ATA6286C

Embedded AVR Microcontroller with LF Receiver and UHF Transmitter

DATASHEET

Features

- 8-bit Atmel® AVR® RISC low-power microcontroller
- Embedded ultra-low-power flash 8-bit AVR (Atmel ATA6289) with on-chip sensitive LF receiver, temperature sensor and integrated UHF transmitter IC
- 8KB of in-system self-programmable Flash memory and 320 Bytes of EEPROM
- 8-bit AVR RISC low-power microcontroller requires typically < 0.5µA sleep current with active interval timer
- 8mA active current requested at 6dBm output power in transmission mode within 432MHz to 448MHz (Atmel ATA5757) frequency range
- ASK and FSK modulation with up to 20Kbaud data rate in Manchester mode
- Programmable 125kHz wake-up receiver channel with typically 2.3µA current consumption in listening mode
- Typically 25mbar ADC resolution (measured with a typical pressure sensor)
- Low-power measurement mode for directly connected capacitive sensors with typically 350µA in 30ms
- Three interfaces for simple capacitive sensors (3pF to 16pF)
- One interface can be configured for motion wake-up (1pF to 4pF)
- Operation voltage 2V to 3.6V for single Li-Cell power supply
- Operating temperature –40°C to +85°C and storage temperature –40°C to +85°C
- Less than 10 external passive components
- QFN 32 (5 x 5) package

Applications

- Active RFID
- Access control

1. Description

The Atmel® ATA6286C is a embedded ultra-low-power AVR 8-bit microcontroller ICs with integrated RF transmission and LF receiving functionality for wake-up purposes in a small QFN32 package.

The RF transmission is based on well-known Atmel IPs for 432MHz to 448MHz (Atmel ATA5757) frequency range with a typical output power up to 6dBm. They are suited for ASK and FSK modulation with up to 20Kbaud data rate in Manchester mode.

The integrated programmable 125kHz LF receiver channel has extremely low current consumption in active listening mode. As a result, the LF receiver is particularly well suited for wake-up purposes.

The programmable AVR 8-bit Flash microcontroller includes 8KB of in-system self-programmable Flash memory and 320 bytes of EEPROM thus allowing the system integrator to install field programmable firmware to meet flexible system requirements on different platforms. The Atmel ATA6286C is configurable to meet extremely low-power requests in sleep mode, measurement mode and transmission mode.

Furthermore, a low-power interval timer and brown-out detector is integrated.

The Atmel ATA6286Cis suited for powering single cell battery applications. Therefore, a single LiMnO2 battery coin cell can supply a whole system.

The AVR 8-bit Flash microcontroller also delivers a dedicated integrated simple capacitive sensor interface, as well as an on-chip calibratable temperature sensor.

Three sensor interfaces are available for capacitive sensing in a range of 3pF to 16pF. In addition, one channel can be configured for motion sensing in a range of 1pF to 4pF.

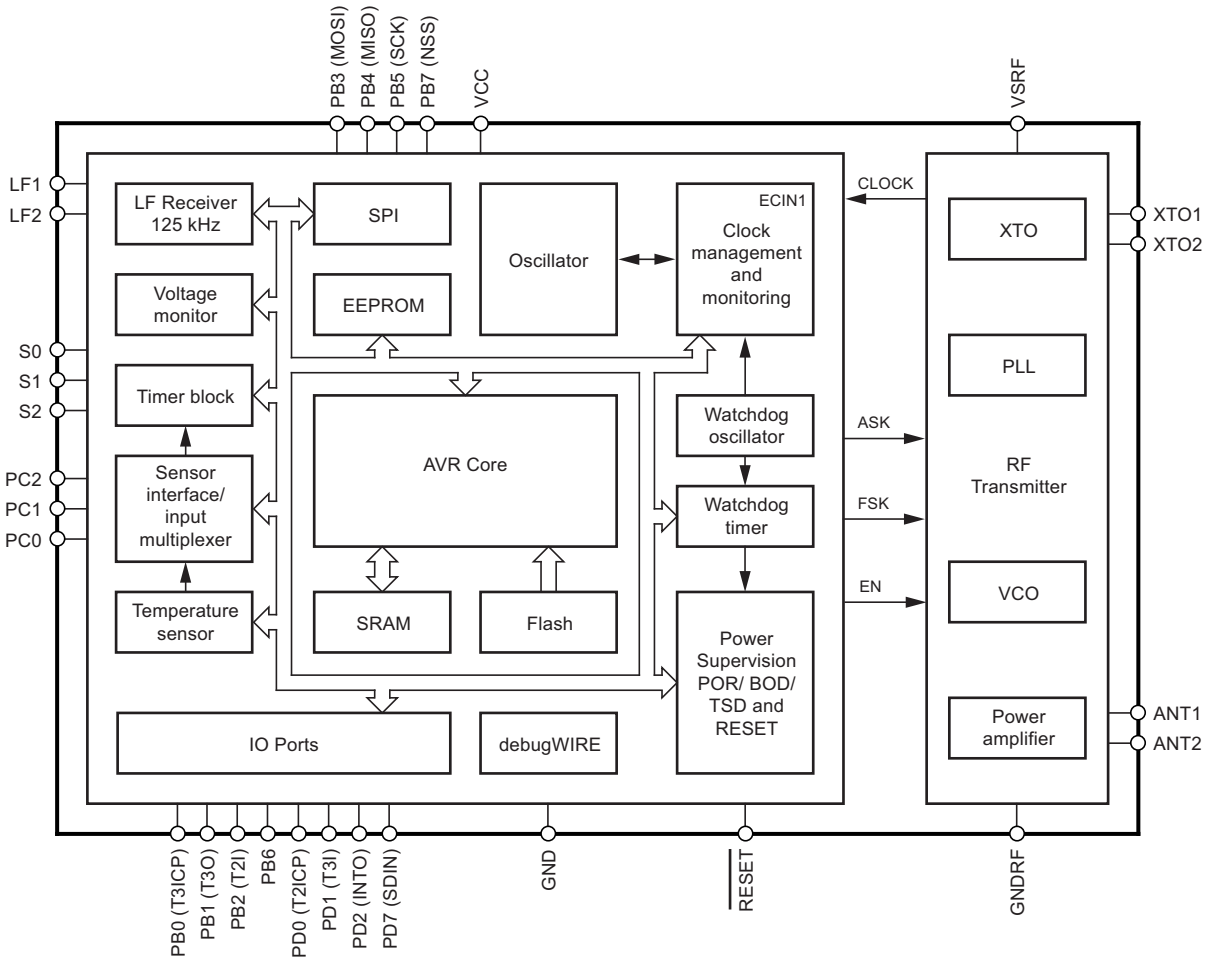
The Atmel ATA6286C is thus proposed for applications requiring very extreme low-power consumption such as active RFID tags with an extended service life.

Owing to the integrated capacitive sensor interface, these ICs may also be used in pressure sensor applications.

2. Overview

2.1 Block Diagram

Figure 2-1. Atmel ATA6286C Block Diagram (MCP)



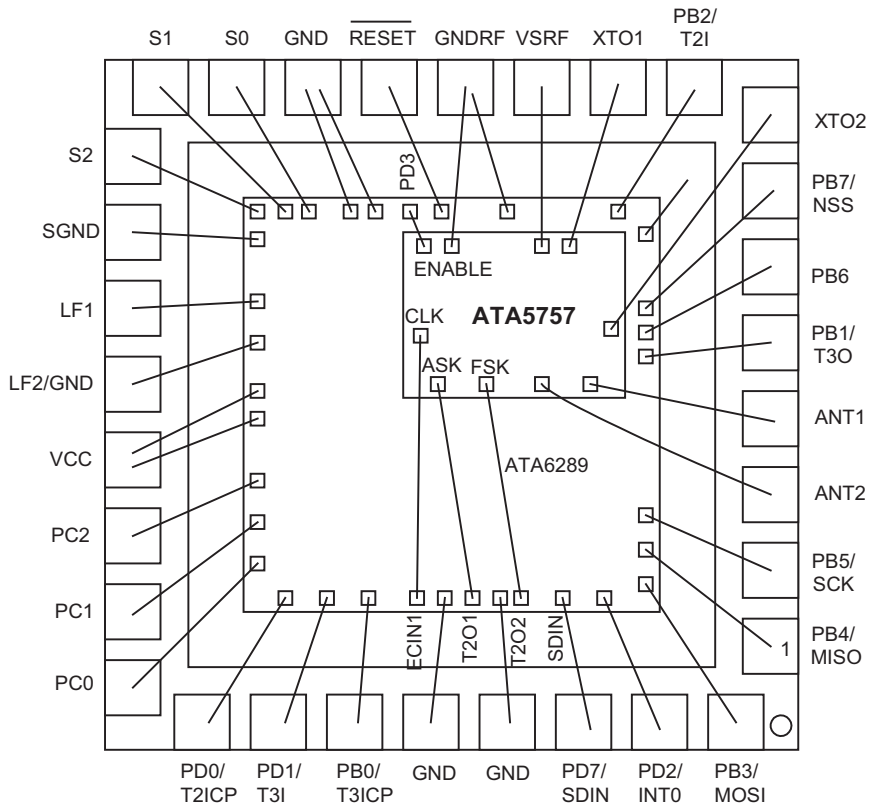
2.2 Bonding Diagram

The Atmel® ATA6286C is a smart RF microtransmitter-based multichip package (MCP). Figure 2-2 on page 4 shows the internal assembly of the MCP. This assembly has two internal dies with four inter-die connections, as shown in Table 2-1.

Table 2-1. Inter-Die Connection Description of the MCP

Atmel ATA6289 Pin	Atmel ATA5757 Pin
PD3 (INT1) – External Interrupt Input 1	EN – Enable Input
PD4 (ECIN1) – External Clock Input 1	CLK – Clock Output Signal
PD5 (T2O1) – Timer2 Modulator Output 1	ASK – Input Signal
PD6 (T2O2) – Timer2 Modulator Output 2	FSK – Input Signal

Figure 2-2. Multichip Package (MCP)



2.3 Pin Configurations

Figure 2-3. Pinout for QFN32 5mm × 5mm Package

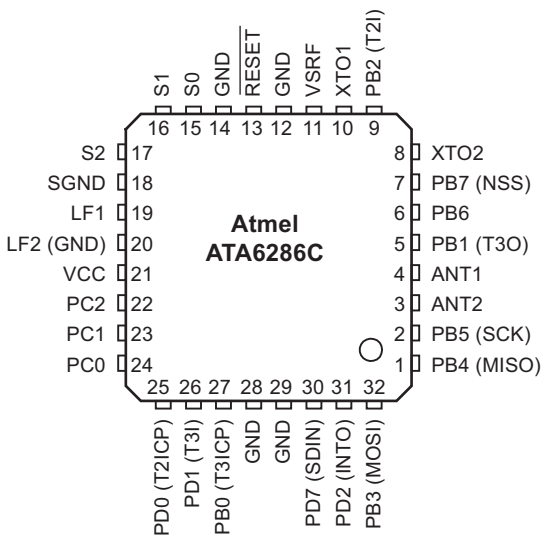


Table 2-2. Pin Description

Pin	Symbol	Alternate Function 1	Alternate Function 2	Function	Comment
1	PB4	MISO	PCINT4	SPI	Port B4
2	PB5	SCK	PCINT5	SPI	Port B5
3	ANT2	-	-	RF antenna 2, emitter of antenna output stage	RF pin
4	ANT1	-	-	RF antenna 1, open collector antenna output	RF pin
5	PB1	T3O	PCINT1	Timer3 output	Port B1
6	PB6	-	PCINT6		Port B6
7	PB7	SS	PCINT7	SPI	Port B7
8	XT02	-	-	Switch for FSK modulation	RF pin
9	PB2	T2I	PCINT2	Timer1, timer2, timer3 external input clock	Port B2
10	XT01	-	-	Connection for crystal	RF pin
11	VSRF	-	-	Power supply voltage for RF	RF pin
12	GNDRF	-	-	Power supply ground for RF	RF pin
13	RESET	debugWIRE	-	Reset input / debugWIRE interface	
14	GND	-	-	Power supply ground	
15	S0	-	-	Sensor input 0 – pressure sensor (cap.)	
16	S1	-	-	Sensor input 1 – sensor (cap.)	
17	S2	-	-	Sensor input 2 – motion sensor (cap.) wake-up	
18	SGND	-	-	Sensor ground	
19	LF1	-	-	LF receiver input 1	
20	LF2 / GND	-	-	LF receiver input 2 internally to GND	
21	V_{CC}	-	-	Power supply voltage (analog + digital)	
22	PC2	-	PCINT10	-	Port C2
23	PC1	CLKO	PCINT9	System clock output	Port C1
24	PC0	ECIN0	PCINT8	External clock input 0	Port C0
25	PD0	T2ICP	PCINT16	Timer2 external input capture	Port D0
26	PD1	T3I	PCINT17	Timer1, timer2, timer3 external input clock	Port D1
27	PB0	T3ICP	PCINT0	Timer3 external input capture	Port B0
28	GND	-	-	Power supply ground	
29	GND	-	-	Power supply ground	
Inter-die[1]	PD3	INT1	PCINT19	External interrupt 1 ⟶ inter-die connection	Port D2
Inter-die[1]	PD4	ECIN1	PCINT20	External clock input 1 ⟶ inter-die connection	Port D4
Inter-die[1]	PD5	T2O1	PCINT21	Timer2 modulator output 1 ⟶ inter-die connection	Port D5
Inter-die[1]	PD6	T2O2	PCINT22	Timer2 modulator output 2 ⟶ inter-die connection	Port D6
30	PD7	SDIN	PCINT23	SSI – serial data input	Port D7
31	PD2	INT0	PCINT18	External interrupt input 0	Port D2
32	PB3	MOSI	PCINT3	SPI	Port B3

Note: 1. Internal inter-die connection of the MCP

2.4 Pin Names

2.4.1 V_{CC}

Supply voltage

2.4.2 GND

Ground

2.4.3 Port B (PB7..0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The port B output buffers have symmetrical drive characteristics with both high-sink and source-current capability. As inputs, port B pins that are pulled low externally will source current if the pull-up resistors are activated. The port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special ATA6289 features as listed in Section 3.12.3.1 “Alternate Functions of Port B” on page 51.

2.4.4 Port C (PC2..0)

Port C is a 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The port C output buffers have symmetrical drive characteristics with both high-sink and source-current capability. As inputs, port C pins that are pulled low externally will source current if the pull-up resistors are activated. The port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port C also serves the functions of various special ATA6289 features as listed in Section 3.12.3.3 “Alternate Functions of Port C” on page 53.

2.4.5 Port D (PD7..0)

Port D is a 8(4)-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PD(6..3) pins are used as internal inter-die connections I/O ports. The port D output buffers have symmetrical drive characteristics with both high-sink and source-current capability. As inputs, port D pins that are pulled low externally will source current if the pull-up resistors are activated. The port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various ATA6289 features as listed in Section 3.12.3.4 “Alternate Functions of Port D” on page 54.

2.4.6 RESET

Reset input. A low level on this pin for longer than the minimum pulse length generates a reset, even if the clock is not running. The minimum pulse length is given in Table 3-13 on page 35. Shorter pulses do not ensure that a reset is generated.

2.4.7 LF (2..1)

Input coil pins for the LF receiver.

2.4.8 S (2..0)

Measuring input pins for external capacitance sensor elements.

2.4.9 ANT(2, 1)

RF antenna pins.

2.4.10 XTO(0, 1)

External crystal pins for the internal RF transmitter IC.

2.5 Disclaimer

Typical values contained in this datasheet are based on simulations and the characterization of other AVR microcontrollers manufactured based on the same process technology. Minimum and maximum values become available after device characterization.

3. AVR Microcontroller ATA6289

3.1 Features

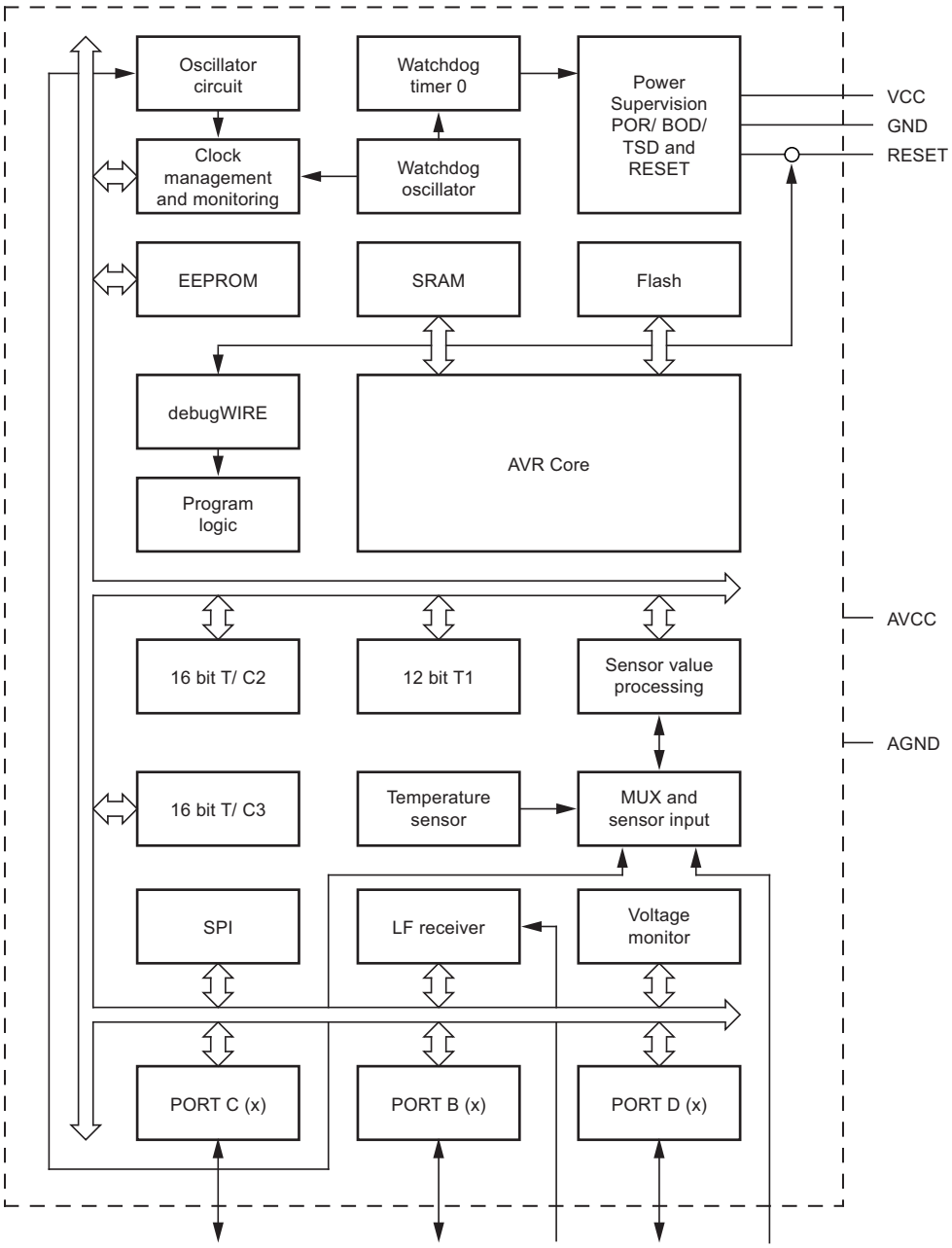
- High performance, extremely low-power Atmel AVR 8-bit microcontroller
- Advanced RISC architecture
  - 131 powerful instructions
  - 32 × 8 general purpose working registers
  - Fully static operation
  - On-chip 2-cycle multiplier
- Non-volatile program and data memories
  - 8KB of in-system self-programmable Flash
  - Optional boot code section with independent lock bits
  - 320 (256 + 64) bytes of EEPROM
  - 512-byte internal SRAM
  - Programming lock for software security
- Peripheral features
  - Programmable watchdog/interval timer with separate internal calibrated extremely low-power oscillator
  - Two 16-bit timer/counter with compare mode, capture mode, and on-chip digital data modulator circuitry
  - Integrated (not calibrated) on-chip temperature sensor with thermal shutdown function
  - Sensor interface for external pressure sensor and motion sensor with wake-up function
  - Highly sensitive 1D LF receiver
  - Programmable voltage monitor
  - System clock management and clock monitoring
  - Master/Slave SPI serial interface
  - Integrated debug-wire-interface
  - Interrupt and wake-up on pin change
- Special microcontroller features
  - Power-on reset and programmable brown-out detection
  - Internal calibrated RC oscillator
  - External and internal interrupt sources
  - Three sleep modes: idle, sensor noise reduction, and power-down
- I/O and package
  - 15 (19) programmable I/O lines
  - QFN32 package, 5mm × 5mm
- Operating voltage
  - 1.9V to 3.6V for ADC and LF receiver
  - 1.8V to 3.6V all other components
- Speed
  - 0 to 2MHz (system clock CLK)
  - 0 to 4MHz (timer clock CLT)
- Temperature range
  - –40°C to +85°C

3.2 Overview

The Atmel® ATA6289 is a CMOS 8-bit microcontroller with extremely low-power consumption based on the Atmel AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATA6289 achieves throughputs approaching 1MIPS per MHz allowing the designer to optimize power consumption versus processing speed.

3.3 Block Diagram

Figure 3-1. Block Diagram of Atmel ATA6289



The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU) allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code-efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The Atmel ATA6289 provides the following features: 8KB of in-system programmable Flash with read-while-write capabilities, 320 (256+64) bytes of EEPROM, 512 bytes of SRAM, 15 (19) general purpose I/O lines, 32 general purpose working registers, on-chip debugging support and programming, three flexible timers/counters, two of them with compare modes, internal and external interrupts, a sensor interface for the external pressure sensor and an acceleration/motion sensor, a programmable watchdog timer with internally calibrated oscillator, an SPI serial port, and three software-selectable power-saving modes.

The device is manufactured using Atmel high-density non-volatile memory technology. On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional non-volatile memory programmer, or by an on-chip boot program running on the Atmel AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the boot Flash section continues to run while the application Flash section is updated, providing true read-while-write operation. By combining an 8-bit RISC CPU with in-system self-programmable Flash on a monolithic chip, the Atmel ATA6289 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The Atmel ATA6289 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

3.4 About Code Examples

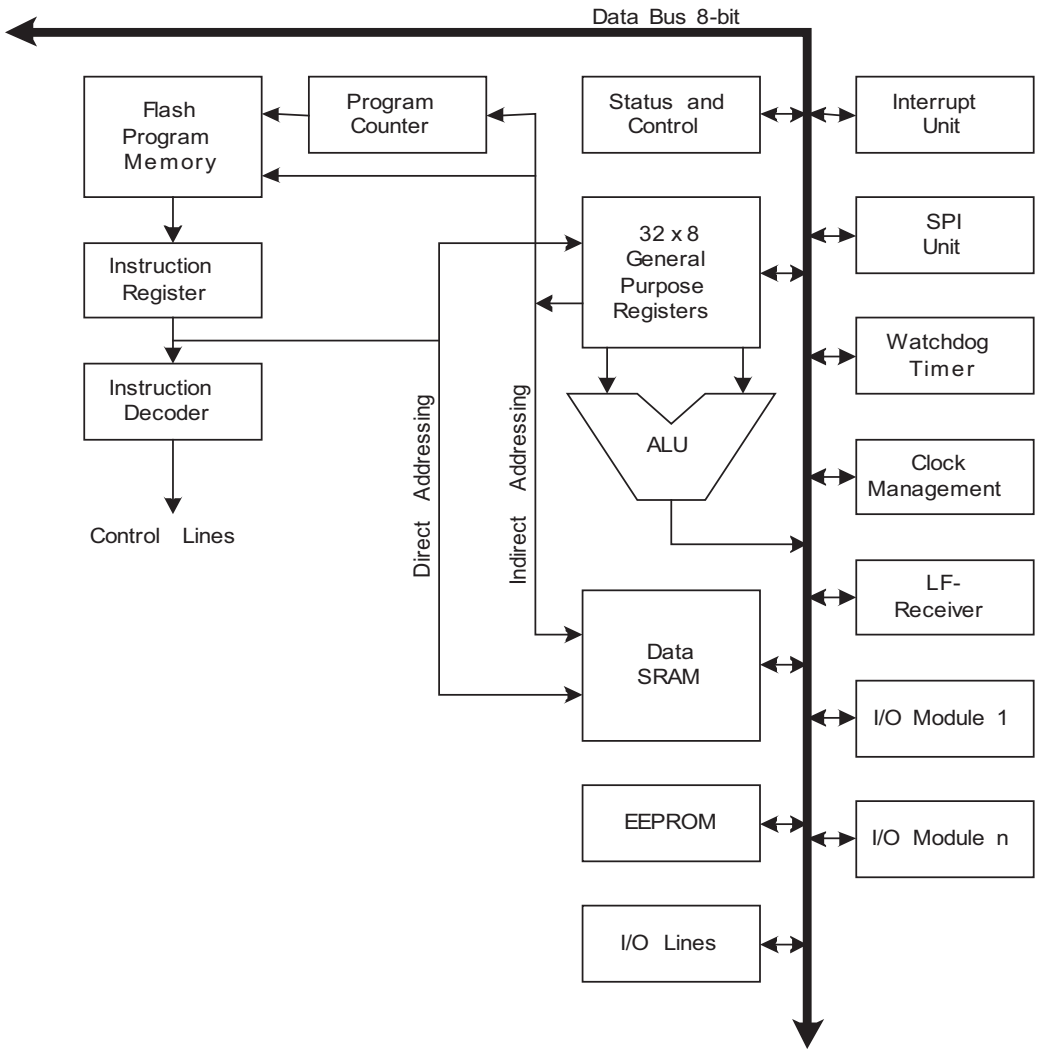
This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part-specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler-dependent. Please refer to the C compiler documentation for more details.

The Atmel AVR Studio® can be used for code development. Please select the Atmel AVR device “ATA6289.”

3.5 Atmel AVR CPU Core

3.5.1 Architectural Overview

Figure 3-2. Block Diagram of the Atmel AVR Architecture



In order to maximize performance and parallelism, the Atmel® AVR® uses Harvard architecture with separate memories and buses for program and data. Instructions in the program memory are executed with single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is in-system reprogrammable Flash memory.

The fast-access register file contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle ALU operation. In a typical ALU operation, two operands are output from the register file, the operation is executed and the result is stored back in the register file—all in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address-register pointers for data space addressing—enabling efficient address calculations. One of these address pointers can also be used as an address pointer to look up tables in the Flash program memory. These added function registers are the 16-bit X, Y and Z registers described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single-register operations can also be executed in the ALU. After an arithmetic operation, the status register is updated to reflect information about the operation outcome.

Program flow is enabled by conditional and unconditional jump and call instructions, allowing the entire address space to be addressed directly. Most Atmel® AVR® instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided into two sections, the boot program section and the application program section. Both sections have dedicated lock bits for write and read/write protection. The Store Program Memory (SPM) instruction that writes into the application Flash memory section must reside in the boot program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the Stack Pointer (SP) in the reset routine (before subroutines or interrupts are executed). The SP is read/write-accessible in the I/O space. The data SRAM can be accessed easily through the five different addressing modes supported in the Atmel AVR architecture.

The memory spaces in the Atmel AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as control registers, SPI and other I/O functions. The I/O memory can be accessed directly, or as the data space locations following those of the register file, 0x20 - 0x5F. In addition, the ATA6289 has extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

3.5.2 ALU – Arithmetic Logic Unit

The high-performance Atmel AVR ALU operates in direct connection with all 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories—arithmetic, logic and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. For a detailed description, see Section 3.22 “Instruction Set Summary” on page 156.

3.5.3 Status Register

The status register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering the program flow in order to perform conditional operations. Note that the status register is updated after all ALU operations, as specified in the instruction set reference. In many cases—this eliminates the need to use the dedicated compare instructions, resulting in faster and more compact code. The status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt.This must be handled by software.

3.5.3.1 The Atmel AVR Status Register (SREG):

Bit	7	6	5	4	3	2	1	0	
	I	**T**	**H**	**S**	**V**	**N**	**Z**	**C**	**SREG**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – I: Global Interrupt Enable
The global interrupt enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared, none of the interrupts are enabled independently of the individual interrupt enable settings. The I bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

Bit 6 – T: Bit Copy Storage
The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T bit as the source or destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied by the BLD instruction into a bit in a register in the register file.

Bit 5 – H: Half Carry Flag
The half carry flag H indicates a half carry in some arithmetic operations. Half carry is useful in BCD arithmetic. See Section 3.22 “Instruction Set Summary” on page 156 for detailed information.

Bit 4 – S: Sign Bit, S = N ⊕ V
The S bit is always exclusive or located between the negative flag N and the two's complement overflow flag V. See Section 3.22 "Instruction Set Summary" on page 156 for detailed information.

Bit 3 – V: Two's Complement Overflow Flag
The two's complement overflow flag V supports two's complement arithmetic. See Section 3.22 "Instruction Set Summary" on page 156 for detailed information.

Bit 2 – N: Negative Flag
The negative flag N indicates a negative result in an arithmetic or logic operation. See Section 3.22 "Instruction Set Summary" on page 156 for detailed information.

Bit 1 – Z: Zero Flag
The zero flag Z indicates a zero result in an arithmetic or logic operation. See Section 3.22 "Instruction Set Summary" on page 156 for detailed information.

Bit 0 – C: Carry Flag
The carry flag C indicates a carry in an arithmetic or logic operation. See Section 3.22 "Instruction Set Summary" on page 156 for detailed information.

3.5.4 General Purpose Register File

The register file is optimized for the Atmel® AVR® enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the register file:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 3-3 shows how the 32 general purpose working registers in the CPU are structured.

Figure 3-3. Atmel AVR CPU General Purpose Working Registers

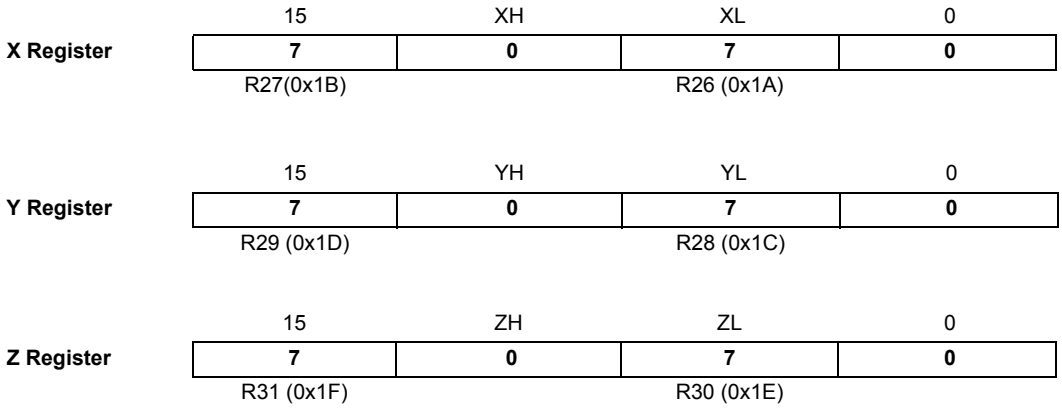
Bit:	7 … 0	Addr.	
General Purpose Working Registers	R0	0x00	
	R1	0x01	
	R2	0x02	
	…		
	R13	0x0D	
	R14	0x0E	
	R15	0x0F	
	R16	0x10	
	R17	0x11	
	…		
	R26	0x1A	X register low byte
	R27	0x1B	X register high byte
	R28	0x1C	Y register low byte
	R29	0x1D	Y register high byte
	R30	0x1E	Z register low byte
	R31	0x1F	Z register high byte

Most of the instructions operating on the register file have direct access to all registers, and most of them are single-cycle instructions. As shown in Figure 3-3, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user data space. Although not physically implemented as SRAM locations, this memory organization provides considerable flexibility in accessing the registers because the X-, Y- and Z-pointer registers can be set to index any register in the file.

3.5.4.1 The X, Y and Z Registers

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y and Z are defined as described in Figure 3-4.

Figure 3-4. The X, Y and Z Registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment and automatic decrement (see the instruction set reference for more information.)

3.5.5 The Stack Pointer

The stack is primarily used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The stack pointer register always points to the top of the stack. Note that the stack is implemented as growing from higher memory locations to lower memory locations. This implies that a stack PUSH command decreases the stack pointer.

The stack pointer points to the data SRAM stack area where the subroutine and interrupt stacks are located. This stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The stack pointer must be set to point above 0x100, preferably RAMEND. The stack pointer is decremented by one when data is pushed onto the stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the stack with a subroutine call or interrupt. The stack pointer is incremented by one when data is popped from the stack with the POP instruction, and it is incremented by two when the return address is popped from the stack with return from subroutine RET or return from interrupt RETI.

The Atmel® AVR® Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation-dependent. Note that the data space in some implementations of the Atmel AVR architecture is so small that only SPL is needed. In this case, the SPH register is not present.

In ATA6289 only SP9 and Sp8 of high byte (SPH) are used.

Bit	15	14	13	12	11	10	9	8	
	SP15	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
	SP7	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
Bit	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	
	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	

3.5.6 Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The Atmel® AVR® CPU is driven by the CPU clock, CLK_{CPU}, directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 3-5 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks and functions per power unit.

Figure 3-5. The Parallel Instruction Fetches and Instruction Executions

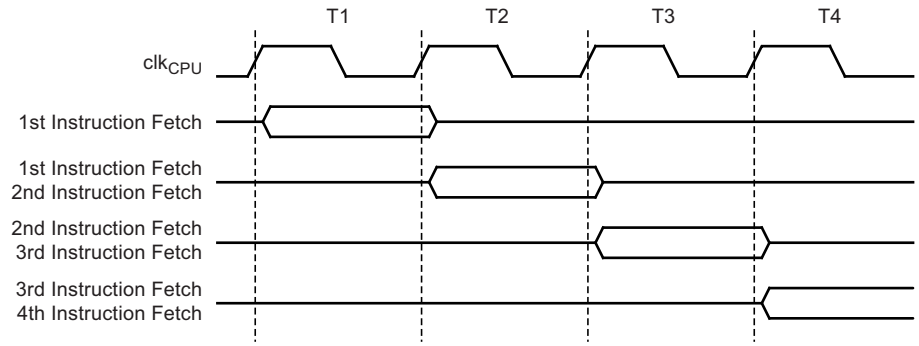
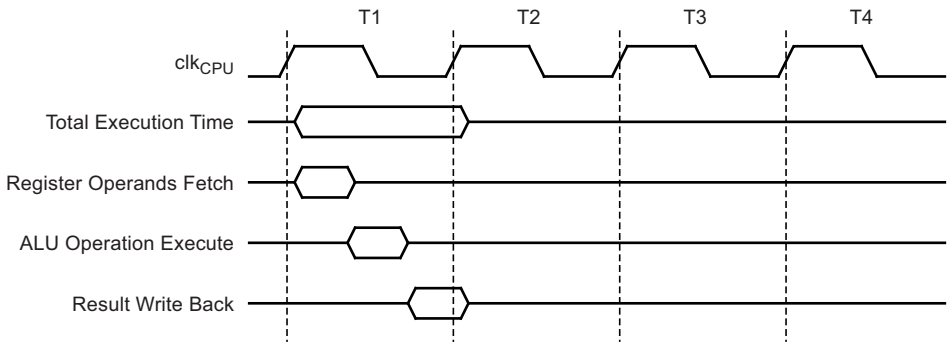


Figure 3-6 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed with the result stored to the destination register.

Figure 3-6. Single-Cycle ALU Operation



3.5.7 Reset and Interrupt Handling

The Atmel AVR provides several different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned unique enable bits which must be written logic one together with the global interrupt enable bit in the status register in order to enable the interrupt. Depending on the program counter value, interrupts may be automatically disabled when boot lock bits BLB02 or BLB12 are programmed. This feature improves software security. See Section 3.19.5.1 “Store Program Memory Control and Status Register – SPMCSR” on page 136 for more details.

By default the lowest addresses in the program memory space are defined as the reset and interrupt vectors. The complete list of vectors is found in Section 3.10 “Interrupts” on page 39. This list also determines the priority levels of the different interrupts. The lower the address the higher the priority level. RESET has the highest priority, followed by INT0—the external interrupt request 0. The interrupt vectors can be moved to the start of the boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). For more information, see Section 3.10 “Interrupts” on page 39. The reset vector can also be moved to the start of the boot Flash section by programming the BOOTRST fuse. See Section 3.19 “Boot Loader Support – Read-While-Write Self-Programming” on page 126 for more details.

When an interrupt occurs, the global interrupt enable I bit is cleared and all interrupts are disabled. The user software can write logic one to the I bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I bit is automatically set when a Return from Interrupt instruction (RETI) is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the interrupt flag. For these interrupts, the program counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, and hardware clears the corresponding interrupt flag. Interrupt flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the interrupt flag is set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the global interrupt enable bit is cleared, the corresponding interrupt flag(s) is set and remembered until the global interrupt enable bit is set, and is then executed by order of priority.

The second type of interrupts triggers as long as the interrupt condition is present. These interrupts do not necessarily have interrupt flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt is not triggered.

When the Atmel AVR exits from an interrupt, it always returns to the main program and executes one more instruction before any pending interrupt is served.

Note that the status register is not automatically stored when entering an interrupt routine, and not restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, interrupts are immediately disabled. No interrupt is executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example

```
inr16, SREG; store SREG value
cli ; disable interrupts during timed sequence
sbiEECR, EEMWE; start EEPROM write
sbiEECR, EEWE
outSREG, r16; restore SREG value (I-bit)
```

C Code Example

```
char cSREG;
cSREG = SREG;/* store SREG value */
/* disable interrupts during timed sequence */
_CLI();
EECR |= (1<<EEMWE); /* start EEPROM write */
EECR |= (1<<EEWE);
SREG = cSREG; /* restore SREG value (I-bit) */
```

When using the SEI instruction to enable interrupts, the instruction following SEI is executed before any pending interrupts, as shown in this example.

Assembly Code Example

```
sei; set Global Interrupt Enable
sleep; enter sleep, waiting for interrupt
; note: will enter sleep before any pending
; interrupt(s)
```

C Code Example

```
_SEI(); /* set Global Interrupt Enable */
_SLEEP(); /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
```

3.5.7.1 Interrupt Response Time

The interrupt execution response for all the enabled Atmel® AVR® interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During these four clock cycle periods, the program counter is pushed onto the stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multicycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the startup time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the program counter (two bytes) is popped back from the stack, the stack pointer is incremented by two, and the I bit in SREG is set.

3.6 Atmel AVR ATA6289 Memories

This section describes the different memories in the ATA6289. The Atmel® AVR® architecture has two main memory spaces, the data memory and the program memory space. In addition, the ATA6289 features an EEPROM memory for data storage. All three memory spaces are linear and regular.

3.6.1 In-System Reprogrammable Flash Program Memory

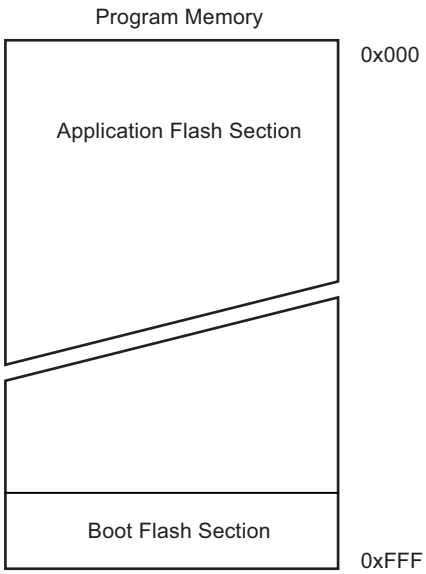
The ATA6289 contains 8KB on-chip in-system reprogrammable Flash memory for program storage. Because all Atmel AVR instructions are 16 bits or 32 bits wide, the Flash is organized as 4K × 16. For software security, the Flash program memory space is divided into two sections, the boot program section and the application program section.

The Flash memory has a service life expectancy of at least 200 write/erase cycles (according to the AECQ100 standard: cycles at room temperature followed by 1,000 hours of High Temperature Operation Lifetime (HTOL) while constantly reading the Flash memory. The ATA6289 Program Counter (PC) is 12 bits wide and thus addresses the 4K program memory locations. The operation of the boot program section and associated boot lock bits for software protection are described in detail in Section 3.19 “Boot Loader Support – Read-While-Write Self-Programming” on page 126. Section 3.21.5 “Serial Downloading” on page 149 contains a detailed description about Flash data serial downloading using the SPI pins.

Constant tables can be allocated within the entire program memory address space (see the Load Program Memory (LPM) instruction description (Table 3-83 on page 151).)

Timing diagrams for instruction fetch and execution are presented in Section 3.5.6 “Instruction Execution Timing” on page 14.

Figure 3-7. Program Memory Map

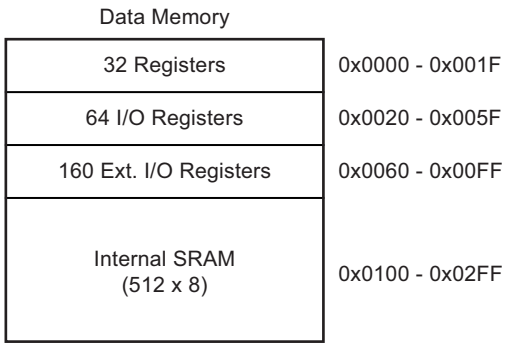


3.6.2 SRAM Data Memory

Figure 3-8 shows how the ATA6289 SRAM memory is organized. The ATA6289 is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in the opcode for the IN and OUT instructions. For the extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used. The lower 768 data memory locations address not only the register file, the I/O memory and extended I/O memory, but also the internal data SRAM. The first 32 locations address the register file, the next 64 location the standard I/O memory. This is followed by 160 locations of extended I/O memory, with the next 512 locations addressing the internal data SRAM. The five different addressing modes for the data memory cover: direct, indirect with displacement, indirect, indirect with pre-decrement and indirect with post-increment. In the register file, registers R26 to R31 feature the indirect addressing pointer registers. The direct addressing reaches the entire data space. The indirect with displacement mode reaches 63 address locations from the base address given by the Y or Z register. When using register indirect addressing modes with automatic pre-decrement and post- increment, the address registers X, Y and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O registers, 160 extended I/O registers, and the 512 bytes of internal SRAM in the ATA6289 are all accessible through all these addressing modes. The register file is described in Section 3.5.4 “General Purpose Register File” on page 12.

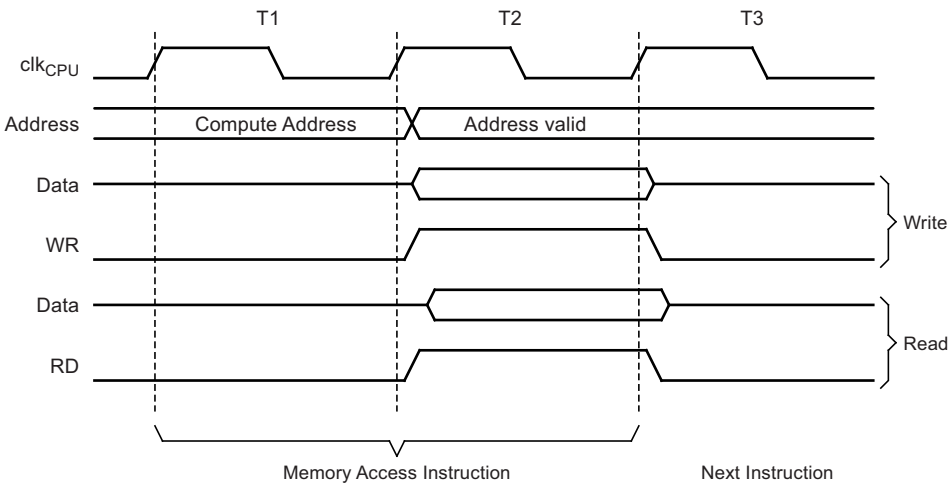
Figure 3-8. Data Memory Map



3.6.2.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two CLK_{CPU} cycles as described in Figure 3-9 on page 17.

Figure 3-9. On-Chip Data SRAM Access Cycles



3.6.3 I/O Memory

The I/O space definition of the ATA6289 is shown in Section 3.21.6 “Register Summary” on page 153. All ATA6289 I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers the value of single bits can be checked using the SBIS and SBIC instructions. Refer to Section 3.22 “Instruction Set Summary” on page 156 for more details. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATA6289 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in opcode for the IN and OUT instructions. For the extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the status flags are cleared by writing a logic one to them. Note that, unlike most other Atmel® AVR®s, the CBI and SBI instructions only operate on the specified bit and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and peripherals control registers are explained in later sections.

3.6.4 General Purpose I/O Registers

The Atmel ATA6289 contains three general purpose I/O registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and status flags. General purpose I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

3.6.4.1 General Purpose I/O Register 2 – GPIOR2

Bit	7	6	5	4	3	2	1	0	
	GPIOR2[7..0]								GPIOR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.6.4.2 General Purpose I/O Register 1 – GPIOR1

Bit	7	6	5	4	3	2	1	0	
	GPIOR1[7..0]								GPIOR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

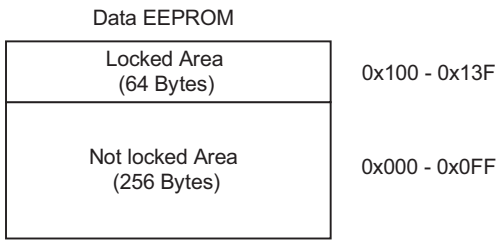
3.6.4.3 General Purpose I/O Register 0 – GPIOR0

Bit	7	6	5	4	3	2	1	0	
	GPIOR0[7..0]								GPIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.6.5 EEPROM Data Memory

The Atmel® ATA6289 contains 320 (256 + 64) bytes of data EEPROM memory. It is organized as a separate data space in which single bytes can be read and written. As shown in Figure 3-10, the EEPROM contains a locked area of 64 bytes. The EEPROM addresses above 0x0FF are only readable if the fuse bit EELOCK is programmed. The EELOCK bit is located in the fuse high byte. For more information, see Table 3-71 on page 138. If the EELOCK bit is unprogrammed (default), the EEPROM area above 0x0FF is also writable. The EEPROM has a service life expectancy of at least 2,000 write/erase cycles (according to the AECQ100 standard: cycles at room temperature followed by 1,000 hours of High Temperature Operation Lifetime (HTOL) while constantly reading the Flash memory). The access between the EEPROM and the CPU is described in the following, with information about the EEPROM address registers, the EEPROM data register and the EEPROM control register.

Figure 3-10. EEPROM Map



3.6.5.1 EEPROM Read/Write Access

The EEPROM address registers are accessible in the I/O space. The write access time for the EEPROM is given in Table 3-1 on page 20. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies, VCC is likely to rise or fall slowly on power-up/down. For a period of time this causes the device to run at a voltage lower than specified as minimum for the clock frequency used. See Section 3.6.5.5 “Preventing EEPROM Corruption” on page 23 for more information on how to avoid problems when encountering these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. For more information, see Section 3.6.5.4 “The EEPROM Control Register – EECR” on page 20.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

3.6.5.2 The EEPROM Address Register – EEARH, EEARL

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	-	**EEAR8**	**EEARH**
	EEAR[7..0]								**EEARL**
Bit	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	

Bits 15..9 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bits 8..0 – EEAR7..0: EEPROM Address
The EEPROM address register (EEAR) specifies the EEPROM address in the 320(256)-byte EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 319(255). The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

3.6.5.3 The EEPROM Data Register – EEDR

Bit	7	6	5	4	3	2	1	0	
	EEDR[7..0]								EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..0 – EEDR7.0: EEPROM Data
For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

3.6.5.4 The EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
	-	-	EEPM1	EEPM0	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	X	X	0	0	X	0	

Bits 7..6 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bits 5, 4 – EEPM1 and EEPM0: EEPROM Programming Mode Bits
The EEPROM programming mode bits setting defines which programming action is triggered when writing EEWE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the erase and write operations into two different operations. The programming times for the different modes are shown in Table 3-1. While EEWE is set, any write command to EEPMn is ignored. During reset, the EEPMn bits are reset to 0b00 unless the EEPROM is busy programming.

Table 3-1. EEPROM Mode Bits

EEPM1	EEPM0	Programming Time	Operation
0	0	3.4ms	Erase and write in one operation (atomic operation)
0	1	1.8ms	Erase only
1	0	1.8ms	Write only
1	1	-	Reserved for future use

Bit 3 – EERIE: EEPROM Ready Interrupt Enable
Writing EERIE to one enables the EEPROM ready interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM ready interrupt generates a constant interrupt when EEWE is cleared.

Bit 2 – EEMWE: EEPROM Master Write Enable
The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set, setting EEWE within four clock cycles writes data to the EEPROM at the address selected. If EEMWE is zero, setting EEWE has no effect. When EEMWE has been written to one by the software, the hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for an EEPROM write procedure.

Bit 1 – EEWE: EEPROM Write Enable

The EEPROM Write Enable (EEWE) signal is the write strobe to the EEPROM. When the address and data are correctly set up, the EEWE bit must be written to one to write the value into the EEPROM. The EEMWE bit must be written to one before a logic one is written to EEWE; otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (steps 3 and 4 can be done in any order):

- Wait until EEWE becomes zero.
- Wait until SELFPRGEN in the SPMCSR register becomes zero.
- Write the new EEPROM address to the EEAR register (optional).
- Write the new EEPROM data to the EEDR register (optional).
- Write a logic one to the EEMWE bit while writing a zero to the EEWE bit in the EECR register.
- Within four clock cycles after setting EEMWE, write a logic one to EEWE.

The EEPROM cannot be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a boot loader allowing the CPU to program the Flash. If the Flash is never updated by the CPU, step 2 can be omitted. See Section 3.19 “Boot Loader Support – Read-While-Write Self-Programming” on page 126 for more information about boot programming.

Caution: An interrupt between step 5 and step 6 makes the write cycle fail because the EEPROM master write enable times out. If an interrupt routine accessing the EEPROM interrupts another EEPROM access, the EEAR or EEDR register is modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt flag cleared during all the steps to avoid these problems.

When the write access time has elapsed, the EEWE bit is cleared by the hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE has been set, the CPU is halted for two cycles before the next instruction is executed.

Bit 0 – EERE: EEPROM Read Enable

The EEPROM Read Enable (EERE) signal is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction with the requested data becoming available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed. The user should poll the EEWE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM nor change the EEAR register.

The calibrated oscillator is used for timing the EEPROM accesses. Table 3-2 lists the typical programming time for EEPROM access from the CPU.

Table 3-2. EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles	Typ. Programming Time
EEPROM write (from CPU)	67 584	8.5ms

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (e.g., by disabling interrupts globally) so that no interrupts occur during execution of these functions. The examples also assume that no Flash boot loader is present in the software. If there is such code present, the EEPROM write function must also wait for any ongoing SPM command to finish.

Assembly Code Example

```
EEPROM_write:
; Wait for completion of previous write
sbic EECR,EEWE
rjmp EEPROM_write
; Set up address (r18:r17) in address register
out EEARH, r18
out EEARL, r17
; Write data (r16) to Data Register
out EEDR,r16
; Write logical one to EEMWE
sbi EECR,EEMWE
; Start eeprom write by setting EEWE
sbi EECR,EEWE
ret
```

C Code Example

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
/* Wait for completion of previous write */
while(EECR & (1<<EEWE))
;
/* Set up address and Data Registers */
EEAR = uiAddress;
EEDR = ucData;
/* Write logical one to EEMWE */
EECR |= (1<<EEMWE);
/* Start eeprom write by setting EEWE */
EECR |= (1<<EEWE);
}
```

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts occur while these functions are executed.

Assembly Code Example

```
EEPROM_read:
; Wait for completion of previous write
sbic EECR,EEWE
rjmp EEPROM_read
; Set up address (r18:r17) in address register
out EEARH, r18
out EEARL, r17
; Start eeprom read by writing EERE
sbi EECR,EERE
; Read data from Data Register
in r16,EEDR
ret
```

C Code Example

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
/* Wait for completion of previous write */
while(EECR & (1<<EEWE))
;
/* Set up address register */
EEAR = uiAddress;
/* Start eeprom read by writing EERE */
EECR |= (1<<EERE);
/* Return data from Data Register */
return EEDR;
}
```

3.6.5.5 Preventing EEPROM Corruption

During periods of low V_{CC}, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board-level systems using EEPROM, and the same design solutions should be applied.

EEPROM data corruption can be caused by two situations if there is insufficient voltage. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the Atmel® AVR® $\overline{\text{RESET}}$ active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-Out Detector (BOD). If the detection level of the internal BOD does not match the detection level needed, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation is completed, provided the power supply voltage is sufficient.

3.7 Clock Generation

3.7.1 Clock Module

The ATA6289 contains a clock module with two internal oscillator types:

FRC: Fast running, programmable and calibrated RC oscillator (1MHz/4MHz ±10%)

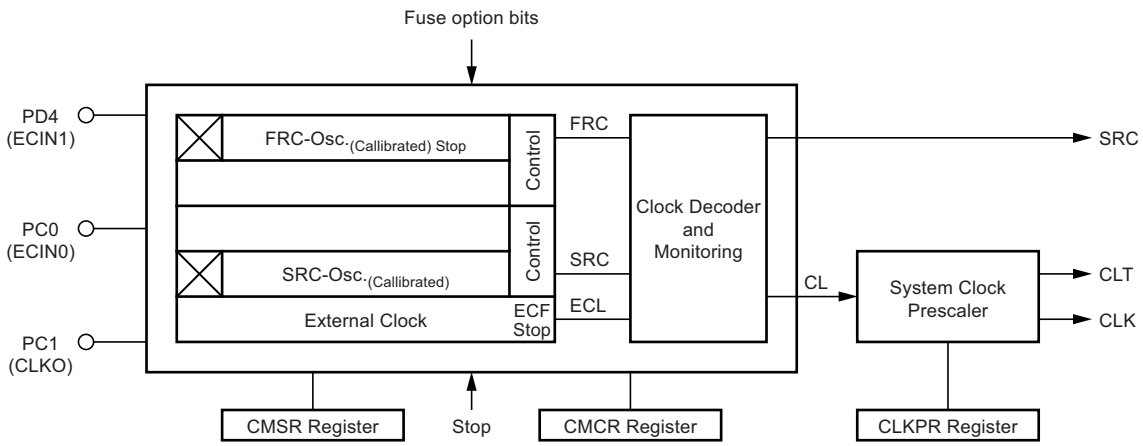
SRC: Slow running and calibrated RC oscillator (90kHz ±10%)

The PC1/ECIN0 pin and PD4/ECIN1 pin can be used as input for two different external clocks and the PC1/CLKO pin as output for the divided system clock. All of these oscillator types and external input clocks can be selected to generate the system Clock (CLK). A special feature of the clock management is the capability to switch between these different clock sources at run time. This new feature offers the advantage that the controller can now start operation after the wake-up signal with the calibrated internal RC oscillator and can switch to an external clock as its system clock. A synchronization stage avoids clock periods of insufficient length if the clock source or the clock speed is changed. If an external input clock is selected, a supervisor circuit monitors the external input and automatically switches to an internal RC oscillator clock if the external clock source fails. The ECF bit indicates the condition of the external input clock monitoring circuit in the CMSR register. If the accessory interrupt enable is set, the corresponding monitoring interrupt is executed.

In applications that do not require exact timing, it is possible to use the fully integrated RC oscillators. The center frequency tolerance of both RC oscillators can be calibrated by VCC = 3V/25°C within ±1% accuracy. The SRC and the timer0 can work together as an ultra-low-power watchdog/interval timer stage.

The clock module is programmable via software with the Clock Management Control Register (CMCR) and the Clock Prescaler Register (CLKPR). The required oscillator configuration can be selected with the CMM[1..0] bits in the CMCR. A system clock prescaler contains a programmable 7-bit divider stage. This stage subdivides the system clock by setting the CLKPR and allows the adjustment of the system Clock speed (CLK) and also the Timer Clock speed (CLT). This can be used with all clock source options and affects the clock frequency of the CPU, with all synchronous peripherals. $CLK_{I/O}$, CLK_{CPU} and CLK_{Flash} divided by a factor shown in Table 3-10 on page 31.

Figure 3-11. Clock Module Unit



3.7.1.1 External Clock Monitor

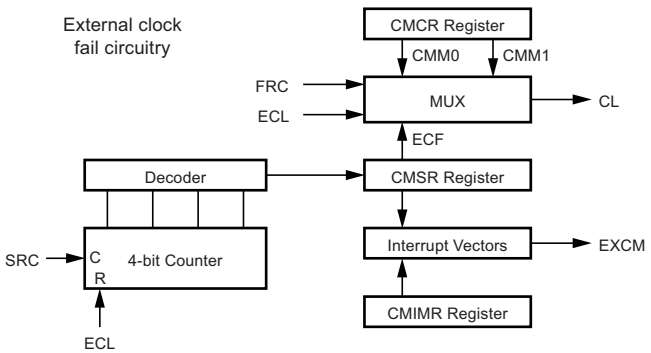
If an external clock is used as the system clock, internal clock monitor circuitry is activated. If the external clock fails for a given time, the ECF bit is set in the Clock Management Status Register (CMSR). After an external clock fail is detected, the system uses the internal RC Oscillator (FRC) as the system clock by switching the CCS bit in the CMCR to zero.

The external clock monitor circuit uses the internal SRC oscillator (90kHz) as the clock source for a 4-bit counter. If the external clock does not reset the internal 4-bit counter periodically, a counter value is reached which triggers the external clock fail bit ECF. For more information, see Figure 3-12.

A typical time value for the external clock fail detection is 100µs. Therefore the minimum external clock frequency is limited to typically 10kHz.

An external frequency < 10kHz forces a clock fail reset.

Figure 3-12. External Clock Fail Circuitry



3.7.1.2 Clock Management Control Register – CMCR

Bit	7	6	5	4	3	2	1	0	
	CMCCE	**-**	**ECINS**	**CCS**	**CMONEN**	**SRCD**	**CMM1**	**CMM0**	**CMCR**
Read/Write	RW	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – CMCCE: Clock Management Control Change Enable Bit
The CMCCE bit must be written to logic one to enable change of the CMMn bits, SRCD bit, CMONEN, CCS bit, and ECINS bit. The CMCCE bit is only updated when the other bits in CMCR are simultaneously written to zero. CMCCE is cleared by the hardware four cycles after it is written or when CMMn, CCS and ECINS bits are written. Rewriting the CMCCE bit within this time-out period neither extends the time-out period nor clears the CMCCE bit.

Bit 6 – Res: Reserved Bit
This bit is a reserved bit on the ATA6289 and is always read as zero.

Bit 5 – ECINS: External Clock Input Select Bit
This bit selects one of the two external clock input PC0(ECIN0) or PD4(ECIN1). The ECINS bit must be written to logic one to enable the ECIN1 clock input, and if the ECINS bit is written to logic zero, then ECIN0 clock input is enabled. The ECINS bit should be only changed if the CCS bit has been cleared. If the CCS bit is set to zero, the internal FRC is activated during bit change for synchronization reasons.

Table 3-3. External Clock Input Select Bit Description

ECINS	Description
0	PC0 ––> External input clock 0 (ECIN0)
1	PD4 ––> External input clock 1 (ECIN1)

Bit 4 – CCS: Core Clock Select Bit
This bit selects from the FRC oscillator clock and all other clock sources. The CCS bit must be written to logic one to enable the mode selected with the CMM[1..0] bits. If the CCS bit is written to logic zero, the FRC oscillator clock is enabled. When the CCS bit is logic one, the CMM[1..0] bits in CMCR must not be changed. After external clock fail detection, the CCS bit is written to zero.

Table 3-4. Core Clock Select Bit Description

CCS	Description
0	The FRC oscillator generates CL
1	The SRC oscillator or an External Clock source (ECL) generates depending on the setting of the CMM[1..0] bits.

Bit 3 – CMONEN: Clock Monitoring Enable
This bit controls clock monitoring. The CMONEN bit must be written to logic one to enable clock monitoring. If the CMONEN bit is written to logic zero, clock monitoring is always disabled.

Bit 2 – SRCD: Slow RC oscillator (SRC) Disable Bit
This bit controls the SRC oscillator used as the clock source for the watchdog (also called WDRC). The SRCD bit must be written to logic one to disable (stop) the SRC, and if the SRCD bit is written to logic zero, the SRC is always enabled (running). The SRC oscillator cannot be disabled if the fuse bit WDRCON is programmed.

Bits 1..0 – CMM1..0: Clock Management Mode Bits 1 - 0
These bits select the input clock source (CL) of the system clock prescaler. Bits CMM1 and CMM0 are not affected by an external clock fail. Before changing the CMM1..0 bits, CCS must first be cleared. After the CMM1..0 bits have been changed, CCS can be set to logic one.

Table 3-5. Clock Source of the System Clock Prescaler Select Bit Description

Mode	Clock Source for System Clock Prescaler (CL)			
	CMM1	CMM0	CCS = 0	CCS = 1
0	0	0	FRC	Reserved
1	0	1	FRC	Reserved
2	1	0	FRC	SRC
3	1	1	FRC	ECL

3.7.1.3 Clock Management Status Register – CMSR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	ECF	CMSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7 to 1 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 0 – ECF: External Clock Input Flag Bit
This bit is set if the clock monitoring circuit detects a breakdown of the selected external input clock (ECIN0 or ECIN1). ECF is automatically cleared when the clock monitoring interrupt vector is executed. Alternatively, ECF can be cleared by writing a logic one to its bit location.

3.7.1.4 Clock Management Interrupt Mask Register – CMIMR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	ECIE	CMIMR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

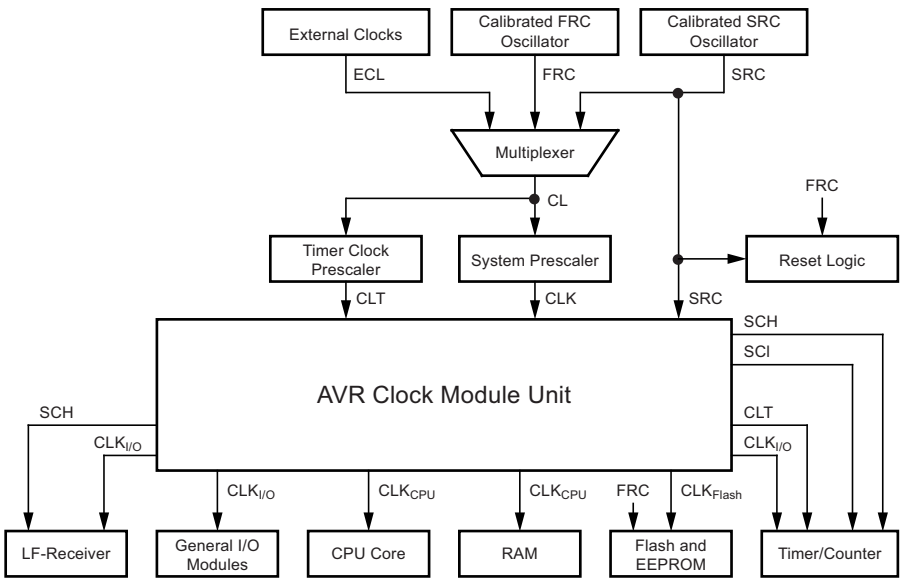
Bits 7 to 1 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 0 – ECIE: External Clock Input Interrupt Enable Bit
Writing ECIE to one enables the clock monitoring interrupt vector if the I bit in SREG is set. Writing ECIE to zero disables the interrupt. The corresponding interrupt vector is executed when the ECF flag, located in CMSR, is set.

3.7.2 Clock Systems and Their Distribution

Figure 3-13 presents the principal clock systems in the Atmel® AVR® and their distribution. All of the clocks need not be active at any given time. In order to reduce power consumption, the clocks for modules not in use can be halted by using different sleep modes described in Section 3.8 “Power Management and Sleep Modes” on page 32. The clock systems are described in detail below.

Figure 3-13. Clock Distribution



System Clock Prescaler Output – CLK
The system clock prescaler output signal (CLK) is used as clock sources for the microcontroller and affects the clock frequency of the CPU and all synchronous peripherals with $CLK_{I/O}$, CLK_{CPU}, and CLK_{Flash} divided by a factor shown in Table 3-10 on page 31.

CPU Clock – CLK_{CPU}
The CPU clock is routed to parts of the system concerned with operation of the Atmel AVR core. Examples of such modules are the general purpose register file, the status register and the data memory holding the stack pointer. Halting the CPU clock keeps the core from performing general operations and calculations.

I/O Clock – $CLK_{I/O}$
The I/O clock is used by the majority of the I/O modules, such as timers/counters, SPIs and ports. The I/O clock is also used by the external interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected, even if the I/O clock is halted.

Flash Clock – CLK_{FLASH}
The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

Timer Clock Prescaler Output – CLT
The timer clock allows the asynchronous timer3/counter3 to be clocked with a clock faster than the I/O clock ($CLK_{I/O}$). The timer clock is usually active simultaneously with the CPU clock.

Calibrated Internal Low-Level Slow Frequency RC Oscillator Clock (90kHz) – SCL
The slow asynchronous timer clock (SCL) supplies only the watchdog/interval timer. This allows the watchdog/interval timer to work when the device is in sleep mode.

Calibrated Internal High-Level Slow Frequency RC Oscillator Clock (90kHz) – SCH
The slow asynchronous timer clock (SCH) is a high supply voltage output clock of the slow frequency calibrated internal RC oscillator (SRC).

3.7.3 Clock Sources

The device has the following clock source options, selectable by the clock module unit. This special feature of the clock management is capability of switching between these different clock sources at run time.

3.7.3.1 Default Clock Source

The device is shipped with an internal FRC oscillator at 4MHz and with the fuse CKDIV8 programmed, resulting in a 0.5MHz system clock. The startup time is set to maximum and time-out period enabled (FRCFS = “0,” SUT = “10,” CKDIV8 = “0”). The default setting ensures that all users can make their desired clock source setting using any available programming interface.

3.7.3.2 Clock Startup Sequence

Any clock source needs a sufficient V_{CC} to start to oscillate and a minimum number of oscillation cycles before it can be considered stable.

To ensure sufficient V_{CC}, the device issues an internal reset with a time-out delay (t_{TOUT}) after the device reset has been released by all other reset sources. Section 3.9 “System Control and Reset” on page 34 describes the start conditions for the internal reset. The delay (t_{TOUT}) is timed from the SRC oscillator and the number of cycles in the delay is set by the SUTx and FRCFS fuse bits. The selectable delays are shown in Table 3-6. The frequency of the SRC oscillator depends on whether the calibration data value from the oscillator is loaded into the calibration register.

Table 3-6. Number of SRC Oscillator Cycles

After Power-On Reset Typ. Time-Out	After All Other Resets Typ. Time-Out (V_{CC} = 3.0V)	Number of Cycles
10ms	5.7ms	512
140ms	91ms	8192

The main purpose of the delay is to keep the Atmel® AVR® in reset until it is supplied with minimum V_{CC}. The delay does not monitor the actual voltage and it is required to select a delay longer than the VCC rise time. If this is not possible, an internal or external brown-out detection circuit should be used. A BOD circuit ensures sufficient VCC before it releases the reset so that the time-out delay can be disabled. Disabling the time-out delay without utilizing a BOD circuit is not recommended.

The oscillator must oscillate for a minimum number of cycles before the clock is considered stable. An internal ripple counter monitors the oscillator output clock while keeping the internal reset active for a given number of clock cycles. The reset is then released and the device starts to execute.

The startup sequence for the clock includes both the time-out delay and the startup time when the device starts up from reset. When starting up from power-down mode, VCC is assumed to be at a sufficient level and only the startup time is included (see Table 3-7).

Table 3-7. Power-Down and Reset Delays

Power Conditions	Startup Time from Sensor Noise Reduction and Power-Down	Additional Delay from Reset (POR), V_{CC} = 3.0V	SUT1..0
Reserved			00
Fast rising power	6CLK	14CLK(1) + 5.7ms (10ms)	01
Slowly rising power	6CLK	14CLK(1) + 91ms (140ms)	10 (default)
Reserved			11

Note: 1. CLK = clock from prescaler output

3.7.4 Calibrated Internal Fast RC Oscillator (FRC)

The calibrated internal FRC oscillator has two selectable fundamental speed frequencies (1MHz/4MHz) which are determined by the FRCFS fuse. The device is shipped with the CKDIV8 fuse programmed. For more information, see Section 3.7.7 “System Clock Prescaler (CLK)” on page 30. During reset, the hardware loads the calibration byte into the FRCCAL register and by doing so automatically calibrates the FRC oscillator. Over a supply range of 1.9V to 3.6V and within a temperature range of –40°C to +85°C, the calibration results in a frequency of 1MHz/4MHz ±10% (ensured by final test). By changing the FRCCAL register (typical value, not measured in final test) the oscillator can be calibrated for any frequency within the range of 0.9MHz to 1.1MHz/3.6MHz to 4.4MHz with a ±1% accuracy for a fixed supply voltage and temperature.

3.7.4.1 Fast Frequency RC – Oscillator Calibration Register – FRCCAL

Bit	7	6	5	4	3	2	1	0	
	-	-	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0	FRCCAL
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7, 6 – Res: Reserved Bits
These bits are reserved and are always read as zero.

Bits 5..0 – FCAL: Fast Frequency RC Oscillator Calibration Value
The oscillator calibration register is used to trim the calibrated internal FRC oscillator to remove process variations from the oscillator frequency. The factory-calibrated value is automatically written to this register during chip reset, giving an oscillator frequency of 1MHz/4MHz. The application software can write this register to change the oscillator frequency. The oscillator can be calibrated to any frequency in the range of 0.9MHz to 1.1MHz/3.6MHz to 4.4MHz within 1% accuracy for a fixed supply voltage and temperature. Calibration outside that range is not guaranteed.

Note that this oscillator is used to time EEPROM and Flash write accesses, and these write times are affected accordingly.

3.7.5 Calibrated Internal Slow RC Oscillator (SRC)

The calibrated internal SRC oscillator is an ultra-low-power oscillator providing a clock of 90kHz. During reset, hardware loads the calibration byte into the SRCCAL register, thus automatically calibrating the SRC oscillator. Over a supply range of 1.9V to 3.6V and within a temperature range of –40°C to +85°C the calibration results in a frequency of 90kHz ±10% (ensured by final test). By changing the SRCCAL register, the oscillator can be calibrated to any frequency in the range of 81kHz – 99kHz within ±1% accuracy (typical value, not measured in final test) for a fixed supply voltage and temperature. This oscillator can be used as a watchdog oscillator, interval timer, start measurement timer for motion sensor, reset time-out, and also as system clock.

3.7.5.1 Slow Frequency RC – Oscillator Calibration Register – SRCCAL

Bit	7	6	5	4	3	2	1	0	
	SCAL7	**SCAL6**	**SCAL5**	**SCAL4**	**SCAL3**	**SCAL2**	**SCAL1**	**SCAL0**	**SRCCAL**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..0 – SCAL7..0: Slow Frequency RC Oscillator Calibration Value
The oscillator calibration register is used to trim the calibrated internal FRC oscillator to remove process variations from the oscillator frequency. The factory-calibrated value is automatically written to this register during chip reset, resulting in an oscillator frequency of 90kHz. The application software can write this register to change the oscillator frequency. The oscillator can be calibrated to any frequency in the range of 81kHz to 99kHz within 1% accuracy for a fixed supply voltage and temperature. Calibration outside that range is not guaranteed.

The SCAL7 bit determines the range of operation for the oscillator. Setting this bit to 0 gives the lowest frequency range, setting this bit to 1 achieves the highest frequency range. The two frequency ranges are overlapping, in other words a setting of SRCCAL = 0x7F results in a higher frequency than SRCCAL = 0x80.

3.7.6 Clock Output Buffer

The device can output the system clock on the PC1/CLKO pin. The CKOUT fuse bit has to be programmed to enable the output. This mode is suitable when the chip clock is used to drive other circuits on the system. The clock is also output during reset and normal operation of the I/O pin is overridden when the fuse is programmed. Any clock source, including the internal RC oscillators, can be selected when the clock is output on CLKO. If the system clock prescaler is used, it is the divided system clock that is output.

Table 3-8. Device Clock Output Select Description(1)

Device Clock Output	CKOUT Fuse	Clock Name	Description
PC1(CLKO)	1	CLK	Disabled
	0	CLK	Enabled

Note: 1. For all fuses “1” means unprogrammed while “0” means programmed

3.7.7 System Clock Prescaler (CLK)

The Atmel® ATA6289 has a system clock prescaler. For more information, see Figure 3-14 on page 32. The system clock can be divided by setting the CLKPR register (see Section 3.7.8.1 “Clock Prescaler Register – CLPR” on page 30). This feature can be used to decrease the system clock frequency and the power consumption when the requirement for processing power is low. The Input Clock (CL) for the prescaler is selectable by the clock module unit and affects the CPU clock frequency and all synchronous peripherals. $CLK_{I/O}$, CLK_{CPU} and CLK_{Flash} are divided by a factor shown in Table 3-10 on page 31.

When switching between prescaler settings, the system clock prescaler ensures that no glitches occur in the clock system. It also ensures that no intermediate frequency exceeds either the clock frequency corresponding to the previous setting or the clock frequency corresponding to the new setting. The ripple counter that implements the prescaler runs at the frequency of the undivided clock, which may be faster than the CPU’s clock frequency. It is therefore not possible to determine the state of the prescaler—even if it were readable—and the exact time it takes to switch from one clock division to the other cannot be exactly predicted. A special write procedure must be followed to change the CLKPS[2..0] bits to avoid unintentional changes of clock frequency:

1. Write the Clock Prescaler Change Enable (CLPCE) bit to one and all other bits in CLKPR to zero.
2. Within four cycles, write the desired value to CLKPS[2..0] while writing a zero to CLPCE.

Interrupts must be disabled when changing the prescaler setting to make sure the write procedure is not interrupted.

3.7.8 Timer Clock Prescaler (CLT)

The Atmel ATA6289 has a timer clock prescaler (Figure 3-14 on page 32) and the timer clock can be divided by setting the CLKPR register. For more information, see Section 3.7.8.1 “Clock Prescaler Register – CLPR” on page 30. This feature can be used to decrease the timer clock frequency. The Input Clock (CL) for the prescaler is selectable by the clock module unit and affects the clock frequency of the timers, which are divided by a factor shown in Table 3-9 on page 31. When switching between prescaler settings, the timer clock prescaler ensures that no glitches occur in the clock system. It also ensures that no intermediate frequency is higher than neither the clock frequency corresponding to the previous setting, nor the clock frequency corresponding to the new setting. The ripple-counter that implements the prescaler runs at the frequency of the undivided clock, which may be faster than the timer's clock frequency. Hence, it is not possible to determine the state of the prescaler—even if it were readable, and the exact time it takes to switch from one clock division to the other cannot be exactly predicted. To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLTPS[2..0] bits:

1. Write the Clock Prescaler Change Enable (CLPCE) bit to one and all other bits in CLKPR to zero.
2. Within four cycles, write the desired value to CLTPS[2..0] while writing a zero to CLPCE.

Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted.

3.7.8.1 Clock Prescaler Register – CLPR

Bit	7	6	5	4	3	2	1	0	
	CLPCE	**-**	**CLTPS2**	**CLTPS1**	**CLTPS0**	**CLKPS2**	**CLKPS1**	**CLKPS0**	**CLPR**
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – CLPCE: CLock Prescaler Change Enable Bit
The CLPCE bit must be written to logic one to enable change of the CLTPS[2..0] bits and CLKPS[2..0] bits. The CLPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLPCE is cleared by hardware four cycles after it is written or when CLTPS[2..0] bits and CLKPS[2..0] bits are written. Rewriting the CLPCE bit within this time-out period does neither extend the time-out period, nor clear the CLPCE bit.

Bit 6 – Res: Reserved Bit
This bit is a reserved bit on the ATA6289 and is always read as zero.

Bits 5..3 – CLTPS2..0: CLock Timer Prescaler Select Bits 2 - 0
These bits select the timer output clock (CLT) of the timer clock prescaler as shown in Table 3-9.

Table 3-9. Timer Clock Prescaler Select Bit Description

CLTPS2	CLTPS1	CLTPS0	Description
0	0	0	disable
0	0	1	1
0	1	0	2
0	1	1	4
1	0	0	8
1	0	1	16
1	1	0	32
1	1	1	64

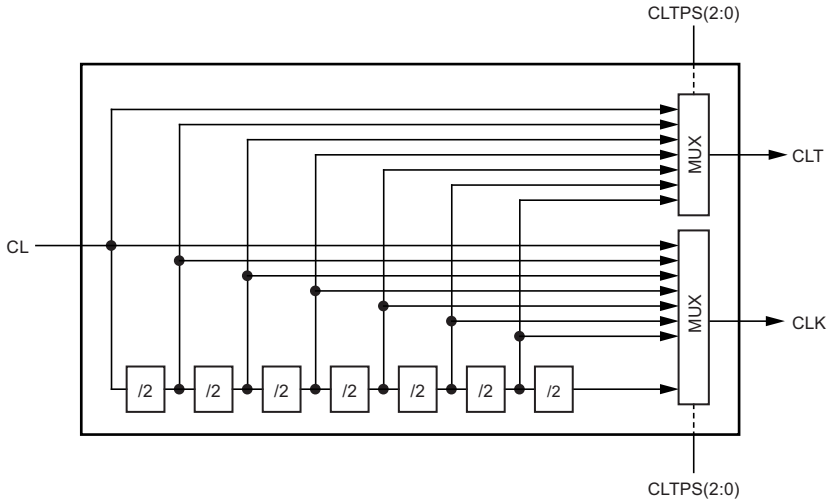
Bits 2..0 – CLKPS2..0: CLocK System Prescaler Select Bits 2 - 0
These bits select the system output clock (CLK) of the system clock prescaler as shown in Table 3-10.

Table 3-10. System Clock Prescaler Select Bit Description

CLKPS2	CLKPS1	CLKPS0	Description
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The CKDIV8 fuse determines the initial value of the CLKPS bits. If CKDIV8 is unprogrammed, the CLKPS bits are reset to “000.” If CKDIV8 is programmed, CLKPS bits are reset to “011,” resulting in a division factor of 8 at startup. This feature should be used if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. Note that any value can be written to the CLKPS bits regardless of the CKDIV8 fuse setting. The application software must ensure that a sufficient division factor is chosen if the selected clock source has a higher frequency than the maximum frequency of the device under the present operating conditions. The device is shipped with the CKDIV8 fuse programmed.

Figure 3-14. System Clock Prescaler and Timer Clock Prescaler



3.8 Power Management and Sleep Modes

Sleep modes enable the application to shut down unused modules in the MCU to save power. The Atmel® ATA6289 provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

To enter one out of three sleep modes, the SE bit in SMCR must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1 and SM0 bits in the SMCR register select which sleep mode (idle, sensor noise reduction, power-down) is activated by the SLEEP instruction (see Table 3-11 on page 33 for more information). If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the startup time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM remain unchanged when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the reset vector.

Figure 3-13 on page 26 shows the different clock systems and their distribution. Refer to this Figure and to Table 3-11 on page 33 for assistance in selecting the appropriate sleep mode.

3.8.1 Sleep Mode Control Register – SMCR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The sleep mode control register contains control bits for power management.

Bits 7..4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bits 3, 2, 1 – SM2..0: Sleep Mode Select Bits 2, 1 and 0
These bits select between the three available sleep modes as shown in Table 3-11.

Table 3-11. Sleep Mode Select

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	Sensor Noise Reduction
0	1	0	Power-Down (lowest current)
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

Bit 0 – SE: Sleep Enable

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode, unless it is the programmer's intention, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after wake-up.

3.8.2 Idle Mode

When all SM2 ...0 bits are written to zero, the SLEEP instruction makes the MCU enter idle mode, stopping the CPU but allowing the SPI, LF receiver, brown-out, voltage monitor, sensor interface, timer/counters, watchdog, and the interrupt system to continue operating. This sleep mode basically halts CLK_{CPU} and CLK_{FLASH}, while allowing the other clocks to run. Idle mode enables the MCU to wake up from externally triggered interrupts as well as internal interrupts such as timer overflow and a transmit buffer empty interrupt or a receive buffer full interrupt of the on-chip digital data modulator.

3.8.3 Sensor Noise Reduction Mode

When the SM2 ... 0 bits are written to "001," the SLEEP instruction makes the MCU enter sensor noise reduction mode. This sleep mode basically halts CLK_{CPU}, CLK_{FLASH} and $CLK_{I/O}$ while allowing the other clocks to run. The FRC oscillator is running and supplies the timers with the CLK_{CLT}.

If timer2/3 are enabled, they keep running during sleep. The device can wake up either due to timer overflow, a capture event or output compare event from timer2/3 if the corresponding timer2/3 interrupt enable bits are set in the T3IMR or T2IMR register, and the global interrupt enable bit in the SREG register is set.

3.8.4 Power-Down Mode

When the SM2..0 bits are written to "010," the SLEEP instruction makes the MCU enter power-down mode. In this mode, the FRC oscillator, an external input clock at ECIN0/ECIN1, the voltage monitor and the brown-out detection are stopped (when the BODPD bit is cleared), while the external interrupts and the watchdog continue operating (if enabled). Only an external reset, a watchdog reset, LF receiver start condition interrupt, an external level interrupt on INT0 or INT1, or a pin change interrupt can wake up the MCU. This sleep mode basically halts all generated clocks except SCL, if enabled, allowing operation of asynchronous modules only. Note that if a level-triggered interrupt is used for wake-up from power-down mode, the changed level must be held for some time to wake up the MCU. For more information, see Section 3.11 "External Interrupts" on page 41.

When waking up from power-down mode, there is a delay of the wake-up condition until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same SUT1..0 fuse bits that define the reset time-out period described in Section 3.7.3 "Clock Sources" on page 27.

Table 3-12. Active Clock Domains and Wake-Up Sources in the Different Sleep Modes

Sleep Mode	Active Clock Domains				Oscillators and External Clocks					Wake-Up Sources									
	clk_{CPU}	clk_{FLASH}	$clk_{I/O}$	clk_{CLT}	External Clock	FRC	SCH	SCL		INT0, INT1 and Pin Change	LF receiver	Voltage Monitor	Thermal Shutdown	Sensor Interface	WDT	BOT	Timer0	Timer1	Timer2/3
IDLE			X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X
Sensor Noise Reduction				X	X	X	X	X		X(1)	X	X	X	X	X	X	X	X	X
Power-Down								X		X(1)	X	X(2)	X(2)	X	X	X(2)	X	X	X

Notes: 1. For INT1 and INT0, only level interrupt.

2. Only, when the BODPD is set bit in the VMCSR register.

3.9 System Control and Reset

3.9.1 Resetting the Atmel AVR

During reset, all I/O registers are set to their initial values and the program starts execution from the reset vector. The instruction placed at the reset vector must be a JMP—Absolute Jump—instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. This is also the case if the reset vector is in the application section while the interrupt vectors are in the boot section or vice versa. The circuit diagram in Figure 3-15 on page 35 shows the reset logic. Table 3-13 on page 35 defines typical electrical parameters of the reset circuitry (see Section 5.2 “Operating Characteristics of Atmel ATA6289” on page 169ff for more information).

The I/O ports of the Atmel® AVR® are immediately reset to their initial state when a reset source is activated. This does not require that any clock source is running.

After all reset sources have been deactivated, a delay counter is invoked, extending the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT and CKSEL fuses. The different selections for the delay period are presented in Section 3.7.3 “Clock Sources” on page 27.

3.9.2 Reset Sources

The ATA6289 has five sources of reset:

Power-On Reset: The MCU is reset when the supply voltage is below the power-on reset threshold (V_{POT}).

External Reset: The MCU is reset when a low level is present on the $\overline{\text{RESET}}$ pin for longer than the minimum pulse length.

Watchdog Reset: The MCU is reset when the watchdog timer period expires and the watchdog is enabled.

Brown-Out Reset: The MCU is reset when the supply voltage V_{CC} is below the brown-out reset threshold (V_{BOT}) and the BOD is enabled.

Temperature Shutdown Reset: The MCU is reset when the ambient temperature exceeds the specified threshold temperature (TSD) and temperature shutdown is enabled.

Figure 3-15. Reset Logic

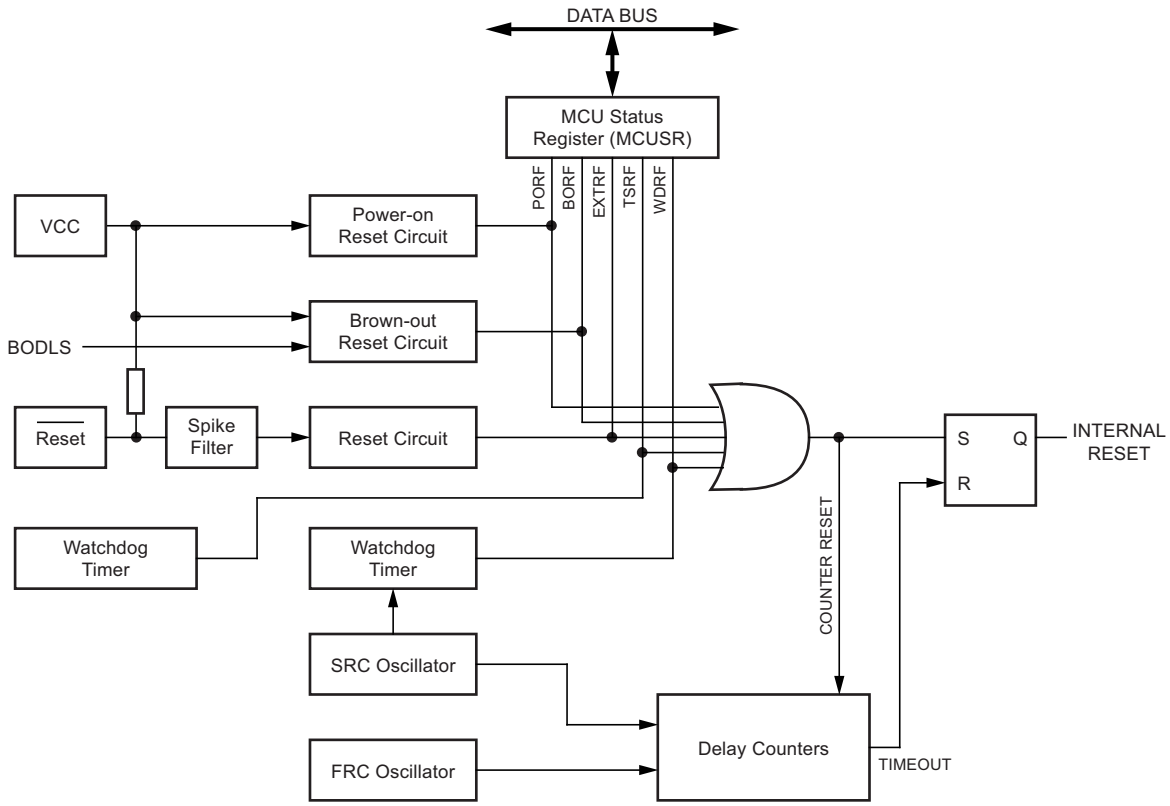


Table 3-13. Reset Characteristics

Parameters	Symbol	Min.	Typ.	Max.	Unit
Power-on reset threshold voltage (rising)	V_{POT}	-	1.15	-	V
Power-on reset threshold voltage (falling)		-	1.15	-	V
$\overline{\text{RESET}}$ pin threshold voltage	V_{RST}	0.2VCC	-	0.8xVCC	V
Minimum pulse width on $\overline{\text{RESET}}$ pin	t_{RST}	2	-	-	µs

3.9.2.1 Power-On Reset

A Power-On Reset (POR) pulse is generated by an on-chip detection circuit. The detection level is defined in Table 3-13. The POR is activated whenever V_{CC} is below the detection level $V_{POT(rising)}$. The POR circuit can be used to trigger the startup reset as well as to detect a failure in the supply voltage.

A POR circuit ensures that the device is reset from power-on. Reaching the power-on reset threshold voltage $V_{POT(rising)}$ invokes the delay counter, determining how long the device is kept in internal RESET after V_{CC} rises. The internal reset signal is activated again without any delay when V_{CC} decreases below the detection level $V_{POT(falling)}$.

Figure 3-16. MCU Startup, RESET Tied To V_{CC} with Internal Pull-Up at $\overline{\text{RESET}}$ Pin

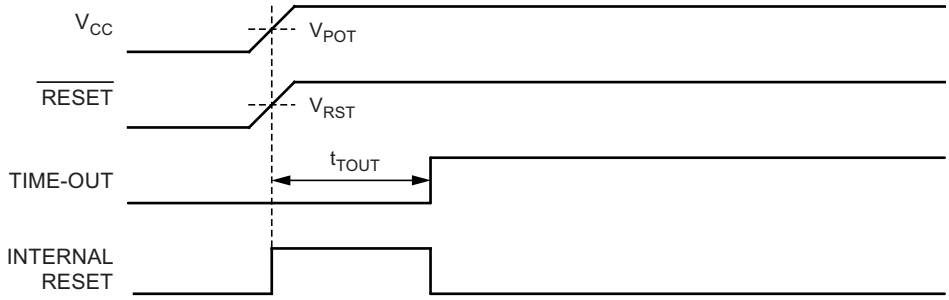
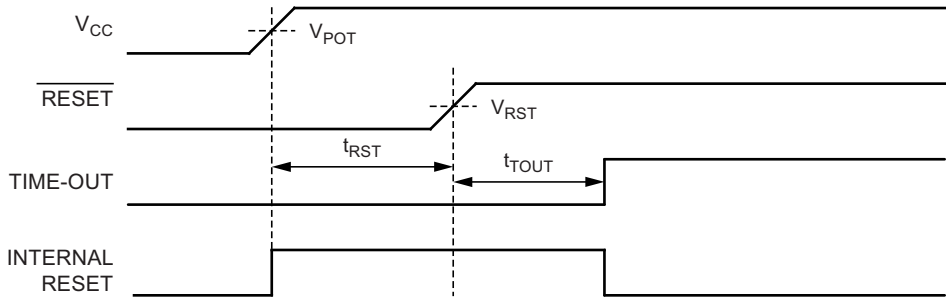


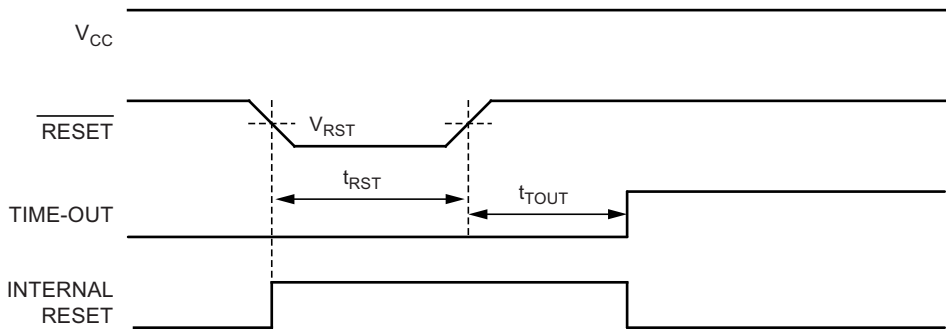
Figure 3-17. MCU Startup, RESET Extended Externally at $\overline{\text{RESET}}$ Pin



3.9.2.2 External Reset

An external reset is generated by a low level at the $\overline{\text{RESET}}$ pin. $\overline{\text{RESET}}$ pulses longer than the minimum pulse width (see Table 3-13 on page 35) generate a reset, even if the clock is not running. Shorter pulses do not ensure a reset is generated. When the applied signal reaches the reset threshold voltage—V_{RST}—on its positive edge, the delay counter starts the MCU after the time-out period—t_{TOUT}—has expired.

Figure 3-18. External Reset During Operation at $\overline{\text{RESET}}$ Pin



3.9.2.3 Brown-Out Detection/Reset

The ATA6289 has an on-chip BOD circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BODLS bits in the VMCSR register. The trigger level has a hysteresis to ensure spike-free brown-out detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST/2}$ and $V_{BOT-} = V_{BOT} - V_{HYST/2}$.

Table 3-14. Brown-Out Detection Level Select Bit Description

BODLS	Typ. V_{BOT}	Unit
0	1.8	V
1	2.0	V

Table 3-15. Brown-Out Detection Characteristics

Parameters	Symbol	Typ.	Unit
Brown-out detector hysteresis	V_{HYST}	50	mV
Min. pulse width on brown-out reset	t_{BOD}	2	µs

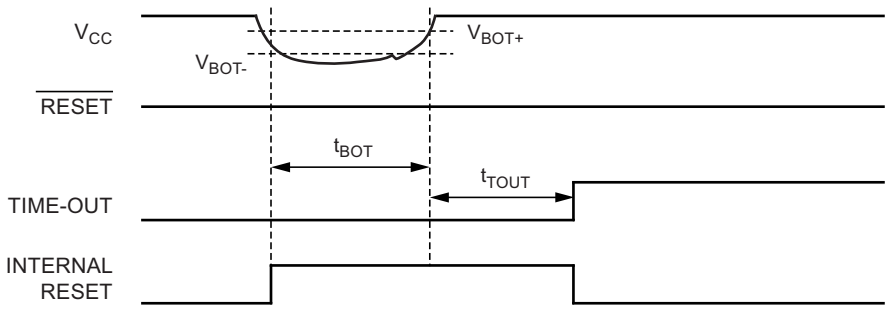
For more information, see Section 5.2 “Operating Characteristics of Atmel ATA6289” on page 169.

When the BOD is enabled, and VCC decreases to a value below the trigger level (V_{BOT-} in Figure 3-19), the brown-out reset is immediately activated. When VCC increases above the trigger level (V_{BOT+} in Figure 3-19), the delay counter starts the MCU after the time-out period t_{TOUT} has expired.

The BOD circuit only detects a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} as shown in Table 3-15.

For more information on BOD, see Section 3.16 “Voltage Monitor and Brown-Out Detection” on page 109.

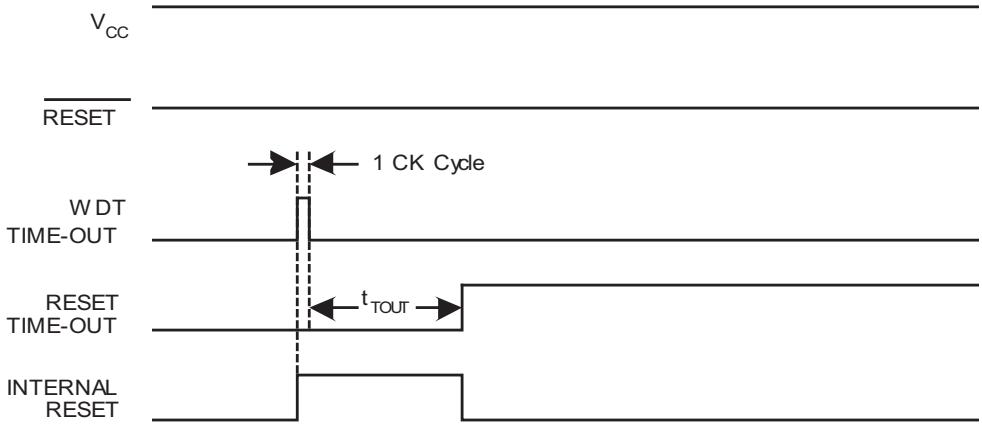
Figure 3-19. Brown-Out Reset During Operation



3.9.2.4 Watchdog Reset

When the watchdog times out, it generates a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the time-out period t_{TOUT}. For more details on operating the watchdog timer, see Section 3.13.3 “Timer0 with Watchdog/Interval Timer” on page 62.

Figure 3-20. Watchdog Reset During Operation



3.9.2.5 Temperature Shutdown Reset

An internal temperature shutdown circuit avoids an undefined device operation above the normal operating temperature range. In cases where the ambient temperature exceeds the specified threshold temperature ($T_{SD} \approx 120°C$, not calibrated yet in production), the device can be forced into reset state and the flag can be set. This immediately stops device operation. As a result, all chip circuitry and port lines are disabled, including the interval timer. Once the device has entered temperature shutdown mode, only the temperature sensor circuitry remains active.

For more information on temperature shutdown, see Section 3.15.6 “Temperature Shutdown Mode” on page 108.

3.9.2.6 MCU Status Register – MCUSR

The MCU status register provides information on which reset source caused an MCU reset

Bit	7	6	5	4	3	2	1	0	
	-	-	TSRF	-	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7, 6, 4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 5 – TSRF: Temperature Shutdown Reset Flag
This bit is set if the ambient temperature exceeds the specified threshold temperature (T_{SD}) and temperature shutdown mode is enabled by the TSSD bit in the TSCR register. This bit is reset by a power-on reset or by writing a logic zero to the flag.

Bit 3 – WDRF: Watchdog Reset Flag
This bit is set if a watchdog reset occurs and watchdog is enabled. The bit is reset by a power-on reset, or by writing a logic zero to the flag.

Bit 2 – BORF: Brown-Out Reset Flag
This bit is set if a brown-out reset occurs and BOD is enabled. The bit is reset by a power-on reset, or by writing a logic zero to the flag.

Bit 1 – EXTRF: External Reset Flag
This bit is set if an external reset occurs. The bit is reset by a power-on reset, or by writing a logic zero to the flag.

Bit 0 – PORF: Power-On Reset Flag
This bit is set if a power-on reset occurs. The bit is reset only by writing a logic zero to the flag.

The user should read and then reset the MCUSR in the program as early as possible to utilize reset flags for identifying a reset condition. If the register is cleared before another reset occurs, the source of the reset can be found by examining the reset flags.

3.10 Interrupts

This section describes the specifics of the interrupt handling as performed in ATA6289. For a general explanation of Atmel® AVR® interrupt handling, see Section 3.5.7 “Reset and Interrupt Handling” on page 14. On the ATA6289 the reset vector is affected by the BOOTRST fuse and the interrupt vector start address is affected by the IVSEL bit in the MCUCR register. For more information, see Section 3.10.2.1 “MCU Control Register – MCUCR” on page 40.

3.10.1 Interrupt Vectors

Table 3-16. Reset and Interrupt Vectors

Vector No.	Program Address(1)	Source	Interrupt Definition
1	0x000(2)	RESET	External pin, power-on reset, brown-out reset, watchdog reset, and temperature shutdown reset
2	0x001	INT0	External interrupt request 0
3	0x002	INT1	External interrupt request 1
4	0x003	PCINT0	Pin change interrupt request 0
5	0x004	PCINT1	Pin change interrupt request 1
6	0x005	PCINT2	Pin change interrupt request 2
7	0x006	INTVM	Voltage monitoring interrupt
8	0x007	SENINT	Sensor interface interrupt
9	0x008	INTT0	Timer0 interval interrupt
10	0x009	LFWP	LF receiver wake-up interrupt
11	0x00A	T3CAP	Timer/Counter3 capture event
12	0x00B	T3COMA	Timer/Counter3 compare match A
13	0x00C	T3COMB	Timer/Counter3 compare match B
14	0x00D	T3OVF	Timer/Counter3 overflow
15	0x00E	T2CAP	Timer/Counter2 capture event
16	0x00F	T2COM	Timer/Counter2 compare match
17	0x010	T2OVF	Timer/Counter2 overflow
18	0x011	SPISTC	SPI serial transfer complete
19	0x012	LFRXB	LF receiver receive buffer interrupt
20	0x013	INTT1	Timer1 interval interrupt
21	0x014	T2RXB	Timer2 SSI receive buffer interrupt
22	0x015	T2TXB	Timer2 SSI transmit buffer interrupt
23	0x016	T2TXC	Timer2 SSI transmit complete interrupt
24	0x017	LFREOB	LF receiver end of burst interrupt
25	0x018	EXCM	External input clock break down interrupt
26	0x019	EEREADY	EE ready interrupt
27	0x01A	SPM READY	Store program memory ready

Notes:
1. When the IVSEL bit in MCUCR is set, interrupt vectors are moved to the start of the boot Flash section. The address of each interrupt vector is then the address in this table added to the start address of the boot Flash section.
2. When the BOOTRST fuse is programmed, the device jumps to the boot loader address at reset. For more information, see Section 3.19 “Boot Loader Support – Read-While-Write Self-Programming” on page 126.

Table 3-17 shows reset and interrupt vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, interrupt vectors are not used and regular program code can be placed at these locations. This is also the case if the reset vector is in the application section while the interrupt vectors are in the boot section or vice versa.

Table 3-17. Reset and Interrupt Vectors Placement in ATA6289(1)

BOOTRST	IVSEL	Reset Address	Interrupt Vector Start Address
1	0	0x0000	0x0001
1	1	0x0000	Boot Reset Address + 0x0001
0	0	Boot Reset Address	0x0001
0	1	Boot Reset Address	Boot Reset Address + 0x0001

Note: 1. For the BOOTRST fuse, “1” means unprogrammed while “0” means programmed.

3.10.2 Moving Interrupts Between Application and Boot Space

The general interrupt control register controls the placement of the interrupt vector table.

3.10.2.1 MCU Control Register – MCUCR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..5 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 4 – PUD: Pull-Up Disable
This bit is described in the I/O port section.

Bits 3..2 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 1 – IVSEL: Interrupt Vector Select
When the IVSEL bit is cleared (zero), the interrupt vectors are placed at the start of the Flash memory. When this bit is set (one), the interrupt vectors are moved to the beginning of the boot loader section of the Flash. The actual address of the start of the boot Flash section is determined by the BOOTSZ fuses. See Section 3.19 “Boot Loader Support – Read-While-Write Self-Programming” on page 126 for more information. To avoid unintentional changes of interrupt vector tables, a special write procedure must be followed to change the IVSEL bit:

Write the Interrupt Vector Change Enable (IVCE) bit to one.

Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts are automatically disabled while this sequence is executed. Interrupts are disabled if the cycle IVCE is set and remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I bit in the status register is unaffected by the automatic disabling.

Note: If interrupt vectors are placed in the boot loader section and boot lock bit BLB02 is programmed, interrupts are disabled while executing from the application section. If interrupt vectors are placed in the application section and boot lock bit BLB12 is programmed, interrupts are disabled while executing from the boot loader section. See Section 3.19 “Boot Loader Support – Read-While-Write Self-Programming” on page 126 for more information on boot lock bits.

Bit 0 – IVCE: Interrupt Vector Change Enable
The IVCE bit must be written to logic one to enable the IVSEL bit to be changed. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. As explained in the IVSEL description above, setting the IVCE bit disables interrupts.

3.11 External Interrupts

The external interrupts are triggered by the INT0 pin or INT1 pin or any of the PCINT23..16,10..0 pins. Note that, if enabled, the interrupts trigger, even if the INT0 or INT1 or PCINT23..16,10..0 pins are configured as outputs. This feature makes it possible to generate a software interrupt. The pin change interrupt PCI2 triggers if any enabled PCINT23..16 pin toggles. The pin change interrupt PCI1 triggers if any enabled PCINT10..8 pin toggles. Pin change interrupt PCI0 triggers if any enabled PCINT7..0 pin toggles. The PCMSK2, PCMSK1 and PCMSK0 registers control which pins contribute to the pin-change interrupts. Pin-change interrupts on PCINT23..16,10..0 are detected asynchronously. This implies that these interrupts can also be used for waking the part from sleep modes other than idle mode.

The INT0 or INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the External Interrupt Control Register A (EICRA). When the INT0 or INT1 interrupt is enabled and is configured as level-triggered, the interrupt triggers as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock. This is described in Section 3.7.2 “Clock Systems and Their Distribution” on page 26. A low-level interrupt on INT0 or INT1 is detected asynchronously. This implies that this interrupt can also be used for waking the part from sleep modes other than idle mode. The I/O clock is halted in all sleep modes except idle mode.

Note that if a level-triggered interrupt is used for wake-up from power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the startup time, the MCU still wakes up, but no interrupt is generated. The startup time is defined by the SUT fuses and CMM2..0 as described in Section 3.7 “Clock Generation” on page 23.

3.11.1 External Interrupt Control Register A – EICRA

The external interrupt control register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 3, 2 – ISC11, ISC10: Interrupt 1 Sense Control Bit 1 and Bit 0
The external interrupt 1 is activated by the external pin INT1 if the SREG I flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined by ISC11 and ISC10 (see Table 3-18). The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 3-18. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request
0	1	Any logical change on INT1 generates an interrupt request
1	0	The falling edge of INT1 generates an interrupt request
1	1	The rising edge of INT1 generates an interrupt request

Bit 1, 0 – ISC01, ISC00: Interrupt 0 Sense Control Bit 1 and Bit 0
The external interrupt 0 is activated by the external pin INT0 if the SREG I flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined by ISC00 and ISC01 (see Table 3-19). The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low-level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 3-19. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request
0	1	Any logical change on INT0 generates an interrupt request
1	0	The falling edge of INT0 generates an interrupt request
1	1	The rising edge of INT0 generates an interrupt request

3.11.2 External Interrupt Mask Register – EIMSK

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	**INT1**	**INT0**	**EIMSK**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..2 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 1 – INT1: External Interrupt Request 1 Enable
When the INT1 bit is set (one) and the I bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The interrupt sense control 1 bits ISC11 and ISC10 in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on the rising and/or falling edge of the INT1 pin or level-sensed. Activity at the pin causes an interrupt request, even if INT1 is configured as an output. The corresponding interrupt of external interrupt request 1 is executed from the INT1 interrupt vector.

Bit 0 – INT0: External Interrupt Request 0 Enable
When the INT0 bit is set (one) and the I bit in the SREG is set (one), the external pin interrupt is enabled. The interrupt sense control 0 bits ISC01 and ISC00 in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on the rising and/or falling edge of the INT0 pin or level-sensed. Activity at the pin causes an interrupt request, even if INT0 is configured as an output. The corresponding interrupt of external interrupt request 0 is executed from the INT0 interrupt vector.

3.11.3 External Interrupt Flag Register – EIFR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	**INTF1**	**INTF0**	**EIFR**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..2 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 1 – INTF1: External Interrupt Flag 1
When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 is set (one). If the I bit in SREG and the INT1 bit in EIMSK are set (one), the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it. This flag is always cleared when INT1 is configured as a level interrupt.

Bit 0 – INTF0: External Interrupt Flag 0
When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I bit in SREG and the INT0 bit in EIMSK are set (one), the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it. This flag is always cleared when INT0 is configured as a level interrupt.

3.11.4 Pin Change Interrupt Control Register – PCICR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	**PCIE2**	**PCIE1**	**PCIE0**	**PCICR**
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..3 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 2 – PCIE2: Pin Change Interrupt Enable 2
When the PCIE2 bit is set (one) and the I bit in the SREG is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT23..16 pin causes an interrupt. The corresponding interrupt of the pin-change interrupt request is executed from the PCI2 interrupt vector. PCINT23..16 pins are enabled individually by the PCMSK2 register.

Bit 1 – PCIE1: Pin Change Interrupt Enable 1
When the PCIE1 bit is set (one) and the I bit in the SREG is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT10..8 pin causes an interrupt. The corresponding pin change interrupt is executed from the PCI1 interrupt vector. PCINT10..8 pins are enabled individually by the PCMSK1 register.

Bit 0 – PCIE0: Pin Change Interrupt Enable 0
When the PCIE0 bit is set (one) and the I bit in the SREG is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7..0 pin causes an interrupt. The corresponding pin change interrupt request is executed from the PCI0 interrupt vector. PCINT7..0 pins are enabled individually by the PCMSK0 register.

3.11.5 Pin Change Interrupt Flag Register – PCIFR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	**PCIF2**	**PCIF1**	**PCIF0**	**PCIFR**
Read/Write	R/W	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..3 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 2 – PCIF2: Pin Change Interrupt Flag 2
When a logic change on any PCINT23..16 pin triggers an interrupt request, PCIF2 becomes set (one). If the I bit in SREG and the PCIE2 bit in PCICR are set (one), the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it.

Bit 1 – PCIF1: Pin Change Interrupt Flag 1
When a logic change on any PCINT10..8 pin triggers an interrupt request, PCIF1 is set (one). If the I bit in SREG and the PCIE1 bit in PCICR are set (one), the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it.

Bit 0 – PCIF0: Pin Change Interrupt Flag 0
When a logic change on any PCINT7..0 pin triggers an interrupt request, PCIF0 is set (one). If the I bit in SREG and the PCIE0 bit in PCICR are set (one), the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it.

3.11.6 Pin Change Mask Register 0 – PCMSK0

Bit	7	6	5	4	3	2	1	0	
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7..0 – PCINT7..0: Pin Change Interrupt Enable Mask 7..0
Each PCINT7..0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

3.11.7 Pin Change Mask Register 1 – PCMSK1

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..3 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 2..0 – PCINT10..8: Pin Change Interrupt Enable Mask 10..8
Each PCINT10..8-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT10..8 is set and the PCIE1 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT10..8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

3.11.8 Pin Change Mask Register 2 – PCMSK2

Bit	7	6	5	4	3	2	1	0	
	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

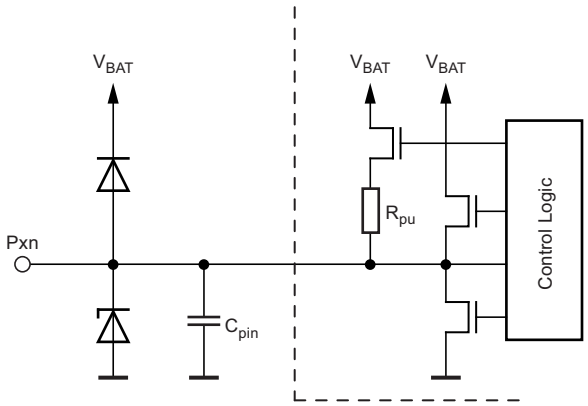
Bit 7..0 – PCINT23..16: Pin Change Interrupt Enable Mask 23..16
Each PCINT23..16-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is set and the PCIE2 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

3.12 I/O Ports

3.12.1 Introduction

All Atmel® AVR® ports have true read-modify-write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high-sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and ground as indicated in Figure 3-21. Refer to Section 5.2 “Operating Characteristics of Atmel ATA6289” on page 169ff for a complete list of electrical parameters.

Figure 3-21. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 is used for bit no. 3 in port B, here documented generally as PORTxn. The physical I/O registers and bit locations are listed in Section 3.12.4 “Register Description for I/O Ports” on page 56.

Three I/O memory address locations are allocated for each port, one each for the data register (PORTx), data direction register (DDRx) and the port input pins (PINx). The port input pins I/O location is read-only, while the data register and the data direction register are read/write. However, writing a logic one to a bit in the PINx register results in a toggle in the corresponding bit of the register. In addition, the Pull-Up Disable (PUD) bit in MCUCR disables the pull-up function for all pins in all ports when set.

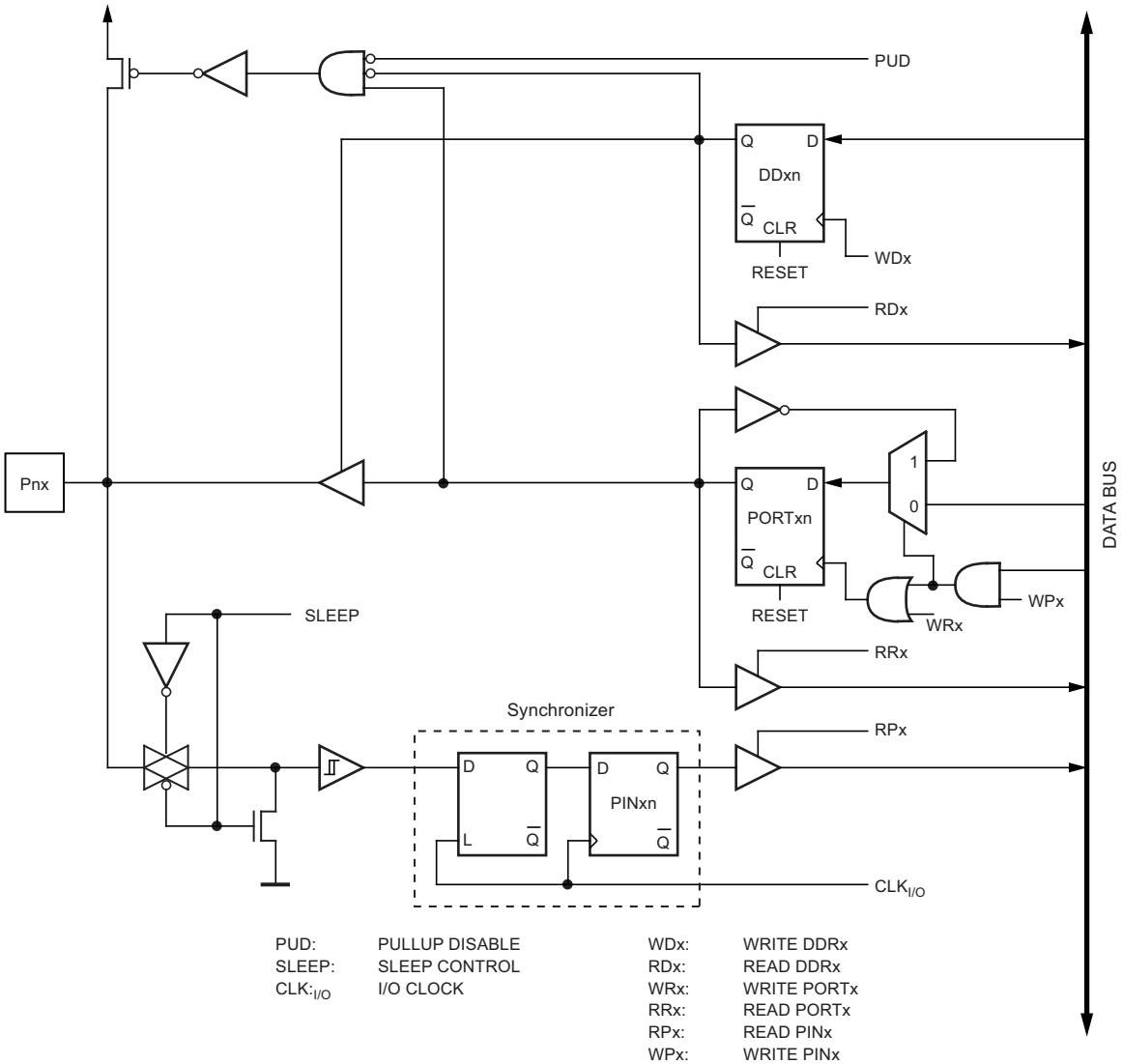
Using the I/O port as the general digital I/O is described in Section 3.12.2 “Ports as General Digital I/O” on page 46. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in Section 3.12.3 “Alternate Port Functions” on page 49. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as the general digital I/O.

3.12.2 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 3-22 shows a functional description of one I/O port pin, here generically called Pxn.

Figure 3-22. General Digital I/O(1)



WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $CLK_{I/O}$, SLEEP and PUD are common to all ports.

3.12.2.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn and PINxn. As shown in Section 3.12.4 “Register Description for I/O Ports” on page 56, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch off the pull-up resistor, PORTxn has to be written logic zero or the pin has to be configured as an output pin or PUD must be set to logic one. The port pins are tri-stated when the reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one with the pin configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero with the pin configured as an output pin, the port pin is driven low (zero).

3.12.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, regardless of the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

3.12.2.3 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable because a high-impedance environment does not detect the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR register can be set to disable all pull-ups in all ports.

Switching between an input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.

Table 3-20 summarizes the control signals for the pin value.

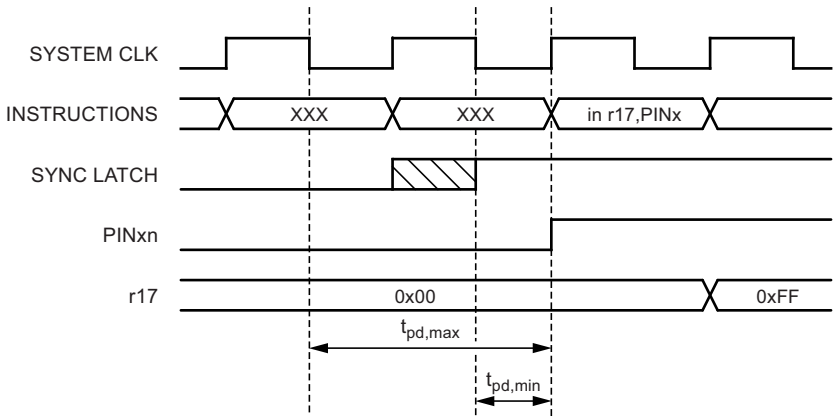
Table 3-20. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-Up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn sources current if pulled low externally
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output low (sink)
1	1	X	Output	No	Output high (source)

3.12.2.4 Reading the Pin Value

The port pin can be read through the PINxn register bit regardless of how the data direction bit DDxn is set. As shown in Figure 3-22 on page 46, the PINxn register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also generates a delay. Figure 3-23 on page 47 shows a timing diagram of the synchronization when reading a pin value applied externally. The maximum and minimum propagation delays are denoted as tpd, max and tpd,min respectively.

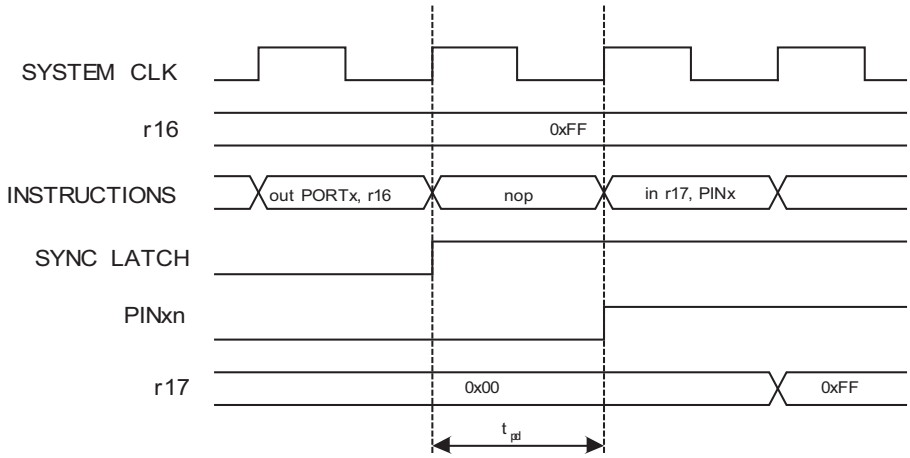
Figure 3-23. Synchronization when Reading an Externally Applied Pin Value



Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low and becomes transparent when the clock is high, as indicated by the shaded region of the “SYNC LATCH” signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn register at the succeeding positive clock edge. As indicated by the two arrows tpd,max and tpd,min, a single signal transition at the pin is delayed anywhere from a ½ to 1½ system clock periods depending upon the time of assertion.

When reading back a software-assigned pin value, a nop instruction must be inserted as indicated in Figure 3-24. The out instruction sets the “SYNC LATCH” signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is 1 system clock period.

Figure 3-24. Synchronization when Reading a Software-Assigned Pin Value



3.12.2.5 Digital Input Enable and Sleep Modes

As shown in Figure 3-22 on page 46, the digital input signal can be clamped to ground at the input of the Schmitt trigger. The signal marked SLEEP in the Figure is set by the MCU sleep controller in power-down mode, power-save mode and standby mode to avoid high power consumption, even if some input signals are left floating or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is also active for these pins. SLEEP is also overridden by various other alternate functions as described in Section 3.12.3 “Alternate Port Functions” on page 49.

If a logic high level (“one”) is present on an asynchronous external interrupt pin configured as “Interrupt on Rising Edge, Falling Edge or Any Logic Change on Pin” while the external interrupt is not enabled, the corresponding external interrupt flag is set when resuming from the above-mentioned sleep mode because the clamping in this sleep mode produces the requested logic change.

3.12.2.6 Unconnected Pins

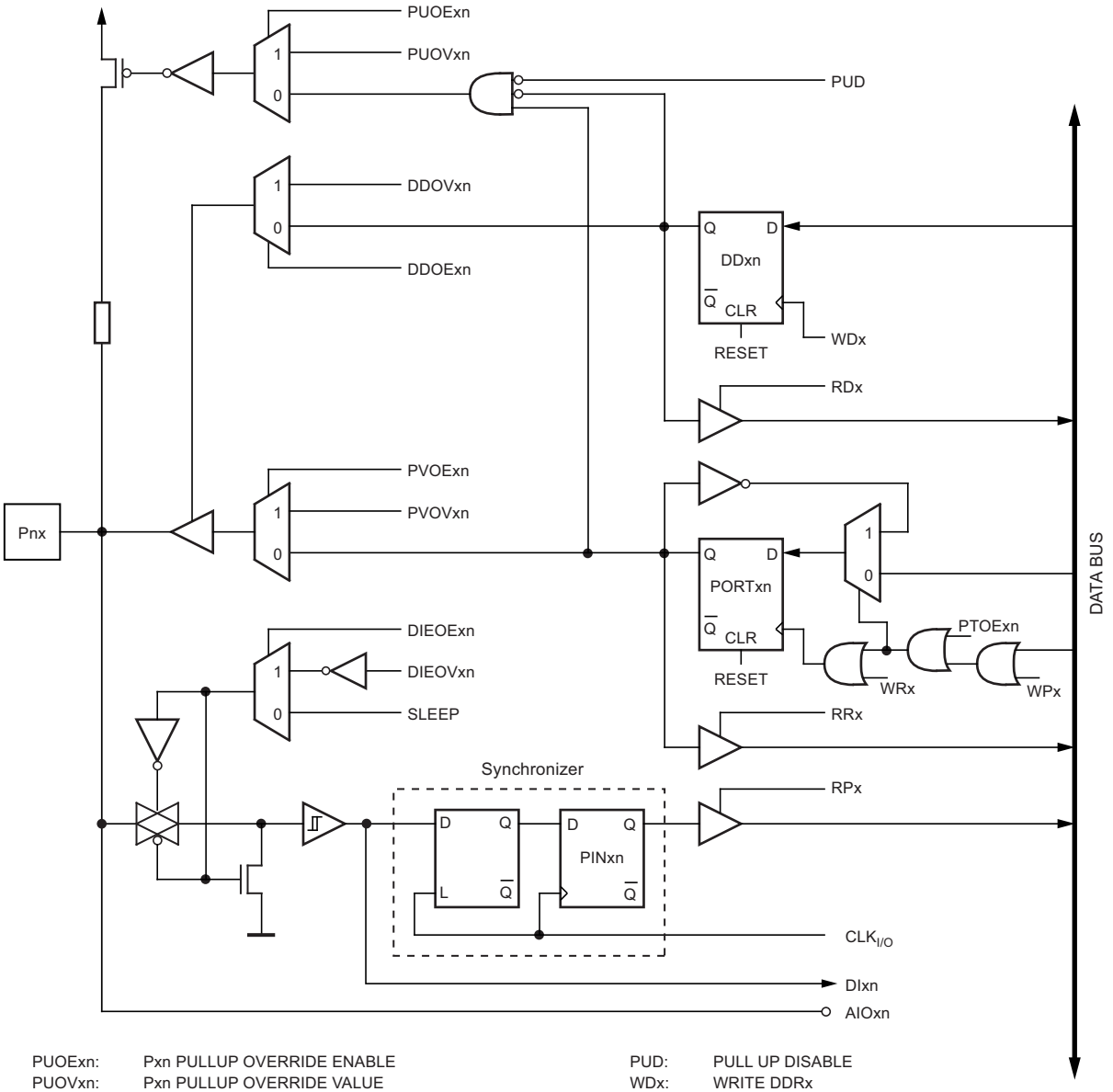
If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (reset, active mode and idle mode).

The most straightforward method for ensuring a defined level of an unused pin is to enable the internal pull-up. In this case, the pull-up is disabled during reset. If low-power consumption during reset is important, use of an external pull-up or pull-down is recommended. It is not advisable to connect unused pins directly to VCC or GND because this may cause excessive currents if the pin is accidentally configured as an output.

3.12.3 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. Figure 3-25 shows how the port pin control signals from the simplified Figure 3-22 on page 46 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the Figure serves as a generic description applicable to all port pins in the Atmel® AVR® microcontroller family.

Figure 3-25. Alternate Port Functions(1)



PUOExn:	Pxn PULLUP OVERRIDE ENABLE	PUD:	PULL UP DISABLE
PUOVxn:	Pxn PULLUP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn PDATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WRx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn DIGITAL INPUT ENABLE OVERRIDE ENABLE	WPx:	WRITE PORT REGISTER
DIEOVxn:	Pxn DIGITAL INPUT ENABLE OVERRIDE VALUE	CLK:$_{I/O}$	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	DIGITAL INPUT PINn ON PORTx
PTOExn:	Pxn PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG INPUT/OUTPUT PINn ON PORTx

WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $CLK_{I/O}$, SLEEP and PUD are common to all ports. All other signals are unique for each pin.

Table 3-21 summarizes the function of the overriding signals. The pin and port indexes are not shown in the succeeding tables. The overriding signals are generated internally in the modules which have the alternate function.

Table 3-21. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-Up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-Up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn and PUD register bits.
DDOE	Data Direction Override Enable	If this signal is set, the output driver enable is controlled by the DDOV signal. If this signal is cleared, the output driver is enabled by the DDxn register bit.
DDOV	Data Direction Override Value	If DDOE is set, the output driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn register bit.
PVOE	Port Value Override Enable	If this signal is set and the output driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the output driver is enabled, the port value is controlled by the PORTxn register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn register bit.
PTOE	Port Toggle Override Enable	If PTOE is set, the PORTxn register bit is inverted.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the digital input enable is controlled by the DIEOV signal. If this signal is cleared, the digital input enable is determined by MCU state (normal mode, sleep mode).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the digital input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (normal mode, sleep mode).
DI	Digital Input	This is the digital input for alternate functions. In the figure, the signal is connected to the output of the Schmitt trigger but before the synchronizer. Unless the digital input is used as a clock source, the module with the alternate function uses its own synchronizer.
AIO	Analog Input/Output	This is the analog input/output to/from alternate functions. The signal is connected directly to the pad and can be used bi-directionally.

The following subsections provide a brief description of alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

3.12.3.1 Alternate Functions of Port B

The port B pins with alternate functions are shown in Table 3-22.

Table 3-22. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	$\overline{SS}$ (SPI Bus Master Slave select) PCINT7 (Pin Change Interrupt 7)
PB6	PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master Output Clock/Slave Input Clock) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) PCINT3 (Pin Change Interrupt 3)
PB2	T2I (Timer1, Timer2, Timer3 External Input Clock Input) PCINT2 (Pin Change Interrupt 2)
PB1	T3O (Timer3 Output) PCINT1 (Pin Change Interrupt 1)
PB0	T3ICP (Timer3 External Input Capture Input) PCINT0 (Pin Change Interrupt 0)

3.12.3.2 The Alternate Pin Configuration on Port B:

$\overline{SS}$/PCINT7 – Port B, Bit 7

$\overline{SS}$: Not Slave Port Select Input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB7. As a slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB7. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB7 bit.

PCINT7, Pin Change Interrupt Source 7: The PB7 pin can serve as an external interrupt source.

PCINT6 – Port B, Bit 6

PCINT6, Pin Change Interrupt Source 6: The PB6 pin can serve as an external interrupt source.

SCK/PCINT5 – Port B, Bit 5

SCK: Master Clock Output, Slave Clock Input Pin for SPI Channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB5. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB5 bit.

PCINT5, Pin Change Interrupt Source 5: The PB5 pin can serve as an external interrupt source.

MISO/PCINT4 – Port B, Bit 4

MISO: Master Data Input, Slave Data Output Pin for SPI Channel. When the SPI is enabled as a master, this pin is configured as an input regardless of the setting of DDB4. When the SPI is enabled as a slave, the data direction of this pin is controlled by DDB4. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB4 bit.

PCINT4, Pin Change Interrupt Source 4: The PB4 pin can serve as an external interrupt source.

MOSI/PCINT3 – Port B, Bit 3

MOSI: SPI Master Data Output, Slave Data Input for SPI Channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB3. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB3 bit.

PCINT3, Pin Change Interrupt Source 3: The PB3 pin can serve as an external interrupt source.

T2I/PCINT2 – Port B, Bit 2

T2I: External Input Clock for Timer1, Timer2 or Timer3. The pin has to be configured as an input (DDB2 set (zero)) to serve this function.

PCINT2, Pin Change Interrupt Source 2: The PB2 pin can serve as an external interrupt source.

T3O/PCINT1 – Port B, Bit 1
T3O, Timer3 Modulator Output: The PB1 pin can serve as an external output for the timer/counter3 modulator. The pin has to be configured as an output (DDB1 set (one)) to serve this function.

PCINT1, Pin Change Interrupt Source 1: The PB1 pin can serve as an external interrupt source.

T3ICP/PCINT0 – Port B, Bit 0
T3ICP – Timer3 Input Capture Pin: The PB0 pin can act as an external input capture pin for timer/counter3. The pin has to be configured as an input (DDB0 set (zero)) to serve this function.

PCINT0, Pin Change Interrupt Source 0: The PB0 pin can serve as an external interrupt source.

Table 3-23 and Table 3-29 show the alternate functions of port B to the overriding signals shown in Table 3-21 on page 50. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 3-23. Over Alternate Functions in PB7..PB4

Signal Name	PB7/SS/PCINT7	PB6/PCINT6	PB5/SCK/PCINT5	PB4/MISO/PCINT4
PUOE	SPE × NMSTR	0	SPE × NMSTR	SPE × MSTR
PUOV	PORTB7 × NPUD	0	PORTB5 × NPUD	PORTB4 × NPUD
DDOE	SPE × NMSTR	0	SPE × NMSTR	SPE × MSTR
DDOV	0	0	0	0
PVOE	0	0	SPE × MSTR	SPE × NMSTR
PVOV	0	0	SCK OUTPUT	SPI SLAVE OUTPUT
DIEOE	PCINT7 × PCIE0	PCINT6 × PCIE0	PCINT5 × PCIE0	PCINT4 × PCIE0
DIEOV	1	1	1	1
DI	PCINT7 INPUT SPI $\overline{SS}$	PCINT6 INPUT	PCINT5 INPUT SCK INPUT	PCINT4 INPUT SPI MSTR INPUT
AIO	-	-	-	-

Table 3-24. Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/MOSI/PCINT3	PB2/T2I/PCINT2	PB1/T3O/PCINT1	PB0/T3ICP/PCINT0
PUOE	SPE × NMSTR	0	0	0
PUOV	PORT3 × NPUD	0	0	0
DDOE	SPE × NMSTR	0	0	0
DDOV	0	0	0	0
PVOE	SPE × MSTR	0	T3O ENABLE	0
PVOV	SPI MSTR OUTPUT	0	T3O	0
DIEOE	PCINT3 × PCIE0	PCINT2 × PCIE0	PCINT1 × PCIE0	PCINT0 × PCIE0
DIEOV	1	1	1	1
DI	PCINT3 INPUT SPI SLAVE INPUT	PCINT2 INPUT T2I INPUT	PCINT1 INPUT	PCINT0 INPUT T3ICP INPUT
AIO	-	-	-	-

3.12.3.3 Alternate Functions of Port C

The port C pins with alternate functions are shown in Table 3-25.

Table 3-25. Port C Pins Alternate Functions

Port Pin	Alternate Functions
PC2	PCINT10 (Pin Change Interrupt 10)
PC1	PCINT9 (Pin Change Interrupt 9) CLKO (Clock Output Pin for System Clock)
PC0	PCINT8 (Pin Change Interrupt 8) EXIN0 (External Clock Input 0 Pin)

The alternate pin configuration on port C:

PCINT10 – Port C, Bit 2
PCINT10, Pin Change Interrupt Source 10: The PC2 pin can serve as an external interrupt source.

CLKO/PCINT9 – Port C, Bit 1
CLKO: Divided System Clock Output. The divided system clock can be output on the PC1 pin. The divided system clock is output if the CKOUT fuse bit is set (one), regardless of the PORTC1 and DDC1 setting. It is also output during reset.

PCINT9, Pin Change Interrupt Source 9: The PC1 pin can serve as an external interrupt source.

ECIN0/PCINT8 – Port C, Bit 0
ECIN0: External Clock Input 0. External system clock input 0 for the clock module on PC0 pin. When used as a clock pin, the pin cannot be used as an I/O pin. The clock is input if the CMM[1..0] bits are set (one), the ECINS bit is cleared (zero) and CSS bit is set (one) in the CMCR register, regardless of the PORTC0 and DDC0 setting, and PORTC0, DDC0 and PINC0 all read 0.

PCINT8, Pin Change Interrupt Source 8: The PC0 pin can serve as an external interrupt source.

Table 3-26 relates the alternate functions of port C to the overriding signals shown in Table 3-21 on page 50.

Table 3-26. Overriding Signals for Alternate Functions in PC2..PC0

Signal Name	PC2/PCINT10	PC1/CLKO/PCINT9	PC0/EXIN0/PCINT8
PUOE		CLKO	
PUOV		0	0
DDOE		CLKO(1)	CMM[1..0] × NECINS × CSS
DDOV		1	0
PVOE		CLKO(1)	0
PVOV		$CLK_{I/O}$	0
DIEOE	PCINT10 × PCIE1	PCINT9 × PCIE1	PCINT8 × PCIE1
DIEOV	1	1	1
DI	PCINT10 INPUT	PCINT9 INPUT CLOCK OUTPUT	PCINT8 INPUT EXIN0 INPUT
AIO	Sensor input 4	Oscillator output	Oscillator input

Note: 1. CLKO is one if the CKOUT fuse is programmed.

3.12.3.4 Alternate Functions of Port D

The port D pins with alternate functions are shown in Table 3-27.

Table 3-27. Port D Pins Alternate Functions

Port Pin	Alternate Functions
PD7	SDIN (Timer2 SSI Serial Input Data Input) PCINT23 (Pin Change Interrupt 23)
PD6	T2O2 (Timer2 Output 2) PCINT22 (Pin Change Interrupt 22)
PD5	T2O1 (Timer2 Output 1) PCINT21 (Pin Change Interrupt 21)
PD4	ECIN1 (External System Input Clock 1 Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Input Interrupt 1 Input) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Input Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	T3I (Timer1, Timer2, Timer3 External Input Clock Input) PCINT17 (Pin Change Interrupt 17)
PD0	T2ICP (Timer2 External Input Capture Input) PCINT16 (Pin Change Interrupt 16)

The alternate pin configuration on port D:

SDIN/PCINT23 – Port D, Bit 7
The external Serial Data Input (SDI) of the SSI on PD7 pin. When the SPI mode of the SSI is selected (T2M[3..0] = 0b1001) in the timer2 modulator, this pin is configured as an input regardless of the setting of DDD7. When the pin is forced to be an input, the pull-up can still be controlled by the PORTD7 bit.

PCINT23, Pin Change Interrupt Source 23: The PD7 pin can serve as an external interrupt source.

T2O2/PCINT22 – Port D, Bit 6
T2O2: Timer2 Output 2. The PD6 pin can serve as an external output for the timer2 modulator output 2. The function of the modulator output pin is dependent on the timer2 mode bits T2M[3..0] bits in the T2MRB register. The PD6 pin has to be configured as an output (DDD6 set [one]) if the timer mode relies on the output pin to perform this function. When PD6 is not used as modulator output pin, it is an I/O pin.

PCINT22, Pin Change Interrupt Source 22: The PD6 pin can serve as an external interrupt source.

T2O1/PCINT21 – Port D, Bit 5
T2O1: Timer2 Output 1. The PD5 pin can serve as an external output for the timer2 modulator output 1. The function of the modulator output pin is dependent on the timer2 mode bits T2M[3..0] bits in the T2MRB register. The PD5 pin has to be configured as an output (DDD5 set [one]) if the timer mode relies on the output pin to perform this function. When PD5 is not used as modulator output pin, it is an I/O pin.

PCINT21, Pin Change Interrupt Source 21: The PD5 pin can serve as an external interrupt source.

ECIN1/PCINT20 – Port D, Bit 4
ECIN1: External Clock Input. External system clock input for the clock module on PD4 pin. When used as a clock pin, the pin cannot be used as an I/O pin. The clock is input if the CMM[1..0] bits are set (one), the ECINS bit is set (one) and CCS bit is set (one) in the CMCR register, regardless of the PORTD4 and DDD4 setting, and PORTD4, DDD4 and PIND4 all read 0.

PCINT20, Pin Change Interrupt Source 20: The PD4 pin can serve as an external interrupt source.

INT1/PCINT19 – Port D, Bit 3
INT1: External Interrupt Source 1. The PD3 pin can serve as an external interrupt source.

The pin has to be configured as an input (DDD3 set (zero)) to serve this function.

PCINT19, Pin Change Interrupt Source 19: The PD3 pin can serve as an external interrupt source.

INT0/PCINT18 – Port D, Bit 2
INT0: External Interrupt Source 1. The PD2 pin can serve as an external interrupt source.

The pin has to be configured as an input (DDD2 set (zero)) to serve this function.

PCINT18, Pin Change Interrupt Source 18: The PD2 pin can serve as an external interrupt source.

T3I/PCINT17 – Port D, Bit 1
T3I: External Input Clock for Timer1, Timer2 or Timer3.

The pin has to be configured as an input (DDD1 set (zero)) to serve this function.

PCINT17, Pin Change Interrupt Source 17: The PD1 pin can serve as an external interrupt source.

T2ICP/PCINT16 – Port D, Bit 0
T2ICP – Timer2 Input Capture Pin: The PD0 pin can act as an external input capture pin for timer/counter2. The pin has to be configured as an input (DDD0 set (zero)) to serve this function.

PCINT16, Pin Change Interrupt Source 16: The PD0 pin can serve as an external interrupt source.

Table 3-28 and Table 3-29 places the alternate functions of port D in relation to the overriding signals shown in Table 3-21 on page 50.

Table 3-28. Overriding Signals for Alternate Functions in PD7..PD4

Signal Name	PD7/SDIN/PCINT23	PD6/T2O2/PCINT15	PD5/T2O1/PCINT14	PD4/ECIN1/PCINT13
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	SPI (T2M[3..0] = 0b1001)	0	0	CMM[1..0] × ECINS × CSS
DDOV	0	0	0	0
PVOE	0	T2O2 ENABLE	T2O1 ENABLE	0
PVOV	0	T2O2	T2O1	0
DIEOE	PCINT23 × PCIE2	PCINT22 × PCIE2	PCINT21 × PCIE2	PCINT20 × PCIE2
DIEOV	1	1	1	1
DI	PCINT23 INPUT SDIN INPUT	PCINT22 INPUT	PCINT21 INPUT	PCINT20 INPUT ECIN1 INPUT
AIO	–	–	–	–

Table 3-29. Overriding Signals for Alternate Functions in PD3..PD0

Signal Name	PD3/INT1/PCINT19	PD2/INT0/PCINT18	PD1/T3I/PCINT17	PD0/T2ICP/PCINT16
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	PCINT19 × PCIE2 + INT1 ENABLE	PCINT18 × PCIE2 + INT0 ENABLE	PCINT17 × PCIE2	PCINT16 × PCIE2
DIEOV	1	1	1	1
DI	PCINT19 INPUT INT1 INPUT	PCINT18 INPUT INT0 INPUT	PCINT17 INPUT T3I INPUT	PCINT16 INPUT T2ICP INPUT
AIO	–	–	–	–

3.12.4 Register Description for I/O Ports

3.12.4.1 MCU Control Register – MCUCR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 4 – PUD: Pull-Up Disable
When this bit is written to one, the pull-ups in the I/O ports are disabled, even if the DDxn and PORTxn registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See Section 3.12.2.1 “Configuring the Pin” on page 46 for more details about this feature.

3.12.4.2 Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.3 Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.4 Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.5 Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.6 Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.7 Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.8 Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.9 Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.12.4.10 Port D Input Pins Address – PIND

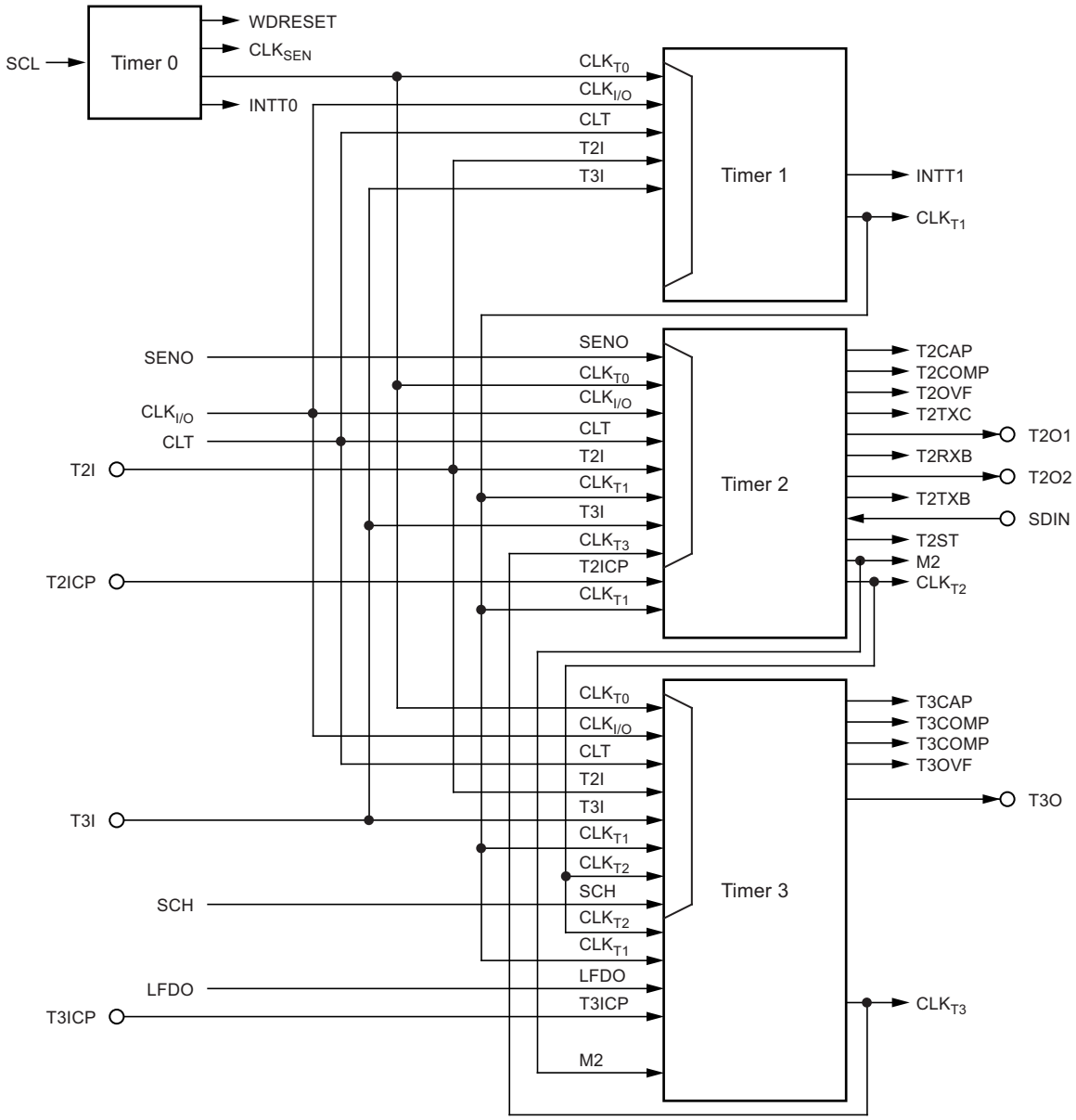
Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.13 Timer Module

3.13.1 Overview

The timer module includes timer0 with watchdog, timer1, timer2 and timer3. The timer module is shown in Figure 3-26.

Figure 3-26. Universal Timer/Counter Module Structure



The timer module offers many operating modes for generating intervals, counting events, interrupts, PWM, carrier frequency burst modulation, Manchester modulation, and bi-phase modulation. It uses a variety of clock sources and many combination modes that enable a lot of interactions between the single timers. The timer module is fully controlled by external Atmel® AVR® MCU, using a set of control registers available in the MCU I/O address space.

3.13.2 Accessing 16-Bit Registers

Before describing the different timers in detail more information must be provided about accessing 16-bit registers because the compare registers and capture registers of timer2 (T2ICR, T2COR) and timer3 (T3ICR, T3CORA, T3CORB) are 16-bit registers that can be accessed by the Atmel® AVR® CPU via the 8-bit databus. The 16-bit register must be byte-accessed using two read or write operations. Each 16-bit timer has a single 8-bit register for temporarily storing the high byte of the 16-bit access. The same temporary register is shared between all 16-bit registers within each 16-bit timer. Accessing the low byte triggers the 16-bit read or write operation. When the low byte of a 16-bit register is written by the CPU, the high byte stored in the temporary register and the low byte written are both copied into the 16-bit register in the same clock cycle. When the low byte of a 16-bit register is read by the CPU, the high byte of the 16-bit register is copied into the temporary register in the same clock cycle in which the low byte is read.

Therefore to do a 16-bit write, the high byte must be written before the low byte. For a 16-bit read, the low byte must be read before the high byte.

The following code examples show how to access the 16-bit timer registers assuming that no interrupts update the temporary register. Note that when using “C,” the compiler handles the 16-bit access.

Assembly Code Examples[1]

```
...
; Set T2COR to 0x01FF
Ldi r17, 0x01
Ldi r16, 0xFF
STS T2CORH, r17
STS T2CORL, r16
; Read T2COR into r17:r16
LDS r16, T2CORL
LDS r17, T2CORH
...
```

C Code Examples[1]

```
unsigned int i;
...
/* Set T2COR to 0x01FF */
T2COR = 0x1FF;
/* Read T2COR into i */
i = T2COR;
...
```

Note: 1. The example code assumes that the part-specific header file is included. For I/O registers located in the extended I/O map, “IN,” “OUT,” “SBIS,” “SBIC,” “CBI,” and “SBI” instructions must be replaced with instructions that allow access to extended I/O—“LDS” and “STS” are typically combined with “SBRS,” “SBRC,” “SBR,” and “CBR.”
The assembly code example returns the T2COR value in the r17:r16 register pair.

It is important to notice that accessing 16-bit registers is an atomic operation. If an interrupt occurs between the two instructions accessing the 16-bit register, and the interrupt code updates the temporary register by accessing the same or any other of the 16-bit timer registers, then the result of the access outside the interrupt is corrupted. Therefore, when both the main code and the interrupt code update the temporary register, the main code must disable the interrupts during the 16-bit access.

The following code examples show how to do an atomic read of the T2COR register contents. Reading any of the 16-bit registers can be done based on the same principle.

Assembly Code Example[1]

```
TIM16_ReadT2COR:
; Save global interrupt flag
In r18, SREG
; Disable interrupts
cli
; Read T2COR into r17:r16
LDS r16,T2CORL
LDS r17,T2CORH
; Restore global interrupt flag
Out SREG, r18
ret
```

C Code Example[1]

```
unsigned int TIM16_ReadT2COR( void )
{
unsigned char sreg;
unsigned int i;
/* Save global interrupt flag */
sreg = SREG;
/* Disable interrupts */
_CLI();
/* Read T2COR into i */
i = T2COR;
/* Restore global interrupt flag */
SREG = sreg;
return i;
}
```

Note: 1. The example code assumes that the part-specific header file is included. For I/O registers located in extended I/O map, “IN,” “OUT,” “SBIS,” “SBIC,” “CBI,” and “SBI” instructions must be replaced with instructions that allow access to extended I/O—“LDS” and “STS” are typically combined with “SBRS,” “SBRC,” “SBR,” and “CBR.” The assembly code example returns the T2COR value in the r17:r16 register pair.

The following code examples show how to do an atomic write of the T2COR register contents. Reading any of the 16-bit registers can be done based on the same principle.

Assembly Code Example[1]

```
TIM16_WriteT2COR:
; Save global interrupt flag
In r18, SREG
; Disable interrupts
cli
; Set T2COR to r17:r16
STS T2CORH, r17
STS T2CORL, r16
; Restore global interrupt flag
Out SREG, r18
ret
```

C Code Example[1]

```
void TIM16_WriteT2COR( unsigned int i )
{
unsigned char sreg;
unsigned int i;
/* Save global interrupt flag */
sreg = SREG;
/* Disable interrupts */
_CLI();
/* Set T2COR to i */
T2COR = i;
/* Restore global interrupt flag */
SREG = sreg;
}
```

Note: 1. The example code assumes that the part-specific header file is included. For I/O registers located in the extended I/O map, “IN,” “OUT,” “SBIS,” “SBIC,” “CBI,” and “SBI” instructions must be replaced with instructions that allow access to extended I/O—“LDS” and “STS” are typically combined with “SBRS,” “SBRC,” “SBR,” and “CBR.”
The assembly code example requires that the r17:r16 register pair contains the value to be written to T2COR.

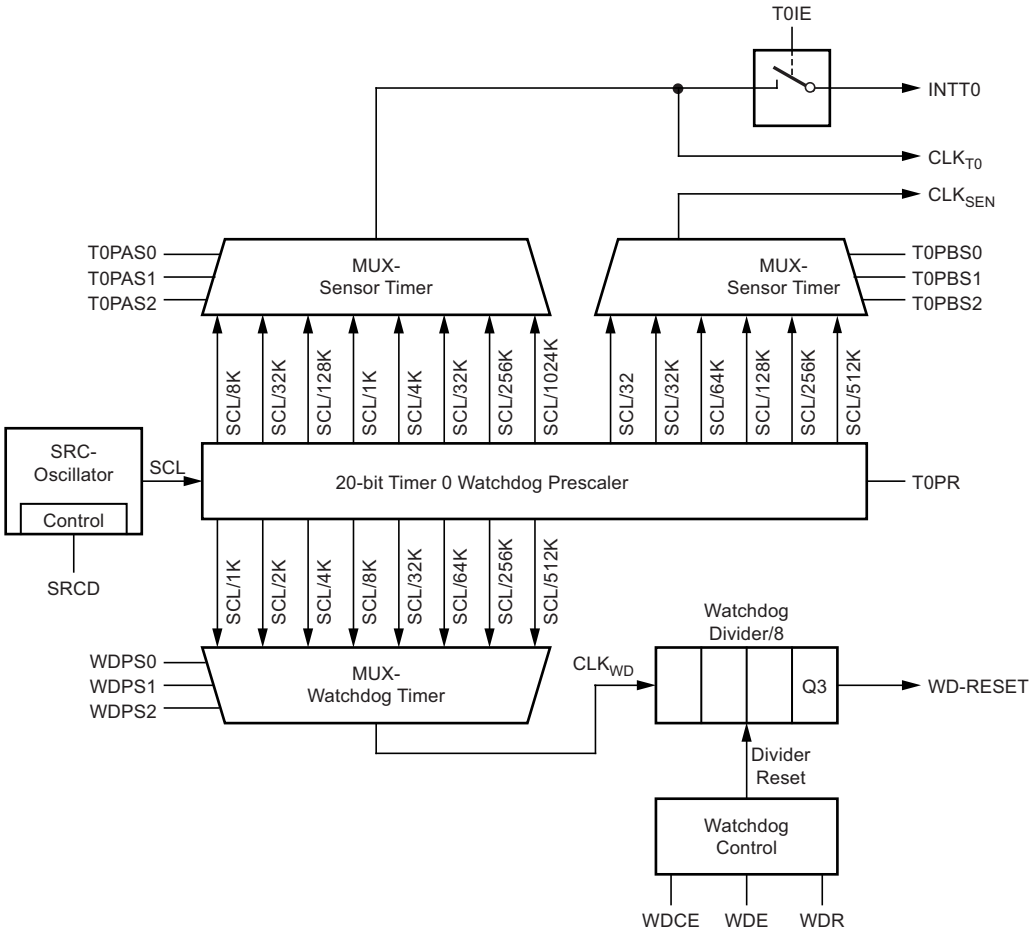
3.13.2.1 Reusing the Temporary High Byte Register

When writing to more than one 16-bit register with the high byte the same for all registers written, the high byte only needs to be written once. However, note that the same rule of atomic operation described previously also applies in this case.

3.13.3 Timer0 with Watchdog/Interval Timer

The timer0 is a watchdog/interval timer which can be used to generate periodical interrupts and used as a prescaler for timer1, timer2, timer3, and the watchdog function.

Figure 3-27. Timer0 Block Diagram



The watchdog/interval timer0 is clocked from a Separately Calibrated On-Chip RC Oscillator (SRC) which runs at f_{SCL} = 90kHz. The SRC and the timer0 can work together as an ultra-low-power watchdog/interval timer stage.

The timer0 consists of a programmable 20-stage divider that is driven by the SCL. The timer output signal (CLK_{T0}) can be used as a prescaler clock and as the source for the timer0 interrupt. The interrupt is able to mask via the T0IE bit and the time interval for the timer output can also be adjusted as shown in Table 3-32 and Table 3-33 on page 66f via the T0PAS[2..0] and T0PBS[2..0] bits in the timer0 control register T0CR.

This timer starts running automatically after any power-on reset. If the watchdog function is not activated, the timer can be restarted by writing a logic one to the T0PR bit in the T0CR register.

The timer0 can also be used as a watchdog timer to prevent a system from stalling. The watchdog divider is a 3-bit counter that is supplied by a separate output clock (CLK_{WD}) of timer0. It generates a system reset when the 3-bit counter overflows. To avoid this, the 3-bit counter must be reset before it overflows. The application software has to accomplish this by executing the Watchdog Reset (WDR) instruction to restart the watchdog counter before the time-out value is reached. The watchdog counter is also reset when it is disabled and when a chip reset occurs. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another reset, the ATA6289 resets and executes from the reset vector. By controlling the watchdog timer0 prescaler, the watchdog reset interval can be adjusted as shown in Table 3-31 on page 65 via the WDPS[2..0] bits in the timer0 watchdog control register WDTCR.

To prevent unintentional disabling of the watchdog or unintentional change of time-out period, two different safety levels are selected by the fuse WDTON as shown in Table 3-30.

Table 3-30. Configuration As a Function of the Fuse Settings of WDTON

WDTON	Safety Level	WDT Initial State	How to Disable the WDT	How to Change Time-Out
Unprogrammed	1	Disabled	Timed sequence	Timed sequence
Programmed	2	Enabled	Always enabled	Timed sequence

3.13.3.1 Watchdog Timer0 Control Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	WDCE	WDE	WDPS2	WDPS1	WDPS0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..5 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 4 – WDCE: WatchDog Change Enable Bit
This bit must be set when the WDE bit is written to logic zero. Otherwise, the watchdog is not disabled. Once written to one, the hardware clears this bit after four clock cycles. Refer to the description of the WDE bit for a watchdog disable procedure. This bit must also be set when changing the watchdog prescaler bits. For more information, see Table 3-30.

Bit 3 – WDE: WatchDog Enable Bit
When the WDE bit is written to logic one, the watchdog timer is enabled, and if the WDE bit is written to logic zero, the watchdog timer function is disabled. WDE can only be cleared if the WDCE bit has logic level one. To further ensure program security, the watchdog setup must follow a special timed sequence. The following procedure must be executed to disable the watchdog timer by clearing the WDE bit:

1. In one operation, write a logic one to WDCE and WDE. A logic one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write logic zero to WDE. The watchdog is then disabled.

In safety level 2, it is not possible to disable the watchdog timer, not even with the algorithm described above. See Table 3-30 on page 63 for more information.

WDE is overridden by WDRF in MCUSR. This means the WDE is always set when the WDRF is set. In order to clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a save startup after the failure.

The following code example shows one assembly and one C function for turning off the watchdog timer. The example includes the disabling of global interrupts so that no interrupt occurs during the execution of these functions.

Assembly Code Example[1]

```
WDT_off:
; Turn off global interrupt
cli
; Reset Watchdog Timer
wdr
; Clear WDRF in MCUSR
in r16, MCUSR
andi r16, (0xff & (0<<WDRF))
out MCUSR, r16
; Write logical one to WDCE and WDE
; Keep old prescaler setting to prevent unintentional time-out
lds r16, WDTCR
ori r16, (1<<WDCE) | (1<<WDE)
sts WDTCR, r16
; Turn off WDT
ldi r16, (0<<WDE)
sts WDTCR, r16
; Turn on global interrupt
sei
ret
```

C Code Example[1]

```
void WDT_off(void)
{
__disable_interrupt();
__watchdog_reset();
/* Clear WDRF in MCUSR */
MCUSR &= ~(1<<WDRF);
/* Write logical one to WDCE and WDE */
/* Keep old prescaler setting to prevent unintentional time-out */
WDTCR |= (1<<WDCE) | (1<<WDE);
/* Turn off WDT */
WDTCR = 0x00;
__enable_interrupt();
}
```

Note: 1. See Section 3.4 “About Code Examples” on page 9.

If the watchdog is accidentally enabled, for example, by a runaway pointer or brown-out condition, the device is reset and the watchdog timer remains enabled. If the code is not set up to handle the watchdog, this might lead to an eternal loop of time-out resets. To avoid this situation, the application software should always clear the Watchdog System Reset Flag (WDRF) and the WDE control bit in the initialization routine, even if the watchdog is not in use.

The following code example shows one assembly and one C function for changing the time-out value of the watchdog timer. The watchdog timer should be reset before any change of the WDPS bits because a change in the WDPS bits can result in a time-out when switching to a shorter time-out period:

Assembly Code Example[1]

```
WDT_Prescaler_Change:
; Turn off global interrupt
cli
; Reset Watchdog Timer
wdr
; Start timed sequence
lds r16, WDTCR
ori r16, (1<<WDCE) | (1<<WDE)
sts WDTCR, r16
; -- Got four cycles to set the new values from here -
; Set new prescaler(time-out) value = 64K cycles (~0.5 s)
ldi r16, (1<<WDE) | (1<<WDPS2) | (1<<WDPS0)
sts WDTCR, r16
; -- Finished setting new values, used 2 cycles -
; Turn on global interrupt
sei
```

C Code Example[1]

```
void WDT_Prescaler_Change(void)
{
__disable_interrupt();
__watchdog_reset();
/* Start timed equence */
WDTCR |= (1<<WDCE) | (1<<WDE);
/* Set new prescaler(time-out) value = 64K cycles (~0.5 s) */
WDTCR = (1<<WDE) | (1<<WDPS2) | (1<<WDPS0);
__enable_interrupt();
}
```

Note: 1. See Section 3.4 “About Code Examples” on page 9.

Bits 2..0 – WDPS2..0: WatchDog Prescaler Select Bits 2 - 0

The WDPS2, WDPS1 and WDPS0 bits determine the watchdog timer prescaling clock output (CLK_{WD}) when the watchdog timer is enabled. The different prescaling values and their corresponding time-out periods are shown in Table 3-31.

Table 3-31. Watchdog Timer Prescaler Select Bit Description

WDPS2	WDPS1	WDPS0	Number of Oscillator Cycles (SCL)	Total Numbers of WDT Cycles 8 × SCL = WDT	Typical Time-Out at V_{CC} = 3V/25°C and T_{SCL} = 1 / 90kHz
0	0	0	1K cycles	8 × 1K = 8K cycles	90ms
0	0	1	2K cycles	8 × 2K = 16K cycles	180ms
0	1	0	4K cycles	8 × 4K = 32K cycles	365ms
0	1	1	8K cycles	8 × 8K = 64K cycles	730ms
1	0	0	32K cycles	8 × 32K = 256K cycles	2.9s
1	0	1	64K cycles	8 × 64K = 512K cycles	5.8s
1	1	0	256K cycles	8 × 256K = 1024K cycles	23s
1	1	1	512K cycles	8 × 512K = 4096K cycles	47s

3.13.3.2 Timer0 Control Register – T0CR

Bit	7	6	5	4	3	2	1	0	
	T0PBS2	T0PBS1	T0PBS0	T0PR	T0IE	T0PAS2	T0PAS1	T0PAS0	T0CR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7 to 5 – T0PBS2..0: Timer0 Prescaler B Select Bits 2 to 0
The T0PBS2, T0PBS1 and T0PBS0 bits determine the timer0 prescaling clock output (CLK_{SEN}) as well as the mode of the motion sensor. The different prescaling values and their corresponding time-out periods are shown in Table 3-32.

Table 3-32. Timer0 Prescaler B Select Bit Description

T0PBS2	T0PBS1	T0PBS0	Motion Sensor Mode	Number of Oscillator Cycles (SCL)	Typical Time-Out at V_{CC} = 3V/25°C and $T_{SCL} \approx$ 1 / 90kHz for CLK_{SEN}
0	0	0	Disabled	None	Disabled
0	0	1	Running	32K cycles	0.365s
0	1	0	Running	64K cycles	0.730s
0	1	1	Running	128K cycles	1.46s
1	0	0	Running	256K cycles	2.9s
1	0	1	Running	512K cycles	5.8s
1	1	0	Running	32 cycles (used for test)	Reserved (355µs)
1	1	1	Disabled	Reserved	Reserved

Bit 4 – T0PR: Timer0 Prescaler Reset Bit
Writing T0PR to one restarts the timer0 watchdog prescaler. If T0PR is written to one, the hardware clears this bit after four clock cycles. Only if the watchdog function is disabled can the timer0 watchdog prescaler be restarted. Additionally, the T0PR bit should only be set to one if the motion sensor functionality is disabled via the SMEN bit located in the SCR register. Otherwise a malfunction could occur in the motion sensor circuitry due to an asynchronous reset of timer0.

Bit 3 – T0IE: Timer0 Interrupt Enable Bit
Writing T0IE to one enables an interval timer interrupt if the I bit in SREG is set. Writing T0IE to zero disables the interrupt. The corresponding interrupt vector is executed when the T0F flag, located in T10IFR, is set.

Bits 2..0 – T0PAS2..0: Timer0 Prescaler A Select Bits 2 – 0
The T0PAS2, T0PSA1 and T0PAS0 bits determine the timer0 prescaling clock output (CLK_{T0}). The different prescaling values and their corresponding time-out periods are shown in Table 3-33.

Table 3-33. Timer0 Prescaler A Select Bit Description

T0PAS2	T0PAS1	T0PAS0	Number of Oscillator Cycles (SCL)	Typical Time-Out at V_{CC} = 3V/25°C and $T_{SCL} \approx$ 1 / 90kHz for CLK_{T0}
0	0	0	8 cycles	89µs
0	0	1	32 cycles	350µs
0	1	0	128 cycles	1.4ms
0	1	1	1K cycles	11.4ms
1	0	0	4K cycles	45.5ms
1	0	1	32K cycles	365ms
1	1	0	256K cycles	2.9s
1	1	1	1024K cycles	11.65s

3.13.3.3 Timer1/0 Interrupt Flag Register – T10IFR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	**T1F**	**T0F**	**T10IFR**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..2 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and always read as zero.

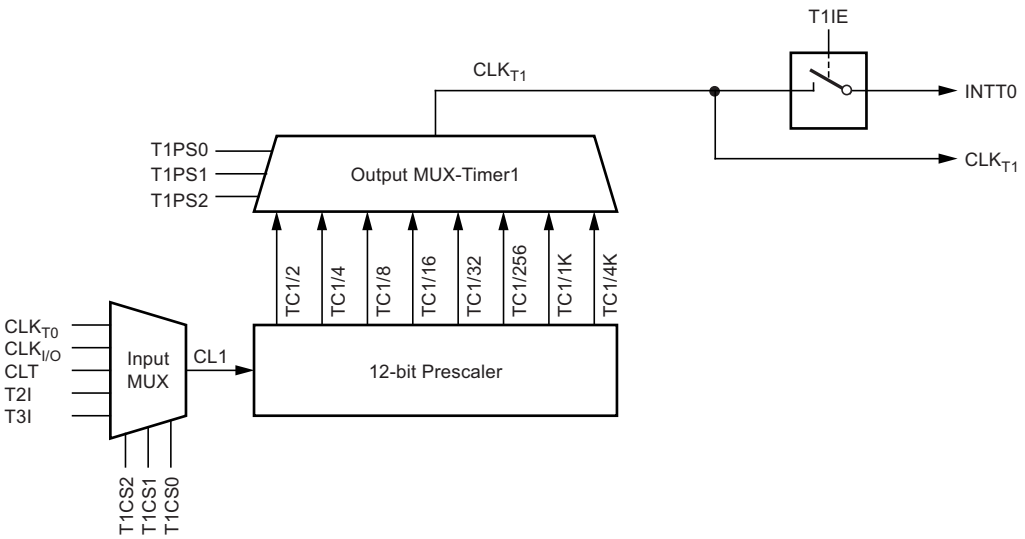
Bit 1 – T1F: Timer1 Flag Bit
When the interval timer in timer1 generates an output clock pulse (CLK_{T1}), the T1F bit is set (one). If the I bit in SREG and the T1IE bit is set (one) at the T1CR, the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it.

Bit 0 – T0F: Timer0 Flag Bit
When the interval timer in timer0 generates an output clock pulse (CLK_{T0}), the T0F bit is set (one). If the I bit in SREG and the T0IE bit is set (one) at the T0CR, the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it.

3.13.4 Timer1

Timer1 is an interval timer which can be used to generate periodical interrupts and used as a prescaler for timer2 and timer3. Timer1 consists of a programmable 12-bit divider that allows input clock (CL1) to be driven by the timer0 output clock (CLK_{T0}), I/O clock ($CLK_{I/O}$), timer clock (CLT), the external input clock (T2I), and the external input clock (T3I). The three bits T1CS[2..0] select the input clock (CL1) for timer1. The timer output signal can be used as a prescaler clock and as the source for the timer1 interrupt. The interrupt is maskable via the T1IE bit and also the time interval for the timer output can be adjusted as shown in Figure 3-28 via the T1PS[2..0] bits in the timer1 control register T1CR. The timer interrupt flag bit (T1F) is located in the T10IFR register.

Figure 3-28. Timer1 Block Diagram



3.13.4.1 Timer1 Control Register – T1CR

Bit	7	6	5	4	3	2	1	0	
	T1IE	-	T1CS2	T1CS1	T1CS0	T1PS2	T1PS1	T1PS0	T1CR
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – T1IE: Timer1 Interrupt Enable Bit
Writing T1IE to one enables an interval timer interrupt if the I bit in SREG is set. Writing T1IE to zero disables the interrupt. The corresponding interrupt vector is executed when the T1F flag, located in T10IFR, is set.

Bit6 – Res: Reserved Bit
This is a reserved bit on the ATA6289 and is always read as zero.

Bits 5..3 – T1CS2..0: Timer1 Clock Select Bits 2 – 0
The T1CS2, T1CS1 and T1CS0 bits select the input clock (CL1) of the timer1 as shown in Table 3-34.

Table 3-34. Timer1 Input Clock Select Bit Description

T1CS2	T1CS1	T1CS0	Input Clock (CL1) of 12-Bit Prescaler
0	0	0	Disable (no clock source)
0	0	1	CLK_{T0}
0	1	0	$CLK_{I/O}$
0	1	1	CLT
1	0	0	T2I
1	0	1	T3I
1	1	0	Reserved
1	1	1	Reserved

Bits 2..0 – T1PS2..0: Timer1 Prescaler Select Bits 2 - 0
The T1PS2, T1PS1 and T1PS0 bits determine the timer1 prescaling clock output (CLK_{T1}). The different prescaling values are shown in Table 3-35.

Table 3-35. Timer1 Prescaler Select Bit Description

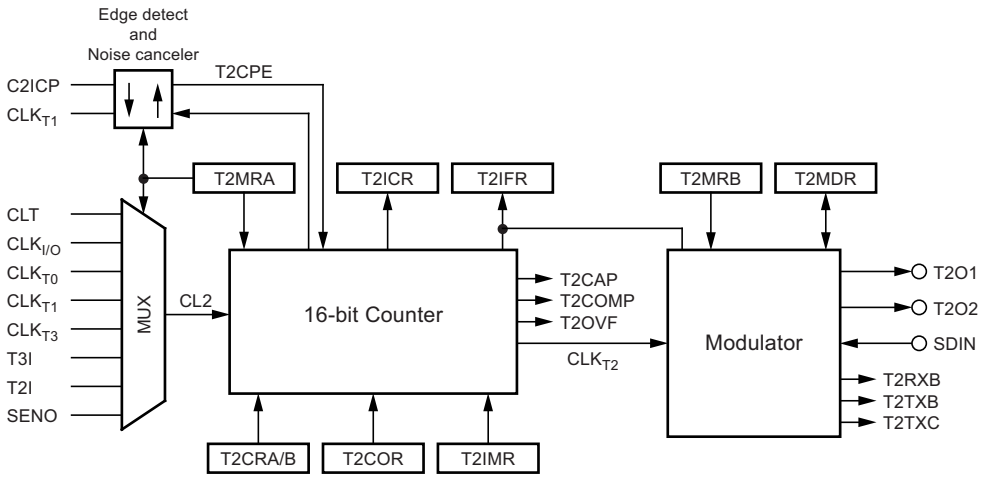
T1PS2	T1PS1	T1PS0	Divider
0	0	0	CL1/2
0	0	1	CL1/4
0	1	0	CL1/8
0	1	1	CL1/16
1	0	0	CL1/32
1	0	1	CL1/256
1	1	0	CL1/1K
1	1	1	CL1/4K

3.13.5 Timer2/Counter2

16-bit timer/counter with capture/compare modes and continuous bit-stream modulator

- True 16-bit design (i.e., allows 16-bit PWM)
- Eight different selectable input clocks for timer2/counter2
- One output compare unit
- Bit-stream modulator stage
- One input capture unit
- Input capture noise canceler
- Clear timer on compare match or capture event
- Different programmable PWM period
- Frequency generator
- External event counter
- Six independent interrupt sources (T2CAP, T2COM, T2OVF, T2RXB, T2TXB, T2TXC)

Figure 3-29. Timer2 Block Diagram



Timer2 consists of a 16-bit up counter stage with compare register (T2COR) and capture register (T2ICR). The timer can be used as an event counter, as a timer and as a signal generator. The counter can be driven by internal and external clock sources. For the capture signals (T2ICP, CLK_{T1}) it has a programmable edge-sensitive input with digital filtering unit (noise canceler) for reducing the chance of capturing noise spikes. In the capture mode the counter value can be captured by a programmable capture event from the internal timer1 output (CLK_{T1}), by an external event (T2ICP) at the capture input pin or with a software capture event by setting the T2SCE bit. The counter is readable via its capture register while it is running. Its output has a modulator stage with two output pins that not only allows the generation of pulses but also the generation of bi-phase code, Manchester code, or PWM code and together with the data output of the Synchronous Serial Interface (SSI) a continuous serial stream of data. The SSI allows additional synchronous data transfer between the ATA6289 and peripheral devices.

If the “ATA6289” is combined with ATA5757 as a stacked die, the pins PD5/T2O1 and PD6/T2O2 are internally bonded with the “ASK” and “FSK” pins of ATA5757 respectively. See Figure 2-2 on page 4 for more information on how to use the SSI of the timer2 modulator stage to modulate the ATA5757 (ASK or FSK modulation).

The Timer2 Compare Register (T2COR) and Timer2 Input Capture Register (T2ICR) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in Section 3.13.2 “Accessing 16-Bit Registers” on page 59.

The timer2 control/mode registers (T2CRA, T2CRB, T2MRA, T2MRB, T2MDR) are 8-bit registers and have no CPU access restrictions.

The comparator output is controlled by a control register (T2CRA) and contains mask bits for actions (counter reset, output toggle, timer interrupt) which can be triggered by a compare match event or the counter overflow. This architecture enables the timer for various modes.

The functions of the output modulator are controlled by two registers (T2MRB, T2MDR). The SSI Transmit Data Buffer (TXB) register and the SSI Receive Data Buffer (RXB) register share the same I/O address referred to the SSI Data Register (T2MDR). The modes of timer2 operations as well as the modulator I/O pins are controlled by the T2MRB register.

Interrupt request (referred to as “Int.Req.”) signals are all visible in the Timer Interrupt Flag Register (T2IFR). All interrupts are individually masked with the Timer Interrupt Mask Register (T2IMR).

The counter2 input clock (CL2) can be supplied via the I/O clock ($CLK_{I/O}$), the external input clock (T2I), the external input clock (T3I), the timer0 output clock (CLK_{T0}), the timer1 output clock (CLK_{T1}), the timer3 output clock (CLK_{T3}), the output clock of the sensor interface block (SENO), or the timer clock (CLT).

The Output Compare Register (T2COR) is compared with the timer/counter value at all times. The result of the compare can be used by the modulator stage to generate a PWM or variable frequency output on the two selectable modulator output pins (T2O1/T2O2).

In some modes of operation the TOP value or maximum timer/counter value can be defined by either of the T2MRA registers. Timer2 has three selectable fixed TOP values.

3.13.5.1 Definitions

The following definitions are used extensively throughout the section:

Table 3-36. Definitions

Name	Description
MAX	The counter reaches its maximum when it becomes 0xFFFF (decimal 65535).
TOP	The counter reaches the TOP when it becomes equal to the highest value defined by the fixed TOP values—0x00FF, 0x01FF or 0x03FF. The assignment is dependent on the mode of operation.

The timer2 compare data values

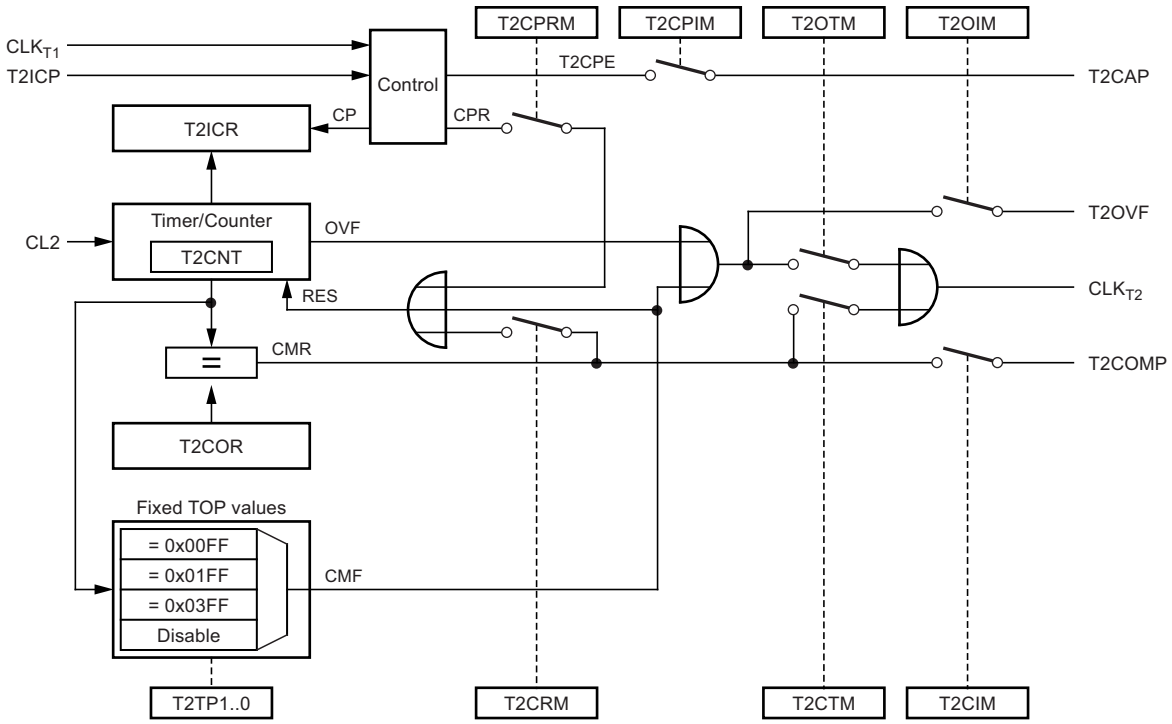
16-bit compare register (T2COR) data value range: $m = y + 1 \rightarrow 0 \leq y \leq 65535$

Three different fixed TOP data values: $n = 0x00FF, 0x01FF \text{ or } 0x03FF$

3.13.5.2 The Counter2 Stage

The counter stage has three input signals (CL2, T2ICP and CLK_{T1}) and four output signals (T2CAP, T2OVF, T2COM and CLK_{T2}). The CL2 supplies the timer/counter (T2CNT) with clocks and the T2ICP signal or CLK_{T1} signal captures the counter value in the capture register. The CLK_{T2} is the output clock and T2CAP, T2OVF and T2COMP are the interrupt request signals of the counter stage.

Figure 3-30. 16-Bit Counter Stage



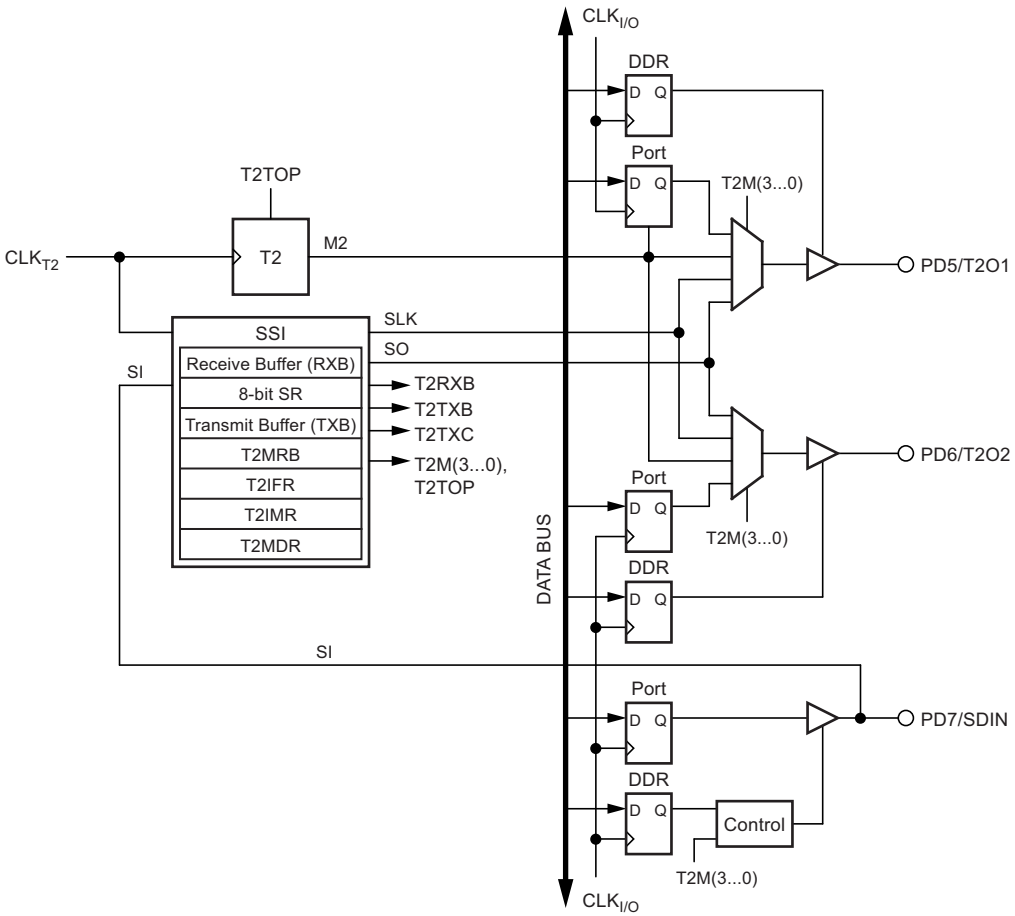
Signal description of the timer/counter2 (internal signals):

CL2	Timer/counter input clock
CLK_{T1}	Timer1 output clock
T2ICP	Timer2 input capture signal
T2CAP	Timer/counter capture event interrupt
T2OVF	Timer/counter overflow interrupt
T2COM	Timer/counter compare match interrupt
CLK_{T2}	Timer/counter stage output clock
CMR	Timer/counter compare register match
CMF	Timer/counter compare fixed TOP match
CPR	Timer/counter reset for capture event
RES	Timer/counter reset (clear all bits)
OVF	Timer/counter overflow
T2CPE	Timer/counter capture signal

3.13.5.3 Timer2 Output Modes

The modulator consists of a toggle flip-flop (T2), a Synchronous Serial Interface (SSI) with control logic and registers (see Section 3.13.5.5 “Modulator Synchronous Serial Interface (SSI)” on page 74), and three external interface pins (T2O1, T2O2, SDIN). T2O1 and T2O2 are output pins and SDIN is an input pin. The modulator has different modes which can be selected with the T2M[3..0] bits in the T2MRB register. Figure 3-31 shows a simplified schematic of how the setting of the T2M[3..0] bits affects the functionality of the I/O ports. Only the parts of the general I/O port control registers (DDR and port) that are affected by the T2M[3..0] bits are shown. The Figure 3-31 shows the timer2 modulator stage.

Figure 3-31. Timer2 Modulator Stage



Signal description (internal signals):

CLK_{T2}	Timer/counter2 stage clock output
SI	SSI serial data input
M2	Output signal of the T2 (toggle flip-flop)
SDIN	External serial data input
SCLK	SSI shift clock output
SO	SSI serial data output
T2RXD	SSI receive buffer full interrupt
T2TXD	SSI transmit buffer empty interrupt
T2TXC	SSI transmit complete interrupt

The general I/O port function is overridden by the timer2 modulator stage in case the selected timer mode is needed for the output pins. However, the pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The data direction register bits for the T2O1 and T2O2 pins (DDD5, DDD6) must be set as output before the value is visible at the pin. When the modulator has selected the SPI mode of the SSI, PD7 pin is configured as an input regardless of the DDD7 bit setting. The port override function is generally independent of the modulator mode, but there are some exceptions. Refer to Table 3-41 on page 81 for more details.

If the “ATA6289” is combined with ATA5757 as a stacked die, the pins PD5/T2O1 and PD6/T2O2 are internally bonded with the “ASK” and “FSK” pins of the Atmel® ATA5757 respectively, see Figure 2-2 on page 4. This makes it possible to use the SSI of the timer2 modulator stage to modulate the ATA5757 (ASK or FSK modulation).

3.13.5.4 Modulator Toggle Flip-Flop (T2)

The toggle flip-flop (T2) consists of a flip-flop with a preset input signal (T2TOP), an input clock (CLK_{T2}) and an output signal (M2). The T2TOP bit at T2MRB register allows the programmer to initialize the toggle output flip-flop signal (M2) only if the timer2 is not running (T2E = '0'). The output signal (M2) was negated with every rising edge of the input clock (CLK_{T2}). The Figure 3-32 shows the toggle flip-flop (T2).

Figure 3-32. Toggle Flip-Flop (T2)

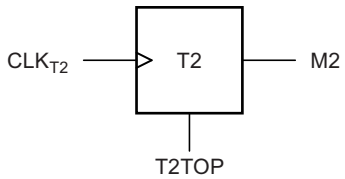
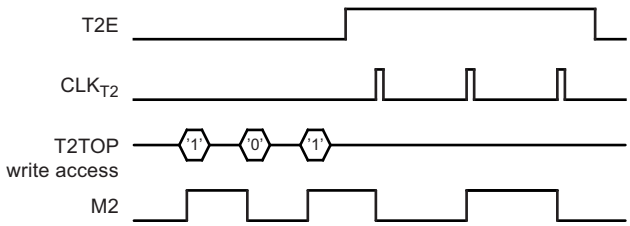


Figure 3-33. Example of Toggle Flip-Flop (T2)



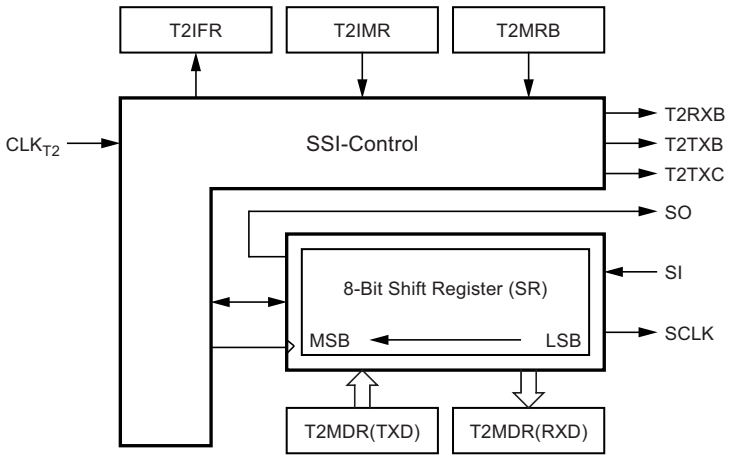
3.13.5.5 Modulator Synchronous Serial Interface (SSI)

The Synchronous Serial Interface (SSI) allows synchronous data transfer between the ATA6289 and the peripheral devices, and also the generation of bi-phase code, Manchester code or PWM code together with the serial data output into a continuous serial stream of data. The SSI consists of an 8-bit shift register (SR), an SSI I/O data register (T2MDR), a mode register (T2MRB), a status register (T2IFR), an interrupt mask register (T2IMR), an input clock (CLK_{T2}), two serial data I/O lines (SI and SO), a shift clock I/O line (SCLK), and three different interrupt request signals (T2RXB, T2TXB, T2TXC). Figure 3-34 shows the SSI.

The SSI includes following features:

- Full-duplex, three-wire synchronous data transfer
- Only master operation
- MSB first data transfer
- Generation of a continuous serial stream of data
- End of transmission interrupt flag

Figure 3-34. Synchronous Serial Interface (SSI)



The SSI contains an 8-bit shift register with two associated 8-bit buffers—the receive buffer T2MDR (RXD) to capture incoming serial data and a transmit buffer T2MDR (TXD) to store the data for the serial data output. Both buffers share the same I/O addresses labeled as the timer2 modulator data register or T2MDR and can be directly accessed by software. The SSI automatically controls the data transfer between transmit and receive buffer and the 8-bit shift register. This makes it possible to support single byte transfers or continuous bit streams.

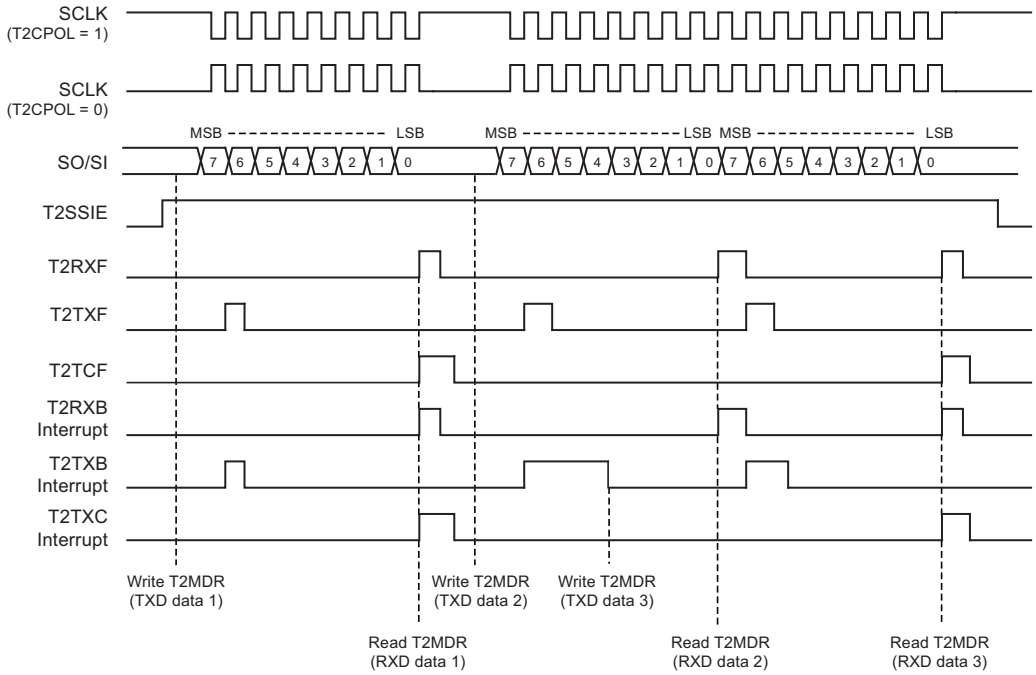
The SSI is always master. The required clock for the data interchange is accessible on the SCLK line from the timer2/counter2 stage output clock (CLK_{T2}). SCLK is half the clock of CLK_{T2}. With this additional division by 2 a duty cycle of 50% for SCLK can be ensured which is important for the SSI data transfer (see Figure 3-36 on page 80). The data is always shifted from master to slave on the Serial Data Output (SO) line and from slave to master on the Serial Data Input (SI) line. Serial data is organized in 8-bit telegrams which are shifted with the Most Significant Bit (MSB) first.

At the beginning of a telegram, the SSI control loads the transmit buffer into the shift register and immediately begins shifting data out. At the same time, incoming data is shifted into the shift register. This incoming data is automatically loaded into the receive buffer when the complete telegram has been received. In that way data can be simultaneously received and transmitted.

The system is double buffered in the transmit direction and double buffered in the receive direction. When receiving data, however, a received character must be read from the SSI Data Register (T2MDR) before the next character has been completely shifted in. Otherwise, the first byte is lost.

Before data can be transferred, the SSI must first be activated. This is performed by means of the SSI enable control bit (T2SSIE in the T2MRB register). There are two combinations of SCLK polarity with respect to serial data. These combinations are determined by the control bit (T2CPOL in the T2MRB register). The SSI has three status flags (T2RXF, T2TXF and T2TCF) in the status register (T2IFR) as well as three interrupt mask bits (T2RXIM, T2TXIM and T2TCIM) in the interrupt mask register (T2IMR). The status of the SSI buffer registers is shown by the T2TXF bit for the transmit buffer register (TXD) and the T2RXF bit for receive buffer register (RXD). The T2TCF bit indicates the present status of the serial communication. Figure 3-35 shows a sample transmit/receive operation by the SSI.

Figure 3-35. Example of Transmit/Receive Operation



For a serial data stream without a gap you must write T2MDR (TXD data) just after a T2TXB interrupt or after a TXF flag is set. For sending data byte-wise you must write T2MDR (TXD data) after a T2TXC interrupt or after the T2TCF flag is set.

3.13.5.6 Timer2 Registers

Timer2 has five control/mode registers, an interrupt flag register, a modulator receive and transmit buffer register, a 16-bit compare register, and a 16-bit capture register.

3.13.5.7 Timer2 Control Register A – T2CRA

Bit	7	6	5	4	3	2	1	0	
	T2E	**T2TS**	**T2ICS**		**T2CRM**	**T2CR**	**T2CTM**	**T2OTM**	**T2CRA**
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – T2E: Timer2 Enable Bit
This bit controls the timer2 block. The T2E bit must be written to logic one to enable timer2, and if the T2E bit is written to logic zero, the timer2 is disabled.

Bit 6 – T2TS: Timer2 Toggle with Start Bit
If the modulator output of timer2 is toggled, the T2TS bit must be written to logic one when the timer is enabled with T2E. If the T2TS bit is written to logic zero, the modulator output of timer2 is not toggled with the timer enabled.

Bit 5 – T2ICS: Timer Input Capture Select Bit
The T2ICS bit selects the input capture signal of timer2 as shown in Table 3-37.

Table 3-37. Input Capture Signal Select Bit Description

T2ICS	Description
0	CLK_{T1}
1	T2ICP

Bit4 – Res: Reserved Bit
This bit is a reserved bit on the ATA6289 and is always read as zero.

Bit 3 – T2CRM: Timer2 Compare Reset Mask Bit
If a match of the counter with the compare register occurs, the T2CRM bit must be written to logic one to enable the counter reset. If the T2CRM bit is written to logic zero, the counter reset is disabled.

Bit 2 – T2CR: Timer2 Counter Reset
The T2CR bit resets counter2 asynchronously if this bit is set to logic one.

Bit 1 – T2CTM: Timer2 Compare Toggle Mask Bit
The T2CTM bit must be written to logic one to enable the compare toggle, and if the T2CTM bit is written to logic zero, the compare toggle is disabled. A match of the counter with the compare register toggles the output flip-flop in the modulator of timer2.

Bit 0 – T2OTM: Timer2 Overflow Toggle Mask Bit
The T2OTM bit must be written to logic one to enable the overflow toggle, and if the T2OTM bit is written to logic zero, the overflow toggle is disabled. A counter overflow toggles the output flip-flop in the modulator of timer2.

3.13.5.8 Timer2 Control Register B – T2CRB

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	T2SCE	T2CRB
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..1 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 0 – T2SCE: Timer2 Software Capture Enable Bit
The T2SCE bit must be written to logic one to generate a software capture event. The T2SCE bit is cleared after the counter value is saved in the capture register. The timer2 counter value is readable via its capture register at run time.

3.13.5.9 Timer2 Modulator Data Register – T2MDR

Bit	7	6	5	4	3	2	1	0	
	T2MDR [7..0]								T2MDR (read)
	T2MDR [7..0]								T2MDR (write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The modulator transmit data buffer register and the modulator receive data buffer register share the same I/O address labeled as modulator data register or T2MDR. The Transmit Data Buffer Register (TXB) is the destination for the data written to the T2MDR register location. Reading the T2MDR register location returns the contents of the Receive Data Buffer Register (RXB).

3.13.5.10 Timer2 Input Capture Register – T2ICR

Bit	7	6	5	4	3	2	1	0	
	T2ICRH [15..8]								T2ICRH
	T2ICRL [7..0]								T2ICRL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The input capture register is updated with the counter value (T2CNT) each time an event occurs at the T2ICP pin, timer1 output clock CLK_{T1} or after a software capture event is generated with the T2SCE bit. The input capture register is a 16-bit register. To ensure that both the high and low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit Temporary High Byte Register (TEMP). This temporary register is shared by all 16-bit registers (see Section 3.13.2 “Accessing 16-Bit Registers” on page 59).

3.13.5.11 Timer2 Compare Register – T2COR

Bit	7	6	5	4	3	2	1	0	
	T2CORH [15..8]								T2CORH
	T2CORL [7..0]								T2CORL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The compare registers contain a 16-bit value that is continuously compared with the counter value (T2CNT). A counter match can be used to generate a compare interrupt, a counter reset, an output clock CLK_{T2}, or to generate a waveform with the modulator at the output pins (T2O1, T2O2).

The compare register is a 16-bit register. To ensure that both the high and low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit TEMP. This temporary register is shared by all 16-bit registers. For more information, see Section 3.13.2 “Accessing 16-Bit Registers” on page 59.

3.13.5.12 Timer2 Interrupt Flag Register – T2IFR

Bit	7	6	5	4	3	2	1	0	
	-	-	T2TCF	T2TXF	T2RXF	T2ICF	T2COF	T2OFF	T2IFR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	1	0	0	0	0	

Bits 7..6 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 5 – T2TCF: Timer2 SSI Transmit Complete Flag Bit
This flag bit is set when the entire frame in the SSI shift register has been shifted out and there is no new data currently present in the transmit buffer (T2MDR). The T2TCF flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The T2TCF flag can generate a transmit complete interrupt (for a description of the T2TCIM bit, see Section 3.13.5.13 “Timer2 Interrupt Mask Register – T2IMR” on page 78.)

Bit 4 – T2TXF: Timer2 SSI Transmit Flag Bit
The T2TXF flag indicates if the transmit buffer (T2MDR) is ready to receive new data. If T2TXF is one, the buffer is empty, and therefore ready to be written. The T2TXF flag can generate a data register empty interrupt (for a description of the T2TXIM bit, see Section 3.13.5.13 “Timer2 Interrupt Mask Register – T2IMR” on page 78.)

T2TXF is set after a reset to indicate that the SSI transmitter is ready.

Bit 3 – T2RXF: Timer2 SSI Receive Flag Bit
This flag bit is set when there is unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the timer2 SSI is disabled (T2SSIE = “0” in the T2MRB register), the receive buffer is cleared and as a result the T2RXF bit becomes zero. The T2RXF flag can be used to generate a receive complete interrupt (for a description of the T2RXIM bit, Section 3.13.5.13 “Timer2 Interrupt Mask Register – T2IMR” on page 78.)

Bit 2 – T2ICF: Timer2 Input Capture Flag Bit
This flag is set (one) and indicates that the timer2/counter2 value has been transferred to the capture register (T2ICR) when a capture event from the T2ICP pin, an output clock of the timer1 (CLK_{T1}) or a software capture event generated by the T2SCE bit occurs. T2ICF is automatically cleared when the interrupt routine is executed. Alternatively, T2ICF can be cleared by writing a logic one to this bit location. The T2ICF flag can be used to generate a timer2 capture interrupt (for a description of the T2CPIM bit, see Section 3.13.5.13 “Timer2 Interrupt Mask Register – T2IMR” on page 78.)

Bit 1 – T2COF: Timer2 COmpare Flag Bit
This flag is set (one) if the timer2/counter2 value (T2CNT) matches the compare register value. The flag (T2COF) is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it. The T2COF flag can be used to generate a timer2 compare interrupt (for a description of the T2CIM bit, see Section 3.13.5.13 “Timer2 Interrupt Mask Register – T2IMR” on page 78.)

Bit 0 – T2OFF: Timer2 OverFlow Flag Bit
This flag is set (one) if the timer2/counter2 Overflow (OVF) occurs. An OVF is generated if the “MAX” value or the “TOP” value of timer2 is reached (see Table 3-36 on page 70). The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one to it. The T2OFF flag can be used to generate a timer2 overflow interrupt (for the description of the T2OIM bit, see the following section.)

3.13.5.13 Timer2 Interrupt Mask Register – T2IMR

Bit	7	6	5	4	3	2	1	0	
	-	-	T2TCIM	T2TXIM	T2RXIM	T2CPIM	T2CIM	T2OIM	T2IMR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..6 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 5 – T2TCIM: Timer2 SSI Transmit Complete Interrupt Mask Bit
Writing this bit to one enables an interrupt on the T2TCF flag. A timer2 SSI Transmit Complete Interrupt (T2TXC) is generated only if the T2TCIM bit is written to one, the global interrupt flag in SREG is written to one and the T2TCF bit in T2IFR is set.

Bit 4 – T2TXIM: Timer2 SSI Transmit Interrupt Mask Bit
Writing this bit to one enables an interrupt on the T2TXF flag. A timer2 SSI Transmit Buffer Empty Interrupt (T2TXB) is generated only if the T2TXIM bit is written to one, the global interrupt flag in SREG is written to one and the T2TXF bit in T2IFR is set.

Bit 3 – T2RXIM: Timer2 SSI Receive Interrupt Mask Bit
Writing this bit to one enables an interrupt on the T2RXF flag. A timer2 SSI Receive Buffer Full Interrupt (T2RXB) is generated only if the T2RXIM bit is written to one, the global interrupt flag in SREG is written to one and the T2RXF bit in T2IFR is set.

Bit 2 – T2CPIM: Timer2 Capture Interrupt Mask Bit
Writing this bit to one enables an interrupt on the T2ICF flag. A timer2 Capture Interrupt (T2CAP) is generated only if the T2CPIM bit is written to one, the global interrupt flag in SREG is written to one and the T2ICF bit in T2IFR is set.

Bit 1 – T2CIM: Timer2 Compare Interrupt Mask Bit
Writing this bit to one enables an interrupt on the T2COF flag. A timer2 Compare Interrupt (T2COM) is generated only if the T2CIM bit is written to one, the global interrupt flag in SREG is written to one and the T2COF bit in T2IFR is set.

Bit 0 – T2OIM: Timer2 Overflow Interrupt Mask Bit
Writing this bit to one enables an interrupt on the T2OFF flag. A timer2 Overflow Interrupt (T2OVF) is generated only if the T2OIM bit is written to one, the global interrupt flag in SREG is written to one and the T2OFF bit in T2IFR is set.

3.13.5.14 Timer2 Mode Register A – T2MRA

Bit	7	6	5	4	3	2	1	0	
	T2TP1	T2TP0	T2CNC	T2CE1	T2CE0	T2CS2	T2CS1	T2CS0	T2MRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7 to 6 – T2TP1..0: Timer 2 ToP Select Bits 1 to 0
The T2TP1 and T2TP0 bits select the fixed TOP compare value of timer2 as shown in Table 3-38.

Table 3-38. Timer2 TOP Select Bit Description

T2TP1	T2TP0	Fixed TOP Value
0	0	Disable fixed value
0	1	0x00FF
1	0	0x01FF
1	1	0x03FF

Bit5 – T2CNC: Timer2 Input Capture Noise Canceler Bit
Setting this bit (one) activates the input capture noise canceler. When the noise canceler is activated, the input from the input capture pin (T2ICP, CLK_{T1}) is filtered. The filter function requires four successive equal valued samples of the T2ICP pin for changing its output. The input capture is therefore delayed by four Counter Clock (CL2) cycles when the noise canceler is enabled.

Bits 4 to 3 – T2CE1..0: Timer2 Capture Edge Select Bits 1 to 0
The T2CE1 and T2CE0 bits select the edge from the capture input signal (T2ICP, CLK_{T1}) of timer2 as shown in Table 3-39.

Table 3-39. Timer2 Capture Edge Select Bit Description

T2CE1	T2CE0	Input Capture Edge Signal (T2ICP) of Timer2
0	0	Disable edge detect
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Bits 2 to 0 – T2CS2..0: Timer2 Clock Select Bits 2 to 0
The T2CS2, T2CS1 and T2CS0 bits select the input clock (CL2) of timer2 as shown in Table 3-40.

Table 3-40. Timer2 Input Clock Select Bit Description

T2CS2	T2CS1	T2CS0	Input Clock (CL2) of TCNT2
0	0	0	CLT
0	0	1	$CLK_{I/O}$
0	1	0	CLK_{T0}
0	1	1	CLK_{T1}
1	0	0	CLK_{T3}
1	0	1	T2I
1	1	0	T3I
1	1	1	SENO

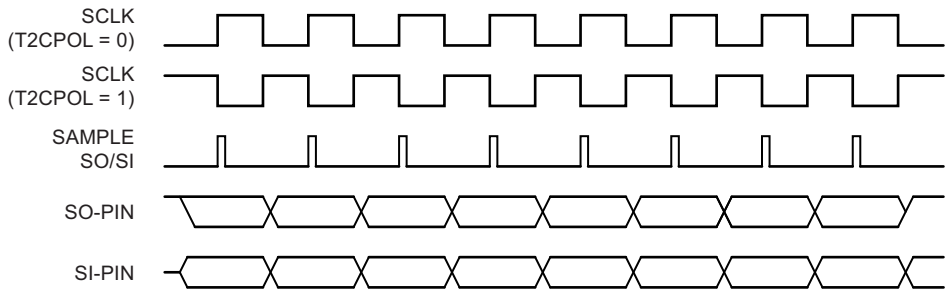
3.13.5.15 Timer2 Mode Register B – T2MRB

Bit	7	6	5	4	3	2	1	0	
	T2SSIE	T2CPOL	-	T2TOP	T2M3	T2M2	T2M1	T2M0	T2MRB
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – T2SSIE: Timer2 SSI Enable Bit
This bit must be set to enable any SSI operation. When this bit is written to low, the SSI is disabled.

Bit 6 – T2CPOL: Timer2 Clock Polarity for SSI Shift Clock
When this bit is written to one, SCLK is high when idle. When T2CPOL is written to zero, SCLK is low when idle. Refer to Figure 3-36 for an example. The T2CPOL functionality is summarized below:

Figure 3-36. SSI Data Transfer Format



Data Modes
There are two combinations of SCLK polarity regarding the serial data. These combinations are determined by the control bit T2CPOL (see Figure 3-36 on page 80). To ensure sufficient time for the data signal to stabilize, sample SO/SI with the opposite edge that is used to shift out for external peripheral devices.

Bit 5 – Res: Reserved Bit
This bit is a reserved bit on the ATA6289 and is always read as zero.

Bit 4 – T2TOP: Timer2 Toggle Output Preset Bit
The T2TOP bit must be written to logic one to set the toggle flip-flop, and if the T2TOP bit is written to logic zero, toggle flip-flop is reset. This bit allows the programmer to preset the toggle output flip-flop in the modulator of timer2.

Note: If T2E = ‘1,’ no output preset is possible

Bits 3..0 – T2M3..0: Timer2 Mode Bits 3 - 0
The T2M[3..0] bits select the modes of timer2 and, in addition, the configuration of the modulator I/O-pins (T2O1, T2O2 and SDIN) as shown in Table 3-41.

Table 3-41. Timer2 Mode Bit Description

Mode	T2M3	T2M2	T2M1	T2M0	Timer2 Mode	Timer2 Modulator I/O Pins		
						T2O1[1]	T2O2[1]	SDIN[1]
1	0	0	0	0	Timer/counter mode	PD5	PD6	PD7
2	0	0	0	1	Toggle mode	M2	PD6	PD7
3	0	0	1	0	Toggle mode	PD5	M2	PD7
4	0	0	1	1	PWM mode	M2	PD6	PD7
5	0	1	0	0	PWM mode	PD5	M2	PD7
6	0	1	0	1	Modulator mode with SSI (ASK modulation of ATA57575[2])	SO	PD6	PD7
7	0	1	1	0	Modulator mode with SSI (FSK modulation of ATA5757[2])	PD5	SO	PD7
8	0	1	1	1	SSI transmit mode	SO	SCLK	PD7
9	1	0	0	0	SSI transmit mode	SCLK	SO	PD7
10	1	0	0	1	SPI mode	SO	SCLK	SI
11	1	0	1	0	Capture mode	PD5	M2	PD7
12	1	0	1	1	Capture mode	M2	PD6	PD7
13	1	1	0	0	Sensor measurement mode together with timer3	PD5	PD6	PD7
14	1	1	0	1	Reserved	PD5	PD6	PD7
15	1	1	1	0	Reserved	PD5	PD6	PD7
16	1	1	1	1	Reserved	PD5	PD6	PD7

Notes: 1. General port I/O: PD5, PD6, PD7
Alternate port function:
M2 → Output signal of T2 (toggle flip-flop)
SO → SSI serial data output
SI → SSI serial data input
SCLK → SSI clock

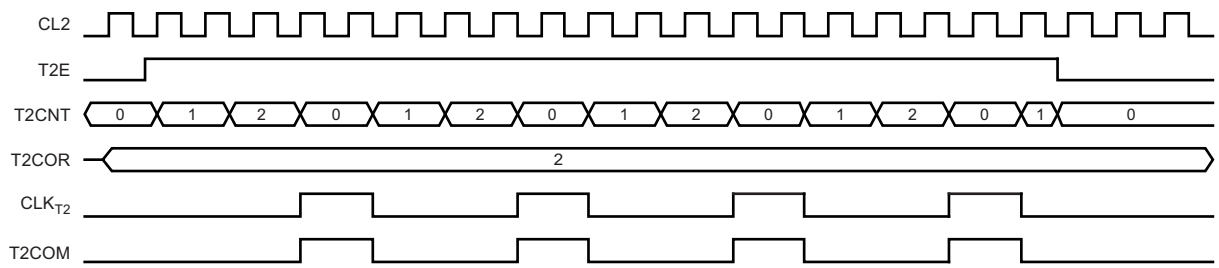
2. The Atmel ATA5757 can be modulated in these modes with the Atmel ATA6286C.

3.13.6 Timer2 Modes

3.13.6.1 Mode 1: Timer/Counter Mode

The timer2/counter2 can be supplied with internal/external clocks. In mode 1 timer2 does not use any modulator I/O pins (T2O1; T2O2, SDIN) but instead works as an interval timer. The port pins (PD5, PD6, PD7) can be used as general digital I/Os. Figure 3-37 shows an example of the timer/counter mode.

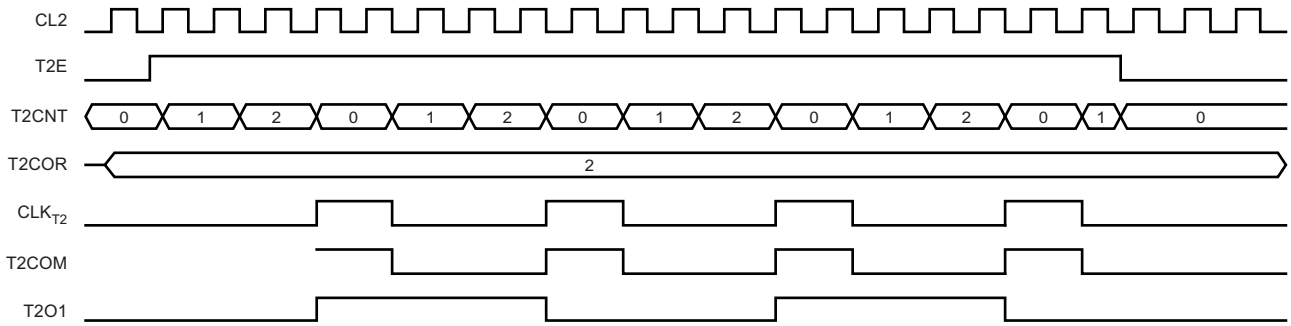
Figure 3-37. Timer/Counter Mode, Timing Diagram



3.13.6.2 Mode 2: Toggle Mode with Modulator Output T2O1 at the PD5 Pin

The timer2/counter2 can be supplied with internal/external clocks. Mode 2 uses only one modulator I/O pin (M2 --> T2O1) to build a waveform generator. The port pins (PD6, PD7) can be used as a general digital I/O. Figure 3-38 shows an example of the toggle mode.

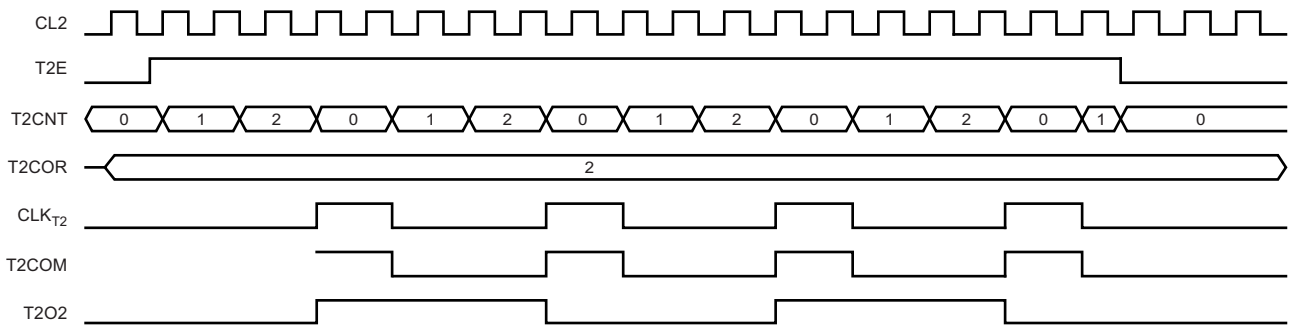
Figure 3-38. Toggle Mode and Modulator Output at the T2O1 Pin, Timing Diagram



3.13.6.3 Mode 3: Toggle Mode with Modulator Output T2O2 at the PD6 Pin

The timer/counter2 can be supplied with internal/external clocks. Mode 3 uses only one modulator I/O pin (M2 --> T2O2) to work as a waveform generator. The port pins (PD5, PD7) can be used as a general digital I/O. Figure 3-39 shows an example of the toggle mode.

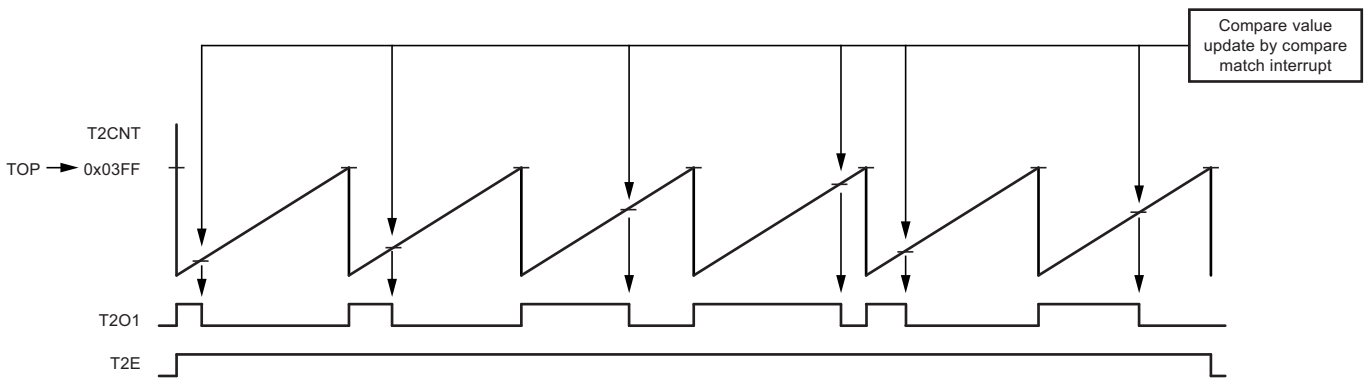
Figure 3-39. Toggle Mode and Modulator Output at the T2O2 Pin, Timing Diagram



3.13.6.4 Mode 4: PWM Mode with Modulator Output T2O1 at the PD5 Pin

Timer2/counter2 can be supplied with internal/external clocks. Mode 4 uses only one modulator I/O pin (M2 --> T2O1) to work as a pulse width modulator. The port pins (PD6, PD7) can be used as a general digital I/O. The timer has four different programmable fixed TOP values which can be selected with the T2TP[1..0] bits in the T2MRA register. Figure 3-40 shows an example of PWM signal generation.

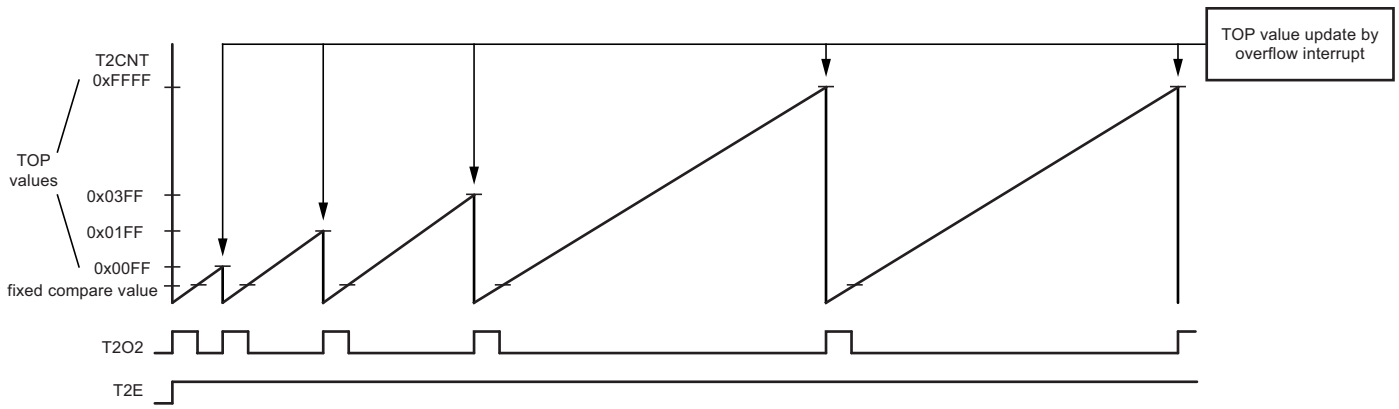
Figure 3-40. PWM Mode and Modulator Output at the T2O1 Pin, Timing Diagram



3.13.6.5 Mode 5: PWM Mode with Modulator Output T2O2 at the PD6 Pin

The timer/counter2 can be supplied with internal/external clocks. Mode 5 uses only one modulator I/O pin (M2 --> T2O2) to work as a pulse width modulator for different frequencies. The port pins (PD5, PD7) can be used as a general digital I/O. The timer has four different programmable fixed TOP values which can be selected with the T2TP[1..0] bits in the T2MRA and the modulator output of timer2 is toggled when the timer is enabled with T2E. Figure 3-41 shows an example of PWM signal generation.

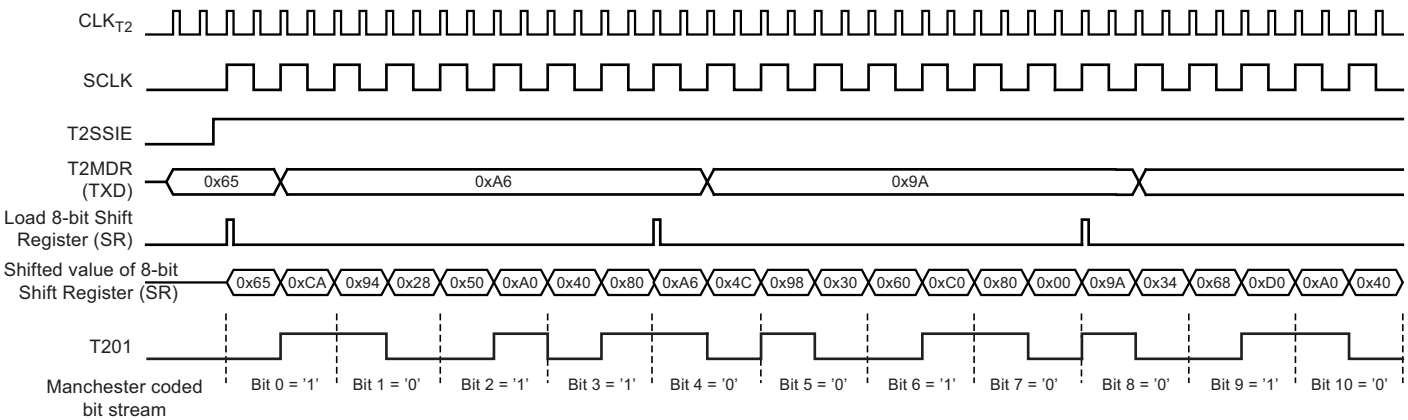
Figure 3-41. PWM Mode and Modulator Output at the T2O2 Pin, Timing Diagram



3.13.6.6 Mode 6: Modulator Mode with SSI and Modulator Output T2O1 at the PD5 Pin

Timer2/counter2 can be supplied with internal/external clocks. Mode 6 uses only one modulator I/O pin (SO --> T2O1). The port pins (PD6, PD7) can be used as a general digital I/O. The timer output clock (CLK_{T2}) can be used to supply the SSI with a shift clock. Together with the continuous serial data stream generated by SSI the modulator mode 6 of the timer allows the generation of a bi-phase code, Manchester code or PWM code. Figure 3-42 shows an example of a Manchester code generation.

Figure 3-42. Modulator Mode and Modulator Output at the T2O1 Pin, Timing Diagram



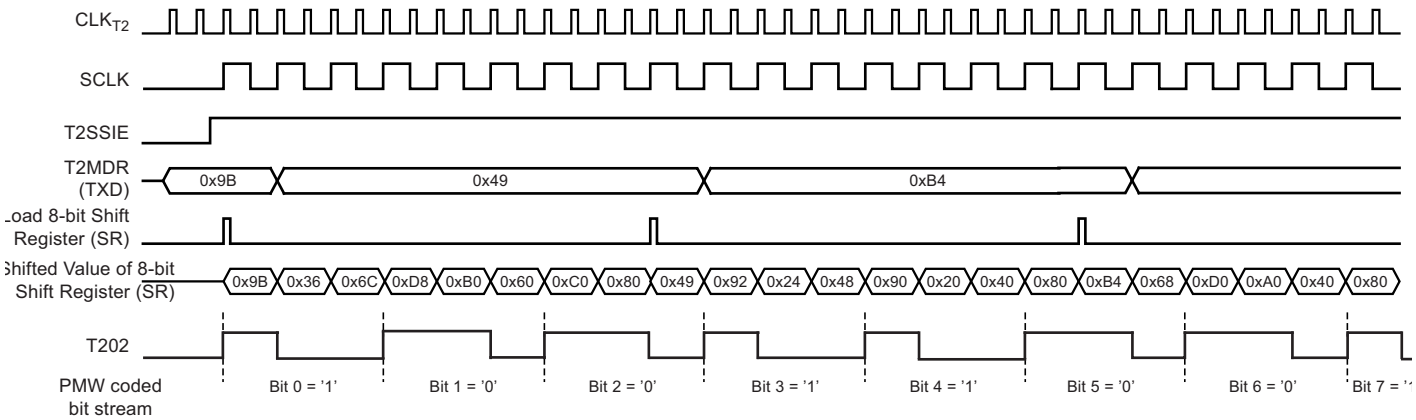
Manchester code: Rising edge in the middle of the bit ⟶ High, falling edge in the middle of the bit ⟶ Low.

At the same time the 8-bit shift register (SR) is loaded with the T2MDR (TXD) data. The first bit (MSB) appears at the T201 output pin.

3.13.6.7 Mode 7: Modulator Mode with SSI and Modulator Output T2O2 at the PD6 Pin

Timer2/counter2 can be supplied with internal/external clocks. Mode 7 uses only one modulator I/O pin (SO --> T2O2). The port pins (PD5, PD7) can be used as a general digital I/O. The timer output clock (CLK_{T2}) can be used to supply the SSI with a shift clock. Together with the continuous serial data stream generated by the SSI the modulator mode 7 of the timer allows the generation of a bi-phase code, Manchester code or PWM code. Figure 3-43 shows an example of a PWM code generation.

Figure 3-43. Modulator Mode and Modulator Output at the T2O2 Pin, Timing Diagram



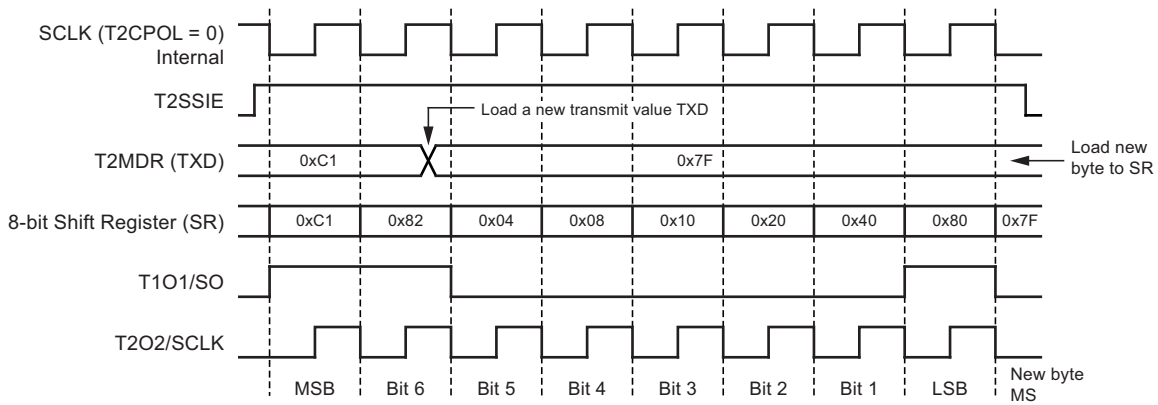
PWM code: Standard practice dictates a 2/3, 1/3 format (of the overall bit period). Each bit consists of a high and a low in a 2/3, 1/3 partition.

At the same time the 8-bit Shift Register (SR) is loaded with the T2MDR (TXD) data. The first bit (MSB) appears at the T202 output pin

3.13.6.8 Mode 8/9: Transmit Mode with SSI and Modulator Outputs T2O1/T2O2 at the PD5/PD6 Pins

Timer2/counter2 can be supplied with internal/external clocks. Mode 8/9 uses two modulator I/O pins (SO --> T2O1 and SCLK --> T2O2 in mode 8 and SCLK --> T2O1 and SO --> T2O2 in mode 9). The port pin PD7 can be used as a general digital I/O. The timer output clock (CLK_{T2}) can be used to supply the SSI with a shift clock. The transmit mode of the timer allows SSI synchronous data transfer between the ATA6289 and peripheral devices. The data is always shifted from master (SSI) to slave on the Serial Data Output (SO) line, synchronized to either the rising or falling edge of the Shift Clock Output (SCLK) line Serial data is organized in 8-bit telegrams which are shifted with the Most Significant Bit (MSB) first to the Serial Data Output (SO) line. Figure 3-44 on page 84 shows an example of a synchronous serial data transmission.

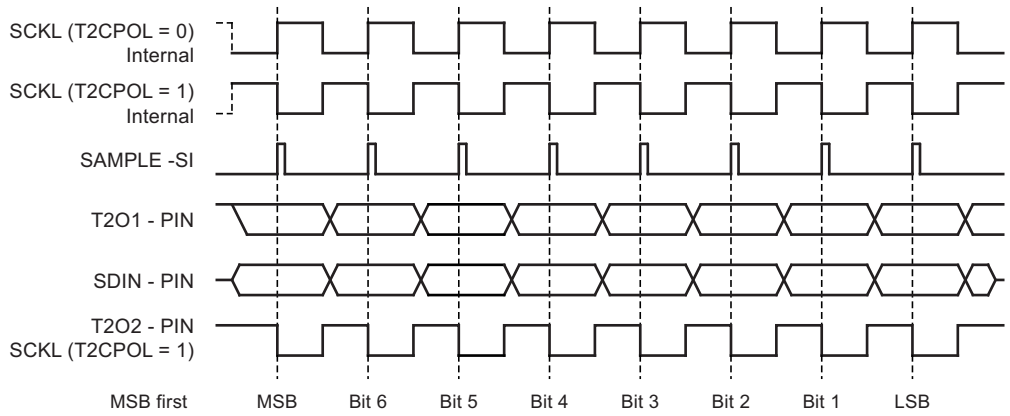
Figure 3-44. Transmit Mode with SSI, Timing Diagram



3.13.6.9 Mode 10: SPI Mode with the Outputs PD5 (T2O1), PD6 (T2O2) and PD7 (SDIN) Pins

Timer2/counter2 can be supplied with internal/external clocks. Mode 10 uses three modulator I/O pins (SO --> T2O1, SCLK --> T2O2 and SI --> SDIN). The timer output clock (CLK_{T2}) can be used to supply the SSI with a shift clock. The SPI mode of the timer allows a SSI synchronous data transfer between the ATA6289 and peripheral devices. The data is always shifted from master (SSI) to slave on the Serial Data Output (SO) line and from slave to master (SSI) on the Serial Data Input (SI) line, synchronized to either the rising or falling edge of the Shift Clock Output (SCLK) line. The serial data is organized in 8-bit telegrams which are shifted with the Most Significant Bit (MSB) first on the Serial Data Output (SO) line. Figure 3-45 shows an example of a three-wire synchronous serial data transfer.

Figure 3-45. SPI Mode with SSI, Timing Diagram



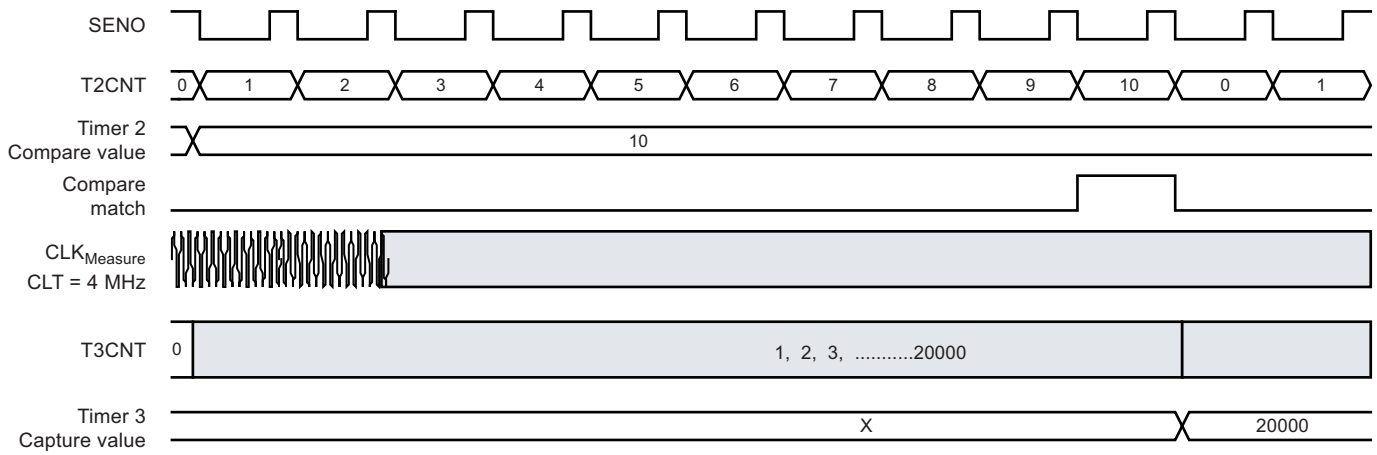
3.13.6.10 Mode 11/12: Capture Mode and Modulator Output T2O1 at the PD5 Pin

Timer2/counter2 can be supplied with internal/external clocks. The trigger source for the input capture signal can be selected by the T2ICS bit in theT2CRA register as well as the active edge for this input signal can be selected with the T2CE(1..0) in the T2MRA register. Mode 11/12 uses one modulator I/O pin (M2 --> T2O1 in mode 12, M2 --> T2O2 in mode 11). The port pins (PD6 and PD7) can be used as a general digital I/O.

3.13.6.11 Mode 13: Sensor Measurement Mode together with Timer3

Timer2/counter2 can be configured via the multiplexer stage to use the SENO signal as input. After a timer2/counter2 compare match is detected the counter3 value can be captured. The synchronization between start of sensor measurement via bit SMS in the sensor control register (SCR) and timer3/ctimer3ounter3 is carried out by hardware. Figure 3-46 shows an example of the sensor measurement mode.

Figure 3-46. Sensor Measurement Mode together with Timer3, Timing Diagram



In sensor measurement mode it is recommended to enable capture noise canceler of timer3 to get correct measure results.

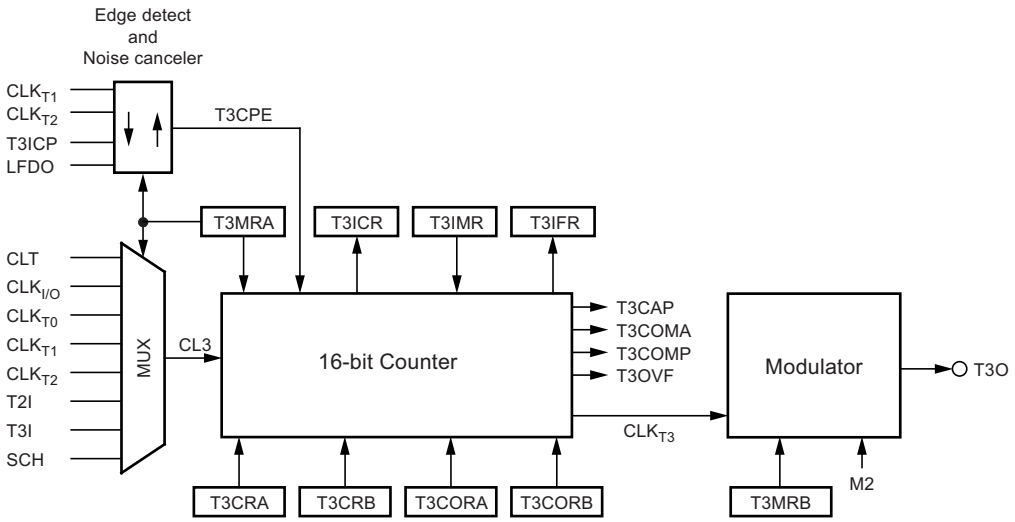
3.13.6.12 Mode 14/15/16: Reserved Modes

3.13.7 Timer3/Counter3

The 16-bit timer/counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement. The main features are:

- True 16-bit design (i.e., allow a 16-bit PWM)
- Eight different selectable input clocks for the timer/counter3
- Two independent compare units
- Input capture noise canceler
- One input capture unit with noise canceler
- Clear timer on compare match or capture event
- Variable PWM period
- Frequency generator
- External event counter
- Four independent interrupt sources (T3CAP, T3COMA, T3COMB, T3OVF)

Figure 3-47. Timer3 Block Diagram



Timer3 consists of an 16-bit up counter with two compare registers (T3CORA, T3CORB) and one capture register (T3ICR). The timer can be used as event counter, timer and signal generator. Its output can be programmed as a modulator. The two compare registers allow various modes of signal generation and modulation. The counter can be driven by internal and external clock sources. For the capture signals (T3ICP, CLK_{T1}, CLK_{T2}, LFDO) it has a programmable edge sensitive input with a digital filtering unit (noise canceler) for reducing the chance of capturing noise spikes. In the capture mode the counter value can be captured by a programmable capture event from the internal timer1 output (CLK_{T1}), timer2 output (CLK_{T2}), by an external event (T3ICP), by the LF receiver output signal (LFDO) at the capture input pin, or with a software capture event by setting the T3SCE bit.

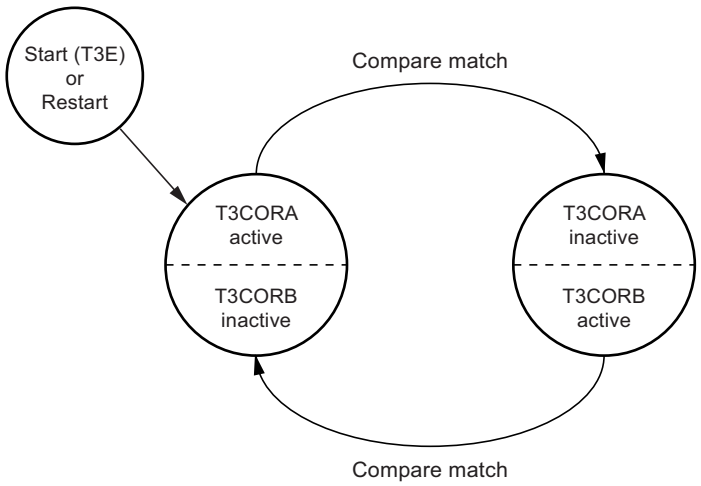
The special functionalities of this timer are the capture event trigger, re-start and single action modes. In the single action mode the counter counts up once to the programmed compare match value. These modes are very useful for modulation, signal generation, signal measurement, and phase controlling. Timer3 has a modulator output stage. As modulator or sensor measurement stage it works together with timer2.

The two compare registers (T3CORA, T3CORB) and the capture register (T3ICR) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in Section 3.13.2 “Accessing 16-Bit Registers” on page 59. The timer3 control, mode, mask, and flag registers (T3CRA, T3CRB, T3MRA, T3MRB, T3IMR, T3IFR) are all 8-bit registers and have no CPU access restrictions.

The comparator outputs are controlled by a control register (T3CRB) and contain mask bits for actions (counter reset, output toggle, single action) which can be triggered by a compare match event or capture event. Every time the output compare registers (T3CORA, T3CORB) are compared with the timer3/counter3 value. The counter can also be enabled to execute single actions with one or both compare registers. If this mode is set, a short period after the counter starts the corresponding match event is generated once.

The timer uses its compare registers alternately if the T3AC bit is set at the T3CRA register. After the timer has been activated, the first comparison is executed by the compare register A, the second is executed by the compare register B, the third is executed by the compare register A and so on as shown in Figure 3-48. This makes it easy to generate signals with constant periods and a variable duty cycle or to generate signals with variable pulse and space widths. If the T3AC bit is cleared at the T3CRA register, the timer does not use its compare registers alternately for compare matches.

Figure 3-48. Timer3 Alternate Compare Register Matches



This architecture enables the timer for various modes. The timer3 operation modes and also the modulator output pin (T3O) are controlled by the T3MRB register.

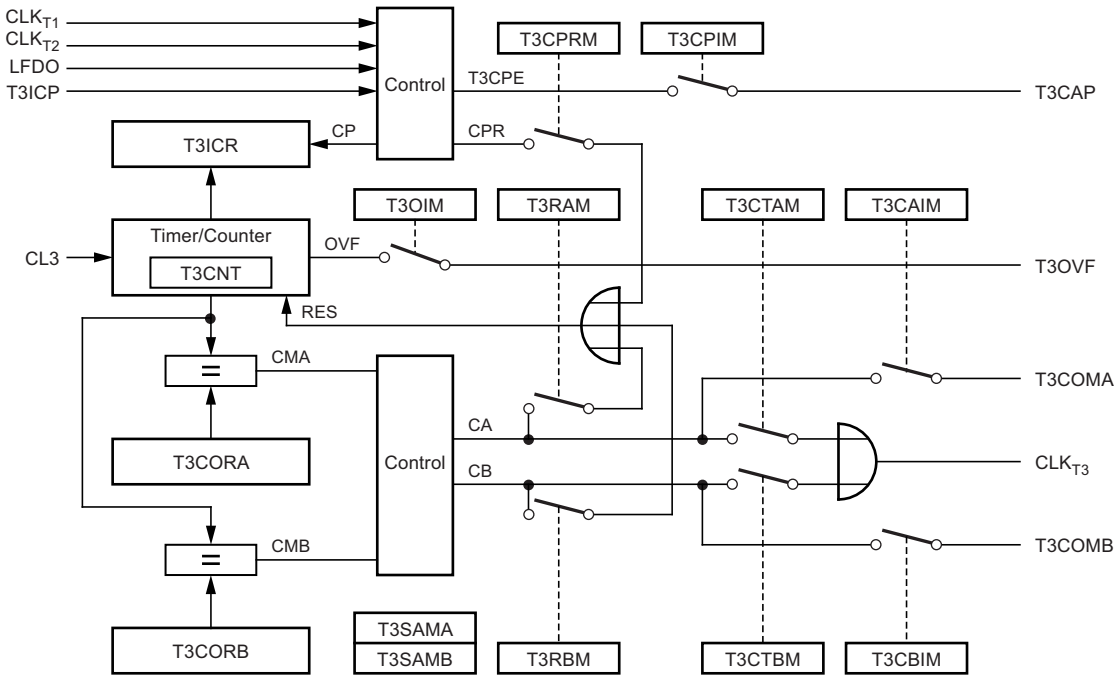
Interrupt request (referred to as “Int. Req.”) signals are all visible in the T3IFR. All interrupts are individually masked with the timer interrupt mask register (T3IMR).

The counter3 input clock (CL3) can be supplied via the I/O Clock ($CLK_{I/O}$), the external input clock (T2I), the external input clock (T3I), the timer0 output clock (CLK_{T0}), the timer1 output clock (CLK_{T1}), the timer2 output clock (CLK_{T2}), the integrated SRC output signal (SCH), or the timer clock (CLT).

3.13.7.1 The Counter3 Stage

The counter stage has five input signals (CL3, T3ICP, LFDO, CLK_{T1}, and CLK_{T2}) and five output signals (T3CAP, T3OVF, T3COMA, T3COMB, and CLK_{T3}). The CL3 supplies timer3/counter3 (T3CNT) with clocks. The external input capture signal (T3ICP), LF Receiver Data Output (LFDO), CLK_{T2} signal, or CLK_{T1} signal capture the counter value in the capture register. CLK_{T3} is the output clock and T3CAP, T3OVF, T3COMA, and T3COMB are the interrupt request signals of the counter stage.

Figure 3-49. 16-Bit Counter3 Stage



3.13.7.2 Signal Description of Timer/Counter3 (Internal Signals)

- CLK_{T1} Timer/counter1 stage clock output
- CLK_{T2} Timer/counter2 stage clock output
- LFDO LF receiver data output
- T3ICP Timer/counter external input capture
- CL3 Timer/counter input clock
- T3CAP Timer/counter capture event interrupt
- T3OVF Timer/counter overflow interrupt
- T3COMA Timer/counter compare match A interrupt
- T3COMB Timer/counter compare match B interrupt
- CLK_{T3} Timer/counter stage clock output
- CMA Timer/counter compare register A match
- CMB Timer/counter compare register B match
- CA Compare match A signal
- CB Compare match B signal
- RES Timer/counter reset (clear all bits)
- OVF Timer/counter overflow
- CP Capture event signal
- CPR Capture event reset signal
- T3CPE Timer/counter capture signal

3.13.7.3 Timer3 Control Register A – T3CRA

Bit	7	6	5	4	3	2	1	0	
	T3E	T3TS	-	-	-	T3CR	T3SCE	T3AC	T3CRA
Read/Write	R/W	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – T3E: Timer3 Enable Bit
This bit controls the timer3 block. The T3E bit must be written to logic one to enable timer3, and if the T3E bit is written to logic zero, the timer3 is disabled.

Bit 6 – T3TS: Timer3 Toggle with Start Bit
The T3TS bit must be written to logic one to toggle the modulator output of timer3 when the timer is enabled with T3E. If the T3TS bit is written to logic zero, the modulator output of timer3 is not toggled with the timer enabled.

Bits 5..3 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and always read as zero.

Bit 2 – T3CR: Timer3 Counter Reset
The T3CR bit resets counter3 asynchronously if this bit is set to logic one.

Bit 1 – T3SCE: Timer3 Software Capture Enable Bit
The T32SCE bit must be written to logic one to generate a software capture event. The T3SCE bit is cleared after the counter value is saved in the capture register. The timer3 counter value is readable via its capture register during run time.

Bit 0 – T3AC: Timer3 Alternate Compare Register Sequence Bit
The T3AC bit must be written to logic one to enable the compare registers alternate mode, and if the T3AC bit is written to logic zero, the alternate mode is disabled.

3.13.7.4 Timer3 Control Register B – T3CRB

Bit	7	6	5	4	3	2	1	0	
	-	T3CPRM	T3CRMB	T3SAMB	T3CTMB	T3CRMA	T3SAMA	T3CTMA	T3CRB
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – Res: Reserved Bit
This bit is a reserved bit on the ATA6289 and is always read as zero.

Bit 6 – T3CPRM: Timer3 Capture Reset Mask Bit
The T3CPRM bit must be written to logic one to enable the counter reset if an internal/external capture event occurs, and if the T3CPRM bit is written to logic zero, the counter reset is disabled.

Bit 5 – T3CRMB: Timer3 Compare Reset Mask Bit B
The T3CRMB bit must be written to logic one to enable the counter reset if a match of the counter with the compare register B (T3CORB) occurs. If the T3CRMB bit is written to logic zero, the counter compare reset is disabled.

Bit 4 – T3SAMB: Timer3 Single Action Mask Bit B
The T3SAMB bit must be written to logic one to enable the single-compare mode, and if the T3SAMB bit is written to logic zero, the single-compare mode is disabled. After this bit is set, a compare match of the counter with register B (T3CORB) is generated only once.

Bit 3 – T3CTMB: Timer3 Compare Toggle Mask Bit B
The T3CTMB bit must be written to logic one to enable the compare toggle, and if the T3CTMB bit is written to logic zero, the compare toggle is disabled. A match of the counter with the compare register B (T3CORB) toggles the output flip-flop in the modulator of timer3.

Bit 2 – T3CRMA: Timer3 Compare Reset Mask Bit A
The T3CRMA bit must be written to logic one to enable the counter reset if a match of the counter with the compare register A (T3CORA) occurs. If the T3CRMA bit is written to logic zero, the counter compare reset is disabled.

Bit 1 – T3SAMA: Timer3 Single Action Mask Bit A
The T3SAMA bit must be written to logic one to enable the single-compare mode, and if the T3SAMA bit is written to logic zero, the single-compare mode is disabled. After this bit is set, a compare match of the counter with register A (T3CORA) is generated only once.

Bit 0 – T3CTMA: Timer3 Compare Toggle Mask Bit A
The T3CTMA bit must be written to logic one to enable the compare toggle, and if the T3CTMA bit is written to logic zero, the compare toggle is disabled. A match of the counter with the compare register A (T3CORA) toggles the output flip-flop in the modulator of timer3.

3.13.7.5 Timer3 Mode Register A – T3MRA

Bit	7	6	5	4	3	2	1	0	
	T3ICS1	**T3ICS0**	**T3CNC**	**T3CE1**	**T3CE0**	**T3CS2**	**T3CS1**	**T3CS0**	**T3MRA**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7 to 6 – T3ICS1..0: Timer 3 Input Capture Select Bits 1 to 0
The T3ICS1 and T3ICS0 bits select the input capture signal of timer3 as shown in Table 3-42.

Table 3-42. Input Capture Signal Select Bit Description

T3ICS1	T3ICS0	Description
0	0	T3ICP
0	1	LFDO
1	0	CLK_{T1}
1	1	CLK_{T2}

Bit5 – T3CNC: Timer3 Input Capture Noise Canceler Bit
Setting this bit (to one) activates the input capture noise canceler. When the noise canceler is activated, the input from the input capture pin is filtered. The filter function requires four successive equal valued samples of the input capture pin for changing its output. The input capture is therefore delayed by four counter clock (CL3) cycles when the noise canceler is enabled.

Bits 4..3 – T3CE1..0: Timer3 Capture Edge Select Bits 1 - 0
The T3CE1 and T3CE0 bits select the edge from all input capture signals of timer3 as shown in Table 3-43.

Table 3-43. Timer3 Capture Edge Select Bit Description

T3CE1	T3CE0	Input Capture Edge of Timer3 Capture Input
0	0	Disable edge detect
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Bits 2 to 0 – T3CS2..0: Timer3 Clock Select Bits 2 to 0
The T3CS2, T3CS1 and T3CS0 bits select the input clock (CL3) of timer3 as shown in Table 3-44.

Table 3-44. Timer3 Input Clock Select Bit Description

T3CS2	T3CS1	T3CS0	Input Clock (CL3) of TCNT3
0	0	0	CLT
0	0	1	$CLK_{I/O}$
0	1	0	CLK_{T0}
0	1	1	CLK_{T1}
1	0	0	CLK_{T2}
1	0	1	T2I
1	1	0	T3I
1	1	1	SCH

3.13.7.6 Timer3 Interrupt Flag Register – T3IFR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	T3ICF	T3COBF	T3COAF	T3OFF	T3IFR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	1	0	0	0	0	

Bits 7..4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and always read as zero.

Bit 3 – T3ICF: Timer3 Input Capture Flag Bit
This flag is set (one) when a capture event occurs at the T3ICP pin indicating that the timer3/counter3 value has been transferred to the capture register (T3ICR). If the I bit in SREG and the T3CPIM bit is set (one) at the T3IMR register, the MCU jumps to the corresponding interrupt vector. T3ICF is automatically cleared when the interrupt routine is executed. Alternatively, T3ICP can be cleared by writing a logic one.

Bit 2 – T3COBF: Timer3 Compare B Flag Bit
This flag is set (one) if the timer3/counter3 value (T3CNT) matches the compare register B value. If the I bit in SREG and the T3CBIM bit is set (one) at theT3IMR register, the MCU jumps to the corresponding interrupt vector. The flag (T3COBF) is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one.

Bit 1 – T3COAF: Timer3 Compare A Flag Bit
This flag is set (one) if the timer3/counter3 value (T3CNT) matches the compare register A value. If the I bit in SREG and the T3CAIM bit is set (one) at the T3IMR register, the MCU jumps to the corresponding interrupt vector. The flag (T3COAF) is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one.

Bit 0 – T3OFF: Timer3 Overflow Flag Bit
This flag is set (one) if the timer3/counter3 overflow (OVF) occurs. If the I bit in SREG and the T3OIM bit is set (one) at the T3IMR register, the MCU jumps to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logic one.

3.13.7.7 Timer3 Interrupt Mask Register – T3IMR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	T3CPIM	T3CBIM	T3CAIM	T3OIM	T3IMR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	1	0	0	0	0	

Bits 7..4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and always read as zero.

Bit 3 – T3CPIM: Timer3 Capture Interrupt Mask Bit
When this bit is written to one, and the I flag in the status register is set (one), the timer3/counter3 input capture interrupt is enabled. The corresponding interrupt vector (see Section 3.10 “Interrupts” on page 39) is executed when the T3ICF flag, located in T3IFR, is set.

Bit 2 – T3CBIM: Timer3 Compare B Interrupt Mask Bit
When the T3CBIM bit is written to one and the I bit in the status register is set (one), the timer3/counter3 compare match B interrupt is enabled. The corresponding interrupt vector is executed if a compare match in timer3/counter3 occurs and the T3COBF bit in the Timer3/Counter3 Interrupt Flag Register (T3IFR) is set.

Bit 1 – T3CAIM: Timer3 Compare A Interrupt Mask Bit
When the T3CAIM bit is written to one and the I bit in the status register is set (one), the timer3/counter3 compare match A interrupt is enabled. The corresponding interrupt vector is executed if a compare match in timer3/counter3 occurs and the T3COAF bit in the Timer3/Counter3 Interrupt Flag Register (T3IFR) is set.

Bit 0 – T3OIM: Timer3 Overflow Interrupt Mask Bit
When the T3OIM bit is written to one and the I bit in the status register is set (one), the timer3/counter3 overflow interrupt is enabled. The corresponding interrupt vector is executed if an overflow in timer3/counter3 occurs and the T3OFF bit in the Timer3/Counter3 Interrupt Flag Register (T3IFR) is set.

3.13.7.8 Timer3 Input Capture Register – T3ICR

Bit	7	6	5	4	3	2	1	0	
	T3ICR [15..8]								T3ICRH
	T3ICR [7..0]								T3ICRL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The input capture register is updated with the counter value (T3CNT) each time an event occurs at the T3ICP pin, the timer1 output clock CLK_{T1}, the timer2 output clock CLK_{T2}, or at the LF receiver output LFDO, or after a software capture event is generated with the T3SCE bit.

The input capture register is 16 bits in size. To ensure that both the high and low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit TEMP. This temporary register is shared by all the other 16-bit registers (see Section 3.13.2 “Accessing 16-Bit Registers” on page 59).

3.13.7.9 Timer3 Compare Register A – T3CORA

Bit	7	6	5	4	3	2	1	0	
	T3CORAH [15..8]								T3CORAH
	T3CORAL [7..0]								T3CORAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The compare registers contain a 16-bit value that is continuously compared with the counter value (T3CNT). A match can be used to generate a compare interrupt, a counter reset, an output clock CLK_{T3} or to generate a waveform with the modulator at the external output pin (T3O).

3.13.7.10 Timer3 Compare Register B – T3CORB

Bit	7	6	5	4	3	2	1	0	
	T3CORBH [15..8]								T3CORBH
	T3CORBL [7..0]								T3CORBL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The compare registers contain a 16-bit value that is compared continuously with the counter value (T3CNT). A match can be used to generate a compare interrupt, a counter reset, an output clock CLK_{T3} or to generate a waveform with the modulator at the external output pin (T3O).

The compare registers are 16 bits in size. To ensure that both the high and low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary TEMP. This temporary register is shared by all the other 16-bit registers (see Section 3.13.2 “Accessing 16-Bit Registers” on page 59).

3.13.7.11 Timer3 Mode Register B – T3MRB

Bit	7	6	5	4	3	2	1	0	
	-	-	-	T3TOP	-	T3M2	T3M1	T3M0	T3MRB
Read/Write	R	R	R	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..5 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 4 – T3TOP: Timer3 Toggle Output Preset Bit
The T3TOP bit must be written to logic one to set the toggle flip-flop, and if the T3TOP bit is written to logic zero, the toggle flip-flop is reset. This bit allows the programmer to preset the toggle output flip-flop in the modulator of the timer3.

Note: No output preset is possible if T3E = “1.”

Bit 3 – Res: Reserved Bit
This bit is a reserved bit on the ATA6289 and is always read as zero.

Bits 2..0 – T3M2..0: Timer3 Mode Bits 2 - 0
The T3M2,T3M1 and T3M0 bits select the output mode of timer3 as shown in Table 3-45.

Table 3-45. Timer3 Mode Bit Description

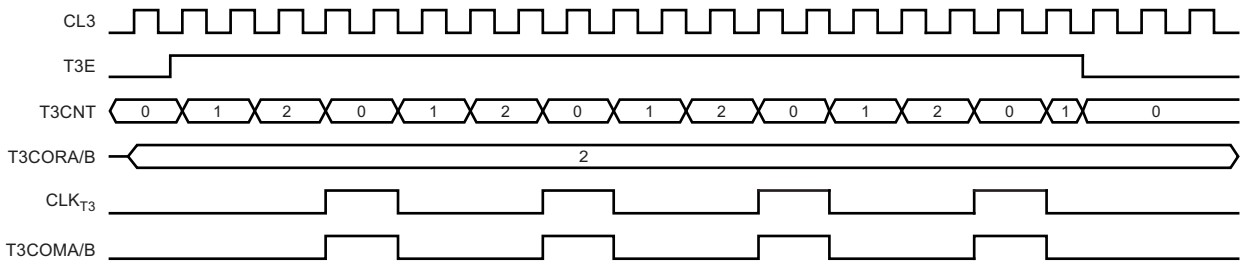
Mode	T3M2	T3M1	T3M0	Timer3 Mode	T3O(1)
1	0	0	0	Timer/Counter mode	PB1
2	0	0	1	Toggle mode	M30
3	0	1	0	Burst modulation with timer2 (M2)	M31
4	0	1	1	Capture mode	PB1
5	1	0	0	Capture mode	M30
6	1	0	1	Sensor measurement mode together with timer2	PB1
7	1	1	0	Reserved	PB1
8	1	1	1	Reserved	PB1

Note: General port I/O: PB1
Alternate port function:
Output signal of the T3 (toggle flip-flop): M30 or M31 = M30 and M2

3.13.7.12 Timer3 Modes

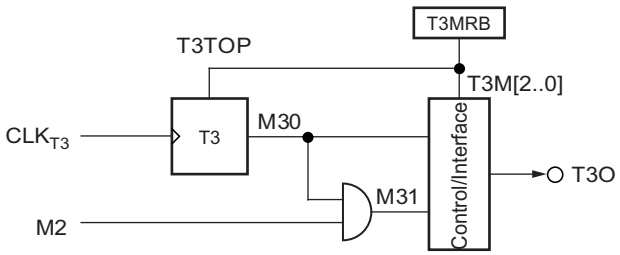
Mode 1: Timer/Counter Mode
In timer/counter mode timer3/counter3 can be supplied with internal/external clocks. The port pin PB1 can be used as a general digital I/O. An example of this mode is shown in Figure 3-50.

Figure 3-50. Timer/Counter Mode, Timing Diagram



Mode 2: Toggle Mode
The modulator consists of a toggle flip-flop (T3), a control logic with the Timer3 Mode Register B (T3MRB) and an interface to the output pin (T3O). The modulators have different modes, which can be selected with the T3M[2..0] bits in the T3MRB register.

Figure 3-51. Timer3 Modulator Stage



In toggle mode timer3/counter3 can be supplied with internal/external clocks. With every clock CLK_{T3} the output of the flip-flop T3 (M30) toggles. The signal M30 is directed to the output T3O.

Mode 3: Burst Modulation Mode with Timer2 (M2)
In burst modulation mode timer3/counter3 can be supplied with internal/external clocks. In this mode the toggled clock of CLK_{T3} (M30) can be bursted (gated) with the modulator output of timer2 (M2) (see Figure 3-51 (M31 = M30 and M2)). The signal M31 is directed to the output T3O.

Mode 4/5: Capture Mode
Timer3/counter3 can be supplied with internal/external clocks. The trigger source for the input capture signal can be selected by the T3ICS[1..0] bits in theT3MRA register as well as the active edge for this input signal can be selected with the T3CE[1..0] in the T3MRA register. In mode 4 the port pin PB1 can be used as a general digital I/O. In mode 5 the output signal of toggle flip-flop (M30) is directed to T3O.

Mode 6: Sensor Measurement Mode together with Timer2
See Figure 3-46 on page 85 (mode 13 description of timer2).

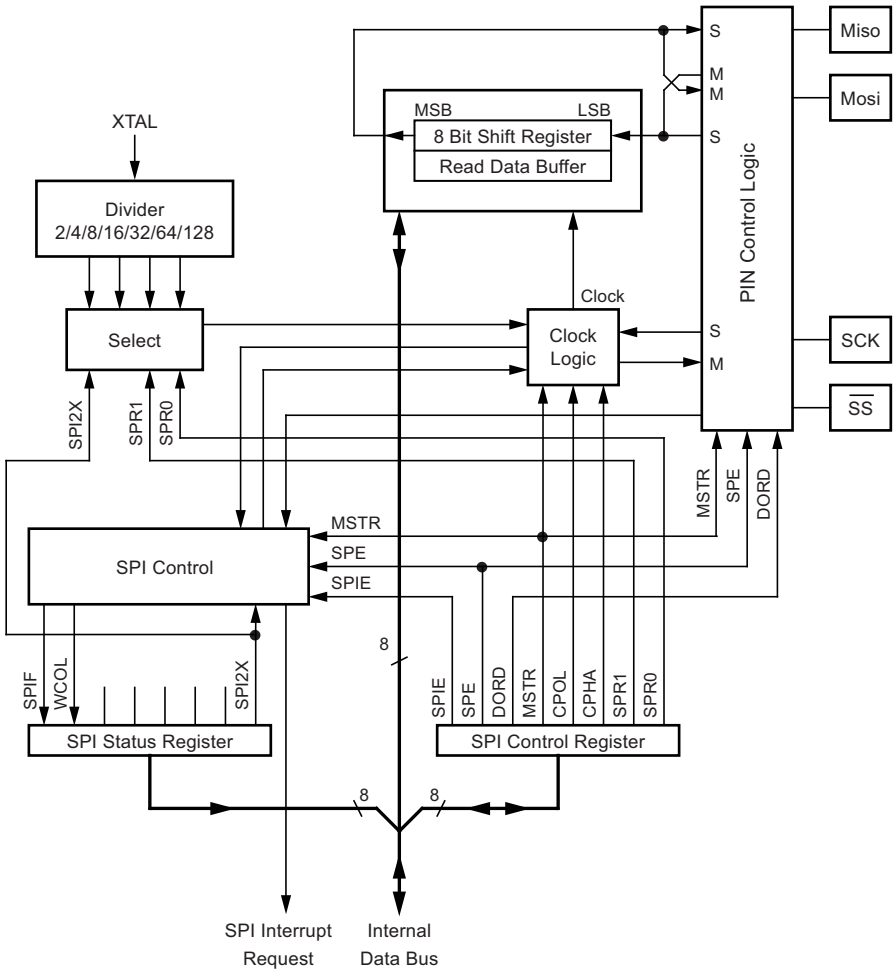
In sensor measurement mode it is recommended to enable the capture noise canceler of timer3 to get correct measure results.

3.14 Serial Peripheral Interface – SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATA6289 and peripheral devices or between several Atmel® AVR® devices. The ATA6289 SPI includes the following features:

- Full-duplex, three-wire synchronous data transfer
- Master or slave operation
- LSB first or MSB first data transfer
- Seven programmable bit rates
- End of transmission interrupt flag
- Write collision flag protection
- Wake-up from idle mode
- Double speed (CK/2) master SPI mode

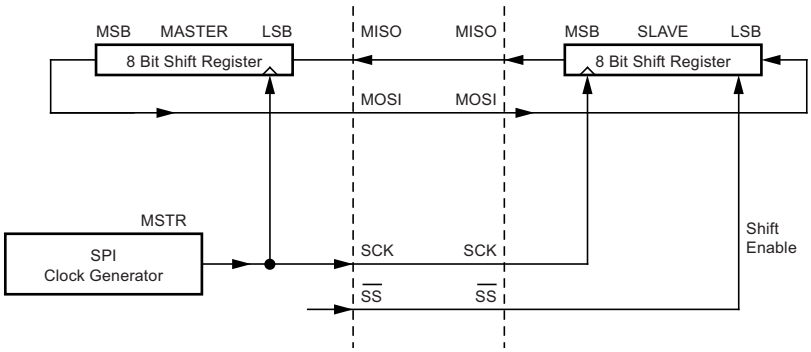
Figure 3-52. SPI Block Diagram



The interconnection between master and slave CPU with the SPI is shown in Figure 3-53. The system consists of two-shift registers and a master clock generator. The SPI master initiates the communication cycle when pulling low the Slave Select ($\overline{SS}$) pin of the desired slave. Master and slave prepare the data to be sent in their respective shift registers and the master generates the required clock pulses on the SCK line to interchange data. Data is always shifted from master to slave on the Master-Out-Slave-In (MOSI) line, and from slave to master on the Master-In-Slave-Out (MISO) line. After each data package, the master synchronizes the slave by pulling the Slave Select ($\overline{SS}$) line high.

When configured as a master, the SPI interface has no automatic control of the $\overline{SS}$ line. This must be handled by user software before communication can start. When this is done, writing a byte to the SPI data register starts the SPI clock generator and the hardware shifts the eight bits into the slave. After shifting one byte, the SPI clock generator stops, setting the end of the Transmission Flag (SPIF) in the SPI Status Register (SPSR). If the SPI Interrupt Enable bit (SPIE) in the SPCR register is set, an interrupt is requested. The master may continue to shift the next byte by writing it into SPDR or signal the end of packet by pulling high the Slave Select ($\overline{SS}$) line. The last incoming byte is kept in the buffer register for later use. When configured as a slave, the SPI interface remains sleeping with MISO tri-stated as long as the $\overline{SS}$ pin is driven high. In that state, software may update the contents of the SPI data register, SPDR register, but the data is not shifted out by incoming clock pulses at the SCK pin until the $\overline{SS}$ pin is driven low. If one byte has been completely shifted, the end of the transmission flag (SPIF) is set. If the SPI interrupt enable bit (SPIE) in the SPCR register is set, an interrupt is requested. The slave may continue to place new data to be sent into SPDR before reading the incoming data. The last incoming byte is kept in the buffer register for later use.

Figure 3-53. Master-Slave Interconnection



The system is single buffered in the transmit direction and double buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register (SPDR) before the entire shift cycle is completed. When receiving data, however, a received character must be read from the SPI data register before the next character has been completely shifted in. Otherwise, the first byte is lost.

In SPI slave mode, the control logic samples the incoming signal of the SCK pin. The frequency of the SPI clock should never exceed $f_{osc}/4$ to ensure correct sampling of the clock signal.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and pin is overridden according to Table 3-46. For more details on automatic port overrides, refer to Section 3.12.3 “Alternate Port Functions” on page 49.

Table 3-46. SPI Pin Overrides

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User defined
SCK	User Defined	Input
SS	User Defined	Input

3.14.1 $\overline{SS}$ Pin Functionality

3.14.1.1 Slave Mode

When the SPI is configured as a slave, the Slave Select ($\overline{SS}$) pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MISO becomes an output if configured by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs and the SPI is passive, which means that it does not receive incoming data. Note that the SPI logic is reset once the $\overline{SS}$ pin is driven high.

The $\overline{SS}$ pin is useful for packet/byte synchronization to keep the slave bit counter synchronous with the master clock generator. When the $\overline{SS}$ pin is driven high, the SPI slave immediately resets the send and receive logic and drops any partially received data in the shift register.

3.14.1.2 Master Mode

When the SPI is configured as a master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin.

If $\overline{SS}$ is configured as an output, the pin is a general output pin which does not affect the SPI system. Typically, the pin drives the $\overline{SS}$ pin of the SPI slave.

If $\overline{SS}$ is configured as an input, it must be held high to ensure master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as a master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another master selecting the SPI as a slave and starting to send data to it. To avoid bus contention, the SPI system performs the following actions:

The MSTR bit in SPCR is cleared and the SPI system becomes a slave. As a result of the SPI becoming a slave, the MOSI and SCK pins become inputs.

The SPIF flag in SPSR is set and the interrupt routine is executed if the SPI interrupt is enabled and the I bit in SREG is set.

Thus, when interrupt-driven SPI transmission is used in master mode and there is the possibility that $\overline{SS}$ is driven low, the interrupt should always make sure that the MSTR bit is still set. If the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable the SPI master mode.

3.14.1.3 SPI Control Register – SPCR

Bit	7	6	5	4	3	2	1	0	
	SPIE	**SPE**	**DORD**	**MSTR**	**CPOL**	**CPHA**	**SPR1**	**SPR0**	**SPCR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – SPIE: SPI Interrupt Enable
This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR register is set and if the global interrupt enable bit in SREG is set.

Bit 6 – SPE: SPI Enable
When the SPE bit is written to one, the SPI is enabled. This bit must be set to enable any SPI operations.

Bit 5 – DORD: Data Order
When the DORD bit is written to one, the LSB of the data word is transmitted first. When the DORD bit is written to zero, the MSB of the data word is transmitted first.

Bit 4 – MSTR: Master/Slave Select Register
This bit selects the master SPI mode when written to logic one, and the slave SPI mode when written logic zero. If $\overline{SS}$ is configured as an input and is driven low while MSTR is set, MSTR is cleared and SPIF in SPSR is set. The user then has to set MSTR to re-enable the SPI master mode.

Bit 3 – CPOL: Clock Polarity
When this bit is written to one, SCK is high when idle. When CPOL is written to zero, SCK is low when idle. Refer to Figure 3-54 and Figure 3-55 on page 100 for an example. The CPOL functionality is summarized in Table 3-47.

Table 3-47. CPOL Functionality

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

Bit 2 – CPHA: Clock Phase

The setting of the Clock Phase bit (CPHA) determines if data is sampled at the leading (first) or trailing (last) edge of SCK. Refer to Figure 3-54 and Figure 3-55 on page 100 for an example. The CPHA functionality is summarized below.

Table 3-48. CPHA Functionality

CPHA	Leading Edge	Trailing Edge
0	Sample	Setup
1	Setup	Sample

Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0

These two bits control the SCK rate of the device configured as a master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator clock frequency f_{osc} is shown in Table 3-49 on page 98.

Table 3-49. Relationship Between SCK and the Oscillator Frequency f_{osc}

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{OSC/4}$
0	0	1	$f_{OSC/16}$
0	1	0	$f_{OSC/64}$
0	1	1	$f_{OSC/128}$
1	0	0	$f_{OSC/2}$
1	0	1	$f_{OSC/8}$
1	1	0	$f_{OSC/32}$
1	1	1	$f_{OSC/64}$

3.14.1.4 SPI Status Register – SPSR

Bit	7	6	5	4	3	2	1	0	
	SPIF	WCOL	-	-	-	-	-	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – SPIF: SPI Interrupt Flag

When a serial transfer is complete, the SPIF flag is set. An interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. The SPIF flag is also set if $\overline{SS}$ is an input and is driven low when the SPI is in master mode. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI status register with SPIF set, then accessing the SPI Data Register (SPDR).

Bit 6 – WCOL: Write Collision Flag

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) is cleared by first reading the SPI status register with WCOL set and then accessing the SPI Data Register (SPDR).

Bit 5..1 – Res: Reserved Bits

These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 0 – SPI2X: Double SPI Speed Bit

If this bit is written logic one, the SPI speed (SCK frequency) is doubled when the SPI is in master mode (see Table 3-49). This means that the minimum SCK period is two CPU clock periods. When the SPI is configured as a slave, the SPI is only guaranteed to work at $f_{osc/4}$ or lower.

The SPI interface on the ATA6289 is also used for program memory and EEPROM downloading or uploading. See Section 3.21.5 “Serial Downloading” on page 149 for serial programming and verification.

3.14.1.5 SPI Data Register – SPDR

Bit	7	6	5	4	3	2	1	0	
	SPDR[7..0]								SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

The SPI data register is a read/write register used for data transfer between the register file and the SPI shift register. Writing to the register initiates data transmission. Reading the register causes the shift register receive buffer to be read.

3.14.2 Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 3-54 and Figure 3-55 on page 100. Data bits are shifted out and latched in on opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize. This is clearly seen by summarizing Table 3-47 and Table 3-48 on page 98, as done in Table 3-50:

Table 3-50. CPOL/CHPA Functionality

	Leading Edge	Trailing Edge	SPI Mode
CPOL = 0, CPHA = 0	Sample (Rising)	Setup (Falling)	0
CPOL = 0, CPHA = 1	Setup (Rising)	Sample (Falling)	1
CPOL = 1, CPHA = 0	Sample (Falling)	Setup (Rising)	2
CPOL = 1, CPHA = 1	Setup (Falling)	Sample (Rising)	3

Figure 3-54. SPI Transfer Format with CPHA = 0

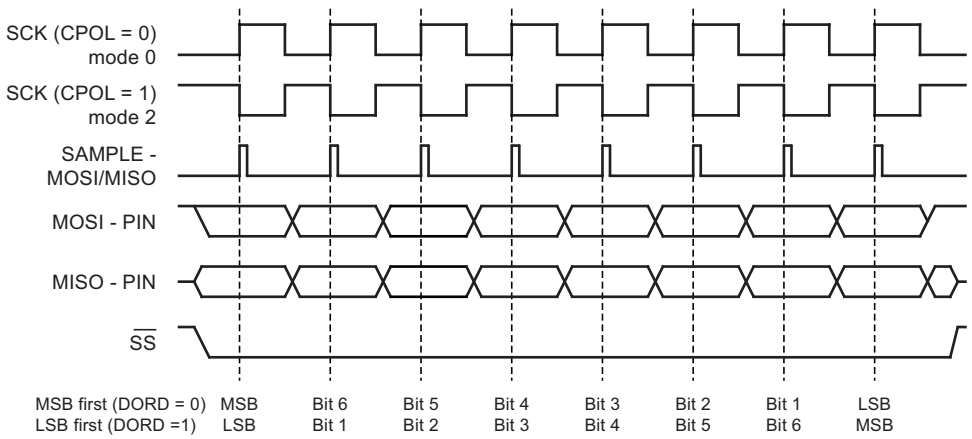
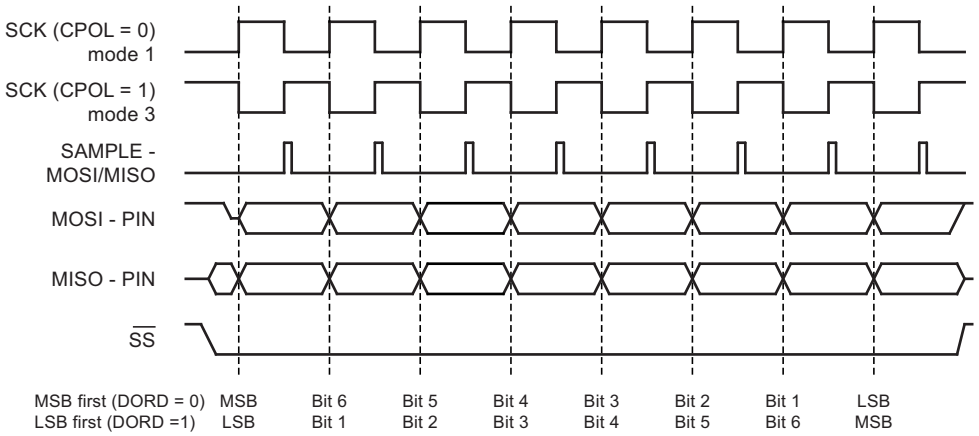


Figure 3-55. SPI Transfer Format with CPHA = 1



3.15 Sensor Interface Block

The sensor interface block contains a Motion Sensor Interface unit (MSIU), a Capacitance Sensor Interface Unit (CSIU), a Voltage Sensor Interface Unit (VSIU), four control registers (SCR, SCCR, SVCR, TSCR), an 8-bit programmable voltage reference calibration register (MSVCAL), a Sensor Interrupt Mask Register (SIMSK), a Sensor Status and Flag Register (SSFR) as well as an output multiplexer to select one of the two different signals f_C (output signal of CSIU) and f_V (output signal of VSIU) for the output signal SENO. The output signal (SENO) is selected by the SEN[1..0] bits in the SCR Register. SENO is directed as input for timer2 (see Figure 3-29 on page 69). The capacitor/voltage sensor interface units in combination with the internal timer modules timer2 and timer3 build an ADC to digitize the different sensor signals, for example, from an external capacitance pressure sensor or an external capacitance acceleration/motion sensor. Additionally, internal voltage levels can be measured and also an internal on-chip temperature sensor can be monitored with the ADC. The general functionality of this ADC is described in the Section 3.15.2 “Capacitance Sensor Interface Unit (CSIU)” on page 103ff.

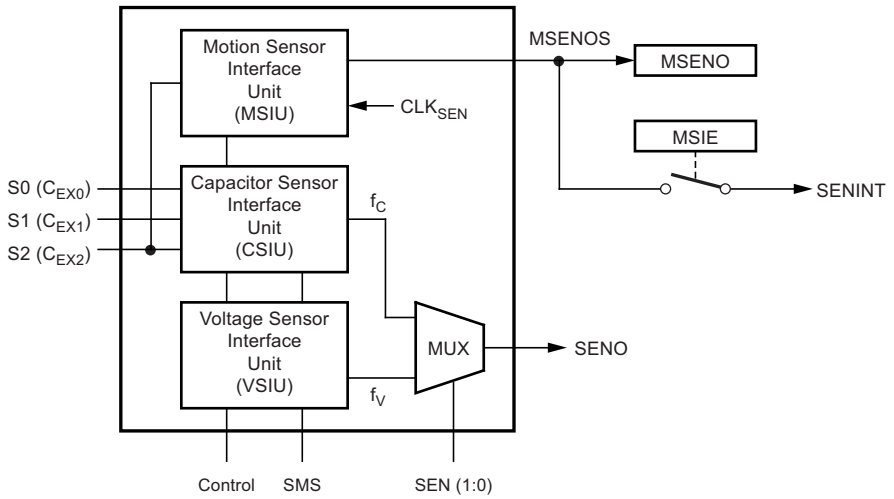
Channel (pin) S2 can be either configured as a motion wake-up with low-power consumption or as a normal capacitive sensor interface. For a motion wake-up application on pin S2 the capacitor sensor interface must be disabled. If the capacitor sensor interface is enabled, the internal circuitry periodically forces all sensor pins to GND. No motion wake-up detection occurs, i.e., motion wake-up fails while all sensor pins are forced to GND.

If motion/acceleration is detected, the output signal of MSU (MSENOS) gets high and the MSENO bit in the SSFR register is set (one)—otherwise MSENOS is low and MSENO bit is cleared (zero).

Additionally, a motion sensor interrupt signal (SENINT) can be enabled if the MSIE bit in the SIMSK register is set (one). The corresponding interrupt vector is executed if MSENF bit in SSFR register is set (one). A more detailed description is given in Section 3.15.1 “Motion Sensor Interface Unit (MSIU)” on page 101.

Configurations of the Atmel ATA6289 need to be carried out to ensure accurate results of sensor interface applications specific sleep mode. Best results can be achieved for a motion wake-up application if ATA6289 is configured into power-down mode. The performance of the capacitive sensor interface is optimized if the ATA6289 is configured into sensor-noise reduction mode.

Figure 3-56. Sensor Interface Block



For further information on the resolution of the three different units MSIU, CSIU and VSIU see Section 5.2 “Operating Characteristics of Atmel ATA6289” on page 169ff.

3.15.1 Motion Sensor Interface Unit (MSIU)

This section describes how a measurement with a motion sensor at pin S2 is executed and how a motion wake-up can be generated.

The SMEN bit in the SCR register must be set (one) to enable the MSIU. The MSIU needs the timer0timer0 clock (CLK_{SEN}) for starting the motion measurements (starting the state machine) periodically, i.e., timer0 has to be activated and TOBS[2..0] bits in the T0CR register must be set so that CLKSEN is directed to the motion sensor interface (see also Section 3.13.3 “Timer0 with Watchdog/Interval Timer” on page 62ff). Before changing the T0CR register the MSIU needs to be disabled. After the T0CR register is set the motion sensor interface can be enabled again. In that way a safe restart of the state machine of MSIU is guaranteed.

The capacitance of the external motion sensor C_M (at pin S2) together with the internal capacitance C_{Ref} forms a capacitive voltage divider (see Figure 3-57 on page 102). If switch Sw1 is closed while controlled by the state machine, a voltage VSEN is generated which is a function of C_M.

$$V_{SEN} = \frac{C_{Ref}}{C_{Ref} + C_M} \times V_{CC}$$

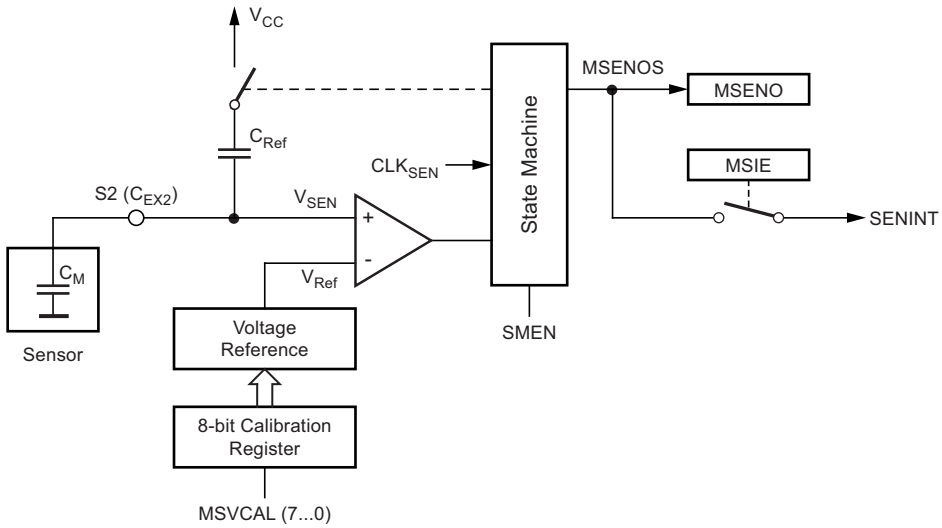
V_{CC} is the supply voltage of the ATA6289.

If the application starts to rotate, the capacitance of the motion/acceleration sensor increases, while at the same time the sensor voltage V_{SEN} decreases. A low-offset-auto-zero comparator stage compares the sensor input voltage on pin S2 with the internally calibrated reference voltage (V_{Ref}). If the sensor voltage on pin S2 is lower than the internal reference voltage V_{Ref},, the MSENOS signal goes to high and the MSENO bit in the SSFR register is set. By polling the MSENO bit the ATA6289 can be woken up if this bit is set.

In addition, a motion Sensor Interrupt signal (SENINT) can be enabled if the MSIE bit in the SIMSK register is set (one). The corresponding interrupt vector is executed if the MSENF bit in the SSFR register is set (one).

The internal reference voltage V_{Ref} is programmable via the 8-bit calibration register (MSVCAL).

Figure 3-57. Motion Sensor Interface Unit



This system of a capacitive voltage divider in combination with the low-power timer0 minimize the power consumption during motion measurement polling (see Section 5.2 “Operating Characteristics of Atmel ATA6289” on page 169ff).

3.15.1.1 Motion Sensor Voltage Calibration Register – MSVCAL

Bit	7	6	5	4	3	2	1	0	
	VRCAL7	VRCAL6	VRCAL5	VRCAL4	VRCAL3	VRCAL2	VRCAL1	VRCAL0	MSVCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..0 – VRCAL7..0: Voltage Reference Calibration Value

The voltage reference calibration register is used to trim the internal voltage reference V_{Ref} for the motion comparator threshold level.

3.15.1.2 Sensor Interrupt Mask Register – SIMSK

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	MSIE	SIMSK
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..1 – Res: Reserved Bits

These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 0 – MSIE: Motion Sensor Interrupt Enable

Writing MSIE to one enables a motion interrupt if the I bit in SREG is set. Writing MSIE to zero disables the interrupt. The corresponding interrupt vector is executed when the MSENF flag, located in SSFR, is set.

3.15.1.3 Sensor Status and Flag Register – SSFR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	MSENO	MSENF	SSFR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..2 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 1 – MSENO: Motion Sensor Output Signal
This bit reads the MSENO signal of the motion sensor. This bit is set when the motion sensor interface unit has detected motion/acceleration.

Bit 0 – MSENF: Motion Sensor Flag
This flag is set when a rising edge occurs at the MSENO output signal. MSENF is automatically cleared when the motion sensor interrupt vector is executed. Alternatively, MSENF can be cleared by writing a logic one to its bit location.

3.15.2 Capacitance Sensor Interface Unit (CSIU)

This section describes how a measurement of a capacitive pressure sensor, connected to one of the sensor pins S0, S1 or S2, is executed.

The CSIU is enabled by setting the SEN[1..0] bits to '01' in the SCR register. It does not consume power when the SEN[1..0] bits are cleared, so it is recommended to switch off the CSIU before entering power-saving sleep modes.

The external capacitance of the pressure sensor connected to S0, S1 or S2 and the internal reference capacitance are measured by means of a single-slope A/D conversion method that converts the capacitance to the frequency f_C. The frequency f_C is generated by periodically charging (and discharging for a constant time $T_{Discharge}$) the capacitance selected by the MUX with a constant current Icharge up to the threshold voltage V_{REF} of a comparator (see Figure 3-58 on page 104).

The frequency signal f_C is directed as input (SENO) to timer2. The measurement stops if the value of timer2 has reached the compare value of the T2COR register (see Figure 3-58). The total time required for this charging (and discharging) procedure is a function of measured capacitance:

$$T_{Total} = \left(\frac{C_X \times V_{REF}}{I_{charge}} + T_{Dischargel}\right) \times T2COR$$

The higher the compare value of timer2 (T2COR) the better is the resolution of this measurement.

The time T_{Total} is determined with timer3 which starts synchronously with timer2 (start of measurement with the SMS bit in the SCR register) and should be clocked with the highest possible frequency $CLK_{Measure}$ (for example, 4MHz) to get the best resolution. The compare match of timer2 generates a capture interrupt event for timer3, i.e., timer3 stops synchronously with timer2. This means that the timer3 value in the capture register T3ICR is a function of measured capacitance (see also Figure 3-59 on page 104).

In combination with the capacitance sensor interface unit, timer2 and timer3 build an ADC that works according to the counting procedure.

Figure 3-58. Capacitance Sensor Interface Unit

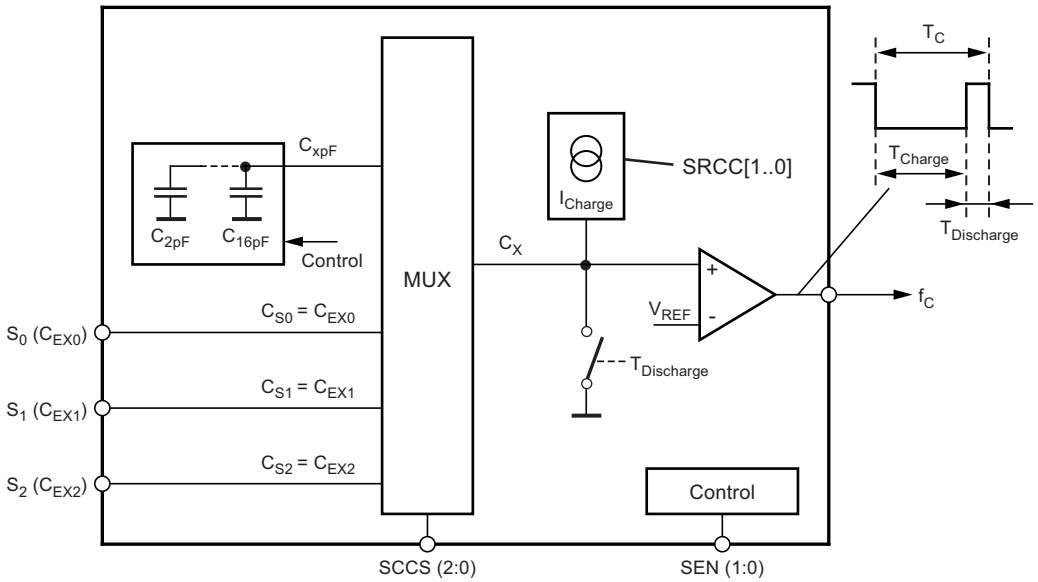
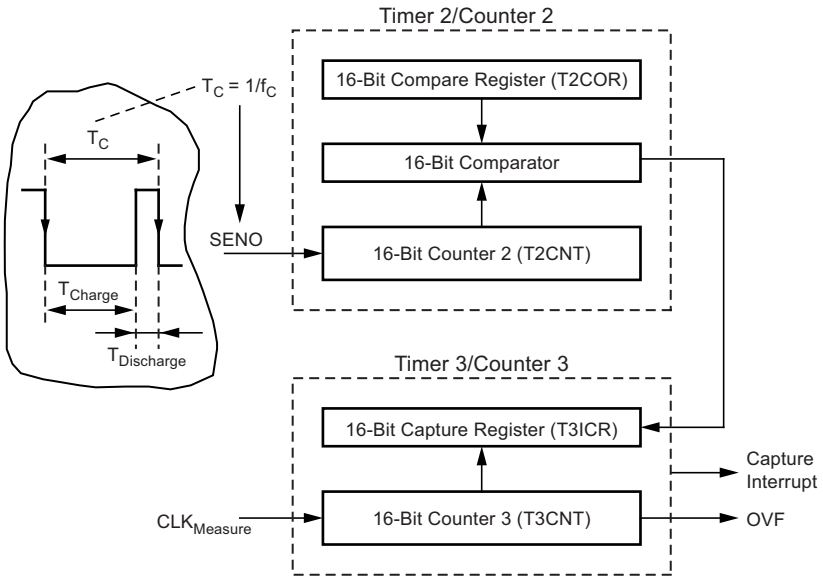


Figure 3-59. Typical Application Example of a Capacitance Measurement



Three consecutive measurement steps are required to determine the value of the external pressure sensor capacitance (see Figure 3-60 on page 105):

1. Measurement of T_{Ref1} for C_{Ref1}
2. Measurement of T_{Ref2} for C_{Ref2}
3. Measurement of T_{Sensor} for C_{Sensor}

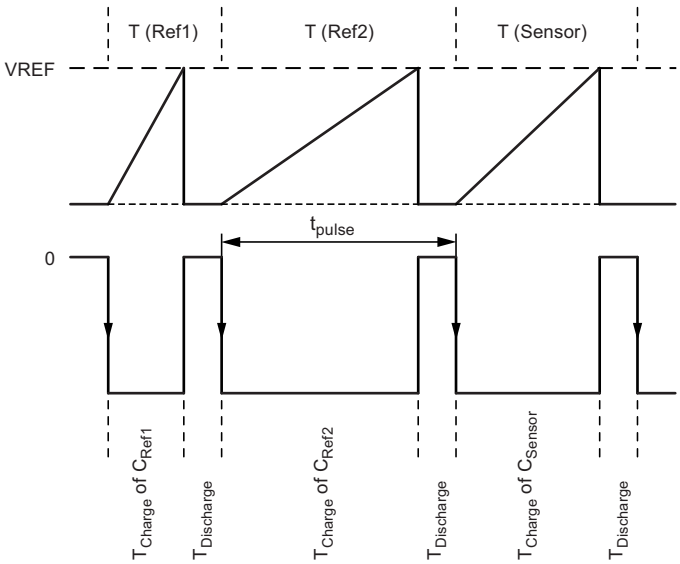
The value of C_{Sensor} can then be calculated as (linear approximation):

$$C_{Sensor} = \frac{T_{Sensor} - T_{Ref1}}{T_{Ref2} - T_{Ref1}} \times (C_{Ref2} - C_{Ref1}) + C_{Ref1}$$

The different capacitance (C_{Ref1}, C_{Ref2}, C_{Sensor}) can be chosen by setting the SCC[2..0] bits in the SCCR register. To get good accuracy over the whole range of the pressure sensor, it is recommended to choose C_{Ref1} close to the minimum capacitance value of the sensor and to choose C_{Ref2} close to the maximum capacitance value of the sensor.

It is also recommended to set T2COR and the current value I_{charge} (with SRCC[1..0] bits in the SCCR register) so that timer3 overflow is avoided and to maximize ADC resolution for a given current consumption.

Figure 3-60. Example of a Capacitance Measurement for Timer2 Compare Register T2COR = 1(1)



Note: 1. In this example T2COR is set to 1, but for a better ADC resolution it is recommended to choose a higher T2COR value (see description above).

3.15.3 CSIU Register Description

3.15.3.1 Sensor Control Register – SCR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	SMEN	SEN1	SEN0	SMS	SCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 3 – SMEN: Sensor Motion Enable Bit
This bit controls the motion sensor interface. The SMEN bit must be written to logic one to enable the interface. If the SMEN bit is written to logic zero, the motion sensor interface is disabled.

Bit 2..1 – SEN1..0: Sensor Enable Bits 1 - 0
These bits control the sensor interface stage as shown in Table 3-51.

Table 3-51. Sensor Stage Enable Bits Description

SEN1	SEN0	Description
0	0	Disabled
0	1	Capacitance sensor enabled
1	0	Reserved
1	1	Voltage sensor enabled

Bit 0- SMS: Sensor Measurement Start Bit
This bit starts the measurement of the selected sensor interface stage. The SMS bit must be written to logic one to start the measurement. If the SMS bit is written to logic zero, the sensor interface stops the measurement.

3.15.3.2 Sensor Capacitance Control Register – SCCR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	SCCS2	SCCS1	SCCS0	SRCC1	SRCC0	SCCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7, 6, 5 – Res: Reserved Bits
These bits are reserved bit on the ATA6289 and are always read as zero.

Bits 5..3 – SCCS2..0: Sensor Capacitance Channel Select Bits 2 - 0
These bits select the input channel of capacitance multiplexer output (CXpF) as shown in Table 3-52.

Table 3-52. Multiplexer Output Channel Select Bit Description

SCCS2	SCCS1	SCCS0	Reference Capacitance Value/pF	C_{Xpf}
0	0	0	-	C_{S0}
0	0	1	-	C_{S1}
0	1	0	-	C_{S2}
0	1	1	2	C_{2pF}
1	0	0	3	C_{3pF}
1	0	1	6	C_{6pF}
1	1	0	10	C_{10pF}
1	1	1	16	C_{16pF}

Bits 1..0 – SRCC1..0: Sensor Reference Charge Current Bits 1 - 0
These bits select the charge current value (Icharge) that is needed for measuring the external sensor capacitance and the internal reference capacitance as shown in Table 3-53.

Table 3-53. Reference Charge Current Bit Description

SRCC1	SRCC0	Charge Current Value (I_{charge}) at V_{CC} = 3V
0	0	150nA
0	1	300nA
1	0	600nA
1	1	900nA

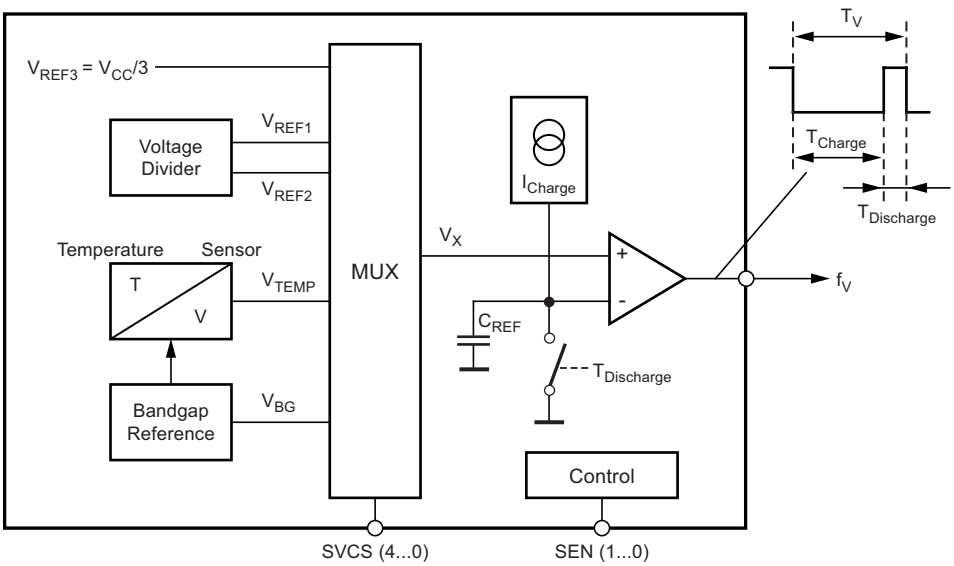
3.15.4 Voltage Sensor Interface Unit (VSIU)

With the VSIU the battery voltage (V_{CC}) and the temperature voltage of an internal PTAT (V_{TEMP}) can be measured.

The VSIU is enabled by setting the SEN[1..0] bits to logic '10' in the SCR register. It does not consume power when the SEN[1..0] bits are cleared, so it is recommended to switch off the VSIU before entering power-saving sleep modes.

The general measurement procedure is equal to that of the measurement of an external capacitance pressure sensor. For a detailed description, see the last section. The difference is that the reference voltage for the comparator is the unknown voltage V_X. A clock signal f_V is generated by periodically charging (and discharging for a constant time $T_{Discharge}$) an internal reference capacitance with a fixed current I_{charge} until the unknown voltage level V_X is reached (see Figure 3-61). The signal f_V is directed as input to timer2 (SENO). The length of the SENO pulse is internally limited to a fixed time value to guarantee that the timer2/counter2 input detects the trigger input signal.

Figure 3-61. Voltage Sensor Interface Unit



Four consecutive measurement steps are required to determine the value of the voltages V_{CC} and V_{TEMP}:

1. Measurement of T_{Ref1} for V_{Ref1}
2. Measurement of T_{Ref2} for V_{Ref2}
3. Measurement of T_{BG} for V_{BG}
4. Measurement of T_{TEMP} or T_{Ref3} for V_{TEMP} or $V_{Ref3} = V_{CC}/3$ respectively

The values of V_{CC} and V_{TEMP} can then be calculated as (linear approximation):

$$V_{CC} = 3 \times \frac{T_{Ref3} + T_{Ref1} - 2 \times T_{Ref2}}{T_{BG} + T_{Ref1} - 2 \times T_{Ref2}} \times V_{BG}$$

$$V_{TEMP} = \frac{T_{TEMP} + T_{Ref1} - 2 \times T_{Ref2}}{T_{BG} + T_{Ref1} - 2 \times T_{Ref2}} \times V_{BG}$$

with V_{BG} = 1.23V (calibrated)

The different voltages (V_{Ref1}, V_{Ref2}, V_{BG}, V_{Ref3}, V_{TEMP}) can be chosen by setting the SVCS[4..0] bits in the SVCR register.

It is also recommended to set T2COR so that timer3 overflow is avoided and to maximize ADC resolution for a given current consumption.

3.15.5 VSIU Register Description

3.15.5.1 Sensor Voltage Control Register – SVCR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	SVCS4	SVCS3	SVCS2	SVCS1	SVCS0	SVCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7 to 4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

These bits control the output of the voltage multiplexer output (V_X) as shown in Table 3-54.

Table 3-54. Multiplexer Output Channel Select Bit Description(1)

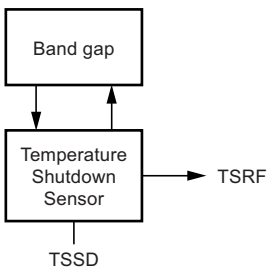
SVCS4	SVCS3	SVCS2	SVCS1	SVCS0	Description	Legend
0	0	0	0	0	open	No channel selected
0	0	0	0	1	V_{BG}	Bandgap voltage
0	0	0	1	0	V_{TEMP}	Temperature voltage
0	0	1	0	0	V_{Ref1}	Internal reference voltage: = typically 900mV at T = 27°C
0	1	0	0	0	V_{Ref2}	Internal reference voltage: = ½ × V_{Ref1}
1	0	0	0	0	V_{Ref3}	Internal reference voltage: = 1/3 × V_{CC}
All other combinations are not allowed					-	Forbidden selection

Note: 1. A direct change between two different input channels is not allowed. It is necessary to apply the following sequence (e.g., V_{BG} --> open --> V_{Temp})

3.15.6 Temperature Shutdown Mode

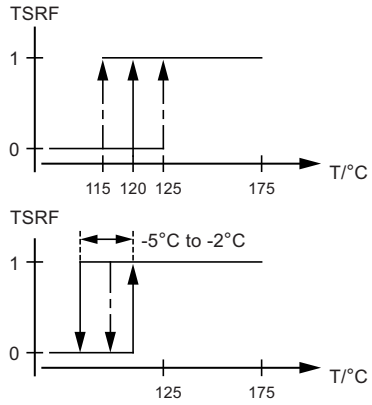
The Temperature Shutdown Sensor unit (TS) contains an on-chip temperature shutdown sensor in combination with a bandgap circuitry. The TS is enabled by clearing the TSSD bit in the TSCR register in the application program. It does not consume power when the TSSD bit is set, so it is recommended to switch off the TS before entering power-saving sleep modes.

Figure 3-62. Temperature Shutdown Sensor Unit



The TS is an internal temperature shutdown circuit to avoid undefined device operation above the normal operating temperature range. The temperature is monitored by the TSRF flag in the MCUSR register only if the TSSD bit has been cleared. The TSRF flag is set in case the ambient temperature exceeds the specified threshold temperature (TSD, not calibrated in production) and the device can be forced into reset state. This immediately stops the operation of the device. As a result, all chip circuitry and port lines are disabled, including the interval timer. Once the device has entered the temperature shutdown mode only the on-chip temperature sensor circuitry (TS) is active.

Figure 3-63. Temperature Shutdown Characteristics



The typical threshold of TSRF is 120°C with a margin of ±5°C (not guaranteed because TSD is not calibrated yet) and the hysteresis range is from –5°C to –2°C (see Figure 3-63).

3.15.6.1 Temperature Sensor Control Register – TSCR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	**TSSD**	**TSCR**
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..1 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 0 – TSSD: Temperature Sensor Shutdown Mode Disable Bit
This bit controls the temperature shutdown mode. The TSSD bit must be written to logic zero to enable the shutdown mode. The temperature shutdown mode is disabled if the TSSD bit is set by writing a logic one. This mode is enabled after a power-on reset or an internal reset.

3.16 Voltage Monitor and Brown-Out Detection

The Voltage Monitor (VM) and Brown-Out Detection (BOD) consist of two separate comparators. Both comparators use the same internal voltage reference. If the same threshold voltage is selected for the BOD and VM, there is no guarantee that both VM and BOD detection is activated at the same V_{CC} voltage level because of the different comparator offset voltages.

3.16.1 Voltage Monitor (VM)

The VMEN bit in the VMCSR register enables or disables the voltage monitor.

The comparator for the VM has eight internal programmable thresholds which are selected by the VMLS[2..0] bits in the VMCSR register. The VMF flag indicates whether the supervised voltage is below or above the threshold. Additionally, an interrupt can be generated when the VMIM bit in the VMCSR register is set (see also Figure 3-64 on page 110).

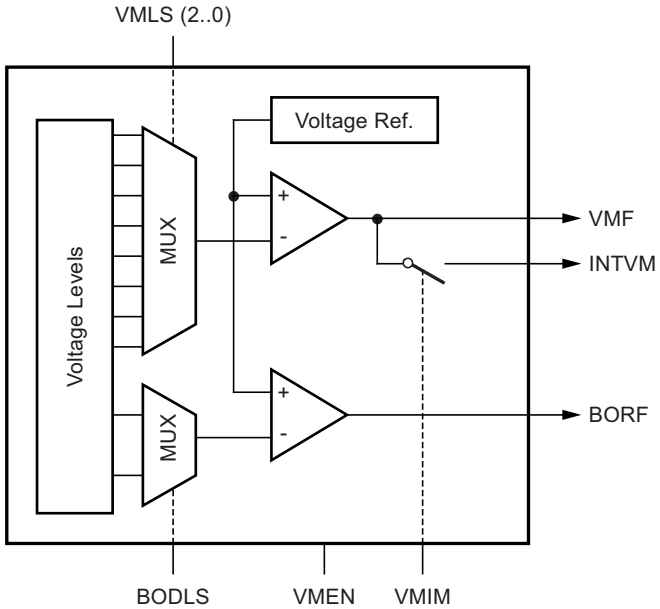
3.16.2 Brown-Out Detection (BOD)

The device has an embedded on-chip brown-out detection circuitry. The BOD circuitry monitors the VCC voltage level during operation. The trigger voltage level for the BOD can be selected by the BODLS bit in the VMCSR register. The trigger level has a hysteresis to ensure spike free brown-out detection. The hysteresis on the detection level should be interpreted as:

$V_{BOT+} = V_{BOT} + V_{HYST/2}$ and $V_{BOT-} = V_{BOT} - V_{HYST/2}$.

The BORF flag indicates if the supervised voltage is below this threshold (see Figure 3-64 on page 110). An internal reset is generated if the BORF bit in the MCUSR register is set. The BODPD bit in the VMCSR register can be used to enable the BOD during power-down mode.

Figure 3-64. Voltage Detector and Brown-Out Detector Stage



3.16.2.1 Voltage Monitor Control and Status Register – VMCSR

Bit	7	6	5	4	3	2	1	0	
	BODLS	**BODPD**	**VMF**	**VMIM**	**VMLS2**	**VMLS1**	**VMLS0**	**VMEN**	**VMCSR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7 – BODLS: Brown-Out Detection Level Select Bit
This bit selects the monitoring level of the programmable brown-out detection circuit as shown in Table 3-55.

Table 3-55. Brown-Out Detection Level Select Bit Description

BODLS	Description
0	1.8V
1	2.0V

Bits 6 – BODPD: Brown-Out Detection on Power-Down Bit
This bit enables the Brown-out detection circuit in the Power-down Mode, is shown in Table 3-56.

Table 3-56. Brown-Out Detection on Power-Down Enable Bit Description

BODPD	Description
0	Stop
1	Running

Bit 5 – VMF: Voltage Monitor Flag
This flag is set if the supervised voltage level is below the programmable threshold. VMF is automatically cleared when the voltage monitor interrupt vector is executed. Alternatively, VMF can be cleared by software if a logic one is written to its bit location.

Bit 4 – VMIM: Voltage Monitor Interrupt Mask Bit
Writing VMIM to one enables a VM interrupt if the I bit in SREG is set. Writing VMIM to zero disables the interrupt. The corresponding interrupt vector is executed when the VMF flag is set.

Bits 3..1 – VMLS2..0: Voltage Monitor Level Select Bits 2 - 0
These bits select the monitor levels of the voltage monitor circuit as shown in Table 3-57.

Table 3-57. Voltage Monitor Level Select Bit Description

VMLS2	VMLS1	VMLS0	Description
0	0	0	1.9V
0	0	1	2.0V
0	1	0	2.2V
0	1	1	2.4V
1	0	0	2.6V
1	0	1	2.9V
1	1	0	Reserved (3.6V)
1	1	1	Reserved (4.5V)

Bit 0 – VMEN: Voltage Monitor Enable Bit
This bit controls the VM. The VMEN bit must be written to logic one to enable the voltage monitor, and if the VMEN bit is written to logic zero, the voltage monitor is disabled.

3.17 LF Receiver

3.17.1 Features

- One input channel for the 1D antenna
- 1.4mV$_{RMS}$ sensitivity typically
- Two modes of preamble detection (continuous, burst) as wake-up for the Manchester decoder
- Three programmable wake-up modes for microcontroller (start gap (SG), SG or start burst (SB) plus header, SG or SB plus header (HD) plus identifier (ID))
- Integrated Manchester decoder for 3.90625Kb/s (±300b/s) baud rate
- Other baud rates possible via software and/or timer3 (transparent mode)
- Digital RSSI for field strength measurement
- Integrated resonance frequency tuning circuit for the 1D antenna

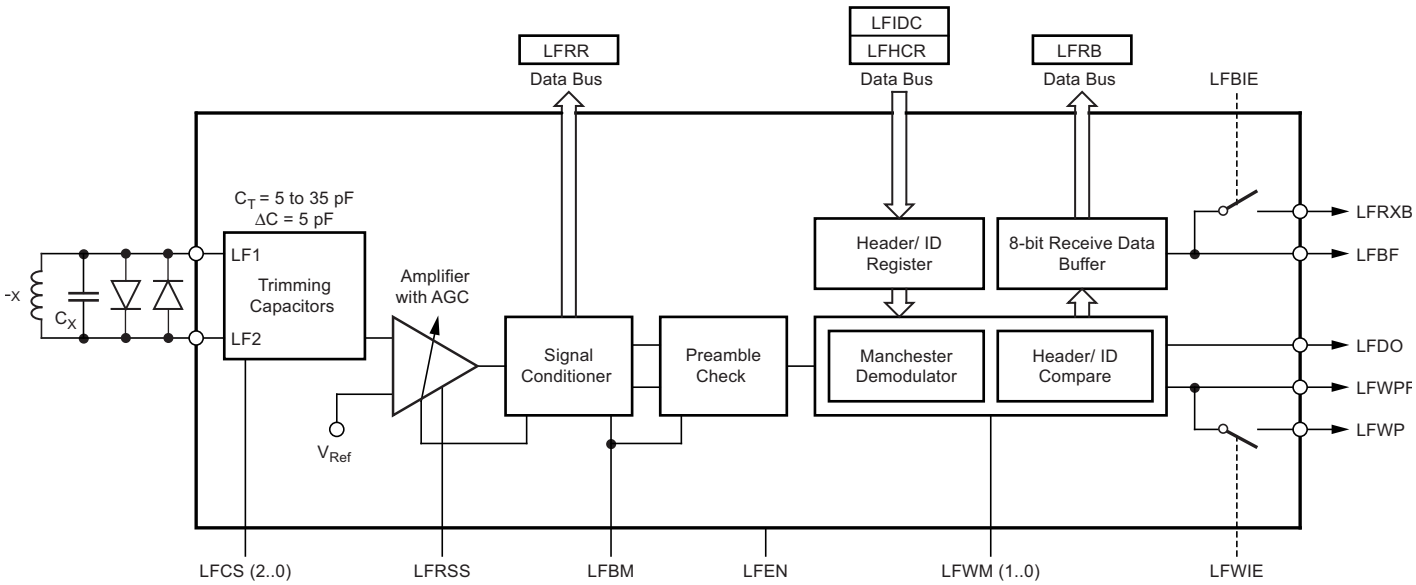
3.17.2 Functional Description

The LF receiver is an ultra-low-power ASK receiver for 125kHz. Without a carrier signal it operates in standby listen mode. In this mode it monitors the coil input with a very low current consumption. The IC supports two modes for waking up the Manchester decoder of the LF receiver—the continuous preamble mode and the burst preamble mode. To wake up the µC with the LF receiver in continuous preamble mode, the transmitter must send a continuous preamble carrier and a synchronization code which can be a Start Gap (SG), an SG followed by a header code or an SG followed by a header and identifier code. In burst preamble mode the transmitter must send a burst preamble, a synchronization code which can be a Start Burst (SB) followed by a header code or an SB followed by a header and identifier code. If no valid signal (code) is detected after a period of about 2ms, the LF receiver is reset and automatically returns to standby listen mode. For more details on the different modes of the LF receiver, see Section 3.17.3 “LF Receiver Wake-Up Modes for Microcontroller” on page 119f. The preamble detection timing is described in Figure 3-65 on page 112.

The on-chip Manchester demodulator decodes the received data stream and compares it with the programmable bit pattern in the LFIDC and/or LFHCR registers. If a valid header or header and identifier frame is detected in the incoming data stream, the Manchester decoder generates an LF receiver wake-up interrupt (LFWP). The successive data are decoded in 8-bit frames and are buffered in the 8-bit receive data register (LFRB). The LFBF flag in the LFFR register indicates that the buffer is full and, in addition, can generate an interrupt (LFRXB) (see also Figure 3-65 on page 112).

Other data protocols and rates are supported by using the LFDO bit in the LFCDR register. This bit can be directed to timer3 as capture input (see Table 3-42 on page 90). The LFDO bit contains the raw data from the receiver front end, i.e., Manchester decoder and header/ID compare circuits are not in use. In this case timer3 is always on and synchronization, identification and decoding must be carried out via software, which consumes some more current. By using the LFDO bit and the transparent mode of the LF receiver (LFWM [1..0] = 11 in the LFRCR register) baud rates < 500b/s can be handled. In this case it is necessary to reset the LF receiver and return it to standby listen mode by setting the LFRST bit to one in the LFCDR register. For more details on this mode, see Section 3.17.3.6 “Wake-Up Mode 6 (LFWM[1..0] = 11 ; LFBM = 0/1):” on page 122.

Figure 3-65. LF Receiver Block

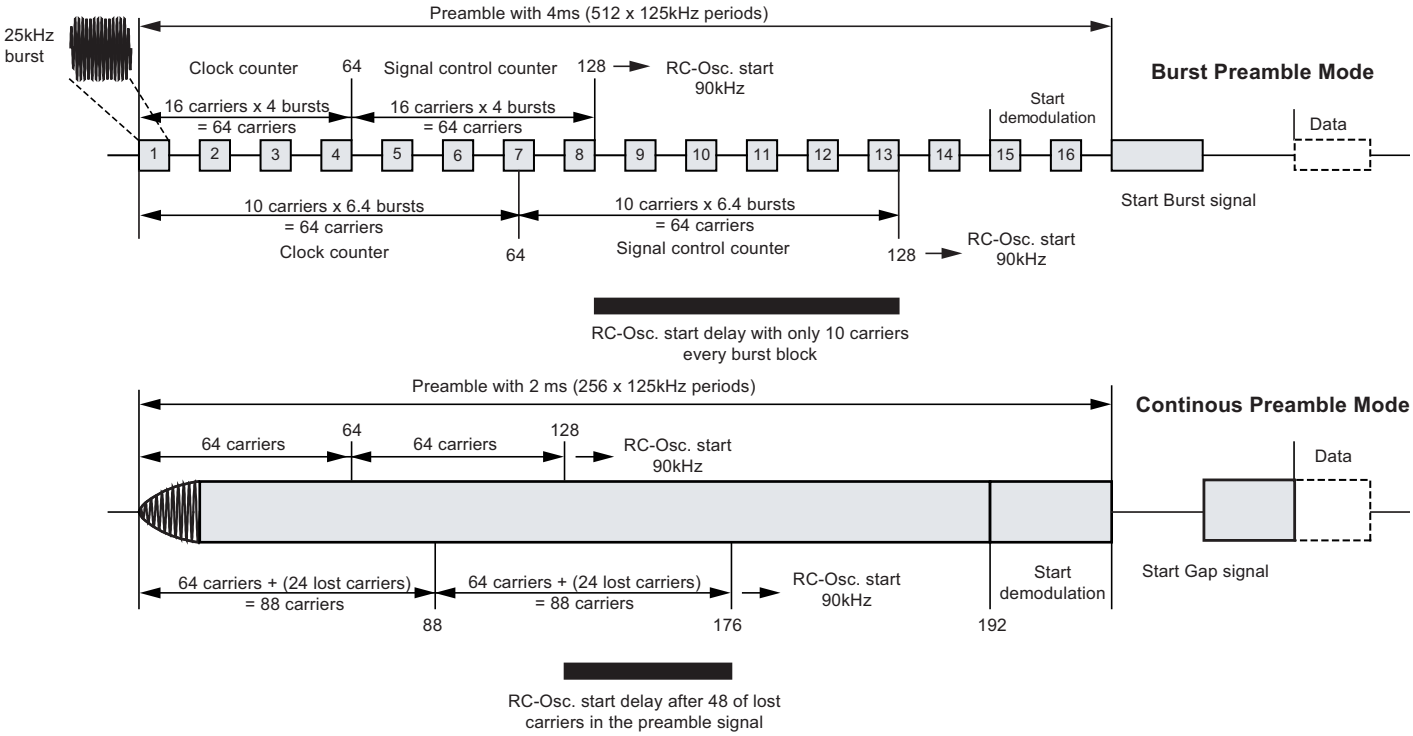


It is recommended to use antiparallel clamping diodes with threshold voltages < 0.7V at the LF input pins to avoid high currents into the LF input pins (which appears at very high field strength).

3.17.2.1 Preamble Detection

To prevent the circuit from unintended operations in a noisy environment the preamble detection circuit checks the input signal. A valid signal is detected by a counter after 128 carrier periods (125kHz periods) without interrupts longer than eight carrier periods in continuous preamble mode and without interrupts longer than 24 carrier periods in burst preamble mode. In the event of longer interrupts the counter is reset. It is recommended that the complete preamble has at least 256 carrier periods in continuous preamble mode and at least 512 carrier periods in burst preamble mode (16 periods ON and 16 periods OFF) (see Figure 3-66 on page 113). When a preamble is detected the 90kHz RC oscillator is started. Upon start of the 90kHz RC oscillator the Manchester decoder is woken up. Then the decoder waits for a valid start gap or start burst. After a start gap or start burst has been detected, the data transfer starts according to the selected wake-up mode which is based on Table 3-60 on page 117. For further details on the protocol and timing of the different wake-up modes of C by the LF receiver, see Section 3.17.3 “LF Receiver Wake-Up Modes for Microcontroller” on page 119.

Figure 3-66. Preamble Timing in the LF Telegram in Burst and Continuos Preamble Mode



3.17.2.2 RSSI – Field Strength Measurement and Trimming Capacitance

The LF receiver contains a digital RSSI field strength measurement circuit. With the RSSI value in the LFRR register the resonance frequency can be optimized to 125kHz by switching the on-chip capacitance according to Table 3-58 on page 116 in addition to the external LC circuit.

The RSSI value is generated by the digital Automatic Gain Control (AGC) circuit. By setting the LFSCE bit in the LFCDR register this value is written into the LFRR register. When loading the RSSI data to the LFRR register is completed, the LFRF flag in the LFFR register is set (one) and the value is readable via software.

3.17.2.3 AGC Amplifier

To achieve data rates of 3.90625kb/s for input signals of about 1.4mV$_{rms}$ to 700mV$_{rms}$, it is necessary to control the gain of the amplifier. The input stage contains an amplifier with a built-in digital Automatic Gain Control (AGC). It is used to adapt the gain to the incoming signal strength and is also used as a digital RSSI for field strength measurements which can be read by the microcontroller (see Section 3.17.2.2 “RSSI – Field Strength Measurement and Trimming Capacitance” on page 113). The gain control circuit regulates the internal signal amplitude to the regulated reference value (gain control reference = 160mV). It decreases the gain by one step if the internal signal exceeds the reference level for one period and it increases the gain by one step if four periods do not achieve the reference level. Regulation starts if the internal signal exceeds the 50% gain control reference level (gap detection reference = 80mV) and stops if the internal signal falls below the gap detection reference (see Figure 3-68 on page 114).

Note: With the variation of gain the coil input impedance changes from high impedance to minimal 80kΩ because of the internal adjustable damping circuit which regulates the gain (see Figure 3-67 on page 114).

Figure 3-67. Coil Input Impedance

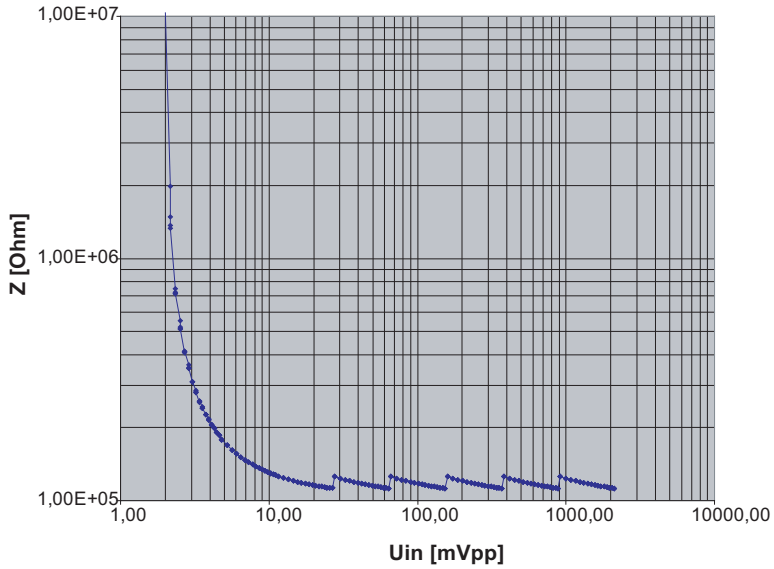
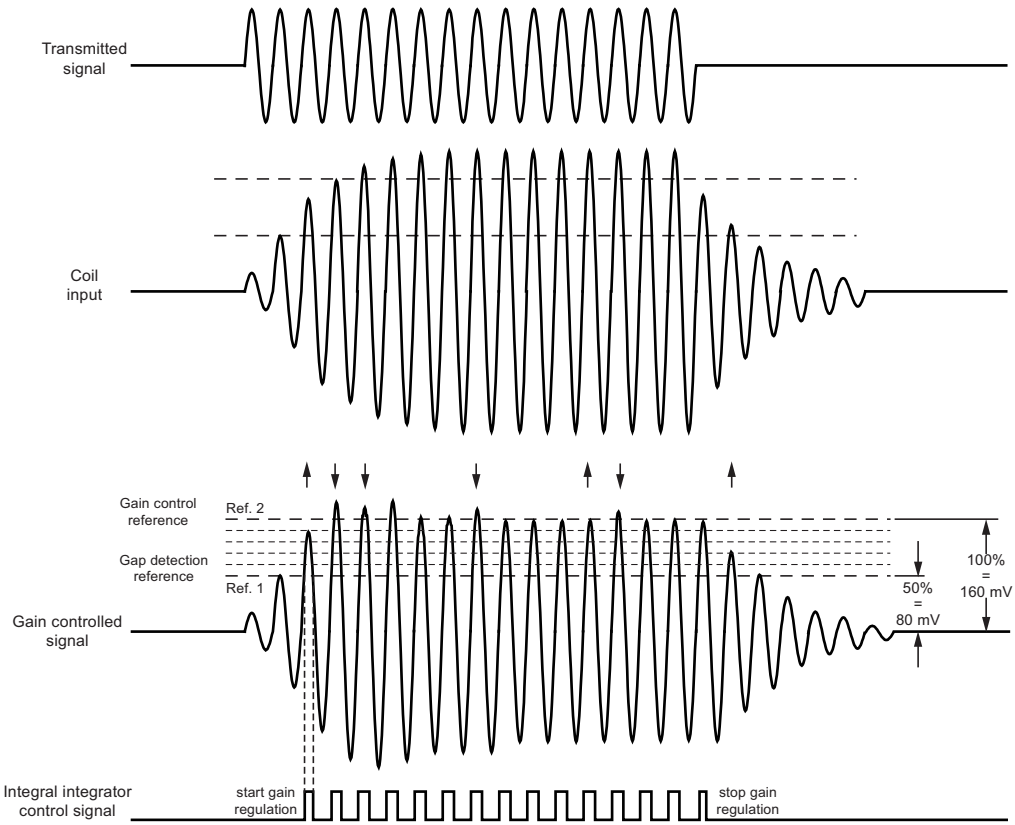


Figure 3-68. Automatic Gain Control



3.17.2.4 Signal Conditioner

The signal conditioner demodulates the amplifier output signal and converts it to a binary signal. It compares the carrier signal with the 50% gain control reference level (80mV) (see Figure 3-69) and delivers a logic zero if the carrier signal stays below the reference level and a logic one if it exceeds the reference level. A smoothing filter generates the envelope signal by suppressing the spaces between the half waves as well as a few missing periods in the carrier and glitches during the gaps.

The output signal of the signal conditioner is used as the internal data signal for data output, wake-up logic and preamble detection.

The delay times t_{on} and t_{off} caused by the rise and fall times of the antenna signal are a function of the Q factor (see Figure 3-70 and Figure 3-71 on page 116).

Figure 3-69. Output Timing

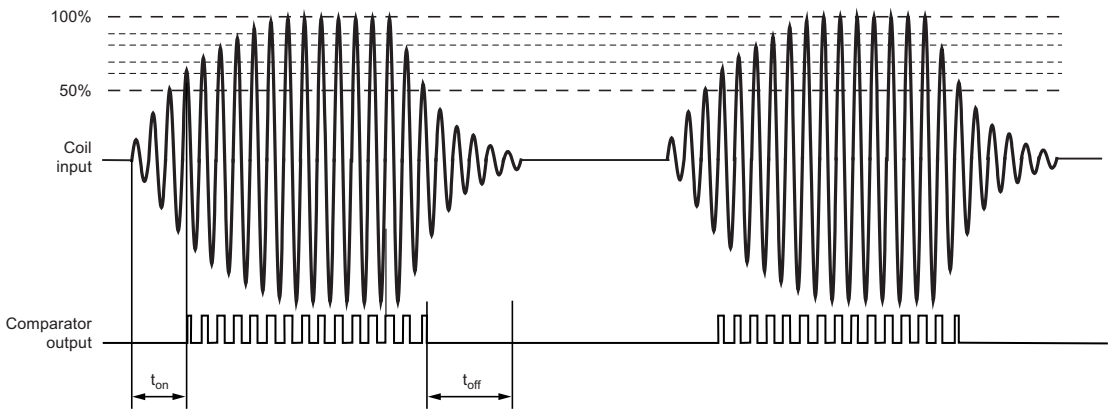


Figure 3-70. Turn-on Delay Time versus Antenna Q Factor (Simulated)

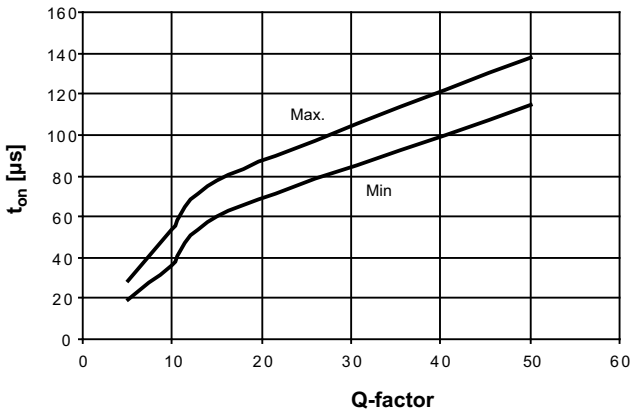
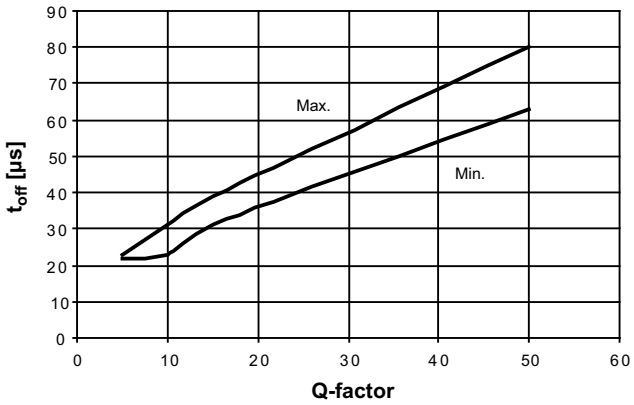


Figure 3-71. Turn-off Delay Time versus Antenna Q Factor (Simulated)

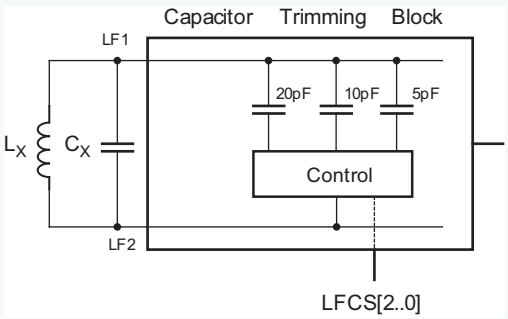


3.17.2.5 Low-Frequency Receiver Control Register – LFRCR

Bit	7	6	5	4	3	2	1	0	
	LFCS2	**LFCS1**	**LFCS0**	**LFRSS**	**LFWM1**	**LFWM0**	**LFBM**	**LFEN**	**LFRCR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..5 – LFCS[2..0]: LF Receiver Capacitance Select Bits 2 - 0
These bits select the capacitance value of the programmable trimming capacitors for the external LC circuit (antenna) as shown in the following Table 3-58.

Table 3-58. Trimming Capacitance Values Select Bit Description

LFCS2	LFCS1	LFCS0	Description	LF Receiver Input Block
0	0	0	0pF	
0	0	1	5pF	
0	1	0	10pF	
0	1	1	15pF	
1	0	0	20pF	
1	0	1	25pF	
1	1	0	30pF	
1	1	1	35pF	

Bits 4 – LFRSS: LF Receiver Sensitivity Select Bit
This bit selects the sensitivity level of the LF receiver input channel (see Table 3-59).

Table 3-59. LF Receiver Sensitivity Adjustment

LFRSS	Default Sensitivity Value
0	High sensitivity (default)
1	Low sensitivity

Bits 3 to 2 – LF receiver Wake-Up Mode Bits 1 to 0
These bits select the LF receiver wake-up modes for microcontroller as shown in the following Table 3-60.

Table 3-60. Wake-Up Mode Select Bit Description

LFWM1	LFWM 0	Description
0	0	Start Gap (SG)
0	1	SG or Start Burst (SB) + Header
1	0	SG or SB + Header + Identifier
1	1	Transparent model

Bit 1 – LFBM: LF Receiver Burst Mode Enable Bit
This bit controls the LF receiver preamble mode. The LFBM bit must be written to logic one to enable the burst preamble mode. If the LFBM bit is written to logic zero, the continuous preamble mode is enabled.

Bit 0 – LFEN: LF Receiver Enable Bit
This bit controls the LF receiver. The LFEN bit must be written to logic one to enable the LF receiver. If the LFEN bit is written to logic zero, the LF receiver is disabled.

3.17.2.6 Low-Frequency Receiver Control and Data Register – LFCDR

Bit	7	6	5	4	3	2	1	0	
	LFSCE	**LFRST**	**-**	**-**	**-**	**-**	**-**	**LFDO**	**LFCDR**
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – LFSCE: LF Receiver RSSI Software Capture Enable Bit
The LFSCE bit must be written to logic one to enable a single capture event of the 7-bit field strength measurement value of the digital AGC to the RSSI data register (LFRR) if the LFEN bit is set to logic one. The LFSCE bit is cleared after the RSSI value is saved in the LFRR register. The RSSI value is readable via its RSSI data register while the LF receiver is running.

Bit 6 – LFRST: LF Receiver Reset Bit
In transparent mode (LFWM[1..0] = 11) this bit must be written to logic one (via software) to reset the LF receiver front end. After writing this bit to logic zero again, the LF receiver is back in standby listen mode. In all other wake-up modes, this bit is ignored and the LF receiver returns to standby listen mode automatically.

Bits 6 ... 1 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 0 – LFDO: LF Receiver Data Output Bit
The demodulated data values of the LF receiver input signal is readable via the LFDO bit of the LFCDR register after the receiver has detected a start gap.

3.17.2.7 Low-Frequency Receive Data Buffer – LFRB

Bit	7	6	5	4	3	2	1	0	
	LFRB[7..0]								**LFRB**
Read	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The LFRB is updated with 8-bit received data frames that are decoded by the on-chip Manchester decoder.

3.17.2.8 Low-Frequency RSSI Data Register – LFRR

Bit	7	6	5	4	3	2	1	0	
	-	LFRR[6..0]							LFRR
Read	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The LFRR is updated with the 7-bit field strength measurement value of the digital AGC when the LFSCE bit is written to logic one.

3.17.2.9 Low-Frequency Header Compare Register – LFHCR

Bit	7	6	5	4	3	2	1	0	
	-	LFHCR[6..0]							LFHCR
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The LFHCR contains 7-bit data of the header compare value.

It is recommended not to use the header or identifier periodically in order to avoid failure of the LF receiver.

3.17.2.10 Low-Frequency Identifier Compare Register – LFIDC

Bit	7	6	5	4	3	2	1	0	
	LFIDCH [15..8]								LFIDCH
	LFIDCL [7..0]								LFIDCL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The LFIDCH register contains the 16-bit identifier compare value. To do a 16-bit write, the high byte must be written before the low byte. For a 16-bit read, the low byte must be read before the high byte.

It is recommended not to use the header or identifier periodically in order to avoid failure of the LF receiver.

3.17.2.11 Low-Frequency Interrupt Mask Register – LFIMR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	LFEIM	LFBIM	LFWIM	LFIMR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..3 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 2 – LFEIM: LF Receiver End of Data Interrupt Mask Bit
Writing LFEIM to one enables the LF receiver end of burst interrupt if the I bit in SREG is set. Writing LFEIM to zero disables the interrupt. The corresponding interrupt vector is executed when the LFEDF flag, located in the LFFR register, is set.

Bit 1 – LFBIM: LF Receiver Data Buffer Interrupt Mask Bit
Writing LFBIM to one enables the LF receiver receive buffer interrupt if the I bit in SREG is set. Writing LFBIM to zero disables the interrupt. The corresponding interrupt vector is executed when the LFBF flag, located in the LFFR register, is set.

Bit 0 – LFWIM: LF Receiver Wake-up Interrupt Mask Bit
Writing LFWIM to one enables an LF receiver wake-up interrupt (LFWP) if the I bit in SREG is set. Writing LFWIM to zero disables the interrupt. The corresponding interrupt vector is executed when the LFWPF flag, located in the LFFR register, is set.

3.17.2.12 Low-Frequency Flag Register – LFFR

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	LFRF	LFEDF	LFBF	LFWPF	LFFR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7…4 – Res: Reserved Bits
These bits are reserved bits on the ATA6289 and are always read as zero.

Bit 3 – LFRF: LF Receiver RSSI Data Flag
This flag is set when the 7-bit field strength measurement value of the digital AGC has been loaded into the RSSI data register (LFRR). It can be cleared by writing a logic one to its bit location.

Bit 2 – LFEDF: LF Receiver End of Data Flag
This flag is set when the data decoder has detected a bit error of the modulation data stream. It can also be set if a regular data stream has been received correctly or if a timeout violation has been detected. The LFEDF is cleared automatically if the LF Receiver End Of Burst Interrupt Vector (LFREOB) is executed. Alternatively, the LFEDF can be cleared by writing a logic one to its bit location.

Bit 1 – LFBF: LF Receiver Data Buffer Full Flag
This flag is set when the on-chip Manchester decoder has loaded a decoded 8-bit dataframe to the receive data buffer (LFRB). The LFBF is cleared automatically if the Receive Buffer Interrupt Vector of the LF receiver (LFRXB) is executed. Alternatively, the LFBF can be cleared by writing a logic one to its bit location.

Bit 0 – LFWPF: LF Receiver Wake-Up Flag
This flag is set once the LF receiver is woken up. The LFWPF is cleared automatically if the LF Receiver Wake-Up Interrupt Vector (LFWP) is executed. Alternatively, LFWPF can be cleared by writing a logic one to its bit location.

3.17.3 LF Receiver Wake-Up Modes for Microcontroller

The combination of the LFBM bit and the LFWM[1..0] bits leads to six different microcontroller wake-up modes, which are described below. For the first five modes it is recommended to use the Manchester decoder to reduce the power consumption to a minimum. In this case the LFWP interrupt can be used to wake-up the microcontroller. If it is necessary to have other data rates and protocols, the LFDO signal can also be used as capture input for timer3 to demodulate incoming data. In this case timer3 capture interrupt (T3CAP) can be used to wake-up the microcontroller (see also transparent mode (wake-up mode 6). This consumes more power because timer3 (with an oscillator as an input clock) must be permanently active and more codes are needed to handle incoming data. The examples below for the first five wake-up modes are included in an active Manchester decoder.

The sixth wake-up mode is a special mode insert for handling data rates of < 500b/s. In this case resetting the front end of the LF receiver in standby listen mode must be carried out with the LFRST bit in the LFCDR register via software (is not done automatically by the LF receiver.) Because of data rates not equal to 3.90625kb/s the Manchester decoder cannot be used, and the demodulation of incoming data should be done with the LFDO signal as capture input for timer3. As mentioned above, in this case the T3CAP interrupt can be used to wake-up the microcontroller.

For all modes with an active Manchester decoder the header, identifier and data must be transmitted in Manchester code with rising edge interpreted as logic one and a data rate of 3.90625kb/s (±300b/s).

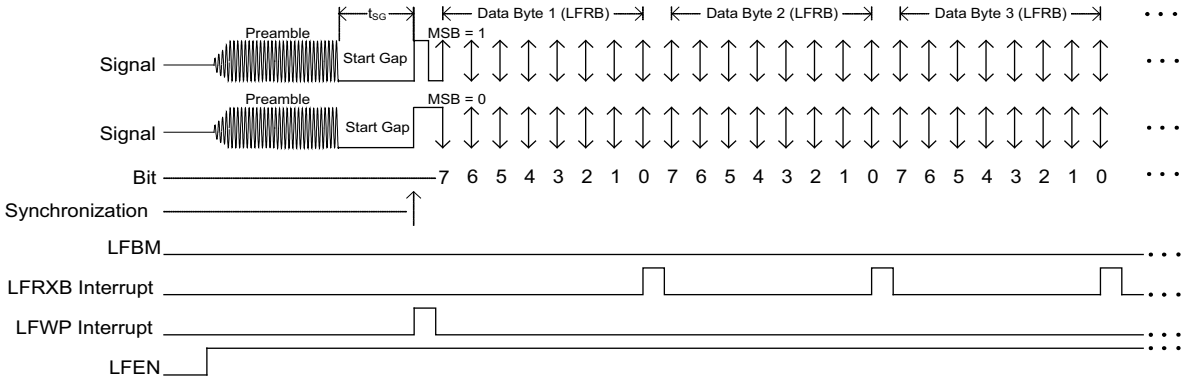
The different signal lengths indicated in the Figures below (Figure 3-72 on page 120 - Figure 3-78 on page 123) derive from the slow RC oscillator frequency which can be in the range of 81kHz < f_{SRC} < 99kHz (typically f_{SRC} = 90kHz) (see Section 5.2 “Operating Characteristics of Atmel ATA6289” on page 169).

3.17.3.1 Wake-Up Mode 1 (LFWM[1..0] = 00 ; LFBM = 0)

After a continuous preamble signal is detected the Manchester decoder starts and waits for a valid start gap (t_{SG}). If the rising edge of the enveloped signal of the first burst after start gap appears within the defined time t_{SG}, the decoder generates an LF Receiver Wake-Up Interrupt (LFWP) if LFWIM bit is set in the LFIMR register. This is when the Manchester decoder is synchronized and data can then be transmitted in Manchester code with 3.90625kb/s. For each 8-bit received data frame an LF Receiver Receive Buffer Interrupt (LFRXB) is generated if the LFBIM bit is set in the LFIMR register, see Figure 3-72 on page 120.

If no valid start gap is detected or if there are data gaps > 2.03ms, the LF receiver returns to standby listen mode automatically. The same occurs if the LFEN bit in the LFRCR register is set to zero.

Figure 3-72. LF Receiver in Continuous Mode with Start Gap Wake-Up



$10 \times 1/f_{SRC\_min}$ periods < t_{SG} < $200 \times 1/f_{SRC\_max}$ periods

with f_{SRC_min} = 81kHz and f_{SRC_max} = 99kHz

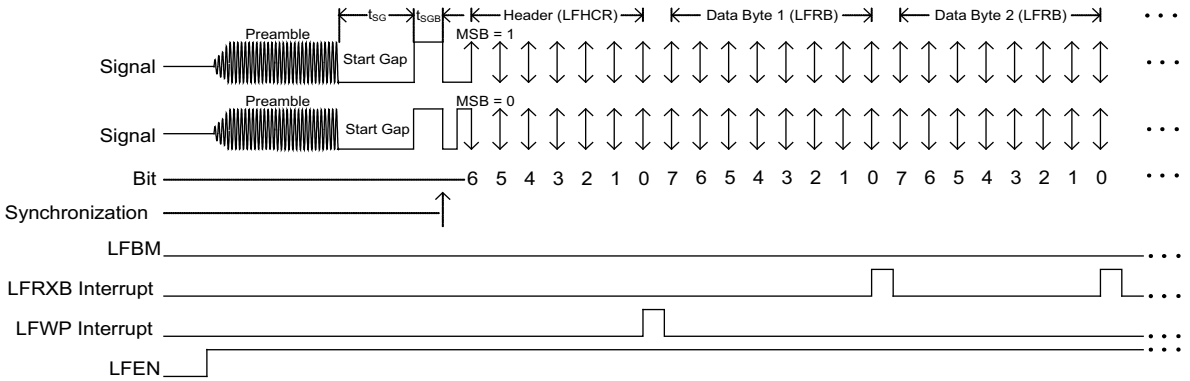
--> 124µs < t_{SG} < 2.03ms

3.17.3.2 Wake-Up Mode 2 (LFWM[1..0] = 01 ; LFBM = 0):

After a continuous preamble signal has been detected the Manchester decoder starts and waits for a valid start gap (t_{SG}). At the falling edge of the enveloped signal of the first burst (with t_{SGB}) after start gap, the Manchester decoder is synchronized and then a valid 7-bit header must be transmitted in Manchester code at 3.90625kb/s.If the LFWIM bit is set in the LFIMR register, an LF Receiver Wake-Up Interrupt (LFWP) is generated at the last bit of the valid header. The followed data is received in 8-bit frames and can generate an LF Receiver Receive Buffer Interrupt (LFRXB) if the LFBIM bit is set in the LFIMR register (see Figure 3-73).

If no valid start gap plus header is detected or if there are data gaps > 2.03ms, the LF receiver returns to standby listen mode automatically. The same happens if the LFEN bit in the LFRCR register is set to zero.

Figure 3-73. LF Receiver in Continuous Mode with Start Gap plus Header Wake-Up



$10 \times 1/f_{SRC\_min}$ periods < t_{SG} < $200 \times 1/f_{SRC\_max}$ periods
1/3.90625kb/s < t_{SGB} < $200 \times 1/f_{SRC\_max}$ periods

with f_{SRC_min} = 81kHz and f_{SRC_max} = 99kHz

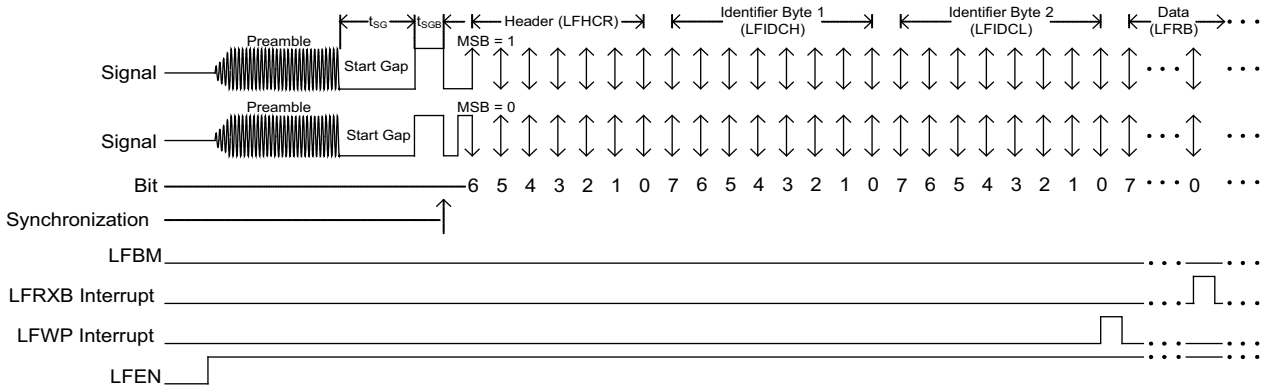
--> 124µs < t_{SG} < 2.03ms and 256µs < t_{SGB} < 2.03ms

3.17.3.3 Wake-Up Mode 3 (LFWM[1..0] = 10 ; LFBM = 0):

After a continuous preamble signal has been detected the Manchester decoder starts and waits for a valid start gap (t_{SG}). At the falling edge of the enveloped signal of the first burst (with t_{SGB}) after start gap, the Manchester decoder is synchronized and then a valid 7-bit header plus a 16-bit identifier must be transmitted in Manchester code at 3.90625kb/s. If the LFWIM bit is set in the LFIMR register, an LF Receiver Wake-up Interrupt (LFWP) is generated at the last bit of the valid header plus identifier. The followed data is received in 8-bit data frames and can generate an LF Receiver Receive Buffer Interrupt (LFRXB) if the LFBIM bit is set in the LFIMR register (see Figure 3-74).

If no valid start gap plus header plus identifier is detected or if there are data gaps > 2.03ms, the LF receiver returns to standby listen mode automatically. The same happens if the LFEN bit in the LFRCR register is set to zero.

Figure 3-74. LF Receiver in Continuous Mode with Start Gap plus Header plus Identifier Wake-up



$10 \times 1/f_{SRC\_min}$ periods < t_{SG} < $200 \times 1/f_{SRC\_max}$ periods
1/3.90625kb/s < t_{SGB} < $200 \times 1/f_{SRC\_max}$ periods

with f_{SRC_min} = 81kHz and f_{SRC_max} = 99kHz

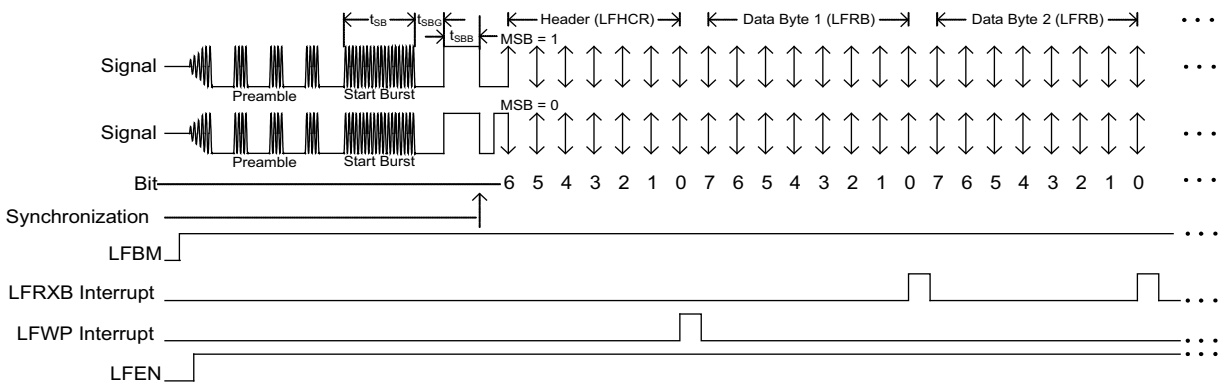
--> 124µs < t_{SG} < 2.03ms and 256µs < t_{SGB} < 2.03ms

3.17.3.4 Wake-Up Mode 4 (LFWM[1..0] = 01 ; LFBM = 1):

After a burst preamble signal has been detected the Manchester decoder starts and waits for a valid start burst (t_{SB}) followed by a gap (t_{SBG}). At the falling edge of the enveloped signal of the first burst (with t_{SBB}) after this gap, the Manchester decoder is synchronized and then a valid 7-bit header must be transmitted in Manchester code at 3.90625kb/s. If LFWIM bit is set in the LFIMR register, an LF Receiver Wake-up Interrupt (LFWP) is generated at the last bit of the valid header. The followed data is received in 8-bit frames and can generate an LF Receiver Receive Buffer Interrupt (LFRXB) if the LFBIM bit is set in the LFIMR register (see Figure 3-75 on page 121).

If no valid start burst plus header is detected or if there are data gaps > 2.03ms, the LF receiver returns to standby listen mode automatically. The same happens if the LFEN bit in the LFRCR register is set to zero.

Figure 3-75. LF Receiver in Burst Mode with Start Gap plus Header Wake-Up



$48 \times 1/f_{SRC\_min}$ periods < t_{SB} < $200 \times 1/f_{SRC\_max}$ periods
$10 \times 1/f_{SRC\_min}$ periods < t_{SBG} < $100 \times 1/f_{SRC\_max}$ periods

1/3.90625kb/s < t_{SBB} < $200 \times 1/f_{SRC\_max}$ periods
with f_{SRC_min} = 81kHz and f_{SRC_max} = 99kHz

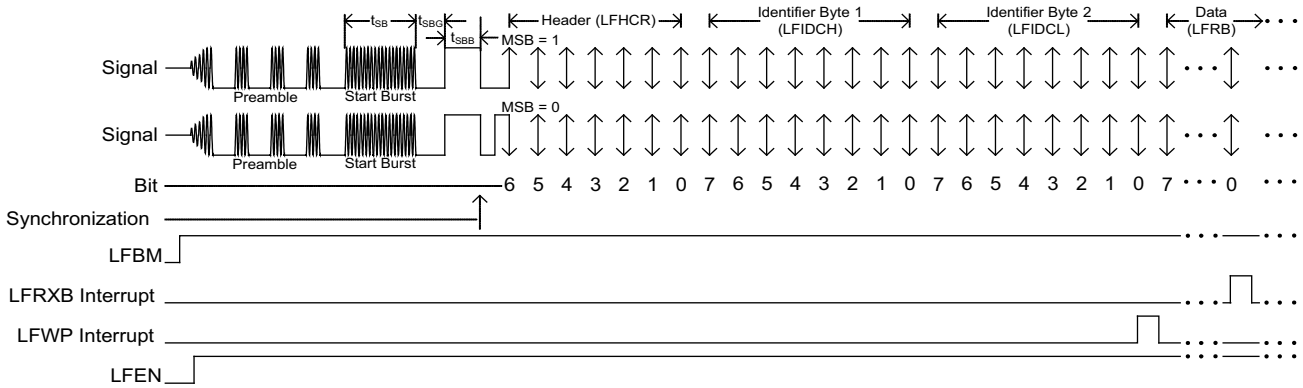
--> 593µs < t_{SB} < 2.03ms and 124µs < t_{SBG} < 1.015ms and 256µs < t_{SBB} < 2.03ms

3.17.3.5 Wake-up Mode 5 (LFWM[1..0] = 10 ; LFBM = 1):

After a burst preamble signal has been detected the Manchester decoder starts and waits for a valid start burst (t_{SB}) followed by a gap (t_{SBG}). At the falling edge of the enveloped signal of the first burst (with t_{SBB}) after this gap, the Manchester decoder is synchronized and then a valid 7-bit header plus 16-bit identifier must be transmitted in Manchester code at 3.90625kb/s. If LFWIM bit is set in the LFIMR register, an LF Receiver Wake-up Interrupt (LFWP) is generated at the last bit of the valid header plus identifier. The followed data is received in 8-bit data frames and can generate an LF Receiver Receive Buffer Interrupt (LFRXB) if the LFBIM bit is set in the LFIMR register (see Figure 3-76).

If no valid start burst plus header plus identifier is detected or if there are data gaps > 2.03ms, the LF receiver returns to standby listen mode automatically. The same happens if the LFEN bit in the LFRCR register is set to zero.

Figure 3-76. LF Receiver in Burst Mode with Start Gap plus Header plus Identifier Wake-Up



$48 \times 1/f_{SRC\_min}$ periods < t_{SB} < $200 \times 1/f_{SRC\_max}$ periods
$10 \times 1/f_{SRC\_min}$ periods < t_{SBG} < $100 \times 1/f_{SRC\_max}$ periods

1/3.90625kb/s < t_{SBB} < $200 \times 1/f_{SRC\_max}$ periods
with f_{SRC_min} = 81kHz and f_{SRC_max} = 99kHz

--> 593µs < t_{SB} < 2.03ms and 124µs < t_{SBG} < 1.015ms and 256µs < t_{SBB} < 2.03ms

3.17.3.6 Wake-Up Mode 6 (LFWM[1..0] = 11 ; LFBM = 0/1):

This mode is inserted especially for handling data rates of < 500b/s, but higher data rates are also possible. In addition, data does not need to be Manchester coded, i.e., other encoding is also possible.

With the transparent mode there are several degrees of freedom to handle incoming data. These cannot be described in full in this datasheet, therefore only two examples are given below.

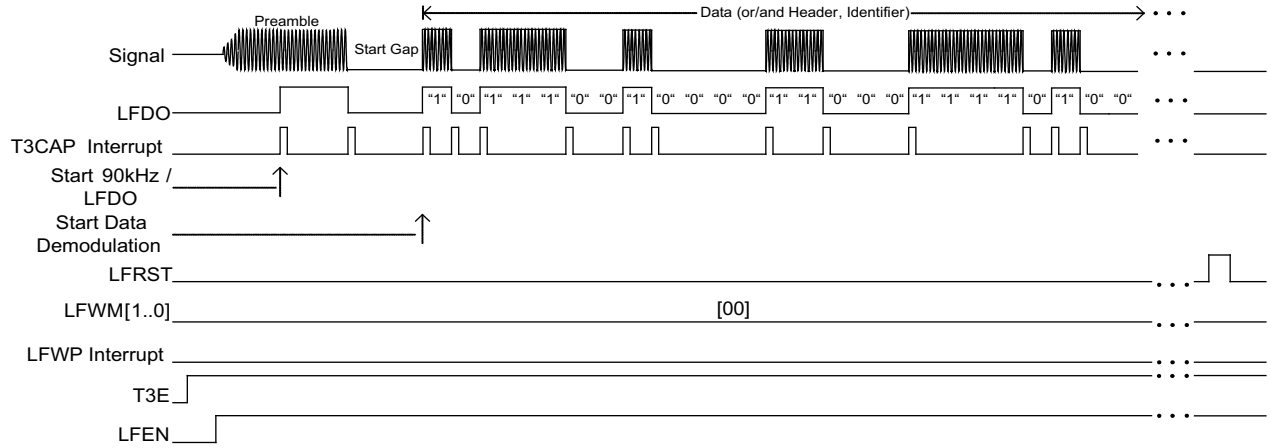
In the examples the continuous preamble mode is used (LFBM = 0). This mode makes it is easier to achieve a defined start condition. In order to save power it is also recommended to use timer3 clocked with the lowest possible frequency, i.e., with a frequency that is just great enough to handle the desired data rate. Timer3, for example, can be clocked with the SRC oscillator or with the output of the low-power timer0. The LFDO signal is used as the capture input for timer3 to demodulate the data.

Important: Once the transparent mode is entered, it is recommended to disable the LF receiver wake-up interrupt to avoid undefined interrupts during the decoding of incoming data. After exiting the transparent mode, the LF receiver wake-up interrupt LFWP can be enabled again.

First example (Figure 3-77):
After a valid preamble has been detected the 90kHz SRC oscillator is started by the LF front end. At the same time the LFDO signal is available and generates the first Timer3 Capture Interrupt (T3CAP). This interrupt can be used, for example, to initialize a gap detection. After the following two successive T3CAP interrupts that meet the gap detection condition, the demodulation of data can start. Gap detection, header and/or identifier compare (if desired) and data demodulation must be carried out by examining the timer3 value during each T3CAP interrupt. After the data transmission has been completed, LF receiver must be reset to standby listen mode with the LFRST bit in the LFCDR register to await the next LF frame.

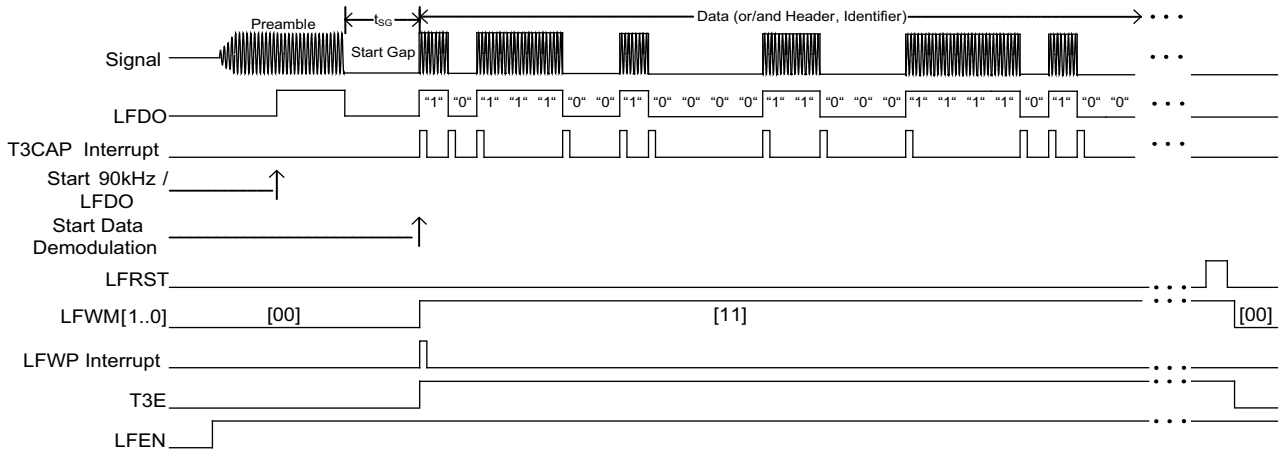
In this example timer3 (with oscillator) must always be enabled (T3E set to logic one).

Figure 3-77. LF Receiver in Transparent Mode with T3CAP Interrupt Wake-up



Second example (Figure 3-78):
A more effective way is to start the LF receiver using mode 1 (LFWM[1..0] = 00 ; LFBM = 0). The LF receiver automatically generates an LFWP interrupt wake-up (see Figure 3-77 on page 123). During the interrupt routine it is possible to switch to the transparent Mode (LFWM[1..0] = 11). Timer3 (with oscillator) must be enabled (T3E set to logic one) and the data demodulation can be carried out as described in the first example (Figure 3-77 on page 123). After the data transmission is completed, the LF receiver must be reset to standby listen mode with the LFRST bit in the LFCDR register. Timer3 (with oscillator) can be disabled (T3E set to logic zero) and the LF receiver must be set back to mode 1 (LFWM[1..0] = 00 ; LFBM = 0) to await the next LF frame.

Figure 3-78. LF Receiver Start in Continuous Mode with Start Gap Wake-Up then Change to Transparent Mode



$124\mu s < t_{SG} < 2.03ms$

3.18 debugWIRE – On-Chip Debug System

3.18.1 Features

- Complete program flow control
- Emulates all on-chip functions, both digital and analog, except $\overline{\text{RESET}}$ pin
- Real-time operation
- Symbolic debugging support (both at C and assembler source level, or for other HLLs)
- Unlimited number of program break points (using software break points)
- Non-intrusive operation
- Electrical characteristics identical to real device
- Automatic configuration system
- High-speed operation
- Programming of non-volatile memories

3.18.2 Overview

The debugWIRE on-chip debug system uses a one-wire, bi-directional interface to control the program flow, execute Atmel® AVR® instructions in the CPU and to program different non-volatile memories.

3.18.3 Physical Interface

When the debugWIRE Enable (DWEN) fuse is programmed and lock bits are unprogrammed, the debugWIRE system within the target device is activated. The $\overline{\text{RESET}}$ port pin is configured as a wire AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.

Figure 3-79. The debugWIRE Setup

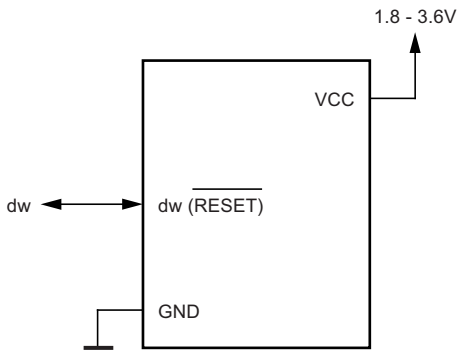


Figure 3-79 shows the schematic of a target MCU with debugWIRE enabled and the emulator connector. The system clock is not affected by the debugWIRE and is always the clock source selected by the CCS, CMM[1..0], CLKPS[2..0] bits and the CKDIV8, FRCFS fuses.

When designing a system in which debugWIRE is used, the following observations must be made for correct operation:

- Pull-up resistors on the dW/($\overline{\text{RESET}}$) line must not be smaller than 10kΩ. The pull-up resistor is not required for debugWIRE functionality.
- Connecting the $\overline{\text{RESET}}$ pin directly to V_{CC} does not work.
- Capacitors connected to the $\overline{\text{RESET}}$ pin must be disconnected when using debugWire.
- All external reset sources must be disconnected.

3.18.4 Software Break Points

By means of the Atmel® AVR® break instruction the debugWIRE supports the program memory break points. Setting a break point in Atmel AVR Studio® inserts a BREAK instruction into the program memory. The BREAK instruction which replaces the original instruction is stored. When program execution is continued, the stored instruction is executed before continuing from the program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The Flash must be re-programmed each time a break point is changed. This is automatically handled by Atmel AVR Studio through the debugWIRE interface. The use of break points therefore reduces Flash data retention. Devices used for debugging purposes should not be shipped to end customers.

3.18.5 Limitations of debugWIRE

The debugWIRE communication pin (dW) is physically located at the same pin as External Reset ($\overline{\text{RESET}}$). An external reset source is therefore not supported when the debugWIRE is enabled.

The debugWIRE system accurately emulates all I/O functions when running at full speed, i.e., when the program in the CPU is running. When the CPU is stopped, care must be taken while accessing some of the I/O registers via the debugger (for example, with Atmel AVR Studio).

A programmed DWEN fuse enables some parts of the clock system to run in all sleep modes. This increases power consumption during sleep mode. Thus, the DWEN fuse should be disabled when debugWire is not being used.

3.18.6 debugWIRE Related Register in I/O Memory

The following section describes the registers used with the debugWire.

3.18.6.1 debugWire Data Register – DWDR

Bit	7	6	5	4	3	2	1	0	
	DWDR [7..0]								DWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The DWDR register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in normal operations.

3.19 Boot Loader Support – Read-While-Write Self-Programming

In the ATA6289, the boot loader support provides a real read-while-write self-programming mechanism for downloading and uploading a program code by the MCU itself. This feature allows flexible application software updates controlled by the MCU using a Flash-resident boot loader program. The boot loader program can use any available data interface and associated protocol to read code and write (program) that code into the Flash memory, or read the code from the program memory. The program code within the boot loader section has the capability to write into the entire Flash, including the boot loader memory. Thus the boot loader can even modify itself, and it can also erase itself from the code if the feature is not needed anymore. The size of the boot loader memory is configurable with fuses and the boot loader has two separate sets of boot lock bits which can be set independently. This gives the user unique flexibility in selecting different levels of protection.

3.19.1 Boot Loader Features

- Read-while-write self-programming
- Flexible boot memory size
- High security (separate boot lock bits for a flexible protection)
- Separate fuse to select reset vector
- Optimized page (1) size
- Code efficient algorithm
- Efficient read-modify-write support

Note: 1. A page is a section in the Flash consisting of several bytes (see Table 3-74 on page 139) used during programming. The page organization does not affect normal operation.

3.19.2 Application and Boot Loader Flash Sections

The Flash memory is organized in two main sections—the application section and the boot loader section (see Figure 3-81 on page 128). The size of the different sections is configured by the BOOTSZ fuses as shown in Table 3-66 on page 135. These two sections can have different level of protection because they have different sets of lock bits.

3.19.2.1 Application Section

The application section is the section of the Flash that is used for storing the application code. The protection level for the application section can be selected by the application boot lock bits (boot lock bits 0) (see Table 3-62 on page 129). The application section can never store any boot loader code because the Store Program Memory (SPM) instruction is disabled when executed from the application section.

3.19.2.2 BLS – Boot Loader Section

While the application section is used for storing the application code, the boot loader software must be located in the Boot Loader Section (BLS) because the SPM instruction can initiate a programming when executing from the BLS only. The SPM instruction can access the entire Flash, including the BLS itself. The protection level for the boot loader section can be selected by the boot loader lock bits (boot lock bits 1) (see Table 3-63 on page 129).

3.19.2.3 Read-While-Write and No Read-While-Write Flash Sections

Whether the CPU supports read-while-write or the CPU is halted during a boot loader software update depends which address it is being programmed. In addition to the two sections that are configurable by the BOOTSZ fuses as described above, the Flash is also divided into two fixed sections, the Read-While-Write (RWW) section and the No Read-While-Write (NRWW) section. The limit between the RWW and NRWW sections is indicated in Figure 3-80 and Figure 3-81 on page 128. The main difference between the two sections is:

1. When erasing or writing a page located inside the RWW section, the NRWW section can be read during the operation.
2. When erasing or writing a page located inside the NRWW section, the CPU is halted during the entire operation.

Note that the user software can never read any code that is located inside the RWW section during a boot loader software operation. The syntax “RWW – Read-While-Write Section” refers to the particular section being programmed (erased or written), not the section actually being read during a boot loader software update.

3.19.2.4 RWW – Read-While-Write Section

If a boot loader software update is programming a page inside the RWW section, it is possible to read the code from the Flash, but only the code that is located in the NRWW section. During on-going programming, the software must ensure that the RWW section is never being read. If the user software is trying to read the code located inside the RWW section (i.e., by a call/jmp/lpm or an interrupt) during programming, the software might end up in an unknown state. To avoid this, the interrupts should either be disabled or moved to the boot loader section. The boot loader section is always located in the NRWW section. The RWW Section Busy (RWWSB) bit in the Store Program Memory Control and Status Register (SPMCSR) is read as logic one as long as the RWW section is blocked for reading. After programming is completed, the RWWSB must be cleared by software before reading the code located in the RWW section. For more information on how to clear RWWSB, see Section 3.19.5 “Store Program Memory Control and Status Register – SPMCSR” on page 136.

3.19.2.5 NRWW – No Read-While-Write Section

The code located in the NRWW section can be read when the boot loader software is updating a page in the RWW section. When the boot loader code updates the NRWW section, the CPU is halted during the entire page erase or page write operation.

Table 3-61. Read-While-Write Features

Which Section Does the Z Pointer Address During Programming?	Which Section Can be Read During Programming?	Is the CPU Halted?	Read-While-Write Supported?
RWW section	NRWW section	No	Yes
NRWW section	None	Yes	No

Figure 3-80. Read-While-Write versus No Read-While-Write

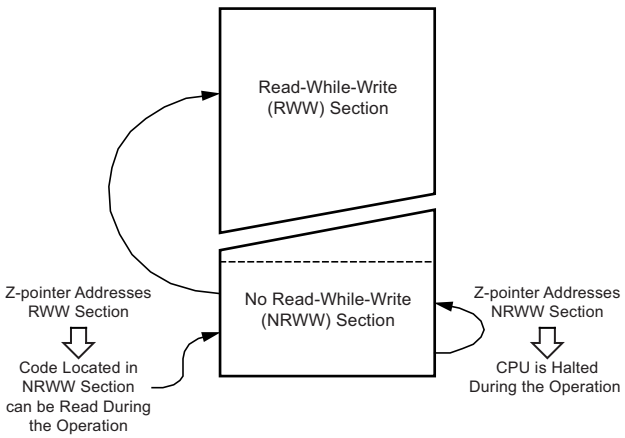
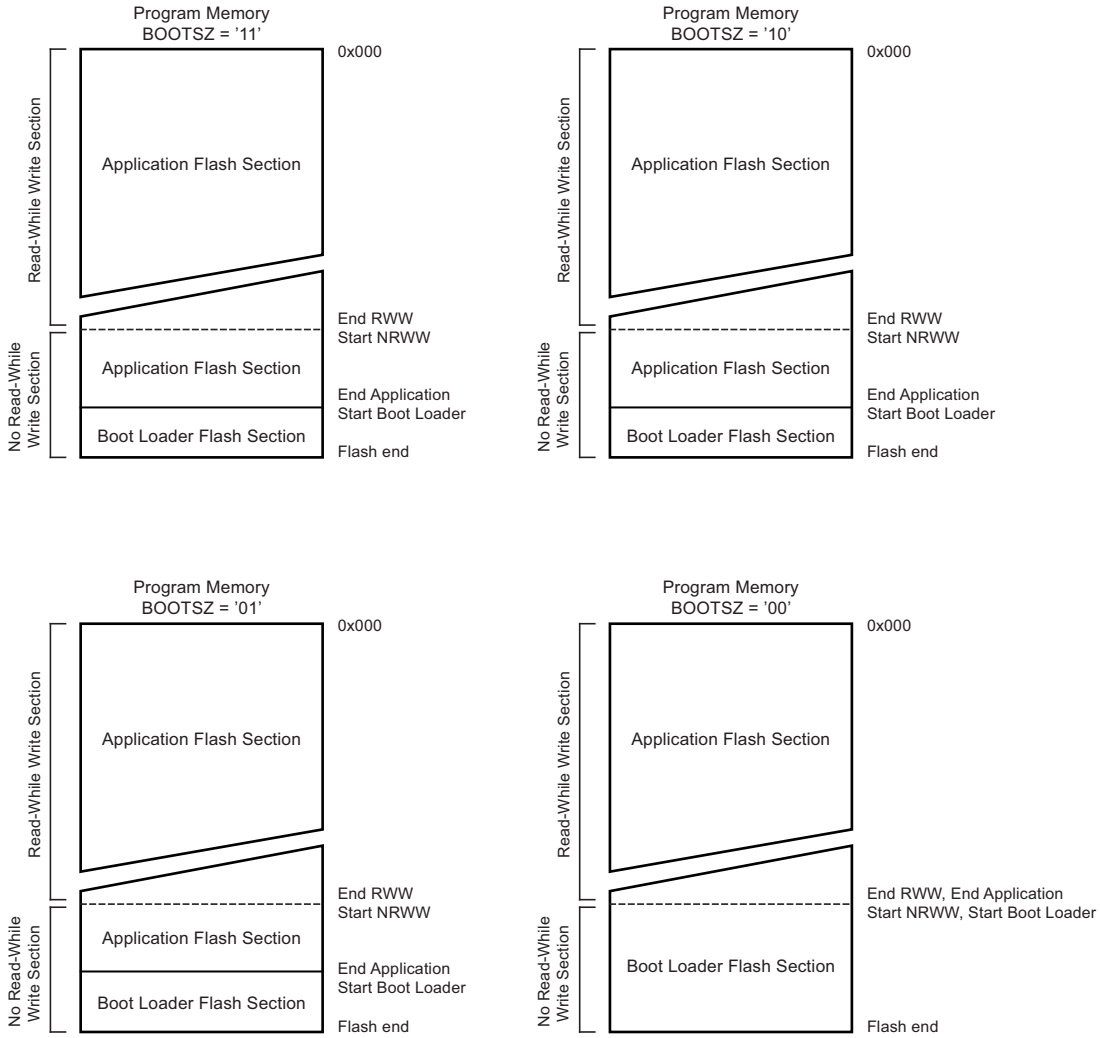


Figure 3-81. Memory Sections(1)



Note: 1. The parameters in Figure 3-81 are indicated in Table 3-66 on page 135.

3.19.2.6 Boot Loader Lock Bits

If no boot loader capability is needed, the entire Flash is available for the application code. The boot loader has two separate sets of boot lock bits which can be set independently. This gives the user unique flexibility in selecting different levels of protection.

The user can select:

- To protect the entire Flash from a software update by the MCU
- To protect only the boot loader Flash section from a software update by the MCU
- To protect only the application Flash section from a software update by the MCU
- Allow software update in the entire Flash

See Table 3-62 and Table 3-63 for further details. The boot lock bits can be set in software and in serial or parallel programming mode, but they can be cleared by a chip erase command only. The general write lock (lock bit mode 2) does not control the programming of the Flash memory by SPM instruction. Similarly, the general read/write lock (lock bit mode 1) controls neither reading nor writing by LPM/SPM if it is attempted.

Table 3-62. Boot Lock Bit 0 Protection Modes (Application Section)(1)

BLB0 Mode	BLB02	BLB01	Protection
1	1	1	No restrictions for SPM or LPM accessing the application section.
2	1	0	The SPM is not allowed to write to the application section.
3	0	0	The SPM is not allowed to write to the application section, and the LPM executing from the boot loader section is not allowed to read from the application section. If interrupt vectors are placed in the boot loader section, interrupts are disabled while executing from the application section.
4	0	1	The LPM executing from the boot loader section is not allowed to read from the application section. If interrupt vectors are placed in the boot loader section, interrupts are disabled while executing from the application section.

Note: 1. “1” means unprogrammed, “0” means programmed

Table 3-63. Boot Lock Bit 1 Protection Modes (Boot Loader Section)(1)

BLB1 Mode	BLB12	BLB11	Protection
1	1	1	No restrictions for SPM or LPM accessing the boot loader section.
2	1	0	The SPM is not allowed to write to the boot loader section.
3	0	0	The SPM is not allowed to write to the boot loader section, and the LPM executing from the application section is not allowed to read from the boot loader section. If interrupt vectors are placed in the application section, interrupts are disabled while executing from the boot loader section.
4	0	1	The LPM executing from the application section is not allowed to read from the boot loader section. If interrupt vectors are placed in the application section, interrupts are disabled while executing from the boot loader section.

Note: 1. “1” means unprogrammed, “0” means programmed

3.19.2.7 Entering the Boot Loader Program

Entering the boot loader takes place by a jump or call from the application program. This may be initiated by a trigger such as a command received via SPI interface. Alternatively, the boot reset fuse can be programmed so that the reset vector is pointing to the boot Flash start address after reset. In this case, the boot loader is started after reset. After the application code is loaded, the program can start executing the application code. Note that the fuses cannot be changed by the MCU itself. This means that once the boot reset fuse is programmed, the reset vector always points to the boot loader reset and the fuse can only be changed through the serial or parallel programming interface.

Table 3-64. Boot Reset Fuse(1)

BOOTRST	Reset Address
1	Reset Vector = Application Reset (address 0x0000)
0	Reset Vector = Boot Loader Reset (see Table 3-66 on page 135)

Note: 1. “1” means unprogrammed, “0” means programmed

3.19.3 Addressing the Flash During Self- Programming

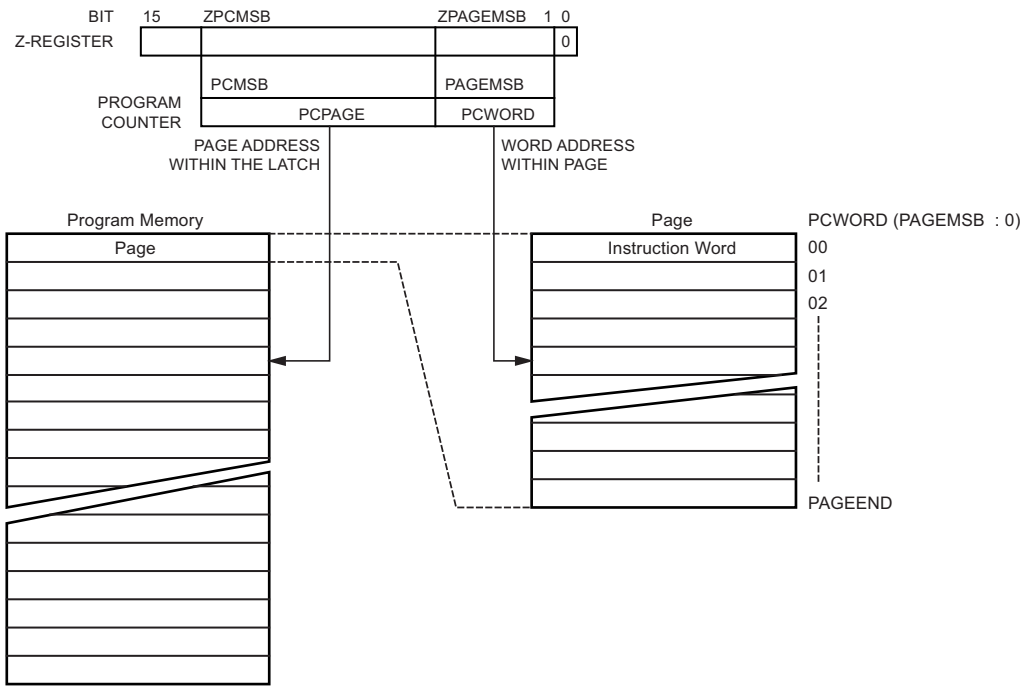
The Z pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	**Z15**	**Z14**	**Z13**	**Z12**	**Z11**	**Z10**	**Z9**	**Z8**
ZL (R30)	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
Bit	7	6	5	4	3	2	1	0

Because the Flash is organized in pages (see Table 3-74 on page 139), the program counter can be treated as having two different sections. One section, consisting of the last significant bits, addresses the words within a page, while the most significant bits address the pages. This is shown in Figure 3-82 on page 130. Note that the page erase and page write operations are addressed independently. Therefore it is of major importance that the boot loader software addresses the same page in both the page erase and page write operation. Once a programming operation is initiated, the address is latched and the Z pointer can be used for other operations.

The only SPM operation that does not use the Z pointer is the setting of the boot loader lock bits. The content of the Z pointer is ignored and has no effect on the operation. The LPM instruction also uses the Z pointer to store the address. Because this instruction addresses the Flash byte-by-byte, the LSB (bit Z0) of the Z pointer is also used.

Figure 3-82. Addressing the Flash During SPM(1)



Note: 1. The different variables used in Figure 3-82 are listed in Table 3-68 on page 135.

3.19.4 Self- Programming the Flash

The program memory is updated in a page-by-page fashion. Before programming a page with the data stored in the temporary page buffer, the page must be erased. The temporary page buffer is filled one word at a time using SPM and the buffer can be filled either before the page erase command or between a page erase and a page write operation:

Alternative 1: fill the buffer before a page erase:

- Fill the temporary page buffer
- Perform a page erase
- Perform a page write

Alternative 2: fill the buffer after page erase:

- Perform a page erase
- Fill temporary page buffer
- Perform a page write

If only a part of the page needs to be changed, the rest of the page must be stored (for example, in the temporary page buffer) before the erase, and then be rewritten. When using alternative 1, the boot loader provides an effective read-modify-write feature which allows the user software to first read the page, do the necessary changes, and then write back the modified data. If alternative 2 is used, it is not possible to read the old data while loading because the page has already been erased. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the page erase and page write operation addresses the same page. See Section 3.19.4.12 “Simple Assembly Code Example for a Boot Loader” on page 133 for an assembly code example.

3.19.4.1 Performing Page Erase by SPM

To execute page erase, set up the address in the Z pointer, write "X0000011" to the SPMCSR and execute SPM within four clock cycles after writing to the SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE in the Z register. Other bits in the Z pointer are ignored during this operation.

Page erase to the RWW section: The NRWW section can be read during the page erase.

Page erase to the NRWW section: The CPU is halted during this operation.

3.19.4.2 Filling the Temporary Buffer (Page Loading)

To write an instruction word, set up the address in the Z pointer and data in R1:R0, write “00000001” to the SPMCSR and execute SPM within four clock cycles after writing to the SPMCSR. The content of PCWORD in the Z register is used to address the data in the temporary buffer. The temporary buffer is auto-erased after a page write operation or by writing the RWWSRE bit in the SPMCSR. It is also erased after a system reset. Note that it is not possible to write more than one time to each address without erasing the temporary buffer.

If the EEPROM is written in the middle of an SPM page load operation, all data loaded is lost.

3.19.4.3 Performing a Page Write

To execute page write, set up the address in the Z-pointer, write “X0000101” to the SPMCSR and execute SPM within four clock cycles after writing to the SPMCSR. The data in R1 and R0 is ignored. The page address must be written to the PCPAGE. Other bits in the Z pointer must be written to zero during this operation.

Page write to the RWW section: The NRWW section can be read during the page write.

Page write to the NRWW section: The CPU is halted during this operation.

3.19.4.4 Using the SPM Interrupt

If the SPM interrupt is enabled, the SPM interrupt generates a constant interrupt when the SELFPRGEN bit in SPMCSR is cleared. This means that the interrupt can be used instead of polling the SPMCSR register in software. When using the SPM interrupt, the interrupt vectors should be moved to the BLS section to avoid that an interrupt is accessing the RWW section when it is blocked for reading. How to move the interrupts is described in Section 3.10 “Interrupts” on page 39.

3.19.4.5 Consideration While Updating BLS

Special care must be taken if the user allows the boot loader section to be updated by leaving boot lock bit11 unprogrammed. An accidental write to the boot loader itself can corrupt the entire boot loader and further software updates might be impossible. If it is not necessary to change the boot loader software itself, it is recommended to program the boot lock bit11 to protect the boot loader software from any internal software changes.

3.19.4.6 Prevent Reading the RWW Section During Self-Programming

During self-programming (either page erase or page write), the RWW section is always blocked for reading. The user software itself must prevent this section from being addressed during the self-programming operation. The RWWSB in the SPMCSR is set as long as the RWW section is busy. During self-programming the interrupt vector table should be moved to the BLS as described in Section 3.10 “Interrupts” on page 39, or the interrupts must be disabled. Before addressing the RWW section after the programming has been completed, the user software must clear the RWWSB by writing the RWWSRE bit. See Section 3.19.4.12 “Simple Assembly Code Example for a Boot Loader” on page 133 for an example.

3.19.4.7 Setting the Boot Loader Lock Bits by SPM

To set the boot loader lock bits and general lock bits, write the desired data to R0, write “X0001001” to the SPMCSR and execute the SPM within four clock cycles after writing to the SPMCSR.

Bit	7	6	5	4	3	2	1	0
R0	**1**	**1**	**BLB12**	**BLB11**	**BLB02**	**BLB01**	**1**	**1**

See Table 3-62 and Table 3-63 on page 129 for how the different settings of the boot loader bits affect the Flash access. If bits 5..2 in R0 are cleared (zero), the corresponding boot lock bit is programmed if an SPM instruction is executed within four cycles after BLBSET and SELFPRGEN have been set in the SPMCSR. The Z pointer will be ignored during this operation, but for future compatibility it is recommended to load the Z pointer with 0x0001 (same as used for reading the lock bits). For future compatibility it is also recommended to set bits 7, 6, 1, and 0 in R0 to “1” when writing the lock bits. When programming the lock bits, the entire Flash can be read during the operation.

3.19.4.8 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation blocks all software programming to Flash. Reading the fuses and lock bits from software is also prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEWE) in the EECR register and verifies that the bit is cleared before writing to the SPMCSR Register.

3.19.4.9 Reading the Fuse and Lock Bits from Software

It is possible to read both the fuse and lock bits from software. To read the lock bits, load the Z pointer with 0x0001 and set the BLBSET and SELFPRGEN bits in the SPMCSR. When an LPM instruction is executed within three CPU cycles after the BLBSET and SELFPRGEN bits have been set in the SPMCSR, the value of the lock bits is loaded into the destination register. The BLBSET and SELFPRGEN bits auto-clear after reading the lock bits has been completed or if no LPM instruction is executed within three CPU cycles or if no SPM instruction is executed within four CPU cycles. When BLBSET and SELFPRGEN are cleared, LPM works as described in the instruction set manual.

Bit	7	6	5	4	3	2	1	0
Rd	**-**	**-**	**BLB12**	**BLB11**	**BLB02**	**BLB01**	**LB2**	**LB1**

The algorithm for reading the fuse low byte is similar to the one described above for reading the lock bits. To read the fuse low byte, load the Z pointer with 0x0000 and set the BLBSET and SELFPRGEN bits in the SPMCSR. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits have been set in the SPMCSR, the value of the fuse low byte (FLB) is loaded into the destination register as shown below. Refer to Table 3-72 on page 138 for a detailed description and mapping of the fuse low byte.

Fuse low bytes:

Bit	7	6	5	4	3	2	1	0
Rd	**FLB7**	**FLB6**	**FLB5**	**FLB4**	**FLB3**	**FLB2**	**FLB1**	**FLB0**

Similarly, when reading the fuse high byte, load 0x0003 in the Z pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits have been set in the SPMCSR, the value of the Fuse High Byte (FHB) is loaded into the destination register as shown below. Refer to Table 3-71 on page 138 for a detailed description and mapping of the fuse high byte.

Fuse high bytes:

Bit	7	6	5	4	3	2	1	0
Rd	**FHB7**	**FHB6**	**FHB5**	**FHB4**	**FHB3**	**FHB2**	**FHB1**	**FHB0**

Fuse and lock bits that are programmed are read as zero. Fuse and lock bits that are unprogrammed are read as one.

3.19.4.10 Preventing Flash Corruption

During periods of low VCC, the Flash program can be corrupted because the supply voltage is too low for the CPU and the Flash to operate properly. These issues are the same as for board level systems using Flash, and the same design solutions should be applied.

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to Flash requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. If there is no need for a boot loader update in the system, program the boot loader lock bits to prevent any boot loader software updates.
2. Keep the Atmel® AVR® $\overline{\text{RESET}}$ pin active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation is completed provided that the power supply voltage is sufficient.
3. Keep the AVR core in power-down sleep mode during periods of low VCC. This prevents the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR register and thus the Flash from unintentional writes.

3.19.4.11 Programming Time for Flash when Using SPM

The calibrated RC oscillator is used to time Flash accesses. Table 3-65 on page 133 shows the typical programming time for Flash accesses from the CPU.

Table 3-65. SPM Programming Time(1)

Symbol	Min. Programming Time	Max. Programming Time
Flash write (page erase, page write and write lock bits by SPM)	3.7ms	4.5ms

Note: 1. Minimum and maximum programming time is per individual operation

3.19.4.12 Simple Assembly Code Example for a Boot Loader

```
;-the routine writes one page of data from RAM to Flash
; the first data location in RAM is pointed to by the Y pointer
; the first data location in Flash is pointed to by the Z-pointer
;-error handling is not included
;-the routine must be placed inside the Boot space
; (at least the Do_spm sub routine). Only code inside NRWW section can
; be read during Self-Programming (Page Erase and Page Write).
;-registers used: r0, r1, temp1 (r16), temp2 (r17), looplo (r24),
; loophi (r25), spmcrval (r20)
; storing and restoring of registers is not included in the routine
; register usage can be optimized at the expense of code size
;-It is assumed that either the interrupt table is moved to the Boot
; loader section or that the interrupts are disabled.
.equ PAGESIZEB = PAGESIZE*2;PAGESIZEB is page size in BYTES, not words
.org SMALLBOOTSTART
Write_page:
; Page Erase
ldi spmcrval, (1<<PGERS) | (1<<SELFPRGEN)
call Do_spm
; re-enable the RWW section
ldi spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
call Do_spm
; transfer data from RAM to Flash page buffer
ldi looplo, low(PAGESIZEB) ;init loop variable
ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
Wrloop:
ld r0, Y+
```

```
ld r1, Y+
ldi spmcrval, (1<<SELFPRGEN)
call Do_spm
adiw ZH:ZL, 2
sbiw loophi:looplo, 2 ;use subi for PAGESIZEB<=256
brne Wrloop
; execute Page Write
subi ZL, low(PAGESIZEB) ;restore pointer
sbci ZH, high(PAGESIZEB) ;not required for PAGESIZEB<=256
ldi spmcrval, (1<<PGWRT) | (1<<SELFPRGEN)
call Do_spm
; re-enable the RWW section
ldi spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
call Do_spm
; read back and check, optional
ldi looplo, low(PAGESIZEB);init loop variable
ldi loophi, high(PAGESIZEB);not required for PAGESIZEB<=256
subi YL, low(PAGESIZEB);restore pointer
sbci YH, high(PAGESIZEB)
Rdloop:
lpm r0, Z+
ld r1, Y+
cpse r0, r1
jmp Error
sbiw loophi:looplo, 1;use subi for PAGESIZEB<=256
brne Rdloop
; return to RWW section
; verify that RWW section is safe to read
Return:
in temp1, SPMCSR
sbrs temp1, RWWSB; If RWWSB is set, the RWW section is not ready yet
ret
; re-enable the RWW section
ldi spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
call Do_spm
rjmp Return
Do_spm:
; check for previous SPM complete
Wait_spm:
in temp1, SPMCSR
sbrc temp1, SELFPRGEN
rjmp Wait_spm
; input: spmcrval determines SPM action
; disable interrupts if enabled, store status
in temp2, SREG
cli
; check that no EEPROM write access is present
Wait_ee:
sbic EECR, EEWE
rjmp Wait_ee
; SPM timed sequence
out SPMCSR, spmcrval
spm
; restore SREG (to enable interrupts if originally enabled)
out SREG, temp2
ret
```

3.19.4.13 Boot Loader Parameters

Parameters used in the description of the self-programming are indicated in Table 3-66 through Table 3-67.

Table 3-66. Boot Size Configuration

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	128 words	4	0x000 – 0xF7F	0xF80 – 0xFFF	0xF7F	0xF80
1	0	256 words	8	0x000 – 0xEFF	0xF00 – 0xFFF	0xEFF	0xF00
0	1	512 words	16	0x000 – 0xDFF	0xE00 – 0xFFF	0xDFF	0xE00
0	0	1024 words	32	0x000 – 0xBFF	0xC00 – 0xFFF	0xBFF	0xC00

Note: See Figure 3-81 on page 128 for different BOOTSZ fuse configurations.

Table 3-67. Read-While-Write Limit

Section	Pages	Address
Read-While-Write section (RWW)	96	0x000 – 0xBFF
No Read-While-Write section (NRWW)	32	0xC00 – 0xFFF

For more information about these two sections, see Section 3.19.2.5 “NRWW – No Read-While-Write Section” on page 127 and Section 3.19.2.4 “RWW – Read-While-Write Section” on page 127.

Table 3-68. Explanation of Different Variables Used in Figure 3-82 on page 130 and the Mapping to Z-pointer

Variable		Corresponding Z Value[1]	Description
PCMSB	11		Most significant bit in the program counter. (The program counter is 12 bits PC[11:0]).
PAGEMSB	4		Most significant bit which is used to address the words within one page (32 words in a page requires 5 bits PC [4:0]).
ZPCMSB		Z12	Bit in Z register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z5	Bit in Z register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.
PCPAGE	PC[11:5]	Z12:Z6	Program counter page address: page select, for page erase and page write.
PCWORD	PC[4:0]	Z5:Z1	Program counter word address: word select, for filling temporary buffer (must be zero during page write operation).

Note: 1. Z15:Z13: always ignored. Z0: should be zero for all SPM commands, byte select for the LPM instruction. See Section 3.19.3 “Addressing the Flash During Self- Programming” on page 130 for more information about the use of the Z pointer during self-programming.

3.19.5 Store Program Memory Control and Status Register – SPMCSR

The store program memory control and status register contain the control bits needed to control the boot loader operations.

3.19.5.1 Store Program Memory Control and Status Register – SPMCSR

Bit	7	6	5	4	3	2	1	0	
	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SELFPRGEN	SPMCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7 – SPMIE: SPM Interrupt Enable Bit
When the SPMIE bit is written to one, and the I bit in the status register is set (one), the SPM ready interrupt is enabled. The SPM ready interrupt is executed as long as the SELFPRGEN bit in the SPMCSR register is cleared.

Bits 6 – RWWSB: Read-While-Write Section Busy Bit
When a self-programming (page erase or page write) operation to the RWW section is initiated, the RWWSB is set (one) by hardware. When the RWWSB bit is set, the RWW section cannot be accessed. The RWWSB bit is cleared if the RWWSRE bit is written to one after a self-programming operation has been completed. Alternatively, the RWWSB bit is automatically cleared if a page load operation is initiated.

Bit 5 – Res: Reserved Bit
This bit is a reserved bit on the ATA6289 and is always read as zero.

Bit 4 – RWWSRE: Read-While-Write Section Read Enable Bit
When programming (page erase or page write) to the RWW section, the RWW section is blocked for reading (the RWWSB is set by hardware). To re-enable the RWW section, the user software must wait until the programming is completed (the SELFPRGEN is cleared). Then, if the RWWSRE bit is written to one at the same time as the SELFPRGEN, the next SPM instruction within four clock cycles re-enables the RWW section. The RWW section cannot be re-enabled while the Flash is busy with a page erase or a page write (SELFPRGEN is set). If the RWWSRE bit is written while the Flash is being loaded, the Flash load operation aborts and the data loaded is lost.

Bit 3 – BLBSET: Boot Lock Bit Set
If this bit is written to one at the same time as the SELFPRGEN, the next SPM instruction within four clock cycles sets boot lock bits and memory lock bits, according to the data in R0. The data in R1 and the address in the Z pointer are ignored. The BLBSET bit is automatically cleared upon completion of the lock bit set, or if no SPM instruction is executed within four clock cycles.

Within three cycles after BLBSET and SELFPRGEN have been set in the SPMCSR register, an LPM instruction reads either the lock bits or the fuse bits (depending on Z0 in the Z pointer) into the destination register. See Section 3.21.3.11 “Reading the Fuse and Lock Bits” on page 146 for more information.

Bit 2 – PGWRT: Page Write Bit
If this bit is written to one at the same time as the SELFPRGEN, the next SPM instruction executes page write within four clock cycles with the data stored in the temporary buffer. The page address is taken from the high part of the Z pointer. The data in R1 and R0 are ignored. The PGWRT bit auto clears upon completion of a page write or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire page write operation if the NRWW section is addressed.

Bit 1 – PGERS: Page Erase Bit
If this bit is written to one at the same time as the SELFPRGEN, the next SPM instruction executes page erase within four clock cycles. The page address is taken from the high part of the Z pointer. The data in R1 and R0 are ignored. The PGERS bit auto clears upon completion of a page erase or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire page write operation if the NRWW section is addressed.

Bit 0 – SELFPRGEN: Self-Programming Bit
This bit enables the SPM instruction for the next four clock cycles. If written to one together with either RWWSRE, BLBSET, PGWRT or PGERS, the following SPM instruction has a special meaning, as described above. If only the SELFPRGEN is written, the following SPM instruction stores the value in R1:R0 in the temporary page buffer addressed by the Z pointer. The LSB of the Z pointer is ignored. The SELFPRGEN bit auto clears upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During page erase and page write, the SELFPRGEN bit remains high until the operation is completed.

Writing any other combination than “10001,” “01001,” “00101,” “00011,” or “00001” in the lower five bits has no effect.

3.20 Memory Programming

3.20.1 Program and Data Memory Lock Bits

The ATA6289 provides six lock bits which can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 3-70 on page 137. The lock bits can only be erased to “1” with the chip erase command. The SPM instruction is enabled for the whole Flash if the SELFPRGEN fuse is programmed (“0”), otherwise it is disabled.

Table 3-69. Lock Bit Byte(1)

Lock Bit Byte	Bit No.	Description	Default value
	7	-	1 (unprogrammed)
	6	-	1 (unprogrammed)
BLB12	5	Boot lock bit	1 (unprogrammed)
BLB11	4	Boot lock bit	1 (unprogrammed)
BLB02	3	Boot lock bit	1 (unprogrammed)
BLB01	2	Boot lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Note: 1. “1” means unprogrammed, “0” means programmed

Table 3-70. Lock Bit Protection Modes(1)(2)

Memory Lock Bits			Protection Type
LB Mode	LB2	LB1	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in parallel and serial programming mode. The fuse bits are locked in both serial and parallel programming mode.(1)
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in parallel and serial programming mode. The boot lock bits and fuse bits are locked in both serial and parallel programming mode.(1)
BLB0 Mode	BLB02	BLB01	
1	1	1	No restrictions for SPM or LPM accessing the application section.
2	1	0	The SPM is not allowed to write to the application section.
3	0	0	The SPM is not allowed to write to the application section, and the LPM executing from the boot loader section is not allowed to read from the application section. If interrupt vectors are placed in the boot loader section, interrupts are disabled while executing from the application section.
4	0	1	The LPM executing from the boot loader section is not allowed to read from the application section. If interrupt vectors are placed in the boot loader section, interrupts are disabled while executing from the application section.
BLB1 Mode	BLB12	BLB11	
1	1	1	No restrictions for SPM or LPM accessing the boot loader section.
2	1	0	The SPM is not allowed to write to the boot loader section.
3	0	0	The SPM is not allowed to write to the boot loader section, and the LPM executing from the application section is not allowed to read from the boot loader section. If interrupt vectors are placed in the application section, interrupts are disabled while executing from the boot loader section.
4	0	1	The LPM executing from the application section is not allowed to read from the boot loader section. If interrupt vectors are placed in the application section, interrupts are disabled while executing from the boot loader section.

Notes: 1. Program the fuse bits and boot lock bits before programming the LB1 and LB2.
2. “1” means unprogrammed, “0” means programmed

3.20.2 Fuse Bits

The ATA6289 has two fuse bytes. Table 3-71 and Table 3-72 describe briefly the functionality of all fuses and how they are mapped into the fuse bytes. Note that the fuses are read as logic zero, “0,” if they are programmed.

Table 3-71. Fuse High Byte

Name	Bit No.	Description	Default value
EELOCK	7	Upper EEPROM locked	1 (unprogrammed, unlocked)
DWEN	6	DebugWIRE enable	1 (unprogrammed, disabled)
SPIEN	5	SPI enable	0 (programmed, SPI programming enabled)
WDTON	4	Watchdog timer always on	1 (unprogrammed, disabled)
EESAVE	3	Keep EEPROM contents during chip erase	1 (unprogrammed, disabled)
BOOTSZ1	2	Boot size select	0 (programmed)(1)
BOOTSZ0	1	Boot size select	0 (programmed)(1)
BOOTRST	0	Select reset vector	1 (unprogrammed, enabled application program reset address vector 0x000)

Note: 1. The default value of BOOTSZ1..0 results in maximum boot size. See Table 3-66 on page 135 for more information.

Table 3-72. Fuse Low Byte

Name	Bit No	Description	Default value
CKDIV8	7	Start up with system clock divided by 8	0 (programmed, enabled)
CKOUT	6	Output internal clock on PC1(CLKO) pin	1 (unprogrammed, disabled)
SUT1	5	Select startup time	1 (unprogrammed)
SUT0	4	Select startup time	0 (programmed)
WDRCON	3	Enable watchdog RC oscillator (SRC)	0 (programmed, enabled)
FRCFS	2	Fast RC oscillator frequency select 1MHz or 4 MHz	0 (programmed, 4MHz enabled)
BODEN	1	Enable brown-out detection	0 (programmed, enabled)
TSRDI	0	Disable temperature shutdown reset	1 (unprogrammed, enabled)

Note: 1. The default value of BOOTSZ1..0 results in maximum boot size. See Table 3-66 on page 135 for more information.

3.20.2.1 Latching of Fuses

The fuse values are latched when the device enters the programming mode and changes of fuse values have no effect until the part leaves the programming mode. This does not apply to the EESAVE fuse which takes effect once it is programmed. The fuses are also latched on power-up in normal mode.

3.20.3 Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial and parallel mode, also when the device is locked. The three bytes reside in a separate address space.

Table 3-73. ATA6289 Signature Bytes Description

Signature Bytes Address	Value	Description
0x000	0x1E	Indicates manufactured by Atmel
0x001	0x93	Indicates 8KB Flash memory
0x002	0x82	Indicates the ATA6289

3.20.4 Calibration Bytes

The ATA6289 has 3-byte calibration values of the two internal RC oscillators (FRC and SRC) and the motion sensor. During reset, these bytes are automatically written into the FRCCAL and SRCCAL registers to ensure correct frequency of the calibrated RC oscillators. The MSVCAL register is also automatically written but with an uncalibrated value (motion sensor default value = 255).

3.20.5 Page Sizes

Table 3-74. Number of Words in a Page and Number of Pages in the Flash

Device	Flash Size	Page Size	PCWORD	No. of Pages	PCPAGE	PCMSB
ATA6289	4Kwords (8KB)	32words (64bytes)	PC[4:0]	128	PC[11:5]	11

Table 3-75. No. of Words in a Page and No. of Pages in the EEPROM

Device	EEPROM Size	Page Size	PCWORD	No. of Pages	PCPAGE	EEAMSB
ATA6289	256 + 64bytes	4bytes	EEA[1:0]	64 + 16	EEA[8:2]	8

3.21 Parallel Programming

This section describes how to parallel program and verify Flash program memory, EEPROM data memory, memory lock bits, and fuse bits in the ATA6289. Unless otherwise noted, pulses are assumed to be at least 1 µs.

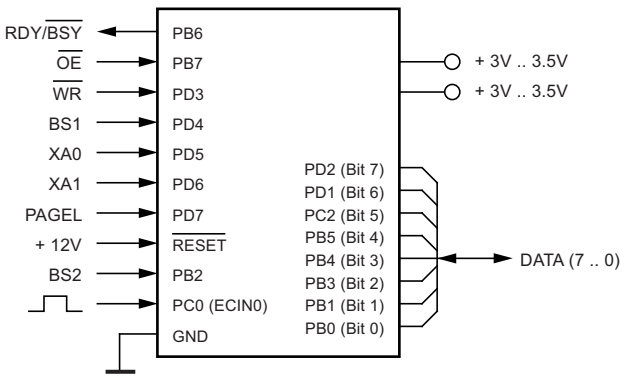
3.21.1 Signal Names

In this section, some pins of the ATA6289 are referenced by signal names describing their functionality during parallel programming (see Figure 3-83 and Table 3-76 on this page). Pins not described in the following tables are referenced by pin names.

The XA1/XA0 pins determine the action executed when the PC0 (ECIN0) pin is given a positive pulse. The bit coding is shown in Table 3-76 on this page.

When pulsing $\overline{WR}$ or $\overline{OE}$, the loaded command determines the action to be executed. The different commands are shown in Table 3-79 on page 140.

Figure 3-83. Parallel Programming



Note: $3.0V < V_{CC} < 3.5V$

3.21.2 Parallel Programming Pin Mapping

Table 3-76. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/$\overline{BSY}$	PB6	O	0: Device is busy programming 1: Device is ready for new command
OE	PB7	I	Output enable (active low)
WR	PD3	I	Write pulse (active low)
BS1	PD4	I	Byte select 1 (“0” selects low byte, “1” selects high byte)
XA0	PD5	I	XTAL action Bit 0
XA1	PD6	I	XTAL action Bit 1
PAGEL	PD7	I	Program memory and EEPROM data page load
BS2	PB2	I	Byte select 2 (“0” selects low byte, “1” selects 2nd high byte)
DATA	Data[7:0] {PD[2:1], PC2, PB[5:3], PB[1:0]}	I/O	Bi-directional databus (output when $\overline{OE}$ is low)

Table 3-77. Pin Values Used to Enter Programming Mode

Pin	Symbol	Value
PAGEL	Prog_enable[3]	0
XA1	Prog_enable[2]	0
XA0	Prog_enable[1]	0
BS2	Prog_enable[0]	0

Table 3-78. XA1 and XA0 Coding

XA1	XA0	Action When PC0(ECIN0) Is Pulsed
0	0	Load Flash or EEPROM address (high or low address byte determined by BS1).
0	1	Load data (high or low data byte for Flash determined by BS1).
1	0	Load command
1	1	No action, idle

Table 3-79. Command Byte Bit Coding

Command Byte	Command Executed
1000 0000	Chip erase
0100 0000	Write fuse bits
0010 0000	Write lock bits
0001 0000	Write flash
0001 0001	Write EEPROM
0000 1000	Read signature bytes and calibration byte
0000 0100	Read fuse and look bits
0000 0010	Read flash
0000 0011	Read EEPROM

3.21.3 Parallel Programming Flow

3.21.3.1 Enter Parallel Programming Mode

The following algorithm puts the device in parallel (high-voltage) programming mode:

1. Set Prog_enable pins listed in Table 3-76 on page 140 to “0000,” $\overline{\text{RESET}}$ pin to 0V and VCC to 0V.
2. Apply 3.0V - 3.5V between V_{CC} and GND. Ensure that VCC reaches at least 1.8V within the next 20µs.
3. Wait 20µs- 60µs, and apply 11.5V - 12.5V to $\overline{\text{RESET}}$ pin.
4. Keep the Prog_enable pins unchanged for at least 10µs after the high voltage has been applied to ensure the Prog_enable signature has been latched.
5. Wait at least 300µs before giving any parallel programming commands.
6. Exit programming mode by powering down the device or by bringing $\overline{\text{RESET}}$ pin to 0V.

If the rise time of the V_{CC} is unable to fulfill the requirements listed above, the following alternative algorithm can be used.

1. Set Prog_enable pins listed in Table 3-76 on page 140 to “0000,” $\overline{\text{RESET}}$ pin to 0V and V_{CC} to 0V.
2. Apply 3.0V - 3.5V between V_{CC} and GND.
3. Monitor VCC and as soon as V_{CC} reaches 0.9V - 1.1V, apply 11.5V - 12.5V to $\overline{\text{RESET}}$ pin.
4. Keep the Prog_enable pins unchanged for at least 10µs after the high voltage has been applied to ensure the Prog_enable signature has been latched.
5. Wait until V_{CC} actually reaches 3.0V - 3.5V before giving any parallel programming commands.
6. Exit programming mode by powering down the device or by bringing $\overline{\text{RESET}}$ pin to 0V.

3.21.3.2 Consideration for Efficient Programming

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command only needs to be loaded once when writing or reading multiple memory locations.
- Skip writing the data value 0xFF that is the contents of the entire EEPROM (unless the EESAVE fuse is programmed) and Flash after a chip erase.
- Address high byte only needs to be loaded before programming or reading a new 256 word window in Flash or 256byte EEPROM. This consideration also applies to signature bytes reading.

3.21.3.3 Chip Erase

Chip erase erases Flash and EEPROM(1) memories including lock bits. The lock bits are not reset until the program memory has been completely erased. The fuse bits are not changed. A chip erase must be performed before the Flash and/or EEPROM are reprogrammed.

Note: 1. The EEPRPOM memory is preserved during chip erase if the EESAVE fuse is programmed.

Load “Chip Erase” command

1. Set XA1, XA0 to “10.” This enables command loading.
2. Set BS1 to “0.”
3. Set DATA to “1000 0000.” This is the command for chip erase.
4. Give PC0 (ECIN0) a positive pulse. This loads the command.
5. Give $\overline{\text{WR}}$ a negative pulse. This starts the chip erase. RDY/$\overline{\text{BSY}}$ goes low.
6. Wait until RDY/$\overline{\text{BSY}}$ goes high before loading a new command.

3.21.3.4 Programming the Flash

The Flash is organized in pages (see Table 3-74 on page 139). When programming the Flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory:

A: Load “Write Flash” command

1. Set XA1, XA0 to “10.” This enables command loading.
2. Set BS1 to “0.”
3. Set DATA to “0001 0000.” This is the command for write Flash.
4. Give PC0 (ECIN0) a positive pulse. This loads the command.

B: Load “Low Byte” address

1. Set XA1, XA0 to “00.” This enables address loading.
2. Set BS1 to “0.” This selects low address.
3. Set DATA = address low byte (0x00 - 0xFF).
4. Give PC0 (ECIN0) a positive pulse. This loads the address low byte.

C: Load “Low Byte” data

1. Set XA1, XA0 to “01.” This enables data loading.
2. Set DATA = data low byte (0x00 - 0xFF).
3. Give PC0 (ECIN0) a positive pulse. This loads the data byte.

D: Load “High Byte” data

1. Set BS1 to “1.” This selects high data byte.
2. Set XA1, XA0 to “01.” This enables data loading.
3. Set DATA = data high byte (0x00 - 0xFF).
4. Give PC0 (ECIN0) a positive pulse. This loads the data byte.

E: Latch data

1. Set BS1 to “1.” This selects high data byte.
2. Give PAGEL a positive pulse. This latches the data bytes. (See Figure 3-85 on page 143 for signal waveforms).

F: Repeat B through E until the entire buffer is filled or until all data within the page is loaded. The lower bits are mapped to words within the page in the address, the higher bits address the pages within the FLASH. This is illustrated in Figure 3-84 on page 143. Note that if less than eight bits are required to address words in the page (page size < 256), the most significant bit(s) in the address low byte are used to address the page when performing a page write (see also Table 3-74 on page 139).

G: Load “High Byte” Address

1. Set XA1, XA0 to “00.” This enables address loading.
2. Set BS1 to “1.” This selects high address.
3. Set DATA = address high byte (0x00 - 0xFF).
4. Give PC0 (ECIN0) a positive pulse. This loads the address high byte.

H: Program page

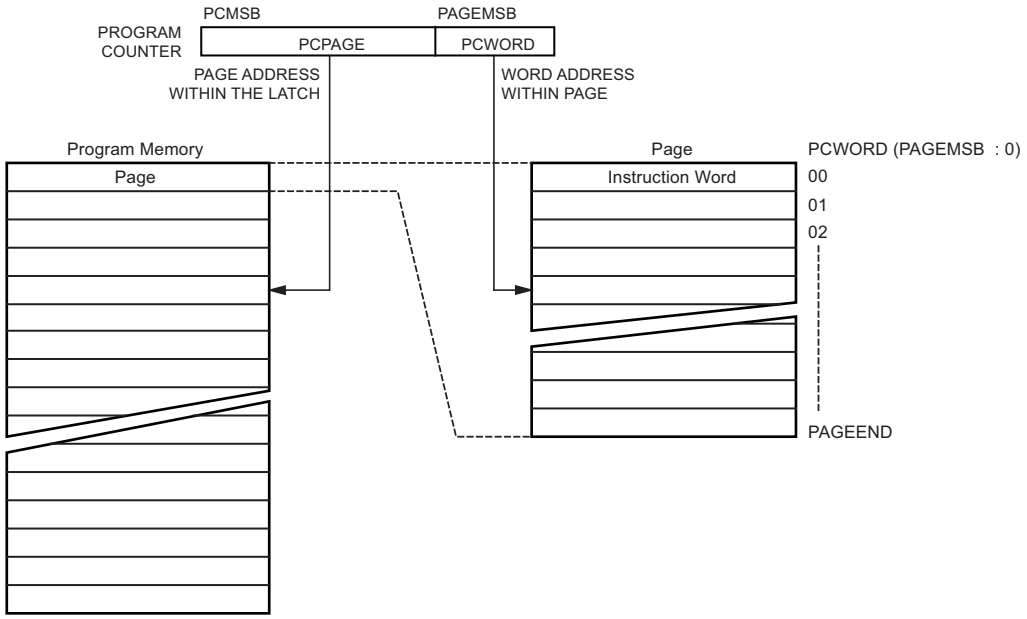
1. Give $\overline{WR}$ a negative pulse. This starts the programming of the entire page of data. RDY/$\overline{BSY}$ goes low.
2. Wait until RDY/$\overline{BSY}$ goes high (See Figure 3-85 on page 143 for signal waveforms).

I: Repeat B through H until the entire Flash or all data is programmed.

J: End page programming

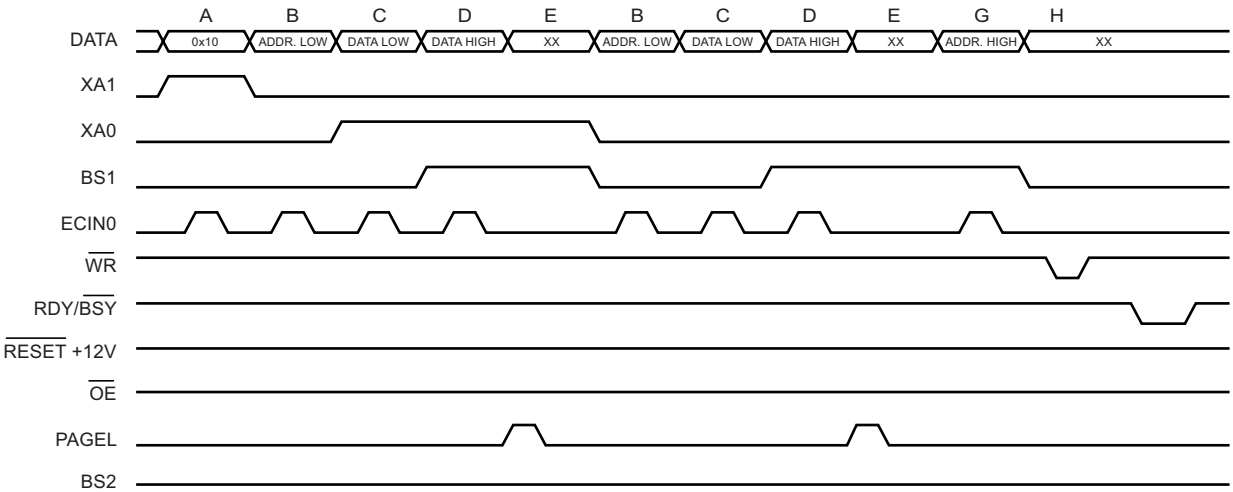
1. Set XA1, XA0 to “10.” This enables command loading.
2. Set DATA to “0000 0000.” This is the command for no operation.
3. Give PC0 (ECIN0) a positive pulse. This loads the command and the internal write signals are reset.

Figure 3-84. Addressing the Flash Which is Organized in Pages(1)



Note: 1. PCPAGE and PCWORD are listed in Table 3-74 on page 139.

Figure 3-85. Programming the Flash Waveforms(1)



Note: 1. “XX” is don't care. The letters refer to the programming description above.

3.21.3.5 Programming the EEPROM

The EEPROM is organized in pages (see Table 3-75 on page 139). When programming the EEPROM, the program data is latched into a page buffer. This allows one page of data to be programmed simultaneously. The programming algorithm for the EEPROM data memory is as follows (see Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command, address and data loading):

A: Load “0001 0001” command.

G: Load “High Byte” address (0x00 - 0xFF).

B: Load “Low Byte” address (0x00 - 0xFF).

C: Load data (0x00 - 0xFF).

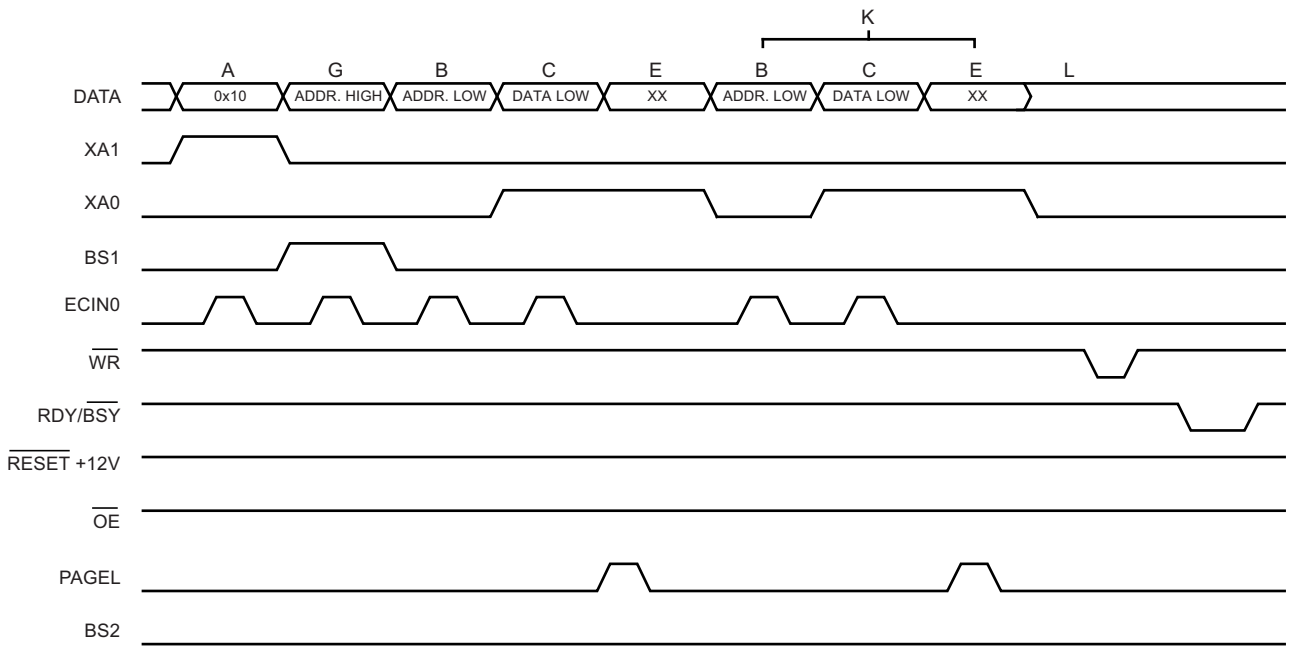
E: Latch data (give PAGEL a positive pulse).

K: Repeat B through E until the entire buffer is filled.

L: Program the EEPROM page.

1. Set BS1 to “0.”
2. Give $\overline{WR}$ a negative pulse. This starts programming of the EEPROM page. RDY/$\overline{BSY}$ goes low.
3. Wait until to RDY/$\overline{BSY}$ goes high before programming the next page (See Figure 3-86 on page 144 for signal waveforms).

Figure 3-86. Programming the EEPROM Waveforms



3.21.3.6 Reading the Flash

The algorithm for reading the Flash memory is as follows (refer to Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command and address loading):

A: Load “0000 0010” command.

G: Load “High Byte” address (0x00 - 0xFF).

B: Load “Low Byte” address (0x00 - 0xFF).

M: Set $\overline{OE}$ to “0” and BS1 to “0.” The Flash word low byte can by now be read at DATA.

N: Set BS1 to “1.” The Flash word high byte can by now be read at DATA.

O: Set $\overline{OE}$ to “1.”

3.21.3.7 Reading the EEPROM

The algorithm for reading the EEPROM memory is as follows (refer to Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command and address loading):

A: Load “0000 0011” command.

G: Load “High Byte” address (0x00 - 0xFF).

B: Load “Low Byte” address (0x00 - 0xFF).

M: Set $\overline{OE}$ to “0” and BS1 to “0.” The EEPROM data byte can by now be read at DATA

O: Set $\overline{OE}$ to “1.”

3.21.3.8 Programming the Fuse Low Bits

The algorithm for programming the fuse low bits is as follows (refer to Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command and data loading):

A: Load “0100 0000” command.

C: Load “Low Byte” data. Bit n = “0” programs and bit n = “1” erases the Fuse bit.

H: Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.

3.21.3.9 Programming the Fuse High Bits

The algorithm for programming the Fuse high bits is as follows (refer to Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command and data loading):

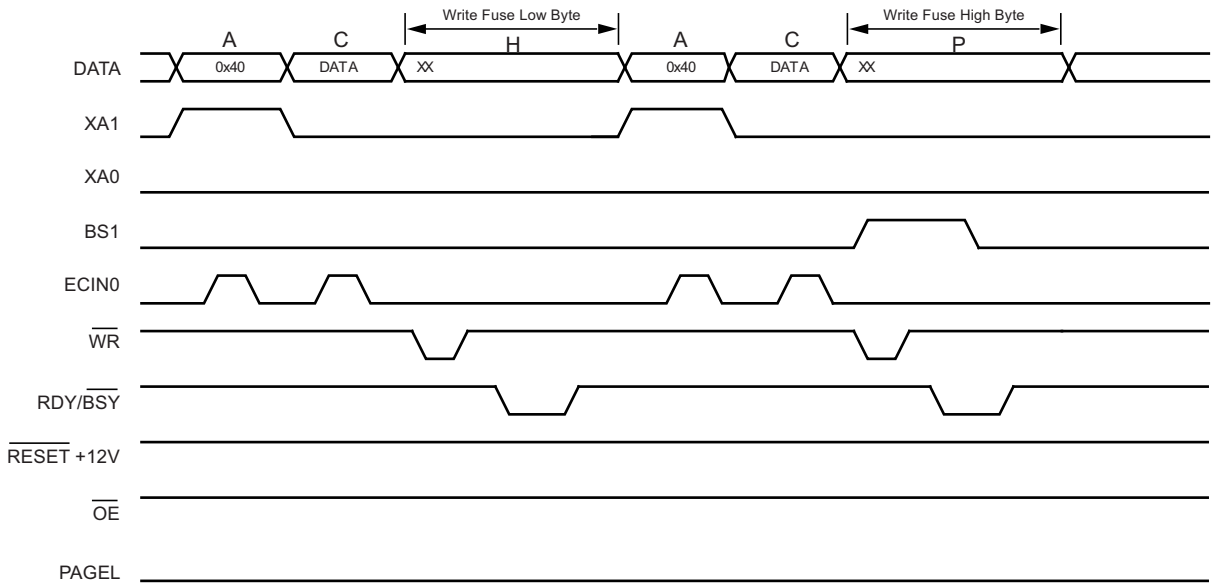
A: Load “0100 0000” command.

C: Load “Low Byte” data. Bit n = “0” programs and bit n = “1” erases the Fuse bit.

P: Write fuse high bits

1. Set BS1 to “1” and BS2 to “0.” This selects the high data byte.
2. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.
3. Set BS1 to “0.” This selects the low data byte.

Figure 3-87. Programming the FUSES Waveforms



3.21.3.10 Programming the Lock Bits

The algorithm for programming the lock bits is as follows (see Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command and data loading):

A: Load “0010 0000” command.

C: Load data low byte. Bit n = “0” programs the lock bit. If LB mode 3 is programmed (LB1 and LB2 is programmed), it is not possible to program the boot lock bits by any external programming mode.

H: Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.

The lock bits can only be cleared by executing chip erase.

3.21.3.11 Reading the Fuse and Lock Bits

The algorithm for reading the fuse and lock bits is as follows (refer to Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command loading):

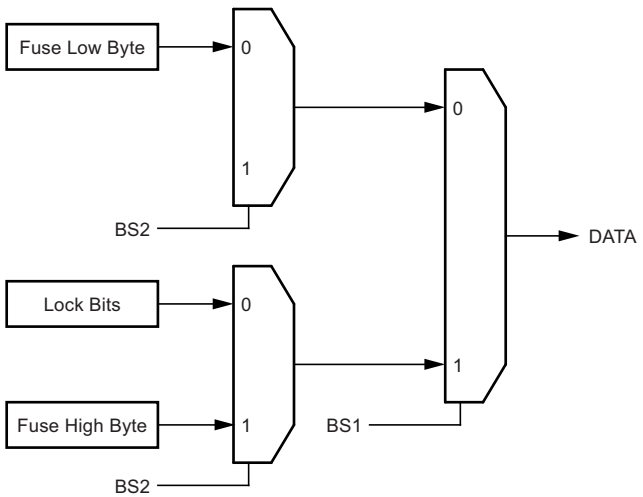
A: Load “0000 0100” command

Q: Read fuse and lock bits

1. Set $\overline{OE}$ to “0,” BS2 to “0” and BS1 to “0.” The status of the Fuse low bits can be read at DATA (“0” means programmed).
2. Set $\overline{OE}$ to “0´,” BS2 to “1” and BS1 to “1.” The status of the Fuse high bits can be read at DATA (“0” means programmed).
3. Set $\overline{OE}$ to “0,” BS2 to “0” and BS1 to “1.” The status of the lock bits can be read at DATA (“0” means programmed).

O: Set $\overline{OE}$ to “1.”

Figure 3-88. Mapping Between BS1, BS2 and the Fuse and Lock Bits During Read



3.21.3.12 Reading the Signature Bytes

The algorithm for reading the signature bytes is as follows (refer to Section 3.21.3.4 “Programming the Flash” on page 142 for more information about command and address loading):

A: Load “0000 1000” command.

B: Load “Low Byte” address (0x00 - 0x02).

M: Set $\overline{OE}$ to “0,” and BS1 to “0.” The selected signature byte can be read at DATA.

O: Set $\overline{OE}$ to “1.”

3.21.4 Parallel Programming Characteristics

Figure 3-89. Parallel Programming Timing, Including Some General Timing Requirements

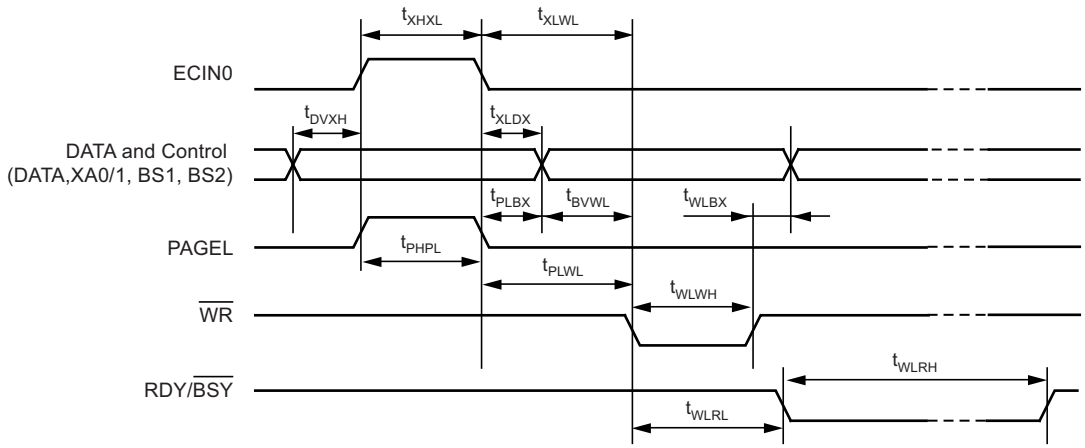
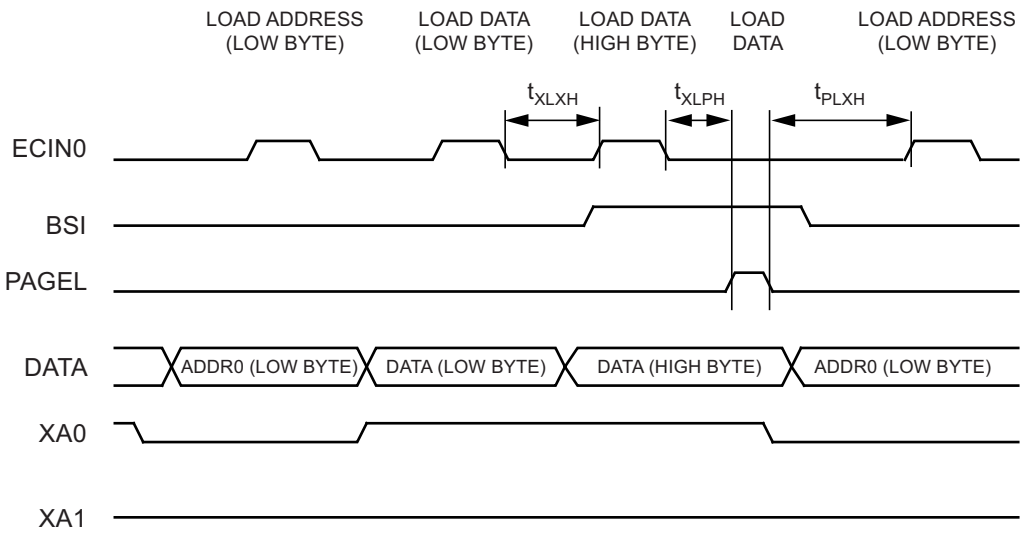
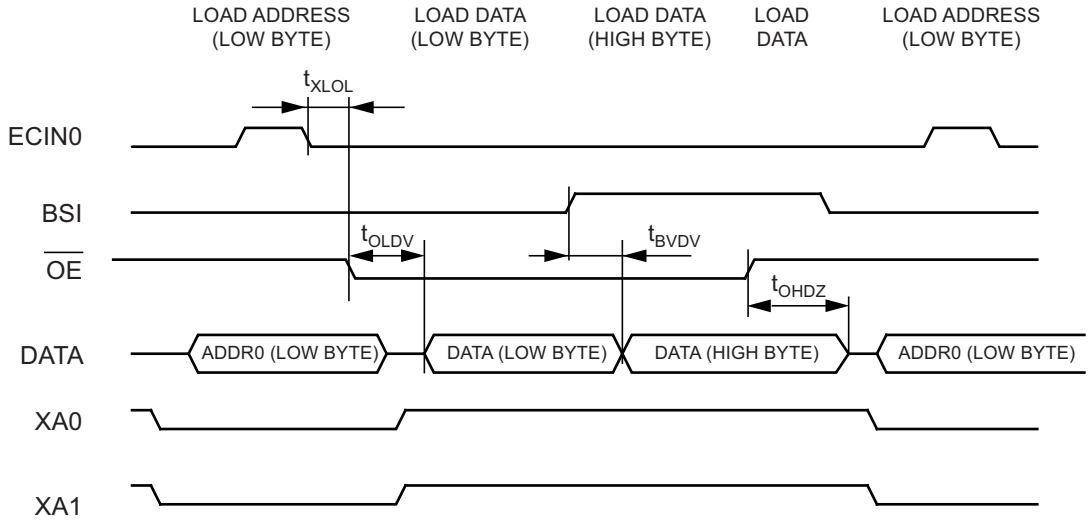


Figure 3-90. Parallel Programming Timing, Loading Sequence with Timing Requirements(1)



Note: 1. The timing requirements shown in Figure 3-89 (i.e., t_{DVXH}, t_{XHX}, and t_{XLDX}) also apply to loading operation.

Figure 3-91. Parallel Programming Timing, Reading Sequence (within the Same Page) with Timing Requirements(1)



Note: The timing requirements shown in Figure 3-89 on page 147 (i.e., t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to reading operation.

Table 3-80. Parallel Programming Characteristics, V_{CC} = 3V ±10%

Symbol	Parameter	Min	Typ	Max	Unit
V_{PP}	Programming enable voltage	11.5		12.5	V
I_{PP}	Programming enable current			250	µA
t_{DVXH}	Data and Control valid before ECIN0 High	67			ns
t_{XLXH}	ECIN0 Low to ECIN0 High	200			ns
t_{XHXL}	ECIN0 Pulse Width High	150			ns
t_{XLDX}	Data and Control Hold after ECIN0 Low	67			ns
t_{XLWL}	ECIN0 Low to $\overline{WR}$ Low	0			ns
t_{XLPH}	ECIN0 Low to PAGEL high	0			ns
t_{PLXH}	PAGEL low to ECIN0 high	150			ns
t_{BVPH}	BS1 Valid before PAGEL High	67			ns
t_{PHPL}	PAGEL Pulse Width High	150			ns
t_{PLBX}	BS1 Hold after PAGEL Low	67			ns
t_{WLBX}	BS2/1 Hold after $\overline{WR}$ Low	67			ns
t_{PLWL}	PAGEL Low to $\overline{WR}$ Low	67			ns
t_{BVWL}	BS1 Valid to $\overline{WR}$ Low	67			ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low	150			ns
t_{WLRL}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ Low	0		1	µs
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High(1)	3.7		4.5	ms
t_{WLRH_CE}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Chip Erase(2)	7.5		9	ms
t_{XLOL}	ECIN0 Low to $\overline{OE}$ Low	0			ns

Notes: 1. t_{WLRH} is valid for the write flash, write EEPROM, write fuse bits and write lock bits commands.
2. t_{WLRH_CE} is valid for the chip erase command.

Table 3-80. Parallel Programming Characteristics, V_{CC} = 3V ±10% (Continued)

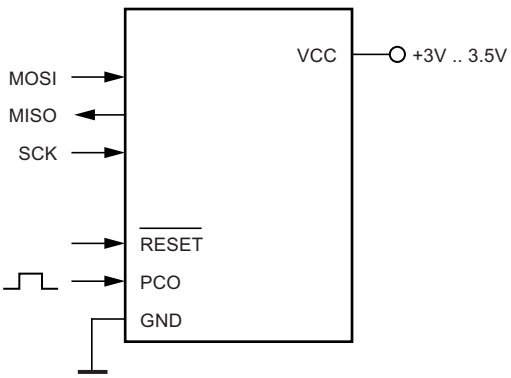
Symbol	Parameter	Min	Typ	Max	Unit
t_{BVDV}	BS1 Valid to DATA Valid	0		250	ns
t_{OLDV}	$\overline{OE}$ Low to DATA Valid			250	ns
t_{OHDZ}	$\overline{OE}$ High to DATA Tri-stated			250	ns

Notes: 1. t_{WLRH} is valid for the write flash, write EEPROM, write fuse bits and write lock bits commands.
2. t_{WLRH_CE} is valid for the chip erase command.

3.21.5 Serial Downloading

Both the Flash and EEPROM memory arrays can be programmed using the serial SPI bus while the $\overline{RESET}$ pin is pulled to GND. The serial interface consists of the SCK, MOSI (input) and MISO (output) pins. After the $\overline{RESET}$ pin has been set low the programming enable instruction needs to be executed first before program/erase operations can be executed. NOTE: The pin mapping for SPI programming is listed in Table 3-81 on page 149. Not all parts use the SPI pins dedicated for the internal SPI interface.

Figure 3-92. Serial Programming and Verify



Notes: 1. If the device is clocked by the internal oscillator, there is no need to connect a clock source to the ECIN0 pin.
2. 3.0V < VCC < 3.5V.

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in serial mode only) and there is no need to first execute the chip erase instruction. The chip erase operation turns the content of every memory location in both the program and EEPROM arrays into 0xFF.

Important:
CLK (SPI-Clock) must be less than ¼ of CPU clock.

3.21.5.1 Serial Programming and Pin Mapping

Table 3-81. Pin Name Mapping Serial Programming

Symbol	Pin Name	I/O	Description
MOSI	PB3	I	Serial data in
MISO	PB4	O	Serial data out
SCK	PB5	I	Serial clock

3.21.5.2 Serial Programming Algorithm

When writing serial data to the ATA6289, data is clocked on the rising edge of the SCK. When reading data from the ATA6289, data is clocked on the falling edge of the SCK. See Figure 3-94 on page 152 for timing details. To program and verify the ATA6289 in serial programming mode, the following sequence is recommended (see “Serial Programming Instruction Set” in Table 3-83 on page 151):

1. Power-up sequence: Apply power between VCC and GND while $\overline{\text{RESET}}$ pin and SCK are set to “0.” In some systems, the programmer cannot guarantee that the SCK is held low during power-up. In this case, the $\overline{\text{RESET}}$ pin must be given a positive pulse for a duration of at least two CPU clock cycles after the SCK has been set to “0.”
2. Wait for at least 20ms and enable serial programming by sending the programming enable serial instruction to the MOSI pin.
3. The serial programming instructions do not work if communication is out of synchronization. If in synchronization, the second byte (0x53) echoes back when the third byte of the programming enable instruction is issued. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the 0x53 did not echo back, give RESET pin a positive pulse and issue a new programming enable command.
4. The Flash is programmed one page at a time. The memory page is loaded one byte at a time by supplying the six LSBs of the address and data together with the load program memory page instruction. To ensure correct loading of the page, the data low byte must be loaded before data high byte is applied to a given address. The program memory page is stored by loading the write program memory page instruction with the seven MSBs of the address. If polling (RDY/$\overline{\text{BSY}}$) is not used, the user must wait at least t_{WD_FLASH} before issuing the next page (see Table 3-82 on page 150). Accessing the serial programming interface before the Flash write operation is completed can result in incorrect programming.
5. A: The EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate write instruction. An EEPROM memory location is first automatically erased before new data is written. If polling (RDY/$\overline{\text{BSY}}$) is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte (see Table 3-82 on page 150). In a chip erased device, no 0xFFs in the data file(s) need to be programmed.
   B: The EEPROM array is programmed one page at a time. The memory page is loaded one byte at a time by supplying the six LSBs of the address and data together with the load EEPROM memory page instruction. The EEPROM memory page is stored by loading the write EEPROM memory page instruction with the seven MSBs of the address. When using EEPROM page access only byte locations loaded with the load EEPROM memory page instruction are altered. The remaining locations remain unchanged. If polling (RDY/$\overline{\text{BSY}}$) is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte (see Table 3-82 on page 150). In a chip erased device, no 0xFFs in the data file(s) need to be programmed.
6. Any memory location can be verified by using the read instruction which returns the content at the selected address at serial output MISO.
7. At the end of the programming session, the $\overline{\text{RESET}}$ pin can be set high to commence normal operation.
8. Power-off sequence (if needed):
   Set $\overline{\text{RESET}}$ pin to “1.”
   Turn VCC power off.

Table 3-82. Minimum Wait Delay Before Writing the next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FLASH}	4.5ms
t_{WD_EEPROM}	3.6ms
t_{WD_ERASE}	9ms

Table 3-83. Serial Programming Instruction Set

Instruction	Instruction Format				
	Byte1	Byte2	Byte3	Byte4	
Programming Enable	1010_1100	0101_0011	xxxx_xxxx	xxxx_xxxx	Enable serial programming after $\overline{\text{RESET}}$ pin goes low
Chip Erase	1010_1100	100x_xxxx	xxxx_xxxx	xxxx_xxxx	Chip erase EEPROM and Flash
Read Program Memory	0010_H000	0000_aaaa	bbbb_bbbb	oooo_oooo	Read H (high or low) data o from Program memory at word address a:b
Load Program Memory Page	0100_H000	000x_xxxx	xxxb_bbbb	iiii_iiii	Write H (high or low) data i to program memory page at word address b. Data low byte must be loaded before data high byte is applied within the same address
Write Program Memory Page	0100_1100	0000_aaaa	bbbx_xxxx	xxxx_xxxx	Write program memory page at address a:b
Read EEPROM Memory	1010_0000	000x_xxaa	bbbb_bbbb	oooo_oooo	Read data o from EEPROM memory at address a:b
Write EEPROM Memory (byte access)	1100_0000	000x_xxaa	bbbb_bbbb	iiii_iiii	Write data i to EEPROM memory at address a:b
Load EEPROM Memory Page (page access)	1100_0001	0000_0000	0000_00bb	iiii_iiii	Load data i to EEPROM memory page buffer. After data is loaded, program EEPROM page
Write EEPROM Memory Page (page access)	1100_0010	00xx_xxaa	bbbb_bb00	xxxx_xxxx	Write EEPROM page at address a:b
Read Lock Bits	0101_0000	0000_0000	xxxx_xxxx	xxoo_oooo	Read lock bits. “0” = programmed, “1” = unprogrammed
Write Lock Bits	1010_1100	111x_xxxx	xxxx_xxxx	11ii_iiii	Write lock bits. Set bits = “0” to program lock bits
Read Signature Byte	0011_0000	000x_xxxx	xxxx_xxbb	oooo_oooo	Read signature byte at address b
Write Fuse bits	1010_1100	1010_0000	xxxx_xxxx	iiii_iiii	Set bits = “0” to program, “1” to unprogram
Write Fuse High bits	1010_1100	1010_1000	xxxx_xxxx	iiii_iiii	Set bits = “0” to program, “1” to unprogram
Read Fuse bits	0101_0000	0000_0000	xxxx_xxxx	oooo_oooo	Read fuse bits. “0” = programmed, “1” = unprogrammed.
Read Fuse High bits	0101_1000	0000_1000	xxxx_xxxx	oooo_oooo	Read Fuse High bits. “0” = programmed, “1” = un- programmed.
Poll $\overline{\text{RDY}}$/$\overline{\text{BSY}}$	1111_0000	0000_0000	xxxx_xxxx	xxxx_xxxo	If o = “1,” a programming operation is still busy. Wait until this bit returns to “0” before applying another command

Notes:
1. Bits are programmed '0,' unprogrammed '1'
2. To ensure future compatibility, unused fuses and lock bits should be unprogrammed (‘1’).
3. Refer to the corresponding section for fuse and lock bits, calibration and signature bytes, and page size.
4. Instructions accessing program memory use a word address. This address may be random within the page range.
5. See http://www.atmel.com/avr for Application Notes regarding programming and programmers.

If the LSB in RDY/$\overline{\text{BSY}}$ data byte out is ‘1,’ a programming operation is still pending. Wait until this bit returns '0' before the next instruction is carried out. Within the same page, the low data byte must be loaded prior to the high data byte. After data is loaded to the page buffer, program the EEPROM page (see Figure 3-93).

Figure 3-93. Serial Programming Instruction Example

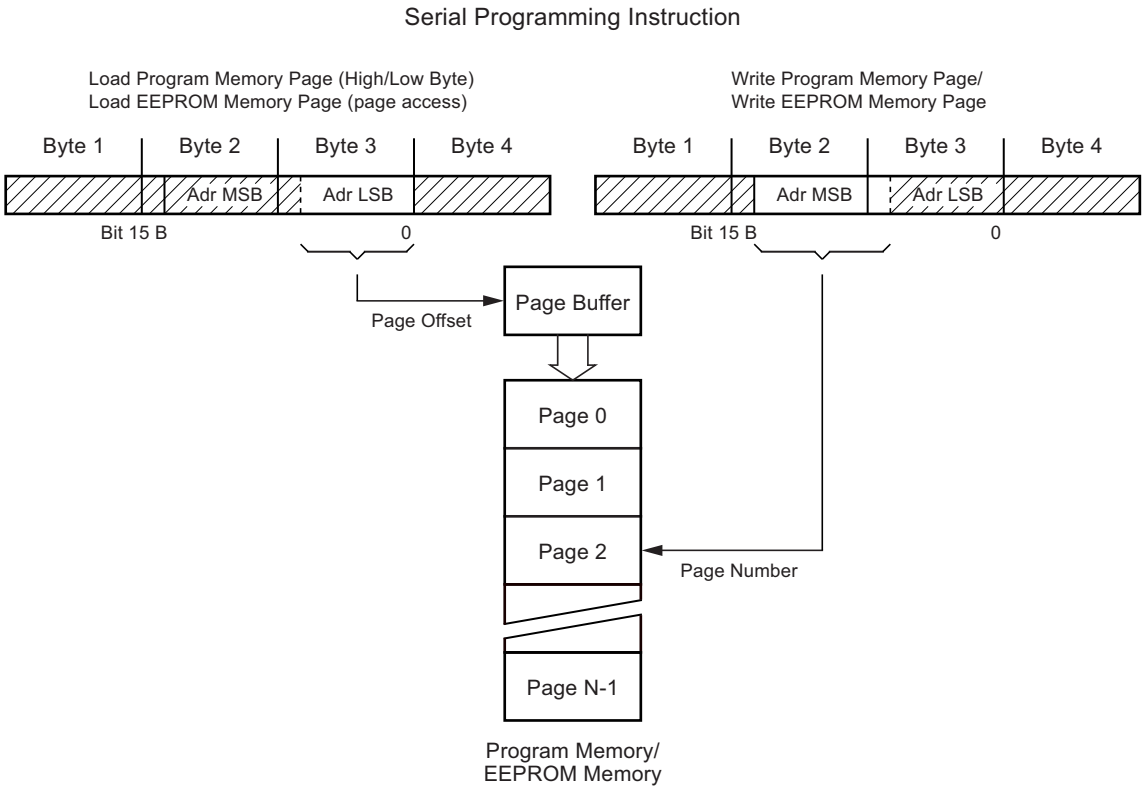
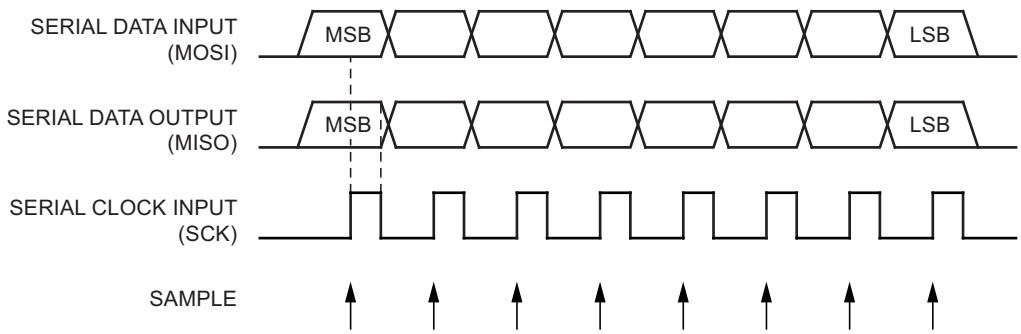


Figure 3-94. Serial Programming Waveforms



3.21.6 Register Summary

Table 3-84. Register Summary

Address	Name	Bi t 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0xFF)	Reserved	-	-	-	-	-	-	-	-
(0xFE)	Reserved	-	-	-	-	-	-	-	-
---	Reserved	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
---	Reserved	-	-	-	-	-	-	-	-
(0x88)	Reserved	-	-	-	-	-	-	-	-
(0x87)	Reserved	-	-	-	-	-	-	-	-
(0x86)	Reserved	-	-	-	-	-	-	-	-
(0x85)	LFIDCH	LF Receiver – Identifier Compare Register High Byte LFIDCH[15..8]							
(0x84)	LFIDCL	LF Receiver – Identifier Compare Register Low Byte LFIDCL[7..0]							
(0x83)	LFHCR	-	LF Receiver – Header Compare Register LFHCR[6..0]						
(0x82)	LFRCR	LFCS2	LFCS1	LFCS0	LFRSS	LFWM1	LFWM0	LFBM	LFEN
(0x81)	LFIMR	-	-	-		-	LFEIM	LFBIM	LFWIM
(0x80)	Reserved	-	-	-	-	-	-	-	-
(0x7F)	T3IMR	-	-	-	-	T3CPIM	T3CBIM	T3CAIM	T3OIM
(0x7E)	T3CRB	-	T3CPRM	T3CRMB	T3SAMB	T3CTMB	T3CRMA	T3SAMA	T3CTMA
(0x7D)	T3MRB	-	-	-	T3TOP	-	T3M2	T3M1	T3M0
(0x7C)	T3MRA	T3ICS1	T3ICS0	T3CNC	T3CE1	T3CE0	T3CS2	T3CS1	T3CS0
(0x7B)	T3CORBH	Timer/Counter3 – Output Compare Register B High Byte T3CORBH[7..0]							
(0x7A)	T3CORBL	Timer/Counter3 – Output Compare Register B Low Byte T3CORBL [7..0]							
(0x79)	T3CORAH	Timer/Counter3 – Output Compare Register A High Byte T3CORAH[7..0]							
(0x78)	T3CORAL	Timer/Counter3 – Output Compare Register A Low Byte T3CORAL[7..0]							
(0x77)	T3ICRH	Timer/Counter3 – Input Capture Register High Byte T3ICRH[7..0]							
(0x76)	T3ICRL	Timer/Counter3 – Input Capture Register Low Byte T3ICRL[7..0]							
(0x75)	Reserved	-	-	-	-	-	-	-	-
(0x74)	T2IMR	-	-	T2TCIM	T2TXIM	T2RXIM	T2CPIM	T2CIM	T2OIM
(0x73)	T2MRB	T2SSIE	T2CPOL	-	T2TOP	T2M3	T2M2	T2M1	T2M0
(0x72)	T2MRA	T2TP1	T2TP0	T2CNC	T2CE1	T2CE0	T2CS2	T2CS1	T2CS0
(0x71)	T2CORH	Timer/Counter2 – Output Compare Register High Byte T2CORH[7..0]							
(0x70)	T2CORL	Timer/Counter2 – Output Compare Register Low Byte T2CORL[7..0]							
(0x6F)	T2ICRH	Timer/Counter2 – Input Capture Register High Byte T2ICRH[7..0]							
(0x6E)	T2ICRL	Timer/Counter2 – Input Capture Register Low Byte T2ICRL[7..0]							
(0x6D)	Reserved	-	-	-	-	-	-	-	-
(0x6C)	PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
(0x6B)	PCMSK1	-	-	-	-	-	PCINT10	PCINT9	PCINT8
(0x6A)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
(0x69)	EICRA	-	-		-	ISC11	ISC10	ISC01	ISC00
(0x68)	Reserved	-	-	-	-	-	-	-	-
(0x67)	MSVCAL	Motion Sensor Voltage Reference Calibration Register VRCAL[7..0]							
(0x66)	FRCCAL	-	-	FRC – Oscillator Calibration Register FCAL[5..0]					
(0x65)	SRCCAL	SRC – Oscillator Calibration Register SCAL[7..0]							

Note: 1. In ATA6289 only bits SP9, SP8 are in use.

Table 3-84. Register Summary (Continued)

Address	Name	Bi t 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x64)	TSCR	-	-	-	-	-	-	-	TSSD
(0x63)	Reserved	-	-	-	-	-	-	-	-
(0x62)	Reserved	-	-	-	-	-	-	-	-
(0x61)	SIMSK	-	-	-	-	-	-	-	MSIE
(0x60)	WDTCR	-	-	-	WDCE	WDE	WDPS2	WDPS1	WDPS0
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C
0x3E (0x5E)	SPH	Stack Pointer High Byte SP[15..8] (1)							
0x3D (0x5D)	SPL	Stack Pointer Low Byte SP[7..0]							
0x3C (0x5C)	CLKPR	CLPCE	-	CLTPS2	CLTPS1	CLTPS0	CLKPS2	CLKPS1	CLKPS0
0x3B (0x5B)	CMIMR	-	-	-	-	-		-	ECIE
0x3A (0x5A)	Reserved	-		-	-	-	-	-	-
0x39 (0x59)	T0CR	T0PBS2	T0PBS1	T0PBS0	T0PR	T0IE	T0PAS2	T0PAS1	T0PAS0
0x38 (0x58)	T1CR	T1IE	-	T1CS2	T1CS1	T1CS0	T1PS2	T1PS1	T1PS0
0x37 (0x57)	SPMCSR	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SELFPRGEN
0x36 (0x56)	LFRB	LF Receiver data Buffer LFRB[7..0]							
0x35 (0x55)	MCUCR	-	-	-	PUD	-	-	IVSEL	IVCE
0x34 (0x54)	MCUSR	-	-	TSRF	-	WDRF	BORF	EXTRF	PORF
0x33 (0x53)	SMCR	-	-	-	-	SM2	SM1	SM0	SE
0x32 (0x52)	LFCDR	LFSCE	LFRST	-	-	-	-	-	LFDO
0x31 (0x51)	Reserved	-		-	-	-	-	-	-
0x30 (0x50)	LFRR	-	LF Receiver RSSI Data Register LFRR[6..0]						
0x2F (0x4F)	T2MDR	Timer2 Modulator Data Register (RXD / TXD) T2MDR[7..0]							
0x2E (0x4E)	SPDR	SPI Data Register SPDR[7..0]							
0x2D (0x4D)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
0x2B (0x4B)	GPIOR2	General Purpose I/O Register 2 GPIOR2[7..0]							
0x2A (0x4A)	GPIOR1	General Purpose I/O Register 1 GPIOR1[7..0]							
0x29 (0x49)	SCCR	-	-	-	SCCS2	SCCS1	SCCS0	SRCC1	SRCC0
0x28 (0x48)	SCR	-	-	-	-	SMEN	SEN1	SEN0	SMS
0x27 (0x47)	SVCR	-	-		SVCS4	SVCS3	SVCS2	SVCS1	SVCS0
0x26 (0x46)	Reserved		-	-	-	-	-	-	-
0x25 (0x45)	Reserved		-	-	-	-	-	-	-
0x24 (0x44)	EIMSK	-	-	-	-	-	-	INT1	INT0
0x23 (0x43)	PCICR	-	-	-	-	-	PCIE2	PCIE1	PCIE0
0x22 (0x42)	EEARH	-	-	-	-			-	EEAR8
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte EEAR[7..0]							
0x20 (0x40)	EEDR	EEPROM Data Register EEDR[7..0]							
0x1F (0x3F)	EECR	-	-	EEPM1	EEPM0	EERIE	EEMWE	EEWE	EERE
0x1E (0x3E)	GPIOR0	General Purpose I/O Register 0 GPIOR0[7..0]							
0x1D (0x3D)	EIFR	-	-		-	-	-	INTF1	INTF0
0x1C (0x3C)	T3IFR	-	-	-	-	T3ICF	T3COBF	T3COAF	T3OFF
0x1B (0x3B)	T2IFR			T2TCF	T2TXF	T2RXF	T2ICF	T2COF	T2OFF
0x1A (0x3A)	T10IFR	-	-	-	-	-	-	T1F	T0F

Note: 1. In ATA6289 only bits SP9, SP8 are in use.

Table 3-84. Register Summary (Continued)

Address	Name	Bi t 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x19 (0x39)	SSFR	-	-	-	-	-		MSENO	MSENF
0x18 (0x38)	LFFR	-	-	-	-	LFRF	LFEDF	LFBF	LFWPF
0x17 (0x37)	PCIFR	-	-	-	-	-	PCIF2	PCIF1	PCIF0
0x16 (0x36)	VMCSR	BODLS	BODPD	VMF	VMIM	VMLS2	VMLS1	VMLS0	VMEN
0x15 (0x35)	Reserved	-		-	-	-	-	-	-
0x14 (0x34)	T3CRA	T3E	T3TS	-	-	-	T3CR	T3SCE	T3AC
0x13 (0x33)	Reserved	-		-	-	-	-	-	-
0x12 (0x32)	T2CRB	-	-	-	-	-	-	-	T2SCE
0x11 (0x31)	T2CRA	T2E	T2TS	T2ICS	-	T2CRM	T2CR	T2CTM	T2OTM
0x10 (0x30)	CMSR	-	-		-	-	-	-	ECF
0x0F (0x2F)	CMCR	CMCCE	-	ECINS	CCS	CMONEN	SRCD	CMM1	CMM0
0x0E (0x2E)	Reserved	-	-	-	-	-	-	-	
0x0D (0x2D)	Reserved		-	-	-	-	-	-	
0x0C (0x2C)	Reserved		-	-	-	-	-	-	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x08 (0x28)	PORTC	-	-	-	-	-	PORTC2	PORTC1	PORTC0
0x07 (0x27)	DDRC	-	-	-		-	DDC2	DDC1	DDC0
0x06 (0x26)	PINC	-	-	-	-	-	PINC2	PINC1	PINC0
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x02 (0x22)	Reserved		-	-	-	-	-	-	
0x01 (0x21)	Reserved		-	-	-	-	-	-	
0x00 (0x20)	Reserved		-	-	-	-	-	-	

Note: 1. In ATA6289 only bits SP9, SP8 are in use.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.

Some of the status flags are cleared by writing a logic one to them. Note that, unlike most other Atmel® AVR®s, the CBI and SBI instructions only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATA6289 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

3.22 Instruction Set Summary

Table 3-85. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
Arithmetic and Logic Instructions					
ADD	Rd, Rr	Add two registers	Rd <-- Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with carry two registers	Rd <-- Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add immediate to word	Rdh:Rdl <-- Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two registers	Rd <-- Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract constant from register	Rd <-- Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with carry two registers	Rd <-- Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with carry constant from register	Rd <-- Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract immediate from word	Rdh:Rdl <-- Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND registers	Rd <-- Rd × Rr	Z,N,V	1
ANDI	Rd, K	Logical AND register and constant	Rd <-- Rd × K	Z,N,V	1
OR	Rd, Rr	Logical OR registers	Rd <-- Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR register and constant	Rd <-- Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR registers	Rd <-- Rd ⊕ Rr	Z,N,V	1
COM	Rd	One’s complement	Rd <-- 0xFF - Rd	Z,C,N,V	1
NEG	Rd	Two’s complement	Rd <-- 0x00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set bit(s) in register	Rd <-- Rd v K	Z,N,V	1
CBR	Rd,K	Clear bit(s) in register	Rd <-- Rd × (0xFF - K)	Z,N,V	1
INC	Rd	Increment	Rd <-- Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd <-- Rd - 1	Z,N,V	1
TST	Rd	Test for zero or minus	Rd <-- Rd × Rd	Z,N,V	1
CLR	Rd	Clear register	Rd <-- Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set register	Rd <-- 0xFF	None	1
MUL	Rd, Rr	Multiply unsigned	R1:R0 <-- Rd × Rr	Z,C	2
MULS	Rd, Rr	Multiply signed	R1:R0 <-- Rd × Rr	Z,C	2
MULSU	Rd, Rr	Multiply signed with unsigned	R1:R0 <-- Rd × Rr	Z,C	2
FMUL	Rd, Rr	Fractional multiply unsigned	R1:R0 <-- (Rd × Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional multiply signed	R1:R0 <-- (Rd × Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional multiply signed with unsigned	R1:R0 <-- (Rd × Rr) << 1	Z,C	2
Branch Instructions					
RJMP	k	Relative jump	PC <-- PC + k + 1	None	2
IJMP		Indirect jump to (Z)	PC <-- Z	None	2
JMP(1)	k	Direct jump	PC <-- k	None	3
RCALL	k	Relative subroutine call	PC <-- PC + k + 1	None	3
ICALL		Indirect call to (Z)	PC <-- Z	None	3
CALL(1)	k	Direct subroutine call	PC <-- k	None	4
RET		Subroutine return	PC <-- STACK	None	4
RETI		Interrupt return	PC <-- STACK	I	4
CPSE	Rd,Rr	Compare, skip if equal	if (Rd = Rr) PC <-- PC + 2 or 3	None	1/2/3

Note: 1. These instructions are only available in ATmega168P and ATmega328P, not in ATA6289.

Table 3-85. Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
CP	Rd,Rr	Compare	Rd - Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with carry	Rd - Rr - C	Z, N,V,C,H	1
CPI	Rd,K	Compare register with immediate	Rd - K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if bit in register cleared	if (Rr(b)=0) PC <-- PC + 2 or 3	None	1/2/3
SBRS	Rr, b	Skip if bit in register is set	if (Rr(b)=1) PC <-- PC + 2 or 3	None	1/2/3
SBIC	P, b	Skip if bit in I/O register cleared	if (P(b)=0) PC <-- PC + 2 or 3	None	1/2/3
SBIS	P, b	Skip if bit in I/O register is set	if (P(b)=1) PC <-- PC + 2 or 3	None	1/2/3
BRBS	s, k	Branch if status flag set	if (SREG(s) = 1) then PC <-- PC+k + 1	None	1/2
BRBC	s, k	Branch if status flag cleared	if (SREG(s) = 0) then PC <-- PC+k + 1	None	1/2
BREQ	k	Branch if equal	if (Z = 1) then PC <-- PC + k + 1	None	1/2
BRNE	k	Branch if not equal	if (Z = 0) then PC <-- PC + k + 1	None	1/2
BRCS	k	Branch if carry set	if (C = 1) then PC <-- PC + k + 1	None	1/2
BRCC	k	Branch if carry cleared	if (C = 0) then PC <-- PC + k + 1	None	1/2
BRSH	k	Branch if same or higher	if (C = 0) then PC <-- PC + k + 1	None	1/2
BRLO	k	Branch if lower	if (C = 1) then PC <-- PC + k + 1	None	1/2
BRMI	k	Branch if minus	if (N = 1) then PC <-- PC + k + 1	None	1/2
BRPL	k	Branch if plus	if (N = 0) then PC <-- PC + k + 1	None	1/2
BRGE	k	Branch if greater or equal, signed	if (N ⊕ V= 0) then PC <-- PC + k + 1	None	1/2
BRLT	k	Branch if less than zero, signed	if (N ⊕ V= 1) then PC <-- PC + k + 1	None	1/2
BRHS	k	Branch if half carry flag set	if (H = 1) then PC <-- PC + k + 1	None	1/2
BRHC	k	Branch if half carry flag cleared	if (H = 0) then PC <-- PC + k + 1	None	1/2
BRTS	k	Branch if T flag set	if (T = 1) then PC <-- PC + k + 1	None	1/2
BRTC	k	Branch if T flag cleared	if (T = 0) then PC <-- PC + k + 1	None	1/2
BRVS	k	Branch if overflow flag is set	if (V = 1) then PC <-- PC + k + 1	None	1/2
BRVC	k	Branch if overflow flag is cleared	if (V = 0) then PC <-- PC + k + 1	None	1/2
BRIE	k	Branch if interrupt enabled	if (I = 1) then PC <-- PC + k + 1	None	1/2
BRID	k	Branch if interrupt disabled	if (I = 0) then PC <-- PC + k + 1	None	1/2
Bit and Bit-Test Instructions					
SBI	P,b	Set bit in I/O register	I/O(P,b) <-- 1	None	2
CBI	P,b	Clear bit in I/O register	I/O(P,b) <-- 0	None	2
LSL	Rd	Logical shift left	Rd(n+1) <-- Rd(n), Rd(0) <-- 0	Z,C,N,V	1
LSR	Rd	Logical shift right	Rd(n) <-- Rd(n+1), Rd(7) <-- 0	Z,C,N,V	1
ROL	Rd	Rotate left through carry	Rd(0) <-- C,Rd(n+1) <-- Rd(n),C <-- Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate right through carry	Rd(7) <-- C,Rd(n) <-- Rd(n+1),C <-- Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic shift right	Rd(n) <-- Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap nibbles	Rd(3..0) <-- Rd(7..4),Rd(7..4) <-- Rd(3..0)	None	1
BSET	s	Flag set	SREG(s) <-- 1	SREG(s)	1

Note: 1. These instructions are only available in ATmega168P and ATmega328P, not in ATA6289.

Table 3-85. Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BCLR	s	Flag clear	SREG(s) <-- 0	SREG(s)	1
BST	Rr, b	Bit store from register to T	T <-- Rr(b)	T	1
BLD	Rd, b	Bit load from T to register	Rd(b) <-- T	None	1
SEC		Set carry	C <-- 1	C	1
CLC		Clear carry	C <-- 0	C	1
SEN		Set negative flag	N <-- 1	N	1
CLN		Clear negative flag	N <-- 0	N	1
SEZ		Set zero flag	Z <-- 1	Z	1
CLZ		Clear zero flag	Z <-- 0	Z	1
SEI		Global interrupt enable	I <-- 1	I	1
CLI		Global interrupt disable	I <-- 0	I	1
SES		Set signed test flag	S <-- 1	S	1
CLS		Clear signed test flag	S <-- 0	S	1
SEV		Set two complement overflow.	V <-- 1	V	1
CLV		Clear two complement overflow	V <-- 0	V	1
SET		Set T in SREG	T <-- 1	T	1
CLT		Clear T in SREG	T <-- 0	T	1
SEH		Set half carry flag in SREG	H <-- 1	H	1
CLH		Clear half carry flag in SREG	H <-- 0	H	1
Data Transfer Instructions					
MOV	Rd, Rr	Move between registers	Rd <-- Rr	None	1
MOVW	Rd, Rr	Copy register word	Rd+1:Rd <-- Rr+1:Rr	None	1
LDI	Rd, K	Load immediate	Rd <-- K	None	1
LD	Rd, X	Load indirect	Rd <-- (X)	None	2
LD	Rd, X+	Load indirect and post-Inc.	Rd <-- (X), X <-- X + 1	None	2
LD	Rd, - X	Load indirect and pre-dec.	X <-- X - 1, Rd <-- (X)	None	2
LD	Rd, Y	Load indirect	Rd <-- (Y)	None	2
LD	Rd, Y+	Load indirect and post-inc.	Rd <-- (Y), Y <-- Y + 1	None	2
LD	Rd, - Y	Load indirect and pre-dec.	Y <-- Y - 1, Rd <-- (Y)	None	2
LDD	Rd,Y+q	Load indirect with displacement	Rd <-- (Y + q)	None	2
LD	Rd, Z	Load indirect	Rd <-- (Z)	None	2
LD	Rd, Z+	Load indirect and post-inc.	Rd <-- (Z), Z <--_ Z+1	None	2
LD	Rd, -Z	Load indirect and pre-dec.	Z <-- Z - 1, Rd <-- (Z)	None	2
LDD	Rd, Z+q	Load indirect with displacement	Rd <-- (Z + q)	None	2
LDS	Rd, k	Load direct from SRAM	Rd <-- (k)	None	2
ST	X, Rr	Store indirect	(X) <-- Rr	None	2
ST	X+, Rr	Store indirect and post-Inc.	(X) <-- Rr, X <-- X + 1	None	2
ST	- X, Rr	Store indirect and pre-dec.	X <-- X - 1, (X) <-- Rr	None	2
ST	Y, Rr	Store indirect	(Y) <-- Rr	None	2
ST	Y+, Rr	Store indirect and post-inc.	(Y) <-- Rr, Y <-- Y + 1	None	2
ST	- Y, Rr	Store indirect and pre-dec.	Y <-- Y - 1, (Y) <-- Rr	None	2

Note: 1. These instructions are only available in ATmega168P and ATmega328P, not in ATA6289.

Table 3-85. Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
STD	Y+q,Rr	Store indirect with displacement	(Y + q) <--_ Rr	None	2
ST	Z, Rr	Store indirect	(Z) <-- Rr	None	2
ST	Z+, Rr	Store indirect and post-inc.	(Z) <-- Rr, Z <-- Z + 1	None	2
ST	-Z, Rr	Store indirect and pre-dec.	Z <-- Z - 1, (Z) <-- Rr	None	2
STD	Z+q,Rr	Store indirect with displacement	(Z + q) <-- Rr	None	2
STS	k, Rr	Store direct to SRAM	(k) <-- Rr	None	2
LPM		Load program memory	R0 <-- (Z)	None	3
LPM	Rd, Z	Load program memory	Rd <-- (Z)	None	3
LPM	Rd, Z+	Load program memory and post-inc	Rd <-- (Z), Z <-- Z+1	None	3
SPM		Store program memory	(Z) <-- R1:R0	None	-
IN	Rd, P	In port	Rd <-- P	None	1
OUT	P, Rr	Out port	P <-- Rr	None	1
PUSH	Rr	Push register on stack	STACK <-- Rr	None	2
POP	Rd	Pop register from stack	Rd <-- STACK	None	2
MCU Control Instructions					
NOP		No operation		None	1
SLEEP		Sleep	See specific description for sleep function	None	1
WDR		Watchdog reset	See specific description for WDR/timer	None	1
BREAK		Break	For on-chip debug only	None	N/A

Note: 1. These instructions are only available in ATmega168P and ATmega328P, not in ATA6289.

4. UHF ASK/FSK Transmitter Atmel ATA5757

4.1 Features

- PLL transmitter IC with single-ended output
- High output power (about 6dBm) at 8.5mA (433MHz) typical value
- Divided by 32 (Atmel ATA5757) blocks for 13MHz crystal frequencies and for low XTO startup times
- Modulation scheme ASK/FSK with internal FSK switch
- Up to 20Kbaud Manchester coding up to 40Kbaud NRZ coding
- Power-down idle and power-up modes to adjust corresponding power consumption through ASK/FSK/ENABLE input pins
- ENABLE input for parallel usage of controlling pins in a 3-wire bus system
- CLK output switches ON if the crystal current amplitude has reached 35% to 80% of its final value
- Crystal oscillator time until CLK output is activated, typically 0.6ms
- Supply voltage 2.0V to 3.6V in operating temperature range of –40°C to +125°C

4.2 Benefits

- Low parasitic FSK switch integrated
- Very short and reproducible time-to-transmit typically < 0.85ms
- 13.56MHz crystals give opportunity for small package sizes

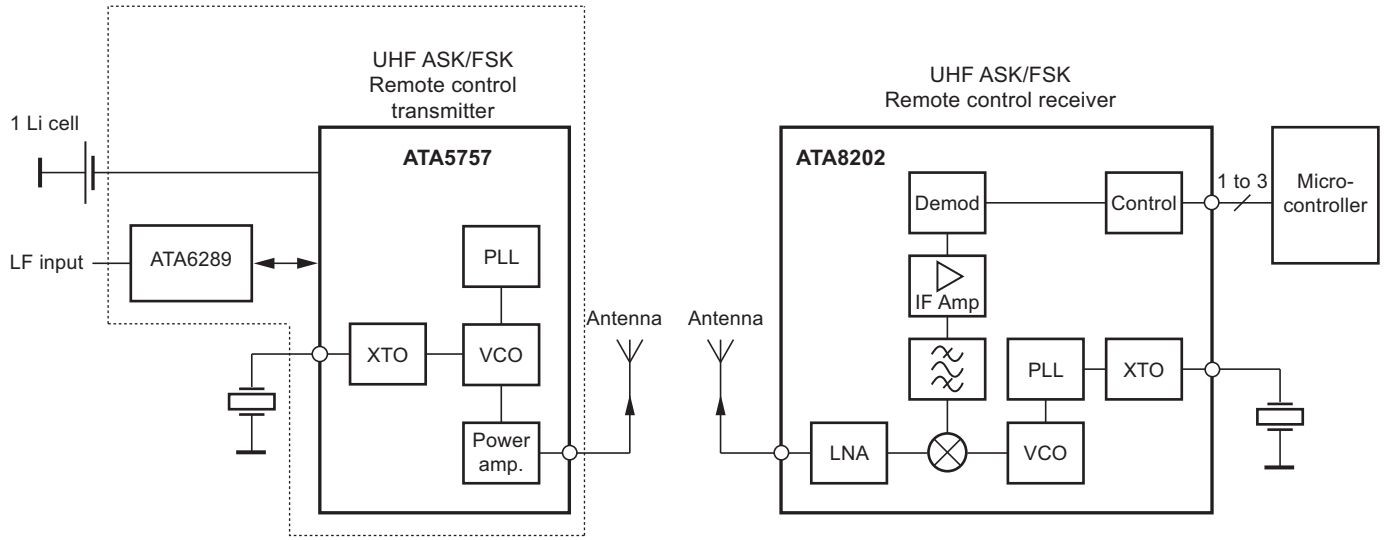
4.3 Short Overview

The Atmel® ATA5757 is a PLL transmitter IC which is placed stacked die on top of the ATA6289 (see Figure 2-2 on page 4). With the three inter-die connections ASK (PD5, T2O1) / FSK (PD6, T2O2) / ENABLE (PD3) the ATA6289 can control the Atmel ATA5757 (see also Table 4-2 on page 163). T2O1, T2O1 are the modulator outputs of timer2, i.e., the Atmel ATA5757 can be directly modulated with timer2. The fourth inter-die connection CLK (PD4) is the clock output of the Atmel ATA5757 and can be used as the external clock for the ATA6289 (see also Section 4.6.2 “CLK Output and Clock Pulse Take-Over by ATA6289” on page 163).

The Atmel ATA5757 has been developed for transmission systems at data rates of up to 20Kbaud Manchester coding and 40Kbaud NRZ coding. The transmitting frequency range is 432MHz to 448MHz (Atmel ATA5757). It can be used in both FSK and ASK systems. With the short settling time of the crystal oscillator the IC is well suited for pressure/motion sensor systems or other remote control applications/systems.

Important: For more detailed information about Atmel ATA5757, please refer to the complete datasheet which can be found on the Atmel website (www.atmel.com).

Figure 4-1. System Block Diagram



4.4 Pin Description

Table 4-1. Pin Description of Atmel ATA5757

Pin	Symbol	Function
Inter-die connection PD4 (ECIN1) to CLK	CLK	Clock output signal for the microcontroller. The clock output frequency is set by the crystal to $f_{XTAL/8}$. The CLK output stays low in power-down mode and after enabling the PLL. The CLK output switches on if the oscillation amplitude of the crystal has reached a certain level.
Inter-die connection PD5 (T2O1) to ASK	ASK	Switches on the power amplifier for ASK modulation and enables the PLL and XTO if the ENABLE pin is high/open (PD3 = high/tri-state).
Inter-die connection PD6 (T2O2) to FSK	FSK	Switches off the FSK switch (switch has high Z if signal at pin FSK is high) and enables the PLL and the XTO if the ENABLE pin is high/open (PD3 = high/ tri-State).
3	ANT2	Emitter of antenna output stage
4	ANT1	Open collector antenna output
8	XTO2	Diode switch, used for FSK modulation
10	XTO1	Connection for crystal
11	VSRF	Supply voltage
12	GNDRF	Ground
Inter-die connection PD3 to ENABLE	ENABLE	ENABLE input If ENABLE is low (PD3 = low) and the ASK or FSK pin is high, the device stays in idle mode. In normal operation ENABLE is set high/open (PD3 = high/tri-state) and ASK or FSK is used to enable the device.

4.5 General Description

The VCO is locked to $32 \times f_{XTAL}$ for Atmel® ATA5757. Thus a 13.56MHz crystal is needed for a 433.92MHz transmitter. All other PLL and VCO peripheral elements are integrated (see Figure 4-2 on page 162).

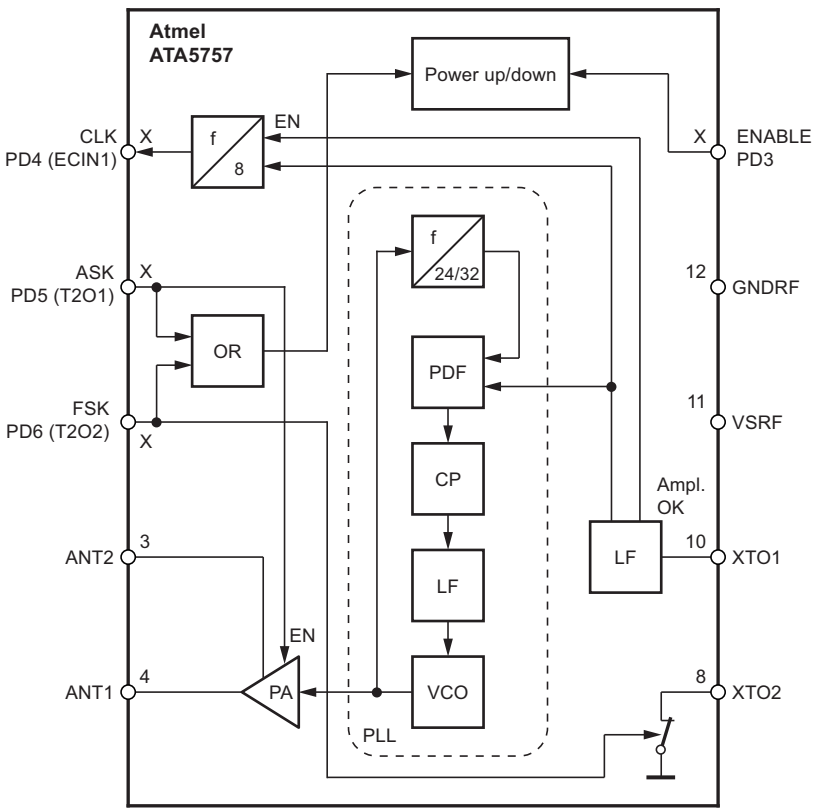
The XTO is a series resonance (current mode) oscillator. Only one capacitor and a crystal connected in series to GND are needed as external elements in an ASK system. The internal FSK switch, together with a second capacitor, can be used for FSK modulation. The crystal oscillator typically needs 0.6ms until the CLK output is activated if a crystal is used as defined in Section 5.3 “Operating Characteristics of Atmel ATA5757” on page 177. For most crystals used in remote control systems, a shorter time outcome results.

The CLK output is switched on if the amplitude of the current flowing through the crystal has reached 35% to 80% of its final value. This is synchronized with the 1.69MHz CLK output. As a result, the first period of the CLK output is always a full period. The PLL is then locked < 250µs after CLK output activation. This means that an additional wait time of ≥ 250µs is necessary before the PA can be switched on and the data transmission can start. This results in a significantly lower time of about 0.85ms between enabling the Atmel ATA5757 and the beginning of the data transmission which saves battery power especially in tire pressure monitoring and remote control systems.

The power amplifier is an open-collector output delivering a current pulse which is nearly independent from the load impedance and therefore the output power can be controlled via the connected load impedance.

This output configuration enables a simple matching to any kind of antenna or to 50Ω. A high power efficiency for the power amplifier can be achieved if an optimized load impedance of $Z_{Load, opt}$ = 280Ω + j310Ω (Atmel ATA5757) at 433.92MHz is used for 3V supply voltage.

Figure 4-2. Block Diagram of Atmel ATA5757



4.6 Functional Description

4.6.1 Atmel ATA5757 Modes

If ASK = Low, FSK = Low and ENABLE = open or low, the circuit is in power-down mode consuming only a very small amount of current so that a lithium cell used as power supply can work for many years.

If the ENABLE pin is left open, power-up of PLL and XTO is the logic OR operation of the ASK and FSK input pins (see Figure 4-2). This means that the IC can be switched on by either the FSK or the ASK input.

If the ENABLE pin is low and ASK or FSK is high, the IC is in idle mode in which the PLL, XTO and power amplifier are off.

With FSK = High and ASK = Low and ENABLE = open or high, the PLL and the XTO are switched on and the power amplifier is off. When the amplitude of the current through the crystal has reached 35% to 80% of its final amplitude, the CLK driver is automatically activated. The CLK output stays low until the CLK driver has been activated. The driver is activated synchronously with the CLK output frequency. Hence the first pulse of the CLK output is a complete period. The PLL is then locked within < 250µs after the CLK driver has been activated, and the transmitter is then ready for data transmission.

With ASK = High the power amplifier is switched on. This is used to perform the ASK modulation and to start FSK modulation (≥ 250 µs after the CLK driver has been activated). During ASK modulation the IC is enabled with the FSK or the ENABLE pin.

With FSK = Low the switch at pin XTO2 is closed, with FSK = High the switch is open. To achieve a faster startup of the crystal oscillator, the FSK pin should be high during startup of the XTO because the series resistance of the resonator seen from pin XTO1 is lower if the switch is off.

The different modes of the Atmel® ATA5757 are listed in Table 4-2 and the corresponding current consumption values can be found in Section 5.3 “Operating Characteristics of Atmel ATA5757” on page 177.

Table 4-2. Atmel ATA5757 Modes

ASK Pin	FSK Pin	ENABLE Pin	Mode
Low	Low	Low/open	Power-down mode, FSK switch high Z
Low	Low	High	Power-up of PLL and XTO, PA off, FSK switch low Z
Low	High	High/open	Power-up of PLL and XTO, PA off, FSK switch high Z
High	Low	High/open	Power-up of PLL and XTO, PA on, FSK switch low Z
High	High	High/open	Power-up of PLL and XTO, PA on, FSK switch high Z
Low/High	High	Low	Idle mode, FSK switch high Z
High	Low/High	Low	Idle mode, ASK switch high Z

4.6.2 CLK Output and Clock Pulse Take-Over by ATA6289

An output CLK signal of 1.69MHz (Atmel ATA5757 operating at 433.92MHz) can be provided for Atmel ATA6289 (PD4, ECIN1).

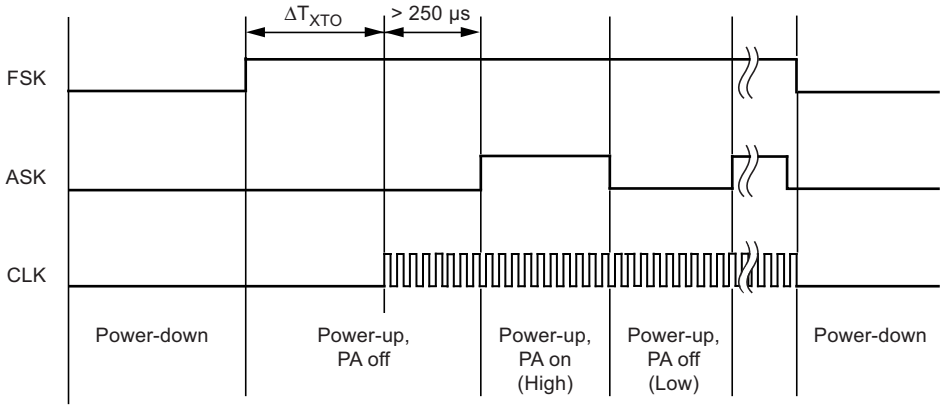
The Atmel ATA6289 can start with its integrated RC oscillator to switch on the Atmel ATA5757. After waiting at least ≥ 0.6ms it can be switched to external clocking (Atmel ATA5757 CLK output) and for the system clock of ATA6289 by setting the corresponding bits in the CMCR register (see also Section 3.7 “Clock Generation” on page 23). After an additional time period of ≥ 250µs the data can be sent via Atmel ATA5757 with crystal accuracy. After the transmission is completed, the Atmel ATA5757 can be set to power-down mode, i.e., no clock at the CLK output. Thus the system clock of the ATA6289 automatically returns to the internal RC oscillator (see also Section 3.7.1.1 “External Clock Monitor” on page 24).

4.6.3 Transmission Modes of ATA5757

4.6.3.1 ASK Mode with ENABLE = Open (PD3 = Tri-State)

The Atmel® ATA5757 is activated by ENABLE = open, FSK = High, ASK = Low. The Atmel ATA6289 can then be switched to external clocking and after typically ΔTXTO = 0.6ms the CLK driver is activated automatically (i.e., the microcontroller must wait until the XTO and CLK are ready.) After another time period of ≤ 250µs, the PLL is locked and ready to transmit. The output power can then be modulated by means of ASK. After transmission the ATA5757 can be switched to power-down mode with ASK = Low and FSK = Low, and the Atmel ATA6289 returns to internal clocking automatically (see also Section 3.7.1.1 “External Clock Monitor” on page 24).

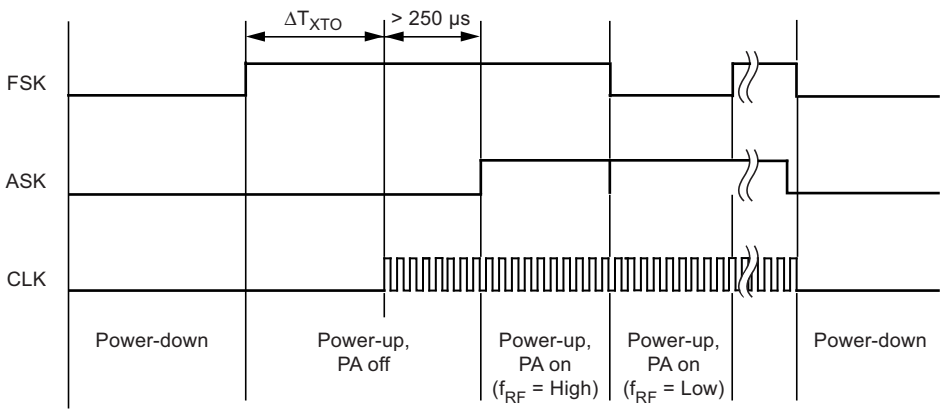
Figure 4-3. Timing ASK Mode with ENABLE = Open (PD3 = Tri-State)



4.6.3.2 FSK Mode with ENABLE = Open (PD3 = Tri-State)

The Atmel ATA5757 is activated by FSK = High, ASK = Low. The Atmel ATA6289 can then be switched to external clocking and after typically ΔTXTO = 0.6ms the CLK driver is activated automatically (i.e., the microcontroller must wait until the XTO and CLK are ready.) After another time period of ≤ 250µs, the PLL is locked and ready to transmit. The power amplifier is switched on with ASK = High. The Atmel ATA5757 is then ready for FSK modulation. The microcontroller starts to switch ON/OFF the capacitor between the crystal load capacitor and GND by means of the FSK pin. This changes the reference frequency of the PLL. If FSK = Low, the output frequency is lower, if FSK = High, the output frequency is higher. After transmission the Atmel ATA5757 can be switched to power-down mode with ASK = Low and FSK = Low, and the ATA6289 returns to internal clocking automatically (see also Section 3.7.1.1 “External Clock Monitor” on page 24).

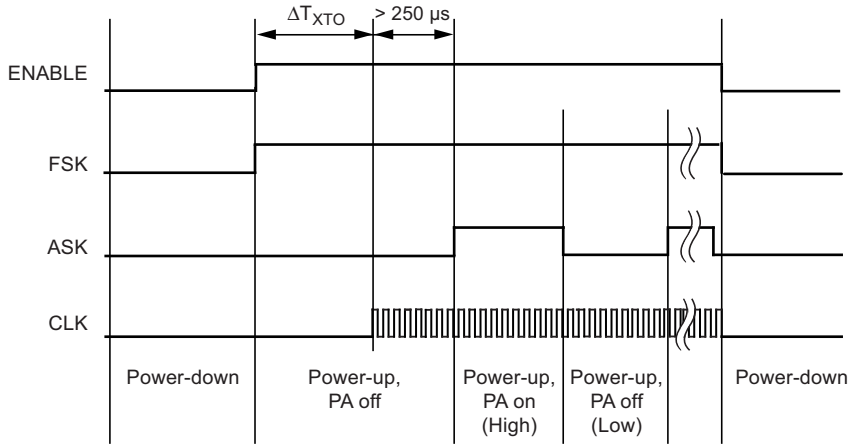
Figure 4-4. Timing FSK Mode with ENABLE = Open (PD3 = Tri-State)



4.6.3.3 ASK Mode with ENABLE = High (PD3 = High)

The Atmel® ATA5757 is activated by ENABLE = High, FSK = High and ASK = Low. After activation the ATA6289 can be switched to external clocking and after typically ΔTXTO = 0.6ms the CLK driver is activated automatically (the microcontroller must wait until the XTO and CLK are ready.) After another time period of ≤ 250µs, the PLL is locked and ready to transmit. The output power can then be modulated by means of the ASK. After transmission the Atmel ATA5757 can be switched to power-down mode with ASK = Low, FSK = Low and ENABLE = Low and the ATA6289 returns to internal clocking automatically (see also Section 3.7.1.1 “External Clock Monitor” on page 24).

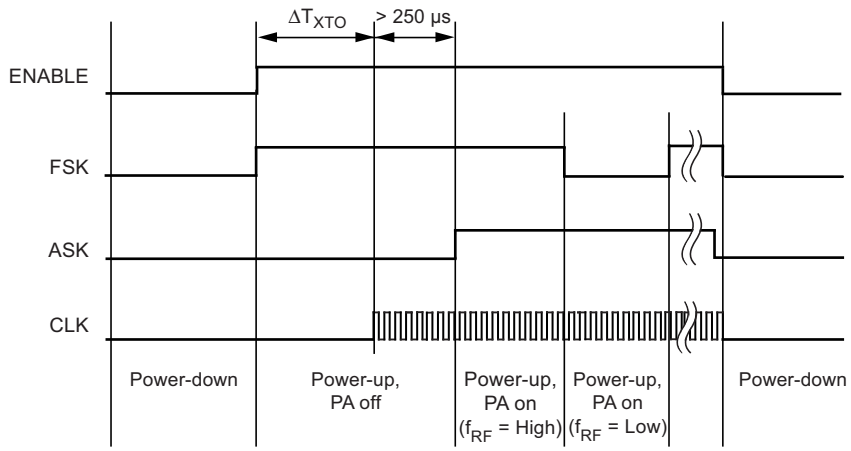
Figure 4-5. Timing ASK Mode with ENABLE = High (PD3 = High)



4.6.3.4 FSK Mode with ENABLE = High (PD3 = High)

The Atmel ATA5757 is activated by ENABLE = High, FSK = High and ASK = Low. The Atmel ATA6289 can be switched to external clocking and after typically ΔTXTO = 0.6ms the CLK driver is activated automatically (i.e., the microcontroller must wait until the XTO and CLK are ready.) After another time period of ≤ 250µs, the PLL is locked and ready to transmit. The power amplifier is switched on with ASK = High. The Atmel ATA5757 is then ready for FSK modulation. The microcontroller starts to switch ON/OFF the capacitor between the crystal load capacitor and GND by means of the FSK pin. This changes the reference frequency of the PLL. If FSK = Low, the output frequency is lower, if FSK = High, the output frequency is higher. After transmission the Atmel ATA5757 can be switched to power-down mode with ASK = Low, FSK = Low and ENABLE = Low and the ATA6289 returns to internal clocking automatically (see also Section 3.7.1.1 “External Clock Monitor” on page 24).

Figure 4-6. Timing FSK Mode with ENABLE = High (PD3 = High)



4.7 Application Circuits

Figure 4-7. ASK Application Circuit

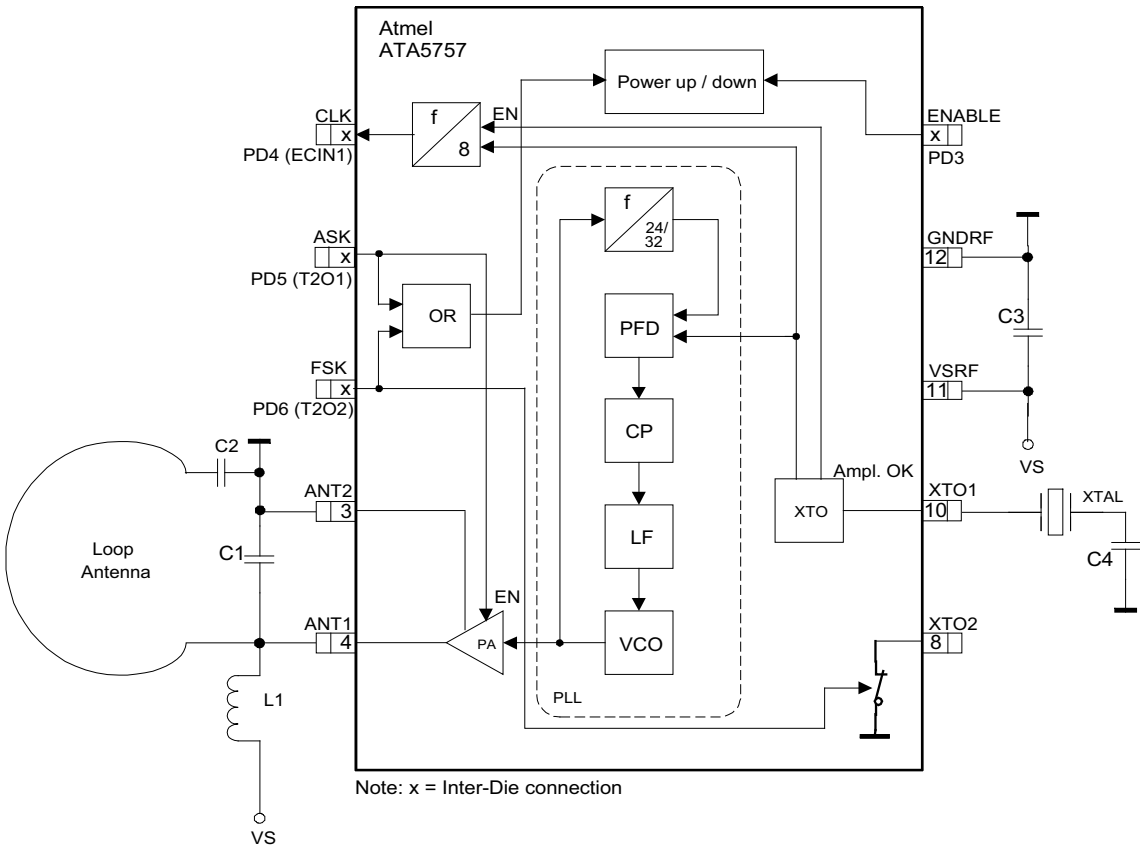
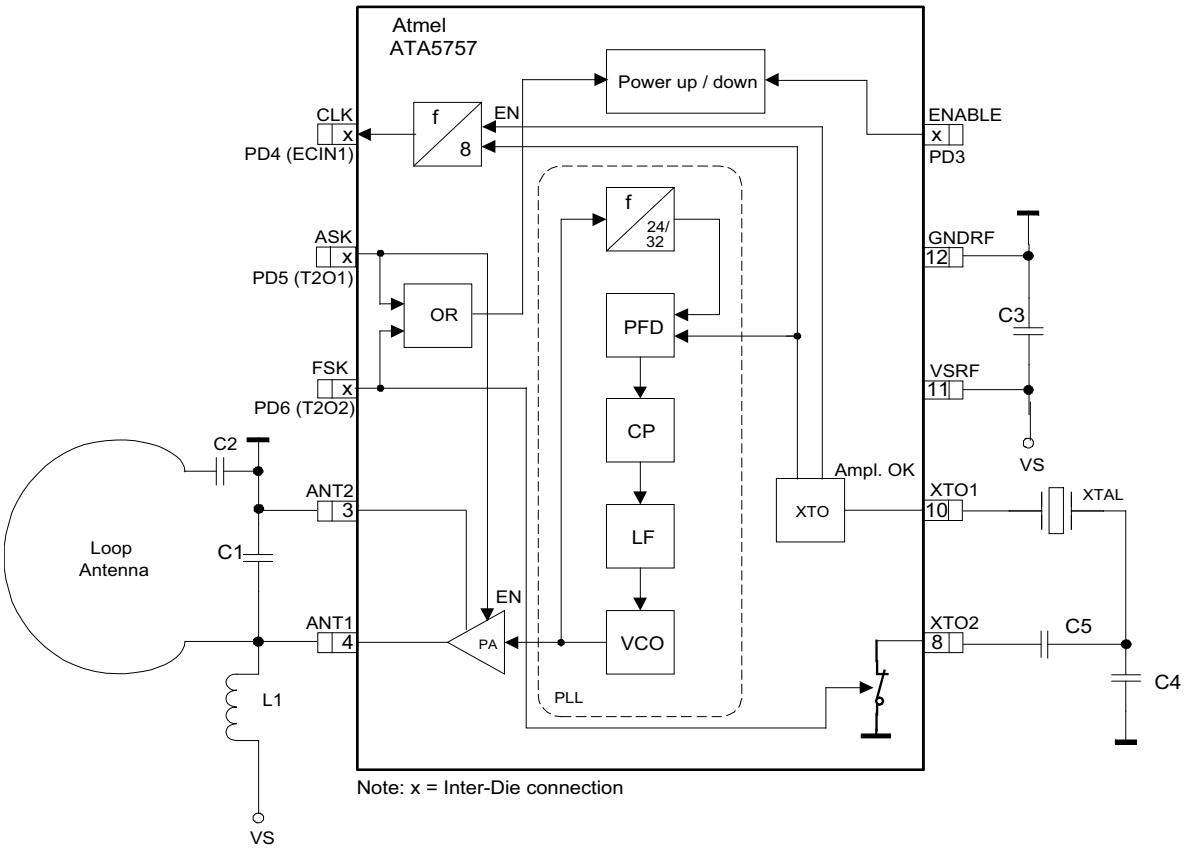


Figure 4-8. FSK Application Circuit



A value of 68nF / X7R is recommended for the supply voltage blocking capacitor C3. C1 and C2 are used to match the loop antenna with the power amplifier. Two capacitors in series should be used for C2 to achieve a better tolerance value and to implement an optimal load impedance $Z_{Load,opt}$ by using capacitors with standard values.

Together with the pins of Atmel® ATA5757 and the PCB board wires, C1 forms a series resonance loop that suppresses the 1st harmonic. Hence the position of C1 on the PCB is important. Normally, the best suppression is achieved when C1 is placed as close as possible to the ANT1 and ANT2 pins. The loop antenna should not exceed a width of 1.5mm, otherwise the Q factor of the loop antenna is too high.

L1 (50nH to 100nH) can be printed on the PCB. C4 should be selected so that the XTO runs on the load resonance frequency of the crystal. Normally, a value of 10pF results in a 12pF load-capacitance crystal due to the board parasitic capacitances and the inductive impedance of the XTO1 pin.

For further information on how to apply Atmel ATA5757 and especially how to get the best values for the external discrete components refer to the complete datasheet which can be found the Atmel website (www.atmel.com).

5. Absolute Maximum Ratings and Operating Characteristics

5.1 Absolute Maximum Ratings

Stresses exceeding those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these or any other conditions exceeding those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

No.	Parameters	Test Conditions	Symbol	Minimum	Typical	Maximum	Unit
Absolute Maximum Ratings of Atmel ATA6286C							
1	Supply voltage		V_{CC}	–0,3		+3.6	V
2	Power dissipation		P_{tot}			200	mW
3	ESD protection at all pins(1)	HBM CDM MM	V_{ESD}		2 750 200		kV V V
4	Junction temperature		T_j			150	°C
5	Storage temperature		T_{stg}	–40		+85	°C
6	Ambient temperature	Max. 15 min	T_{amb}	–40		+85 +170	°C
7	Maximum input current for LF receiver inputs LF1, LF2		$I_{LF1,LF2}$			20	mA
Absolute Maximum Ratings of Atmel ATA5757							
8	Supply voltage		V_{SRF}			5	V
9	Power dissipation		P_{tot}			100	mW
10	Junction temperature		T_j			150	°C
11	Storage temperature		T_{stg}	–40		+125	°C
12	Ambient temperature		T_{amb1}	–40		+85	°C
13	Ambient temperature in power-down mode for 15 minutes without damage	V_{SRF} = 3.2V ENABLE = Low ASK = Low, FSK = Low	T_{amb2}			170	°C

Note: 1. ESD protection for all pins of (ATA6289 and ATA5757)

5.2 Operating Characteristics of Atmel ATA6289

All parameters are referred to GND.
Values characterized for V_{CC} = 2V to 3.6V at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{CC} = 2V and 3.6V at T_{amb} = 85°C unless otherwise specified.
Typical values are characterized for V_{CC} = 3V at T_{amb} = 25°C.

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type[1]
14	Digital I/O							
14.1	Digital input low voltage		VIL	0		$0.2 \times V_{CC}$	V	B
14.2	Digital input high voltage		VIH	$0.8 \times V_{CC}$		V_{CC}	V	B
14.3	Digital input hysteresis		VHYS		450		mV	C
14.4	Digital output low voltage	I_{load} = 2.575mA	VOL			720	mV	A
14.5	Digital output high voltage	I_{load} = 2.575mA	VOH	V_{CC} – 0.72V			V	A
14.6	$\overline{\text{RESET}}$ pin input low voltage		VRL			$0.2 \times V_{CC}$	V	A
14.7	$\overline{\text{RESET}}$ pin input high voltage		VRH	$0.8 \times V_{CC}$			V	A
14.9	Digital input leakage current high level		ILH	–100			nA	A
14.10	Digital input leakage current low level		ILL			100	nA	A
14.11	Digital I/O pull-up resistor		R_{PU}	55	63.5	87	kΩ	A,C
14.12	$\overline{\text{RESET}}$ pin pull-up resistor	V_{RST} = 0.2V V_{RST} = 2.6V	R_{RST}	200 30	367 67	500 100	kΩ	A
15	Internal RC Oscillators							
15.1	Slow RC oscillator calibrated	V_{CC} = 3V	f_{SRC}	81	90	99	kHz	A,C
15.2	Fast RC oscillator calibrated	V_{CC} = 3V	f_{FRC_1MHz} f_{FRC_4MHz}	0.9 3.6	1 4	1.1 4.4	MHz	A,C
16	Voltage Monitoring							
16.1	Voltage monitoring offset error	V_{CC} = 3V VMLS[2..0] = [000] – [101]	VVM	–50		+50	mV	A
16.2	Brown-out detection offset error	V_{CC} = 3V BODLS = 0, 1	BOD	–50		+50	mV	A
16.3	Power-on reset threshold (rising, falling)	V_{CC} = 3V	POT		1.15		V	C
16.4	Time-out delay after reset	see fuse settings	t_{TOUT}		Table 3-6, Table 3-7			D
16.5	Minimum pulse width on $\overline{\text{RESET}}$ pin.		t_{RST}	2			µs	B
16.6	Bandgap voltage calibrated	V_{CC} = 3V	VBG	1.19	1.23	1.27	V	A,C

Notes:
1. Type means: A =100% tested at 85°C, B = 100% correlation tested at 85°C; C = characterized on samples from –40°C to +85°C, typical values at 25°C; D = Design parameter
2. See Section 5.2.1 “Typical Curves” on page 174ff
3. For more details see Section 3.17.3 “LF Receiver Wake-Up Modes for Microcontroller” on page 119ff
4. Calibration must be done by the customer; temperature sensor is not calibrated in production

5.2 Operating Characteristics of Atmel ATA6289 (Continued)

All parameters are referred to GND.
Values characterized for V_{CC} = 2V to 3.6V at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{CC} = 2V and 3.6V at T_{amb} = 85°C unless otherwise specified.
Typical values are characterized for V_{CC} = 3V at T_{amb} = 25°C.

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type[1]
17	Current Consumption in Sleep Modes							
17.1	Power down mode							
17.1.1	Microcontroller sleep, RF transmitter off, LF receiver off, brown-out detection off, Interval timer off	VCC = 3.0V 25°C 85°C	I_{SLEEP}		0.2 0.48	1.5	µA µA	C[2] A
17.1.2	Microcontroller sleep, RF transmitter off, LF receiver off, brown-out detection off, interval timer active	VCC = 3.0V 25°C 85°C	I_{PD1}		0.45 0.67	1.2	µA µA	C[2] A
17.1.3	Microcontroller sleep, RF transmitter off, LF receiver active, brown-out detection off, interval timer off	VCC = 3.0V 25°C 85°C	I_{PD2}		2.7 3.1	6.0	µA	C[2] A
17.2	Sensor noise reduction mode (SNR)							
17.2.1	Supply current (SRC oscillator active, sensor interface active, timers active, BOD, voltage monitor active, temperature shutdown active, PLL off, PA off)	V_{CC} = 3.0V Internal SRC active: f_{SYSCL} = 90kHz 25°C 85°C	I_{SNR_SRC}		12.7 13.4	30.0	µA µA	C A
17.2.2	Supply current (FRC oscillator active, sensor interface active, timers active, BOD, voltage monitor active, temperature shutdown active, PLL off, PA off)	V_{CC} = 3.0V Internal FRC active: f_{SYSCL} = 2MHz 25°C 85°C	I_{SNR_FRC}		255.0 262.0	600.0	µA µA	C A
17.2.3	Supply current (external oscillator, sensor interface active, timers active, BOD, voltage monitor active, temperature shutdown active, PLL off, PA off)	V_{CC} = 3.0V External clock active: f_{ECL} = 2MHz 25°C 85°C	I_{SNR_EXT}		205.0 210.0	555.0	µA µA	C A

Notes:
1. Type means: A =100% tested at 85°C, B = 100% correlation tested at 85°C; C = characterized on samples from –40°C to +85°C, typical values at 25°C; D = Design parameter
2. See Section 5.2.1 “Typical Curves” on page 174ff
3. For more details see Section 3.17.3 “LF Receiver Wake-Up Modes for Microcontroller” on page 119ff
4. Calibration must be done by the customer; temperature sensor is not calibrated in production

5.2 Operating Characteristics of Atmel ATA6289 (Continued)

All parameters are referred to GND.
Values characterized for V_{CC} = 2V to 3.6V at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{CC} = 2V and 3.6V at T_{amb} = 85°C unless otherwise specified.
Typical values are characterized for V_{CC} = 3V at T_{amb} = 25°C.

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type[1]
17.3	Idle mode							
17.3.1	Supply current (microcontroller sleep, all digital modes active, PLL off, PA off)	V_{CC} = 3.0V Internal SRC active: f_{SYSCL} = 90kHz 25°C 85°C	I_{IDLE_SRC}		29.0 30.0	60.0	µA µA	C A
17.3.2	Supply current (microcontroller sleep, all digital modes active, PLL off, PA off)	V_{CC} = 3.0 V Internal FRC active: f_{SYSCL} = 2MHz 25°C 85°C	I_{IDLE_FRC}		620.0 635.0	1000.0	µA µA	C A
17.3.3	Supply current (microcontroller sleep, all digital modes active, PLL off, PA off)	V_{CC} = 3.0 V External clock active: f_{ECL} = 2MHz 25°C 85°C	I_{IDLE_EXT}		560 575	900	µA µA	C A
18	Current Consumption in Active Mode							
18.1	Supply current (microcontroller all digital modes active, PLL off, PA off)	V_{CC} = 3.0V Internal SRC active: f_{SYSCL} = 90kHz 25°C 85°C	I_{ACTIVE_SRC}	n	380 385	600	µA µA	C A
18.2	Supply current (microcontroller all digital modes active, PLL off, PA off)	V_{CC} = 3.0V Internal FRC active: f_{SYSCL} = 2MHz 25°C 85°C	I_{ACTIVE_FRC}		1.55 1.61	2.45	mA mA	C A
18.3	Supply current (microcontroller all digital modes active, PLL off, PA off)	V_{CC} = 3.0V External clock active: f_{ECL} = 2MHz 25°C 85°C	I_{ACTIVE_EXT}		1.45 1.54	2.35	mA mA	C A
19	LF Parameter							
19.1	Power supply current in standby mode	V_{CC} = 3V			see 17.1.3			
19.2	Coil inputs (LF1, LF2)							
19.2.1	Coil input voltage limiter (anti parallel diodes)	V_{CC} = 3V			±0.7		V	D
19.2.2	Coil input current	V_{CC} = 3V	$I_{LF1,LF2}$			2.5	mA	D

Notes:
1. Type means: A =100% tested at 85°C, B = 100% correlation tested at 85°C; C = characterized on samples from –40°C to +85°C, typical values at 25°C; D = Design parameter
2. See Section 5.2.1 “Typical Curves” on page 174ff
3. For more details see Section 3.17.3 “LF Receiver Wake-Up Modes for Microcontroller” on page 119ff
4. Calibration must be done by the customer; temperature sensor is not calibrated in production

5.2 Operating Characteristics of Atmel ATA6289 (Continued)

All parameters are referred to GND.
Values characterized for V_{CC} = 2V to 3.6V at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{CC} = 2V and 3.6V at T_{amb} = 85°C unless otherwise specified.
Typical values are characterized for V_{CC} = 3V at T_{amb} = 25°C.

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type[1]
19.3	Amplifier							
19.3.1	Wake-up sensitivity LFRSS = “0”	V_{CC} = 3V			1.3	2.1	mV_{RMS}	A, C
19.3.2	Wake-up sensitivity LFRSS = “1”	V_{CC} = 3V			5.2	10	mV_{RMS}	A, C
19.3.3	Bandwidth	V_{CC} = 3V Without coil			120		kHz	C
19.3.4	Upper corner frequency	V_{CC} = 3V Without coil			160		kHz	C
19.3.5	Lower corner frequency	V_{CC} = 3V Without coil			40		kHz	C
19.3.6	Input impedance	V_{CC} = 3V f = 125kHz		80 (see also Figure 3-67)			kΩ	D
19.3.7	Input capacitance LFCS[2.0] = 010 (10pF)	V_{CC} = 3V		7.2	8.7	10.8	pF	A,C
19.3.8	Amplifier gain	V_{CC} = 3V		28	32		dB	B,C
19.4	Automatic Gain Control (AGC)							
19.4.1	Preamble detection time	Count. mode (LFBM = 0) Burst mode (LFBM = 1)		128 256			125kHz periods	D
19.4.2	AGC adjustment time (settling time)			128			125kHz periods	D
19.4.3	Signal change rate (LF frontline gap detection)	Count. mode (LFBM = 0) Burst mode (LFBM = 1)		8 24	13 40		125Hz periods	A, C B, C
19.4.4	Data rate (Q < 20)					3.9	Kbaud	D
19.5	LF wake-up of microcontroller							
19.5.1	Start gap detection[3]	Cont. mode (LFBM = 0)		16		204	90kHz periods	D
19.5.2	Start burst detection[3]	Burst mode (LFBM = 1)		48		200	90kHz periods	D
19.5.3	Conversion error of the envelope signal					2	90kHz periods	D

Notes:
1. Type means: A =100% tested at 85°C, B = 100% correlation tested at 85°C; C = characterized on samples from –40°C to +85°C, typical values at 25°C; D = Design parameter
2. See Section 5.2.1 “Typical Curves” on page 174ff
3. For more details see Section 3.17.3 “LF Receiver Wake-Up Modes for Microcontroller” on page 119ff
4. Calibration must be done by the customer; temperature sensor is not calibrated in production

5.2 Operating Characteristics of Atmel ATA6289 (Continued)

All parameters are referred to GND.
Values characterized for V_{CC} = 2V to 3.6V at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{CC} = 2V and 3.6V at T_{amb} = 85°C unless otherwise specified.
Typical values are characterized for V_{CC} = 3V at T_{amb} = 25°C.

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type[1]
20	Capacitance Sensor Unit (CSU)							
20.1	Noise performance	V_{CC} = 3.3V C_{EXT} = 12pF $T_{Measure}$ = 30ms			±2		fF	C[2]
20.2	Accuracy of internal reference capacity	C_{INT} = 16pF		13	16	19	pF	B,C
20.3	External sensor capacity range	Range can be greater but with less accuracy, see values in brackets		3 (2)		16 (25)	pF	D
21	Motion Sensor Unit (MSU)							
21.1	Calibration steps				20		fF	D
21.2	Temperature drift error	C_{EXT} = 3.3pF T_{amb} = –40°C to +85°C			86		fF	C[2]
21.3	V_{CC} drift error	C_{EXT} = 3.3pF V_{CC} = 1.9V to 3.6V			35		fF	C[2]
21.4	External sensor capacity			1		4	pF	C
22	Voltage Sensor Interface							
22.1	Noise performance	V_{CC} = 3.0V Temp = 21°C $T_{Measure}$ = 33ms			±4		mV	C[2]
22.2	Absolute accuracy	Not calibrated V_{CC} = 1.9V V_{CC} = 3.6V			23 35		mV	C
22.3	Voltage sensor range			BOD_{TH}		V_{CCmax}	V	D
23	Temperature Sensor Unit							
23.1	Noise performance	V_{CC} = 3.0V Temp = 21°C $T_{Measure}$ = 33ms		±300			mK	C[2]
23.2	Absolute accuracy	2 point calibration[4] T_{amb} = –40°C to +105°C			3		K	C[2]
23.3	Temperature sensor range			–40		+85	°C	D

Notes:
1. Type means: A =100% tested at 85°C, B = 100% correlation tested at 85°C; C = characterized on samples from –40°C to +85°C, typical values at 25°C; D = Design parameter
2. See Section 5.2.1 “Typical Curves” on page 174ff
3. For more details see Section 3.17.3 “LF Receiver Wake-Up Modes for Microcontroller” on page 119ff
4. Calibration must be done by the customer; temperature sensor is not calibrated in production

5.2.1 Typical Curves

Characterized on samples at V_{CC} = 3V and T_{amb} = 25°C unless otherwise specified.

Figure 5-1. Current I_{sleep} Overtemperature

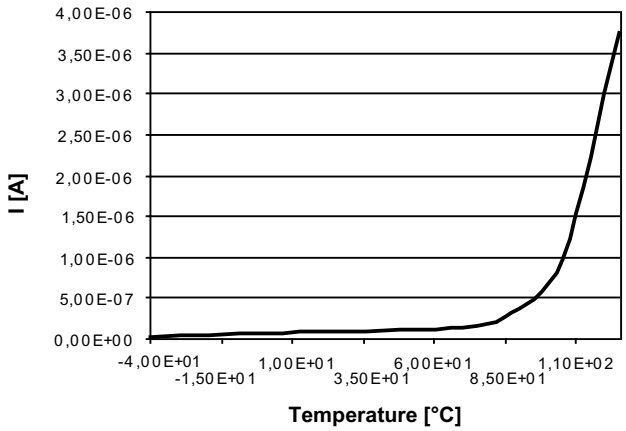


Figure 5-2. Current I_{PD1} Overtemperature

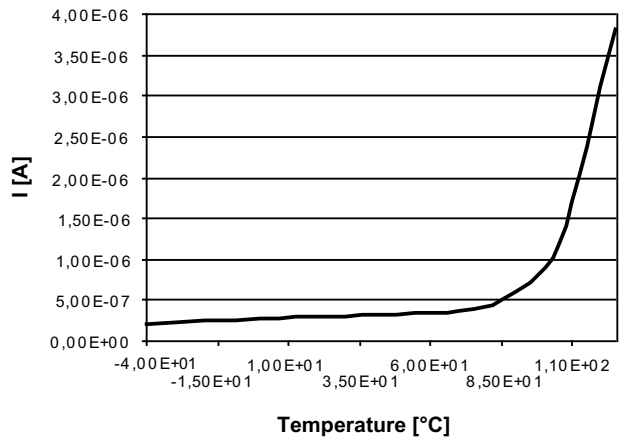


Figure 5-3. Current I_{PD2} Overtemperature

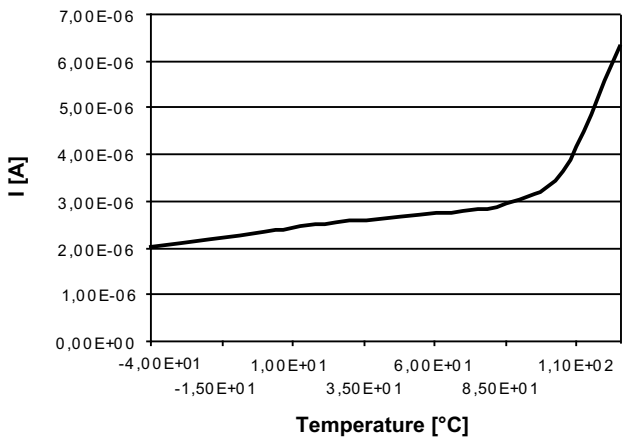


Figure 5-4. Resolution of Capacitance Sensor for C_{EXT} = 12pF, T_{MEAS} = 30ms; V_{CC} = 3.3V

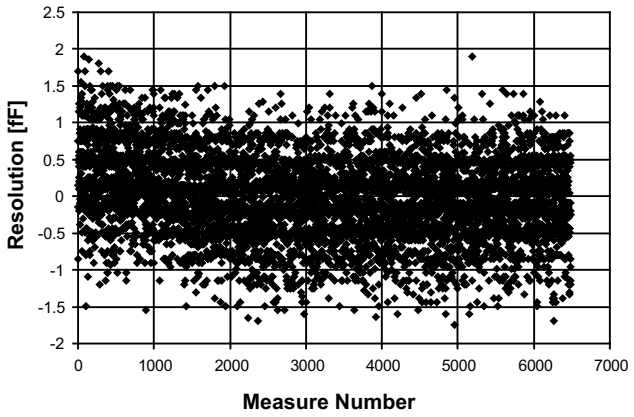


Figure 5-5. Drift Error of Motion Sensor Overtemperature for V_{CC} = 1.9V, 2.0V, 3.6V

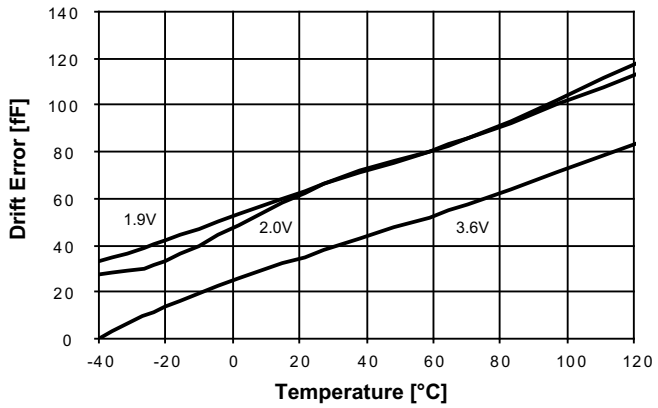


Figure 5-6. Resolution of Temperature Sensor for Temp = 21°C; T_{MEAS} = 33ms; V_{CC} = 3.0V

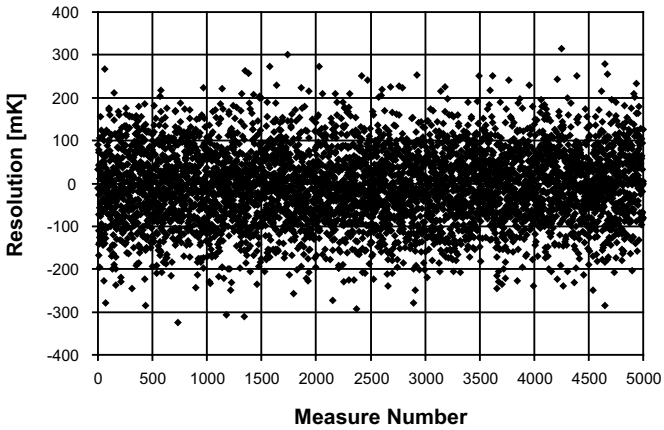


Figure 5-7. Accuracy of Temperature (Two-Point Calibration) for V_{CC} = 1.9V, 3.3V, 4.7V

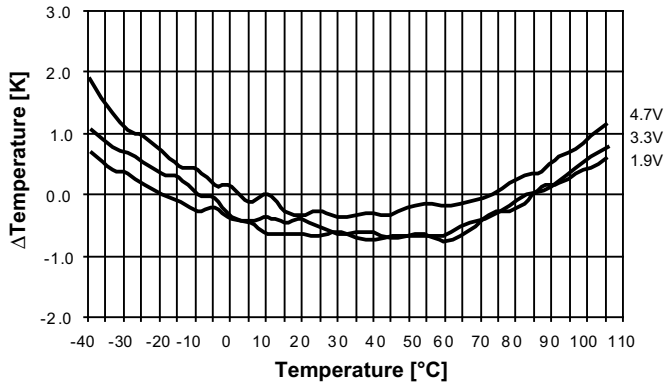
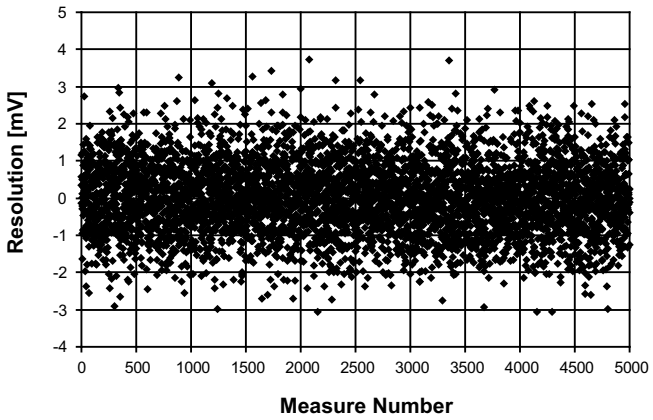


Figure 5-8. Resolution of Voltage Sensor for Temp = 21°C; T_{MEAS} = 33ms; V_{CC} = 3.0V



5.3 Operating Characteristics of Atmel ATA5757

All parameters are referred to GNDRF (pin 12).
Values characterized for V_{SRF} = 2V to 3.6V (pin 11) at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{SRF} = 2V and 3.6V at T_{amb} = 25°C on wafer and T_{amb} = 85°C in final test unless otherwise specified.
Typical values are given for V_{SRF} = 3V and T_{amb} = 25°C.
C_M = 4.37fF, C_0 = 1.3pF, CL_{NOM} = 18pF, C_4 = 10pF, C_5 = 15pF and $R_S \leq 60\Omega$ (these parameters are described in the complete datasheet of Atmel ATA5757 which can be found on the Atmel website at www.atmel.com).

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type(1)
24	Current Consumptions in Sleep, Idle, Active Mode							
24.1	Supply current, power-down mode	ENABLE = Low ASK = Low FSK = Low T_{amb} = 25°C T_{amb} = –40°C to +85°C	I_{S_Off}		1	100 350	nA nA	A A,C
24.2	Supply current, idle mode	ENABLE = Low, ASK, FSK can be Low or High, V_{SRF} = 3.6V	I_{S_IDLE}			100	µA	A(2)
24.3	Supply current, power-up, PA off, FSK switch high Z	ENABLE = High ASK = Low FSK = High V_{SRF} = 3.6V	I_{S_on}		4.1	5.0	mA	A,C
24.4	Supply current, power-up, PA on, FSK switch high Z	ENABLE = High ASK = High FSK = High V_{SRF} = 3.6V, C_{CLK} = 10pF ATA5757	$I_{S\_Transmit1}$		9.2	12.0	mA	A,C
24.5	Supply current, power-up, PA on, FSK low Z	ENABLE = High ASK = High FSK = Low V_{SRF} = 3.6V, C_{CLK} = 10pF ATA5757	$I_{S\_Transmit2}$		9.5	12.3	mA	A,C
25	Power Amplifier (PA)							
25.1	Output power	V_{SRF} = 3.0V, T_{amb} = 25°C, f = 433.92MHz for ATA5757, $Z_{Load, opt}$ = (280 + j310)	P_{Out}	3	5.3	8	dBm	A
25.2	Output power for full temperature and supply voltage range	T_{amb} = –40°C to +85°C, V_{SRF} = 2.0V to 3.6V	P_{Out}	2.5		8.5	dBm	A,C
25.3	Spurious emission	other spurious are lower f_{CLK} = $f_{XT0/8}$ $f_0 \pm f_{CLK}$ $f_0 \pm f_{XT0}$	Spour				dBc	A(3),C
25.4	Harmonics	With 50Ω matching network (see original datasheet) 2nd 3rd			 –16 –15		 dBc dBc	C

*) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

Notes:
1. Type means: A =100% tested at 25°C on wafer and at 85°C in final test; B = 100% correlation tested at 25°C on wafer and at 85°C in final test; C = characterized (or correlation characterized) on samples from –40°C to +85°C and typical values are given for 25°C unless otherwise specified; D = Design parameter
2. Only 100% tested (or 100% correlation tested) on wafer at 25°C
3. Only 100% tested (or 100% correlation tested) in final test at 85°C

5.3 Operating Characteristics of Atmel ATA5757 (Continued)

All parameters are referred to GNDRF (pin 12).
Values characterized for V_{SRF} = 2V to 3.6V (pin 11) at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{SRF} = 2V and 3.6V at T_{amb} = 25°C on wafer and T_{amb} = 85°C in final test unless otherwise specified.
Typical values are given for V_{SRF} = 3V and T_{amb} = 25°C.
C_M = 4.37fF, C_0 = 1.3pF, CL_{NOM} = 18pF, C_4 = 10pF, C_5 = 15pF and $R_S \leq 60\Omega$ (these parameters are described in the complete datasheet of Atmel ATA5757 which can be found on the Atmel website at www.atmel.com).

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type(1)
26	Quartz Oscillator							
26.1	Oscillator frequency XTO (= phase comparator frequency)	$f_{XTO} = f_{0/32}$ ATA5757 f_{XTAL} = resonant frequency of the XTAL, C_M = 4.37fF, load capacitance selected accordingly T_{amb} = –40°C to +85°C	Δf_{XTO}	–50		+50	ppm	A,C
26.2	Imaginary part of XTO1 impedance in steady state oscillation	Because pulling P is $P = -IM_{XTO} = C_M = \pi \times f_{XTO}$ Δf_{XTO} can be calculated out of IM_{XTO} with C_M = 4.37fF	IM_{XTO}	j20	j110	j200	Ω	B,C
26.3	Real part of XTO1 impedance in small signal oscillation	This value is important for crystal oscillator startup	RE_{XTO}	–650	–1100		Ω	B,C
26.4	Crystal oscillator startup time	Time between ENABLE of the IC with FSK = high and activation of the CLK output crystal parameters: C_M = 4.37fF, C_0 = 1.3pF, CL_{NOM} = 18pF, C_4 = 10pF, C_5 = 15pF, $R_S \leq 60\Omega$	ΔT_{XTO}		0.5	2.2	ms	A,C
26.5	XTO drive current	Current flowing through the crystal in steady state oscillation (peak-to-peak value)	I_{XTO1}		215		µApp	A,C
26.6	Series resonance resistance of the resonator seen from pin XTO1	For proper detection of the XTO amplitude	R_{s_max}			150	Ω	D
26.7	Capacitive load at pin XTO1		C_{L_max}			5	pF	D

*) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

Notes:
1. Type means: A =100% tested at 25°C on wafer and at 85°C in final test; B = 100% correlation tested at 25°C on wafer and at 85°C in final test; C = characterized (or correlation characterized) on samples from –40°C to +85°C and typical values are given for 25°C unless otherwise specified; D = Design parameter
2. Only 100% tested (or 100% correlation tested) on wafer at 25°C
3. Only 100% tested (or 100% correlation tested) in final test at 85°C

5.3 Operating Characteristics of Atmel ATA5757 (Continued)

All parameters are referred to GNDRF (pin 12).
Values characterized for V_{SRF} = 2V to 3.6V (pin 11) at T_{amb} = –40°C to +85°C unless otherwise specified.
Values measured for V_{SRF} = 2V and 3.6V at T_{amb} = 25°C on wafer and T_{amb} = 85°C in final test unless otherwise specified.
Typical values are given for V_{SRF} = 3V and T_{amb} = 25°C.
C_M = 4.37fF, C_0 = 1.3pF, CL_{NOM} = 18pF, C_4 = 10pF, C_5 = 15pF and $R_S \leq 60\Omega$ (these parameters are described in the complete datasheet of Atmel ATA5757 which can be found on the Atmel website at www.atmel.com).

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type[1]
27	PLL							
27.1	Locking time of the PLL	Time between the activation of CLK and when the PLL is locked (transmitter ready for data transmission)	ΔT_{PLL}			250	µs	B[3], C
27.2	PLL loop bandwidth		f_{Loop_PLL}		250		kHz	B[3], C
27.3	In loop phase noise PLL	20kHz distance to carrier	L_{PLL}		–84	–76	dBc/Hz	A, C
27.4	Phase noise VCO	at 1MHz at 36MHz	L_{at1M} L_{at36M}		–88 –119	–84 –115	dBc/Hz dBc/Hz	A[3],C B[3], C
27.5	Frequency range of VCO	ATA5757	f_{VCO}	432		448	MHz	A[3], C
27.6	Clock output frequency	ATA5757	f_{CLK}		f_0/256		MHz	D
28	Modulation							
28.1	FSK modulation frequency rate	This corresponds to 20Kbaud in Manchester coding and 40Kbaud in NRZ coding	f_{MOD_FSK}	0		20	kHz	B[3], C
28.2	FSK switch OFF resistance	High Z	R_{SWIT_OFF}	50			kΩ	A, C
28.3	FSK switch OFF capacitance	High Z capacitance	C_{SWIT_OFF}	0.75	0.9	1.1	pF	C
28.4	FSK switch ON resistance	Low Z	R_{SWIT_ON}		130	175	Ω	A, C
28.5	ASK modulation frequency rate	Duty cycle of the modulation signal = 50%, this corresponds to 20Kbaud in Manchester coding and 40Kbaud in NRZ coding	f_{MOD_ASK}	0		20	kHz	C

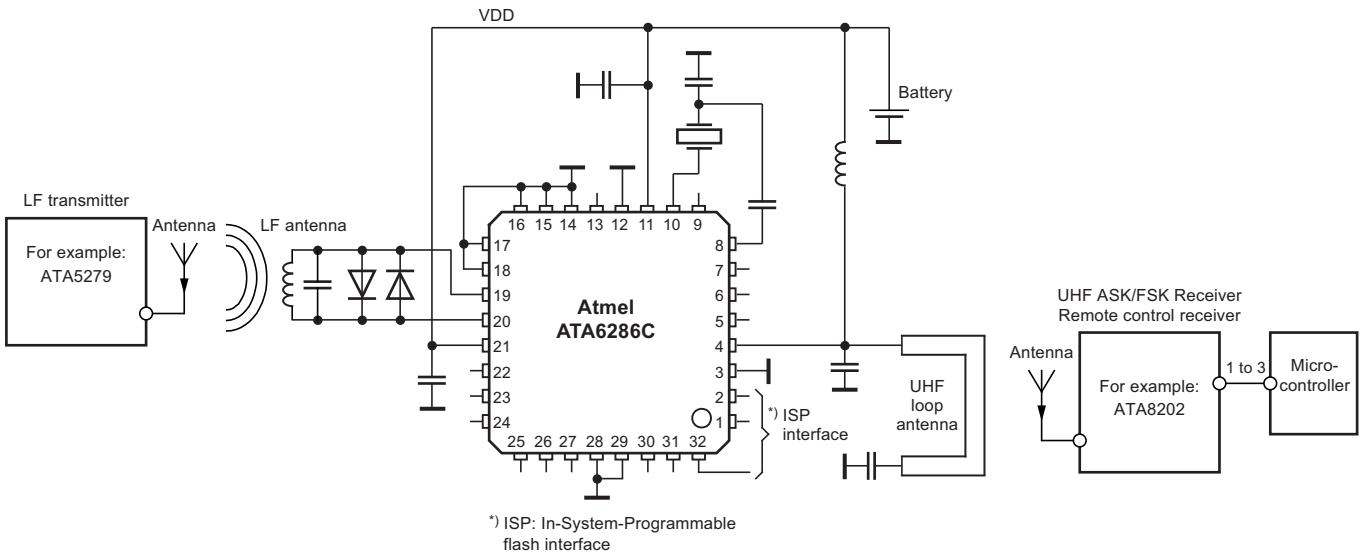
*) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

Notes:
1. Type means: A =100% tested at 25°C on wafer and at 85°C in final test; B = 100% correlation tested at 25°C on wafer and at 85°C in final test; C = characterized (or correlation characterized) on samples from –40°C to +85°C and typical values are given for 25°C unless otherwise specified; D = Design parameter
2. Only 100% tested (or 100% correlation tested) on wafer at 25°C
3. Only 100% tested (or 100% correlation tested) in final test at 85°C

6. Application Examples

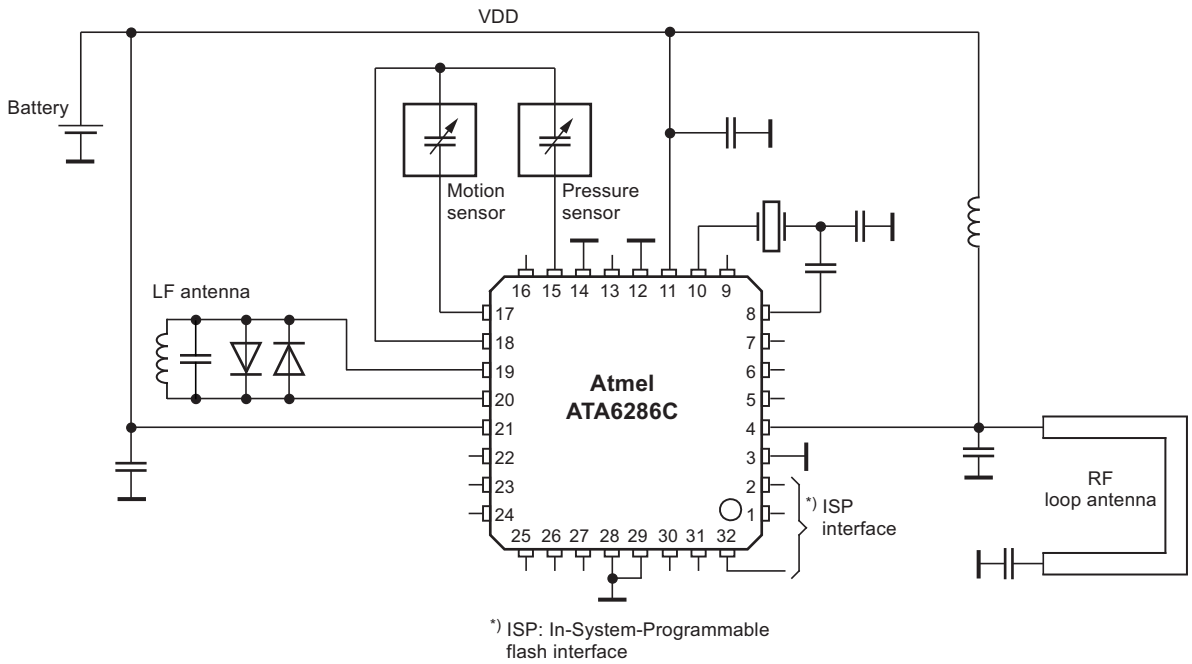
6.1 Active RFID System

Figure 6-1. Active RFID System



6.2 Pressure/Motion Sensor System

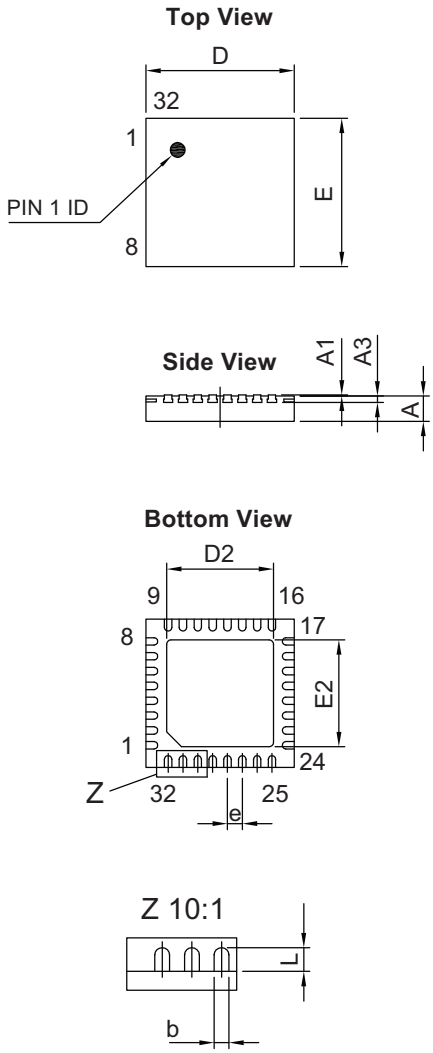
Figure 6-2. Pressure/Motion Sensor System



7. Ordering Information

Extended Type Number	Package	Frequency	MOQ	Remarks
ATA6286C-PNQW-1	QFN32	433MHz	6,000	Taped and reeled

8. Package Information



technical drawings according to DIN specifications

Dimensions in mm

COMMON DIMENSIONS (Unit of Measure = mm)				
Symbol	**MIN**	**NOM**	**MAX**	**NOTE**
A	0.8	0.85	0.9	
A1	0	0.035	0.05	
A3	0.16	0.21	0.26	
D	4.9	5	5.1	
D2	3.5	3.6	3.7	
E	4.9	5	5.1	
E2	3.5	3.6	3.7	
L	0.35	0.4	0.45	
b	0.2	0.25	0.3	
e		0.5		

05/20/14

	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact:** packagedrawings@atmel.com	**Package: QFN_5x5_32L** Exposed pad 3.6x3.6		6.543-5203.01-4	1

9. Revision History

Please note that the following page numbers referred to in this section refer to the specific revision mentioned, not to this document.

Revision No.	History
9308C-RFID-09/14	• Section 7 “Ordering Information” on page 181 updated • Section 8 “Package Information” on page 181 updated
9308B-RFID-07/14	• Put datasheet in the latest template

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**