GPIOR3 – General Purpose I/O Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	GPIOR3
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	

GPIOR4 – General Purpose I/O Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	GPIOR4
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	

GPIOR5 – General Purpose I/O Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	GPIOR5
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	

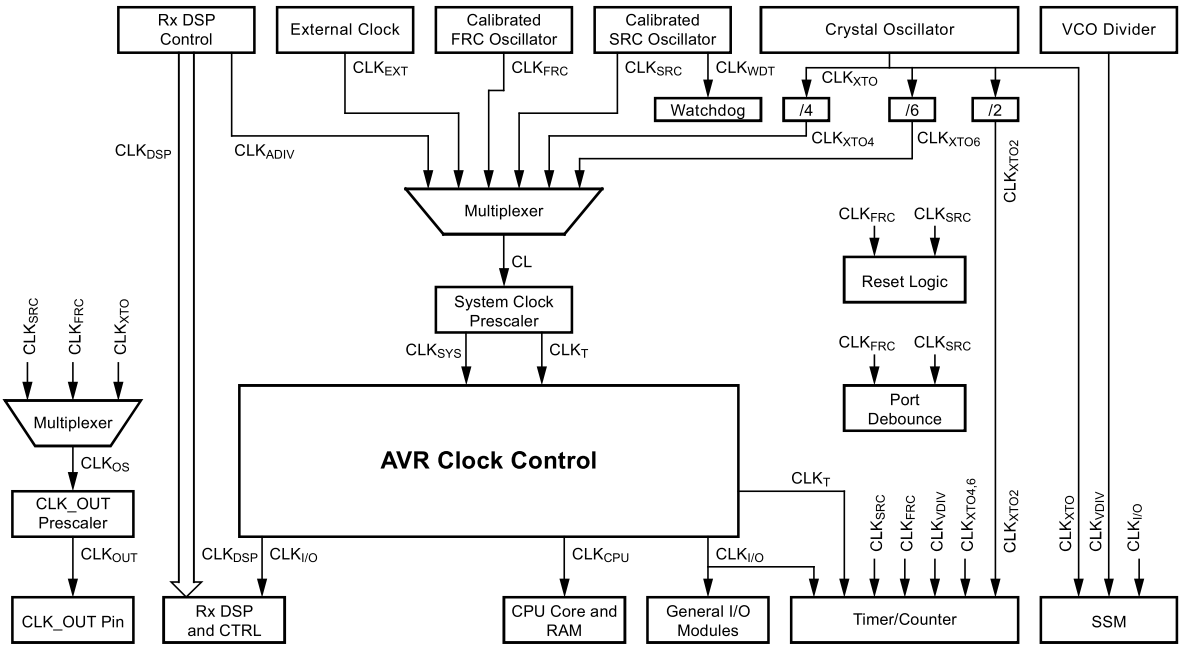
GPIOR6 – General Purpose I/O Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	GPIOR6
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	

3.8.4 System Clock and Clock Options

Figure 3-32 presents the principal clock systems in the AVR® and their distribution. Not all of the clocks need to be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be stopped by using different sleep modes, as described in Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210. Fine-grained clock gating can be performed by using the power reduction registers (PRR0, PRR1 and PRR2). Additional power reduction bits are available for the Rx DSP (Section “RDPR – Rx DSP Power Reduction Register” on page 146).

Figure 3-32. AVR Clock Systems



The AVR clock management supports the switching of the AVR clock source during operation. This allows for a start-up with the fast RC oscillator (CLK_{FRC}) and then switching to the more accurate XTO based clock (CLK_{XTO4}, CLK_{XTO6}) used as the system clock (CLK_{SYS}).

3.8.4.1 Clock Sources

Slow RC Oscillator – CLK_{SRC}

The factory-calibrated internal SRC oscillator is an ultra-low-power oscillator providing a slow clock with 125kHz (typical). It is designed mainly as a:

- Watchdog timer reference clock. The watchdog/interval timer can work in all sleep modes.
- Polling cycle reference for reducing power consumption.

The SRC oscillator requires a short settling time. An internal circuitry enables the clock output only when the oscillator works within the specification limits. The overall accuracy after factory calibration is approximately ±10% over voltage and temperature range (see Table "Electrical Characteristics" on page 300 no. 14.20).

It is not recommended to use the SRC as a system clock during UHF receiver operation. The clock is not fast enough for processing and could lead to reduced sensitivity due to harmonics interfering with the received signal. Measurements show that sensitivity degradation due to clock harmonics does not occur with a correct PCB design and if the CLK_OUT pin is switched off, but it is more secure to use the XTO as clock source during reception.

Fast RC Oscillator – CLK_{FRC}

The FRC oscillator is a low-power RC oscillator with a nominal output frequency of 6.36MHz. It is used for:

- System start-up after reset where CLK_{FRC} is used as system clock

The FRC oscillator requires a short settling time. An internal circuitry enables the clock output only when the oscillator works within specification limits. The overall accuracy after factory calibration is approximately ± 5% (see Table "Electrical Characteristics" on page 300 No 14.30).

It is not recommended to use the FRC as a system clock during UHF receiver operation. This could lead to reduced sensitivity due to harmonics interfering with the received signal. Measurements show that sensitivity degradation due to clock harmonics does not occur with a correct PCB design and if the CLK_OUT pin is switched off, but it is more secure to use the XTO as clock source during reception.

FRC Calibration

The tolerance of the FRC oscillator frequency is ±5% over voltage, temperature range and process tolerances.

To enable using the FRC oscillator for applications requiring ±2% tolerance, the oscillator can be calibrated in-system by the Atmel firmware in the following cases:

- After start-up from OFF mode
- At regular intervals during the system self check and calibration procedure, as described in Section 2.9.1 "TCMode" on page 50.

The FRC calibration must be enabled in the SRAM configuration.

Polling Cycle Calibration Using SRC

The polling cycle can be calibrated to achieve a higher accuracy than the reference clock by correcting the timer compare value. For more details see Section 2.9.2 "Polling Cycle/SRC Calibration" on page 51.

External Clock – CLK_{EXT}

An external clock can be selected as a clock source for the CPU for debugging and testing purposes.

If an external clock is used as system clock, an internal clock monitor circuitry can be activated by setting CMCR.CMONEN to "1". If the external clock fails for a certain period of 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 FRC oscillator as system clock by switching the CCS bit to zero.

The external clock monitor circuitry uses the internal SRC oscillator (125kHz) as clock source for a 4-bit timer. If the external clock does not reset the internal 4-bit timer periodically, a counter value is reached which triggers the external clock fail bit (ECF).

A typical time value for the external clock fail detection is 100µs. Therefore, the minimum external clock frequency is limited to typically 10kHz if external clock monitoring is enabled. An external frequency <10kHz forces a clock fail reset.

Crystal Oscillator – $CLK_{XTO,2,4,6}$

A high-accuracy crystal oscillator is available in the UHF receiver which can be used as a reference for the system clock and the clock output. The XTO frequency divided by 4 and by 6 can be used as a system clock for the AVR®. The XTO divided by 2 (CLK_{XTO2}) can be used only as a reference for Timer3. The undivided CLK_{XTO} can be used only as reference frequency for the clock output pin (CLK_OUT) divider and in the VCO tuning state machine. The CLK_OUT divider has to be programmed to ensure that the frequency at CLK_OUT is below 4.5MHz (see Table "Electrical Characteristics" on page 300 no. 15.90).

Using an XTO clock divided by 8 or more as a system clock is not recommended because doing so can lead to reduced sensitivity of the receiver due to harmonic disturbances. Measurements show that such a sensitivity degradation due to clock harmonics does not occur with a correct PCB design and if the CLK_OUT pin is switched off, but it is more secure to use a frequency above XTO/8 as source during reception. The crystal oscillator in the RF front end as well as the AVCC voltage have to be enabled for this clock to be active.

A detailed description of the XTO can be found in Section 3.2 "Crystal Oscillator" on page 90.

Divided ADC Clock – CLK_{ADIV}

A clock derived from the ADC sampling frequency is provided to be used as a system clock during receiver operation. This keeps harmonics of the system clock from being able to interfere with the received signal.

The RF front end has to be configured to provide a clock for the ADC. The fractional-N-PLL, the ADC, the Rx DSP block and the ADC clock output have to be enabled (RDCR register) for this clock to be active.

Divided VCO Clock – CLK_{VDIV}

The divided VCO clock CLK_{VDIV} is a fast running clock at up to 26.2MHz. It is used to calibrate the VCO frequency by counting the differences versus the CLK_{XTO}.

3.8.4.2 Clock Domains

CLK_{CPU} – CPU Clock

The CPU clock is routed to parts of the system concerned with the operation of the AVR core. Examples of such modules are the general purpose register file, the status register and the data memory holding the stack pointer. Stopping the CPU clock prevents the core from performing general operations and calculations.

$CLK_{I/O}$ – I/O Clock

Like the timers, the I/O clock is used by most I/O modules. The I/O clock is also used by the external interrupt module but some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is stopped (e.g., in sleep mode).

CLK_T – Timer Clock

This clock is derived from the same source as the system clock. Further details on the clock and scaling factor selection can be found in the Section 3.8.4.3 "Clock Switching" on page 204.

CLK_{DSP} – Rx DSP Clocks

The clocks used for the Rx are generated in the internal Rx DSP clock control module. They are configured implicitly by the Rx settings.

CLK_{OUT} – Clock Output

The CLK_{OUT} clock is used to provide a clock reference for external circuitry. The clock output can be derived from three clock sources. The selected clock can be divided by a prescaler and passed to the PB0 pin. Care has to be taken that no configuration is used that leads to an output clock of more than 4.5MHz (see Table "Electrical Characteristics" on page 300 no. 15.90).

Details about the clock source selection can be found in the CLKOCR register description on 205. The output divider settings are described in Section "CLKOD – Clock Output Divider" on page 205.

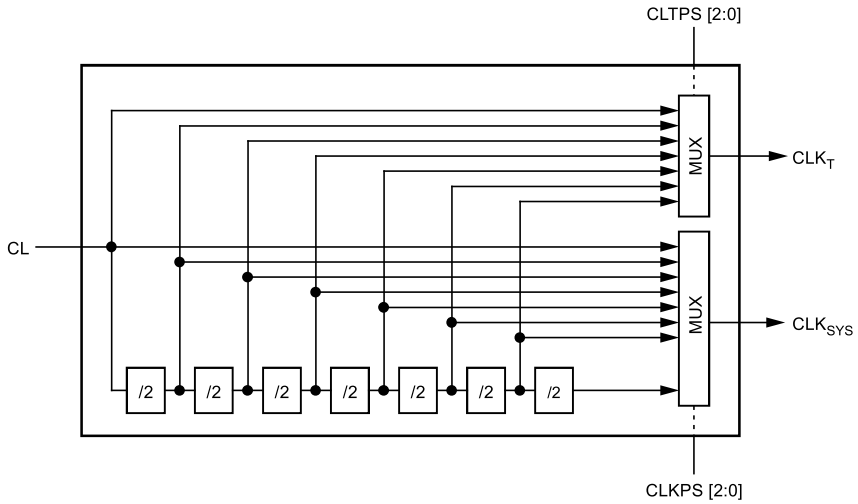
3.8.4.3 Clock Switching

The system start-up is done using the fast RC oscillator. Afterwards the desired clock source can be selected during operation by setting the CMCR register accordingly.

System Clock Prescaler

The system clock prescaler can be used to decrease the system clock frequency and the power consumption when the requirement for processing power are low. The input clock (CL) for the prescaler is selectable via CMCR as described in Section "CMCR – System Clock Management Control Register " on page 207. It affects the clock frequency of the CPU and all synchronous peripherals. $CLK_{I/O}$ and CLK_{CPU} are divided as selected in the CLPR register Section "CLPR – Clock Prescaler Register " on page 209.

Figure 3-33. System Clock Prescaler



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 is higher than 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 clock frequency of the CPU. Hence, it is not possible to determine the state of the prescaler and the exact time it takes to switch from one clock division to the other. To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLKPS[2:0] bits:

1. Write the clock prescaler change enable (CLPCE) bit to one and all other bits in CLPRF to zero.
2. Within four cycles, write the desired value to CLKPS[2:0] while writing a zero to CLPCE.

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

Table 3-43 gives an overview of the default clock sources that are used for each operating mode.

Table 3-43. Main Clock Sources for Operating Mode

Mode	Clock Source
OFFMode	-
IDLEMode(RC)	FRC
IDLEMode(XTO)	XTO/4
PollingMode - Active period - Sleep period	 XTO/4 SRC (AVR in power-down mode!)
RXMode	XTO/4

3.8.4.4 System Clock Register Description

CLKOCR – Clock Output Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	CLKOEN	CLKOS[1:0]		CLKOCR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 2: CLKOEN – Enable Clock Output Driver to CLK_OUT

The corresponding data direction register of the port has to be set as output to enable the driver. The CLK_OUT prescaler settings (CLKOCR.CLKOS and CLKOD) can be modified only if the clock output is disabled (CLKOCR.CLKOEN=0).

Bit 1..0: CLKOS[1:0] – Clock Output Source

Selects the reference clock for the CLK_OUT divider, as described in Table 3-44. This clock output source can be modified only if the clock output is disabled (CLKOCR.CLKOEN=0).

Table 3-44. CLKOS - Clock Output Clock Source Selection

CLKOS[1:0]		Clock Source
0	0	CLK_{SRC}
0	1	CLK_{FRC}
1	0	Reserved
1	1	CLK_{XTO}

CLKOD – Clock Output Divider

It can be modified only if the clock output is disabled (CLKOCR.CLKOEN=0).

Bit	7	6	5	4	3	2	1	0	
	CLKOD[7:0]								CLKOD
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: CLKOD – Clock Output Divider

Divide the selected clock reference by this factor.

The selected clock CLKOS is passed to the CLK_OUT prescaler and the resulting divided clock is visible at the CLK_OUT pin if the output is enabled. See also Figure 3-32 on page 201.

$$fCLK\_OUT = \frac{fCLK_{OS}}{2 \times CLKOD} \qquad (41)$$

CLKOD ∈ {0 .. 255}

If CLKOD=0 is selected, the clock source is passed directly to the pin. The clock output frequency should never exceed the maximum specified frequency (see Section 5. “Electrical Characteristics” on page 300 no. 15.90). For this reason, the divider must stay within the boundaries specified in Table 3-45 on page 206.

Table 3-45. CLKOD – Divider Ranges Depending on Clock Source

Selected CLK Source	Min. CLKOD	Max. CLKOD
CLK_{SRC}	0	255
CLK_{FRC}	1	255
CLK_{XTO}	3	255

For further details see Section 5.7 “I/O Characteristics for Ports PB0 to PB7 and PC0 to PC5” on page 308.

SRCCAL – Slow RC Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0	
	SRCTC[1:0]		SRCCAL[5:0]						SRCCAL
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: SRCTC – Slow RC Oscillator Temperature Compensation

These bits show the temperature compensation value stored during manufacturing. This value is not meant to be changed by the user.

Bit 5..0: SRCCAL – Slow RC Oscillator Calibration

Factory calibration data is written into this register during start-up.

The lowest frequency of the SRC oscillator is available when the calibration registers are in the reset state (0).

SRC parameters can be found in the table “Oscillators and CLK_OUT” on page 307. For this register only the factory calibration values should be used because the Atmel® ATA5785 is not production tested for all other settings of this register.

FRCCAL – Fast RC Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0	
	–	–	FRCTC	FRCCAL[4:0]					FRCCAL
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..6: Reserved Bits

These bits are reserved and read as zero.

Bit 5: FRCTC – Fast RC Oscillator Temperature Compensation

This bit shows the temperature compensation state which has been stored during manufacturing. This value is not meant to be changed by the user.

Bits 4..0: FRCCAL – Fast RC Oscillator Calibration

Factory calibration data is written into this register during start-up. The lowest frequency of the FRC oscillator is available when the calibration registers are in the reset state (0). FRC parameters can be found in the table “Oscillators and CLK_OUT” on page 307. The FRCCAL register should only be modified by the Atmel firmware. The tuning range during operation is limited to ±8 steps from the factory calibration value. All other values are not tested.

CMCR – System Clock Management Control Register

Bit	7	6	5	4	3	2	1	0	
	CMCCE	CMONEN	–	SRCD	CCS	CMM[2:0]			CMCR
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: CMCCE – Clock Management Control Change Enable

The CMCCE bit must be written to logic one to enable the change of the CMCR bits. The CMCCE bit is only updated when the other bits in CMCR are simultaneously written to zero. CMCCE is cleared by hardware four cycles after it is written or when CLPR.CLKPS[2:0] 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: CMONEN – Clock Monitor Enable

This bit controls the clock monitoring of the external clock (CLK_{EXT}). The CMONEN bit must be written to logic one to enable the clock monitoring, if the CMONEN bit is written to logic zero, clock monitoring is always disabled.

Bit 5: Reserved Bit

This bit is reserved and reads as zero.

Bit 4: SRCD – Slow RC Oscillator Disable

Allows the SRC oscillator to stop if it is not used.

SRCD	Description
0	The SRC oscillator is running.
1	The SRC oscillator is stopped if the watchdog is disabled and no other module uses CLK_{SRC}

Bit 3: CCS – Core Clock Select

This bit selects between 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[2:0] bits. CMM[2:0] cannot be modified in this mode. If the CCS bit is written to logic zero, the FRC oscillator clock is enabled and CMM[2:0] can be modified. If the CMM[2:0] bits in CMCR are not changing simultaneously, the CCS bit can only be set to one. After an external clock fail detection the CCS bit is written to zero.

Table 3-46. Core Clock Select Bit

CSS	Description
0	The FRC oscillator is the active clock source and generates the input clock (CL). CMM[2:0] can be modified
1	The clock source selected by CMM[2:0] is active. Possible sources are: CLK_{EXT}, CLK_{SRC}, CLK_{XTO4}, CLK_{XTO6},CLK_{ADIV}. Modification of CMM[2:0] is not possible.

Bit 2..0: CMM – Clock Management Mode

Table 3-47. Clock Source for System Clock Prescaler

Clock Source for System Clock Prescaler (CL)				
CMM[2:0]			CCS = 0	CCS = 1
0	0	0	FRC	CLK_{SRC}
0	0	1	FRC	CLK_{ADIV}
0	1	0	FRC	CLK_{EXT}
0	1	1	FRC	CLK_{XTO6}
1	0	0	FRC	CLK_{XTO4}
1	0	1	FRC	CLK_{XTO4}
1	1	0	FRC	CLK_{XTO4}
1	1	1	FRC	CLK_{XTO4}

CMOCR – Clock Management Override Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	SRCACT	FRCACT	SRCAO	FRCAO	CMOCR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..4: Reserved Bits

These bits are reserved and read as zero.

Bit 3: SRCACT – SRC Oscillator Active

This bit indicates that the SRC oscillator is enabled and active if read as “1”. If this bit is read as “0” the SRC is either disabled or enabled but not active.

Bit 2: FRCACT – FRC Oscillator Active

This bit indicates that the FRC oscillator is enabled and active if read as “1”. If this bit is read as “0” the FRC is either disabled or enabled but not active.

Bit 1: SRCAO – SRC Oscillator Always On

If set to one, this bit enables the SRC oscillator independently of the enabled hardware features.

Bit 0: FRCAO – FRC Oscillator Always On

If set to one, this bit enables the FRC oscillator independently of the enabled hardware features.

CMSR – Clock Management Status Register

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..1: Reserved Bits

These bits are reserved and read as zero.

Bit 0: ECF – External Clock Fail

This bit is set if the clock monitoring circuit detects a breakdown of the external input clock (CLK_{EXT}). 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.

CMIMR – Clock Management Interrupt Mask Register

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..1: Reserved Bits

These bits are reserved and read as zero

Bit 0: ECIE – External Clock Interrupt Enable

Writing ECIE to one enables the clock monitoring interrupt vector if the I bit in SREG is set. The corresponding interrupt vector is executed when the CMSR.ECF flag is set. Writing ECIE to zero disables the interrupt.

CLPR – Clock Prescaler Register

Bit	7	6	5	4	3	2	1	0	
	CLPCE	–	CLTPS[2:0]			CLKPS[2:0]			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

The CLPCE bit must be written to logic one to enable a change of the CLTPS[2:0] and CLKPS[2:0] bits. The CLPCE bit is only updated when the other bits in CLPR 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 neither extends the time-out period nor clears the CLPCE bit.

Bit 6: Reserved Bit

This bit is reserved and reads as zero.

Bits 5..3: CLTPS[2:0] – Timer Clock Prescaler Select

These bits select the division factor for the timer clock (CLK_T) of the system clock prescaler, as shown in Figure 3-33 on page 204.

Table 3-48. CLTPS – Timer Clock Prescaler Select

CLTPS[2:0]			Division Factor
0	0	0	disabled (reset value)
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: CLKPS[2:0] – System Clock Prescaler Select

These bits select the division factor for the system clock (CLK_{SYS}) output as shown in Figure 3-33 on page 204.

Table 3-49. CLKPS – System Clock Prescaler Select

CLKPS[2:0]			System Clock Division Factor
0	0	0	1(reset value)
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

3.8.5 Sleep Modes and Active Power Reduction

Sleep modes enable the application to shut down unused modules in the MCU to save power. The AVR® provides various sleep modes.

To enter any of the sleep modes, the SE bit in SMCR must be written to logic one and a SLEEP instruction must be executed. The SM[2:0] bits in the SMCR register select which sleep mode (idle, extended power save power down, power save) is activated by the AVR SLEEP instruction. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then stopped for four cycles in addition to the start-up time, executes the interrupt routine and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM are not altered when the device wakes up. If a reset occurs during sleep mode, the MCU wakes up and executes from the reset vector. Figure 3-32 on page 201 presents an overview of the clocks and their distribution. Table 3-50 on page 211 shows the available sleep modes with the corresponding active clock domains and wake-up sources.

Idle Mode

When the SM[2:0] bits are written to “0b000”, the SLEEP instruction makes the MCU enter the idle mode, stopping the CPU but allowing the peripherals, e.g., SPI, Rx DSP, timer and the interrupt system to continue operating. This sleep mode basically stops CLK_{CPU} 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 the timer overflow interrupt or a receive buffer full interrupt of the on-chip digital data demodulator.

Extended Power-Save Mode

When the SM[2:0] bits are written to “0b001”, the SLEEP instruction makes the MCU enter the extended power-save mode, stopping the CPU and the peripherals on the I/O bus but allowing the SPI, Rx DSP, timer, watchdog and the interrupt system to continue operating. This sleep mode basically stops CLK_{CPU} and CLK_{IO} while allowing the other clocks to run.

The extended power-save mode enables the MCU to wake up from external triggered interrupts as well as internal interrupts such as the timer overflow interrupt or a receive buffer full interrupt of the on-chip digital data demodulator.

Power-Save Mode

When the SM[2:0] bits are written to “0b011”, the SLEEP instruction makes the MCU enter the power-save mode. In this mode, the FRC oscillator and the external input clock are stopped, while the external interrupts and the watchdog continues operation (if enabled). Only an external reset, a watchdog reset, Rx DSP interrupts, an external level interrupt, or a pin change interrupt can wake up the MCU. This sleep mode basically stops all generated clocks, allowing operation of asynchronous modules only.

Power Down Mode

When the SM[2:0] bits are written to “0b010”, the SLEEP instruction makes the MCU enter power-down mode, stopping the CPU and the peripherals on the I/O bus but allowing the Rx DSP interrupts, timer and the asynchronous interrupts to continue operating. This sleep mode basically stops all generated clocks, allowing operation of asynchronous modules only.

Table 3-50. Sleep Modes: Active Clock Domains and Wake-Up Sources

	Active Clock Domains			Oscillators and External Clocks							Wake-Up Sources								
Sleep Mode	CLK_{CPU}	$CLK_{I/O}$	CLK_T	External Clock	CLK_{FRC}	CLK_{SRC}	CLK_{WDT}	$CLK_{XTO2,4,6}$	CLK_{ADIV}	CLK_{VDIV}	INT0/1 and Pin Change	UHF-Receiver	SPI	SSM	Supply	Ext. input clock monitor	Timer0	Timer1/2/3/4	Timer5
IDLE		X	X	X	X	X	X	X (1)	X	(2)	X	X	X	X	X	X	X	X	X
Extended Power-Save			X	X	X	X	X	X (1)	X	(2)	X				X	X	X	X	
Power-Save					(4)	X	X	X (1)	X	(2)	X				X		X	X	
Power-Down					(4)	X (3)	X (3)	X (1)		(2)	X				X		X	X	

Notes: 1. Only, if XTO and AVCC voltage regulator are enabled in the RF front end.
2. Active only if enabled
3. Turned off if not selected as clock source for the AVR® or any peripheral (e.g., timer), the watchdog is disabled and the CMCR.SRCD bit is set.
4. Active only if forced on by CMOCR.FRCAO or used by the port debouncing logic.

3.8.5.1 Power Management Register Description

SMCR – Sleep Mode Control Register.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	SM[2:0]			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	

Bits 7..4: Reserved Bits

These bits are reserved and read as zero.

Bits 3..1: SM[2:0] – Sleep Mode Select

These bits select between the four available sleep modes as shown in Table 3-51 on page 212.

Table 3-51. Sleep Mode Selection

SM[2:0]			Sleep Mode
0	0	0	Idle
0	0	1	Extended power-save
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

Bit 0: SE– Sleep Mode 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 purpose, 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 waking-up.

PRR0 – Power Reduction Register 0

This register allows a fine-grained clock control. Clocks for various I/O modules can be turned off by setting the corresponding flag to "high". Therefore, the power consumption of these modules is reduced when they are not needed.

All modules except the SPI are disabled by default and they should be enabled if required.

Bit	7	6	5	4	3	2	1	0	
	-	-	PRCO	-	PRCRC	-	PRRXDC	PRSPI	PRR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	1	1	1	1	1	0	

Bits 7..6: Reserved Bits

These bits are reserved and read as zero.

Bit 5: PRCO – Power Reduction Clock Output

Writing a logic one to this bit stops the clock and shuts down the clock output module. When waking up, the module should be reinitialized to ensure proper operation.

Bits 4: Reserved Bit

This bit is reserved and read as "1".

Bit 3: PRCRC – Power Reduction CRC

Writing a logic one to this bit stops the clock to the stage and shuts down the CRC module. When waking up, the module should be reinitialized to ensure proper operation.

Bits 2: Reserved Bit

This bit is reserved and reads as "1".

Bit 1: PRRXDC – Power Reduction Receive DSP Control

Writing a logic one to this bit stops the clock and shuts down the receive DSP control module. When waking up, the module should be reinitialized to ensure proper operation.

Bit 0: PRSPI – Power Reduction Serial Peripheral Interface

Writing a logic one to this bit stops the clock to the module and shuts down the serial peripheral interface. When waking up, the module should be reinitialized to ensure proper operation.

PRR1 – Power Reduction Register 1

This register allows a fine-grained clock control. Clocks for various I/O modules can be turned off by setting the corresponding flag to “high”. Therefore, the power consumption of these modules is reduced when they are not needed.

The modules are disabled by default and they should be enabled only if required.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	PRT5	PRT4	PRT3	PRT2	PRT1	PRR1
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	1	1	1	1	1	

Bits 7..5: Reserved Bits

These bits are reserved and read as zero.

Bit 4: PRT5 – Power Reduction Timer 5

Writing a logic one to this bit stops the clock to the module and shuts down Timer 5. When waking up, the module should be reinitialized to ensure proper operation.

Bit 3: PRT4 – Power Reduction Timer 4

Writing a logic one to this bit stops the clock to the module and shuts down Timer 4. When waking up, the module should be reinitialized to ensure proper operation.

Bit 2: PRT3 – Power Reduction Timer 3

Writing a logic one to this bit stops the clock to the module and shuts down Timer 3. When waking up, the module should be reinitialized to ensure proper operation.

Bit 1: PRT2 – Power Reduction Timer 2

Writing a logic one to this bit stops the clock to the module and shuts down Timer 2. When waking up, the module should be reinitialized to ensure proper operation.

Bit 0: PRT1 – Power Reduction Timer 1

Writing a logic one to this bit stops the clock to the module and shuts down Timer 1. When waking up, the module should be reinitialized to ensure proper operation.

PRR2 – Power Reduction Register 2

This register allows a fine-grained clock control. Clocks for various I/O modules can be turned off by setting the corresponding flag to “high”. Therefore, the power consumption of these modules is reduced when they are not needed.

The modules are disabled by default and they should be enabled only if required.

Bit	7	6	5	4	3	2	1	0	
	PRSSM	**-**	**PRRS**	**PRIDS**	**PRDF**	**PRSF**	**PRXA**	**PRXB**	**PRR2**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

Bit 7: PRSSM – Power Reduction Sequencer State Machine

Writing a logic one to this bit stops the clock to the module and shuts down the sequencer state machine. When waking up, the module should be reinitialized to ensure proper operation.

Bits 6: Reserved Bit

This bit is reserved and reads as zero.

Bit 5: PRRS – Power Reduction RSSI Buffer

Writing a logic one to this bit shuts down the RSSI buffer by stopping the clock to the module and setting the user interface to its reset values. When waking up, the module should be reinitialized to ensure proper operation.

Bit 4: PRIDS – Power Reduction ID Check

Writing a logic one to this bit shuts down the ID check by stopping the clock to the module and setting the user interface to its reset values. When waking up, the module should be reinitialized to ensure proper operation.

Bit 3: PRDF – Power Reduction Data FIFO

Writing a logic one to this bit shuts down the data FIFO by stopping the clock to the module and setting the user interface to its reset values. When waking up, the module should be reinitialized to ensure proper operation.

Bit 2: PRSF – Power Reduction Support FIFO

Writing a logic one to this bit shuts down the support FIFO by stopping the clock to the module and setting the user interface to its reset values. When waking up, the module should be reinitialized to ensure proper operation.

Bit 1: PRXA – Power Reduction Rx Buffer A

Writing a logic one to this bit stops the clock to the stage and shuts down the receive buffer for data path A. When waking up, the module should be reinitialized to ensure proper operation.

Bit 0: PRXB – Power Reduction Rx Buffer B

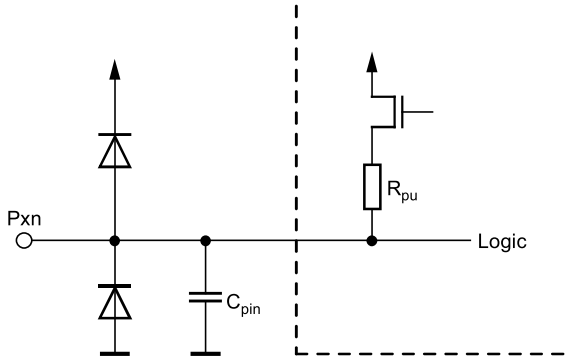
Writing a logic one to this bit stops the clock to the stage and shuts down the receive buffer for data path B. When waking up, the module should be reinitialized to ensure proper operation.

3.8.6 I/O Ports

All 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 AVR instructions. The same applies when changing the drive value (if configured as output) or enabling/disabling pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both, high sink and source capability.

The only exception for this rule is the RX_ACTIVE (PB7) pin when configured as supply. For details, see Section "Alternate Functions of Port B" on page 223. All port pins have individually selectable pull-up resistors with supply-voltage invariant resistance. All I/O pins have protection diodes to both VS and DGND as shown in Figure 3-34 and described in Section 5.1 "ESD Protection Circuits" on page 300. See Section 5.7 "I/O Characteristics for Ports PB0 to PB7 and PC0 to PC5" on page 308 for ports PB0 to PB7 and PC0 to PC5 for a complete list of parameters. The reference pins for the ports are pin 13 (VS) and pin 21 (DGND).

Figure 3-34. 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 within the port. However, when using the register or bit defines in a program, the precise form must be used. For example, as a general rule PORTB3 for bit no. 3 in port B is documented as PORTxn. The physical I/O registers and bit locations are listed in Section 3.8.6.4 "I/O Ports Register Description" on page 228.

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 in the data register. In addition, when it is set the pull-up disable (PUD) bit in MCUCR disables the pull-up function for all pins in all ports.

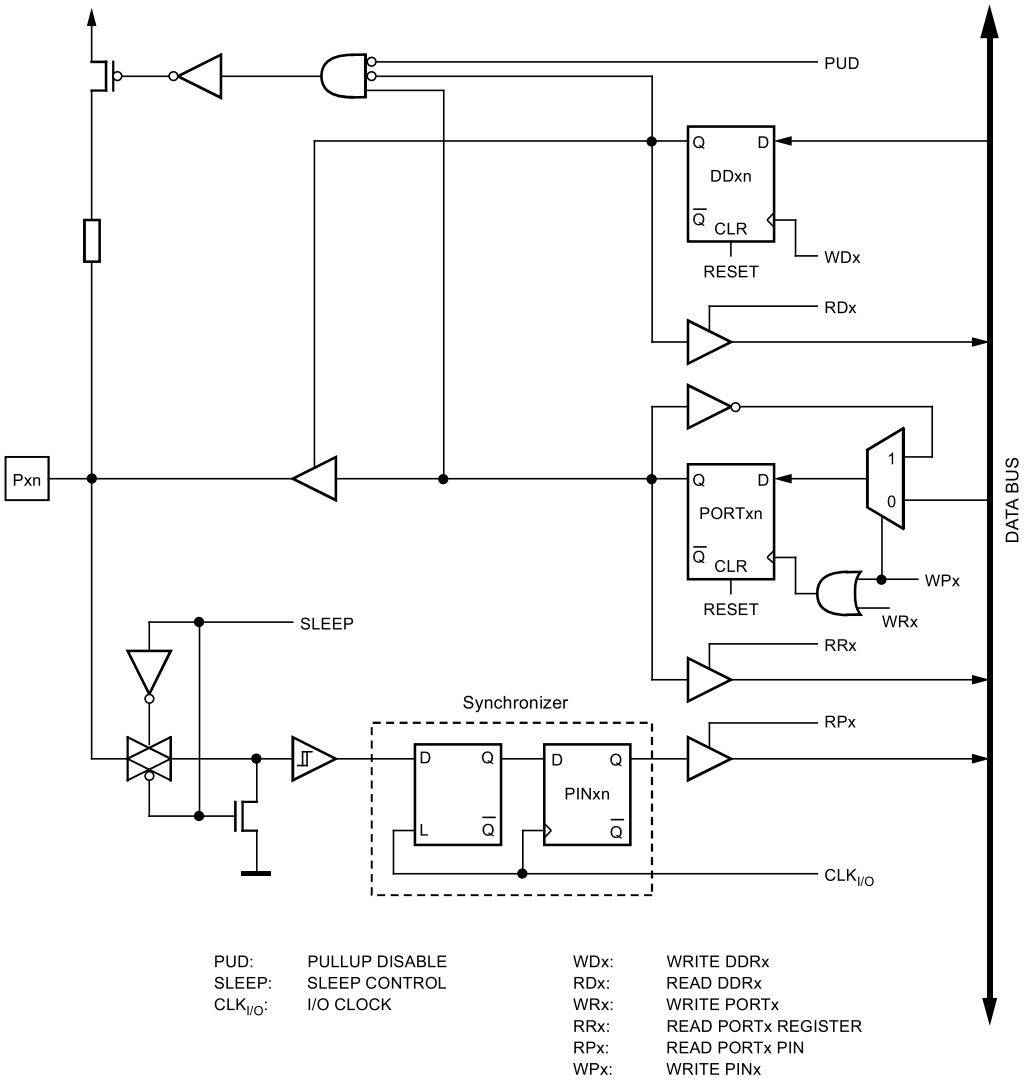
Using the I/O port as general digital I/O is described in Section 3.8.6.1 "Ports as General Digital I/O" on page 216. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in Section 3.8.6.3 "Alternate Port Functions" on page 221. Refer to the individual module sections for a full description of the alternate functions.

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

3.8.6.1 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-up resistors. Figure 3-35 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 3-35. General Digital I/O



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.

Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in Section 3.8.6.4 “I/O Ports Register Description” on page 228, 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 to logic zero or the pin has to be configured as an output pin. A default configuration is applied by the port hardware when the reset condition becomes active, even if no clocks are running. For further details, see Section 3.8.6.3 “Alternate Port Functions” on page 221.

If PORTxn is written to logic one when the pin is configured as an output pin, the port pin is driven HIGH (one). If PORTxn is written to logic zero when the pin is configured as an output pin, the port pin is driven LOW (zero).

All port settings done in DDxn, PORTxn and the port overrides from the alternate port functions need to be globally enabled by the ENPS bit in MCUCR.

Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent of the value of DDRxn.

Note: The SBI instruction can be used to toggle one single bit in a port.

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-up resistors in all ports.

Switching between 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 3-52. Port Pin Configuration

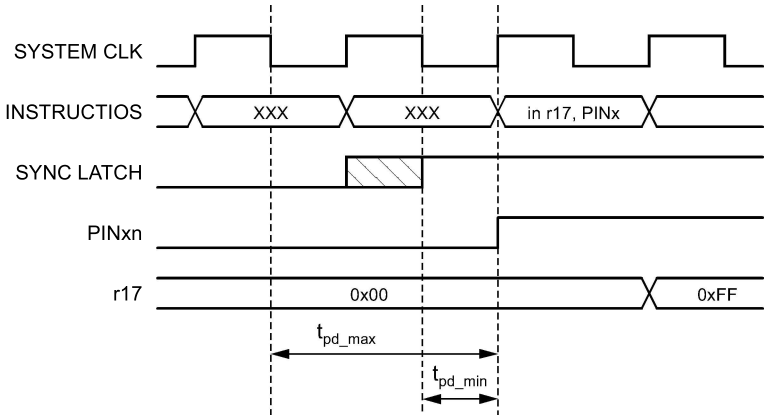
DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-Up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Reading the Pin Value

Independent of the setting of data direction bit DDxn, the port pin can be read through the PINxn register bit. 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 while also introducing a delay.

Figure 3-36 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted as t_{pd_max} and t_{pd_min} respectively.

Figure 3-36. Synchronization when Reading an Externally Applied Pin Value

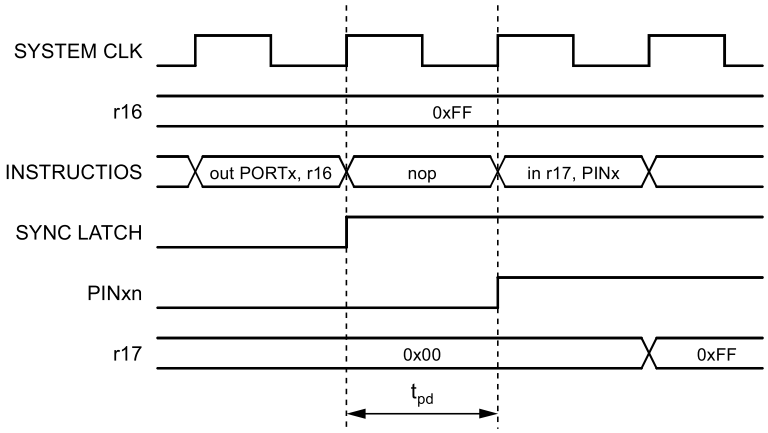


Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the SYNC LATCH signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn register at the following positive clock edge.

As indicated by the two arrows t_{pd_max} and t_{pd_min}, a single signal transition on the pin is delayed between ½ and 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 3-37. The OUT instruction sets the SYNC LATCH signal at the positive edge of the clock. In this case, the propagation delay tpd through the synchronizer is 1 system clock period.

Figure 3-37. Synchronization when Reading a Software Assigned Pin Value



Digital Input Enable and Sleep Modes

As shown in Figure 3-35 on page 216, the digital input signal can be clamped to ground at the input of the Schmitt trigger. The signal labeled SLEEP in the figure is set by the MCU sleep controller in power-down mode, power-save mode and standby mode to avoid increased power consumption if some input signals are left floating or have an analog signal level close to VCC/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 3.8.6.3 “Alternate Port Functions” on page 221.

If a logic HIGH level 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 these sleep mode produces the requested logic change.

Unconnected Pins

If some pins are unused, it is recommended to ensure that these pins have a defined level even if most of the digital inputs are disabled in the deep sleep modes 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 simplest method to ensure a defined level of an unused pin is to enable the internal pull-up. Some pull-ups are disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down resistor. Connecting unused pins directly to VCC or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

3.8.6.2 Port Debouncing

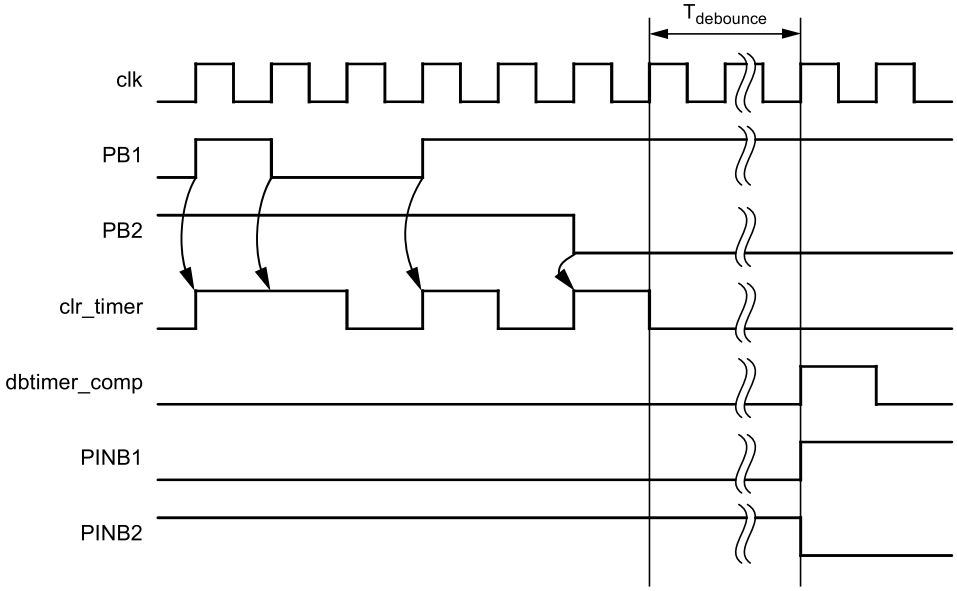
The ports of the Atmel ATA5785 feature hardware debouncing that can be used to avoid false events coming from bouncing mechanical switches. The hardware uses debounce gates for each port pin and a common debounce timer for all ports. The debouncing can be activated and deactivated independently on every port which supports this feature. The debouncing time is configurable in the range of 1ms to 255ms for key debouncing or 5µs to 100µs for LIN-bus debouncing. A level change on any port with activated debouncing resets the debounce timer.

There are two modes for debouncing. In mode 0 a port event is transmitted to the AVR® only if the selected ports are stable (unchanged) for the configured debouncing time and the port values are not the same as the values before the first change (see Section "Debounce Stable Mode (Mode 0)" on page 219). In mode 1 a port event is transmitted to the AVR immediately. All following events are ignored for the configured debouncing time (see Section "Debounce Fast Mode (Mode 1)" on page 220).

Debounce Stable Mode (Mode 0)

All debounce-enabled pins must be stable for the adjusted time $T_{debounce}$ (typ. 10ms). An event on any enabled pin resets the common debounce timer. After the debounce time has elapsed, the pin states are passed through to the AVR bus. An example timing diagram is shown in Figure 3-38. In this debounce mode, short pulses are also suppressed by the debounce logic.

Figure 3-38. Example Timing Diagram for Debounce Stable Mode

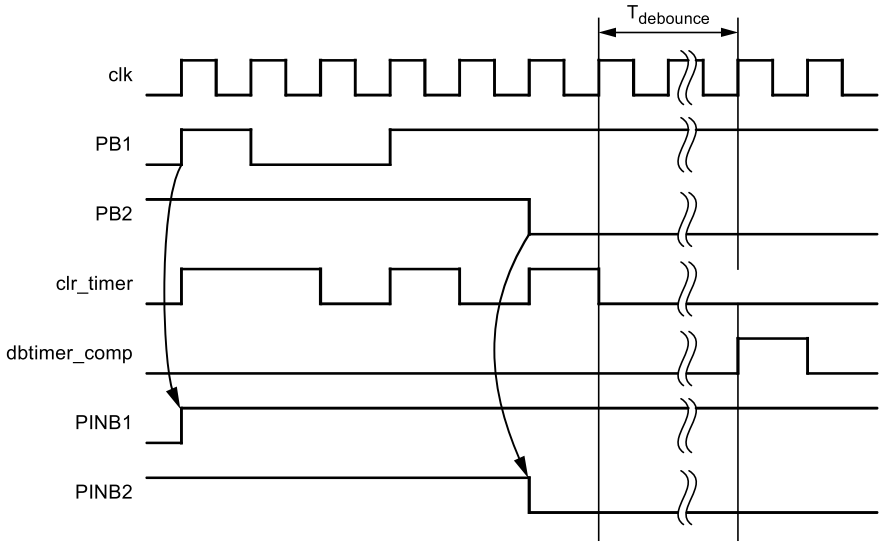


The DBMD bit of the DBCR register must be cleared (DBMD="0") for this mode. The debounce function can be enabled with the DBENx register separately for every pin. The debounce time can be set with the DBTC register and the two control bits DBCS and DBTMS of the DBCR register. The calculation of the debounce time is shown in Table 3-60 on page 231. The debounce function of every pin must be disabled before changing any debounce setting (DBENx="0").

Debounce Fast Mode (Mode 1)

Only the first event on every enabled pin is passed-through to the AVR® core immediately. Every following event on the respective pin will be ignored until the adjustable wait time $T_{debounce}$ (typ. 10ms) has elapsed. Every event resets the common debounce timer. An example timing diagram is shown in Figure 3-39.

Figure 3-39. Example Timing Diagram for Debounce Fast Mode

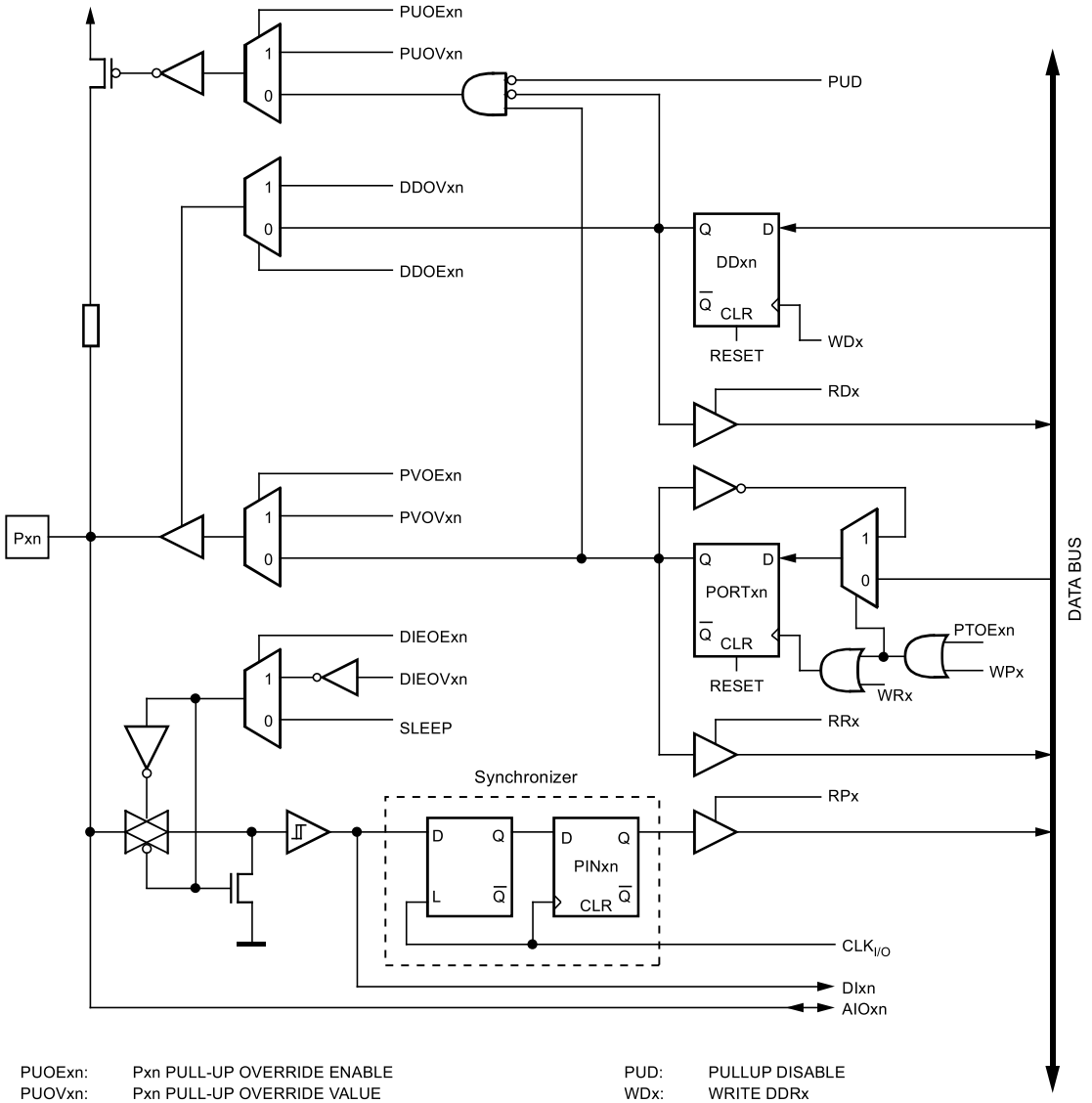


For this mode, bit DBMD of the DBCR register must be set (DBMD="1"). The debounce function can be enabled with the DBENx register separately for every pin. The debounce time can be set with the DBTC register and the two control bits DBCS and DBTMS of the DBCR register. The calculation of the debounce time is shown in Table 3-60 on page 231. The debounce function of every pin must be disabled before changing any debounce setting (DBENx="0").

3.8.6.3 Alternate Port Functions

In addition to being general digital I/Os, most port pins have alternate functions. Figure 3-40 shows how the port pin control signals from the simplified Figure 3-35 on page 216 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 AVR® microcontroller.

Figure 3-40. Alternate Port Functions



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn DATA 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 PINx
DIEOVxn:	Pxn DIGITAL INPUT ENABLE OVERRIDE VALUE	$CLK_{I/O}$:	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	DIGITAL INPUT PIN n ON PORTx
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG INPUT/OUTPUT PIN n 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 3-53 summarizes the function of the overriding signals. The pin and port indices are not shown in the tables below. The overriding signals are generated internally in the modules that have the alternate function.

Table 3-53. 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 to alternate functions. In Figure 3-40 on page 221, 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 briefly describe the alternate functions for each port and explain the relation of the overriding signals to the alternate function.

Alternate Functions of Port B

Port B pins with alternate functions are shown in Table 3-54 and described in more detail farther below.

Table 3-54. Alternate Functions of Port B Pins

Port Pin	Alternate Functions
PB7	**NPWRON6** (not power-on no. 6) **PCINT7** (pin change interrupt 7) **RX_ACTIVE** (strong high-side driver) **LED0** (strong low-side driver) Switchable pull-up resistor
PB6	**PCINT6** (pin change interrupt 6) **EVENT** (firmware controlled event to external microcontroller) Switchable pull-up resistor
PB5	**PCINT5** (pin change interrupt 5) **INT1** (external Interrupt 1) **NSS** (SPI bus master slave select) Switchable pull-up resistor
PB4	**PWRON** (power-on) **PCINT4** (pin change interrupt 4) **LED1** (strong high-side driver) Switchable pull-up resistor
PB3	**PCINT3** (pin change interrupt 3) **MISO** (SPI bus master input/ slave output) Switchable pull-up resistor
PB2	**PCINT2** (pin change interrupt 2) **MOSI** (SPI bus master output/ slave input) Switchable pull-up resistor
PB1	**PCINT1** (pin change interrupt 1) **SCK** (SPI bus master output clock/ slave input clock) Switchable pull-up resistor
PB0	**PCINT0** (pin change interrupt 0) **CLK_OUT** (reference clock output) Switchable pull-up resistor

PB7 – Port B, Bit 7

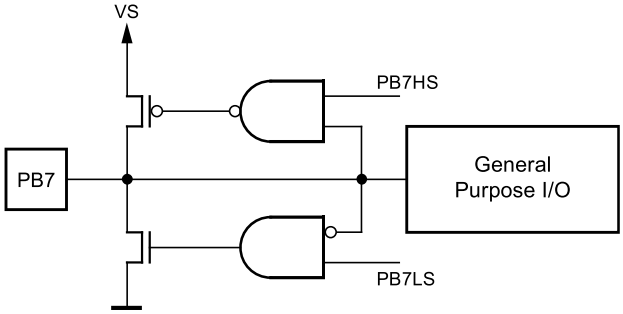
NPWRON6 – Low active power-on signal. Enables the DVCC power supply.

PCINT7 – Pin change interrupt source 7: The PB7 pin can serve as an external interrupt source.

RX_ACTIVE – Firmware controlled, hardware supported function: This indicates that the receiver is active. The pin has a strong low-side driver that can provide more current than the usual pins to supply an external LNA. The high-side driver is enabled on demand by the MCUCR.PB7HS bit (see Section 3.8.6.4 “I/O Ports Register Description” on page 228). It can also be used to drive an external LED. For figures about drive strength, see Section 5.7 “I/O Characteristics for Ports PB0 to PB7 and PC0 to PC5” on page 308.

LED0 – The pin has a strong low-side driver that can provide more current than the usual pins to supply an external LED. The low-side driver is enabled on demand by the MCUCR.PB7LS bit (see Section 3.8.6.4 “I/O Ports Register Description” on page 228). For figures about drive strength, see Section 5.7 “I/O Characteristics for Ports PB0 to PB7 and PC0 to PC5” on page 308.

Figure 3-41. Additional RX_ACTIVE and LED0 Driver



PB6 – Port B, Bit 6

PCINT6 – Pin change interrupt source 6: The PB6 pin can serve as an external interrupt source.

EVENT – Firmware function: Event signalization to external microcontroller. It indicates an important change in the receiver state, for example, valid data detected.

PB5 – Port B, Bit 5

PCINT5 – Pin change interrupt source 5: The PB5 pin can serve as an external interrupt source.

INT1 – External interrupt 1.

NSS – Not slave port select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the DDB5 setting. 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 DDB5. When the pin is forced to be an input, the pull-up resistor can still be controlled by the PORTB5 bit.

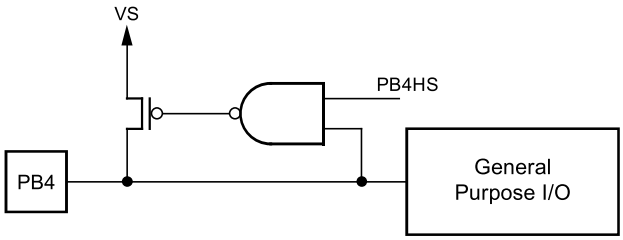
PB4 – Port B, Bit 4

PWRON – High active power-on signal. Enables the DVCC power supply.

PCINT4 – Pin change interrupt source 4: The PB4 pin can serve as an external interrupt source.

LED1 – The pin has a strong high-side driver that can provide more current than the usual pins to supply an external LED. The high-side driver is enabled on demand by the MCUCR.PB4HS bit (see Section 3.8.6.4 “I/O Ports Register Description” on page 228). For figures about drive strength, see Section 5.7 “I/O Characteristics for Ports PB0 to PB7 and PC0 to PC5” on page 308.

Figure 3-42. Additional LED1 Driver



PB3 – Port B, Bit 3

PCINT3 – Pin change interrupt source 3: The PB3 pin can serve as an external interrupt source.

MISO – SPI master data input, slave data output for SPI channel. When the SPI is enabled as a master, this pin is configured as an input regardless of the DDB3 setting. When the SPI is enabled as a slave, the data direction of this pin is controlled by DDB3. When the pin is forced to be an input, the pull-up resistor can still be controlled by the PORTB3 bit.

PB2 – Port B, Bit 2

PCINT2 – Pin change interrupt source 2: The PB2 pin can serve as an external interrupt source.

MOSI – Master data output, slave data input pin for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the DDB2 setting. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB2. When the pin is forced to be an input, the pull-up resistor can still be controlled by the PORTB2 bit.

PB1 – Port B, Bit 1

PCINT1 – Pin change interrupt source 1: The PB1 pin can serve as an external interrupt source.

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 DDB1 setting. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB1. When the pin is forced to be an input, the pull-up resistor can still be controlled by the PORTB1 bit.

PB0 – Port B, Bit 0

PCINT0 – Pin change interrupt source 0: The PB0 pin can serve as an external interrupt source.

CLK_OUT – A divided XTO, ADC or RC oscillator clock can be output on this pin as a reference clock for an external microcontroller. For further details, see Section 3.8.4 “System Clock and Clock Options” on page 201. The corresponding data direction register of the port has to be set as output to enable the driver.

OFFMode Behavior of Port B

The port B functions for the OFFMode are hard wired.

All port pins of port B are inputs in the OFFMode. PB0 to PB6 have no internal pull-up resistor.

Table 3-55. Port B OFFMode Configuration

Port Pin	Internal Pull-Up	OFFMode Function
PB7	Yes	NPWRON6 (low active)
PB6	No	Floating input. Port circuitry avoids internal transverse current.
PB5	No	Floating input. Port circuitry avoids internal transverse current.
PB4	No	PWRON (high active) Input level must be defined by external circuitry to avoid unwanted wake-ups and transverse current. Ground level (low) is required to be able to use the OFFMode.
PB3	No	Floating input. Port circuitry avoids internal transverse current.
PB2	No	MOSI Input level must be defined by external circuitry to avoid internal transverse current in the port.
PB1	No	SCK Input level must be defined by external circuitry to avoid internal transverse current in the port.
PB0	No	Floating input. Port circuitry avoids internal transverse current.

Alternate Functions of Port C

Port C pins with alternate functions are shown in the Table 3-56 and described in more detail farther below.

Table 3-56. Alternate Functions of Port C Pins

Port Pin	Alternate Functions
PC5	**NPWRON5** (not power-on no. 5) **PCINT13** (pin change interrupt 13) **TMDO_CLK** (transparent mode data clock output) **TRPB** (transparent Rx path B data) Switchable pull-up resistor
PC4	**NPWRON4** (not power-on no. 4) **PCINT12** (pin change interrupt 12) **INT0** (external interrupt 0) Switchable pull-up resistor
PC3	**NPWRON3** (not power-on no. 3) **PCINT11** (pin change interrupt 11) **TMDO** (transparent data output) Switchable pull-up resistor
PC2	**NPWRON2** (not power-on no. 2) **PCINT10** (pin change interrupt 10) **TRPA** (transparent Rx path A data) Switchable pull-up resistor
PC1	**NPWRON1** (not power-on no. 1) **PCINT9** (pin change interrupt 9) Switchable pull-up resistor
PC0	**PCINT8** (pin change interrupt 8) **NRESET** (low active microcontroller reset) Switchable pull-up resistor

PC5 – Port C, Bit 5

NPWRON5 – Low active power-on signal. Enables the DVCC power supply.

PCINT13 – Pin change interrupt source 13: The pin can serve as an external interrupt source.

TMDO_CLK – Transparent mode data clock output. Provides a sampling clock for the transparent mode data output (TMDO pin). It can be used to clock the data with the rising clock edge into a receiving shift register. TMDO_CLK has priority over the TRPB output function.

TRPB – The raw data of the transparent receiving path B is provided to an external microcontroller for decoding. The corresponding data direction register of the port has to be set as output to enable the driver. TRPB has lower priority than TMDO_CLK.

PC4 – Port C, Bit 4

NPWRON4 – Low active power-on signal. Enables the DVCC power supply.

PCINT12 – Pin change interrupt source 12: The pin can serve as an external interrupt source.

INT0 – External interrupt 0.

PC3 – Port C, Bit 3

NPWRON3 – Low active power-on signal. Enables the DVCC power supply.

PCINT11 – Pin change interrupt source 11: The pin can serve as an external interrupt source.

TMDO – Transparent mode data output (Rx). Decoded and spike-free ASK or FSK data can be multiplexed to this port. The corresponding data direction register of the port has to be set as output to enable the driver.

PC2 – Port C, Bit 2

NPWRON2 – Low active power-on signal. Enables the DVCC power supply.

PCINT10 – Pin Change Interrupt Source10: The pin can serve as an external interrupt source.

TRPA – The raw data of the transparent receiving path A is provided to an external microcontroller for decoding. The corresponding data direction register of the port has to be set as output to enable the driver.

PC1 – Port C, Bit 1

NPWRON1 – Low active power-on signal. Enables the DVCC power supply.

PCINT9 – Pin change interrupt source 9: The pin can serve as an external interrupt source.

PC0 – Port C, Bit 0

PCINT8 – Pin change interrupt source 8: The pin can serve as an external interrupt source.

NRESET – Nreset input for the AVR®. A low level on this pin for longer than 10µs generates a reset, even if the clock is not running. Shorter pulses could also trigger a reset.

OFFMode Behavior of Port C

The port C functions for the OFFMode are hard wired.

All port pins of port C are inputs in OFFMode.

Table 3-57. Port C OFFMode Configuration

Port Pin	Internal Pull-Up	OFFMode Function
PC5	YES	NPWRON5 (low active)
PC4	YES	NPWRON4 (low active)
PC3	YES	NPWRON3 (low active)
PC2	YES	NPWRON2 (low active)
PC1	YES	NPWRON1 (low active)
PC0	YES	NRESET

3.8.6.4 I/O Ports Register Description

MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
	PB7HS	PB7LS	PB4HS	PUD	ENPS	SPIIO	IVSEL	IVCE	MCUCR
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: PB7HS – Port B7 High-Side Driver Enable

Enables the strong high-side driver on PB7. It can be used to supply an external LNA preamplifier connected to ground. The driver provides a switchable connection to the supply voltage.

Bit 6: PB7LS – Port B7 Low-Side Driver Enable

Enables the strong low-side driver on PB7. It can be used to drive an LED connected to supply. The driver provides a switchable connection to ground.

Bit 5: PB4HS – Port B4 High-Side Driver Enable

Enables the strong high-side driver on PB4. It can be used to supply an external LNA preamplifier connected to ground. The driver provides a switchable connection to the supply voltage.

Bit 4: PUD – Pull-up Resistors 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 3.8.6.1 “Ports as General Digital I/O” on page 216 for details about this feature.

Bit 3: ENPS – Enable Port Settings

Activates the register-defined port settings. If this bit is “0”, hard-wired settings as described in Table 3-55 on page 225 and in Table 3-57 are used as the default.

Bits 2..0: AVR internal bits.

PORTB – Port B Data Register

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	

For a detailed description, see Section 3.8.6.1 “Ports as General Digital I/O” on page 216.

DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	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	

For a detailed description, see Section 3.8.6.1 “Ports as General Digital I/O” on page 216.

PINB – Port B Input Pins Address

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

For a detailed description, see Section 3.8.6.1 “Ports as General Digital I/O” on page 216.

PORTC – Port C Data Register

Bit	7	6	5	4	3	2	1	0	
	-	-	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
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	

For a detailed description, see Section 3.8.6.1 “Ports as General Digital I/O” on page 216.

DDRC – Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
	-	-	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	DDRC
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	

For a detailed description, see Section 3.8.6.1 “Ports as General Digital I/O” on page 216.

PINC – Port C Input Pins Address

Bit	7	6	5	4	3	2	1	0	
	-	-	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

For a detailed description, see Section 3.8.6.1 “Ports as General Digital I/O” on page 216.

DBCR – Debounce Control Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	DBHA	DBTMS	DBCS	DBMD	DBCR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Changes to the DBCR register are only allowed when debouncing is disabled (set all DBENx registers to 0x00)!

Bits 7..4: Reserved Bits

These bits are reserved and read as zero.

Bit 3: DBHA – Debounce Handshake Active Flag

This flag is set to one while the debounce handshake is active. The DBHA flag must be zero before the I/O-clock is disabled (e.g., going to sleep mode other than “idle”). Debouncing does not continue if the handshake of a previous event has not finished.

Bit 2: DBTMS – Debounce Timer Mask Select

The 8-bit debounce timer compare register DBTC is compared with the 15-bit timer value. The range of the comparison between the 8-bit DBTC and the 15-bit timer value is selected by the DBTMS bit as shown in Table 3-58. The resulting debounce timings are shown in Table 3-61 on page 231.

Table 3-58. Debounce Timer Compare Mask Select

DBTMS	Comparison Range	Application Example
0	Bit 14 – Bit 7	Key debouncing
1	Bit 7 – Bit 0	LIN-Bus debouncing

Bit 1: DBCS – Debounce Clock Select

The debounce clock select bit selects the source clock of the debounce block, see Table 3-59.

Table 3-59. Debounce Clock Select

DBCS	Clock Source	Application Example
0	CLK_{SRC} (f_{typ} = 125kHz)	Key debouncing
1	CLK_{FRC} (f_{typ} = 6.36MHz)	LIN-Bus debouncing

Bit 0: DBMD – Debounce Mode

This bit selects the common mode for port debouncing. If this bit is cleared (DBMD=0), debounce mode 0 (stable mode) is active (see Section "Debounce Stable Mode (Mode 0)" on page 219). If this bit is set (DBMD=1), debounce mode 1 (fast mode) is active (see Section "Debounce Fast Mode (Mode 1)" on page 220).

DBENB – Debounce Enable Register Port B

Bit	7	6	5	4	3	2	1	0	
	DBENB7	**DBENB6**	**DBENB5**	**DBENB4**	**DBENB3**	**DBENB2**	**DBENB1**	**DBENB0**	**DBENB**
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: DBENBn – Debounce Enable

These bits enable the debounce function for the respective pin n on port B. The enabled pin is then debounced with the selected mode (DBMD).

DBENC – Debounce Enable Register Port C

Bit	7	6	5	4	3	2	1	0	
	-	**-**	**DBENC5**	**DBENC4**	**DBENC3**	**DBENC2**	**DBENC1**	**DBENC0**	**DBENC**
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: Reserved Bits

These bits are reserved and read as zero.

Bits 5..0: DBENCn – Debounce Enable

These bits enable the debounce function for the respective pin n on port B. The enabled pin is then debounced with the selected mode (DBMD).

DBTC – Debounce Timer Compare Register

Bit	7	6	5	4	3	2	1	0	
	DBTC[7:0]								DBTC
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	

Changes to the DBTC register are only allowed when debouncing is disabled (set all DBENx registers to 0x00).

Bits 7..0: DBTC – Debounce Timer Compare

The Debounce Timer Compare register contains an 8-bit value that is continuously compared with the 15-bit timer value. DBTC can be set from 0 to 255. The resulting debounce timings also depend on the DBCS and DBTMS bits of the debounce control register DBCR (see Table 3-58 on page 229 and Table 3-59 on page 230). The calculation of the debounce time is shown in Table 3-60 and the resulting timing ranges in Table 3-61.

After the debouncing time a wait time caused by a handshake synchronizer needs to be considered before the next debouncing cycle on the appropriate pin is started (approximately 6 clock cycles of debounce clock and 6 clock cycles of I/O-clock).

Table 3-60. Debounce Timing Calculation

DBCS	DBTMS	$T_{debounce\_typ}$
0	0	(DBTC × 128 + 134) × 8µs
0	1	(DBTC + 7) × 8µs
1	0	(DBTC * 128 + 134) × 157ns
1	1	(DBTC + 7) × 157ns

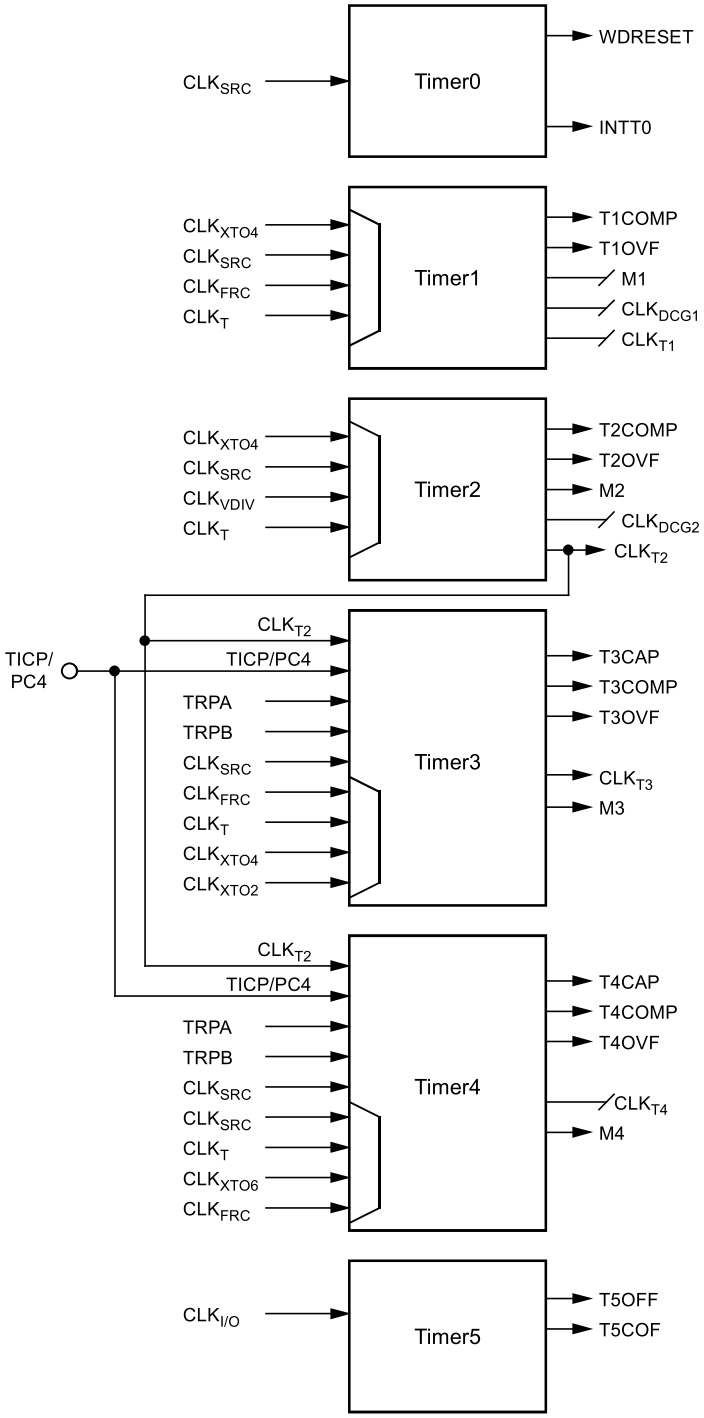
Table 3-61. Debounce Timing Ranges

DBCS	0 (SRC)				1 (FRC)			
DBTMS	0		1		0		1	
DBTC	0	255	0	255	0	255	0	255
Debounce Time	1.1ms	262ms	56µs	2.1ms	21µs	5.1ms	1.1µs	41.1µs

3.8.7 Timer Module

The timer module includes Timer0 with watchdog, the asynchronous 8-bit Timer1 and Timer2, the asynchronous 16-bit Timer3, and Timer4 and a synchronous 16-bit Timer5. Figure 3-43 shows the timer module structure.

Figure 3-43. Timer Module Structure



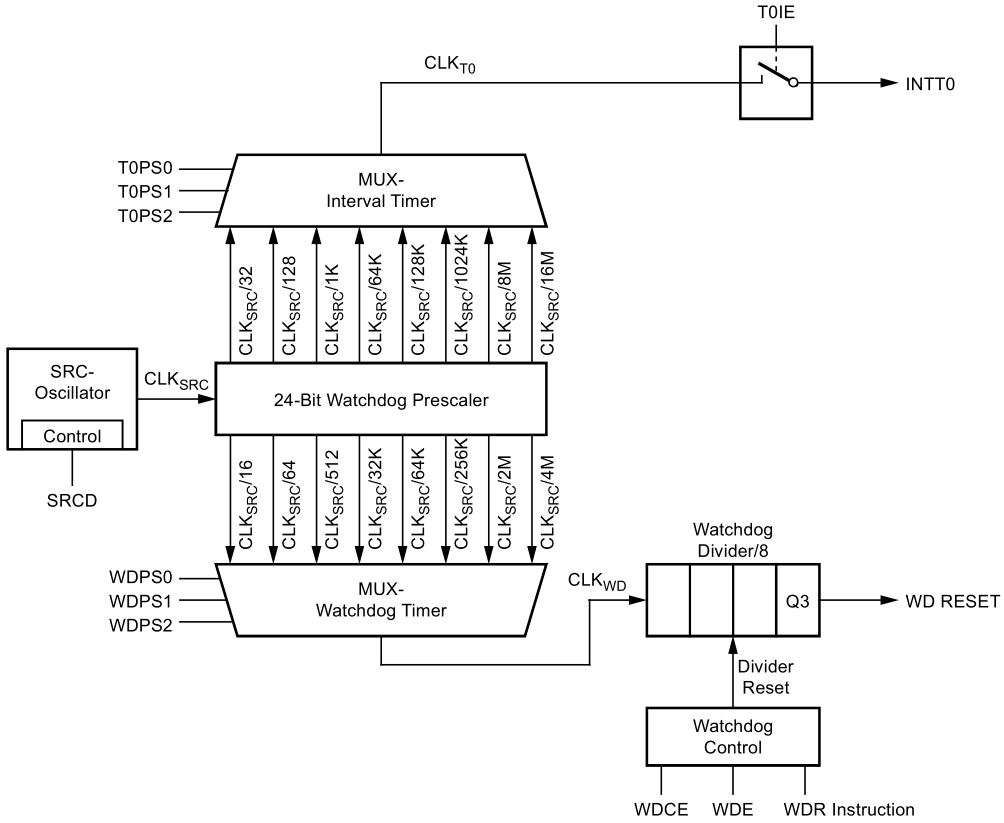
The timer module offers many operating modes for generating intervals, counting events, interrupts, and PWM signals. It uses a variety of clock sources and many modes that enable numerous interactions between the single timers. The timer module is fully controlled by the AVR® MCU by using a set of control registers which are available in MCU I/O address space.

3.8.7.1 Timer0 – Watchdog/Interval Timer

Timer0 is a watchdog/interval timer which can be used to generate periodical interrupts and as a prescaler for the watchdog function. Since this timer also runs during UHF reception it is designed to avoid harmonics and noise generation.

The watchdog/interval is clocked from a calibrated on-chip slow RC (SRC) oscillator with a nominal frequency of f_{SRC} = 125kHz. The oscillator is described in Section 3.8.4.1 “Clock Sources” on page 202.

Figure 3-44. Watchdog Timer



The SRC and the Timer0 can work together as an ultra-low power watchdog/interval timer stage.

Timer0 consists of a programmable 24-stage divider that is driven by CLK_{SRC}. The timer output signal (CLK_{T0}) can be used as a source for the Timer0 interrupt. The interrupt is maskable via the T0IE bit and the interval for the timer output can be adjusted as shown in Section “Timer0 Register Description” on page 234 via the T0PS[2:0] bits in the Timer0 control register T0CR.

The 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.

Timer0 can also be used as a watchdog timer to prevent system stalling. The watchdog divider is a 3-bit counter that is supplied by a separate output clock (CLK_{WD}) of Timer0 and 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 watchdog reset, the microcontroller is reset and executes from the reset vector. By controlling the watchdog timer prescaler, the watchdog reset interval can be adjusted as shown in Section “Timer0 Register Description” on page 234 via the WDPS[2:0] bits in the Timer0 watchdog control register WDTCR.

Timer0 Register Description

WDTCR – Watchdog Timer0 Control Register

Bit	7	6	5	4	3	2	1	0	
	-		-	WDCE	WDE	WDPS[2:0]			WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	1	1	1	

Bits 7..5: Reserved Bits

These bits are reserved and read as zero.

Bit 4: WDCE – Watchdog Change Enable

This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE has to be set. Once written to one, the hardware clears WDCE after four clock cycles.

Bit 3: WDE – Watchdog Enable

The watchdog timer is enabled if the WDE bit is “1”. If the WDE is written to logic zero, the watchdog timer function is disabled. WDE can only be cleared if the WDCE bit has logic level one.

The following procedure must be followed to disable an enabled watchdog timer:

In the same 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.

Within the next four clock cycles, write a logic “0” to WDE. This disables the watchdog.

In safety level 2, it is not possible to disable the watchdog timer, even with the algorithm described above. WDE is overridden by WDRF in the MCUSR. This means the WDE is always set when the WDRF is set. To clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

Bits 2..0: WDPS[2:0] – Watchdog Prescaler Select

The WDPS[2:0] bits determine the watchdog timer prescaling clock output (CLK_{WD}) when the watchdog timer is enabled. The time-out value after a watchdog reset command (WDR) is not precisely known because the WDR command resets only the watchdog divider. The watchdog prescaler is shared by the watchdog and Timer0 and therefore continues counting. For this reason a typical and minimum guaranteed time-out period is given in Table 3-62 on page 234.

Table 3-62. Watchdog Timer Prescaler Select Bit Description

WDPS[2:0]			Prescaler Divider Values (Number of CLK_{SRC} Cycles)	Total Numbers of CLK_{SRC} Cycles Including Watchdog Divider	Time-out at VCC = 3V/25°C and $T_{SRC} \approx$ 1/125KHz	
					Typical	Minimum
0	0	0	16 cycles	8 • 16 = 128 cycles	1ms	0.85ms
0	0	1	64 cycles	8 • 64 = 512 cycles	4ms	3.4ms
0	1	0	512K cycles	8 • 512 = 4K cycles	32ms	27ms
0	1	1	32K cycles	8 • 32K = 256K cycles	2.1s	1.75s
1	0	0	64K cycles	8 • 64K = 512K cycles	4.2s	3.5s
1	0	1	256K cycles	8 • 256K = 2M cycles	16.8s	14s
1	1	0	2M cycles	8 • 2M = 16M cycles	134s	110s
1	1	1	4M cycles	8 • 4M = 32M cycles	268s	220s

T0CR – Timer0 Control Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	T0PR	T0IE	T0PS[2:0]			T0CR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 4: T0PR – Timer0 Prescaler Reset

Writing T0PR to one restarts the watchdog prescaler. Once written to one, the hardware clears this bit after four clock cycles. Only if the watchdog function is disabled can the watchdog prescaler be restarted.

Bit 3: T0IE – Timer0 Interrupt Enable

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 T0FR, is set.

Bits 2..0: T0PS[2:0] – Timer0 Prescaler Select

The T0PS[2:0] bits determine the Timer0 prescaling clock output (CLK_{T0}). The different prescaling values and their corresponding time-out periods are shown in Table 3-63.

Table 3-63. Timer0 Interrupt Prescaler Selection

T0PS2	T0PS1	T0PS0	Number of Oscillator Cycles (CLK_{SRC})	Typical Time-out at T_{SRC} 1/125KHz for CLK_{T0}
0	0	0	32 cycles	0.256ms
0	0	1	128 cycles	1ms
0	1	0	1K cycles	8ms
0	1	1	64K cycles	0.5s
1	0	0	128K cycles	1s
1	0	1	1M cycles	8s
1	1	0	8M cycles	67s
1	1	1	16M cycles	134s

T0IFR – Timer0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	T0F	T0IFR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..1: Reserved Bits

These bits are reserved and read as zero.

Bit 0: T0F – Timer0 Flag

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) in 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 logical one to it.

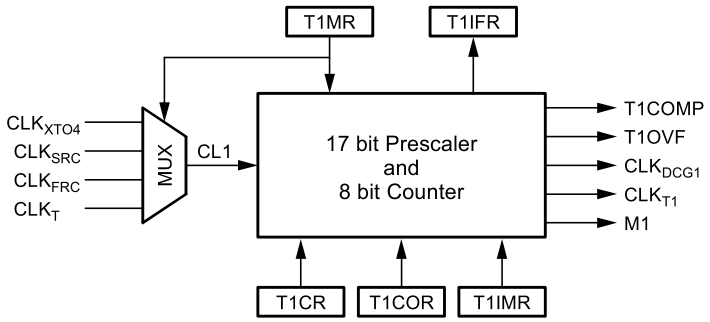
3.8.7.2 Timer1

This timer can be used to generate real time clock (RTC) interrupts and clock output.

Main features:

- 8-bit timer/counter with 17-bit prescaler and compare modes
- True 8-bit design (i.e., allows 8-bit PWM)
- Four different selectable input clocks
- One output compare unit
- Clear timer on compare match
- Programmable PWM period
- Frequency generator
- External event counter
- Two independent interrupt sources (T1COM, T1OVF)

Figure 3-45. Timer1 Block Diagram



Timer1 consists of a 17-bit prescaler and an 8-bit up counter stage with compare register (T1COR). The timer can be used as an event counter, timer and signal generator. The counter can be driven by internal and external clock sources. The 8-bit counter is readable via AVR® only when the timer is stopped.

The Timer1 control and mode registers (T1CR, T1MR) are 8-bit registers. The comparator output is controlled by a control register (T1CR) and contains the mask bits for the actions (counter reset, output toggle, timer interrupt) which can be triggered by a compare match event or by counter overflow. This architecture enables the timer for various modes.

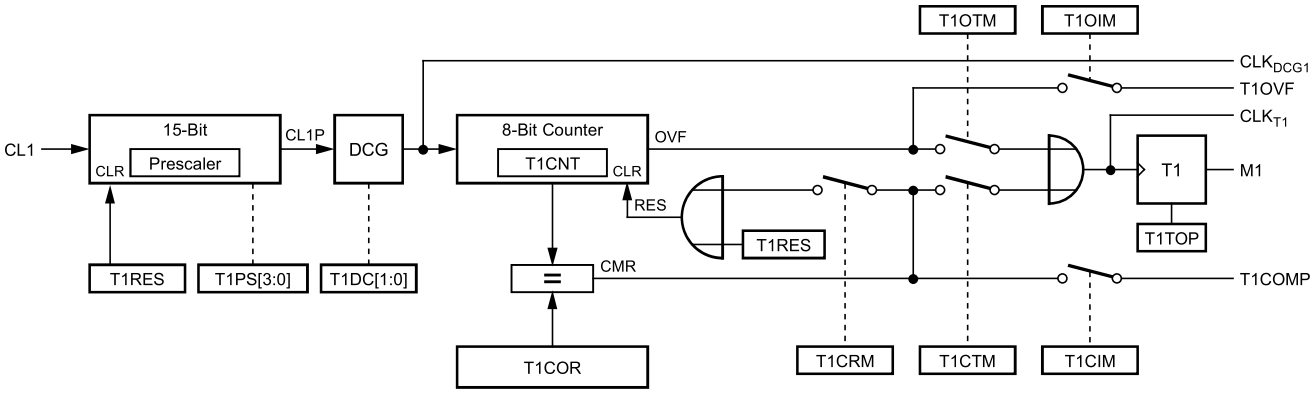
Interrupt request signals are all visible in the Timer1 interrupt flag register (T1IFR). All interrupts are individually maskable with the Timer1 interrupt mask register (T1IMR).

The counter input clock (CL1) can be supplied via the T clock of the system clock prescaler (CLK_T), the crystal-oscillator-based clock (CLK_{XTO4}), the internal SRC clock (CLK_{SRC}) or the internal fast RC clock (CLK_{FRC}).

The output compare register (T1COR) is compared with the counter value at all times. T1COR can be modified while the timer is running.

The 15-bit prescaler works together with an additional 2-bit duty cycle generator as a 16-bit prescaler for 50% duty cycle output or 17-bit prescaler with odd duty cycles. The counter stage has one input signal (CL1) and five output signals (T1OVF, T1COMP, CLK_{DCG1}, M1 and CLK_{T1}). The M1, CLK_{T1} and CLK_{DCG1} are the output clocks, T1OVF, and T1COMP are the interrupt request signals of the counter stage.

Figure 3-46. Timer1 8-Bit Counter Stage



Signal description of the Timer1 (internal signals):

CL1	Prescaler input clock
CLK_{DCG1}	Timer1 input clock and duty cycle generator output clock
CLP1	Prescaler output clock and duty cycle generator (DCG) input clock
T1OVF	Timer1 counter overflow interrupt
T1COMP	Timer1 compare match interrupt
CLK_{T1}	Timer1 counter stage output clock
M1	Modulator output toggle flip-flop
CMR	Counter compare register match
RES	Counter reset (clear all bits)
OVF	Counter overflow

Timer1 Register Description

T1CR – Timer1 Control Register

Bit	7	6	5	4	3	2	1	0	
	T1ENA	**T1TOS**	**T1RES**	**T1TOP**	**-**	**T1CRM**	**T1CTM**	**T1OTM**	**T1CR**
Read/Write	R/W	R/W	W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: T1ENA – Timer1 Enable

This bit controls the Timer1 block. The T1ENA bit must be written to logic one to enable Timer1. If the T1ENA bit is written to logic zero, Timer1 is disabled. Reading this bit shows the actual state of Timer1. Because internal synchronization requires 2½ asynchronous CL1 clock cycles to enable or disable Timer1, it may take some time to read a logic one after having enabled Timer1. The same applies for disabling.

Care has to be taken if the T1ENA or T1CR register is written by consecutive cbi/sbi instructions. For example, clearing the T1ENA bit with a cbi instruction followed by a cbi/sbi instruction on another bit of the T1CR re-enables the timer. (The read-modify-write sequence is still reading the T1ENA=1 and writing it back, thus enabling the timer).

The asynchronous clock can be forced on without synchronization by writing a logic one to the T1ENA bit twice within four AVR® clock cycles. This can be useful for slow clocks, for example, when the first edge of a newly enabled oscillator should already lead to an increased counter value.

Bit 6: T1TOS – Timer1 Toggle with Start

The T1TOS bit must be written to logic one if the modulator output of Timer1 should be toggled when the timer is enabled with T1ENA. If the T1TOS bit is written to logic zero, the modulator output of Timer1 is not toggled with the timer enables.

Bit 5: T1RES – Timer1 Reset

The T1RES bit can be written to logic one to reset the prescaler and counter. This is only allowed if the timer is stopped (T1ENA=0). The T1RES bit is automatically cleared one cycle after the write.

Bit 4: T1TOP – Timer1 Toggle Output Preset

The T1TOP bit must be written to logic one to set the toggle flip-flop. Clearing the T1TOP resets the toggle flip-flop. This bit allows the programmer to preset the toggle output flip-flop in the modulator of Timer1. If the timer is stopped, this bit shows the actual value of the toggle flip-flop.

Note: If T1ENA = “1”, no output preset is possible.

Bit 3: Reserved Bit

This bit is reserved and reads as zero.

Bit 2: T1CRM – Timer1 Compare Reset Mask

The T1CRM bit must be written to logic one to enable the counter reset if a match of the counter with the compare register occurs. If the T1CRM bit is written to logic zero, the counter reset is disabled.

Bit 1: T1CTM – Timer1 Compare Toggle Mask

The T1CTM bit must be written to logic one to enable the compare toggle. If the T1CTM bit is written to logic zero, the compare toggle is disabled. A match of the counter with the compare register generates an output clock of the counter (CLK_{T1}).

Bit 0: T1OTM – Timer1 Overflow Toggle Mask

The T1OTM bit must be written to logic one to enable the overflow toggle. If the T1OTM bit is written to logic zero, the overflow toggle is disabled. A counter overflow generates an output clock of the counter (CLK_{T1}).

T1CNT – Timer1 Counter Register

Bit	7	6	5	4	3	2	1	0	
	T1CNT[7:0]								**T1CNT**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The counter register (T1CNT) contains an 8-bit counter value. It shall only be read when Timer1 is disabled (T1ENA= ”0”). Reading the register during operation may lead to corrupted values since the timer can run on asynchronous clocks.

T1COR – Timer1 Compare Register

Bit	7	6	5	4	3	2	1	0	
	T1COR[7:0]								**T1COR**
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 register contains an 8-bit value that is continuously compared with the counter value (T1CNT). A match can be used to generate a compare interrupt, a counter reset or an output clock CLK_{T1}. The compare register can be written while the timer is running. Potential compare match interrupts generated from glitches on the clock domain crossing are ignored for two AVR® clock cycles.

T1IFR – Timer1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	-		-	-	-	-	T1COF	T1OFF	T1IFR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 1: T1COF – Timer1 Compare Flag

This flag is set to one during the clock cycle after the counter value has matched with the compare register. The flag (T1COF) is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

Bit 0: T1OFF – Timer1 Overflow Flag

This flag is set by the T1OVF signal when the counter reaches its maximum value (0xFF).

If the I bit in SREG and the T1OIM bit are set in T1IMR, 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 logical one to it.

Note: The overflow flag is also set if the compare value is set to FF.

T1IMR – Timer1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	T1CIM	T1OIM	T1IMR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 1: T1CIM – Timer1 Compare Interrupt Mask

If the T1CIM bit is written to one and the I bit in SREG is set, the Timer1 compare match interrupt is enabled. The corresponding interrupt is executed if a compare match occurs, i.e., when the T1COF bit is set in the Timer1 interrupt flag register (T1IFR).

Bit 0: T1OIM – Timer1 Overflow Interrupt Mask

If the T1OIM bit is written to one and the I bit in SREG is set, the Timer1 overflow interrupt is enabled. The corresponding interrupt is executed if a counter overflow occurs, i.e., when the T1OFF bit is set in the Timer1 interrupt flag register (T1IFR).

T1MR – Timer1 Mode Register

This register should only be modified while the timer is disabled (T1CR.T1ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T1DC[1:0]		T1PS[3:0]				T1CS[1:0]		T1MR
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	

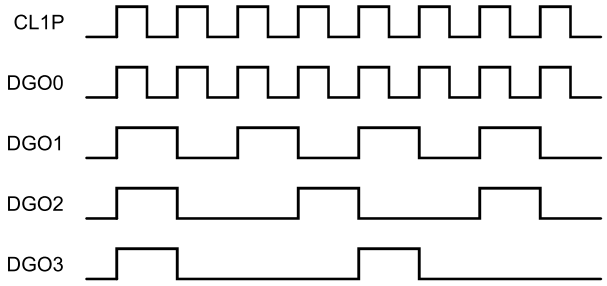
Bits 7..6: T1DC[1:0] – Timer1 Duty Cycle

The T1DC1 and T1DC0 bits select the duty cycle mode of the duty cycle generator as shown in Table 3-64.

Table 3-64. Timer1 Duty Cycle Bit Description

T1DC1	T1DC0	Function of the Duty Cycle Generator	Additional Divider Effect
0	0	Bypassed (DCG0)	1
0	1	Duty cycle 1/1 (DCG1)	2
1	0	Duty cycle 1/2 (DCG2)	3
1	1	Duty cycle 1/3 (DCG3)	4

Figure 3-47. DCG Output Signals



Bits 5..2: T1PS[3:0] – Timer1 Prescaler Select

The T1PS[3:0] bits select the prescaler value of Timer1 as shown in the following formulas:

$$\text{prescaler Value} = 2^{\text{T1PS}[3:0]} \quad (42)$$

$$\text{CL1P}_{\text{Frequency}} = \frac{\text{CL1}_{\text{Frequency}}}{\text{prescaler Value}} = \frac{\text{CL1}_{\text{Frequency}}}{2^{\text{T1PS}[3:0]}} \quad (43)$$

$\text{T1PS}[3:0] \in \{0..15\}$

Bits 1..0: T1CS[1:0] – Timer1 Clock Select

The T1CS[1:0] bits select the input clock (CL1) of Timer1 as shown in Table 3-65.

Table 3-65. Timer1 Input Clock Select Bit Description

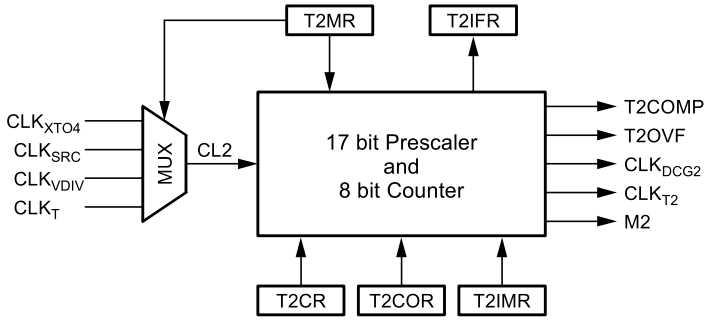
T1CS[1:0]		Input Clock (CL1) of Timer
0	0	CLK_{SRC}
0	1	CLK_{FRC}
1	0	CLK_{T}
1	1	CLK_{XTO4}

3.8.7.3 Timer2

Timer2 is designed to be used mainly to generate periodical interrupts with high accuracy over a wide timing range, for example, sleep and polling times can be realized in 1% steps for periods from 50ms to 1000ms with the SRC (125kHz) and the divided XTO clock. Divider inaccuracy due to the prescaler can be compensated by firmware. This is done by either modifying the compare register value on every n^{th} cycle or by omitting a complete compare match at periodic intervals.

- 8-bit timer/counter with 17-bit prescaler and compare modes
- True 8-bit design (i.e., allows 8-bit PWM)
- Four different selectable input clocks
- One output compare unit
- Clear timer on compare match
- Different programmable PWM period
- Frequency generator
- External event counter
- Two independent interrupt sources (T2COM, T2OVF)

Figure 3-48. Timer2 Block Diagram



Timer2 consists of a 17-bit prescaler and an 8-bit up counter stage with compare register (T2COR). The timer can be used as an event counter, timer, and signal generator. The counter can be driven by internal and external clock sources. The 8-bit counter should be read via AVR® only when the timer is stopped. Otherwise unpredictable values can be read from the asynchronous counter register.

The Timer2 control and mode registers (T2CR, T2MR) are 8-bit registers. The comparator output is controlled by a control register (T2CR) and contains mask bits for the actions (counter reset, output toggle, timer interrupt) which can be triggered by a compare match event or by counter overflow. This architecture enables the timer for various modes.

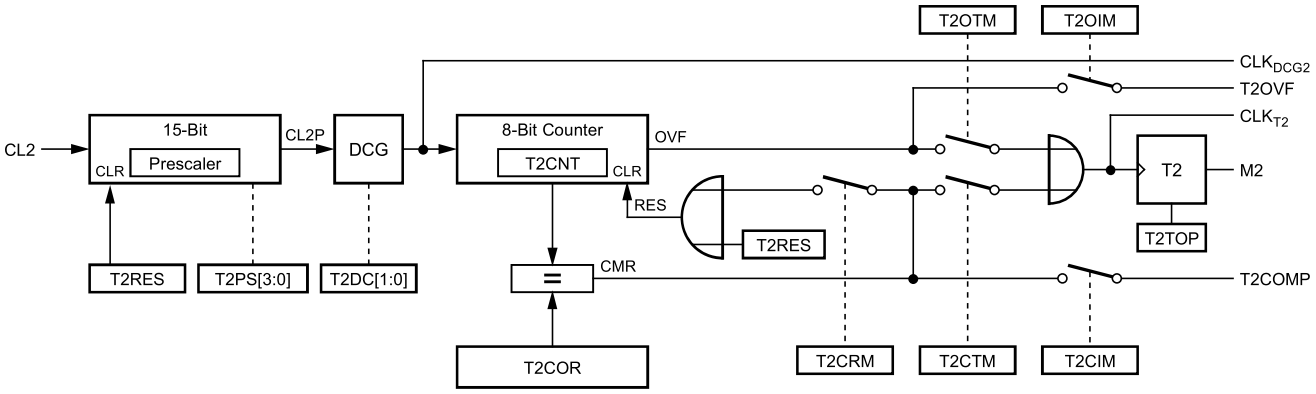
Interrupt request signals are all visible in the timer interrupt flag register (T2IFR). All interrupts are individually maskable with the timer interrupt mask register (T2IMR).

The counter input clock (CL2) can be supplied via the T clock of the system clock prescaler (CLK_T), the crystal-oscillator-based clock (CLK_{XTO4}), the internal SRC clock (CLK_{SRC}), or the internal FRC clock (CLK_{FRC}).

The output compare register (T2COR) is compared with the counter value at all times. T2COR can be modified while the timer is running.

The 15-bit prescaler works together with an additional 2-bit duty cycle generator as a 16-bit prescaler for 50% duty cycle output and it can be used as a 17-bit prescaler with odd duty cycles. The counter stage has one input signal (CL2) and five output signals (T2OVF, T2COMP, CLK_{DCG2}, M2 and CLK_{T2}). The M2, CLK_{T2}, and CLK_{DCG2} are the output clocks. T2OVF and T2COMP are the interrupt request signals of the counter stage.

Figure 3-49. Timer2 8-Bit Counter Stage



Signal description of the Timer2 (internal signals):

CL2	Prescaler input clock
CLK_{DCG2}	Timer2 input clock and duty cycle generator output clock
CLP2	Prescaler output clock and duty cycle generator (DCG) input clock
T2OVF	Timer2 counter overflow interrupt
T2COMP	Timer2 compare match interrupt
CLK_{T2}	Timer2 stage output clock
M2	Modulator output toggle flip-flop
CMR	Counter compare register match
RES	Counter reset (clear all bits)
OVF	Counter overflow

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 the T2CR register allows the programmer to initialize the toggle flip-flop output signal (M2) only if Timer2 is not running (T2ENA = “0”). The output signal (M2) is inverted with every rising edge of the input clock (CLK_{T2}). Figure 3-50 shows the toggle flip-flop (T2).

Figure 3-50. Toggle Flip-Flop (T2)

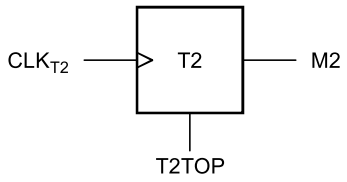
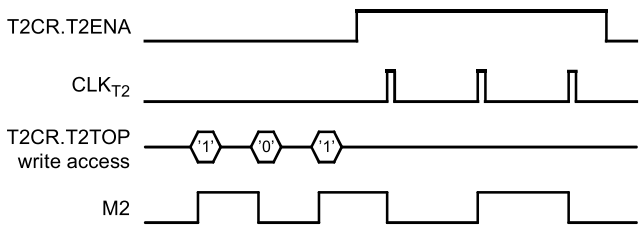


Figure 3-51. Example of Toggle Flip-Flop T2 Operation



Timer2 Register Description

T2CR – Timer2 Control Register

Bit	7	6	5	4	3	2	1	0	
	T2ENA	T2TOS	T2RES	T2TOP	-	T2CRM	T2CTM	T2OTM	T2CR
Read/Write	R/W	R/W	W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: T2ENA – Timer2 Enable

This bit controls the Timer2 block. The T2ENA bit must be written to logic one to enable Timer2. If the T2ENA bit is written to logic zero, Timer2 is disabled. Reading this bit shows the actual state of Timer2. Because internal synchronization requires 2½ asynchronous CL2 clock cycles to enable or disable Timer2, it may take some time to read a logic one after having enabled Timer2. The same applies for disabling.

Care has to be taken if the T2ENA or T2CR register is written by consecutive cbi/sbi instructions. For example, clearing the T2ENA bit with a cbi instruction followed by a cbi/sbi instruction on another bit of the T2CR re-enables the timer. (The read-modify-write sequence is still reading the T2ENA=1 and writing it back, thus enabling the timer.)

The asynchronous clock can be forced on without synchronization by writing a logic one to the T2ENA bit twice within four AVR® clock cycles. This can be useful for slow clocks, for example, when the first edge of a newly enabled oscillator should already lead to an increased counter value.

Bit 6: T2TOS – Timer2 Toggle with Start

The T2TOS bit must be written to logic one if the modulator output of Timer2 should be toggled when the timer is enabled with T2ENA. If the T2TOS bit is written to logic zero, the modulator output of Timer2 is not toggled with the timer enable.

Bit 5: T2RES – Timer2 Reset

The T2RES bit can be written to logic one to reset the prescaler and counter. This is only allowed if the timer is stopped (T2ENA=0). The T2RES bit is automatically cleared one cycle after the write.

Bit 4: T2TOP – Timer2 Toggle Output Preset

The T2TOP bit must be written to logic one to set the toggle flip-flop. Clearing the T2TOP resets the toggle flip-flop. This bit allows the programmer to preset the toggle output flip-flop in the modulator of Timer2. If the timer is stopped, this bit shows the actual value of the toggle flip-flop.

Note: If T2ENA = “1”, no output preset is possible.

Bit 3: Reserved Bit

This bit is reserved and reads as zero.

Bit 2: T2CRM – Timer2 Compare Reset Mask

The T2CRM bit must be written to logic one to enable the counter reset if a match of the counter with the compare register occurs. If the T2CRM bit is written to logic zero, the counter reset is disabled.

Bit 1: T2CTM – Timer2 Compare Toggle Mask

The T2CTM bit must be written to logic one to enable the compare toggle. If the T2CTM bit is written to logic zero, the compare toggle is disabled. A match of the counter with the compare register generates an output clock of the counter (CLK_{T2}).

Bit 0: T2OTM – Timer2 Overflow Toggle Mask

The T2OTM bit must be written to logic one to enable the overflow toggle. If the T2OTM bit is written to logic zero, the overflow toggle is disabled. A counter overflow generates an output clock of the counter (CLK_{T2}).

T2CNT – Timer2 Counter Register

Bit	7	6	5	4	3	2	1	0	
	T2CNT[7:0]								T2CNT
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The counter register (T2CNT) contains an 8-bit counter value. It can be read only when Timer2 is disabled (T2ENA= “0”). Because the timer can run on asynchronous clocks, reading the register during operation may lead to corrupted values.

T2COR – Timer2 Compare Register

Bit	7	6	5	4	3	2	1	0	
	T2COR[7:0]								T2COR
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 register contains an 8-bit value that is continuously compared with the counter value (T2CNT). A match can be used to generate a compare interrupt, a counter reset or an output clock CLK_{T2}. The compare register can be written while the timer is running. Potential compare match interrupts generated from glitches on the clock domain crossing are ignored for two AVR® clock cycles.

T2IFR – Timer2 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	T2COF	T2OFF	T2IFR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 1: T2COF – Timer2 Compare Flag

This flag is set to one during the clock cycle after the counter value has matched with the compare register. The flag (T2COF) is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

Bit 0: T2OFF – Timer2 Overflow Flag

This flag is set by the T2OVF signal when the counter reaches its maximum value (0xFF).

If the I-bit in SREG and the T2OIM bit are set in T2IMR, 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 logical one to it.

T2IMR – Timer2 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	T2CIM	T2OIM	T2IMR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 1: T2CIM – Timer2 Compare Interrupt Mask

If the T2CIM bit is written to one and the I bit in SREG is set, the Timer2 compare match interrupt is enabled. The corresponding interrupt is executed if a compare match occurs, for example, when the T2COF bit is set in the Timer2 interrupt flag register (T2IFR).

Bit 0: T2OIM – Timer2 Overflow Interrupt Mask

If the T2OIM bit is written to one and the I bit in SREG is set (one), the Timer2 overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer2 occurs, i.e., when the T2OFF bit is set in the Timer2 Interrupt Flag Register (T2IFR).

T2MR – Timer 2 Mode Register

This register should only be modified while the timer is disabled (T2CR.T2ENA=”0”). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T2DC[1:0]		T2PS[3:0]				T2CS[1:0]		T2MR
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	

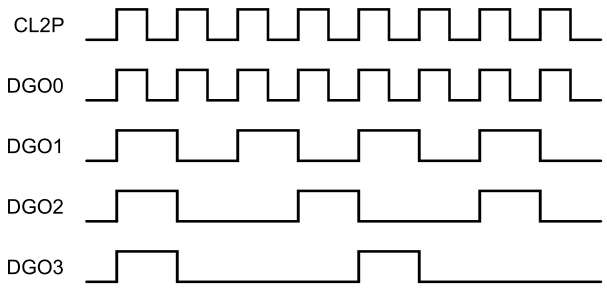
Bits 7..6: T2DC[1:0] – Timer2 Duty Cycle

The T2DC[1:0] bits select the duty cycle mode of the duty cycle generator as shown in Table 3-66.

Table 3-66. Timer2 Duty Cycle Bit Description

T2DC1	T2DC0	Function of the Duty Cycle Generator	Additional Divider Effect
0	0	Bypassed (DCG0)	1
0	1	Duty cycle 1/1 (DCG1)	2
1	0	Duty cycle 1/2 (DCG2)	3
1	1	Duty cycle 1/3 (DCG3)	4

Figure 3-52. DCG Output Signals



Bits 5..2: T2PS[3:0] – Timer2 Prescaler Select

The T2PS[3:0] bits select the prescaler value of Timer2 as shown in the following formula:

$$\text{prescaler Value} = 2^{\text{T2PS}[3:0]} \quad (44)$$

$$\text{CL2P}_{\text{Frequency}} = \frac{\text{CL2}_{\text{Frequency}}}{\text{prescaler Value}} = \frac{\text{CL2}_{\text{Frequency}}}{2^{\text{T2PS}[3:0]}} \quad (45)$$

$\text{T2PS}[3:0] \in \{0..15\}$

Bits 1..0: T2CS[1:0] – Timer 2 Clock Select

The T2CS[1:0] bits select the input clock (CL2) of Timer2 as shown in Table 3-67.

Table 3-67. Timer2 Input Clock Select Bit Description

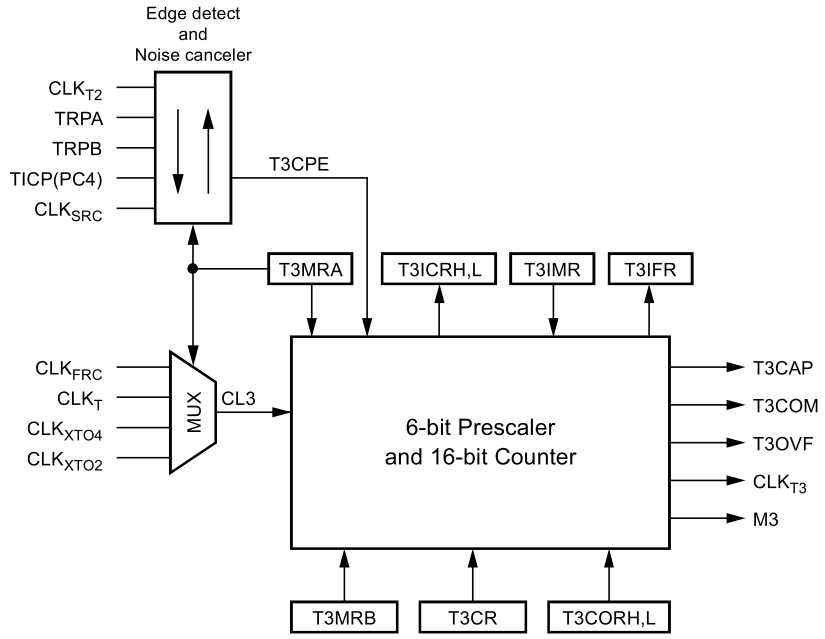
T2CS[1:0]		Input Clock (CL2) of Timer
0	0	CLK_{SRC}
0	1	CLK_{VDIV}
1	0	CLK_{T}
1	1	CLK_{XTO4}

3.8.7.4 Timer3

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

- Four different selectable input clocks
- One output compare unit
- One input capture unit
- Input capture noise canceller
- Clear timer on compare match or capture event
- Variable PWM period
- Frequency generator
- External event counter
- Three independent interrupt sources (T3CAP, T3COM, T3OVF)

Figure 3-53. Timer3 Block Diagram



Timer 3 consists of a 6-bit prescaler and a 16-bit up counter with two compare registers (T3CORH, T3CORL) and two capture registers (T3ICRH, T3ICRL). The timer can be used as an event counter, timer or signal generator. Its output can be programmed as modulator. The compare registers enable it for various modes of signal generation and modulation. The counter can be driven by several clock sources. For an external capture input signal (TICP) it has a programmable edge sensitive input that can be used as capture signal input. The current counter value is readable via its capture register after a software capture event. In the capture mode the counter value can be also captured by a programmable capture event from the Timer2 clock output (CLK_{T2}), transparent Rx path A (TRPA), transparent Rx path B (TRPB) from the UHF receiver or the slow RC oscillator clock (CLK_{SRC}).

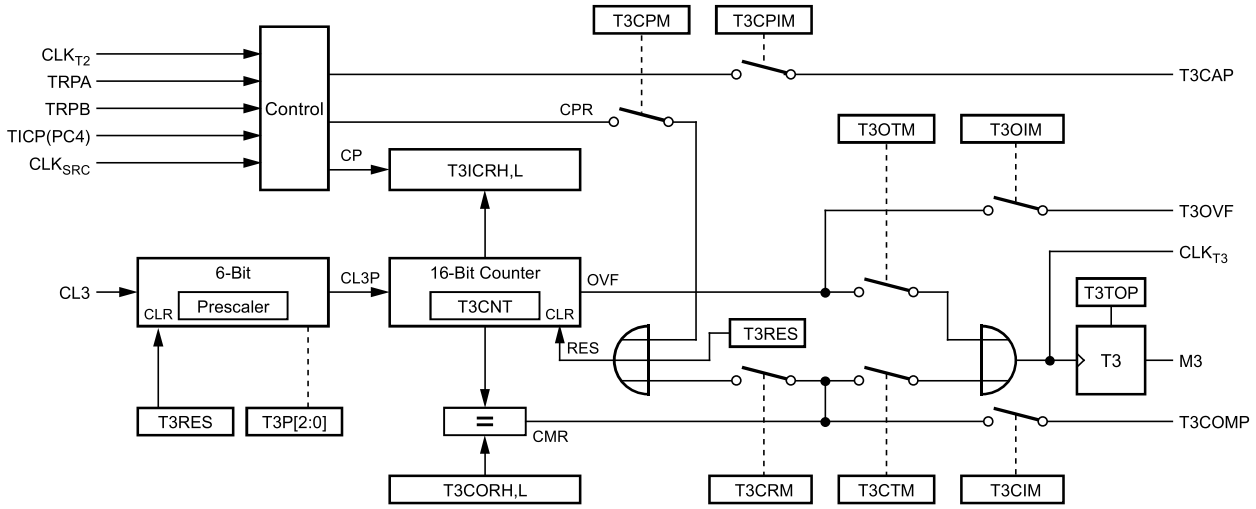
The comparator output is controlled by a control register (T3CR) which contains mask bits for actions (counter reset, output toggle, timer interrupt) that can be triggered by a compare match event, capture event or the counter overflow. The output compare registers (T3CORH, T3CORL) are compared with the counter value at all times.

Interrupt request signals are all visible in the timer interrupt flag register (T3IFR). All interrupts are individually maskable with the timer interrupt mask register (T3IMR).

The counter input clock (CL3) can be supplied via the T clock of the system clock prescaler (CLK_T), the XTO based clock (CLK_{XTO2}, CLK_{XTO4}) or the internal fast oscillator clock (CLK_{FRC}).

The 6-bit prescaler works together with the counter stage (T3CNT). The counter stage has six input signals (CL3, CLK_{T2}, TRPA, TRPB, TICP, CLK_{SRC}) and four output signals (T3CAP, T3OVF, T3COM, and CLK_{T3}). The CL3 supplies the 6-bit prescaler with a clock. The CLK_{T2}, the UHF-receiver-transparent Rx path A,B outputs (TRPA, TRPB), the external input capture signal (TICP), or CLK_{SRC} can capture the counter value in the capture register. The CLK_{T3} is the output clock and T3CAP, T3OVF and T3COM are the interrupt request signals of the counter stage.

Figure 3-54. Timer3 Counter Stage



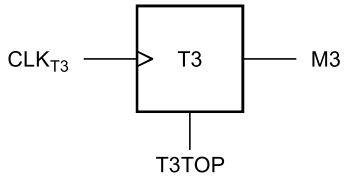
Signal description of Timer3 (internal signals):

CLK_{T2}	Timer2 counter stage clock input
TRPA	Transparent UHF receiver data path A input signal
TRPB	Transparent UHF receiver data path B input signal
TICP	Timer3 external input capture
CLK_{SRC}	Slow RC oscillator clock input
CL3	Selected prescaler input clock
CL3P	Counter input clock
CMR	Counter compare register match
RES	Counter reset (clear all bits)
OVF	Counter overflow
CP	Capture event signal
CPR	Capture event reset signal
CLK_{T3}	Counter stage clock output
T3CAP	Timer3 capture event interrupt
T3OVF	Timer3 counter overflow interrupt
T3COM	Timer3 compare match interrupt
M3	Modulator output toggle flip-flop

Modulator Toggle Flip-Flop (T3)

The toggle flip-flop (T3) consists of a flip-flop with a preset input signal (T3TOP), an input clock (CLK_{T3}) and an output signal (M3). The T3TOP bit in the T3CR register allows the programmer to initialize the toggle flip-flop output (M3) only if Timer3 is not running (T3ENA = “0”). The output signal (M3) is inverted with every rising edge of the input clock (CLK_{T3}). Figure 3-55 shows the toggle flip-flop (T3).

Figure 3-55. Toggle Flip-Flop T3



Timer3 Register Description

T3CR – Timer3 Control Register

Bit	7	6	5	4	3	2	1	0	
	T3ENA	**T3TOS**	**T3RES**	**T3TOP**	**T3CPRM**	**T3CRM**	**T3CTM**	**T3OTM**	**T3CR**
Read/Write	R/W	R/W	W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: T3ENA – Timer3 Enable

This bit controls the Timer3 block. The T3ENA bit must be written to logic one to enable Timer3. If the T3ENA bit is written to logic zero, Timer3 is disabled. Reading this bit shows the actual state of Timer3. Because internal synchronization requires 2½ asynchronous CL3 clock cycles to enable or disable Timer3, it may take some time to read a logic one after having enabled Timer3. The same applies for disabling.

Bit 6: T3TOS – Timer3 Toggle with Start

The T3TOS bit must be written to logic one if the modulator output of Timer3 should toggle when the timer is enabled with T3ENA. If the T3TOS bit is written to logic zero, the modulator output of Timer3 is not toggled on timer enable.

Bit 5: T3RES – Timer3 Reset

The T3RES bit can be written to logic one to reset the prescaler and counter. This is only allowed if the timer is stopped (T3ENA=0). The T3RES bit is automatically cleared one cycle after the write.

Bit 4: T3TOP – T3 Toggle Output Preset

The T3TOP bit must be written to logic one to set the toggle flip-flop. If the T3TOP bit is written to logic zero, it resets the toggle flip-flop. This bit allows the programmer to preset the toggle output flip-flop in the modulator of Timer3.

Note: If T3ENA = “1”, no output preset is possible.

Bit 3: T3CPRM – Timer3 Capture Reset Mask

The T3CPRM bit must be written to logic one to enable the counter reset if an internal/external capture event occurs. If the T3CPRM bit is written to logic zero, the counter reset is disabled.

Bit 2: T3CRM – Timer3 Compare Reset Mask

The T3CRM bit must be written to logic one to enable the counter reset if a match of the counter with the compare register (T3COR) occurs. If the T3CRM bit is written to logic zero, the counter reset is disabled.

Bit 1: T3CTM – Timer3 Compare Toggle Mask

The T3CTM bit must be written to logic one to enable the compare toggle. A match of the counter with the compare register (T3COR) toggles the output flip-flop in the modulator of Timer3. If the T3CTM bit is written to logic zero, the compare toggle is disabled.

Bit 0: T3OTM – Timer3 Overflow Toggle Mask

The T3OTM bit must be written to logic one to enable the overflow toggle. A counter overflow generates an output clock of the counter (CLK_{T3}). If the T3OTM bit is written to logic zero, the overflow toggle is disabled.

T3MRA – Timer3 Mode Register A

This register should be modified only while the timer is disabled (T3CR.T3ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	T3PS[2:0]			T3CS[1:0]		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..5: Reserved Bits

These bits are reserved and read as zero.

Bits 4..2: T3PS[2:0] – Timer3 Prescaler Select

The T3PS[2:0] bits select the prescaler values of Timer3 as shown in Table 3-68.

Table 3-68. Timer3 Prescaler Value Select Bit Description

T3PS[2:0]			Prescaler Value
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	Reserved

Bits 1..0: T3CS[1:0] – Timer 3 Clock Select

The T3CS[1:0] bits select the input clock (CL3) of Timer3 as shown in Table 3-69.

Table 3-69. Timer3 Input Clock Select Bit Description

T3CS[1:0]		Input Clock (CL3) of the Prescaler
0	0	CLK_{FRC}
0	1	CLK_{T}
1	0	CLK_{XTO4}
1	1	CLK_{XTO2}

T3MRB – Timer3 Mode Register B

This register should be modified only while the timer is disabled (T3CR.T3ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T3ICS[2:0]			T3CE[1:0]		T3CNC	T3SCE	-	T3MRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..5: T3ICS[2:0] – Timer3 Input Capture Select

The T3ICS[2:0] bits select the input capture signal of Timer3 as shown in Table 3-70 on page 250.

Table 3-70. Timer3 Input Capture Signal Select Bit Description

T3ICS[2:0]			Description
0	0	0	CLK_{T2}
0	0	1	TRPA
0	1	0	TRPB
0	1	1	TICP
1	0	0	CLK_{SRC}
1	0	1	Reserved (CLK_{SRC})
1	1	0	Reserved (CLK_{SRC})
1	1	1	Reserved (CLK_{SRC})

Bits 4..3: T3CE[1:0] – Timer3 Capture Edge Select

The T3CE1 and T3CE0 bits select the edge from all input signals of Timer3 as shown in Table 3-71.

Table 3-71. Timer3 Capture Edge Select Bit Description

T3CE[1:0]		Input Capture Edge Signal of Timer3
0	0	Disable edge detect
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Bit 2: T3CNC – Timer3 Input Capture Noise Canceller

Setting this bit to one activates the input capture noise canceller. When the noise canceller is activated, the input from the input capture pin is filtered. To change its output, the filter function requires four successive samples of the input capture pin of equal value. The input capture is therefore delayed by four counter clock (CL3) cycles when the noise canceller is enabled.

Bit 1: T3SCE – Timer3 Software Capture Enable

The T3SCE bit must be written to logic one to enable 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 while it is running.

Bit 0: Reserved Bit

This bit is reserved and reads as zero.

T3IFR – Timer3 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	T3ICF	T3COF	T3OFF	T3IFR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 2: T3ICF – Timer3 Input Capture Flag

This flag is set when a capture event occurs on the selected capture source and indicates that the counter value has been transferred to the capture register (T3ICR). If the I bit in SREG and the T3CPIM bit in the T3IMR register are set, the MCU jumps to the corresponding interrupt vector. T3ICF is automatically cleared when the interrupt routine is executed. Alternatively, T3ICF can be cleared by writing a logic one to this bit location.

Bit 1: T3COF – Timer3 Compare Flag

This flag is set during the clock cycle after the Timer3 counter value has matched with the compare register. If the I bit in SREG and the T3CIM bit in the T3IMR register are set, the MCU jumps to the corresponding interrupt vector. The flag (T3COF) is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

Bit 0: T3OFF – Timer3 Overflow Flag

This flag is set by the T3OVF signal when the counter reaches its maximum value (0xFFFF).

If the I bit in SREG and the T3OIM bit in the T3IMR register are set, 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 logical one to it.

T3IMR – Timer 3 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	T3CPIM	T3CIM	T3OIM	T3IMR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 2: T3CPIM – Timer3 Capture Interrupt Mask

If this bit is written to one, and the I flag in SREG is set (interrupts globally enabled), the Timer3 input capture interrupt is enabled. The corresponding interrupt vector is executed when the T3ICF flag, located in T3IFR, is set.

Bit 1: T3CIM – Timer3 Compare Interrupt Mask

If the T3CIM bit is written to one and the I bit in SREG is set, the Timer3 compare match interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer3 occurs, and when the T3COF bit is set in the Timer3 interrupt flag register (T3IFR).

Bit 0: T3OIM – Timer3 Overflow Interrupt Mask

If the T3OIM bit is written to one and the I bit in SREG is set, the Timer3 overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer3 occurs, and when the T3OFF bit is set in the Timer3 interrupt flag register (T3IFR).

T3ICRH – Timer3 Input Capture Register High Byte

The input capture register T3ICR is updated with the counter value (T3CNT) each time a capture trigger event occurs on the selected capture source. Trigger signal and edge selection is done in T3MRA. The T3CNT content can be read via the capture register after a software capture event.

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

T3ICRL – Timer3 Input Capture Register Low Byte

Bit	7	6	5	4	3	2	1	0	
	T3ICR[7:0]								T3ICRL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

T3CORH – Timer3 Compare Register High Byte

The H/L compare registers contain a 16-bit value that is continuously compared with the counter value (T3CNT). The T3CORH represents the upper byte of the compare value. A match can be used to generate a compare interrupt, a counter reset or an output clock.

This register should be modified only while the timer is disabled (T3CR.T3ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T3CORH[7:0]								T3CORH
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	

T3CORL – Timer3 Compare Register Low Byte

The T3CORL represents the lower byte of the compare value.

This register should be modified only while the timer is disabled (T3CR.T3ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T3CORL[7:0]								T3CORL
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	

T3CNTH – Timer3 Counter Register High Byte

Bit	7	6	5	4	3	2	1	0	
	T3CNTH[7:0]								T3CNTH
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The counter register (T3CNTH) contains the upper 8-bit of the counter value. It should be read only when the timer is disabled (T3CR.T3ENA= “0”). Due to the asynchronous implementation it can result in unpredictable read values if ignored.

T3CNTL – Timer3 Counter Register Low Byte

Bit	7	6	5	4	3	2	1	0	
	T3CNTL[7:0]								T3CNTL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The counter register (T3CNTL) contains the lower 8-bit of the counter value. It should be read only when the timer is disabled (T3CR.T3ENA= “0”). Due to the asynchronous implementation it can result in unpredictable read values if ignored.

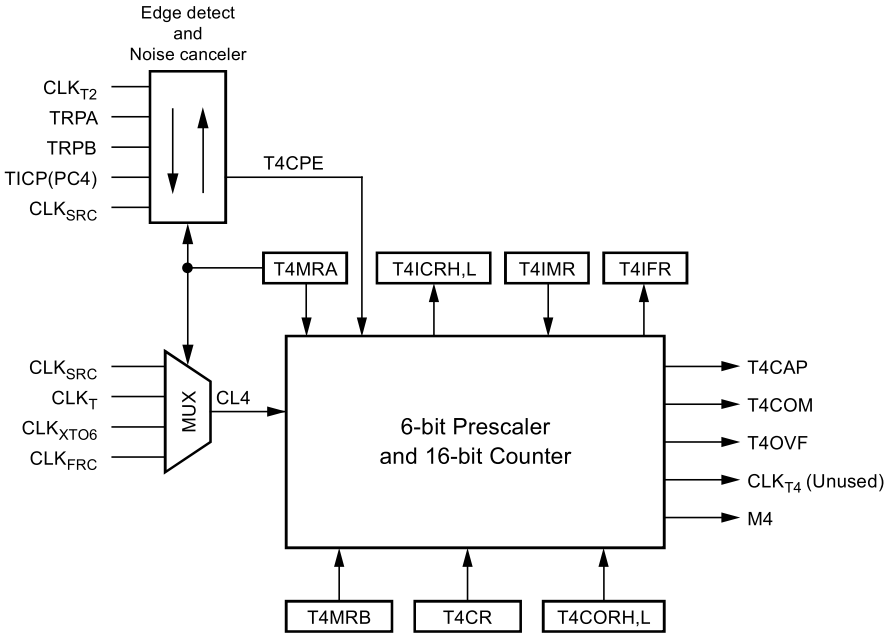
3.8.7.5 Timer4

Timer4 is a copy of Timer3.

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

- Four different selectable input clocks
- One output compare unit
- One input capture unit
- Input capture noise canceller
- Clear timer on compare match or capture event
- Variable PWM period
- Frequency generator
- External event counter
- Three independent interrupt sources (T4CAP, T4COM, T4OVF)

Figure 3-56. Timer4 Block Diagram



Timer 4 consists of a 6-bit prescaler and a 16-bit up counter with two compare registers (T4CORH, T4CORL) and two capture registers (T4ICRH, T4ICRL). The timer can be used as an event counter, timer, or signal generator. Its output can be programmed as modulator. The compare registers enable it for various modes of signal generation and modulation. The counter can be driven by several clock sources. It has a programmable edge sensitive input logic for the capture input signal. The current counter value is readable via its capture register after a software capture event. In the capture mode the counter value can be also captured by a programmable capture event from the Timer2 clock output (CLK_{T2}), transparent Rx path A (TRPA), transparent Rx path B (TRPB) from the UHF receiver or the slow RC oscillator clock (CLK_{SRC}).

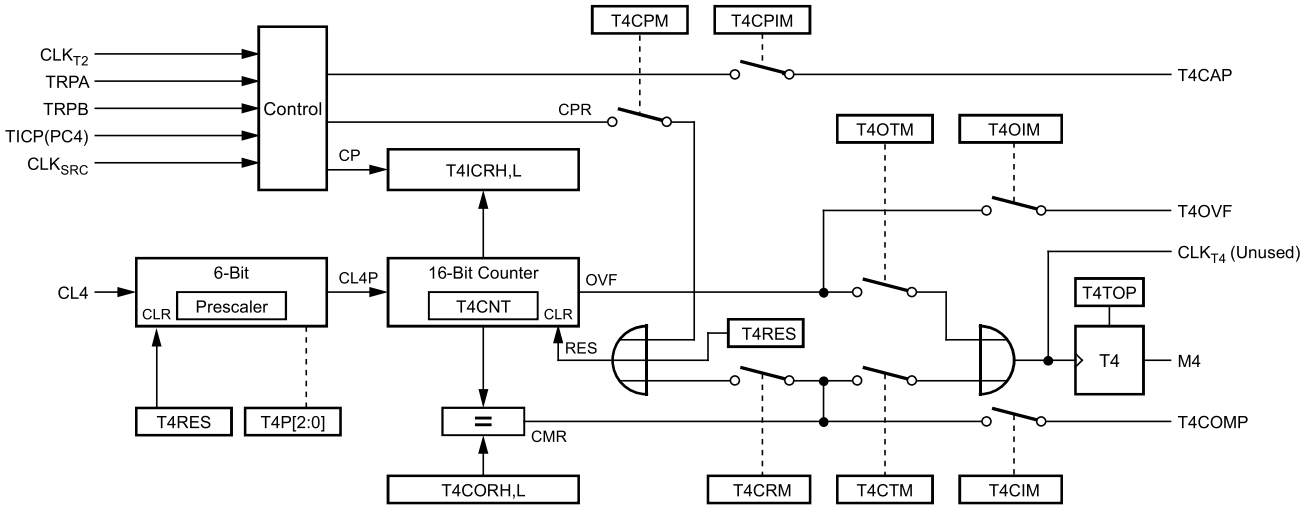
The comparator output is controlled by a control register (T4CR) which contains mask bits for the actions (counter reset, output toggle, timer interrupt) that can be triggered by a compare match event, capture event or the counter overflow. The output compare registers (T4CORH, T4CORL) are compared with the counter value at all times.

Interrupt request signals are all visible in the timer interrupt flag register (T4IFR). All interrupts are individually maskable with the timer interrupt mask register (T4IMR).

The counter input clock (CL4) can be supplied via the T clock of the system clock prescaler (CLK_T), the XTO based clock (CLK_{XTO6}), the internal SRC-clock (CLK_{SRC}), or the internal fast oscillator clock (CLK_{FRC}).

The 6-bit prescaler works together with the counter stage (T4CNT). The counter stage has six input signals (CL4, CLK_{T2}, TRPA, TRPB, TICP, CLK_{SRC}) and four output signals (T4CAP, T4OVF, T4COM, and CLK_{T4}). The CL4 supplies the 6-bit prescaler with a clock. The CLK_{T2}, the UHF-receiver transparent Rx path A/B outputs (TRPA, TRPB), the external input capture signal (TICP), or CLK_{SRC} can capture the counter value in the capture register. The CLK_{T4} is the output clock and T4CAP, T4OVF and T4COM are the interrupt request signals of the counter stage.

Figure 3-57. Timer4 Counter Stage



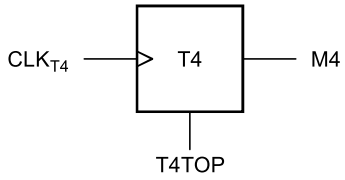
Signal description of Timer4 (internal signals):

CLK_{T2}	Timer2 counter stage clock input
TRPA	Transparent UHF receiver data path A input signal
TRPB	Transparent UHF receiver data path B input signal
TICP	Timer4 external input capture
CLK_{SRC}	Slow RC oscillator clock input
CL4	Selected prescaler input clock
CL4P	Counter input clock
CMR	Counter compare register match
RES	Counter reset (clear all bits)
OVF	Counter overflow
CP	Capture event signal
CPR	Capture event reset signal
CLK_{T4}	Counter stage clock output
T4CAP	Timer4 capture event interrupt
T4OVF	Timer4 counter overflow interrupt
T4COM	Timer4 compare match interrupt
M4	Modulator output toggle flip-flop

Modulator Toggle Flip-Flop (T4)

The toggle flip-flop (T4) consists of a flip-flop with a preset input signal (T4TOP), an input clock (CLK_{T4}) and an output signal (M4). The T4TOP bit in the T4CR register allows the programmer to initialize the toggle flip-flop output (M4) only if Timer4 is not running (T4ENA = “0”). The output signal (M4) is inverted with every rising edge of the input clock (CLK_{T4}). Figure 3-58 on page 255 shows the toggle flip-flop (T4).

Figure 3-58. Toggle Flip-Flop T4



Timer4 Register Description

T4CR – Timer 4 Control Register

Bit	7	6	5	4	3	2	1	0	
	T4ENA	**T4TOS**	**T4RES**	**T4TOP**	**T4CPRM**	**T4CRM**	**T4CTM**	**T4OTM**	**T4CR**
Read/Write	R/W	R/W	W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: T4ENA – Timer4 Enable

This bit controls the Timer4 block. The T4ENA bit must be written to logic one to enable Timer4. If the T4ENA bit is written to logic zero, Timer4 is disabled. Reading this bit shows the actual state of Timer4. Because internal synchronization requires 2½ asynchronous CL4 clock cycles to enable or disable Timer4, it may take some time to read a logic one after having enabled Timer4. The same applies for disabling.

Bit 6: T4TOS – Timer4 Toggle with Start

The T4TOS bit must be written to logic one if the modulator output of Timer4 should toggle when the timer is enabled with T4ENA. If the T4TOS bit is written to logic zero, the modulator output of Timer4 is not toggled on timer enable.

Bit 5: T4RES – Timer4 Reset

The T4RES bit can be written to logic one to reset the prescaler and counter. This is only allowed if the timer is stopped (T4ENA=0). The T4RES bit is automatically cleared one cycle after the write.

Bit 4: T4TOP – T4 Toggle Output Preset

The T4TOP bit must be written to logic one to set the toggle flip-flop. If the T4TOP bit is written to logic zero, it resets the toggle flip-flop. This bit allows the programmer to preset the toggle output flip-flop in the modulator of Timer4.

Note: If T4ENA = “1”, no output preset is possible.

Bit 3: T4CPRM – Timer4 Capture Reset Mask

The T4CPRM bit must be written to logic one to enable the counter reset if an internal/external capture event occurs. If the T4CPRM bit is written to logic zero, the counter reset is disabled.

Bit 2: T4CRM – Timer4 Compare Reset Mask

The T4CRM bit must be written to logic one to enable the counter reset if a match of the counter with the compare register (T4COR) occurs. If the T4CRM bit is written to logic zero, the counter reset is disabled.

Bit 1: T4CTM – Timer4 Compare Toggle Mask

The T4CTM bit must be written to logic one to enable the compare toggle. A match of the counter with the compare register (T4COR) toggles the output flip-flop in the modulator of Timer4. If the T4CTM bit is written to logic zero, the compare toggle is disabled.

Bit 0: T4OTM – Timer4 Overflow Toggle Mask

The T4OTM bit must be written to logic one to enable the overflow toggle. A counter overflow generates an output clock of the counter (CLK_{T4}). If the T4OTM bit is written to logic zero, the overflow toggle is disabled.

T4MRA – Timer4 Mode Register A

This register should be modified only while the timer is disabled (T4CR.T4ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	T4PS[2:0]			T4CS[1:0]		T4MRA
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: Reserved Bits

These bits are reserved and read as zero.

Bits 4..2: T4PS[2:0] – Timer 4 Prescaler Select

The T4PS[2:0] bits select the prescaler value of Timer4 as shown in Table 3-72.

Table 3-72. Timer 4 Prescaler Value Select Bit Description

T4PS[2:0]			Prescaler Value
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	Reserved

Bits 1..0: T4CS[1:0] – Timer 4 Clock Select

The T4CS[1:0] bits select the input clock (CL4) of Timer4 as shown in Table 3-73.

Table 3-73. Timer4 Input Clock Select Bit Description

T4CS[1:0]		Input Clock (CL4) of the Prescaler
0	0	CLK_{SRC}
0	1	CLK_{T}
1	0	CLK_{XTO6}
1	1	CLK_{FRC}

T4MRB – Timer4 Mode Register B

This register should be modified only while the timer is disabled (T4CR.T4ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T4ICS[2:0]			T4CE[1:0]		T4CNC	T4SCE	-	T4MRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..5: T4ICS[2:0] – Timer4 Input Capture Select

The T4ICS[2:0] bits select the input capture signal of Timer4, shown in Table 3-74 on page 257.

Table 3-74. Timer4 Input Capture Signal Select Bit Description

T4ICS[2:0]			Description
0	0	0	CLK_{T2}
0	0	1	TRPA
0	1	0	TRPB
0	1	1	TICP
1	0	0	CLK_{SRC}
1	0	1	Reserved (CLK_{SRC})
1	1	0	Reserved (CLK_{SRC})
1	1	1	Reserved (CLK_{SRC})

Bits 4..3: T4CE[1:0] – Timer4 Capture Edge Select

The T4CE1 and T4CE0 bits select the edge from all input signals of Timer4 as shown in Table 3-75.

Table 3-75. Timer4 Capture Edge Select Bit Description

T4CE[1:0]		Input Capture Edge Signal of Timer4
0	0	Disable edge detect
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Bit 2: T4CNC – Timer4 Input Capture Noise Canceller

Setting this bit to one activates the input capture noise canceller. When the noise canceller is activated, the input from the selected capture source is filtered. To change its output, the filter function requires four successive samples of the input capture signal of equal value. The input capture is therefore delayed by four counter clock (CL4) cycles when the noise canceller is enabled.

Bit 1: T4SCE – Timer4 Software Capture Enable

The T4SCE bit must be written to logic one to enable a software capture event. The T4SCE bit is cleared after the counter value is saved in the capture register. The Timer4 counter value is readable via its capture register while it is running.

Bit 0: Reserved Bit

This bit is reserved and reads as zero.

T4IFR – Timer4 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	T4ICF	T4COF	T4OFF	T4IFR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 2: T4ICF – Timer4 Input Capture Flag

This flag is set when a capture event occurs on the selected capture source, indicating that the counter value has been transferred to the capture register (T4ICR). If the I bit in SREG and the T4CPIM bit in T4IMR register are set, the MCU jumps to the corresponding interrupt vector. T4ICF is automatically cleared when the interrupt routine is executed. Alternatively, T4ICF can be cleared by writing a logic one to this bit location.

Bit 1: T4COF – Timer4 Compare Flag

This flag is set during the clock cycle after the Timer4 counter value has matched with the compare register. If the I bit in SREG and the T4CIM bit in the T4IMR register are set, the MCU jumps to the corresponding interrupt vector. The flag (T4COF) is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

Bit 0: T4OFF – Timer4 Overflow Flag

This flag is set by the T4OVF signal when the counter reaches its maximum value (0xFFFF).

If the I bit in SREG and the T4OIM bit in the T4IMR register are set, 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 logical one to it.

T4IMR – Timer4 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	T4CPIM	T4CIM	T4OIM	T4IMR
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: Reserved Bits

These bits are reserved and read as zero.

Bit 2: T4CPIM – Timer4 Capture Interrupt Mask

If this bit is written to one and the I flag in SREG is set (interrupts globally enabled), the Timer4 input capture interrupt is enabled. The corresponding interrupt vector is executed when the T4ICF Flag, located in T4IFR, is set.

Bit 1: T4CIM – Timer4 Compare Interrupt Mask

If the T4CIM bit is written to one and the I bit in SREG is set, the Timer4 compare match interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer4 occurs, and when the T4COF bit is set in the Timer4 interrupt flag register (T4IFR).

Bit 0: T4OIM – Timer4 Overflow Interrupt Mask

If the T4OIM bit is written to one and the I bit in SREG is set, the Timer4 overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer4 occurs, and when the T4OFF bit is set in the Timer4 interrupt flag register (T4IFR).

T4ICRH – Timer4 Input Capture Register High Byte

The input capture register T4ICR is updated with the counter value (T4CNT) each time a capture trigger event occurs on the selected capture source. Trigger signal and edge selection is done in T4MRA. The T4CNT content can be read via the capture register after a software capture event.

Bit	7	6	5	4	3	2	1	0	
	T4ICR[15:8]								T4ICRH
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

T4ICRL – Timer4 Input Capture Register Low Byte

Bit	7	6	5	4	3	2	1	0	
	T4ICR[7:0]								T4ICRL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

T4CORH – Timer4 Compare Register High Byte

The compare registers H/L contain a 16-bit value that is continuously compared with the counter value (T4CNT). The T4CORH represents the upper byte of the compare value. A match can be used to generate a compare interrupt, a counter reset or an output clock CLK_{T4}.

This register should be modified only while the timer is disabled (T4CR.T4ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T4CORH[7:0]								T4CORH
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	

T4CORL – Timer4 Compare Register Low Byte

The T4CORL represents the lower byte of the compare value.

This register should be modified only while the timer is disabled (T4CR.T4ENA=0). Modifying the bits during operation leads to unpredictable operation.

Bit	7	6	5	4	3	2	1	0	
	T4CORL[7:0]								T4CORL
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	

T4CNTH – Timer4 Counter Register High Byte

Bit	7	6	5	4	3	2	1	0	
	T4CNTH[7:0]								T4CNTH
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The counter register (T4CNTH) contains the upper 8-bit of the counter value. It should be read only when the timer is disabled (T4ENA= “0”). Due to the asynchronous implementation it can result in unpredictable read values if ignored.

T4CNTL – Timer4 Counter Register Low Byte

Bit	7	6	5	4	3	2	1	0	
	T4CNTL[7:0]								T4CNTL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The counter register (T4CNTL) contains the lower 8-bit of the counter value. It should be read only when the timer is disabled (T4ENA= “0”). Due to the asynchronous implementation, it can result in unpredictable read values if ignored.

3.8.7.6 Timer5

The 16-bit Timer5 unit allows accurate program execution timing (event management).

The main features are:

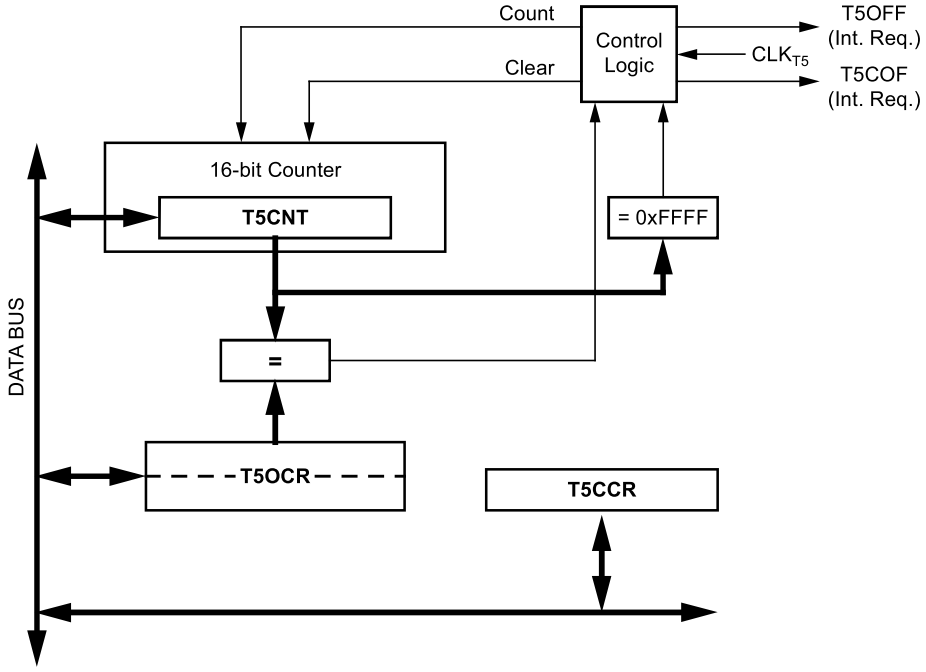
- Synchronous operation
- One output compare unit
- Clear timer on compare match (Auto Reload)
- Two independent interrupt sources (T5OFF and T5COF)

Overview

A simplified block diagram of the synchronous 16-bit Timer5 is shown in Figure 3-59. The CPU-accessible I/O registers, including I/O bits and I/O pins, are shown in bold.

The PRT5 bit in ”PRR1 – Power reduction register 1” must be written to “0” to enable the Timer5 module.

Figure 3-59. Timer5 Block Diagram



The counter register (T5CNT) and the output compare register (T5OCR) are both 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. The Timer5 control register (T5CCR) is an 8-bit register and has no CPU access restrictions. Interrupt requests (abbreviated to Int.Req. in the figure) signals are visible in the Timer5 interrupt flag register (T5IFR). Both interrupts are individually masked with the timer interrupt mask register (T5IMR). T5IFR and T5IMR are not shown in the figure.

Timer5 is clocked internally via the prescaler. The clock select logic block controls which clock source the counter uses to increment its value. The counter is inactive when no clock source is selected. The output from the clock select logic is referred to as the timer clock (CLK_{T5}).

The output compare register (T5OCR) is compared with the counter value at all time. The compare match event sets the compare match flag (T5COF), which can be used to generate an output compare interrupt request.

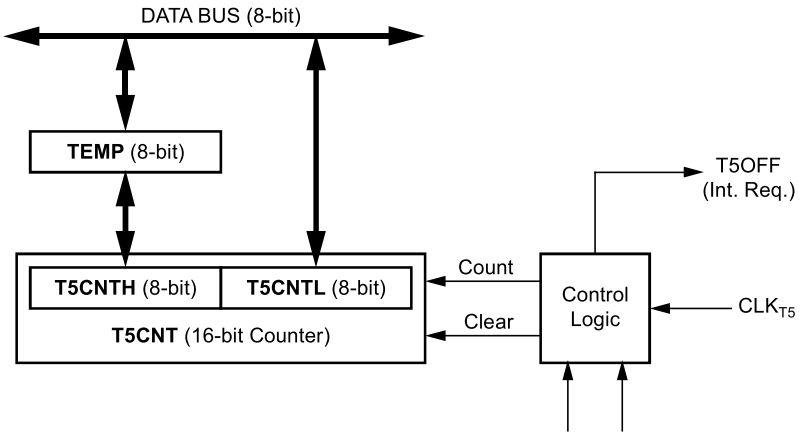
Timer/Counter5 Clock Sources

Timer5 is clocked by an internal clock source. The clock source is selected by the clock select logic which is controlled by the clock select (T5CS[2:0]) bits located in the Timer5 control register (T5CCR). For details on clock sources and the prescaler, see Section "Timer5 Prescaler" on page 262.

Counter Unit

The main part of the 16-bit timer is the programmable 16-bit counter unit. Figure 3-60 shows a block diagram of the counter and its surroundings.

Figure 3-60. Timer5 Counter Unit Block Diagram



Signal description (internal signals):

Count	Increment T5CNT by 1.
Clear	Clear T5CNT (set all bits to zero).
CLK_{T5}	Timer5 clock.

The 16-bit counter is mapped into two 8-bit I/O memory locations: counter high (T5CNTH) containing the upper eight bits of the counter, and counter low (T5CNTL) containing the lower eight bits. The T5CNTH register can only be indirectly accessed by the CPU. When the CPU does an access to the T5CNTH I/O location, the CPU accesses the high byte temporary register (TEMP). The temporary register is updated with the T5CNTH value when the T5CNTL is read, and T5CNTH is updated with the temporary register value when T5CNTL is written. This allows the CPU to read or write the entire 16-bit counter value within one clock cycle via the 8-bit data bus. It is important to note that in some circumstances writing to the T5CNT register while the counter is counting produces unpredictable results. The special cases are described in the sections where they are of importance.

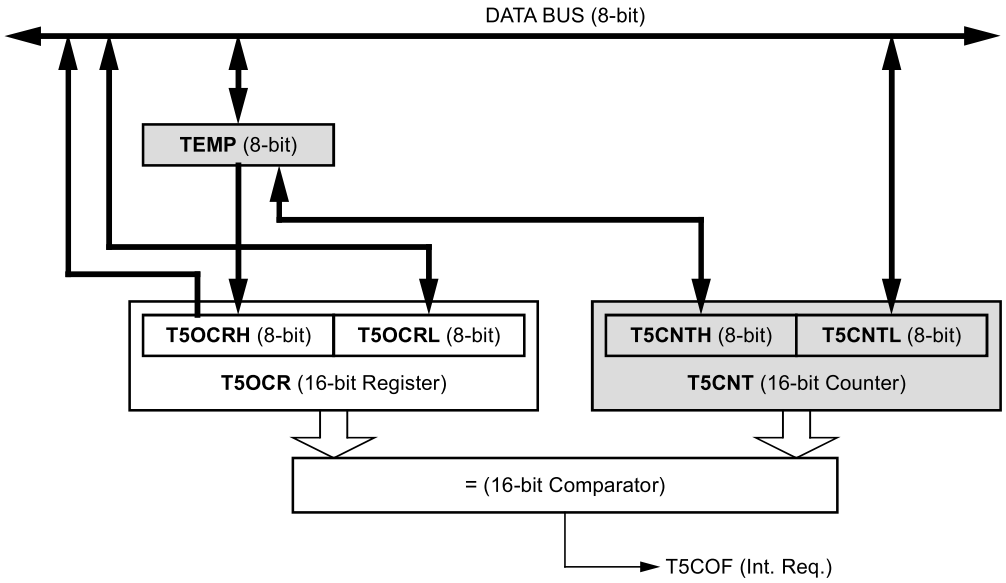
Depending on the operating modes used, the counter is cleared or incremented at each timer clock (CLK_{T5}). The CLK_{T5} is generated from an internal clock source, selected by the clock select bits (T5CS[2:0]). When no clock source is selected (T5CS[2:0] = "0b000"), the timer is stopped. However, the T5CNT value can be accessed by the CPU, independent of whether CLK_{T5} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

Output Compare Unit

The 16-bit comparator continuously compares T5CNT with the output compare register (T5OCR). If T5CNT equals T5OCR, the comparator signals a match. A match sets the output compare flag (T5COF) at the next timer clock cycle. If enabled (T5CIM = 1), the output compare flag generates an output compare interrupt. The T5COF flag is automatically cleared when the interrupt is executed. Alternatively, the T5COF flag can be cleared by software by writing a logical "1" to its I/O bit location.

Figure 3-61 on page 262 shows a block diagram of the output compare unit. The elements of the block diagram that are not directly a part of the output compare unit are shaded in gray.

Figure 3-61. Timer5 Output Compare Unit



Compare Match Blocking by T5CNT Write

All CPU writes to the T5CNT register blocks any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows T5OCR to be initialized to the same value as T5CNT without triggering an interrupt when the Timer5 clock is enabled.

Using the Output Compare Unit

Because writing T5CNT in any operating mode blocks all compare matches for one timer clock cycle, changing T5CNT poses risks when using any of the output compare channels, regardless of whether the timer is running or not. If the value written to T5CNT equals the T5OCR value, the compare match is missed.

Timer5 Prescaler

Internal Clock Source

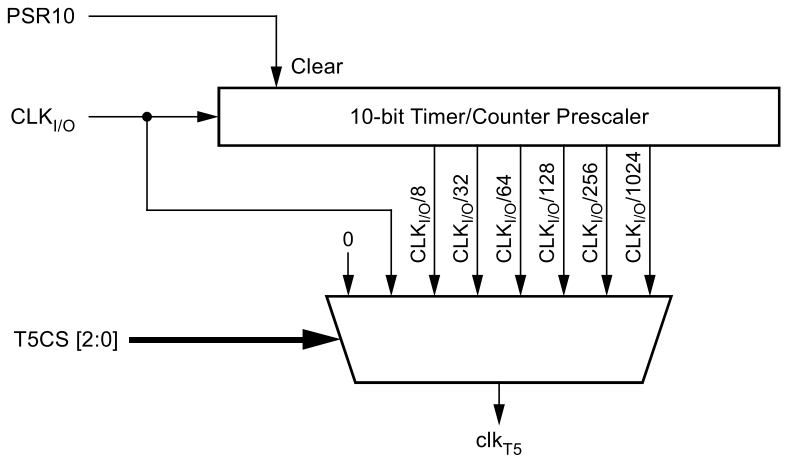
Timer5 can be clocked directly by the system clock (by setting the T5CS[2:0] = “0b001”). This provides the fastest operation, with a maximum clock frequency equal to system clock frequency ($f_{CLK\_I/O}$). Alternatively, one of the taps from the prescaler can be used as a clock source by setting the T5CS[2:0]. See Table 3-76 on page 263 for Timer5 settings. The prescaled clock has a frequency of either $f_{CLK\_I/O}/8$, $f_{CLK\_I/O}/32$, $f_{CLK\_I/O}/64$, $f_{CLK\_I/O}/128$, $f_{CLK\_I/O}/256$, or $f_{CLK\_I/O}/1024$.

Prescaler Reset

The prescaler is free-running, meaning it operates independently of the clock select logic of Timer5. Since the prescaler is not affected by the clock select of the timer, the state of the prescaler has implications for situations where a prescaled clock is used. One example of prescaling artifacts occurs when the timer is enabled and clocked by the prescaler (6 > T5CS[2:0] > 1). The number of system clock cycles from when the timer is enabled to the first count occurs can be from 1 to N+1 system clock cycles, where N equals the prescaler divisor.

It is possible to use the prescaler reset for synchronizing Timer5 to program execution.

Figure 3-62. Timer5 Prescaler



Timer5 Register Description

T5CCR – Timer5 Configuration and Control Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	T5CTC	T5CS[2:0]			T5CCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7..4: Reserved Bits

These bits are reserved and read as zero.

Bit 3: T5CTC – Clear Counter on Compare Match

When the T5CTC control bit is set, the counter is reset to 0x00 in the CPU clock cycle after a compare match.

Bit 2..0: T5CS[2:0]: Clock Select

The three clock select bits select the clock source to be used by the timer according to Table 3-76.

Table 3-76. T5CS[2:0] – Clock Select Bit Description

T5CS[2:0]			Description
0	0	0	No clock source (Timer/counter stopped)
0	0	1	CLK$_{I/O}$/1 (No prescaling)
0	1	0	CLK$_{I/O}$/8 (From prescaler)
0	1	1	CLK$_{I/O}$/32 (From prescaler)
1	0	0	CLK$_{I/O}$/64 (From prescaler)
1	0	1	CLK$_{I/O}$/128 (From prescaler)
1	1	0	CLK$_{I/O}$/256 (From prescaler)
1	1	1	CLK$_{I/O}$/1024 (From prescaler)

T5IFR – Timer5 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	T5COF	T5OFF	T5IFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7..2: Reserved Bits

These bits are reserved and read as zero.

Bit 1: T5COF – Timer5 Output Compare Match Flag

This flag is set in the timer clock cycle after the counter (T5CNT) value matches the output compare register (T5OCR).

T5COF is automatically cleared when the output compare match interrupt vector is executed. Alternatively, T5COF can be cleared by writing a logic “1” to its bit location.

Bit 0: T5OFF – Timer5 Overflow Flag

T5OFF flag is set when the timer overflows.

T5OFF is automatically cleared when the Timer5 overflow interrupt vector is executed. Alternatively, T5OFF can be cleared by writing a logic “1” to its bit location.

T5IMR – Timer5 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	T5CIM	T5OIM	T5IMR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7..2: Reserved Bits

These bits are reserved and read as zero.

Bit 1: T5CIM – Timer5 Output Compare Match Interrupt Enable

If this bit is written to one, and the I flag in SREG is set (interrupts globally enabled), the Timer5 output compare match interrupt is enabled. The corresponding interrupt vector is executed when the T5IFR.T5COF flag is set.

Bit 0: T5OIM – Timer5 Overflow Interrupt Enable

If this bit is written to one, and the I flag in SREG is set (interrupts globally enabled), the Timer5 overflow interrupt is enabled. The corresponding interrupt vector is executed when the T5IFR.T5OFF flag is set.

T5OCRH – Timer5 Output Compare Register High Byte

Bit	7	6	5	4	3	2	1	0	
	T5OCR[15:8]								T5OCRH
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 output compare register contains a 16-bit value that is continuously compared with the counter value (T5CNT). The T5OCRH represents the upper byte of the compare value.

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 high-byte register (TEMP). This temporary register is shared by all the other 16-bit registers.

T5OCRL – Timer5 Output Compare Register Low Byte

Bit	7	6	5	4	3	2	1	0	
	T5OCR[7:0]								T5OCRL
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 output compare register contains a 16-bit value that is continuously compared with the counter value (T5CNT). The T5OCRL represents the lower byte of the compare value.

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 high-byte register (TEMP). This temporary register is shared by all the other 16-bit registers.

T5CNTH – Timer5 Counter Register High Byte

Bit	7	6	5	4	3	2	1	0	
	T5CNT[15:8]								T5CNTH
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 T5CNTH/L registers provide direct access, both for read and for write operations to the 16-bit counter. The T5CNTH register represents the upper byte of the counter value.

To ensure that both the high and low bytes are read and written 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 the other 16-bit registers.

Modifying T5CNTH while the counter is running processes the risk of missing a compare match between T5CNT and the T5OCR register.

Writing to the T5CNTH register blocks (removes) the compare match on the following timer clock.

T5CNTL – Timer5 Counter Register Low Byte

Bit	7	6	5	4	3	2	1	0	
	T5CNT[7:0]								T5CNTL
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 T5CNTH/L registers provide direct access, both for read and for write operations to the 16-bit counter. The T5CNTL register represents the lower byte of the counter value.

To ensure that both the high and low bytes are read and written 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 the other 16-bit registers.

Modifying T5CNTL while the counter is running processes the risk of missing a compare match between T5CNT and the T5OCR register.

Writing to the T5CNTL register blocks (removes) the compare match on the following timer clock.

GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
	TSM	-	-	-	-	-	-	**PSR10**	**GTCCR**
Read/Write	R/W	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: TSM – Timer Synchronization Mode

Writing the TSM bit to “1” activates the timer synchronization mode. In this mode, the value that is written to the PSR10 bit is retained and thus keeps the corresponding prescaler reset signals asserted. This ensures that the corresponding timers are stopped and can be configured to the same value without the risk of one of them advancing during configuration. When the TSM bit is written to zero, the PSR10 bit is cleared by hardware, and the timers start counting simultaneously.

Bit 6..1: Reserved Bits

These bits are reserved and read as zero.

Bit 0: PSR10 – Prescaler Reset

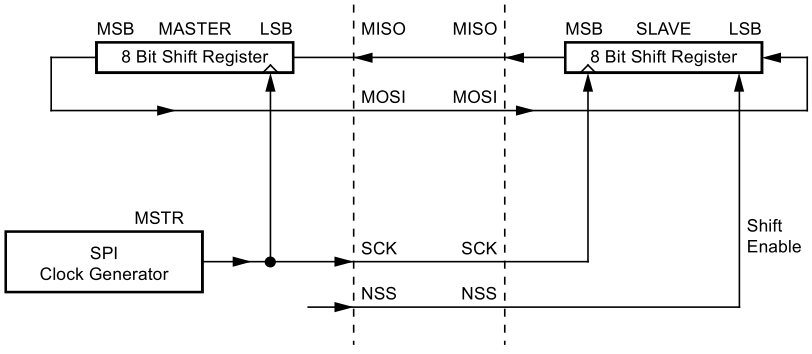
When this bit is “1”, the Timer5 prescaler is reset. This bit is normally cleared immediately by hardware, unless the TSM bit is set.

3.8.8 SPI – Serial Peripheral Interface

The serial peripheral interface (SPI) allows for a synchronous data transfer between the Atmel ATA5785 and peripheral devices or external microcontrollers. The SPI includes the following features:

- Full-duplex, three-wire synchronous data transfer
- Master or slave operation
- LSB first or MSB first data transfer
- Programmable bit rates
- 4 bytes receive buffer, 4 bytes transmit buffer
- EOT interrupt flag
- Configurable fill status interrupt flags
- Wake-up from idle mode
- Double speed (CK/2) master SPI mode

Figure 3-63. SPI Master – Slave Interconnect

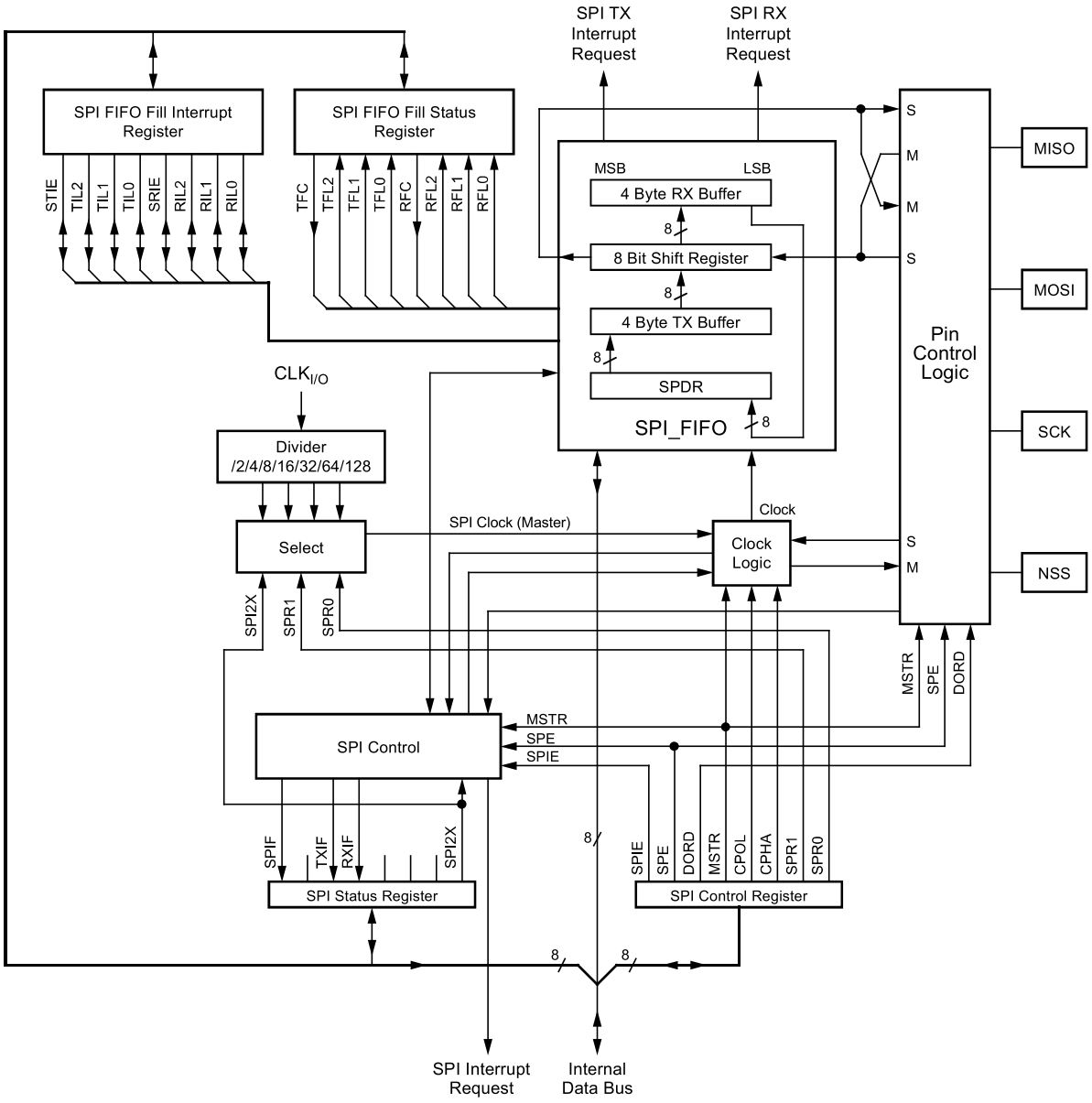


The interconnection between master and slave CPUs with SPI is shown in Figure 3-63. The system consists of two shift registers and a master clock generator. The SPI master initiates the communication cycle when pulling low the low active slave select (NSS) pin of the desired slave. Master and slave prepare the data to be sent in their respective shift registers. Additionally, 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 packet, for example, a telegram or any number of data bytes forming a packet, the master synchronizes the slave by pulling high the NSS line.

When configured as a master, the SPI interface has no automatic control of the NSS 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 8 bits into the slave. After shifting one byte, the next byte is fetched from the transmit buffer. If it is empty, the SPI clock generator stops. The master may continue to shift the next byte by writing it into SPDR or signal the end of packet by pulling high the NSS line. The last incoming bytes are kept in the receive buffer for later use.

When configured as a slave, the SPI interface remains sleeping with MISO tri-stated as long as the NSS pin is driven high. In this state, firmware may fill the transmit buffer through the SPI data register (SPDR), but the data is not be shifted out by incoming clock pulses on the SCK pin until the NSS pin is driven low. As one byte has been completely shifted, the end of transmission flag (SPIF) is set. If the SPI interrupt enable bit, SPIE, in the SPCR register is set, an interrupt is requested. This can be used for byte-wise transmission when the 4-byte receive and transmit buffers are not used.

Figure 3-64. SPI Block Diagram



As shown in Figure 3-64, the system is buffered independently in the transmit direction and the receive direction. Both buffers are accessed by firmware through the SPDR register. The transmit buffer can be completely filled by the firmware. If the buffer is empty when a new byte should be transferred to the shift register, the latest data are sent multiple times. The transmit buffer state can be controlled by the fill status (TFL) and the corresponding fill status interrupt (TXIF).

The receive buffer can be read as long as it contains data. If it is empty the latest byte is repeated. If the receive buffer is full and a new complete byte is received before the buffer is read, the oldest data in the buffer gets lost. The new byte is transferred from the shift register into the receive buffer in any case. The receive buffer state can be controlled by the fill status (RFL) and the corresponding fill status interrupt (RXIF).

In SPI slave mode, the control logic samples the incoming signal of the SCK pin. To ensure correct sampling of the clock signal, the minimum low and high period should be:

Low period: longer than 2 CPU clock cycles.

High period: longer than 2 CPU clock cycles.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK, and NSS pins are overridden according to Table 3-77. For more details on automatic port overrides, refer to Section 3.8.6.3 “Alternate Port Functions” on page 221.

Table 3-77. SPI Pin Overrides

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

3.8.8.1 NSS Pin Functionality

Slave Mode

When the SPI is configured as a slave, the not slave select (NSS) pin is an input. When NSS is held low, the SPI is activated and MISO becomes an output if configured that way by the user in the corresponding DDR register. All other pins are inputs. When NSS is driven high, all pins are inputs except MISO which must be user configured as an output in the corresponding DDR register. SPI is passive then, meaning it does not receive incoming data.

Note: The slave mode is the only mode supported by the Atmel ATA5785 firmware. The SPI logic is reset once the NSS pin is driven high.

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

Master Mode

When the SPI is configured as a master (MSTR in SPCR is set), the user can determine the direction of the NSS pin.

If NSS is configured as an output, the pin is a general output pin which does not affect the SPI system. Typically, the pin drives the NSS pin of the SPI slave.

If NSS is configured as an input, it must be held high to ensure master SPI operation. If the NSS pin is driven low by peripheral circuitry when the SPI is configured as a master with the NSS 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 takes 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 if the SPI interrupt is enabled and the I bit in SREG is set, the interrupt routine is executed.

Thus, when interrupt-driven SPI transmission is used in master mode and NSS might be driven low, the interrupt should always check 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 SPI master mode.

3.8.8.2 Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by the control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 3-65 and Figure 3-66 on page 269. Data bits are shifted out and latched in on opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize. This functionality is shown in detail in Table 3-78.

Table 3-78. CPOL and CPHA Functionality

	Leading Edge	Trailing Edge	SPI Mode
CPOL=0, CPHA=0	Sample (rising)	Setup (falling)	0 (default mode for Atmel ATA5785)
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 3-65. SPI Transfer Format with CPHA = 0

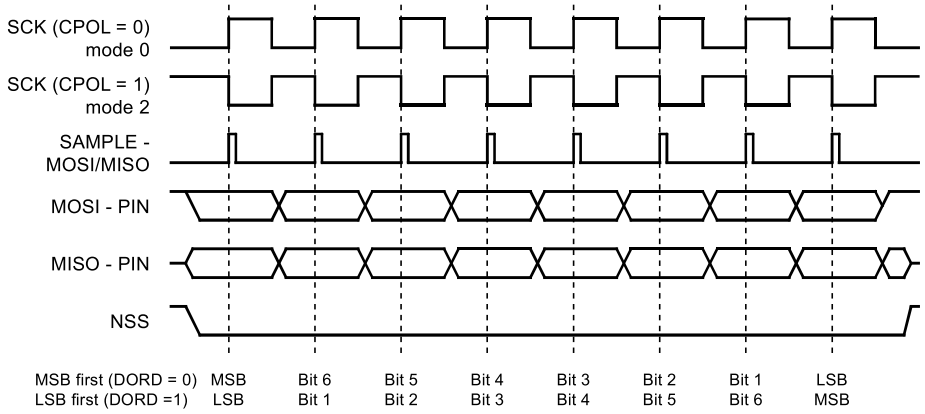
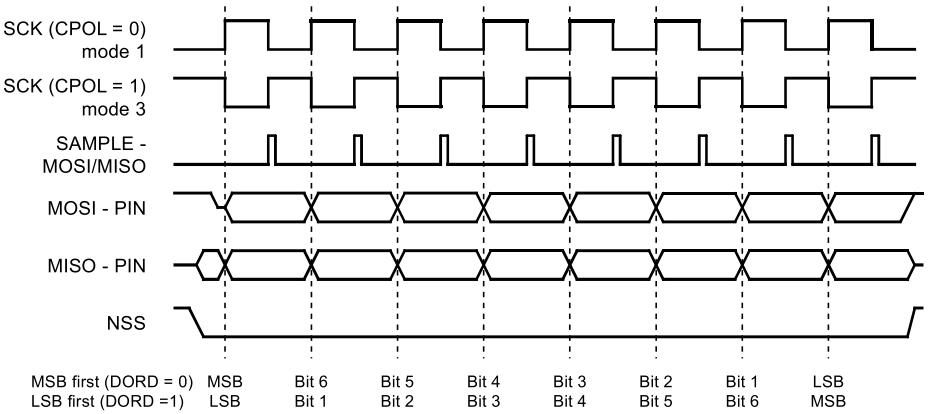


Figure 3-66. SPI Transfer Format with CPHA = 1



Note: In Atmel® ATA5785 the SPI can only work in slave mode with the following settings:CPOL=0, CPHA=0 and DORD=0

3.8.8.3 SPI Register Description

SPCR – SPI Configuration Register

Bit	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR[1:0]		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 the 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

If the DORD bit is written to one, the LSB of the data word is transmitted first.

If the DORD bit is written to zero, the MSB of the data word is transmitted first.

Bit 4: MSTR – Master/Slave Select

This bit selects master SPI mode when written to one and slave SPI mode when written to logic zero. If NSS 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 SPI master mode.

Bit 3: CPOL – Clock Polarity

If this bit is written to one, SCK is high when idle. If CPOL is written to zero, SCK is low when idle. Refer to Figure 3-65 on page 269 and Figure 3-66 on page 269 for an example. The CPOL functionality is summarized in Table 3-79:

Table 3-79. CPOL Functionality

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

Bit 2: CPHA – Clock Phase

The setting of the clock phase (CPHA) determines if data is sampled on the leading (first) or trailing (last) edge of SCK. Refer to Figure 3-65 on page 269 and Figure 3-66 on page 269 for an example. The CPHA functionality is summarized in Table 3-80:

Table 3-80. CPHA Functionality

CPHA	Leading Edge	Trailing Edge
0	Sample	Setup
1	Setup	Sample

Bits 1..0: SPR[1:0] – SPI Clock Rate Select

These two bits control the SCK rate of the device configured as a master. SPR[1:0] has no effect on the slave. The relationship between SCK and the $CLK_{I/O}$ is shown in Table 3-81.

Table 3-81. Relationship Between SCK and Oscillator Frequency

SPSR.SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$CLK_{I/O}/4$
0	0	1	$CLK_{I/O}/16$
0	1	0	$CLK_{I/O}/64$
0	1	1	$CLK_{I/O}/128$
1	0	0	$CLK_{I/O}/2$
1	0	1	$CLK_{I/O}/8$
1	1	0	$CLK_{I/O}/32$
1	1	1	$CLK_{I/O}/64$

SPSR – SPI Status Register

Bit	7	6	5	4	3	2	1	0	
	SPIF	–	TXIF	RXIF	–	–	–	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. If NSS is an input and is driven low when the SPI is in master mode, this also sets the SPIF flag. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit can be cleared by first reading the SPI status register with SPIF set, then accessing the SPI data register (SPDR).

Bit 6: Reserved Bit

This bit is reserved and reads as zero.

Bit 5: TXIF – Transmit Buffer Interrupt Flag

The TXIF bit is set if the SPI transmit buffer has reached the defined fill level as in TIL[2:0] of SFIR. An interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. The TXIF bit is cleared if new data is written to the SPDR and the fill level of the buffer has exceeded the defined interrupt level.

Bit 4: RXIF – Receive Buffer Interrupt Flag

The RXIF bit is set if the SPI receive buffer has reached the defined fill level as in RIL[2:0] of SFIR. An interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. The RXIF bit is cleared if data has been read from SPDR and the fill level of the buffer has fallen below the defined interrupt level.

Bits 3..1: Reserved Bits

These bits are reserved and read as zero.

Bit 0: SPI2X – Double SPI Speed

If this bit is written to logic one the SPI speed (SCK frequency) will be doubled when the SPI is in master mode. This means that the minimum SCK period will be two CPU clock periods. When the SPI is configured as slave, the SPI is only guaranteed to work at $CLK_{I/O}/4$ or lower.

SFFR – SPI FIFO Fill Status Register

Bit	7	6	5	4	3	2	1	0	
	TFC	TFL[2:0]			RFC	RFL[2:0]			SFFR
Read/Write	W	R	R	R	W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: TFC – Transmit Buffer Clear

Writing a “1” to this bit clears the transmit buffer and resets the fill level counter to “0”.

Bits 6..4: TFL[2:0] – Transmit Buffer Fill Level

The SPI transmit buffer fill level is a read-only register, which holds the current fill level of the SPI transmit buffer.

Bit 3: RFC – Receive Buffer Clear

Writing a “1” to this bit clears the receive buffer and resets the fill level counter to “0”.

Bits 2..0: RFL[2:0] – Receive Buffer Fill Level

The SPI receive buffer fill level is a read only register, which holds the current fill level of the SPI receive buffer.

SFIR – SPI FIFO Interrupt Register

Bit	7	6	5	4	3	2	1	0	
	STIE	TIL[2:0]			SRIE	RIL[2:0]			SFIR
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: STIE – SPI Transmit Buffer Interrupt Enable

This bit causes the SPI transmit buffer interrupt to be executed if the TXIF bit in the SPSR register is set and if the global interrupt enable bit in SREG is set.

Bits 6..4: TIL[2:0] – Transmit Buffer Interrupt Level

The SPI transmit buffer interrupt level is a read/write register, which holds the fill level that triggers an SPI interrupt. If it is set to 0, an interrupt is triggered from the transmit FIFO when the last byte has been transferred from the transmit buffer to the serial shift register. If it is set to 1, it issues an interrupt when only one byte is left in the buffer.

Bit 3: SRIE – SPI Receive Buffer Interrupt Enable

This bit causes the SPI receive buffer interrupt to be executed if the RXIF bit in the SPSR register is set and if the global interrupt enable bit in SREG is set.

Bits 2..0: RIL[2:0] – Receive Buffer Interrupt Level

The SPI receive buffer interrupt level is a read/write register, which holds the fill level that triggers an SPI interrupt. If it is set to 0, no interrupt is triggered from the receive FIFO. Set to 1 it issues an interrupt, when 1 byte has been captured and written to the buffer.

SPDR - SPI Data Register

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

Bit 7..0: SPDR[7:0] – SPI Data Register

The SPI data register is a read/write register used for data transfer between the register file and the SPI transmit and receive buffer. Writing to the register puts data into the transmit buffer and initiates data transmission. Reading the register causes the receive buffer to be read. The fill levels in SFFR are automatically incremented or decremented, respectively.

3.8.9 CRC

An 8-bit CRC generator with byte access is provided as standard AVR® peripheral module. The CRC module is used to verify the SRAM content in the SPI command “Start SRAM CRC calculation”. This CRC generator is not intended to be used to check the received telegrams as there are more flexible CRC modules available in the Rx data path of the Atmel ATA5785.

Key properties:

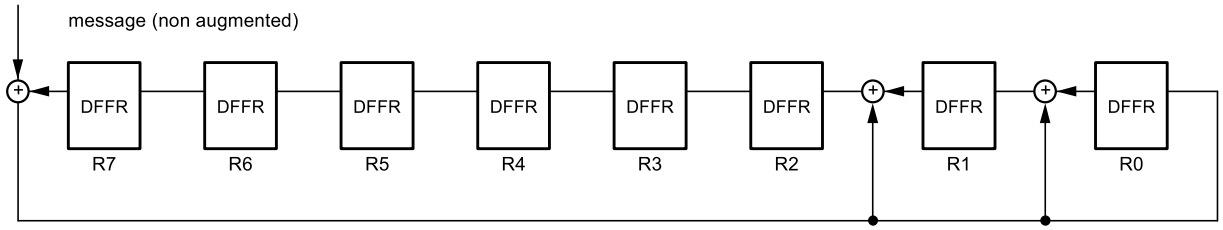
- Very fast parallel CRC computation
- Accessible via AVR bus
- Fixed polynomial

Generally, CRC checkers detect the following types of errors:

1. Single-bit errors
2. Two-bit errors
3. Three-bit and other odd-number-bit errors
4. Burst errors that are less than or equal to the CRC length
5. Most burst errors that are greater than the CRC length
6. The types of errors that CRC checker detects depending on the generator polynomial.

The implemented CRC checker is using the following polynomial: CRC(x) = x^8 + x^2 + x^1 + 1.

Figure 3-67. CRC Equivalent Block Diagram



The parallel implementation of the polynomial is equivalent to the linear feedback shift register (LFSR) in Figure 3-67. The initial value of the 8-bit CRC data register is 0x00. With some standards, the data is expected to enter the processor LSBF (least significant bit first). In this case the incoming data and the resulting CRC must be bit reflected. A binary number is reflected by swapping all of its bits around the central point. For example, 1101 is the reflection of 1011. The bits REFLI and REFLO of the CRC control register are required to configure the bit order of incoming and outgoing data. If REFLI is FALSE (initial value), input bytes are processed with bit 7 being treated as the most significant bit (MSB) and bit 0 being treated as the least significant bit (LSB). If REFLI is TRUE, each byte is reflected before being processed. If REFLO is set to FALSE (initial value), the final value in the CRC register is fed into the output stage directly, otherwise, if this parameter is TRUE, the final CRC register value is reflected first.

CRC computation:

1. Initialize the CRC data register
2. Write message byte to CRC data register
3. Continue with step 2 until there are no message bytes left
4. Read CRC data register
5. Continue with step 1

3.8.9.1 CRC Register Description

CRCCR – CRC Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	REFLO	REFLI	CRCRS	CRCCR
Read/Write	R	R	R	R	R	R/W	R/W	W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..3: Reserved Bits

These bits are reserved and read as zero.

Bit 2: REFLO – Reflect Data Output Byte

If this parameter is 0, the final value in the CRC register is fed into the output stage directly, otherwise, if this parameter is set to 1, the final register value is reflected first.

Bit 1: REFLI – Reflect Data Input Byte

If this parameter is 0, input bytes are processed with bit 7 being treated as the MSB and bit 0 being treated as the LSB. If it is set to 1, each data input byte is reflected before being processed.

Bit 0: CRCRS – CRC Data Register Reset

This bit is used to reset the CRC data input register and CRC data output register (CRCDIR, CRCDOR). Once written to one, hardware clears this bit after one clock cycle.

CRCDIR – CRC Data Input Register

Bit	7	6	5	4	3	2	1	0	
	CRCDIR[7:0]								CRCDIR
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: CRCDIR[7:0] – CRC Data Input Register

Incoming bytes are fed into this register for processing by the CRC block.

CRCDOR – CRC Data Output Register

Bit	7	6	5	4	3	2	1	0	
	CRCDOR[7:0]								CRCDOR
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..0: CRCDOR[7:0] – CRC Data Output Register

The CRC computation result is available in this register one clock cycle after the last byte was fed into the CRCDIR register.

3.9 Power Management

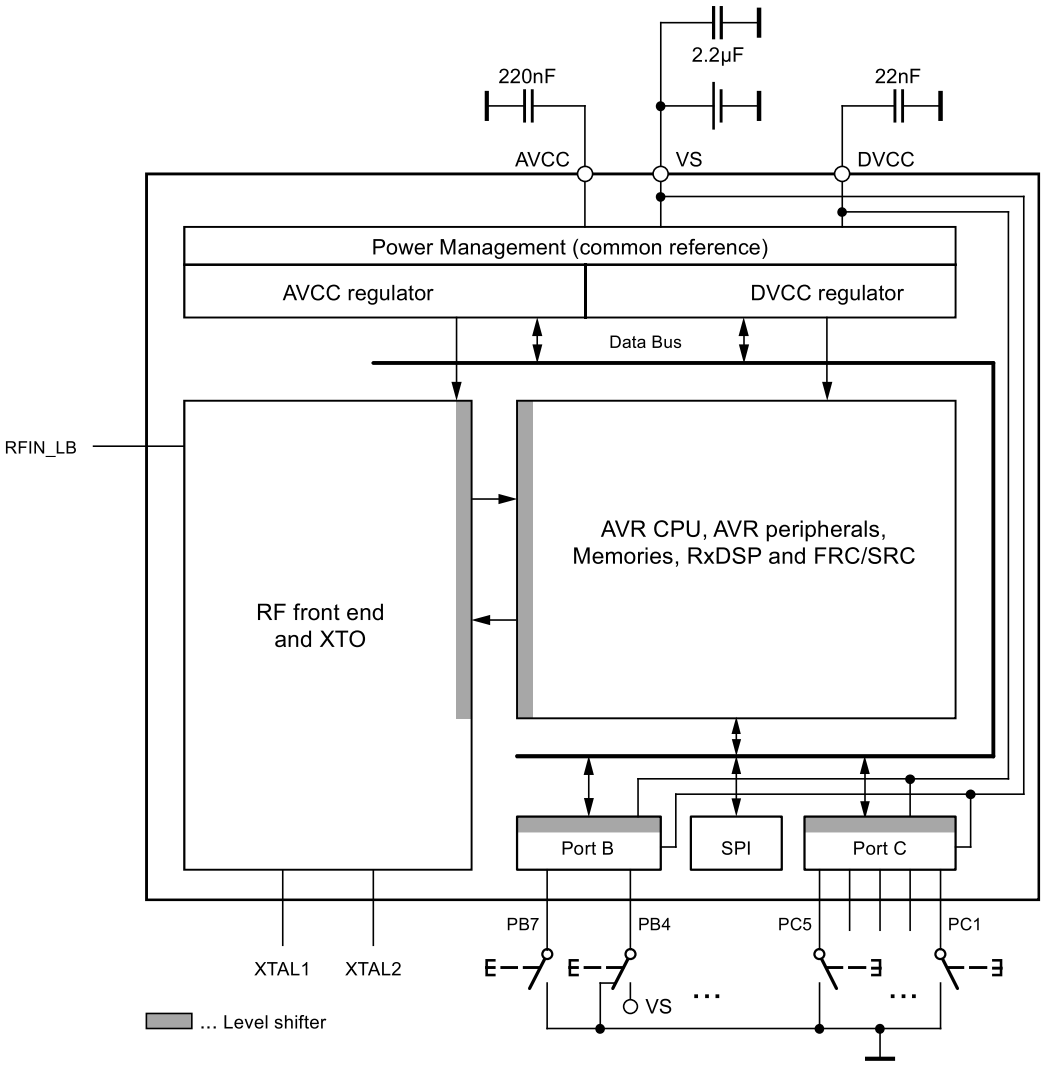
3.9.1 Overview

The IC has three power domains:

1. VS – Unregulated battery voltage input
2. DVCC – Internally regulated digital supply voltage. Typical value is 1.35V.
3. AVCC – Internally regulated RF front end and XTO supply. Typical value is 1.85V.

The Atmel® ATA5785 can be operated from V_S= 1.9V to 5.5V (3V and 5V applications).

Figure 3-68. Power Supply Management



3.9.2 Power Supplies

3.9.2.1 VS Battery Voltage Input

The VS pin should be blocked by an external 2.2µF capacitor in order to prevent the IC from strong supply voltage drops during the start-up procedure.

3.9.2.2 DVCC Supply Voltage Regulator

The DVCC supply voltage is enabled by a power-on. This happens when the PWRON pin (port PB4, as depicted in Figure 3-68 on page 275) is set to high or an NPWRONx pin (Ports PC1 to PC5 and PB7) is set to low for a time period longer than $T_{power\_on\_REQ}$. This time is needed to allow detection of the wake-up pin. The customer application software or an external microcontroller can verify the validity of a power-up by reading the corresponding event flags provided by the Atmel® ATA5785 firmware. An external 22nF blocking capacitor guarantees the stability of the internal DVCC supply voltage regulator circuit. Other values than 22nF for this capacitor should not be used.

The DVCC supply voltage can be turned off by setting the SUPCR.DVDIS register bit if no NPWRONx or PWRON pin is active or by using the OFF command.

3.9.2.3 AVCC Supply Voltage Regulator

The AVCC supply voltage regulator provides the supply voltage for the RF front end and XTO. It delivers high output current while maintaining tight control over voltage. The 220nF external blocking capacitor guarantees the stability of the internal AVCC voltage regulator circuit. Other values than 220nF for this capacitor should not be used.

The AVCC supply voltage can be enabled and disabled by using the SUPCR.AVEN bit. This is done automatically by firmware when starting the XTO or the RXMode.

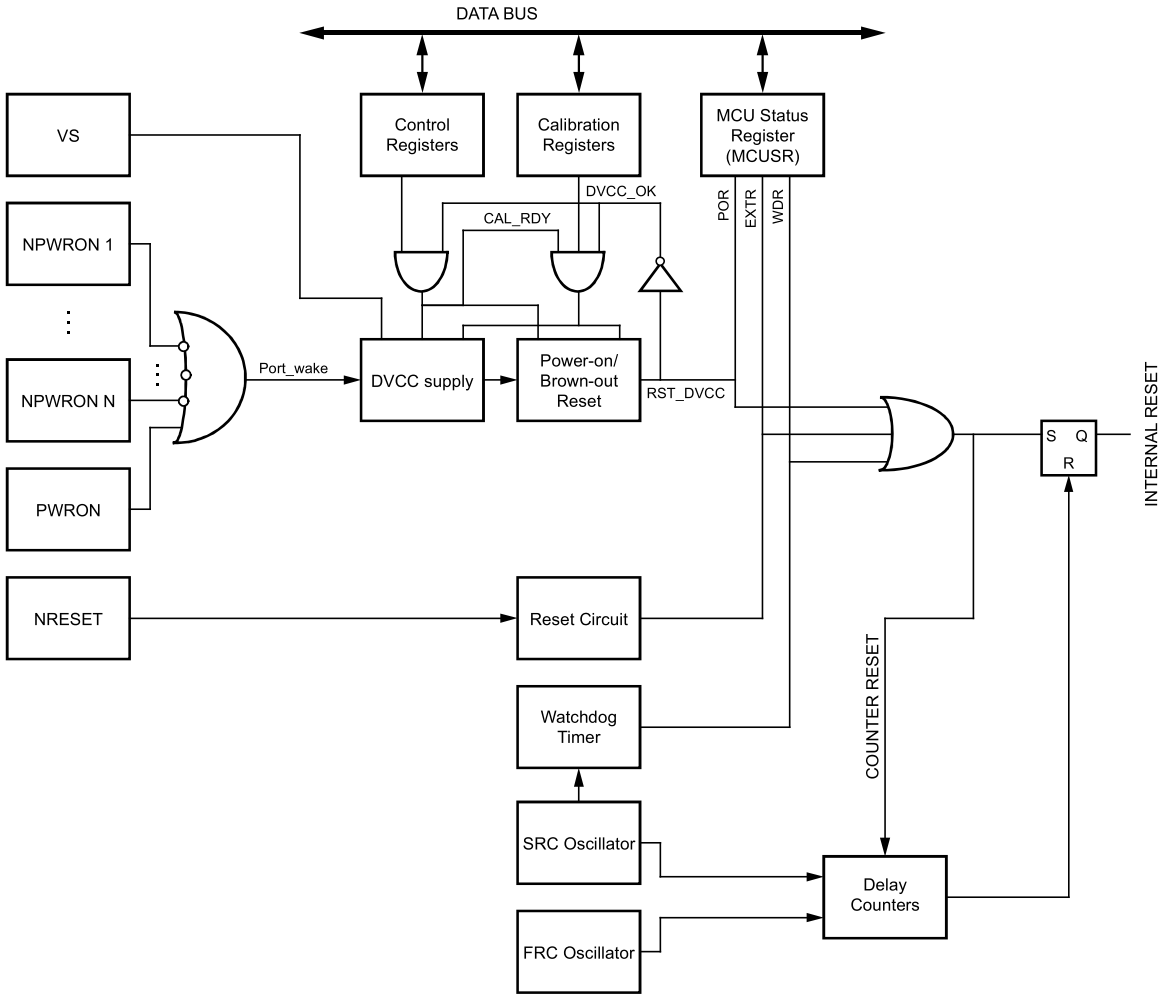
3.9.3 AVR Reset

During AVR® reset all I/O registers are set to their initial (hardware-defined) values and the AVR program starts execution from the reset vector. The circuit diagram in Figure 3-69 on page 277 shows the reset logic of the Atmel ATA5785.

Three sources can generate an AVR reset:

- DVCC-related power on reset (POR):
  The AVR reset is generated when the supply voltage DVCC is below the brown-out reset threshold (V_{BOT-}). The brown-out detector is always enabled.
- External reset (EXTR):
  The AVR reset is generated when a low level is detected on the NRESET pin for longer than 10 µs.
- Watchdog reset (WDR):
  The AVR reset is generated when the watchdog is enabled and its timer period expires.

Figure 3-69. Reset Logic of the Atmel ATA5785



3.9.3.1 Power ON and Brown-Out Reset (POR)

The DVCC supply voltage regulator output is monitored permanently by the brown-out detector (BOD). This monitor circuit is also used during the power-up sequence. After a PWRON or NPWRONx edge event, the DVCC voltage ramps up. When reaching the V_{BOT+} voltage (DVCC reset level, typically 1.27V), the internal reset switches to “1” and an internal timer is started. The time depends on the reset source as described in Table 3-82 on page 281 in more detail. After the time period of t_{TOUT}, the negative edge of the internal reset starts the AVR® operation. The complete power-up timing is shown in Figure 3-76 on page 281.

Figure 3-70. AVR Power-Up with PWRON and NRESET

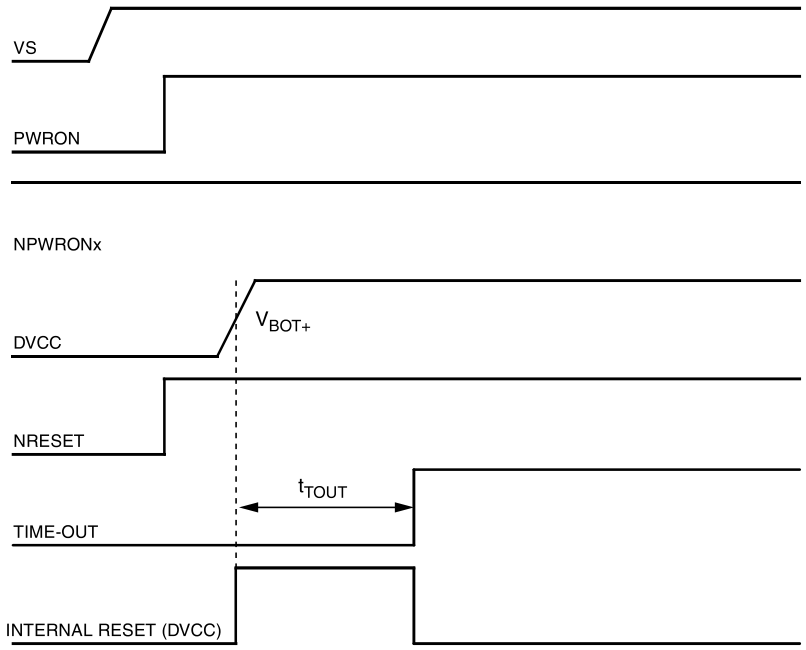
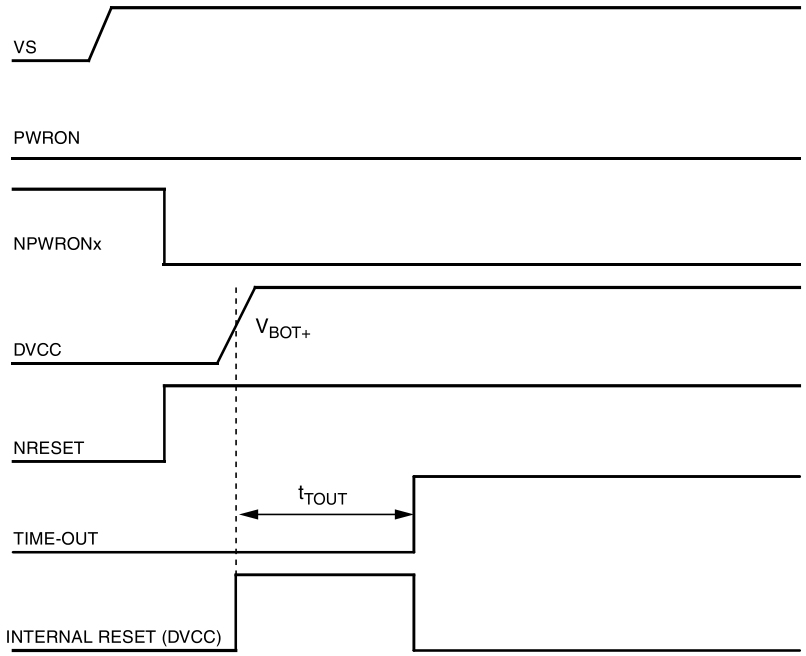
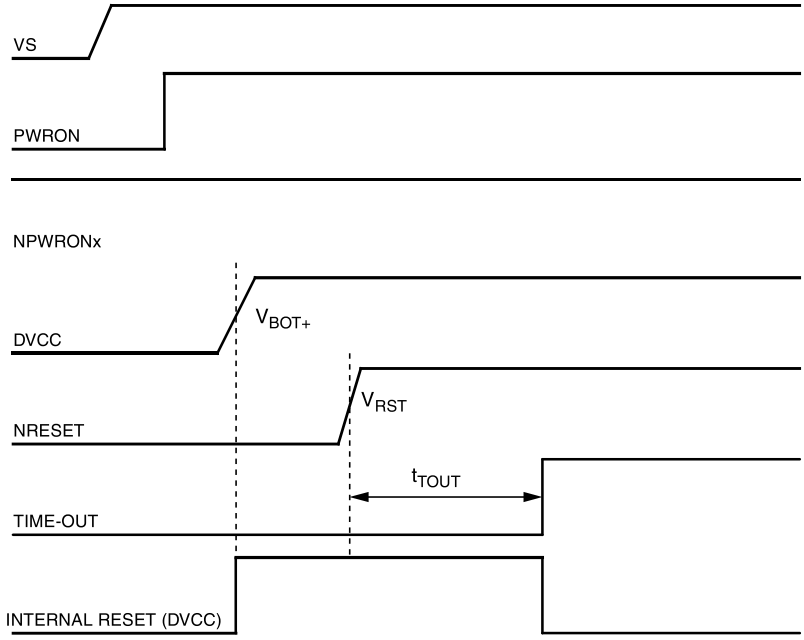


Figure 3-71. AVR Power-Up with NPWRON and NRESET



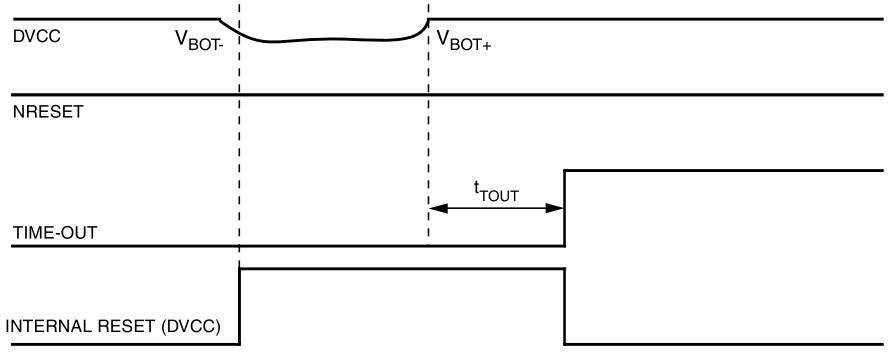
The duration of the internal reset delay t_{TOUT} can be extended by an external signal applied to the NRESET pin. The DVCC voltage must be already stable. If the signal at NRESET reaches the voltage level V_{RST} (approximately VS/2), the reset timer t_{TOUT} is re-triggered to cause the internal reset.

Figure 3-72. AVR Power-Up with Externally Extended Reset



A reset is generated when DVCC voltage drops below the internal threshold level (V_{BOT-}, typically 1.25V). If the DVCC voltage increases again (V_{BOT+}, typically 1.27V, 20mV hysteresis) after the time period of t_{TOUT}, the AVR® system resumes operation. The timing is described in Section 3.9.3.4 “AVR Reset Timing” on page 281. A typical DVCC voltage drop scenario is shown in Figure 3-73 on page 279.

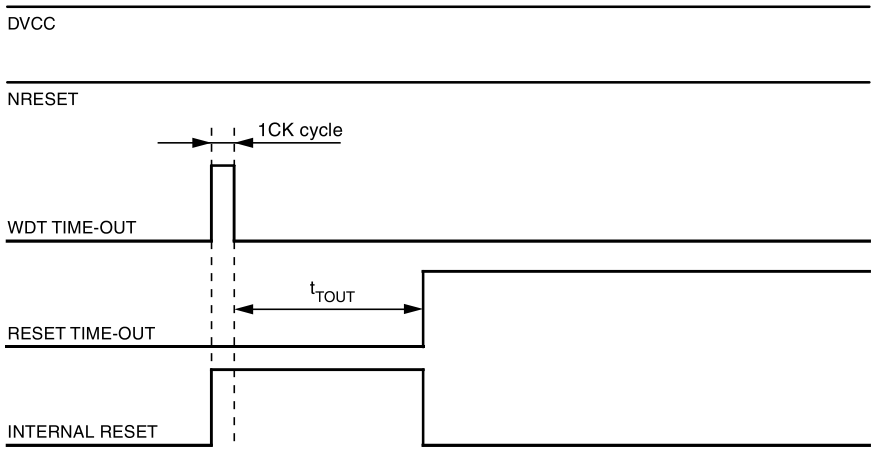
Figure 3-73. DVCC Drop during Operation



3.9.3.2 Watchdog Reset (WDTRESET)

When the watchdog time-out occurs, a short reset pulse of one clock cycle duration is generated. The delay timer starts counting the time-out period t_{TOUT} on the falling edge of this pulse as described in Section 3.9.3.4 “AVR Reset Timing” on page 281. For more information about operating the watchdog timer, see Section 3.8.7.1 “Timer0 – Watchdog/Interval Timer” on page 233.

Figure 3-74. Watchdog Reset during Operation

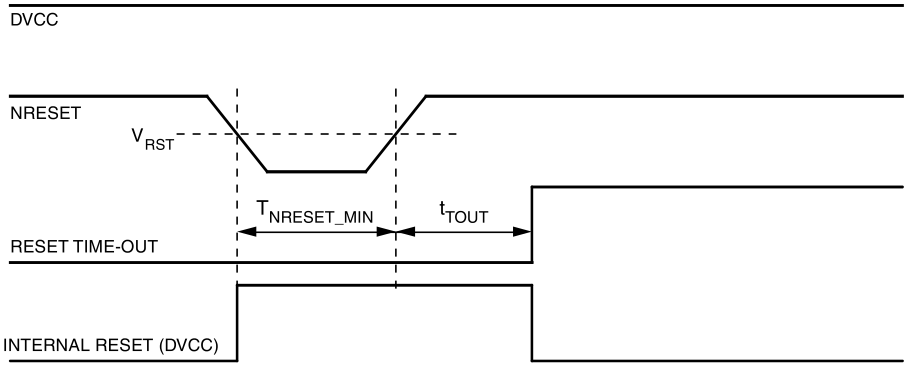


3.9.3.3 External Reset (NRESET)

An external reset is generated by a low level on the NRESET pin. The external reset is always activated.

Reset pulses longer than the minimum pulse width (T_{NRESET_MIN} of 10µs) trigger a reset even if the internal clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal on the NRESET pin reaches the reset threshold voltage V_{RST} (approximately VS/2) on its positive edge, the delay counter starts the AVR after the time-out period t_{TOUT} has expired.

Figure 3-75. NRESET during Operation



3.9.3.4 AVR Reset Timing

When an AVR® reset source becomes active, the AVR registers are reset. Operation is started again at the next rising edge of the selected clock source. After all reset sources become inactive, a delay counter is invoked which extends the internal reset by t_{TOUT}. This method allows all supplies to reach a stable level before normal operation can be started.

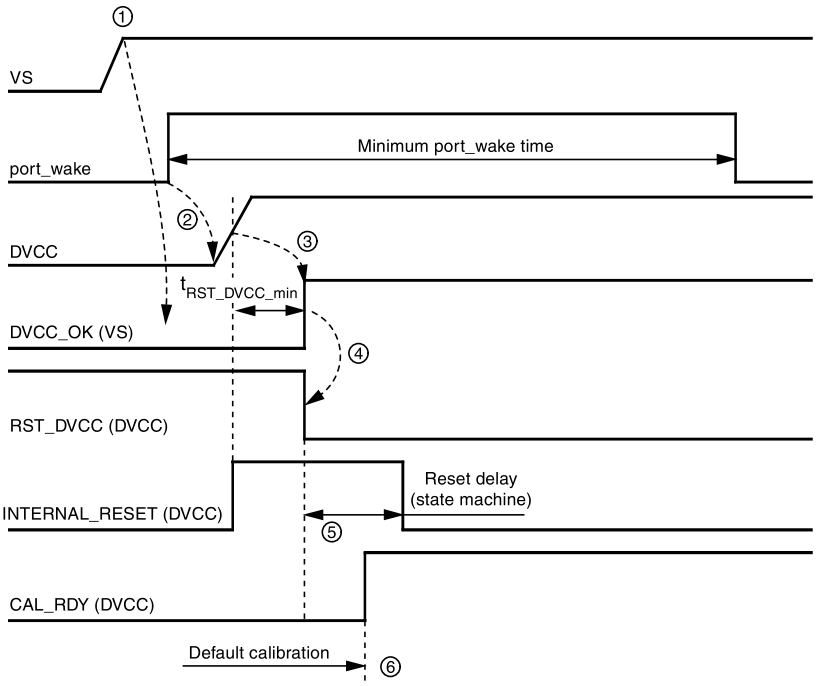
The table below calculates the time t_{TOUT} until the internal reset is removed.

Table 3-82. Delay Time t_{TOUT} until Internal Reset is Removed

Reset Source	t_{TOUT} in Numbers of Clocks
POR	$142 \times CLK_{FRC}$
NRESET	$140 \times CLK_{FRC}$
WDTRESET	$512 \times CLK_{FRC} + 140 \times CLK_{FRC}$

Figure 3-76 illustrates a typical power-up sequence. The power supply domain of the circuit delivering the signal is included in brackets.

Figure 3-76. Power-Up Sequence



1. The main supply is applied to the VS pin. The internal supply generators remain disabled. The DVCC_OK = LOW indicates that the DVCC is not yet active and therefore not high enough for operation.
2. The PWRON pin is set to high or an NPWRON pin is pulled to low (button pressed). This triggers the port_wake signal, enabling the DVCC regulator.
3. When the DVCC voltage rises above the reset threshold V_{BOT+}, the DVCC_OK signal switches to high, indicating a valid operation voltage. An analog delay $t_{RST\_DVCC\_min}$ is implemented to guarantee enough settling time for all signals.
4. The DVCC reset (RST_DVCC) is set as early as possible with the rising DVCC voltage and cleared by the high level of DVCC_OK.
5. The RST_DVCC signal triggers the signal INTERNAL_RESET. The internal AVR® reset is extended by a reset state machine that clears the registers and loads the calibration settings of all regulators, reset monitors and RC oscillators. The CAL_RDY signal is set to high after the calibration data is available to the power management circuit.

6. The internal AVR reset is released after reset and calibration data of voltage regulators, reset levels and RC oscillators are available. The AVR firmware starts operation. The PORT_RDY signal activates the port configuration.

3.9.4 AVCC Voltage Supervision

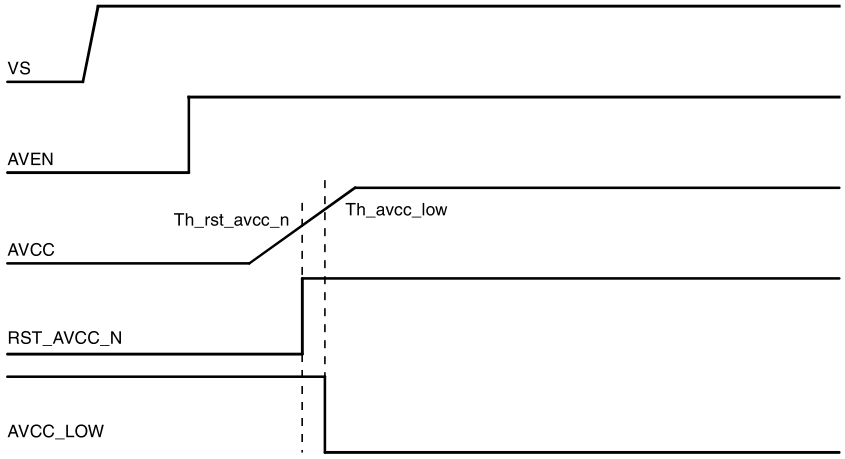
The AVCC voltage is enabled by the AVEN bit in the SUPCR register. Two thresholds are defined for the AVCC voltage supervision:

- Th_rst_avcc_n – Indicates that the voltage is below the safe operating voltage range of the digital circuits.
- Th_avcc_low – Indicates that the voltage is below the safe operating voltage of the analog RF front end and XTO circuits.

The reset circuits are used to allow proper operation of the analog and digital portion of the RF front end and XTO. Both thresholds have a hysteresis of at least 10mV.

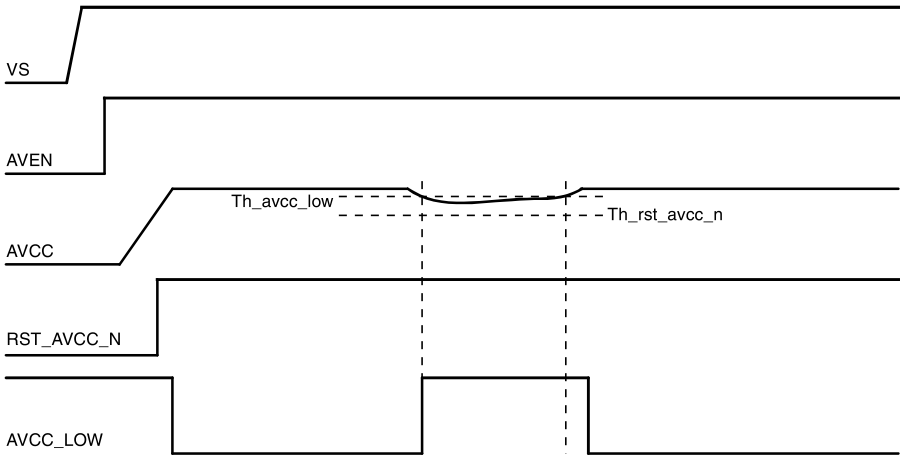
A typical AVCC power-up sequence is shown in Figure 3-77. The battery voltage VS is applied and the AVR® is ready for operation. The SUPCR.AVEN signal is set by firmware and the AVCC voltage starts ramping up. When the reset threshold Th_rst_avcc_n is reached, the AVCC reset is set, and the RF front end and XTO configuration via the internal front-end control can be started. The analog circuits are ready to be enabled when the Th_avcc_low threshold is reached.

Figure 3-77. AVCC Power-Up Sequence



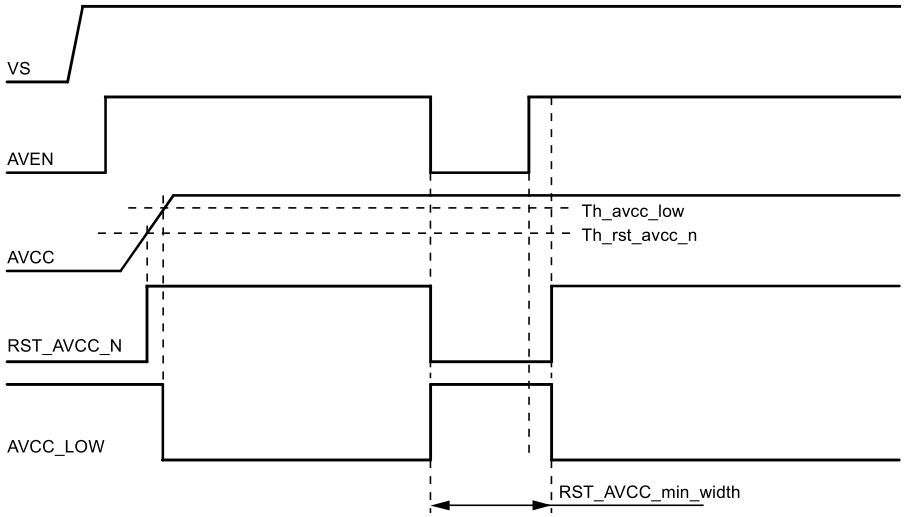
The firmware activates all front-end blocks in a way that no undesired AVCC voltage drop occurs. In case of a weak battery, the AVCC voltage might drop below the Th_avcc_low threshold during operation (see Figure 3-78 on page 282). The AVCC low interrupt flag AVCCLF in the SUPFR byte is set by the AVCC_LOW signal.

Figure 3-78. AVCC Drop during Operation and Violating Low Threshold



If the SUPCR.AVEN bit is cleared, a RST_AVCC_N and AVCC_LOW indication occurs even if the AVCC voltage does not drop below the reset threshold as depicted at the first falling edge of AVEN in Figure 3-79. This ensures a reliable reset each time the AVCC is turned off. The reset circuit reliably maintains a minimum pulse width RST_AVCC_min_width at RST_AVCC_N and AVCC_LOW. The AVCC reset interrupt flag (AVCCRF) in the SUPFR is set by the RST_AVCC_N signal.

Figure 3-79. RST_AVCC and AVCC_LOW Controlled by AVEN



3.9.5 Power Management Register Description

SUPCR – Supply Control Register

Bit	7	6	5	4	3	2	1	0	
	-	**AVDIC**	**AVEN**	**DVDIS**	**-**	**-**	**AVCCLM**	**AVCCRM**	**SUPCR**
Read/Write	R	R/W	R/W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: Reserved Bit

This bit is reserved for future use and read as zero.

Bit 6: AVDIC – AVCC Double Internal Current

The internal bias current of the AVCC regulator is increased if this bit is set to “1”. Higher bias current provides more AVCC stability on fast load transitions. This bit must always be “0” in the ATA5785.

Bit 5: AVEN – AVCC Enable

The RF front end and XTO power supply providing AVCC is enabled by setting this bit. The bit is set automatically if IDLEMode(XTO), or RXMode, are started. Clearing this bit disables the AVCC power supply and enables a reset of the RF front end and the XTO.

Bit 4: DVDIS – DVCC Disable

The digital power supply is disabled when setting this bit under the condition that no NPWRONx=0 or PWRON=1 pin is currently set. In addition, the AVR® switches to the reset state and the Atmel ATA5785 switches to OFFMode. The digital power supply is enabled again using the PWRON and NPWRONx pins which are controlled externally.

Bits 3..2: Reserved Bits

These bits are reserved for future use and read as zero.

Bit 1: AVCCLM – AVCC Low Interrupt Mask Bit

When the AVCCLM bit and the I-bit in the AVR status register are set to “1”, the AVCC low interrupt is enabled. The corresponding interrupt is executed if the AVCC voltage falls below the AVCC low threshold, which is typically 100mV below the nominal AVCC voltage.

Bit 0: AVCCRM – AVCC Reset Interrupt Mask Bit

This bit is reserved for future use and read as zero.

SUPFR – Supply Interrupt Flag Register

The SUPFR provides the status of the AVCC supply, but does not trigger a reset of the AVR directly.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	AVCCLF	AVCCRF	SUPFR
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: Reserved Bits

These bits are reserved for future use and read as zero.

Bit 1: AVCCLF – AVCC Low Interrupt Flag

This flag is set if the AVCC level falls below the lowest limit of the operating range of the analog circuits in the RF front end and XTO. The circuits are disabled. The settling procedure of the circuits must thus be performed again before valid data can be processed. If the I-bit in SREG is set to “1”, the AVCCLM bit in the SUPCR register is set to “1”, sysEventConf.AVCCLOWM is set to “1”, and the AVR then 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 logical “1” to its bit location.

Bit 0: AVCCRF – AVCC Reset Interrupt Flag

This bit is set if an AVCC reset occurs. When AVCC is turned off or falls below the reset threshold during operation, the AVCCRF flag is set. In this case the operation of the RF front end and XTO is stopped by the hardware reset and the digital settings of the RF front end and XTO are lost.

MCUSR – MCU Status Register

The MCU status register provides the information which reset source caused an AVR reset.

PORF, EXTRF, and WDRF trigger the non-maskable reset interrupt.

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

Bits 7 to 4: Reserved Bits

These bits are reserved for future use and read as zero.

Bit 3: WDRF – Watchdog Reset Flag

This bit is set if a watchdog reset occurs. The bit is cleared by a power-on reset or by writing a logic “0” to the flag.

Bit 2: Reserved Bit

This bit is reserved for future use and reads as zero.

Bit 1: EXTRF – External Reset Flag

This bit is set if an external reset occurs. The bit is cleared by a power-on reset or by writing a logic “0” to the flag.

Bit 0: PORF – Power-On Reset Flag

This bit is set if a power-on reset or a brown-out reset occurs due to a low DVCC voltage. The brown-out circuit is always active when the DVCC voltage is enabled. The bit is cleared only by writing a logic “0” to the flag.

In order to use the reset flags for identifying a reset condition, the user must read and then reset the MCUSR in a customer-defined program as soon as possible. If the register is cleared before another reset occurs, the source of the reset is found by examining the reset flags.

SUPCA2 – Supply Calibration Register 2

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	BGCAL[3:0]				SUPCA2
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: Reserved Bits

These bits are reserved for future use and read as zero.

Bits 3..0: BGCAL

Factory calibration settings. The customer should not modify these register bits.

SUPCA3 – Supply Calibration Register 3

Bit	7	6	5	4	3	2	1	0	
	-	DCAL[6:4]			ACAL[7:4]				SUPCA3
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	

Bits 7: Reserved Bit

This bit is reserved for future use and reads as zero.

Bits 6..0: DCAL and ACAL

Factory calibration settings. The customer should not modify these register bits.

SUPCA4 – Supply Calibration Register 4

Bit	7	6	5	4	3	2	1	0	
	DCAL[3:0]				ACAL[3:0]				SUPCA4
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: DCAL and ACAL

Factory calibration settings. The customer should not modify these register bits.

CALRDY – Calibration Ready Signature

Bit	7	6	5	4	3	2	1	0	
	CALRDY[7:0]								CALRDY
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: CALRDY

Factory calibration settings. The customer should not modify these register bits.

4. Application

4.1 Rx-Tx Setup

This section describes application relevant information for the use of the RF circuits within the Atmel® ATA5785. In a transceiver system the frequency planning and the frequency accuracy are of importance. This is discussed in Section 4.1.1 “Frequency Accuracy in System Design” on page 286. The frequency can be chosen in a wide range. Because of the chip architecture not all variations are reasonable. The limitations are shown in Section 4.1.2 “Fractional-N Divider and SDM” on page 291. The RF circuits of the IC must be matched to the antenna. High-Q surface-mounted devices (SMD) should be used on the board to match the LNA inputs. The matching components for the Atmel application board (described in Section 4.2 “Application Board Design” on page 295) are summarized for a wide variety of frequencies. The data for these matching devices are found in Section 4.1.3 “Impedance Matching of RFIN” on page 292.

4.1.1 Frequency Accuracy in System Design

4.1.1.1 System Design with Compensation of Initial XTAL and XTO Tolerances

The initial frequency tolerance of the system is given by the crystal tolerance, by the integrated capacitors on pins XTAL1 and XTAL2 and by the initial transconductance tolerance of the XTO. The tolerance can be lowered to ±1ppm by measuring the CLK_OUT frequency at the end of the module production and by correcting the frequency programming of the fractional-N PLL. This initial tolerance depends on the measuring accuracy and the resolution of the fractional-N PLL.

The XTO has a remaining influence of less than ±4ppm (see parameter no. 13.50 in Section 5. “Electrical Characteristics” on page 300).

The resulting tolerance is given by

- ±5ppm residual XTO tolerance from above
- Crystal frequency temperature tolerance, for example, ±30ppm in the temperature range of –40°C and +85°C(1)
- Crystal aging of ±10ppm.

Note: 1. The influence of the temperature tolerance can be reduced by using the RF calibration feature of the Atmel ATA5785.

In this example, ±45ppm of total tolerance must be compensated. In a 433.92MHz system this results in a ±19.52kHz total frequency tolerance on one side of the transceiver system. If both sides of the system have the same tolerances, a maximum frequency offset between transmitter and receiver of ±2 × 19.52kHz = ±39.05kHz has to be taken into account.

Using 10kBit/s FSK with ±10kHz frequency deviation results in a required IF bandwidth of 106.38kHz. In this example, IF bandwidth of either 110kHz or 123kHz can be chosen. Section 4.1.1.3 “Estimation of Maximum Frequency Offset between Receiver and Transmitter” on page 287 describes the impact of the IF bandwidth on system performance. If choosing 110kHz IF bandwidth, 1.5dB less sensitivity has to be considered for the link budget under worst case frequency tolerance conditions. On the other hand, no loss in sensitivity due to frequency offsets has to be considered if 123kHz IF bandwidth is selected.

4.1.1.2 System Design without Compensation of Initial XTAL and XTO Tolerances

If tuning at the end of module production is not possible, additional tolerances have to be considered and result in broader system bandwidth. The resulting tolerance is given by

- The internal load capacitance variation and the XTO variation amounts for a motional capacitance of 4fF in the crystal to ±10ppm (see parameter no. 13.40 in Section 5. “Electrical Characteristics” on page 300) + ±4ppm (see parameter no. 13.50 in Section 5. “Electrical Characteristics” on page 300) = ±14ppm
- The initial frequency tolerance of the crystal of, for example, ±40 ppm
- The temperature tolerance of the crystal, for example, ±30ppm in the temperature range of –40°C to +85°C(1)
- Crystal aging, for example, ±10ppm

Note: 1. The influence of the temperature tolerance can be reduced by using the RF calibration feature of the Atmel ATA5785.

The total frequency tolerance is ±94ppm. For a 433.92MHz system the total frequency tolerance on one side of the transceiver system is ±40.78kHz. If both sides of the system have the same tolerances, a maximum frequency offset between transmitter and receiver of 2 × ±40.78kHz = ±81.57kHz has to be taken into account.

Finally, using 10kBit/s FSK with ±10kHz frequency deviation results in required IF bandwidth of 191.4kHz. For the system design IF bandwidth of 219kHz IF bandwidth can be selected. No loss in sensitivity due to frequency offsets has to be considered due to the 27.6kHz bandwidth margin (see Section 4.1.1.3 “Estimation of Maximum Frequency Offset between Receiver and Transmitter” on page 287).

4.1.1.3 Estimation of Maximum Frequency Offset between Receiver and Transmitter

The required channel filter bandwidth depends on the frequency error between receiver and transmitter and the occupied bandwidth of the modulation signal used.

Section 4.1.1.1 “System Design with Compensation of Initial XTAL and XTO Tolerances” on page 286 shows the main impact of the XTO on the worst case frequency error between receiver and transmitter. The maximum frequency error between receiver and transmitter has to be determined for the individual application and the channel filter bandwidth has to be chosen with enough width to prevent unintentional loss of sensitivity, in the case of worst case frequency errors. The channel bandwidths of the Atmel ATA5785 can be configured between 25kHz and 360kHz. The required register settings are described in Section “Channel Filter Bandwidths” on page 99.

Table 4-1 shows the measured frequency offset values with 1.5dB sensitivity loss for FSK systems compared with values calculated using equation (46). If using NRZ encoding instead of Manchester encoding, equation (47) can be used, replacing data rate (DR) by symbol rate (SR). As can be seen, the calculated maximum frequency offset differs from measurement in cases of high occupied bandwidth compared to the channel filter bandwidth and in cases of high modulation indices. Because this behavior is a property of the RxDSP, it is highly reproducible and a measurement result can be used for system design without the need to consider part-to-part variations.

$$\Delta f_{S1.5dBFSK} = \frac{BW_{IF}}{2} - \sqrt{DR_{FSK}^{2} + f_{DEV}^{2}} \qquad (46)$$

$$DR = \frac{SR}{2} \qquad (47)$$

Table 4-1. Filter Settings for each Supported IF Bandwidth

Channel Filter Bandwidth	Manchester-Coded FSK Data Rate and Frequency Deviation	Measured Frequency Offset $\Delta f_{S1.5dBFSK\_measured}$	Calculated Frequency Offset with Equation
25kHz	0.75kBit ±0.75kHz	±10.7kHz	±11.44kHz
	2.4kBit ±2.4kHz	±9.4kHz	±9.10kHz
	5kBit ±2.4kHz	±6.8kHz	±6.95kHz
80kHz	2.4kBit ±2.4kHz	±35kHz	±36.6kHz
	10kBit ±10kHz	±27.5kHz	±25.8kHz
	20kBit ±20kHz	±17.5kHz	±11.7kHz
165kHz	5kBit ±5kHz	±72.5kHz	±75.4kHz
	10kBit ±10kHz	±70kHz	±68.3kHz
	20kBit ±20kHz	±57.5kHz	±54.2kHz
	1kBit ±16kHz	±60kHz	±66.5kHz
	10kBit ±16kHz	±70kHz	±63.6kHz
366kHz	10kBit ±10kHz	±170kHz	±168.8kHz
	20kBit ±20kHz	±155kHz	±154.7kHz
	10kBit ±38kHz	±135kHz	±143.7kHz

Figure 4-1 shows the sensitivity loss versus frequency offset for FSK with DR = 2.4kBit/s and f_{DEV} = ±2.4kHz using a 80kHz channel filter bandwidth. Figure 4-2 shows the sensitivity loss versus frequency offset for DR = 10kBit/s and f_{DEV} = ±10kHz using a 165kHz channel filter bandwidth. It can be seen that the 1.5dB additional loss in the link budget can be avoided using a slightly higher bandwidth as already discussed in Section 4.1.1.1 “System Design with Compensation of Initial XTAL and XTO Tolerances” on page 286. In the 80kHz bandwidth, FSK DR = 2.4kBit/s and f_{DEV} = ±2.4kHz case using only ±30kHz maximum frequency offset instead of ±35kHz avoids the additional 1.5dB sensitivity loss in the link budget. For the other channel filter bandwidth and data rate examples the behavior is equivalent. This can be taken into account by lowering the maximum frequency offset by $BW_{IF}/16$. Equation (48) should be used to incorporate this into the system design. The ripple in the sensitivity shown in Figure 4-1 and Figure 4-2 is caused by ±0.5dB measurement uncertainty and is not a property of the Atmel® ATA5785.

Figure 4-1. Measured Sensitivity Loss versus Frequency Offset (BW = 80kHz, FSK, DR = 2.4kBit/s, fDEV = ±2.4kHz)

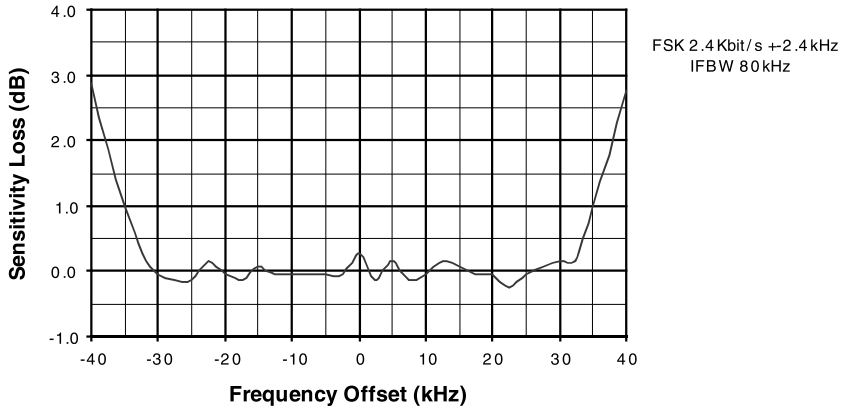
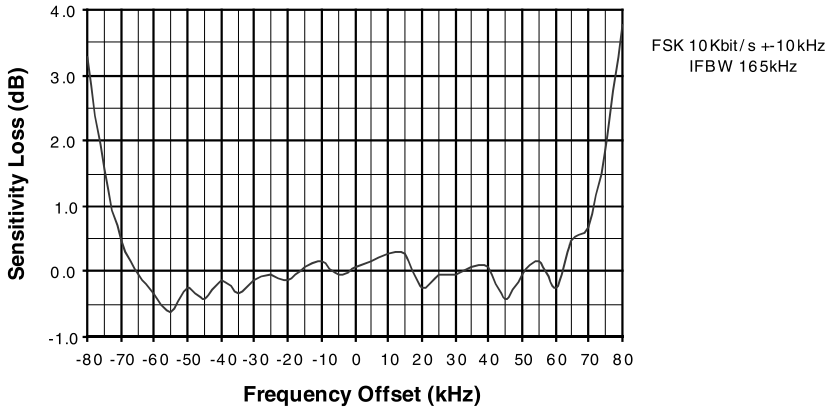


Figure 4-2. Measured Sensitivity Loss versus Frequency Offset (BW = 165kHz, FSK, DR = 10kBit/s, fDEV = ±10kHz)



$$\Delta f_{max\ FSK} = \frac{BW_{IF}}{2} - \sqrt{DR_{FSK}^{2} + f_{DEV}^{2}} - \frac{BW_{IF}}{16} \qquad (48)$$

Table 4-2 on page 289 shows the measured and calculated maximum frequency offsets derived from equation (48) for low loss in sensitivity. Atmel recommends using these numbers for system design.

Table 4-2. Measured Frequency Offset for Low Sensitivity Loss Using an FSK System

Channel Filter Bandwidth	Manchester-Coded FSK Data Rate and Frequency Deviation	Recommended maximum Frequency Offset for Low Loss Based on Measurement	Calculated Frequency Offset with Equation
25kHz	0.75kBit ±0.75kHz	±9.1kHz	±9.9kHz
	2.4kBit ±2.4kHz	±7.8kHz	±7.5kHz
	5kBit ±2.4kHz	±5.2kHz	±5.4kHz
80kHz	2.4kBit ±2.4kHz	±30kHz	±31.6kHz
	10kBit ±10kHz	±22.5kHz	±20.8kHz
	20kBit ±20kHz	±12.5kHz	±6.7kHz
165kHz	5kBit ±5kHz	±62.2kHz	±65.1kHz
	10kBit ±10kHz	±59.7kHz	±58kHz
	20kBit 20kHz	±47.2kHz	±43.9kHz
	1kBit 16kHz	±49.7 kHz	±56.2kHz
	10kBit ±16kHz	±59.7kHz	±53.3kHz
366kHz	10kBit ±10kHz	±147kHz	±145.9kHz
	20kBit ±20kHz	±132kHz	±131.8kHz
	10kBit ±38kHz	±112kHz	±120.8kHz

Table 4-3 shows the measured frequency offset values for 1.5dB sensitivity loss compared with values calculated with equation (52). The calculated maximum frequency offsets are in most cases lower than the measured ones. Therefore the calculated maximum frequency offsets can be used for system design using ASK.

$$\Delta f_{S1.5dBASK} = \frac{BW_{IF}}{2} - DR_{ASK} \quad (49)$$

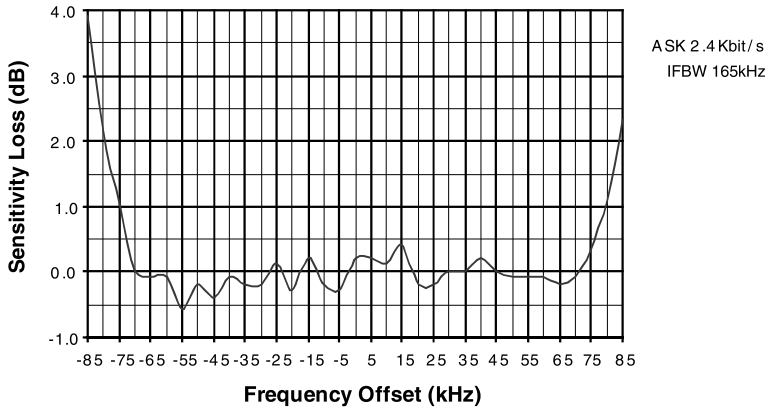
Table 4-3. Frequency Offset for 1.5dB Sensitivity Loss Using an ASK System

Channel Filter Bandwidth	Manchester-Coded ASK Data Rate	Measured Frequency Offset $\Delta f_{S1.5dBASK\_measured}$	Calculated Frequency Offset with Equation (52)
25kHz	2.4kBit/s	±11.2kHz	±10.1kHz
	5kBit/s(1)	±11.2*1)kHz	±7.5kHz
80kHz	2.4kBit/s	±37.5kHz	±37.6kHz
	10kBit/s	±36kHz	±30kHz
	20kBit/s(1)	±36(1)kHz	±20kHz
165kHz	2.4kBit/s	±77.5kHz	±80kHz
	20kBit/s	±66kHz	±62.5kHz
366kHz	2.4kBit/s	±180kHz	±180.6kHz
	20kBit/s	±170kHz	±163kHz

Note: 1. In the case of high ASK data rates compared with the bandwidth, the sensitivity versus frequency offset curve is not as flat as in other cases. For these data rates the frequency offset for 3dB loss of sensitivity is given in Table 4-3 and Table 4-4.

Figure 4-3 shows the sensitivity versus frequency offset for ASK with DR = 2.4kBit/s using 165kHz channel filter bandwidth. It can be seen that the 1.5dB additional loss in the link budget can also be avoided using $BW_{IF}/16$ lower frequency offset as in the FSK case, therefore equation (50) should be used for system design.

Figure 4-3. Measured Sensitivity Loss versus Frequency Offset (BW = 165kHz, ASK, DR = 2.4kBit/s)



$$\Delta f_{max\ ASK} = \frac{BW_{IF}}{2} - DR_{ASK} - \frac{BW_{IF}}{16} \qquad (50)$$

Figure 4-4. Measured Sensitivity Loss versus Frequency Offset (BW = 80kHz, ASK, DR = 20kBit/s)

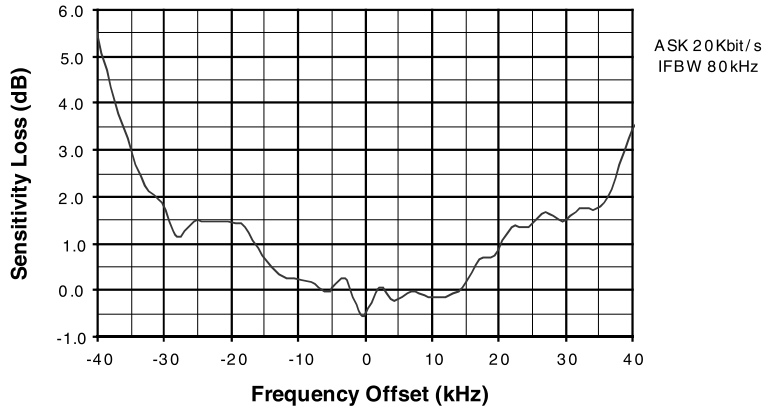


Table 4-4 shows the measured and calculated maximum frequency offset derived from equation (50) for low loss in ASK sensitivity. It is advisable to use this for system design.

Table 4-4. Max Frequency Offset for Low Sensitivity Loss Using ASK

Channel Filter Bandwidth	Manchester Data Rate	Recommended Maximum Frequency Offset for Low Loss Based on Measurement	Calculated Frequency Offset with Equation
25kHz	2.4kBit/s	±9.6kHz	±8.5kHz
	5kBit/s*1)	±9.6kHz *1)	±5.9kHz
80kHz	2.4kBit/s	±32.5kHz	±32.5kHz
	10kBit/s	±31kHz	±25kHz
	20kBit/s *1)	±31kHz(1)	±15kHz
165kHz	2.4kBit/s	±67kHz	±69.5kHz
	20kBit/s	±55.5kHz	±52.0kHz
366kHz	2.4kBit/s	±157kHz	±158kHz
	20kBit/s	±147kHz	±140kHz

Note: 1. See description in Table 4-3 on page 289.

4.1.2 Fractional-N Divider and SDM

4.1.2.1 Frequency Setting

Because the phase-locked loop (PLL) guarantees $f_{DIV} = f_{XTO}$, the fractional divider setting and the sigma delta modulator (SDM) together control the output frequency.

The frequency adjustment is done by the ffreq[17:0] control signal which defines the fractional part of the division factor. FEMS.M[3:0] and FEMS.S[3:0] define the integer portion of the division factor. For each channel in service 3 and 4 a set of data exists in the SRAM and the firmware copies these values into the front end registers before activating the PLL.

The RF divider and the fractional divider settings of the PLL are shown in Table 4-5. Values of FEMS.M[3:0] and FEMS.S[3:0] are ambiguous and should therefore be calculated by using the Atmel configuration tool.

Table 4-5. RF Divider and Main Swallow Counter Setting of the Fractional-N PLL

Frequency Bands [MHz]	RF Divider Ratio	FEMS.M[3:0]	FEMS.S[3:0]	FECR.S4N3
310.00 … 318.00	6	6	0;1;2;3	0
418.00 … 477.00	4	8;9	0;1;2; … ;(M-3)	1

The fractional-N PLL output frequency $f_{out} = f_{LO}$ as a function of the settings can be calculated with:

$$f_{OUT} = f_{XTO} \times \left(\frac{ffreq + 0.5}{2^{17 + LBNHB}} + M \times 2^{2 - LBNHB} + S \times 2^{-(1 + LBNHB)} \right) \quad (51)$$

ffreq as a function of frequency and other settings can be calculated with:

$$f_{freq} = \left(\frac{f_{OUT}}{f_{XTO}} - M \times 2^{2 - LBNHB} - S \times 2^{-(1 + LBNHB)} \right) \times 2^{(17 + LBNHB)} - 0.5 \quad (52)$$

using:

ffreq = ffreq[17:0] = {16384,16385,…, 180224}

LBNHB = 1

f_{XTO} = {23.800,…., 26.200} MHz; PLL reference frequency

M = FEMS.M[3:0] = {2, 3, …, 15};

S = FEMS.S[3:0] = {0, 1, …, M-3};

The FFREQ values which are configured in the SRAM services correspond to the receive mode and represent the center receive frequency minus the intermediate frequency (≈251kHz).

If required, e.g., for XTO temperature compensation, these values are modified automatically by the firmware before they are written to the hardware registers.

4.1.2.2 SDM Integer Boundaries and Instable Control Signal ffreq[17:0] Ranges

The range of the sigma delta modulator control word ffreq[17:0] is from 0 to 262143. The range from 16384 to 180224 must be used to allow stable operation of the SDM. Values outside this range shall not be used.

If certain ranges of ffreq[17:0] are used, the fractional-N PLL works close to frequencies at which integer boundary spurious arises that cannot be dithered by the sigma delta modulator. These ranges should therefore be avoided by choosing another XTAL frequency. In the Atmel configuration tool a warning is shown if these forbidden ranges are used.

Table 4-6. Ranges of ffreq[17:0] Causing Integer Boundary Spurious of the Fractional-N PLL

Integer Dividing ffreq[17:0] Value	Range ffreq[17:0] that Should not Be Used	FEMS.S[3:0] Value
2^{16} = 65536	58982 to 72089	3 or 7
2^{16} = 65536	62259 to 68812	other values
$2*2^{16}$ = 131072	124518 to 137625	2 or 6
$2*2^{16}$ = 131072	127795 to 134348	other values

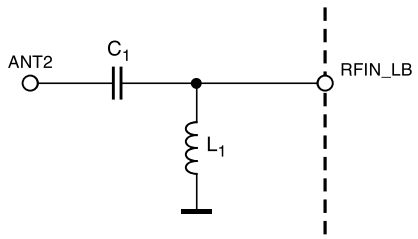
4.1.3 Impedance Matching of RFIN

4.1.3.1 Overview

In Atmel® ATA5785 the RF input must be connected directly to the antenna as depicted in Figure 4-6.

Figure 4-5 indicates the simplified matching circuit from the application board as shown in Section 4.2 “Application Board Design” on page 295.

Figure 4-5. Input Matching on Application Board



4.1.3.2 RFIN Matching

The input impedance of the LNA depends on the RF frequency. The values shown in parameter no. 8.50 in Section 5. “Electrical Characteristics” on page 300 are measured on the application board described in Section 4.2 “Application Board Design” on page 295 by connecting the measurement ground to the ground of L1 and measuring the impedance at the other connection of L1. The capacitance of the application board is calibrated by measuring the unpopulated board and removing the capacitance value from the measurement above.

Most RF performance data found in Section 5.5 “RF Receiving Characteristics” on page 302 of the electrical characteristics is measured on the application board with RFIN_LB power matched to Ant2 of the application board.

To match the Atmel ATA5785 RF inputs to Ant2 of the application board, the components described in Table 4-7 are required. It is important for the Atmel ATA5785 that a DC path from pin 1 to ground exists. If the matching circuit is changed, an inductor should always be placed from RFIN_LB to ground. The quality factor Q_L of the matching inductors causes losses which result in diminished sensitivity. These losses can be estimated when calculating the parallel equivalent resistance of the inductor with $R_{Loss} = 2 \times \pi \times f \times L \times Q_L$ and the matching loss with $10 \times \log10(1 + Rp/R_{Loss})$. Care must therefore be taken when choosing the component type of these inductors. For 315MHz the R_{Loss} calculates to 2.3kΩ using the values from Table 4-7, Rp = 870Ω is taken from parameter no. 8.50 in Section 5. “Electrical Characteristics” on page 300 and the resulting loss is 1.4dB. This means that sensitivity can be increased by choosing a higher grade inductor (such as a wire-wound inductor).

Table 4-7. Rx Matching Element Values (Ant2)

Frequency f_{RF} [MHz]	L_1	C_1	Inductor	Q Inductor at f_{RF}
315	39nH	3.3pF	TOKO LL1608FS39N	30
433.92	22nH	2.7pF	TOKO LL1608FS22N	35

4.1.4 External LNA Support

The sensitivity of the receiver can be increased by using an external low-noise amplifier. The Atmel® ATA5785 offers the output pin 29/RX_ACTIVE which can be used to bias the external LNA. The pin signalizes the active time of the receive path. It can supply 5mA current for a 4.5V to 5.5V application and 1.5mA for a 1.9V to 3.6V application while delivering a minimum of $0.9 \times V_{VS}$ as supply voltage for the LNA (for more information, see parameter no. 15.60 in Section 5. “Electrical Characteristics” on page 300).

The achievable sensitivity improvement can be calculated using a cascaded noise figure calculation according to Friis's formula. Be aware that Friis's formula uses the noise factors instead of noise figures and a linear power gain. The noise figure of Atmel ATA5785 is 7.5dB in Low-Band.

For example, an external LNA with a 3dB noise figure and 10dB gain improves the system noise figure in Low-Band from typical 7.5dB to 3.9dB. Therefore, the sensitivity values of parameters no. 4.90 to no. 5.60 in Section 5. “Electrical Characteristics” on page 300 are improved by typically 3.6dB.

4.1.5 Receiver Blocking and Selectivity Characteristics

This section describes the blocking behavior of the Atmel® ATA5785. The numbers listed are the 3dB blocking characteristics. The measurements below are carried out with a useful signal at a frequency of f_{RF} and a level of 3dB above the BER = 10^{-3} sensitivity levels as shown in parameter nos. 4.90 to 5.20 in Section 5. “Electrical Characteristics” on page 300. The values indicated describe how much a continuous wave disturber can be larger than these useful signal levels until the BER drops back to 10^{-3}. For example, the sensitivity level is –106.5dBm (see parameter no. 5.10 in Section 5. “Electrical Characteristics” on page 300) and at a distance of 1MHz the blocking is 64dBC (see parameter no. 7.10 in Section 5. “Electrical Characteristics” on page 300 or Figure 4-6). The useful signal applied in the measurement is –106.5dBm + 3dB = –103.5dBm and the measured absolute blocker level for BER 10^{-3} is –103.5dBm + 64dBC = –39.5dBm.

Figure 4-6, Figure 4-7, and Figure 4-8 on page 294 show the blocking behavior for 433.92MHz with 165kHz bandwidth using FSK modulation with a data rate of DR = 20kBit/s and a frequency deviation of f_{DEV} = ±20KHz. All measurements are with high resolution to capture all spurious receiving frequencies.

Figure 4-6. Measured Blocking at ±1MHz Frequency Offset (f_{RF} = 433.92MHz, BW = 165kHz, FSK, DR = 20kBit/s, f_{DEV} = ±20kHz)

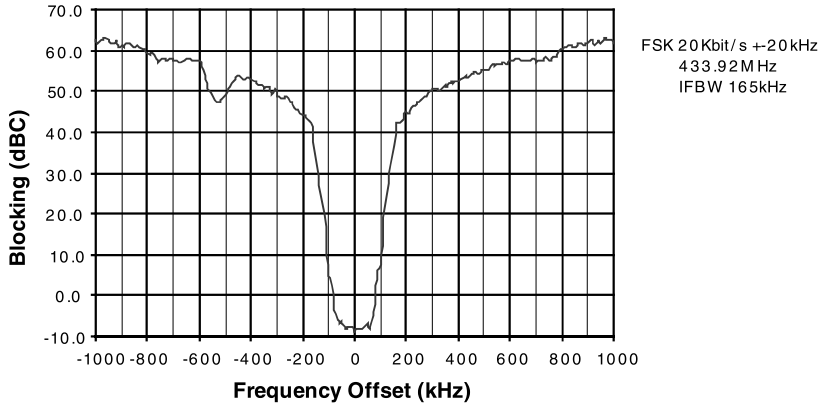


Figure 4-7. Measured Blocking at ±10MHz Frequency Offset (f_{RF} = 433.92MHz, BW = 165kHz, FSK, DR = 20kBit/s, f_{DEV} = ±20kHz)

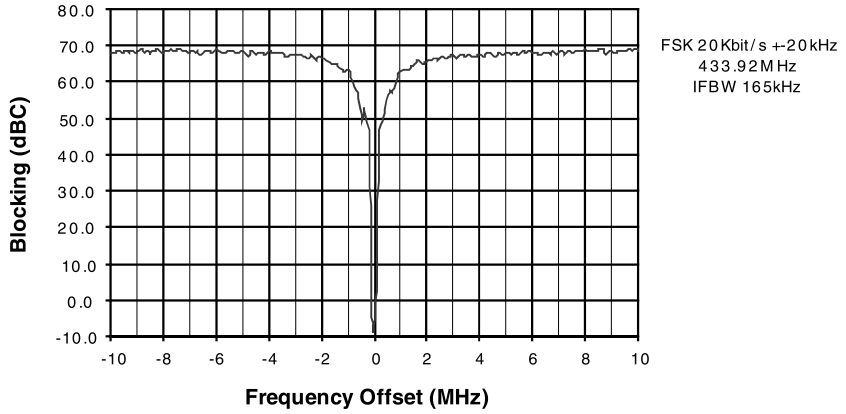
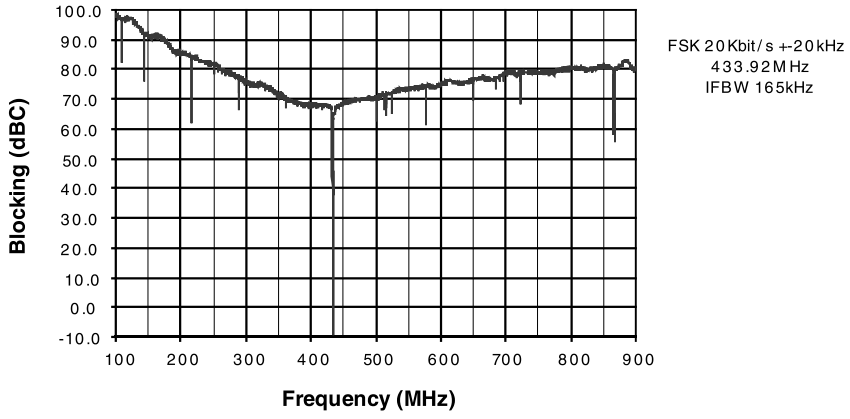


Figure 4-8. Measured Blocking in the Frequency Range from 100MHz to 900MHz (f_{RF}=433.92MHz, BW = 165kHz, FSK, DR = 20kBit/s, f_{DEV} = ±20kHz)



The differences in the blocking characteristics for other RF frequencies or bandwidths are shown in parameter nos. 6.90 to 7.80 in Section 5. “Electrical Characteristics” on page 300. To describe the differences more clearly the blocking for 25kHz IF bandwidth and 360kHz IF bandwidth is depicted in Figure 4-9 and Figure 4-10.

Figure 4-9. Measured Blocking at ±1MHz Frequency Offset (f_{RF}=433.92MHz, BW=25kHz, FSK, DR=2.4kBit/s, f_{DEV}=±2.4kHz)

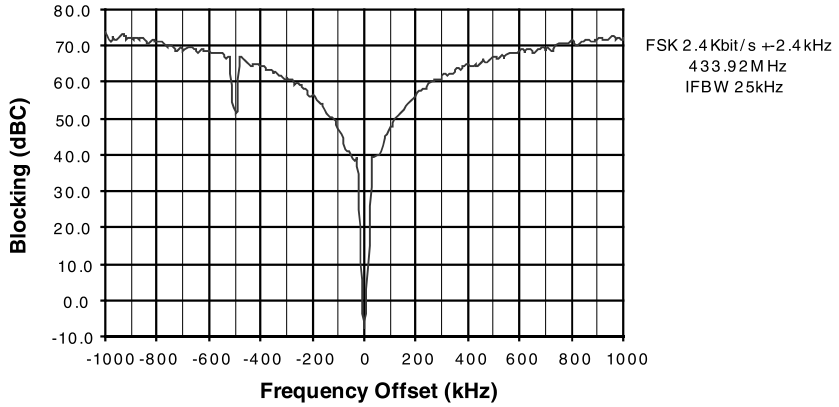
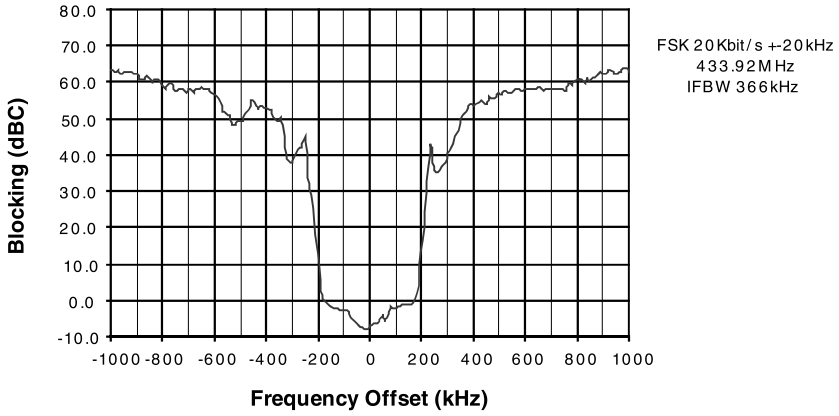


Figure 4-10. Measured Blocking at ±1MHz Frequency Offset (f_{RF}=433.92MHz, BW=366kHz, FSK, DR=20kBit/s, f_{DEV}=±20kHz)



4.2 Application Board Design

This section describes the application and reference board design for the Atmel® ATA5785 and gives recommendations about optimizing the layout for the Atmel ATA5785 application.

Figure 4-11 shows the schematic of the Atmel ATA5785 application board. The layout is illustrated in Figure 4-12 on page 297 and the component diagram is depicted in Figure 4-13 on page 297. Table 4-8 on page 298 and Table 4-9 on page 299 summarize the corresponding board populations for 315MHz and 433.92MHz respectively. The boards are manufactured using FR4 material with a PCB thickness of 1mm.

The electrical characteristics in Section 5. “Electrical Characteristics” on page 300 are measured using a matching network between RFIN_LB and Ant2.

The matching networks of the receive path is found in Section 3.4.2.1 “LNA and Mixer” on page 96.

Figure 4-11. Schematic of the ATA5785 Application Board

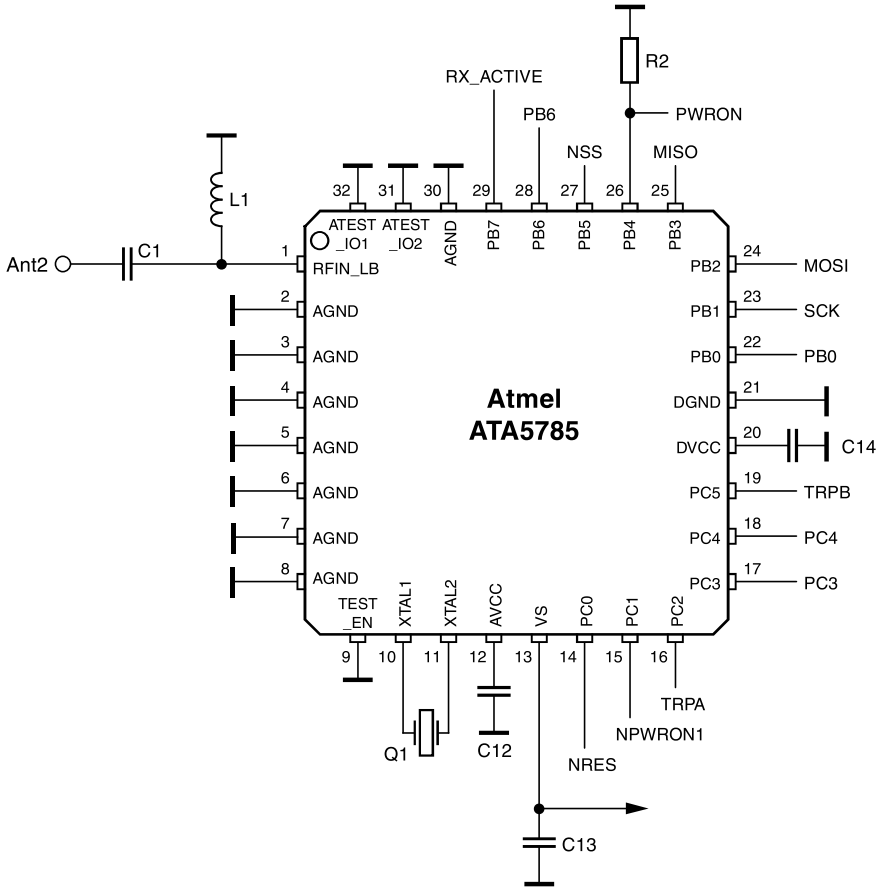


Figure 4-12. Layout of the ATA5785 Application Board

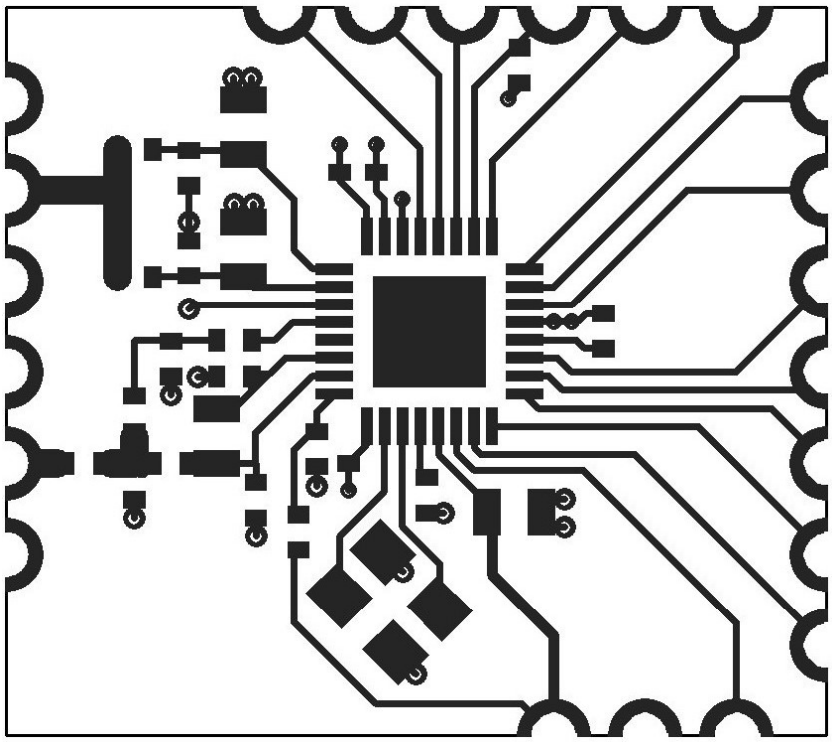
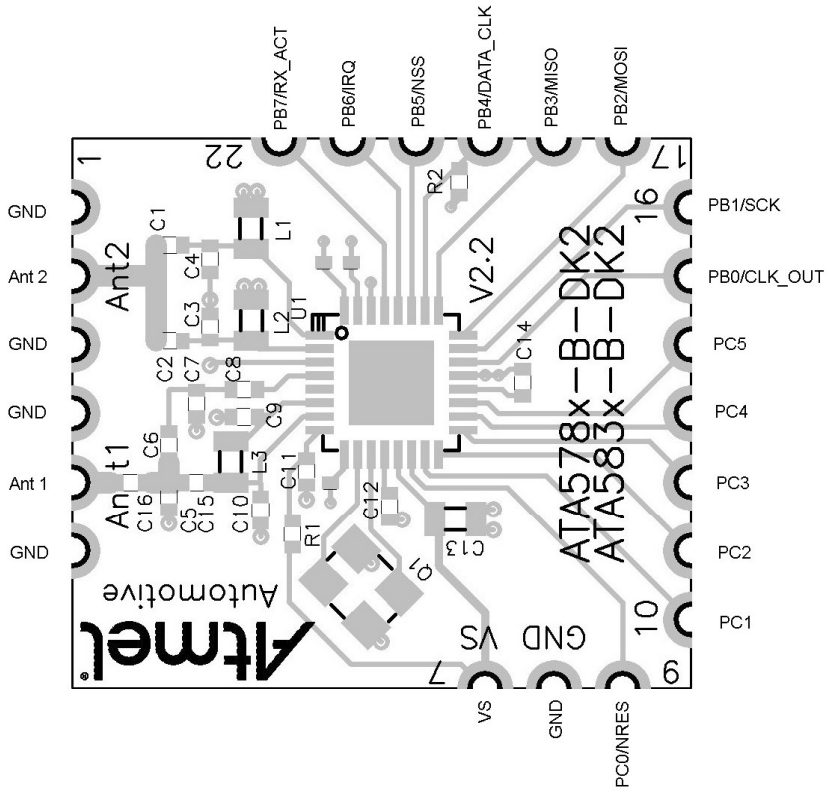


Figure 4-13. Component Diagram of the ATA5785 Application Board



The following items must be considered during the design of a layout for an Atmel® ATA5785 application board:

1. The decoupling capacitor of AVCC, C_{12} must be placed as close as possible to the pin 12 because otherwise the series inductance is too high and the supply bypassing is no longer effective at high frequencies.
2. The decoupling capacitor of DVCC, C_{14} must be placed as close as possible to the pin 20. This decoupling capacitor should be connected directly to the DGND pin and ground layer using vias as seen in Figure 4-12 on page 297. Otherwise, the sensitivity of the receiver may be worsened by the spurious clock emissions of the integrated AVR.
3. The decoupling capacitor of VS, C_{13} must be placed as close as possible to the pin 13, because otherwise the series inductance is to high and supply bypassing is no longer effective at high frequencies.
4. Direct connection of the DGND pin to the exposed die pad must be avoided and at least four vias must be placed under the exposed die pad.If this is not done, the isolation of the integrated AVR from the RF frontend is worse. The exposed die pad is also the RF ground and reduced sensitivity may result from bad ground connection on the exposed die pad.
5. The crystal must be placed as close as possible to the IC to avoid extra capacitance on XTAL1 and XTAL2.
6. It is advisable to design the lines carrying the RF signal as short as possible and place the elements of the matching networks as close as possible to the IC.
7. Avoid routing XTAL, AVCC, and VS lines in parallel and close to each other over long distances; doing so reduces the coupling of the XTO signals to the supply voltage. Failing to do so may cause spurious receiver emissions.
8. Avoid routing XTAL1 and XTAL2 lines in parallel and close to each other over long distances, to avoid a reduction of the XTO oscillation margin.

Table 4-8 and Table 4-9 on page 299 show the bill of material of the application boards for 315MHz and 433.92MHz applications. The required power supply for the application board is 3V. Connecting 5V to the Atmel ATA5785 PCB populated for 3V application destroys the IC. Components not mentioned in Table 4-8 and Table 4-9 on page 299 are not mounted.

Table 4-8. Bill of Material of the Application Board for 315MHz, 3V Application

Operating Frequency is 315MHz				
Component	**Value**	**Material / Series**	**Housing**	**Manufacturer / Distributor**
R2	100k		0402	
Q1	24.305MHz	DSX321SL		KDS
C1	3.3pF	COG	0402	Murata
C3	0Ω		0402	
C7	0Ω		0402	
C8	0Ω		0402	
C9	0Ω		0402	
C10	0Ω		0603	
C11	0Ω		0402	
C12	220nF	X7R	0402	TY
C13	2.2µF	X5R	0402	Murata
C14	22nF	X7R	0402	Murata
L1	39nH	LL-1608-FSH	0603	TOKO

Table 4-9. Bill of Material of the Application Board for 433.92MHz, 3V Application

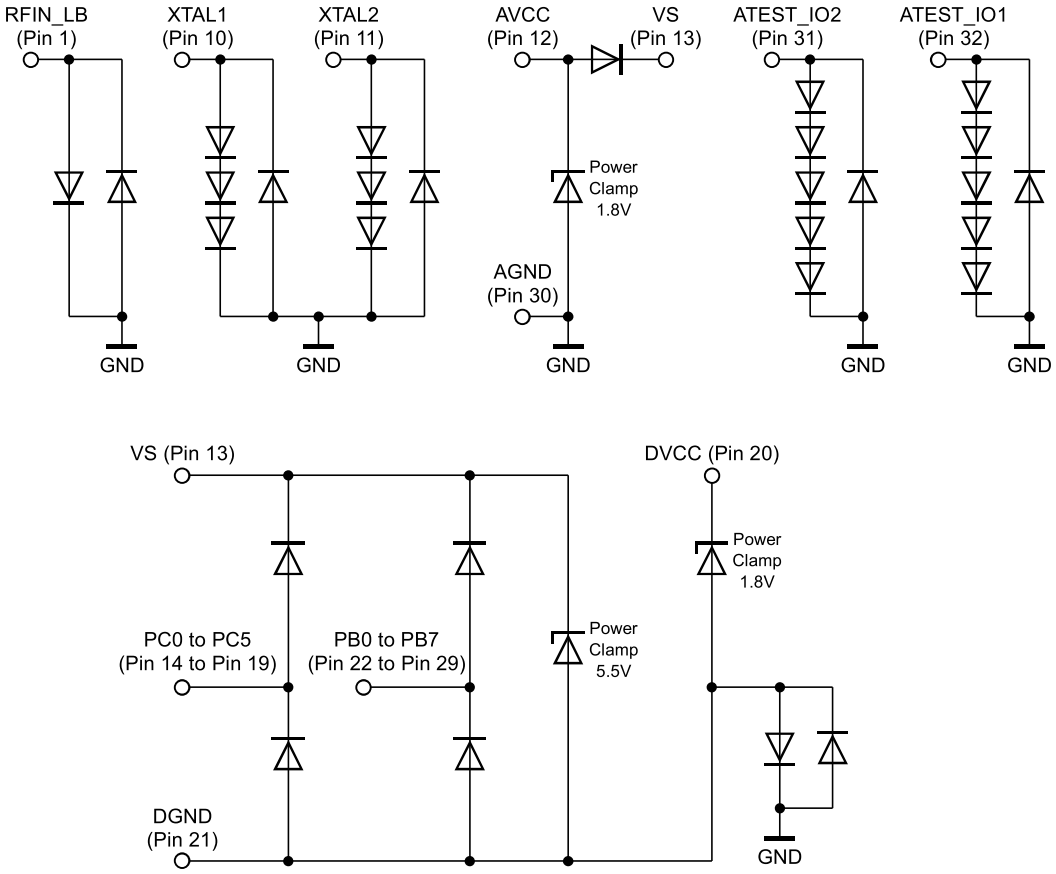
Operating Frequency is 315MHz				
Component	**Value**	**Material / Series**	**Housing**	**Manufacturer / Distributor**
R2	100k		0402	
Q1	24.305MHz	DSX321SL		KDS
C1	2.7pF	COG	0402	Murata
C3	0Ω		0402	
C7	0Ω		0402	
C8	0Ω		0402	
C9	0Ω		0402	
C10	0Ω		0603	
C11	0Ω		0402	
C12	220nF	X7R	0402	TY
C13	2.2µF	X5R	0402	Murata
C14	22nF	X7R	0402	Murata
L1	22nH	LL-1608-FSH	0603	TOKO

5. Electrical Characteristics

5.1 ESD Protection Circuits

GND is the exposed die pad of the Atmel® ATA5785 which is internally connected to AGND (pin 30). All Zener diodes shown in Figure 5-1 (marked as power clamps) are realized with dynamic clamping circuits and not physical Zener diodes. Therefore, DC currents are not clamped to the shown voltages.

Figure 5-1. Atmel ATA5785 ESD Protection Circuit



5.2 Absolute Maximum Ratings

Stresses beyond 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 at these or any other conditions beyond 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.

Parameters	Symbol	Min.	Max.	Unit
Junction temperature	Tj		+150	°C
Storage temperature	Tstg	–55	+125	°C
Ambient temperature	Tamb	–40	+105	°C
Supply voltage	V_{VS}	–0.3	6.0	V
Maximum input level (input matched to 50Ω)	P_{in_max}		+10	dBm
ESD (Human Body Model) all pins	HBM	–4	+4	kV
ESD (Machine Model) all pins	MM	–200	+200	V
ESD (Field Induced Charged Device Model) all pins	FCDM	–750	+750	V

5.3 Thermal Resistance

Parameters	Symbol	Value	Unit
Thermal resistance, junction ambient, soldered according to JEDEC	R_{th_JA}	35	K/W

5.4 Supply Voltages and Current Consumption

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel® settings are used unless marked with *1.

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
1.00	Supply voltage range VS		13	V_{VS}	1.9	3.0	5.5	V	A
1.05	Supply voltage rise time		13	V_{VS_rise}			1	V/µs	D
1.20	OFFMode Current consumption	T_{amb} = 25°C T_{amb} = 85°C T_{amb} = 105°C	8, 13	$I_{OFFMode}$		5	150 600 5,000	nA	B B A
1.30	IDLEMode(RC) Current consumption	SRC active, AVR in power-down mode, temperature range –40°C to +65°C	13	$I_{IDLEMode(RC)}$		50	90	µA	B
1.40	IDLEMode(XTO) Current consumption	XTO active, AVR in power-down mode	13	$I_{IDLEMode(XTO)}$		250	400	µA	B
1.60	IDLEMode(XTO) Current consumption	With active CLK_OUT f_{XTO}/6 = 4.05MHz $C_{LOAD\_CLK\_OUT}$ = 10pF V_{VS} = 5.5V AVR running with f_{XTO}/4 = 6.076MHz	13, 22	$I_{IDLEMode(XTO)\_CLK\_OUT2}$		1.3	2.5	mA	B
1.80	RXMode Current consumption	AVR running with f_{XTO}/4 f_{RF} = 315MHz *1 f_{RF} = 433.92MHz	13	$I_{RXMode1}$		9.2 9.8	12.7 13.2	mA	B A

Note: *) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter
Pin number in brackets mean, that they are measured matched to 50Ω on the application board.

5.5 RF Receiving Characteristics

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with [*1].

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
Frequency Ranges and Frequency Resolution of PLL for RXMode and PollingMode									
3.00	RF operating frequency range 315MHz Low-Band	FECR.S4N3 = '0'	1, 7	$f_{Range\_LB1\_315}$	310	315	318	MHz	A
3.10	RF operating frequency range 433MHz Low-Band	FECR.S4N3 = '1'	1, 7	$f_{Range\_LB2\_433}$	418	433.92	477	MHz	B
3.40	Frequency resolution PLL	Low-Band $f_{XTO}/2^{18}$	1, 2, 7	DF_{PLL}		92.72		Hz	B
RXMode and PollingMode Receive Characteristics IF bandwidth specifications are examples usable for parameter extrapolation if other IF bandwidth values are used									
4.00	Receiver 3dB bandwidth	Programmable digital IF filter	1, 2	BW_{IF}	25		366	kHz	B
4.10	ASK and FSK transparent mode data rate Manchester mode	at 25kHz IF-BW at 50kHz IF-BW at 80kHz IF-BW at 165kHz IF-BW at 237kHz IF-BW at 366kHz IF-BW	1, 2	DR_{TM}	0.25		7 14 20 50 80 80	Kbit/s	B B B B B B
4.20	Modulation index FSK	η = frequency_deviation / symbol_rate recommended	1, 2	η	0.5 0.75	1	360 1.25		B B
4.30	Frequency deviation	Maximum usable frequency deviation is baseband clock dependant f_{DEV_Max} = CLK_BB/8 at 25kHz IF-BW at 50kHz IF-BW at 80kHz IF-BW at 165kHz IF-BW at 237kHz IF-BW at 366kHz IF-BW	1, 2	f_{DEV}	 ±0.375 ±0.75 ±1.2 ±2.5 ±3.5 ±5.4		 ±9 ±18 ±26 ±60 ±93 ±93	kHz	 B B B B B B
4.40	ASK and FSK transparent mode symbol rate NRZ mode	Used to receive NRZ, Keyloq, PPM, 1/3 2/3 Coded telegrams at 25kHz IF-BW at 50kHz IF-BW at 80kHz IF-BW at 165kHz IF-BW at 237kHz IF-BW at 366kHz IF-BW	1, 2	SR_{TM_OPT}	0.5		 14 28 40 100 160 160	Ksym/s	 B B B B B B

Note: *) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter
Pin number in brackets mean, that they are measured matched to 50Ω on the application board.

5.5 RF Receiving Characteristics (Continued)

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with [*1].

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
4.70	Data rate tolerance FSK and ASK	Loss of sensitivity <1dB	1, 2	DR_{TOL}	–10		+10	%	B
4.80	Buffered data rate Manchester and NRZ mode	TMDO output will be buffered internally and readout via SPI interface Manchester mode NRZ mode	1, 2	$DR_{Buffered}$	 0.25 0.5		 80 120	 Kbit/s Ksym/s	 B B
4.90	ASK/FSK Sensitivity level 315MHz/ 433.92MHz Manchester encoded Receiving 100bit Packets with 9 of 10 Packets Error Free or BER = 10^-3 Continuous RX measured at TMDO output or buffered via SPI for DR < $DR_{Buffered}$	FSK at 25kHz IF bandwidth T_{amb} = 25°C 0.75Kbit/s ± 0.75kHz 5Kbit/s ± 2.4kHz	(1), 17, 19	$SFSK_{B25\_R0.75}$ $SFSK_{B25\_R5\_2.4}$	–1.5dB	–122.5 –113.5	+1.5dB	dBm	B B
5.00		FSK at 80kHz IF bandwidth T_{amb} = 25°C 2.4Kbit/s ± 2.4kHz 20Kbit/s ± 20kHz	(1), 17, 19	$SFSK_{B80\_R2\_4}$ $SFSK_{B80\_R20}$	–1.5dB	–117 –108.5	+1.5dB	dBm	B B
5.10		FSK at 165kHz IF bandwidth T_{amb} = 25°C 5Kbit/s ± 5kHz 40Kbit/s ± 40kHz	(1), 17, 19	$SFSK_{B165\_R5}$ $SFSKB_{165\_R40}$	–1.5dB	–114 –105.5	+1.5dB	dBm	B B

Note: *) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter
Pin number in brackets mean, that they are measured matched to 50Ω on the application board.

5.5 RF Receiving Characteristics (Continued)

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with *1.

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
5.20	ASK/FSK Sensitivity level 315MHz/ 433.92MHz Manchester encoded Receiving 100bit Packets with 9 of 10 Packets Error Free or BER = 10^-3 Continuous RX measured at TMDO output or buffered via SPI for DR < $DR_{Buffered}$	FSK at 366kHz IF bandwidth T_{amb} = 25°C 20Kbit/s ± 20kHz 80Kbit/s ± 80 kHz	(1), 17, 19	$SFSK_{B366\_R20}$ $SFSK_{B366\_R80}$	–1.5dB	–107.5 –100.5	+1.5dB	dBm	B B
5.30		ASK at 25kHz IF bandwidth (100% ASK level of carrier value) T_{amb} = 25°C 0.5Kbit/s 5Kbit/s	(1), 17, 19	$SASK_{B25\_R0\_5}$ $SASK_{B25\_R5}$	–1.5dB	–125 –117.5	+1.5dB	dBm	B B
5.40		ASK at 80kHz IF bandwidth (100% ASK level of carrier value) T_{amb} = 25°C 1Kbit/s 20Kbit/s	(1), 17, 19	$SASK_{B80\_R1}$ $SASK_{B80\_R20}$	–1.5dB	–121.5 –110.5	+1.5dB	dBm	B B
5.50		ASK at 165kHz IF bandwidth (100% ASK level of carrier value) T_{amb} = 25°C 1Kbit/s 40Kbit/s	(1), 17, 19	$SASK_{B165\_R1}$ $SASK_{B165\_R40}$	–1.5dB	–120.5 –107.5	+1.5dB	dBm	B B
5.60		ASK at 366kHz IF bandwidth (100% ASK level of carrier value) T_{amb} = 25°C 1Kbit/s 80Kbit/s	(1), 17, 19	$SASK_{B366\_R1}$ $SASK_{B366\_R80}$	–1.5dB	–118.5 –103.5	+1.5dB	dBm	B B
5.80	Sensitivity change Full ambient temperature range	T_{amb} = –40°C to +105°C Low-Band *1 $S = S_{FSK\_ASK} + \Delta S$	(1)	ΔS_{Tamb_LB}	–1.5		2	dB	C
5.90	Sensitivity change NRZ	Compared to Machester NRZ using no more than 8 succeeding ‘0’ or ‘1’ symbols FSK ASK $S = S_{FSK\_ASK} + \Delta S$	(1, 2)	ΔS_{NRZ}	–1 0	0 2	2 4	dB	C C
6.00	Sensitivity change TRPA/B raw data	Compared to matched filter TMDO signal on pin 17, Manchester encoded ASK FSK $S = S_{FSK\_ASK} + \Delta S$	(1, 2), 16, 17, 19	ΔS_{RAW_DATA}		2.5 1.5		dB	B B

Note: *) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter
Pin number in brackets mean, that they are measured matched to 50Ω on the application board.

5.5 RF Receiving Characteristics (Continued)

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with [*1].

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
6.20	Sensitivity change for frequency deviations lower than configured	Configured for maximum f_{DEVM}· sensitivity degradation at f_{DEVM} / 3 compared to f_{DEVM} $S = S_{FSK\_ASK} + \Delta S$	(1, 2)	$\Delta S_{:fdevM\_3}$		2	3	dB	C
6.30	Value change from ASK level to OOK level	To calculate OOK values from ASK 100% level of carrier values Example: 2.4Kbit at 165kHz IF bandwidth ASK: 100% level of Carrier –117dBm = OOK: –111dBm $S_{OOK} = S_{ASK} + \Delta_{OOK}$	(1, 2)	Δ_{OOK}	6	6	6	dB	D
6.90	315MHz/ 433MHz blocking Manchester encoded useful signal level increased 3dB above sensitivity level blocking measured relative to useful signal level Receiving 100bit Packets with 9 of 10 Packets Error Free or BER = 10^-3 Continuous RX Excluding spurious receiving frequencies	At 25kHz IF bandwidth, FSK, T_{amb} = 25°C 2.4Kbit/s ± 2.4kHz	(1, 2)	fdist. ≥ 50kHz fdist. ≥ 100kHz fdist. ≥ 225kHz fdist. ≥ 450kHz fdist. ≥ 1MHz fdist. ≥ 4MHz fdist. ≥ 10MHz		40 46 58 64 73 78 78		dBc	C C C C C C C
7.00		At 80kHz IF bandwidth, FSK, T_{amb} = 25°C 10Kbit/s ± 10kHz	(1, 2)	fdist. ≥ 150kHz fdist. ≥ 225kHz fdist. ≥ 450kHz fdist. ≥ 1MHz fdist. ≥ 4MHz fdist. ≥ 10MHz		45 52 58 67 71 71		dBc	C C C C C C
7.10		At 165kHz IF bandwidth, FSK, T_{amb} = 25°C 20Kbit/s ± 20kHz	(1, 2)	fdist. ≥ 225kHz fdist. ≥ 450kHz fdist. ≥ 1MHz fdist. ≥ 4MHz fdist. ≥ 10MHz		48 54 64 68 68		dBc	C C C C C
7.20		At 366kHz IF bandwidth, FSK, T_{amb} = 25°C 20Kbit/s ± 20kHz	(1, 2)	fdist. ≥ 500kHz fdist. ≥ 1MHz fdist. ≥ 4MHz fdist. ≥ 10MHz		55 64 68 68		dBc	C C C C
7.70	Image rejection	Low-Band no adaptive algorithm used, therefore, numbers valid if large disturber applied before useful signal	(1, 2)	IM_{RED}	45	55		dB	A
7.80	Blocking $3f_{LO}$, $5f_{LO}$	Low-Band: $3*f_{LO}-f_{IF}$ $5*f_{LO}+f_{IF}$	(1, 2)	BL_{NfLO}	27 28	32 33	37 38	dB dB	C C

Note: *) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter
Pin number in brackets mean, that they are measured matched to 50Ω on the application board.

5.5 RF Receiving Characteristics (Continued)

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with [*1].

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
7.90	Nominal IF frequency	RxDSP property depends on nominal RF frequency and DIV_IF $f_{IF} = f_{RF} / (DIV\_IF*6)$		f_{IF}	242	251	276	kHz	B
8.10	System input referred compression point	No AGC is used, therefore, the full dynamic is available receiving signals at sensitivity level on pin	(1, 2)	ICP_{1dB}		–45		dBm	B
8.15	System input referred out-of band compression point	1MHz distance to carrier	(1, 2)	ICP_{1dB_1MHz}		–35		dBm	C
8.20	System input referred 3rd-order intercept point	Low-Band	(1, 2)	IIP3		–35		dBm	C
8.30	Max, useful RX input level	System works from sensitivity level up to that level with BER = 10^{-3}	(1, 2)	P_{In_max1}	–10	+10		dBm	C
8.50	Input impedance	Measured on application board, RC parallel equivalent circuit 315MHz 433.92MHz	 1 1	Z_{in}	–20%	 870 2.9 400 2.9	 +20%	 Ω pF Ω pF	 C C C C
8.90	LO spurious at LNA input	freq > 1GHz freq < 1GHz	(1, 2)	$P_{LO\_LNA\_IN}$		–60 –86	–50 –60	dBm	C C
9.00	RSSI accuracy	P_{RFIN_LB} = –70dBm Low-Band	(1, 2), 4	$RSSI_{ABS\_ACCU}$	–5.0		+5.0	dB	B
9.10	RSSI relative accuracy	Measurement range –100dBm to –50dBm	(1, 2), 4	$RSSI_{REL\_ACCU}$	–1		+1	dB	B
9.20	RSSI resolution	DSP property	(1, 2), 4	$RSSI_{RES}$		0.5		dB/ value	D

Note: *) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter
Pin number in brackets mean, that they are measured matched to 50Ω on the application board.

5.6 Oscillators and CLK_OUT

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances, quartz parameters C_m = 4fF and C_0 = 1pF unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with *1.

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
13.00	CLK_OUT equivalent internal capacitance	Used for current calculation	13, 22	C_{CLK}		7.5	10	pF	C
13.10	Supply current increase CLK_OUT active	Calculation can be applied to all operation modes except OFFMode	13	ΔI_{CLK}	$(C_{CLK} + C_{LOAD\_CLK\_OUT}) \times V_{VS} \times f_{OUT}$			A	C
13.30	XTO frequency range		10, 11	f_{xto}	23.8	24.305	26.2	MHz	C
13.40	XTO pulling due to internal capacitance and XTO tolerance	C_m = 4fF, T_{amb} = 25°C	10, 11	ΔF_{XTO1}	–10		+10	ppm	B
13.50	XTO pulling due to temperature and supply voltage	C_m = 4fF T_{amb} = –40°C to +105°C	10, 11	ΔF_{XTO2}	–4		+4	ppm	B
13.60	Maximum C_0 of XTAL	XTAL parameter	10, 11	C_{0_max}		1	2	pF	D
13.70	XTAL, C_m motional capacitance	XTAL parameter	10, 11	C_m		4	10	fF	D
13.80	XTAL, real part of XTO impedance at start-up	C_m = 4fF, C_0 = 1pF	10, 11	R_{m_start1}	950			Ω	B
13.90	XTAL, real part of XTO impedance at start-up	C_m = 4fF, C_0 = 1pF, T_{amb} < 85°C	10, 11	R_{e_start2}	1100			Ω	B
14.00	XTAL, maximum R_m after start-up	XTAL parameter	10, 11	R_{m_max}	110			Ω	D
14.10	Internal load capacitors	Including ESD and package capacitance. XTAL has to be specified for 7.5pF load capacitance (incl. 1pF PCB capacitance per pin)	10, 11	C_{L1}, C_{L2}	13.3	14	14.7	pF	B
14.20	Slow RC oscillator frequency	Polling cycle can be calibrated ±2% accurate with f_{XTO}	22	f_{SRC}	–10%	125	+10%	kHz	A
14.30	Fast RC oscillator frequency	FRC oscillator can be calibrated ±2% accurate with f_{XTO}	22	f_{FRC}	–5%	6.36	+5%	MHz	A

Note: *) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

5.7 I/O Characteristics for Ports PB0 to PB7 and PC0 to PC5

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with [*1].

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
15.00	Input low voltage	PC0 to PC5 PB0 to PB7	14-19 22-29	V_{IL}	–0.3		0.2 x V_{VS}	V	A
15.05	Input low leakage current I/O pin	PC0 to PC5 PB0 to PB7	14-19 22-29	I_{IL}			–1	µA	A
15.10	Input high voltage	PC0 to PC5 PB0 to PB7	14-19 22-29	V_{IH}	0.8 x V_{VS}		V_{VS} + 0.3	V	A
15.15	Input high leakage current I/O pin	PC0 to PC5 PB0 to PB7	14-19 22-29	I_{IH}			1	µA	A
15.20	Output low voltage	3V application: I_{OL} = 0.2mA 5V application: I_{OL} = 0.8mA	14-19 22-29	V_{OL_3V} V_{OL_5V}			0.1 x V_{VS} 0.1 x V_{VS}	V V	A A
15.30	Output high voltage	3V application: I_{OH} = –0.2mA 5V application: I_{OH} = –0.8mA	14-19 22-29	V_{OH_3V} V_{OH_5V}	0.9 x V_{VS} 0.9 x V_{VS}			V V	A A
15.40	I/O pin pull-up resistor	OFFMode: see port B and port C	14-19 22-29	R_{PU}	30	50	70	kΩ	A
15.50	Output low voltage for strong LED low-side driver (PB7)	Configurable on pin PB7 3V application: I_{LOAD} = 1.5mA 5V application: I_{LOAD} = 5mA	29	V_{OL_STR1}			0.1 x V_{VS} 0.1 x V_{VS}	V V	A A
15.60	Output high voltage for strong LED/LNA high-side driver (PB7, PB4)	Configurable on pin PB7 and PB4 3V application: I_{LOAD} = –1.5mA 5V application: I_{LOAD} = –5mA	26, 29	V_{OH_STR1}	0.9 x V_{VS} 0.9 x V_{VS}			V V	A A
15.70	Output low voltage for strong ISP low-side driver (PB3)	Activated in ISP mode I_{OL} = 1.7mA, V_{Vs} > 2.5V T_{amb} = –40°C to +65°C	25	V_{OL_STR2}			0.1 x V_{VS} 0.1 x V_{VS}	V V	B B
15.80	Output high voltage for strong ISP high-side driver (PB3)	Activated in ISP mode I_{OH} = –1.7mA, V_{Vs} > 2.5V T_{amb} = –40°C to +65°C	25	V_{OH_STR2}	0.9 x V_{VS} 0.9 x V_{VS}			V V	B B
15.90	CLK_OUT output frequency	XTO, FRC or SRC related clock f_{CLK_OUT} = f_{OSC}/(2*CLKOD)	22	f_{CLK_OUT}			4.5	MHz	B
16.00	CLK_OUT duty cycle	$C_{LOAD\_CLK\_OUT}$ = 10pF f_{CLK_OUT} = 4.5MHz	22	DTY_{CLK_OUT}	45		55	%	A

Note: *) Type means: A = 100% tested at voltage and temperature limits, B = 100 % correlation tested, C = Characterized on samples, D = Design parameter

5.7 I/O Characteristics for Ports PB0 to PB7 and PC0 to PC5 (Continued)

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V over all process tolerances unless otherwise specified. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz. Standard Atmel settings are used unless marked with [*1].

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
16.10	I/O pin output delay time (rising edge)	3V application C_{Load} = 10pF	14-19	$T_{del\_rise\_3V}$	13.6	17.5	22.4	ns	D
		5V application C_{Load} = 10pF	22-29	$T_{del\_rise\_5V}$	9.7	12.4	15.7	ns	D
16.20	I/O pin rise time ($0.1 \times V_{VS}$ to $0.9 \times V_{VS}$)	3V application C_{Load} = 10pF	14-19	T_{rise_3V}	20.7	23.9	28.4	ns	D
		5V application C_{Load} = 10pF	22-29	T_{rise_5V}	12.7	14.3	16.6	ns	D
16.30	I/O pin slew rate (rising edge)	3V application C_{Load} = 10pF	14-19	$T_{sr\_rise\_3V}$	0.115	0.100	0.084	V/ns	D
		5V application C_{Load} = 10pF	22-29	$T_{sr\_rise\_5V}$	0.315	0.280	0.240	V/ns	D
16.40	I/O pin output delay time (falling edge)	3V application C_{Load} = 10pF	14-19	$T_{del\_fall\_3V}$	13.7	17.4	22.7	ns	D
		5V application C_{Load} = 10pF	22-29	$T_{del\_fall\_5V}$	10.4	12.2	16.0	ns	D
16.50	I/O pin fall time ($0.9 \times V_{VS}$ to $0.1 \times V_{VS}$)	3V application C_{Load} = 10pF	14-19	T_{fall_3V}	16.2	19.2	22.5	ns	D
		5V application C_{Load} = 10pF	22-29	T_{fall_5V}	10.4	12.4	13.7	ns	D
16.60	I/O pin slew rate (falling edge)	3V application C_{Load} = 10pF	14-19	$T_{sr\_fall\_3V}$	0.148	0.125	0.106	V/ns	D
		5V application C_{Load} = 10pF	22-29	$T_{sr\_fall\_5V}$	0.384	0.322	0.292	V/ns	D

Note: *) Type means: A = 100% tested at voltage and temperature limits, B = 100 % correlation tested, C = Characterized on samples, D = Design parameter

6. Timing Characteristics

6.1 Hardware Timings

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to + 105°C, V_{VS} =1.9V to 5.5V over all process tolerances. Typical values are given at V_{VS} = 5V, T_{amb} = 25°C, and for a typical process unless otherwise specified. Crystal oscillator frequency f_{XTO} = 24.305MHz, FRC frequency f_{FRC} = 6.0MHz. Standard Atmel Settings are used unless marked with *1.

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
17.00	Start-up Time XTO	AVCC already enabled and ready C0 < 1.5pF 4fF < Cm < 15fF R_m < 110Ω R_m < 800Ω	10, 11	T_{Start_XTO}	90	130	250 1500	µs µs	B C
17.50	System Initialisation Startup Time	PWRON = ‘1’ or NPWRON = ‘0’ to INTERNAL RESET removal	13,20	$T_{SYSINIT1}$	80		200	µs	B

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

6.2 System Timings

All parameters refer to GND (backplane) and are valid for T_{amb} = –40°C to +105°C, V_{VS} = 1.9V to 5.5V across all process tolerances unless otherwise specified. Standard Atmel settings are used unless marked with *1. Minimum timing values refer to maximum values of f_{FRC}; maximum timing values refer to minimum values of f_{FRC} (see Section 5. “Electrical Characteristics” on page 424 number 14.30).

Crystal Oscillator frequency f_{XTO} is set to 24.305MHz for all timing values.

6.2.1 OFFMode -> IDLEMode

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Max.	Unit	Type*
Timing Group: OFFMode --> IDLEMode(RC)								
18.10	NPWRONx LOW / PWRON HIGH --> SYS_RDY EVENT		15 (16, 17, 18, 19, 26, 29), 28	$T_{OFF2IDLERC}$	180	370	µs	C, D

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

6.2.2 IDLEMode -> RXMode

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Max.	Unit	Type*
Timing Group: IDLEMode(XTO) --> RXMode								
21.00	Last SPI clock --> TRPA first edge	Raw transparent output activated	23, 16	$T_{IDLEXTO2RX}$	430	590	µs	C, D
	Additional time for: • Channel filter bandwidth 25kHz..45kHz • Channel filter bandwidth 50kHz..71kHz • Channel filter bandwidth 80kHz..146kHz • Channel filter bandwidth 165kHz..219kHz (default) • Channel filter bandwidth 237kHz..366kHz • RF calibration • VCO tuning • Load of current service from SRAM • Load of current channel from SRAM				280 150 110 0 –10 70 50 80 0	300 160 120 0 –20 100 70 100 10	µs	C, D
Timing Group: IDLEMode(RC) --> RXMode								
21.10	Last SPI clock --> TRPA first edge	Raw transparent output activated	23, 16	$T_{IDLERC2RX}$	640	1030	µs	C, D
	Additional time for: • Channel filter bandwidth 25kHz..45kHz • Channel filter bandwidth 50kHz..71kHz • Channel filter bandwidth 80kHz..146kHz • Channel filter bandwidth 165kHz..219kHz (default) • Channel filter bandwidth 237kHz..366kHz • RF calibration • VCO tuning				280 150 110 0 –10 70 50	300 160 120 0 –20 100 70	µs	C, D

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

6.2.3 RXMode -> IDLEMode

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Max.	Unit	Type*
Timing Group: RXMode --> IDLEMode								
22.00	EOT --> AVR power-down mode enable		1, 2, 13	$T_{RX2IDLE}$	110	180	µs	C, D
22.10	Last SPI clock --> AVR power-down mode enable	SPI command is used to switch to IDLEMode	23, 13	$T_{RX2IDLE}$	90	140	µs	C, D

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

6.2.4 RXMode -> RXMode

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Max.	Unit	Type*
Timing Group: RXMode --> RXMode								
26.00	Last SPI clock/EOT -> TRPA first edge	Raw transparent output activated	23, 16	T_{RX2RX}	380	530	µs	C, D
	Additional time for: • Channel filter bandwidth 25kHz..45kHz • Channel filter bandwidth 50kHz..71kHz • Channel filter bandwidth 80kHz..146kHz • Channel filter bandwidth 165kHz..219kHz (default) • Channel filter bandwidth 237kHz..366kHz • RF calibration • VCO tuning • Load of current service from SRAM (Rx1/Rx2 service not equal) • Load of current channel (Rx1/Rx2 channel not equal)				280 150 110 0 –10 70 50 80 0	300 160 120 0 –20 100 70 100 10	µs	C, D

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

6.2.5 IDLEMode -> PollingMode

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Max.	Unit	Type*
Timing Group: IDLEMode(XTO) --> PollingMode								
27.00	Last SPI clock --> TRPA first edge	Raw transparent output activated	23, 16	$T_{IDLEXTO2POLL}$	490	640	µs	C, D
	Additional time for: • Channel filter bandwidth 25kHz..45kHz • Channel filter bandwidth 50kHz..71kHz • Channel filter bandwidth 80kHz..146kHz • Channel filter bandwidth 165kHz..219kHz (default) • Channel filter bandwidth 237kHz..366kHz • RF calibration • VCO tuning				280 150 110 0 –10 70 50	300 160 120 0 –20 100 70	µs	C, D
Timing Group: IDLEMode(RC) --> PollingMode								
27.10	Last SPI clock --> TRPA first edge	Raw transparent output activated	23, 16	$T_{IDLERC2POLL}$	710	1080	µs	C, D
	Additional time for: • Channel filter bandwidth 25kHz..45kHz • Channel filter bandwidth 50kHz..71kHz • Channel filter bandwidth 80kHz..146kHz • Channel filter bandwidth 165kHz..219kHz (default) • Channel filter bandwidth 237kHz..366kHz • RF calibration • VCO tuning				280 150 110 0 –10 70 50	300 160 120 0 –20 100 70	µs	C, D

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

6.2.6 PollingMode -> IDLEMode

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Max.	Unit	Type*
Timing Group: PollingMode --> IDLEMode(XTO)								
28.00	EOT --> AVR power-down mode enable		1, 2, 13	$T_{POLL2IDLEXTO}$	140	190	µs	C, D
28.10	Last SPI clock --> AVR power-down mode enable	SPI command is used to switch to IDLEMode	23, 13	$T_{POLL2IDLEXTO2}$	90	120	µs	C, D
Timing Group: PollingMode --> IDLEMode(RC)								
28.20	EOT --> AVR power-down mode enable		1, 2, 13	$T_{POLL2IDLERC}$	150	210	µs	C, D
28.30	Last SPI clock --> AVR power-down mode enable	SPI command is used to switch to IDLEMode	23, 13	$T_{POLL2IDLERC2}$	100	140	µs	C, D

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

6.2.7 Tune and Check

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Max.	Unit	Type*
Timing Group: SRC Calibration								
29.20	Last SPI clock --> AVR power-down mode enable	IDLEMode(XTO)	23, 13	$T_{TCSRCCAL}$	160	200	µs	C, D
29.25	Last SPI clock --> AVR power-down mode enable	IDLEMode(RC)	23, 13	$T_{TCSRCCAL2}$	520	650	µs	C, D
Timing Group: FRC Calibration								
29.30	Last SPI clock --> AVR power-down mode enable	IDLEMode(XTO)	23, 13	$T_{TCFRCCAL}$	160	210	µs	C, D
29.35	Last SPI clock --> AVR power-down mode enable	IDLEMode(RC)	23, 13	$T_{TCFRCCAL2}$	520	650	µs	C, D
Timing Group: VCO Tuning								
29.40	Last SPI clock --> AVR power-down mode enable	IDLEMode(XTO)	23, 13	$T_{TCVCOTUNE}$	540	720	µs	C, D
29.45	Last SPI clock --> AVR power-down mode enable	IDLEMode(RC)	23, 13	$T_{TCVCOTUNE2}$	770	1140	µs	C, D

*) Type means: A = 100% tested at voltage and temperature limits, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

7. Appendix

7.1 Abbreviations

AC:	Alternating Current
ADC:	Analog-to-Digital Converter
AEC:	Automotive Electronics Council
AGC:	Analog Gain Control
ASK:	Amplitude Shift Keying
BER:	Bit Error Rate
BOD:	Brown Out Detector
BOM:	Bill of Material
BT:	Bandwidth Time Product
BW_{IF}:	Intermediate Frequency Bandwidth
CDM:	Charge Device Model (ESD)
CLK_OUT:	Clock Output
CNR:	Carrier-to-Noise Ratio
CP:	Charge Pump
CPU:	Central Processing Unit
CRC:	Cyclic Redundancy Check
DFIFO:	Data FIFO
DAC:	Digital-to-Analog Converter
DC:	Direct Current
DPSK:	Dual Phase Shift Keying
DR:	Data Rate
DSP:	Digital Signal Processing
EEPROM:	Electrical Erasable Programmable Read Only Memory
EOT:	End of Telegram
ESD:	Electro Static Discharge
FCDM:	Field Induce Charge Device Model (ESD)
FIFO:	First In First Out
FRC:	Fast Resistor Capacitor (oscillator)
FSK:	Frequency Shift Keying
GFSK:	Gaussians Frequency Shift Keying
GND:	Ground
HBM:	Human Body Model (ESD)
IC:	Integrated Circuit
ID:	Identification
IF:	Intermediate Frequency
IQ:	In-phase Quadrature
ISM:	Industrial Scientific and Medical
ISP:	In System Programming
JEDEC:	Joint Electron Device Engineering Council
LDO:	Low Dropout Voltage Regulator
LED:	Light Emitting Diode

LIN:	Local Interconnect Network
LNA:	Low Noise Amplifier
LO:	Local Oscillator
LSB:	Least Significant Bit
MCU:	Microcontroller
MISO:	Master In / Slave Out (SPI)
MM:	Machine Model (ESD)
MOSI:	Master Out / Slave In (SPI)
MSB:	Most Significant Bit
NRZ:	Non Return to Zero
NSS:	Not Slave Select (SPI)
OOK:	On Off Keying
PC:	Program counter
PCB:	Printed Circuit Board
PEG:	Passive Entry Go
PFD:	Phase Frequency Detector
PLL:	Phase Locked Loop
POR:	Power On Reset
ppm:	Parts Per Million
PSK:	Phase Shift Keying
RF:	Radio Frequency
RKE:	Remote Keyless Entry
ROM:	Read Only Memory
RS:	Remote Start
RSSI:	Received Signal Strength Indicator
RST:	Reset
Rx:	Receive
SFIFO:	Support FIFO
SAW:	Surface Acoustic Wave
SCK:	SPI Clock
SDM:	Sigma Delta Modulator
SOT:	Start of Telegram
SPDT:	Single Pole Double Throw
SPI:	Serial Programming Interface
SR:	Symbol Rate
SRAM:	Static Random Access Memory
SRC:	Slow Resistor Capacitor (oscillator)
SSI:	Synchronous Serial Interface
TC:	Transceiver Configuration
TMDI:	Transparent Mode Data Input
TMDO:	Transparent Mode Data Output
TMDO_CLK:	Transparent Mode Data Output Clock
TPM, TPMS:	Tire Pressure Monitoring System
Tx:	Transmit

µC:	Microcontroller
UHF:	Ultra High Frequency
VCO:	Voltage Controlled Oscillator
VS:	Voltage Supply
WDR:	Watchdog Reset
WCO:	Wake Check OK
WUP:	Wake-Up Pattern
XTAL:	Crystal
XTO:	Crystal Oscillator
∆f:	Frequency Deviation

7.2 Register Map

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x1FF)	Reserved	-	-	-	-	-	-	-	-
---	Reserved	-	-	-	-	-	-	-	-
(0x160)	Reserved	-	-	-	-	-	-	-	-
(0x15F)	RSCOM	-	-	-	-	-	-	RSIFC	0
(0x15E)	RSHDV	RSHDV[7:0]							
(0x15D)	RSLDV	RSLDV[7:0]							
(0x15C)	RSIFG	RSIFG[7:0]							
(0x15B)	Reserved	-	-	-	-	-	-	-	-
(0x15A)	Reserved	-	-	-	-	-	-	-	-
(0x159)	Reserved	-	-	-	-	-	-	-	-
(0x158)	SFIR	STIE	TIL[2:0]			SRIE	RIL[2:0]		
(0x157)	SFFR	TFC	TFL[2:0]			RFC	RFL[2:0]		
(0x156)	Reserved	-	-	-	-	-	-	-	-
(0x155)	DBENC	-	-	DBENC5	DBENC4	DBENC3	DBENC2	DBENC1	DBENC0
(0x154)	DBENB	DBENB7	DBENB6	DBENB5	DBENB4	DBENB3	DBENB2	DBENB1	DBENB0
(0x153)	DBTC	DBTC[7:0]							
(0x152)	DBCR	-	-	-	-	DBHA	DBTMS	DBCS	DBMD
(0x151)	RSSC	-	RSPKF	0	RSWLH	RSUP[3:0]			
(0x150)	RSSH	RSSH[7:0]							
(0x14F)	RSSL	RSSL[7:0]							
(0x14E)	RSSPK	RSSPK[7:0]							
(0x14D)	RSSAV	RSSAV[7:0]							
(0x14C)	IDS	-	-	-	-	-	-	IDFULL	IDOK
(0x14B)	IDC	IDCE	IDCLR	IDFIM	-	IDBO[1:0]		IDL[1:0]	
(0x14A)	IDB3	IDB3[7:0]							
(0x149)	IDB2	IDB2[7:0]							
(0x148)	IDB1	IDB1[7:0]							
(0x147)	IDB0	IDB0[7:0]							
(0x146)	CRCDOR	CRCDOR[7:0]							
(0x145)	CRCCR	-	-	-	-	-	REFLO	REFLI	CRCRS
(0x144)	RXDSA	RXDSA[7:0]							
(0x143)	RXCPHA	RXCPHA[7:0]							
(0x142)	RXCPLA	RXCPLA[7:1]							-
(0x141)	RXCIHA	RXCIHA[7:0]							
(0x140)	RXCILA	RXCILA[7:0]							
(0x13F)	RXCSBA	RXCSBA[7:0]							
(0x13E)	RXCRHA	RXCRHA[7:0]							
(0x13D)	RXCRLA	RXCRLA[7:0]							
(0x13C)	RXTLHA	-	-	-	-	RXTLHA[3:0]			
(0x13B)	RXTLLA	RXTLLA[7:0]							
(0x13A)	RXDSB	RXDSB[7:0]							
(0x139)	RXCPHB	RXCPHB[7:0]							
(0x138)	RXCPLB	RXCPLB[7:1]							-
(0x137)	RXCIHB	RXCIHB[7:0]							

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.2 Register Map (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x136)	RXCILB	RXCILB[7:0]							
(0x135)	RXCSBB	RXCSBB[7:0]							
(0x134)	RXCRHB	RXCRHB[7:0]							
(0x133)	RXCRLB	RXCRLB[7:0]							
(0x132)	RXTLHB	-	-	-	-	RXTLHB[3:0]			
(0x131)	RXTLLB	RXTLLB[7.0]							
(0x130)	RXBC2	-	-	-	-	-	RXBCLR	RXBF	RXBPB
(0x12F)	RXBC1	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA
(0x12E)	Reserved	-	-	-	-	-	-	-	-
---	Reserved	-	-	-	-	-	-	-	-
(0x10F)	Reserved	-	-	-	-	-	-	-	-
(0x10E)	FEBIA(1)	IFAEN	-	-	-	-	-	-	-
(0x10D)	Reserved	-	-	-	-	-	-	-	-
(0x10C)	Reserved	-	-	-	-	-	-	-	-
(0x10B)	FEVCO(1)	VCOB[3:0]				CPCC[3:0]			
(0x10A)	FECR(1)	-	-	ANPS	PLCKG	ADHS	0	S4N3	1
(0x109)	FETN4(1)	RTN4[3:0]				CTN4[3:0]			
(0x108)	FEMS(1)	PLLM[3:0]				PLLS[3:0]			
(0x107)	FEBT(1)	-	-	-	-	RTN2[1:0]		CTN2[1:0]	
(0x106)	FEVCT(1)	-	-	-	-	FEVCT[3:0]			
(0x105)	Reserved	-	-	-	-	-	-	-	-
(0x104)	Reserved	-	-	-	-	-	-	-	-
(0x103)	FELNA(1)	LBL[3:0]				-	-	-	-
(0x102)	FEEN2(1)	-	-	0	PLPEN	0	-	0	0
(0x101)	FEEN1(1)	-	PLSP1	ADCLK	ADEN	LNAEN	XTOEN	PLCAL	PLEN
(0x100)	FESR(1)	-	-	-	-	PLCK	XRDY	-	-
(0x0FF)	SSMFCR	-	-	-	-	-	-	SSMIDSF	SSMIDSO
(0x0FE)	SOTTOB	SOTTOB[7:0]							
(0x0FD)	SOTTOA	SOTTOA[7:0]							
(0x0FC)	WCOTOB	WCOTOB[7:0]							
(0x0FB)	WCOTOA	WCOTOA[7:0]							
(0x0FA)	EOTC3B	EOTAFE3	RRFEB3	TELREB3	TMOFEB3	MANFEB3	SYTFEB3	AMPFEB3	CARFEB3
(0x0F9)	EOTC2B	EOTAFE2	RRFEB2	TELREB2	TMOFEB2	MANFEB2	SYTFEB2	AMPFEB2	CARFEB2
(0x0F8)	EOTC1B	EOTAFE1	RRFEB1	TELREB1	TMOFEB1	MANFEB1	SYTFEB1	AMPFEB1	CARFEB1
(0x0F7)	EOTC3A	EOTBFE3	RRFEA3	TELREA3	TMOFEA3	MANFEA3	SYTFEA3	AMPFEA3	CARFEA3
(0x0F6)	EOTC2A	EOTBFE2	RRFEA2	TELREA2	TMOFEA2	MANFEA2	SYTFEA2	AMPFEA2	CARFEA2
(0x0F5)	EOTC1A	EOTBFE1	RRFEA1	TELREA1	TMOFEA1	MANFEA1	SYTFEA1	AMPFEA1	CARFEA1
(0x0F4)	SOTC2B	WCOAOE2	RROEB2	SFIDEB2	WUPEB2	MANOEB2	SYTOEB2	AMPOEB2	CAROEB2
(0x0F3)	SOTC1B	WCOAOE1	RROEB1	SFIDEB1	WUPEB1	MANOEB1	SYTOEB1	AMPOEB1	CAROEB1
(0x0F2)	SOTC2A	WCOBOE2	RROEA2	SFIDEA2	WUPEA2	MANOEA2	SYTOEA2	AMPOEA2	CAROEA2
(0x0F1)	SOTC1A	WCOBOE1	RROEA1	SFIDEA1	WUPEA1	MANOEA1	SYTOEA1	AMPOEA1	CAROEA1
(0x0F0)	GTCR	IWUPB	DARB	GAPMB	RXTEHB	IWUPA	DARA	GAPMA	RXTEHA
(0x0EF)	MSMCR4	MSMSM7[3:0]				MSMSM6[3:0]			
(0x0EE)	MSMCR3	MSMSM5[3:0]				MSMSM4[3:0]			

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.2 Register Map (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x0ED)	MSMCR2	MSMSM3[3:0]				MSMSM2[3:0]			
(0x0EC)	MSMCR1	MSMSM1[3:0]				MSMSM0[3:0]			
(0x0EB)	SSMXSR	-	-	SSMSTB[5:0]					
(0x0EA)	SSMSTR	-	-	SSMSTA[5:0]					
(0x0E9)	MSMSTR	-	-	-	SSMMST[4:0]				
(0x0E8)	SSMIMR	-	-	-	-	-	-	-	SSMIM
(0x0E7)	SSMIFR	-	-	-	-	-	-	-	SSMIF
(0x0E6)	SSMSR	SSMERR	-	-	-	SSMESM[3:0]			
(0x0E5)	SSMRR	-	-	-	-	-	-	SSMST	SSMR
(0x0E4)	SSMFBR	-	-	SSMPLDT	-	SSMDFDT	SSMFID[2:0]		
(0x0E3)	SSMRCR	SSMTMOE	SSMIDSE	SSMIFA	SSMPVS	1	1	SSMPB	SSMPA
(0x0E2)	SSMCR	SETRPB	SETRPA	-	SSMPVE	-	-	0	0
(0x0E1)	SFC	SFDRA	-	-	SFFLC[4:0]				
(0x0E0)	SFI	-	-	-	-	-	-	SFERIM	SFFLIM
(0x0DF)	SFD	SFD[7:0]							
(0x0DE)	SFRP	-	-	-	SFRP[4:0]				
(0x0DD)	SFWP	-	-	-	SFWP[4:0]				
(0x0DC)	SFL	SFCLR	-	-	SFFLS[4:0]				
(0x0DB)	SFS	-	-	-	-	-	SFOFL	SFUFL	SFFLRF
(0x0DA)	DFC	DFDRA	-	DFFLC[5:0]					
(0x0D9)	DFI	-	-	-	-	-	-	DFERIM	DFFLIM
(0x0D8)	DFD	DFD[7:0]							
(0x0D7)	DFRP	-	-	DFRP[5:0]					
(0x0D6)	DFWP	-	-	DFWP[5:0]					
(0x0D5)	DFL	DFCLR	-	DFFLS[5:0]					
(0x0D4)	DFTLH	-	-	-	-	DFTLH[3:0]			
(0x0D3)	DFTLL	DFTLL[7:0]							
(0x0D2)	DFS	-	-	-	-	-	DFOFL	DFUFL	DFFLRF
(0x0D1)	Reserved	-	-	-	-	-	-	-	-
(0x0D0)	CALRDY	CALRDY[7:0]							
(0x0CF)	SUPCA4	DCAL[3:0]				ACAL[3:0]			
(0x0CE)	SUPCA3	-	DCAL[6:4]			ACAL[7:4]			
(0x0CD)	SUPCA2	-	-	-	-	BGCAL[3:0]			
(0x0CC)	Reserved	-	-	-	-	-	-	-	-
(0x0CB)	SUPCR	-	AVDIC	AVEN	DVDIS	-	-	AVCCLM	AVCCRM
(0x0CA)	SUPFR	-	-	-	-	-	-	AVCCLF	AVCCRF
(0x0C9)	CMOCR	-	-	-	-	SRCACT	FRCACT	SRCAO	FRCAO
(0x0C8)	CMSR	-	-	-	-	-	-	-	ECF
(0x0C7)	FRCCAL	-	-	FRCTC	FRCCAL[4:0]				
(0x0C6)	SRCCAL	SRCTC[1:0]		SRCCAL[5:0]					
(0x0C5)	Reserved	-	-	-	-	-	-	-	-
(0x0C4)	CLKOCR	-	-	-	-	-	CLKOEN	CLKOS[1:0]	
(0x0C3)	CLKOD	CLKOD[7:0]							
(0x0C2)	WUP4A	WUP4A[7:0]							

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.2 Register Map (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x0C1)	WUP3A	WUP3A[7:0]							
(0x0C0)	WUP2A	WUP2A[7:0]							
(0x0BF)	WUP1A	WUP1A[7:0]							
(0x0BE)	SFID4A	SFID4A[7:0]							
(0x0BD)	SFID3A	SFID3A[7:0]							
(0x0BC)	SFID2A	SFID2A[7:0]							
(0x0BB)	SFID1A	SFID1A[7:0]							
(0x0BA)	WUPLA	-	-	WUPLA[5:0]					
(0x0B9)	WUPTA	-	-	-	WUPTA[4:0]				
(0x0B8)	SFIDLA	-	-	SFIDLA[5:0]					
(0x0B7)	SFIDCA	SEMEA	-	-	SFIDTA[4:0]				
(0x0B6)	WUP4B	WUP4B[7:0]							
(0x0B5)	WUP3B	WUP3B[7:0]							
(0x0B4)	WUP2B	WUP2B[7:0]							
(0x0B3)	WUP1B	WUP1B[7:0]							
(0x0B2)	SFID4B	SFID4B[7:0]							
(0x0B1)	SFID3B	SFID3B[7:0]							
(0x0B0)	SFID2B	SFID2B[7:0]							
(0x0AF)	SFID1B	SFID1B[7:0]							
(0x0AE)	WUPLB	-	-	WUPLB[5:0]					
(0x0AD)	WUPTB	-	-	-	WUPTB[4:0]				
(0x0AC)	SFIDLB	-	-	SFIDLB[5:0]					
(0x0AB)	SFIDCB	SEMEB	-	-	SFIDTB[4:0]				
(0x0AA)	CHDN	-	-	ADCDN	BBDN[4:0]				
(0x0A9)	CHCR	-	-	-	-	BWM[3:0]			
(0x0A8)	DMDRA	DMDNA[3:0]				DMAA[3:0]			
(0x0A7)	DMDRB	DMDNB[3:0]				DMAB[3:0]			
(0x0A6)	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
(0x0A5)	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				
(0x0A4)	DMCDA	DMCTA[2:0]			DMCLA[4:0]				
(0x0A3)	DMCDB	DMCTB[2:0]			DMCLB[4:0]				
(0x0A2)	DMMA	DMNEA	DMHA	DMPA	DMATA[4:0]				
(0x0A1)	DMMB	DMNEB	DMHB	DMPB	DMATB[4:0]				
(0x0A0)	RXFOA	RXFOA[7:0]							
(0x09F)	RXFOB	RXFOB[7:0]							
(0x09E)	SYCA	SYTLA[3:0]				SYCSA[3:0]			
(0x09D)	SYCB	SYTLB[3:0]				SYCSB[3:0]			
(0x09C)	Reserved	-	-	-	-	-	-	-	-
(0x09B)	Reserved	-	-	-	-	-	-	-	-
(0x09A)	Reserved	-	-	-	-	-	-	-	-
(0x099)	RDOCR	-	0	0	ETRPB	ETRPA	TMDS[1:0]		-
(0x098)	RDSIMR	WCOBM	WCOAM	SOTBM	SOTAM	EOTBM	EOTAM	NBITBM	NBITAM
(0x097)	Reserved	-	-	-	-	-	-	-	-
(0x096)	TESRA	-	-	-	-	-	EOTLA[1:0]		CRCOA

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.2 Register Map (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x095)	TESRB	-	-	-	-	-	EOTLB[1:0]		CRCOB
(0x094)	SOTCA	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
(0x093)	SOTCB	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB
(0x092)	SOTSA	WCOBO	RROA	SFIDOA	WUPOA	MANOA	SYTOA	AMPOA	CAROA
(0x091)	SOTSB	WCOAO	RROB	SFIDOB	WUPOB	MANOB	SYTOB	AMPOB	CAROB
(0x090)	GTCCR	TSM	-	-	-	-	-	-	PSR10
(0x08F)	T5IMR	-	-	-	-	-	-	T5CIM	T5OIM
(0x08E)	T5CNTH	T5CNTH[7:0]							
(0x08D)	T5CNTL	T5CNTL[7:0]							
(0x08C)	T5CCR	-	-	-	-	T5CTC	T5CS[2:0]		
(0x08B)	T5OCRH	T5OCRH[7:0]							
(0x08A)	T5OCRL	T5OCRL[7:0]							
(0x089)	Reserved	-	-	-	-	-	-	-	-
(0x088)	T4IMR	-	-	-	-	-	T4CPIM	T4CIM	T4OIM
(0x087)	T4MRB	T4ICS[2:0]			T4CE[1:0]		T4CNC	T4SCE	-
(0x086)	T4MRA	-	-	-	T4PS[2:0]			T4CS[1:0]	
(0x085)	T4ICRH	T4ICRH[7:0]							
(0x084)	T4ICRL	T4ICRL[7:0]							
(0x083)	T4CORH	T4CORH[7:0]							
(0x082)	T4CORL	T4CORL[7:0]							
(0x081)	T4CNTH	T4CNTH[7:0]							
(0x080)	T4CNTL	T4CNTL[7:0]							
(0x07F)	T3IMR	-	-	-	-	-	T3CPIM	T3CIM	T3OIM
(0x07E)	T3MRB	T3ICS[2:0]			T3CE[1:0]		T3CNC	T3SCE	-
(0x07D)	T3MRA	-	-	-	T3PS[2:0]			T3CS[1:0]	
(0x07C)	T3ICRH	T3ICRH[7:0]							
(0x07B)	T3ICRL	T3ICRL[7:0]							
(0x07A)	T3CORH	T3CORH[7:0]							
(0x079)	T3CORL	T3CORL[7:0]							
(0x078)	T3CNTH	T3CNTH[7:0]							
(0x077)	T3CNTL	T3CNTL[7:0]							
(0x076)	T2IMR	-	-	-	-	-	-	T2CIM	T2OIM
(0x075)	T2MR	T2DC[1:0]		T2PS[3:0]				T2CS[1:0]	
(0x074)	T2COR	T2COR[7:0]							
(0x073)	T2CNT	T2CNT[7:0]							
(0x072)	T1IMR	-	-	-	-	-	-	T1CIM	T1OIM
(0x071)	T1MR	T1DC[1:0]		T1PS[3:0]				T1CS[1:0]	
(0x070)	T1COR	T1COR[7:0]							
(0x06F)	T1CNT	T1CNT[7:0]							
(0x06E)	WDTCR	-	-	-	WDCE	WDE	WDPS[2:0]		
(0x06D)	PCMSK1	-	-	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
(0x06C)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
(0x06B)	EICRA	-	-	-	-	ISC11	ISC10	ISC01	ISC00
(0x06A)	Reserved	-	-	-	-	-	-	-	-

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.2 Register Map (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x069)	FFREQ2H	-	-	-	-	-	-	FFREQ2H[1:0]	
(0x068)	FFREQ2M	FFREQ2M[7:0]							
(0x067)	FFREQ2L	FFREQ2L[7:0]							
(0x066)	FFREQ1H	-	-	-	-	-	-	FFREQ1H[1:0]	
(0x065)	FFREQ1M	FFREQ1M[7:0]							
(0x064)	FFREQ1L	FFREQ1L[7:0]							
(0x063)	Reserved	-	-	-	-	-	-	-	-
(0x062)	Reserved	-	-	-	-	-	-	-	-
(0x061)	Reserved	-	-	-	-	-	-	-	-
(0x060)	FSEN	-	-	-	-	-	-	SDEN	SDPU
0x03F (0x05F)	SREG	I	T	H	S	V	N	Z	C
0x03E (0x05E)	SPH	-	-	-	-	-	SP10	SP9	SP8
0x03D (0x05D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
0x03C (0x05C)	Reserved	-	-	-	-	-	-	-	-
0x03B (0x05B)	CLPR	CLPCE	-	CLTPS[2:0]			CLKPS[2:0]		
0x03A (0x05A)	CMIMR	-	-	-	-	-	-	-	ECIE
0x039 (0x059)	CMCR	CMCCE	CMONEN	-	SRCD	CCS	CMM[2:0]		
0x038 (0x058)	SMCR	-	-	-	-	SM[2:0]			SE
0x037 (0x057)	EOTCB	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x036 (0x056)	EOTSB	EOTAF	RRFB	TELRB	TMOFB	MANFB	SYTFB	AMPFB	CARFB
0x035 (0x055)	EOTCA	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x034 (0x054)	EOTSA	EOTBF	RRFA	TELRA	TMOFA	MANFA	SYTFA	AMPFA	CARFA
0x033 (0x053)	RDCR	-	-	-	-	-	RDEN	ADIVEN	RDPU
0x032 (0x052)	Reserved	-	-	-	-	-	-	-	-
0x031 (0x051)	Reserved	-	-	-	-	-	-	-	-
0x030 (0x050)	Reserved	-	-	-	-	-	-	-	-
0x02F (0x04F)	T0IFR	-	-	-	-	-	-	-	T0F
0x02E (0x04E)	SPDR	SPDR[7:0]							
0x02D (0x04D)	SPSR	SPIF	-	TXIF	RXIF	-	-	-	SPI2X
0x02C (0x04C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR[1:0]	

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.2 Register Map (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02B (0x04B)	MCUSR	-	-	-	-	WDRF	-	EXTRF	PORF
0x02A (0x04A)	Reserved	-	-	-	-	-	-	-	-
0x029 (0x049)	CRCDIR	CRCDIR[7:0]							
0x028 (0x048)	EIFR	-	-	-	-	-	-	INTF1	INTF0
0x027 (0x047)	EIMSK	-	-	-	-	-	-	INT1	INT0
0x026 (0x046)	PCICR	-	-	-	-	-	-	PCIE1	PCIE0
0x025 (0x045)	GPIOR2	GPIOR2[7:0]							
0x024 (0x044)	GPIOR1	GPIOR1[7:0]							
0x023 (0x043)	Reserved	-	-	-	-	-	-	-	-
0x022 (0x042)	Reserved	-	-	-	-	-	-	-	-
0x021 (0x041)	Reserved	-	-	-	-	-	-	-	-
0x020 (0x040)	Reserved	-	-	-	-	-	-	-	-
0x01F (0x03F)	Reserved	-	-	-	-	-	-	-	-
0x01E (0x03E)	GPIOR6	GPIOR6[7:0]							
0x01D (0x03D)	GPIOR5	GPIOR5[7:0]							
0x01C (0x03C)	GPIOR4	GPIOR4[7:0]							
0x01B (0x03B)	GPIOR3	GPIOR3[7:0]							
0x01A (0x03A)	GPIOR0	GPIOR0[7:0]							
0x019 (0x039)	T5IFR	-	-	-	-	-	-	T5COF	T5OFF
0x018 (0x038)	T4IFR	-	-	-	-	-	T4ICF	T4COF	T4OFF
0x017 (0x037)	T3IFR	-	-	-	-	-	T3ICF	T3COF	T3OFF
0x016 (0x036)	T2IFR	-	-	-	-	-	-	T2COF	T2OFF
0x015 (0x035)	T1IFR	-	-	-	-	-	-	T1COF	T1OFF
0x014 (0x034)	T4CR	T4ENA	T4TOS	T4RES	T4TOP	T4CPRM	T4CRM	T4CTM	T4OTM
0x013 (0x033)	T3CR	T3ENA	T3TOS	T3RES	T3TOP	T3CPRM	T3CRM	T3CTM	T3OTM

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.2 Register Map (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x012 (0x032)	T2CR	T2ENA	T2TOS	T2RES	T2TOP	-	T2CRM	T2CTM	T2OTM
0x011 (0x031)	T1CR	T1ENA	T1TOS	T1RES	T1TOP	-	T1CRM	T1CTM	T1OTM
0x010 (0x030)	T0CR	-	-	-	T0PR	T0IE	T0PS[2:0]		
0x00F (0x02F)	PCIFR	-	-	-	-	-	-	PCIF1	PCIF0
0x00E (0x02E)	MCUCR	PB7HS	PB7LS	PB4HS	PUD	ENPS	SPIIO	IVSEL	IVCE
0x00D (0x02D)	RDSIFR	WCOB	WCOA	SOTB	SOTA	EOTB	EOTA	NBITB	NBITA
0x00C (0x02C)	Reserved	-	-	-	-	-	-	-	-
0x00B (0x02B)	Reserved	-	-	-	-	-	-	-	-
0x00A (0x02A)	PORTC	-	-	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
0x009 (0x029)	DDRC	-	-	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
0x008 (0x028)	PINC	-	-	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
0x007 (0x027)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x006 (0x026)	DDRB	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
0x005 (0x025)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x004 (0x024)	RDPR	RDPRF	ARDPRF	APRPTA	APRPTB	1	PRFLT	PRPTA	PRPTB
0x003 (0x023)	PRR2	PRSSM	-	PRRS	PRIDS	PRDF	PRSF	PRXA	PRXB
0x002 (0x022)	PRR1	-	-	-	PRT5	PRT4	PRT3	PRT2	PRT1
0x001 (0x021)	PRR0	-	-	PRCO	1	PRCRC	1	PRRXDC	PRSPI
0x000 (0x020)	Reserved	-	-	-	-	-	-	-	-

Notes: 1. Only accessible if AVCC is on, see Section 3.6 “RF Front-End Register Description” on page 171

7.3 SRAM Map

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0200	internal										
...											
0x020C	internal										
0x020D	events	pinEvent Mask	0x00	PWRON	-	NPWRON6	NPWRON5	NPWRON4	NPWRON3	NPWRON2	NPWRON1
0x020E	events	sysEvent Mask	0xA2	SYS_ERR	CMD_RDY	SYS_RDY	AVCC LOWM	-	RX_ACTIVE_EN	RX_ACTIVE_POL	IRQ_POL
0x020F	events	cmdRdy Conf	0x00	-	-	SRC_CAL	FRC_CAL	VCO_CAL	RF_CAL	SELF CHECK	-
0x0210	internal										
...											
0x0214	internal										
0x0215	current Service	CHCR	0x00	-	-	-	-	BWM[3:0]			
0x0216	current Service	CHDN	0x00	-	-	ADCDN	BBDN[4:0]				
0x0217	current Service	CHSTART FILTER	0x00	CHSTART FILTER[7:0]							
0x0218	current Service	DMCDA	0x00	DMCTA[2:0]			DMCLA[4:0]				
0x0219	current Service	DMCDB	0x00	DMCTB[2:0]			DMCLB[4:0]				
0x021A	current Service	DMCRA	0x00	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x021B	current Service	DMCRB	0x00	DMARB	SY1TB	SASKB	DMPGB[4:0]				
0x021C	current Service	DMDRA	0x00	DMDNA[3:0]				DMAA[3:0]			
0x021D	current Service	DMDRB	0x00	DMDNB[3:0]				DMAB[3:0]			
0x021E	current Service	DMMA	0x00	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x021F	current Service	DMMB	0x00	DMNEB	DMHB	DMPB	DMATB[4:0]				
0x0220	current Service	EOT1A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x0221	current Service	EOT1B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x0222	current Service	EOT2A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x0223	current Service	EOT2B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x0224	current Service	EOT3A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x0225	current Service	EOT3B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x0226	internal										
0x0227	internal										

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0228	internal										
0x0229	current Service	FEVCO	0x00	VCOB[3:0]				CPCC[3:0]			
0x022A	current Service	FEVCT	0x00	-	-	-	-	FEVCT[3:0]			
0x022B	internal										
...											
0x0234	internal										
0x0235	current Service	IF[0] (low byte)	0x00	IF[7:0]							
0x0236	current Service	IF[1] (high byte)	0x00	IF[15:8]							
0x0237	current Service	RDOCR	0x00	-	0	0	ETRPB	ETRPA	TMDS[1:0]		-
0x0238	current Service	rssiSys Conf	0x00	RssiEnable	-	rssiBufEv Mask	RSSIbuf [4:0]				
0x0239	current Service	rxSet PathA[0]	0x00	-	rxBufEv MaskA	RXbufA[5:0]					
0x023A	current Service	rxSet PathA[1]	0x00	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x023B	current Service	rxSet PathB[0]	0x00	-	rxBufEv MaskB	RXbufB[5:0]					
0x023C	current Service	rxSet PathB[1]	0x00	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB
0x023D	current Service	rxSys Event	0x00	IDCHKA_ Mask	WCOKA_ Mask	SOTA_ Mask	EOTA_Mask	IDCHKB_ Mask	WCOKB_ Mask	SOTB_ Mask	EOTB_ Mask
0x023E	current Service	rxSysSet	0x00	0	IdScan_ ENA	IFAmplifier_ ENA	PathValidAft erSOT_ENA	1	1	0	0
0x023F	current Service	SFIDA[0]	0x00	SFID1A[7:0]							
0x0240	current Service	SFIDA[1]	0x00	SFID2A[7:0]							
0x0241	current Service	SFIDA[2]	0x00	SFID3A[7:0]							
0x0242	current Service	SFIDA[3]	0x00	SFID4A[7:0]							
0x0243	current Service	SFIDB[0]	0x00	SFID1B[7:0]							
0x0244	current Service	SFIDB[1]	0x00	SFID2B[7:0]							
0x0245	current Service	SFIDB[2]	0x00	SFID3B[7:0]							
0x0246	current Service	SFIDB[3]	0x00	SFID4B[7:0]							
0x0247	current Service	SFIDCA	0x00	SEMEA	-	-	SFIDTA[4:0]				
0x0248	current Service	SFIDCB	0x00	SEMEB	-	-	SFIDTB[4:0]				

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0249	current Service	SFIDLA	0x00	-	-	SFIDLA[5:0]					
0x024A	current Service	SFIDLB	0x00	-	-	SFIDLB[5:0]					
0x024B	current Service	SOT1A	0x00	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x024C	current Service	SOT1B	0x00	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB
0x024D	current Service	SOT2A	0x00	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x024E	current Service	SOT2B	0x00	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB
0x024F	current Service	SOTtime OutA	0x00	SOTTOA[7:0]							
0x0250	current Service	SOTtime OutB	0x00	SOTTOB[7:0]							
0x0251	current Service	SYCA	0x00	SYTLA[3:0]				SYCSA[3:0]			
0x0252	current Service	SYCB	0x00	SYTLB[3:0]				SYCSB[3:0]			
0x0253	internal										
0x0254	internal										
0x0255	current Service	TMUL	0x00	TMUL[7:0]							
0x0256	current Service	trxSys Conf	0x00	-	-	-	-	TRPB_ENA	TRPA_ENA	0	0
0x0257	internal										
...											
0x0262	internal										
0x0263	current Service	WCOtime OutA	0x00	WCOTOA[7:0]							
0x0264	current Service	WCOtime OutB	0x00	WCOTOB[7:0]							
0x0265	current Service	WUPA[0]	0x00	WUP1A[7:0]							
0x0266	current Service	WUPA[1]	0x00	WUP2A[7:0]							
0x0267	current Service	WUPA[2]	0x00	WUP3A[7:0]							
0x0268	current Service	WUPA[3]	0x00	WUP4A[7:0]							
0x0269	current Service	WUPB[0]	0x00	WUP1B[7:0]							
0x026A	current Service	WUPB[1]	0x00	WUP2B[7:0]							
0x026B	current Service	WUPB[2]	0x00	WUP3B[7:0]							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x026C	current Service	WUPB[3]	0x00	WUP4B[7:0]							
0x026D	current Service	WUPLA	0x00	-	-	WUPLA[5:0]					
0x026E	current Service	WUPLB	0x00	-	-	WUPLB[5:0]					
0x026F	current Service	WUPTA	0x00	-	-	-	WUPTA[4:0]				
0x0270	current Service	WUPTB	0x00	-	-	-	WUPTB[4:0]				
0x0271	current Service	RXCPA[0] (low byte)	0x00	RXCPLA[7:0]							
0x0272	current Service	RXCPA[1] (high byte)	0x00	RXCPHA[7:0]							
0x0273	current Service	RXCIA[0] (low byte)	0x00	RXCILA[7:0]							
0x0274	current Service	RXCIA[1] (high byte)	0x00	RXCIHA[7:0]							
0x0275	current Service	RXCSBA	0x00	RXCSBA[7:0]							
0x0276	current Service	RXTLA[0] (low byte)	0x00	RXTLLA[7:0]							
0x0277	current Service	RXTLA[1] (high byte)	0x00	-	-	-	-	RXTLHA[3:0]			
0x0278	current Service	RXCPB[0] (low byte)	0x00	RXCPLB[7:0]							
0x0279	current Service	RXCPB[1] (high byte)	0x00	RXCPHB[7:0]							
0x027A	current Service	RXCIB[0] (low byte)	0x00	RXCILB[7:0]							
0x027B	current Service	RXCIB[1] (high byte)	0x00	RXCIHB[7:0]							
0x027C	current Service	RXCSBB	0x00	RXCSBB[7:0]							
0x027D	current Service	RXTLB[0] (low byte)	0x00	RXTLLB[7:0]							
0x027E	current Service	RXTLB[1] (high byte)	0x00	-	-	-	-	RXTLHB[3:0]			
0x027F	current Service	RXBC1	0x00	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA
0x0280	internal										
...											
0x0291	internal										
0x0292	current Service	RSSC	0x00	-	RSPKF	RSHRX	RSWLH	RSUP[3:0]			
0x0293	current Service	FFREQ[0] (low byte)	0x00	FFREQL[7:0]							
0x0294	current Service	FFREQ[1] (mid byte)	0x00	FFREQM[7:0]							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0295	current Service	FFREQ[2] (high byte)	0x00	FFREQH[7:0]							
0x0296	current Service	FEMS	0x00	PLLM[3:0]				PLLS[3:0]			
0x0297	current Service	FECR	0x00	-	-	ANPS	PLCKG	ADHS	-	S4N3	1
0x0298	internal										
...											
0x02C2	internal										
0x02C3	calib	tempMeas	0x00	8 bit temperature measurement result (Range 0:175)							
0x02C4	calib	srcRes	0x00	SRC frequency measurement result							
0x02C5	calib	srcCorVal	0x00	correction value for Timer1							
0x02C6	internal										
0x02C7	temperatureCalibration	temperatureCalibration[0]	0x00	temperature deviation value byte 0							
0x02C8	temperatureCalibration	temperatureCalibration[1]	0x00	temperature deviation value byte 1							
0x02C9	temperatureCalibration	temperatureCalibration[2]	0x00	temperature deviation value byte 2							
0x02CA	temperatureCalibration	temperatureCalibration[3]	0x00	temperature deviation value byte 3							
0x02CB	temperatureCalibration	temperatureCalibration[4]	0x00	temperature deviation value byte 4							
0x02CC	temperatureCalibration	temperatureCalibration[5]	0x00	temperature deviation value byte 5							
0x02CD	temperatureCalibration	temperatureCalibration[6]	0x00	temperature deviation value byte 6							
0x02CE	temperatureCalibration	temperatureCalibration[7]	0x00	temperature deviation value byte 7							
0x02CF	temperatureCalibration	temperatureCalibration[8]	0x00	temperature deviation value byte 8							
0x02D0	temperatureCalibration	temperatureCalibration[9]	0x00	temperature deviation value byte 9							
0x02D1	temperatureCalibration	temperatureCalibration[10]	0x00	temperature deviation value byte 10							
0x02D2	temperatureCalibration	temperatureCalibration[11]	0x00	temperature deviation value byte 11							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x02D3	temperatureCalibration	temperature Calibration[12]	0x00	temperature deviation value byte 12							
0x02D4	temperatureCalibration	temperature Calibration[13]	0x00	temperature deviation value byte 13							
0x02D5	temperatureCalibration	temperature Calibration[14]	0x00	temperature deviation value byte 14							
0x02D6	temperatureCalibration	temperature Calibration[15]	0x00	temperature deviation value byte 15							
0x02D7	temperatureCalibration	temperature Calibration[16]	0x00	temperature deviation value byte 16							
0x02D8	temperatureCalibration	temperature Calibration[17]	0x00	temperature deviation value byte 17							
0x02D9	temperatureCalibration	temperature Calibration[18]	0x00	temperature deviation value byte 18							
0x02DA	temperatureCalibration	temperature Calibration[19]	0x00	temperature deviation value byte 19							
0x02DB	temperatureCalibration	temperature Calibration[20]	0x00	temperature deviation value byte 20							
0x02DC	temperatureCalibration	temperature Calibration[21]	0x00	temperature deviation value byte 21							
0x02DD	temperatureCalibration	temperature Calibration[22]	0x00	temperature deviation value byte 22							
0x02DE	trxCalibConfiguration	calConf1	0x00	0	0	EN_SRCCAL	EN_FRCCAL	EN_VCOCAL	EN_RFCAL	EN_SELFCHECK	EN_REGREFRESH
0x02DF	trxCalibConfiguration	calConf2	0x85	SRC[7:0]							
0x02E0	trxCalibConfiguration	selfChk	0x00	SC[7:0]							
0x02E1	frcCalibrationConfiguration	frcCalib Gate	0xBF	data[7:0]							
0x02E2	internal										
0x02E3	internal										
0x02E4	internal										
0x02E5	trxConf	system Config	0x00	-	-	SRCAO	FRCAO	SFIFO_OFL_UFL_RX_DISABLE	DFIFO_OFL_UFL_RX_DISABLE	Current idle mode selector	-

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x02E6	internal										
...											
0x02EA	internal										
0x02EB	sleepModeConfig	sleepModeConfig	0x80	FirmwareSleepModeEnable	-	-	-	sleep Mode[2:0]			-
0x02EC	internal										
...											
0x02F3	internal										
0x02F4	debug	errorCode	0x00	contains the error information of last system error							
0x02F5	debug	ssmErrorCode	0x00	contains the detailed error information of SSM error							
0x02F6	poll Config	confT1COR	0x00	T1COR[7:0]							
0x02F7	poll Config	confT1MR	0x00	T1MR[7:0]							
0x02F8	poll Config	pollChanConf[0].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x02F9	poll Config	pollChanConf[0].svcChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x02FA	poll Config	pollChanConf[1].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x02FB	poll Config	pollChanConf[1].svcChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x02FC	poll Config	pollChanConf[2].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x02FD	poll Config	pollChanConf[2].svcChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x02FE	poll Config	pollChanConf[3].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x02FF	poll Config	pollChanConf[3].svcChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0300	poll Config	pollChanConf[4].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x0301	poll Config	pollChanConf[4].svcChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0302	poll Config	pollChanConf[5].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0303	poll Config	pollChanConf [5].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0304	poll Config	pollChanConf[6].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x0305	poll Config	pollChanConf [6].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0306	poll Config	pollChanConf [7].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x0307	poll Config	pollChanConf [7].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0308	poll Config	pollChanConf[8].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x0309	poll Config	pollChanConf [8].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x030A	poll Config	pollChanConf [9].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x030B	poll Config	pollChanConf [9].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x030C	poll Config	pollChanConf [10].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x030D	poll Config	pollChanConf [10].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x030E	poll Config	pollChanConf [11].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x030F	poll Config	pollChanConf[11].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0310	poll Config	pollChanConf [12].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x0311	poll Config	pollChanConf [12].svcCh Config	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0312	poll Config	pollChanConf [13].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x0313	poll Config	pollChanConf [13].svc ChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0314	poll Config	pollChanConf [14].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP
0x0315	poll Config	pollChanConf [14].svc ChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0316	poll Config	pollChanConf [15].config	0x00	RfCalib	-	VCO_TUNE	-	-	-	EOL	EOP

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0317	poll Config	pollChanConf [15].svc ChConfig	0x00	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
0x0318	internal										
...											
0x031B	internal										
0x031C	sramServices[3]	CHCR	0x00	-	-	-	-	BWM[3:0]			
0x031D	sramServices[3]	CHDN	0x00	-	-	ADCDN	BBDN[4:0]				
0x031E	sramServices[3]	CHSTARTFILTER	0x00	CHSTARTFILTER[7:0]							
0x031F	sramServices[3]	DMCDA	0x00	DMCTA[2:0]			DMCLA[4:0]				
0x0320	sramServices[3]	DMCDB	0x00	DMCTB[2:0]			DMCLB[4:0]				
0x0321	sramServices[3]	DMCRA	0x00	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	sramServices[3]	DMCRB	0x00	DMARB	SY1TB	SASKB	DMPGB[4:0]				
0x0323	sramServices[3]	DMDRA	0x00	DMDNA[3:0]				DMAA[3:0]			
0x0324	sramServices[3]	DMDRB	0x00	DMDNB[3:0]				DMAB[3:0]			
0x0325	sramServices[3]	DMMA	0x00	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x0326	sramServices[3]	DMMB	0x00	DMNEB	DMHB	DMPB	DMATB[4:0]				
0x0327	sramServices[3]	EOT1A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x0328	sramServices[3]	EOT1B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x0329	sramServices[3]	EOT2A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x032A	sramServices[3]	EOT2B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x032B	sramServices[3]	EOT3A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x032C	sramServices[3]	EOT3B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x032D	internal										
0x032E	internal										
0x032F	internal										
0x0330	sramServices[3]	FEVCO	0x00	VCOB[3:0]				CPCC[3:0]			
0x0331	sramServices[3]	FEVCT	0x00	-	-	-	-	FEVCT[3:0]			
0x0332	internal										

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
...											
0x033B	internal										
0x033C	sramServices[3]	IF[0] (low byte)	0x00	IF[7:0]							
0x033D	sramServices[3]	IF[1] (high byte)	0x00	IF[15:8]							
0x033E	sramServices[3]	RDOCR	0x00	-	0	0	ETRPB	ETRPA	TMDS[1:0]		-
0x033F	sramServices[3]	rssiSysConf	0x00	RssiEnable	-	rssiBufEvMask	RSSIbuf[4:0]				
0x0340	sramServices[3]	rxSetPathA[0]	0x00	-	rxBufEvMaskA	RXbufA[5:0]					
0x0341	sramServices[3]	rxSetPathA[1]	0x00	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x0342	sramServices[3]	rxSetPathB[0]	0x00	-	rxBufEvMaskB	RXbufB[5:0]					
0x0343	sramServices[3]	rxSetPathB[1]	0x00	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB
0x0344	sramServices[3]	rxSysEvent	0x00	IDCHKA_Mask	WCOKA_Mask	SOTA_Mask	EOTA_Mask	IDCHKB_Mask	WCOKB_Mask	SOTB_Mask	EOTB_Mask
0x0345	sramServices[3]	rxSysSet	0x00	0	IdScan_ENA	IFAmplifier_ENA	PathValidAfterSOT_ENA	1	1	0	0
0x0346	sramServices[3]	SFIDA[0]	0x00	SFID1A[7:0]							
0x0347	sramServices[3]	SFIDA[1]	0x00	SFID2A[7:0]							
0x0348	sramServices[3]	SFIDA[2]	0x00	SFID3A[7:0]							
0x0349	sramServices[3]	SFIDA[3]	0x00	SFID4A[7:0]							
0x034A	sramServices[3]	SFIDB[0]	0x00	SFID1B[7:0]							
0x034B	sramServices[3]	SFIDB[1]	0x00	SFID2B[7:0]							
0x034C	sramServices[3]	SFIDB[2]	0x00	SFID3B[7:0]							
0x034D	sramServices[3]	SFIDB[3]	0x00	SFID4B[7:0]							
0x034E	sramServices[3]	SFIDCA	0x00	SEMEA	-	-	SFIDTA[4:0]				
0x034F	sramServices[3]	SFIDCB	0x00	SEMEB	-	-	SFIDTB[4:0]				
0x0350	sramServices[3]	SFIDLA	0x00	-	-	SFIDLA[5:0]					
0x0351	sramServices[3]	SFIDLB	0x00	-	-	SFIDLB[5:0]					
0x0352	sramServices[3]	SOT1A	0x00	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0353	sramServices[3]	SOT1B	0x00	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB
0x0354	sramServices[3]	SOT2A	0x00	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x0355	sramServices[3]	SOT2B	0x00	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB
0x0356	sramServices[3]	SOTtimeOutA	0x00	SOTTOA[7:0]							
0x0357	sramServices[3]	SOTtimeOutB	0x00	SOTTOB[7:0]							
0x0358	sramServices[3]	SYCA	0x00	SYTLA[3:0]				SYCSA[3:0]			
0x0359	sramServices[3]	SYCB	0x00	SYTLB[3:0]				SYCSB[3:0]			
0x035A	internal										
0x035B	internal										
0x035C	sramServices[3]	TMUL	0x00	TMUL[7:0]							
0x035D	sramServices[3]	trxSysConf	0x00	-	-	-	-	TRPB_ENA	TRPA_ENA	0	0
0x035E	internal										
...											
0x0369	internal										
0x036A	sramServices[3]	WCOtimeOutA	0x00	WCOTOA[7:0]							
0x036B	sramServices[3]	WCOtimeOutB	0x00	WCOTOB[7:0]							
0x036C	sramServices[3]	WUPA[0]	0x00	WUP1A[7:0]							
0x036D	sramServices[3]	WUPA[1]	0x00	WUP2A[7:0]							
0x036E	sramServices[3]	WUPA[2]	0x00	WUP3A[7:0]							
0x036F	sramServices[3]	WUPA[3]	0x00	WUP4A[7:0]							
0x0370	sramServices[3]	WUPB[0]	0x00	WUP1B[7:0]							
0x0371	sramServices[3]	WUPB[1]	0x00	WUP2B[7:0]							
0x0372	sramServices[3]	WUPB[2]	0x00	WUP3B[7:0]							
0x0373	sramServices[3]	WUPB[3]	0x00	WUP4B[7:0]							
0x0374	sramServices[3]	WUPLA	0x00	-	-	WUPLA[5:0]					
0x0375	sramServices[3]	WUPLB	0x00	-	-	WUPLB[5:0]					

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0376	sramServices[3]	WUPTA	0x00	-	-	-	WUPTA[4:0]				
0x0377	sramServices[3]	WUPTB	0x00	-	-	-	WUPTB[4:0]				
0x0378	sramServices[3]	RXCPA[0] (low byte)	0x00	RXCPLA[7:0]							
0x0379	sramServices[3]	RXCPA[1] (high byte)	0x00	RXCPHA[7:0]							
0x037A	sramServices[3]	RXCIA[0] (low byte)	0x00	RXCILA[7:0]							
0x037B	sramServices[3]	RXCIA[1] (high byte)	0x00	RXCIHA[7:0]							
0x037C	sramServices[3]	RXCSBA	0x00	RXCSBA[7:0]							
0x037D	sramServices[3]	RXTLA[0] (low byte)	0x00	RXTLLA[7:0]							
0x037E	sramServices[3]	RXTLA[1] (high byte)	0x00	-	-	-	-	RXTLHA[3:0]			
0x037F	sramServices[3]	RXCPB[0] (low byte)	0x00	RXCPLB[7:0]							
0x0380	sramServices[3]	RXCPB[1] (high byte)	0x00	RXCPHB[7:0]							
0x0381	sramServices[3]	RXCIB[0] (low byte)	0x00	RXCILB[7:0]							
0x0382	sramServices[3]	RXCIB[1] (high byte)	0x00	RXCIHB[7:0]							
0x0383	sramServices[3]	RXCSBB	0x00	RXCSBB[7:0]							
0x0384	sramServices[3]	RXTLB[0] (low byte)	0x00	RXTLLB[7:0]							
0x0385	sramServices[3]	RXTLB[1] (high byte)	0x00	-	-	-	-	RXTLHB[3:0]			
0x0386	sramServices[3]	RXBC1	0x00	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA
0x0387	internal										
...											
0x0398	internal										
0x0399	sramServices[3]	RSSC	0x00	-	RSPKF	RSHRX	RSWLH	RSUP[3:0]			
0x039A	sramServices[3]	FFREQ[0] (low byte)	0x00	FFREQL[7:0]							
0x039B	sramServices[3]	FFREQ[1] (mid byte)	0x00	FFREQM[7:0]							
0x039C	sramServices[3]	FFREQ[2] (high byte)	0x00	FFREQH[7:0]							
0x039D	sramServices[3]	FEMS	0x00	PLLM[3:0]				PLLS[3:0]			
0x039E	sramServices[3]	FECR	0x00	-	-	ANPS	PLCKG	ADHS	0	S4N3	1

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x039F	sramServices[3]	FFREQ[0] (low byte)	0x00	FFREQL[7:0]							
0x03A0	sramServices[3]	FFREQ[1] (mid byte)	0x00	FFREQM[7:0]							
0x03A1	sramServices[3]	FFREQ[2] (high byte)	0x00	FFREQH[7:0]							
0x03A2	sramServices[3]	FEMS	0x00	PLLM[3:0]				PLLS[3:0]			
0x03A3	sramServices[3]	FECR	0x00	-	-	ANPS	PLCKG	ADHS	0	S4N3	1
0x03A4	sramServices[3]	FFREQ[0] (low byte)	0x00	FFREQL[7:0]							
0x03A5	sramServices[3]	FFREQ[1] (mid byte)	0x00	FFREQM[7:0]							
0x03A6	sramServices[3]	FFREQ[2] (high byte)	0x00	FFREQH[7:0]							
0x03A7	sramServices[3]	FEMS	0x00	PLLM[3:0]				PLLS[3:0]			
0x03A8	sramServices[3]	FECR	0x00	-	-	ANPS	PLCKG	ADHS	0	S4N3	1
0x03A9	sramServices[4]	CHCR	0x00	-	-	-	-	BWM[3:0]			
0x03AA	sramServices[4]	CHDN	0x00	-	-	ADCDN	BBDN[4:0]				
0x03AB	sramServices[4]	CHSTART FILTER	0x00	CHSTART FILTER[7:0]							
0x03AC	sramServices[4]	DMCDA	0x00	DMCTA[2:0]			DMCLA[4:0]				
0x03AD	sramServices[4]	DMCDB	0x00	DMCTB[2:0]			DMCLB[4:0]				
0x03AE	sramServices[4]	DMCRA	0x00	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x03AF	sramServices[4]	DMCRB	0x00	DMARB	SY1TB	SASKB	DMPGB[4:0]				
0x03B0	sramServices[4]	DMDRA	0x00	DMDNA[3:0]				DMAA[3:0]			
0x03B1	sramServices[4]	DMDRB	0x00	DMDNB[3:0]				DMAB[3:0]			
0x03B2	sramServices[4]	DMMA	0x00	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x03B3	sramServices[4]	DMMB	0x00	DMNEB	DMHB	DMPB	DMATB[4:0]				
0x03B4	sramServices[4]	EOT1A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x03B5	sramServices[4]	EOT1B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x03B6	sramServices[4]	EOT2A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x03B7	sramServices[4]	EOT2B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x03B8	sramServices[4]	EOT3A	0x00	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x03B9	sramServices[4]	EOT3B	0x00	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB
0x03BA	internal										
0x03BB	internal										
0x03BC	internal										
0x03BD	sramServices[4]	FEVCO	0x00	VCOB[3:0]				CPCC[3:0]			
0x03BE	sramServices[4]	FEVCT	0x00	-	-	-	-	FEVCT[3:0]			
0x03BF	internal										
...											
0x03C8	internal										
0x03C9	sramServices[4]	IF[0] (low byte)	0x00	IF[7:0]							
0x03CA	sramServices[4]	IF[1] (high byte)	0x00	IF[15:8]							
0x03CB	sramServices[4]	RDOCR	0x00	-	0	0	ETRPB	ETRPA	TMDS[1:0]		-
0x03CC	sramServices[4]	rssiSysConf	0x00	RssiEnable	-	rssiBufEvMask	RSSIbuf[4:0]				
0x03CD	sramServices[4]	rxSetPathA[0]	0x00	-	rxBufEvMaskA	RXbufA[5:0]					
0x03CE	sramServices[4]	rxSetPathA[1]	0x00	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x03CF	sramServices[4]	rxSetPathB[0]	0x00	-	rxBufEvMaskB	RXbufB[5:0]					
0x03D0	sramServices[4]	rxSetPathB[1]	0x00	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB
0x03D1	sramServices[4]	rxSysEvent	0x00	IDCHKA_Mask	WCOKA_Mask	SOTA_Mask	EOTA_Mask	IDCHKB_Mask	WCOKB_Mask	SOTB_Mask	EOTB_Mask
0x03D2	sramServices[4]	rxSysSet	0x00	0	IdScan_ENA	IFAmplifier_ENA	PathValidAfterSOT_ENA	1	1	0	0
0x03D3	sramServices[4]	SFIDA[0]	0x00	SFID1A[7:0]							
0x03D4	sramServices[4]	SFIDA[1]	0x00	SFID2A[7:0]							
0x03D5	sramServices[4]	SFIDA[2]	0x00	SFID3A[7:0]							
0x03D6	sramServices[4]	SFIDA[3]	0x00	SFID4A[7:0]							
0x03D7	sramServices[4]	SFIDB[0]	0x00	SFID1B[7:0]							
0x03D8	sramServices[4]	SFIDB[1]	0x00	SFID2B[7:0]							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x03D9	sramServices[4]	SFIDB[2]	0x00	SFID3B[7:0]							
0x03DA	sramServices[4]	SFIDB[3]	0x00	SFID4B[7:0]							
0x03DB	sramServices[4]	SFIDCA	0x00	SEMEA	-	-	SFIDTA[4:0]				
0x03DC	sramServices[4]	SFIDCB	0x00	SEMEB	-	-	SFIDTB[4:0]				
0x03DD	sramServices[4]	SFIDLA	0x00	-	-	SFIDLA[5:0]					
0x03DE	sramServices[4]	SFIDLB	0x00	-	-	SFIDLB[5:0]					
0x03DF	sramServices[4]	SOT1A	0x00	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x03E0	sramServices[4]	SOT1B	0x00	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB
0x03E1	sramServices[4]	SOT2A	0x00	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x03E2	sramServices[4]	SOT2B	0x00	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB
0x03E3	sramServices[4]	SOTtimeOutA	0x00	SOTTOA[7:0]							
0x03E4	sramServices[4]	SOTtimeOutB	0x00	SOTTOB[7:0]							
0x03E5	sramServices[4]	SYCA	0x00	SYTLA[3:0]				SYCSA[3:0]			
0x03E6	sramServices[4]	SYCB	0x00	SYTLB[3:0]				SYCSB[3:0]			
0x03E7	internal										
0x03E8	internal										
0x03E9	sramServices[4]	TMUL	0x00	TMUL[7:0]							
0x03EA	sramServices[4]	trxSysConf	0x00	-	-	-	-	TRPB_ENA	TRPA_ENA	0	0
0x03EB	internal										
...											
0x03F6	internal										
0x03F7	sramServices[4]	WCOtimeOutA	0x00	WCOTOA[7:0]							
0x03F8	sramServices[4]	WCOtimeOutB	0x00	WCOTOB[7:0]							
0x03F9	sramServices[4]	WUPA[0]	0x00	WUP1A[7:0]							
0x03FA	sramServices[4]	WUPA[1]	0x00	WUP2A[7:0]							
0x03FB	sramServices[4]	WUPA[2]	0x00	WUP3A[7:0]							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x03FC	sramServices[4]	WUPA[3]	0x00	WUP4A[7:0]							
0x03FD	sramServices[4]	WUPB[0]	0x00	WUP1B[7:0]							
0x03FE	sramServices[4]	WUPB[1]	0x00	WUP2B[7:0]							
0x03FF	sramServices[4]	WUPB[2]	0x00	WUP3B[7:0]							
0x0400	sramServices[4]	WUPB[3]	0x00	WUP4B[7:0]							
0x0401	sramServices[4]	WUPLA	0x00	-	-	WUPLA[5:0]					
0x0402	sramServices[4]	WUPLB	0x00	-	-	WUPLB[5:0]					
0x0403	sramServices[4]	WUPTA	0x00	-	-	-	WUPTA[4:0]				
0x0404	sramServices[4]	WUPTB	0x00	-	-	-	WUPTB[4:0]				
0x0405	sramServices[4]	RXCPA[0] (low byte)	0x00	RXCPLA[7:0]							
0x0406	sramServices[4]	RXCPA[1] (high byte)	0x00	RXCPHA[7:0]							
0x0407	sramServices[4]	RXCIA[0] (low byte)	0x00	RXCILA[7:0]							
0x0408	sramServices[4]	RXCIA[1] (high byte)	0x00	RXCIHA[7:0]							
0x0409	sramServices[4]	RXCSBA	0x00	RXCSBA[7:0]							
0x040A	sramServices[4]	RXTLA[0] (low byte)	0x00	RXTLLA[7:0]							
0x040B	sramServices[4]	RXTLA[1] (high byte)	0x00	-	-	-	-	RXTLHA[3:0]			
0x040C	sramServices[4]	RXCPB[0] (low byte)	0x00	RXCPLB[7:0]							
0x040D	sramServices[4]	RXCPB[1] (high byte)	0x00	RXCPHB[7:0]							
0x040E	sramServices[4]	RXCIB[0] (low byte)	0x00	RXCILB[7:0]							
0x040F	sramServices[4]	RXCIB[1] (high byte)	0x00	RXCIHB[7:0]							
0x0410	sramServices[4]	RXCSBB	0x00	RXCSBB[7:0]							
0x0411	sramServices[4]	RXTLB[0] (low byte)	0x00	RXTLLB[7:0]							
0x0412	sramServices[4]	RXTLB[1] (high byte)	0x00	-	-	-	-	RXTLHB[3:0]			
0x0413	sramServices[4]	RXBC1	0x00	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA
0x0414	internal										

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
...											
0x0425	internal										
0x0426	sramServices[4]	RSSC	0x00	-	RSPKF	RSHRX	RSWLH	RSUP[3:0]			
0x0427	sramServices[4]	FFREQ[0] (low byte)	0x00	FFREQL[7:0]							
0x0428	sramServices[4]	FFREQ[1] (mid byte)	0x00	FFREQM[7:0]							
0x0429	sramServices[4]	FFREQ[2] (high byte)	0x00	FFREQH[7:0]							
0x042A	sramServices[4]	FEMS	0x00	PLLM[3:0]				PLLS[3:0]			
0x042B	sramServices[4]	FECR	0x00	-	-	ANPS	PLCKG	ADHS	0	S4N3	1
0x042C	sramServices[4]	FFREQ[0] (low byte)	0x00	FFREQL[7:0]							
0x042D	sramServices[4]	FFREQ[1] (mid byte)	0x00	FFREQM[7:0]							
0x042E	sramServices[4]	FFREQ[2] (high byte)	0x00	FFREQH[7:0]							
0x042F	sramServices[4]	FEMS	0x00	PLLM[3:0]				PLLS[3:0]			
0x0430	sramServices[4]	FECR	0x00	-	-	ANPS	PLCKG	ADHS	0	S4N3	1
0x0431	sramServices[4]	FFREQ[0] (low byte)	0x00	FFREQL[7:0]							
0x0432	sramServices[4]	FFREQ[1] (mid byte)	0x00	FFREQM[7:0]							
0x0433	sramServices[4]	FFREQ[2] (high byte)	0x00	FFREQH[7:0]							
0x0434	sramServices[4]	FEMS	0x00	PLLM[3:0]				PLLS[3:0]			
0x0435	sramServices[4]	FECR	0x00	-	-	ANPS	PLCKG	ADHS	0	S4N3	1
0x0436	internal										
...											
0x0447	internal										
0x0448	rssiThresholds	rssiThresholds[3][0].RSSL	0x00	RSSL for service 3 and channel 0							
0x0449	rssiThresholds	rssiThresholds[3][0].RSSH	0xFF	RSSH for service 3 and channel 0							
0x044A	rssiThresholds	rssiThresholds[3][1].RSSL	0x00	RSSL for service 3 and channel 1							
0x044B	rssiThresholds	rssiThresholds[3][1].RSSH	0xFF	RSSH for service 3 and channel 1							
0x044C	rssiThresholds	rssiThresholds[3][2].RSSL	0x00	RSSL for service 3 and channel 2							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x044D	rssiThresholds	rssiThresholds[3][2].RSSH	0xFF	RSSH for service 3 and channel 2							
0x044E	rssiThresholds	rssiThresholds[4][0].RSSL	0x00	RSSL for service 4 and channel 0							
0x044F	rssiThresholds	rssiThresholds[4][0].RSSH	0xFF	RSSH for service 4 and channel 0							
0x0450	rssiThresholds	rssiThresholds[4][1].RSSL	0x00	RSSL for service 4 and channel 1							
0x0451	rssiThresholds	rssiThresholds[4][1].RSSH	0xFF	RSSH for service 4 and channel 1							
0x0452	rssiThresholds	rssiThresholds[4][2].RSSL	0x00	RSSL for service 4 and channel 2							
0x0453	rssiThresholds	rssiThresholds[4][2].RSSH	0xFF	RSSH for service 4 and channel 2							
0x0454	internal										
...											
0x046B	internal										
0x046C	sramID	idEna	0x00	-	-	-	ID[4:0]				
0x046D	sramID	data[0][0]	0x00	data0_byte0[7:0]							
0x046E	sramID	data[0][1]	0x00	data0_byte1[7:0]							
0x046F	sramID	data[0][2]	0x00	data0_byte2[7:0]							
0x0470	sramID	data[0][3]	0x00	data0_byte3[7:0]							
0x0471	sramID	config[0]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x0472	sramID	data[1][0]	0x00	data1_byte0[7:0]							
0x0473	sramID	data[1][1]	0x00	data1_byte1[7:0]							
0x0474	sramID	data[1][2]	0x00	data1_byte2[7:0]							
0x0475	sramID	data[1][3]	0x00	data1_byte3[7:0]							
0x0476	sramID	config[1]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x0477	sramID	data[2][0]	0x00	data2_byte0[7:0]							
0x0478	sramID	data[2][1]	0x00	data2_byte1[7:0]							
0x0479	sramID	data[2][2]	0x00	data2_byte2[7:0]							
0x047A	sramID	data[2][3]	0x00	data2_byte3[7:0]							
0x047B	sramID	config[2]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x047C	sramID	data[3][0]	0x00	data3_byte0[7:0]							
0x047D	sramID	data[3][1]	0x00	data3_byte1[7:0]							
0x047E	sramID	data[3][2]	0x00	data3_byte2[7:0]							
0x047F	sramID	data[3][3]	0x00	data3_byte3[7:0]							
0x0480	sramID	config[3]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x0481	sramID	data[4][0]	0x00	data4_byte0[7:0]							
0x0482	sramID	data[4][1]	0x00	data4_byte1[7:0]							
0x0483	sramID	data[4][2]	0x00	data4_byte2[7:0]							
0x0484	sramID	data[4][3]	0x00	data4_byte3[7:0]							
0x0485	sramID	config[4]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x0486	sramID	data[5][0]	0x00	data5_byte0[7:0]							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0487	sramID	data[5][1]	0x00	data5_byte1[7:0]							
0x0488	sramID	data[5][2]	0x00	data5_byte2[7:0]							
0x0489	sramID	data[5][3]	0x00	data5_byte3[7:0]							
0x048A	sramID	config[5]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x048B	sramID	data[6][0]	0x00	data6_byte0[7:0]							
0x048C	sramID	data[6][1]	0x00	data6_byte1[7:0]							
0x048D	sramID	data[6][2]	0x00	data6_byte2[7:0]							
0x048E	sramID	data[6][3]	0x00	data6_byte3[7:0]							
0x048F	sramID	config[6]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x0490	sramID	data[7][0]	0x00	data7_byte0[7:0]							
0x0491	sramID	data[7][1]	0x00	data7_byte1[7:0]							
0x0492	sramID	data[7][2]	0x00	data7_byte2[7:0]							
0x0493	sramID	data[7][3]	0x00	data7_byte3[7:0]							
0x0494	sramID	config[7]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x0495	sramID	data[8][0]	0x00	data8_byte0[7:0]							
0x0496	sramID	data[8][1]	0x00	data8_byte1[7:0]							
0x0497	sramID	data[8][2]	0x00	data8_byte2[7:0]							
0x0498	sramID	data[8][3]	0x00	data8_byte3[7:0]							
0x0499	sramID	config[8]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x049A	sramID	data[9][0]	0x00	data9_byte0[7:0]							
0x049B	sramID	data[9][1]	0x00	data9_byte1[7:0]							
0x049C	sramID	data[9][2]	0x00	data9_byte2[7:0]							
0x049D	sramID	data[9][3]	0x00	data9_byte3[7:0]							
0x049E	sramID	config[9]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x049F	sramID	data[10][0]	0x00	data10_byte0[7:0]							
0x04A0	sramID	data[10][1]	0x00	data10_byte1[7:0]							
0x04A1	sramID	data[10][2]	0x00	data10_byte2[7:0]							
0x04A2	sramID	data[10][3]	0x00	data10_byte3[7:0]							
0x04A3	sramID	config[10]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04A4	sramID	data[11][0]	0x00	data11_byte0[7:0]							
0x04A5	sramID	data[11][1]	0x00	data11_byte1[7:0]							
0x04A6	sramID	data[11][2]	0x00	data11_byte2[7:0]							
0x04A7	sramID	data[11][3]	0x00	data11_byte3[7:0]							
0x04A8	sramID	config[11]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04A9	sramID	data[12][0]	0x00	data12_byte0[7:0]							
0x04AA	sramID	data[12][1]	0x00	data12_byte1[7:0]							
0x04AB	sramID	data[12][2]	0x00	data12_byte2[7:0]							
0x04AC	sramID	data[12][3]	0x00	data12_byte3[7:0]							
0x04AD	sramID	config[12]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04AE	sramID	data[13][0]	0x00	data13_byte0[7:0]							
0x04AF	sramID	data[13][1]	0x00	data13_byte1[7:0]							
0x04B0	sramID	data[13][2]	0x00	data13_byte2[7:0]							

7.3 SRAM Map (Continued)

The variables located in the SRAM, contain all necessary information to control the receiver in IDLEMode, RXMode and PollingMode.

				Description							
Address	Structure	Variable	Init Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x04B1	sramID	data[13][3]	0x00	data13_byte3[7:0]							
0x04B2	sramID	config[13]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04B3	sramID	data[14][0]	0x00	data14_byte0[7:0]							
0x04B4	sramID	data[14][1]	0x00	data14_byte1[7:0]							
0x04B5	sramID	data[14][2]	0x00	data14_byte2[7:0]							
0x04B6	sramID	data[14][3]	0x00	data14_byte3[7:0]							
0x04B7	sramID	config[14]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04B8	sramID	data[15][0]	0x00	data15_byte0[7:0]							
0x04B9	sramID	data[15][1]	0x00	data15_byte1[7:0]							
0x04BA	sramID	data[15][2]	0x00	data15_byte2[7:0]							
0x04BB	sramID	data[15][3]	0x00	data15_byte3[7:0]							
0x04BC	sramID	config[15]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04BD	sramID	data[16][0]	0x00	data16_byte0[7:0]							
0x04BE	sramID	data[16][1]	0x00	data16_byte1[7:0]							
0x04BF	sramID	data[16][2]	0x00	data16_byte2[7:0]							
0x04C0	sramID	data[16][3]	0x00	data16_byte3[7:0]							
0x04C1	sramID	config[16]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04C2	sramID	data[17][0]	0x00	data17_byte0[7:0]							
0x04C3	sramID	data[17][1]	0x00	data17_byte1[7:0]							
0x04C4	sramID	data[17][2]	0x00	data17_byte2[7:0]							
0x04C5	sramID	data[17][3]	0x00	data17_byte3[7:0]							
0x04C6	sramID	config[17]	0x00	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
0x04C7	internal										
...											
0x04F8	internal										
0x04F9	sramcrc	result	0x00	Calculated SRAM CRC value							
0x04FA	sramcrc	status	0x00								CRCstatus
0x04FB	internal										
...											
0x0518	internal										
0x0519	unused										
...											
0x05FF	unused										

7.4 SPI Command Reference

7.4.1 Read Fill Level Rx FIFO

Description

The SPI command “Read Fill Level Rx FIFO” returns the fill level of the Rx data FIFO.

Command Frame

Master	0x01	0x00	0x00
ATA5785	events.system	events.events	bytes

Parameters

- 0x01 Command ID
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- bytes Fill level of Rx DFIFO

7.4.2 Read Fill Level RSSI FIFO

Description

The SPI command “Read Fill Level RSSI FIFO” returns the fill level of the support FIFO in Rx direction.

Command Frame

Master	0x03	0x00	0x00
ATA5785	events.system	events.events	bytes

Parameters

- 0x03 Command ID
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- bytes Fill level of SFIFO

7.4.3 Get Event Bytes

Description

The SPI command “Get Event Bytes” returns the actual events.system, events.events, events.power and events.config bytes to the master. After the SPI command “Get Event Bytes”, the event bytes events.system, event.events and events.power are cleared. The event byte events.config is not cleared as it contains the configuration of the currently loaded service/channel configuration.

Command Frame

Master	0x04	0x00	0x00	0x00
ATA5785	events.system	events.events	events.power	events.config

Parameters

- 0x04 Command ID
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte for system events
- events.events Event byte for event events
- events.power Event byte for power events
- events.config Event byte for config events

Register/Variable

Event Bytes								
system (R15)	SYS_ERR	CMD_RDY	SYS_RDY	AVCCLOW	-	SFIFO	DFIFO_RX	-
events (R14)	IDCHKA	WCOKA	SOTA	EOTA	IDCHKB	WCOKB	SOTB	EOTB
power (R13)	PWRON	-	NPWRON6	NPWRON5	NPWRON4	NPWRON3	NPWRON2	NPWRON1
config (R12)	PathB	PathA	ch[1:0]		-	ser[2:0]		

7.4.4 Read RSSI FIFO

Description

The SPI command “Read RSSI FIFO” reads n data bytes from the SFIFO. The desired number of data bytes to be returned is given in the length parameter. The access direction switch of the FIFO is set automatically and restored to the origin at the end of the command (direction temporarily stored in the T-bit).

Command Frame

Master	0x05	length	0x00	0x00	…	0x00
ATA5785	events.system	events.events	dummy	data	…	data

Parameters

- 0x05 Command ID
- length Number of bytes to be read from the SFIFO
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)
- …
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- 0x00 Dummy byte (no information)
- data Data byte 1
- data Data byte 2
- …
- data Data byte n (n = length)

7.4.5 Read Rx FIFO

Description

The SPI command “Read Rx FIFO” reads n data bytes from the DFIFO. The desired number of data bytes to be returned is given in the length parameter. The access direction switch of the FIFO is set automatically and restored to the origin at the end of the command (direction temporarily stored in the T-bit).

Command Frame

Master	0x06	length	0x00	0x00	…	0x00
ATA5785	events.system	events.events	dummy	data	…	data

Parameters

- 0x06 Command ID
- length Number of bytes to be read from DFIFO
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)
- …
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- 0x00 Dummy byte (no information)
- data Data byte 1
- data Data byte 2
- …
- data Data byte n (n = length)

7.4.6 Write SRAM / Register

Description

The SPI command “Write SRAM / Register” writes n consecutive data bytes to the SRAM or hardware registers.

Command Frame

Master	0x07	length	addr_high	addr_low	data	…	data
ATA5785	events.system	events.events	dummy	dummy	0x00	…	0x00

Parameters

- 0x07 Command ID
- length Number of bytes to be written to SRAM or registers
- addr_high High byte of the destination start address
- addr_low Low byte of the destination start address
- data Data byte 1
- data Data byte 2
- …
- data Data byte n (n = length)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- dummy Dummy byte (no information)
- dummy Dummy byte (no information)
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)
- …
- 0x00 Dummy byte (no information)

7.4.7 Read SRAM / Register

Description

The SPI command “Read SRAM / Register” reads n consecutive data bytes from the SRAM or hardware registers, and sends them back to the master.

Command Frame

Master	0x08	length	addr_high	addr_low	0x00	0x00	…	0x00
ATA5785	events.system	events.events	dummy	dummy	dummy	data	…	data

Parameters

- 0x08 Command ID
- length Number of bytes to be read from SRAM or registers
- addr_high High byte of the destination start address
- addr_low Low byte of the destination start address
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)
- …
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- dummy Dummy byte (no information)
- dummy Dummy byte (no information)
- dummy Dummy byte (no information)
- data Data byte
- …
- data Data byte n (n = length)

7.4.8 Set System Mode

Description

The SPI command “Set System Mode” is used to switch the following modes:

1. IDLEMode
   a. IDLEMode(RC)
   b. IDLEMode(XTO)
2. RXMode
3. PollingMode

Note: 1. The IDLEMode is the only mode that allows reading/writing from/to the SRAM and registers.

The parameters are copied as follows:

extReq.systemModeConfig = systemModeConfig
extReq.serviceChannelConfig = serviceChannelConfig

Command Frame

Master	0x0D	systemModeConfig	serviceChannelConfig
ATA5785	events.system	events.events	dummy

Parameters

- 0x0D Command ID
- systemModeConfig Parameter for extReq.systemModeConfig variable

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
systemModeConfig	RF_CAL	-	VCO_TUNE	IDLEMode Selector	-	TMDEN	OPM[1:0]	

Bit 7: RF_CAL – RF Calibration
0 RF Calibration before changing OPM disabled
1 RF Calibration before changing OPM enabled

Bit 6: Reserved Bit
This bit is reserved for future use and shall be set to zero

Bit 5: VCO_TUNE – VCO Tuning
0 VCO tuning before changing OPM disabled
1 VCO tuning before changing OPM enabled

Bit 4: IDLEModeSelector – IDLEMode Selector
0 IDLEMode(RC) selected
1 IDLEMode(XTO) selected

Bit 3: Reserved Bit
This bit is reserved for future use and shall be set to zero

Bit 2: TMDEN – Transparent Mode Data Enable
0 Buffered mode is enabled
1 Transparent mode is enabled

Bit 1..0: OPM[1:0] – Operation Mode

OPM1	OPM0	Function
0	0	IDLEMode
0	1	Reserved
1	0	RXMode
1	1	RX PollingMode

- serviceChannelConfig Parameter for extReq.serviceChannelConfig variable (all modes except PollingMode)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
serviceChannelConfig	enaPathB	enaPathA	ch[1:0]		-	ser[2:0]		

Bit 7: enaPathB – Enable Path B
0 Path B disabled
1 Path B enabled

Bit 6: enaPathA – Enable Path A
0 Path A disabled
1 Path A enabled

Bits 5..4: ch[1:0] – Channel

ch1	ch0	Function
0	0	Channel 0
0	1	Channel 1
1	0	Channel 2
1	1	Channel 0 - default

Bit 3: Reserved Bit
This bit is reserved for future use and shall be set to zero.

Bits 2..0: ser[2:0] – Service[2:0]

Ser2	Ser1	Ser0	Function
0	0	0	Reserved
0	0	1	Reserved
0	1	0	Reserved
0	1	1	Service3
1	0	0	Service4
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

- serviceChannelConfig Parameter for extReq.serviceChannelConfig variable (PollingMode only)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
serviceChannelConfig	-	-	-	-	startPollingIndex			

Bits 7..4: Reserved Bits
These bits are reserved for future use and shall be set to zero.

Bits 3..0: startPollingIndex – Start Index in PollingMode

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- 0x00 Dummy byte (no information)

Set System Mode (IDLEMode)

If Operation Mode OPM IDLEMode is selected, the firmware switches from any mode to IDLEMode. This command is accepted always to set the system to a defined mode.

The following parameters are **used**:

- systemModeConfig
  - IDLEModeSelector:
    - 0 switch to IDLEMode(RC)
    - 1 switch to IDLEMode(XTO)
  - OPM Operation Mode (set to 0b00)
- OPM serviceChannelConfig
  - none

The following parameters are **ignored**:

- systemModeConfig
  - RF_CAL
  - VCO_TUNE
  - TMDEN
- serviceChannelConfig
  - enaPathB
  - enaPathA
  - ch[1:0]
  - ser[2:0]

Set System Mode (RXMode)

If the operation mode OPM RXMode is selected, the firmware switches from IDLEMode or RXMode to RXMode. The firmware decides if normal switching or direct switching is used.

The following parameters are **used**:

- systemModeConfig
  - RF_CAL: if set, the firmware does a correction of the frequency words according to the available temperature value.
  - VCO_TUNE: if set, the firmware configures the Rx flow with VCO tuning SSM.
  - IDLEModeSelector: if set, the system returns to IDLEMode(XTO) after RXMode has finished. If cleared, the system returns to IDLEMode(RC).
  - TMDEN Transparent Data Mode Enable: if set, the system is configured for RXMode(transparent). If cleared, the system is configured for RXMode(buffered).
  - OPM Operation Mode (set to 0b10).
- serviceChannelConfig
  - enaPathB: If set, path B configuration is used for RXMode.
  - enaPathA: If set, path A configuration is used for RXMode.
  - ch[1:0]: defines the used channel configuration for RXMode.
  - ser[2:0]: defines the used service configuration for RXMode.

The following parameters are **ignored**:

- none

Set System Mode (PollingMode)

If the operation mode OPM PollingMode is selected, the firmware switches from IDLEMode to PollingMode. After the SPI command is accepted the firmware configures the SSM with VCO tuning (if enabled in systemModeConfig) for the 1st service/channel configuration. The following service/channel configurations are configured according to their configuration in the SRAM array pollConfig.pollChanConf.

The following parameters are **used**:

- systemModeConfig
  - VCO_TUNE: if set, the firmware configures the 1st service/channel configuration with VCO tuning SSM.
  - IDLEModeSelector: if set, the system remains in IDLEMode(XTO) during the sleep period. If cleared, the system remains in IDLEMode(RC) during the sleep period.
  - TMDEN Transparent Data Mode Enable: If set, the system is configured for PollingMode(transparent).
  - OPM Operation Mode (set to 0b11).
- serviceChannelConfig
  - startPollingIndex[3:0] is used as array index to start PollingMode

The following parameters are **ignored**:

- systemModeConfig
  - RF_CAL: RF calibration is not available in PollingMode

7.4.9 Calibrate and Check

Description

The SPI command “Calibrate and Check” is used to trigger one of the following features.

- SRC calibration
- FRC calibration
- VCO tuning
- Self check

The parameters are copied as follows:

extReq.tuneCheckConfig = tuneCheckConfig
extReq.serviceChannelConfig = serviceChannelConfig

Command Frame

Master	0x0E	tuneCheckConfig	serviceChannelConfig
ATA5785	events.system	events.events	dummy

Parameters

- 0x0E Command ID
- tuneCheckConfig Parameter for extReq.tuneCheckConfig

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
tuneCheckConfig	-	-	EN_ SRCCAL	EN_ FRCCAL	EN_ VCOCAL	-	EN_SELF CHECK	-
Note: Only one bit is allowed to be set in the tuneCheckConfig.								

Bit 7: Reserved Bit
This bit is reserved for future use and shall be set to zero.

Bit 6: Reserved Bit
This bit is reserved for future use and shall be set to zero.

Bit 5: **EN_SRCCAL – Enable SRC Calibration**
0 SRC calibration disabled
1 SRC calibration enabled

Bit 4: **EN_FRCCAL – Enable FRC Calibration**
0 FRC calibration disabled
1 FRC calibration enabled

Bit 3: **EN_VCOCAL – Enable VCO Tuning**
0 VCO tuning disabled
1 VCO tuning enabled

Bit 2: Reserved Bit
This bit is reserved for future use and shall be set to zero.

Bit 1: **EN_SELFCHECK – Enable Self Check**
0 Self check disabled
1 Self check enabled

Bit 0: Reserved Bit
This bit is reserved for future use and shall be set to zero.

- serviceChannelConfig Parameter for extReq.serviceChannelConfig

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
serviceChannelConfig	enaPathB	enaPathA	ch[1:0]		-	ser[2:0]		

Bit 7: enaPathB – Enable Path B
0 Path B disabled
1 Path B enabled

Bit 6: enaPathA – Enable Path A
0 Path A disabled
1 Path A enabled

Bits 5..4: Ch[1:0] – Channel[1:0]

Bit 3: Reserved Bit
This bit is reserved for future use and shall be set to zero.

Bits 2..0: Ser[2:0] – Service[2:0]

Return Values
- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- 0x00 Dummy byte (no information)

7.4.10 Set Watchdog

Description
The SPI command “Set Watchdog” is used to configure, enable and disable the internal watchdog timer.

Command Frame

Master	0x10	confWDTCR
ATA5785	events.system	events.events

Parameters
- 0x10 Command ID
- confWDTCR Watchdog timer configuration

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
confWDTCR	-	-	-	-	WDE	WDPS[2:0]		

Bits 7..4: Reserved Bit**s**
These bits are reserved for future use and shall be set to zero.

Bit 3: WDE – Watchdog Enable
0 Watchdog disabled
1 Watchdog enabled

Bits 2..0: WDPS[2:0] – Watchdog Prescaler Select

WDPS[2:0]			Time-out at VCC = 3V/25°C and T_{SRC} ≈ 1/125kHz	
			Typical	Minimum
0	0	0	1ms	0.85ms
0	0	1	4ms	3.4ms
0	1	0	32ms	27ms
0	1	1	2.1s	1.75s
1	0	0	4.2s	3.5s
1	0	1	16.8s	14s
1	1	0	134s	110s
1	1	1	268s	220s

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)

7.4.11 Get Version ROM

Description

The SPI command “Get Version ROM” returns the ROM version to the master.

Command Frame

Master	0x12	0x00	0x00
ATA5785	events.system	events.events	version

Parameters

- 0x12 Command ID
- 0x00 Dummy bytes (no information)
- 0x00 Dummy bytes (no information)

Return Values

- events.system Event bytes (for details see SPI command “Get Event Bytes”)
- events.events Event bytes (for details see SPI command “Get Event Bytes”)
- version ROM version

7.4.12 System Reset

Description

The SPI command “System Reset” configures the watchdog with the minimum timeout period and waits for a watchdog reset in an endless loop.

Command Frame

Master	0x15	0x00
ATA5785	events.system	events.events

Parameters

- 0x15 Command ID
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)

7.4.13 OFF Command

Description

This SPI command “OFF Command” sets the Atmel® ATA5785 to OFFMode. Switching to OFFMode is realized as follows:

```
DDRB = 0x00;
DDRC = 0x00;
PORTB = 0xFF;
PORTC = 0xFF;
SUPCR |= BM_DVDIS;
```

Command Frame

Master	0x18	0x00
ATA5785	events.system	events.events

Parameters

- 0x18 Command ID
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)

7.4.14 Start RSSI Measurement

Description

The SPI command “Start RSSI Measurement” starts the RSSI measurement of the service/channel configuration parameter of the transferred serviceChannelConfig. The result is stored in the uint8_t rfRssiValue and can be read out with the SPI command “Get RSSI Value”.

Note: The RSSI measurement must be activated and configured in the rssiSysConf SRAM variable of the corresponding service. Otherwise incorrect values will be determined.

Command Frame

Master	0x1B	serviceChannelConfig
ATA5785	events.system	events.events

Parameters

- 0x1B Command ID
- serviceChannelConfig Service ChannelConfig of the extReq structure

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
serviceChannelConfig	enaPathB	enaPathA	ch[1:0]		-	ser[2:0]		

Bit 7: enaPathB – Enable Path B

0 Path B disabled

1 Path B enabled

Bit 6: enaPathA – Enable Path A

0 Path A disabled

1 Path A enabled

Bits 5..4: ch[1:0] – Channel[1:0]

Bit 3: Reserved Bit

This bit is reserved for future use and shall be set to zero.

Bits 2..0: ser[2:0] – Service[2:0]

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)

7.4.15 Get RSSI Value

Description

The SPI command “Get RSSI Value” returns the current RSSI average and peak value. To trigger a RSSI measurement the SPI command “Start RSSI Measurement” can be used. The signal power of the watched 50Ω antenna input can be calculated by formula (1) on page 47.

Command Frame

Master	0x1C	0x00	0x00	0x00
ATA5785	events.system	events.events	RSSAV	RSSPK

Parameters

- 0x1C Command ID
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)
- 0x00 Dummy byte (no information)

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- RSSAV Last measured RSSI average value
- RSSPK Last measured RSSI peak value

7.4.16 Start SRAM CRC Calculation

Description

The SPI command “Start SRAM CRC Calculation” triggers a CRC calculation of “length” bytes in the SRAM starting from a configurable address. The final CRC value can be read from the SRAM variable sramcrc.result after the status bit sramcrc.status is “0” to indicate that calculation has finished.

For calculation of the CRC value the following fixed polynomial is used: $x^8 + x^2 + x + 1$

Command Frame

Master	0x1D	length	addr_high	addr_low
ATA5785	events.system	events.events	dummy	dummy

Parameters

- 0x1D Command ID
- length Number of bytes used for CRC calculation
- addr_high High byte of start address for CRC calculation
- addr_low Low byte of start address for CRC calculation

Return Values

- events.system Event byte (for details see SPI command “Get Event Bytes”)
- events.events Event byte (for details see SPI command “Get Event Bytes”)
- dummy Dummy byte (no information)
- dummy Dummy byte (no information)

7.5 Error Codes

In case of an internal error the SYS_ERR event flag in the event byte events.system is set and an error code is stored to the SRAM variable debug:

Table 7-1. Error Code Variables

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F4	errorCode	contains the error information of the last system error							
0x02F5	ssmErrorCode	contains the last internal state of the hardware SSM							

Table 7-2. Error Codes Description

errorCode	Error Description
0	Reserved
1	DEBUG_ERROR_CODE_SETIDLEMODE_MISS_SSMRDY This error code is set if the ShutDown SSM doesn't signal SSM_RDY.
2	DEBUG_ERROR_CODE_SETIDLEMODE_TIMER_LOCKED This error code is set if the SSM watchdog timer (Timer2) is not available for ShutDown.
3	DEBUG_ERROR_CODE_RX_STATESTARTSSM_TIMER_LOCKED This error code is set if the SSM watchdog timer (Timer2) is not available for RX.
4	DEBUG_ERROR_CODE_RX_STATEWAIT4SSMSTATE_GETTELEGRAM_TIMEOUT This error code is set if the GET_TELEGRAM SSM is not started in the right time. If this error occurs, the system is switched to IDLEMode via ShutDown SSM.
5	Reserved
6	Reserved
7	Reserved
8	DEBUG_ERROR_CODE_POLL_STATEWAIT4SSMSTATE_GETTELEGRAM_TIMEOUT This error code is set if the GET_TELEGRAM isn't started in the right time. If this error occurs, the system is switched to IDLEMode via ShutDown SSM.
9	Reserved
10	Reserved
11	DEBUG_ERROR_CODE_VCOCAL_STATESTARTSSM_TIMER_LOCKED This error code is set if the SSM watchdog timer (Timer2) is not available for VCO tuning.
12	DEBUG_ERROR_CODE_VCOCAL_STATEWAIT4SSMRDY_TIMEOUT This error code is set if the VCO tuning SSM isn't finished in the right time. If this error occurs, the system is switched to IDLEMode via ShutDown SSM.
13	DEBUG_ERROR_CODE_SHUTDOWNTRX_TIMER_LOCKED This error code is set if the SSM watchdog timer (Timer2) is not available for ShutDown SSM.
14	DEBUG_ERROR_CODE_SHUTDOWNTRX_TIMEOUT This error code is set if the ShutDown SSM isn't finished in the right time.
15	DEBUG_ERROR_CODE_POLLING_TIMER1_LOCKED This error code is set if the polling timer (Timer1) is not available for polling.
16	DEBUG_ERROR_CODE_INVALID_OPM_SWITCHING This error code is set if the OPM switching is not allowed.
17	DEBUG_ERROR_CODE_INVALID_OPM_MODE_DURING_TUNE_AND_CHECK This error code is set if TuneAndCheck is requested via SPI and the system is not in IDLEMode.
18	Reserved
19	Reserved
20	DEBUG_ERROR_CODE_DFIFO_OVER_UNDER_FLOW This error code is set if a DFIFO overflow/underflow is detected. The corresponding interrupt is disabled to avoid continuous overflow/underflow interrupts.

Table 7-2. Error Codes Description (Continued)

errorCode	Error Description
21	DEBUG_ERROR_CODE_SFIFO_OVER_UNDER_FLOW This error code is set if a SFIFO overflow/underflow is detected. The corresponding interrupt is disabled to avoid continuous overflow/underflow interrupts.
22	DEBUG_ERROR_CODE_RSSI_STATESTARTSSM_TIMER_LOCKED This error code is set if the SSM watchdog timer (Timer2) is not available for RSSI measurement.
23	DEBUG_ERROR_CODE_SRC_FRC_CALIB_FAILED This error code is set if Timer2 and Timer3 are no available for SRC/FRC calibration
24	DEBUG_ERROR_CODE_GETRXTELEGRAM_SSM_ERROR This error code is set if GetRxTelegram SSM returns with error code and SSM_RDY
25	Reserved
26	Reserved
27	DEBUG_ERROR_CODE_AVCCLOW_TIMEOUT This error code is set if AVCC isn't enabled in the right time (180µs).
28	Reserved
29	Reserved
30	Reserved
31	Reserved
32	Reserved
33	Reserved

Table 7-3. SSM Error Codes Description

ssmErrorCode	Error Description
0x81	This error code is set if an error occurs in SSM "PLL enable"
0x82	This error code is set if an error occurs in SSM "PLL lock"
0x83	This error code is set if an error occurs in SSM "Rx DSP enable"
0x84	This error code is set if an error occurs in SSM "Rx DSP disable"
0x85	Reserved
0x86	Reserved
0x87	Reserved
0x88	Reserved
0x89	This error code is set if an error occurs in SSM "Get telegram"
0x8A	Reserved
0x8B	This error code is set if an error occurs in SSM "Shut down"
0x8C	This error code is set if an error occurs in SSM "VCO-Tuning"
0x8D	Reserved

7.6 AVR Timer Assignment

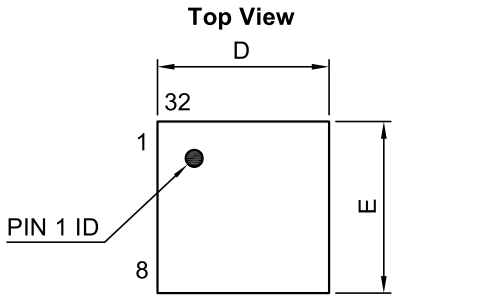
Table 7-4. AVR Timer Assignment Description

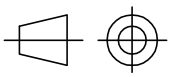
Timer	Function	Comment
0	Watchdog	Reserved for firmware usage
1	Polling Cycle	Reserved for firmware usage
2	FRC Calibration SRC Calibration SSM Watchdog	Reserved for firmware usage
3	FRC Calibration SRC Calibration	Reserved for firmware usage
4	-	Reserved for future use
5	-	Reserved for future use

8. Ordering Information

Extended Type Number	Package	Remarks
ATA5785-GHQW	QFN32	5mm x 5mm, 6k tape and reel, PB-free, wettable flanks

9. Package Information

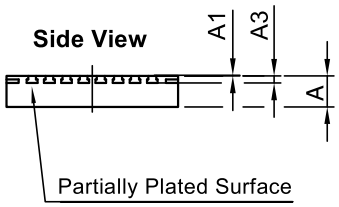


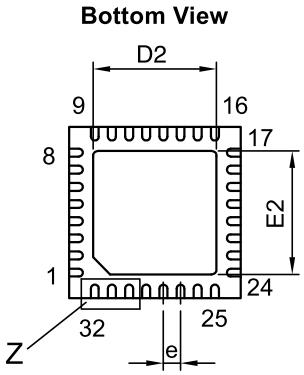


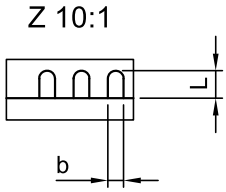
technical drawings
according to DIN
specifications

Dimensions in mm

Two Step Singulation process







COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.8	0.9	1	
A1	0.0	0.02	0.05	
A3	0.15	0.2	0.25	
D	4.9	5	5.1	
D2	3.45	3.6	3.75	
E	4.9	5	5.1	
E2	3.45	3.6	3.75	
L	0.35	0.4	0.45	
b	0.16	0.23	0.3	
e		0.5 BSC		

11/30/11

Atmel Package Drawing Contact: packagedrawings@atmel.com	TITLE **Package: VQFN_5x5_32L** Exposed pad 3.6x3.6	GPC	DRAWING NO. 6.543-5124.02-4	REV. 2

10. 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
9360BX-RKE-11/14	• Section 2.8.7.9 “External LNA Control” on page 49 updated • Section 4.1.4 “External LNA Support” on page 293 updated

11. Table of Contents

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**