ATA5785

UHF ASK/FSK Receiver

USER MANUAL

Features

- AVR® microcontroller core with 1Kbyte SRAM and 24Kbyte RF library in firmware (ROM)
- Supported frequency ranges
  - Low-Band 310MHz to 318MHz, 418MHz to 477MHz
  - 315.00MHz/433.92MHz with one 24.305MHz crystal
- Low current consumption
  - 9.8mA for RXMode (Low-Band), 1.2mA for 21ms cycle three-channel polling
- Typical OFFMode current of 5nA (maximum 600nA at Vs = 3.6V and T = 85°C)
- Input 1dB compression point
  - –48dBm (full sensitivity level)
- Programmable channel frequency with fractional-N PLL
  - 93Hz resolution for Low-Band
- FSK deviation ±0.375kHz to ±93kHz
- FSK sensitivity (Manchester coded) at 433.92MHz
  - –108.5dBm at 20Kbit/s Δf = ± 20kHz BWIF = 165kHz
  - –111dBm at 10Kbit/s Δf = ± 10kHz BWIF = 165kHz
  - –114dBm at 5Kbit/s Δf = ± 5kHz BWIF = 165kHz
  - –122.5dBm at 0.75Kbit/s Δf = ± 0.75kHz BWIF = 25kHz
- ASK sensitivity (Manchester coded) at 433.92MHz
  - –110.5dBm at 20Kbit/s BWIF = 80kHz
  - –125dBm at 0.5Kbit/s BWIF = 25kHz
- Programmable Rx-IF bandwidth 25kHz to 366kHz (approximately 10% steps)
- Blocking (BWIF = 165kHz): 64dBc at frequency offset = 1MHz and 48dBc at 225kHz
- High image rejection: 55dB at 315MHz/433.92MHz without calibration
- Supported data rate in buffered mode 0.5Kbit/s to 80Kbit/s (120Kbit/s NRZ)
- Supports pattern-based wake-up and start of frame identification

- Flexible service configuration concept with on-the-fly (OTF) modification (in IDLEMode) of SRAM service parameters (data rate, …)
    - Two service configurations are located in SRAM and can be modified via SPI
- Digital RSSI with very high relative accuracy of ±1dB thanks to digitized IF processing
- Programmable clock output derived from crystal frequency
- SPI interface for Rx data access and receiver configuration
- 500Kbit SPI data rate for short periods on SPI bus and host controller
- Configurable EVENT signal indicates the status of the IC to an external microcontroller
- Automatic low-power channel polling with flexible configuration concerning timing, order and participating channels
- Fast reaction time
- Support of mixed ASK/FSK telegrams
- Non-byte aligned data reception
- Supply voltage ranges 1.9V to 5.5V
- Temperature range –40°C to +105°C
- ESD protection at all pins (±4kV HBM, ±200V MM, ±750V FCDM)
- Small 5×5mm QFN32 package/pitch 0.5mm
- Package and pin-to-pin compatibility with Atmel® ATA5780N and ATA5781/2/3
- Backward RF matching compatibility with Atmel ATA5780N and ATA5781/2/3 (RF redesign not needed)
- Suitable for applications governed by EN 300 220 and FCC part 15, title 47

1. General Product Description

1.1 Introduction

The Atmel® ATA5785 is a highly integrated, low-power UHF ASK/FSK RF receiver with an integrated AVR® microcontroller. It is package and pin-to-pin compatible with the previous generation of RF receivers (Atmel ATA5780N and ATA5781/2/3).

The Atmel ATA5785 is partitioned into three sections; an RF front end, a digital baseband and the low-power 8-bit AVR microcontroller. The product is designed for the ISM frequency bands in the ranges of 310MHz to 318MHz and 418MHz to 477MHz. The external part count is kept to a minimum due to the very high level of integration in this device. By combining outstanding RF performance with highly sophisticated baseband signal processing, robust wireless communication can be easily achieved. The receive path uses a low-IF architecture with an integrated double quadrature receiver and digitized IF processing. This results in high image rejection and excellent blocking performance. In addition, highly flexible and configurable baseband signal processing allows the receiver to operate in several scanning, wake-up and automatic self-polling scenarios. For example, during polling the IC can scan for specific message content (IDs) and save valid telegram data in the FIFO buffer for later retrieval. The device integrates two receive paths that enable a parallel search for two telegrams with different modulations, data rates, wake-up conditions, etc.

The Atmel ATA5785 implements a flexible service configuration concept and supports up to 6 channels. The channels are grouped into two service configurations with three channels each. The service configurations are located in the SRAM and allow on-the-fly modifications during IDLEMode via SPI commands. Highly configurable and autonomous scanning capability enables flexible polling scenarios with up to 15 channels. The SPI interface allows external control and device configuration.

1.2 Document Guide

Chapter 1 “General Product Description”

Chapter 1 provides an initial overview of the functionality, features, and application areas of the Atmel® ATA5785. Pin descriptions are included in this chapter.

Chapter 2 “System Functional Description”

Chapter 2 provides a functional description of the entire system. All features, operating modes, and configuration options are described. This chapter will interest

- Standard users who want to gain a functional understanding of the IC
- Users who want set up a configuration for a specific application
- Users who want to use the built-in firmware library by controlling the IC via the SPI interface

Chapter 3 “Hardware”

Chapter 3 provides a detailed description of the Atmel ATA5785 hardware. Every hardware block is described including all registers. This chapter will interest

- Advanced users who want to gain a deep understanding of the IC
- Users who want to manipulate internal hardware settings during operation to achieve customized behavior of the IC

Chapter 4 “Application”

Chapter 4 covers the application board design and general information on Rx-Tx system design. The chapter includes examples and will interest

- Application engineers
- System engineers who design the overall Rx-Tx link of the transmission system

Chapter 5 “Electrical Characteristics”

Chapter 5 lists all parameters of the Atmel ATA5785 including absolute maximum ratings, supply voltages, current consumption, and RF characteristics.

Chapter 6 “Timing Characteristics”

Chapter 6 contains sample timing values for start-up of operating modes and switch between operating modes.

Chapter 7 “Appendix”

The Appendix contains a list of abbreviations, SPI commands, and tables with various memory maps. The chapter is intended to be a quick reference guide for commands, variables, and registers.

Chapter 8 “Ordering Information”

Chapter 8 provides the order numbers for the Atmel ATA5785.

Chapter 9 “Package Information”

Chapter 9 shows the precise dimensions of the Atmel ATA5785 package. This information is useful for the application board design.

Chapter 10 “Revision History”

Chapter 10 contains the version history and change notes of all releases of this document.

Chapter 11 “Table of Contents”

Chapter 11 shows an outline of this document down to the third heading level. A complete bookmark list with all heading levels is included in the PDF version of this document.

1.3 System Overview

Figure 1-1. Circuit Overview

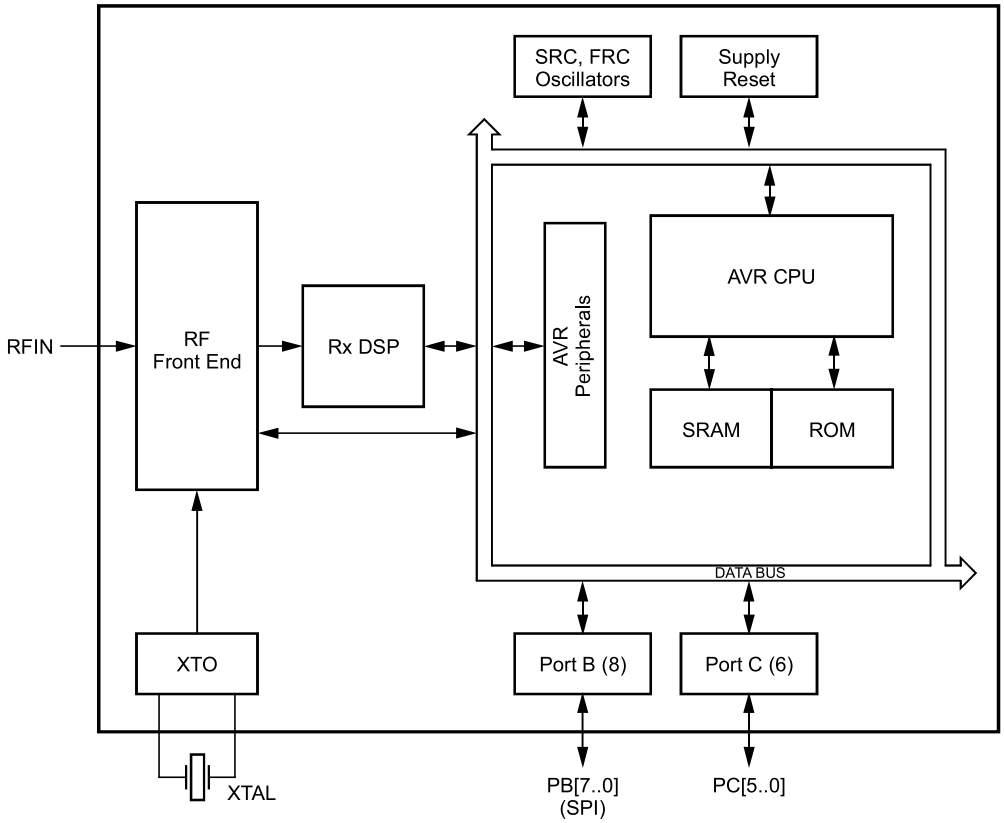


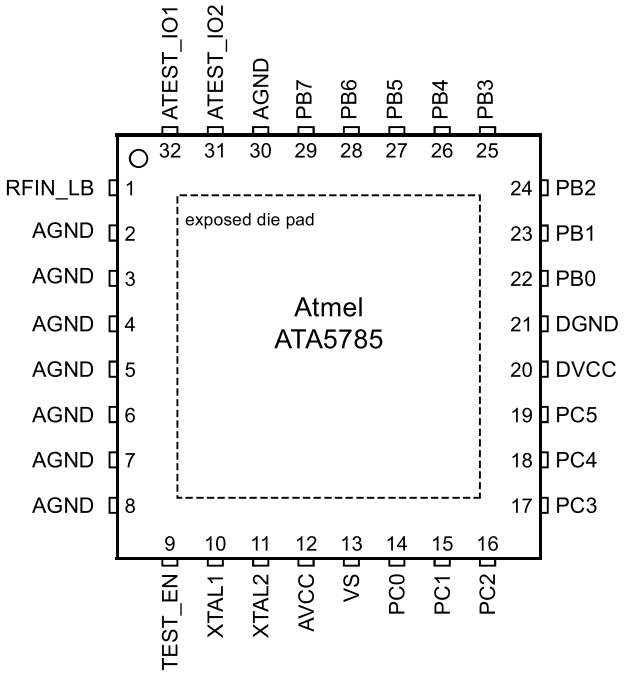
Figure 1-1 on page 5 shows an overview of the main functional blocks of the Atmel® ATA5785. External configuration and control of the Atmel ATA5785 is performed through the SPI pins SCK, MOSI, MISO, and NSS on port B. A large portion of the functionality is defined by the firmware located in the ROM and processed by the AVR®. The service configuration is written to the SRAM via SPI after power-on by an external host microcontroller. A subsequent SPI command can trigger the AVR to configure the hardware according to these settings and start up a given system mode (e.g., RXMode, or PollingMode). Internal events such as “Start of Telegram” or “FIFO empty” are signaled to the external microcontroller on pin 28 (PB6/EVENT).

It is important to note that all PWRON and NPWRON pins (PC1..5, PB4, PB7) are active in OFFMode. This means that even if the Atmel ATA5785 is in OFFMode and the DVCC voltage is switched off, the power management circuitry within the Atmel ATA5785 biases these pins with VS.

AVR ports can be used as button inputs, external LNA supply voltage (RX_ACTIVE), LED drivers, EVENT pin, general purpose digital inputs, or wake-up inputs, etc.

1.4 Pinning

Figure 1-2. Pin Diagram



Note: The exposed die pad is connected to the internal die.

Table 1-1. Pin Description

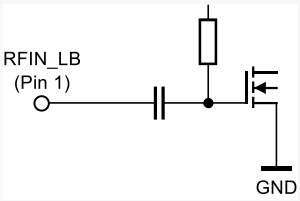
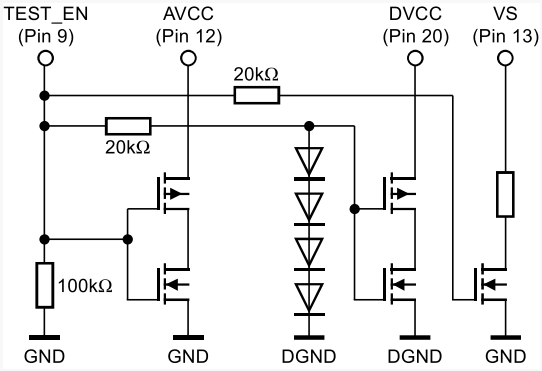
Pin No.	Pin Name	Type	Equivalent Circuit	Description
1	RFIN_LB	Analog		LNA input for Low-Band frequency range (< 500MHz)
2-8	AGND	–		Connect to GND in application
9	TEST_EN	–		Test enable, connected to GND in application

Table 1-1. Pin Description (Continued)

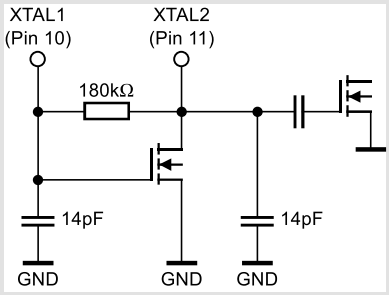
Pin No.	Pin Name	Type	Equivalent Circuit	Description	
10	XTAL1	Analog		Crystal oscillator pin 1 (input)	
11	XTAL2	Analog		Crystal oscillator pin 2 (output)	
12	AVCC	Analog		RF front end supply regulator output	
13	VS	Analog		Main supply voltage input	
14	PC0	Digital		Main Alternate	: AVR Port C0 : PCINT8 / NRESET
15	PC1	Digital		Main Alternate	: AVR Port C1 : NPWRON1 / PCINT9 / EXT_CLK
16	PC2	Digital		Main Alternate	: AVR Port C2 : NPWRON2 / PCINT10 / TRPA
17	PC3	Digital		Main Alternate	: AVR Port C3 : NPWRON3 / PCINT11 / TMDO
18	PC4	Digital		Main Alternate	: AVR Port C4 : NPWRON4 / PCINT12 / INT0
19	PC5	Digital		Main Alternate	: AVR Port C5 : NPWRON5 / PCINT13 / TRPB / TMDO_CLK
20	DVCC	–		Digital supply voltage regulator output	
21	DGND	–		Digital ground	
22	PB0	Digital		Main Alternate	: AVR Port B0 : PCINT0 / CLK_OUT
23	PB1	Digital		Main Alternate	: AVR Port B1 : PCINT1 / SCK
24	PB2	Digital		Main Alternate	: AVR Port B2 : PCINT2 / MOSI (SPI Master Out Slave In)
25	PB3	Digital		Main Alternate	: AVR Port B3 : PCINT3 / MISO (SPI Master In Slave Out)
26	PB4	Digital		Main Alternate	: AVR Port B4 : PWRON / PCINT4 / LED1 (strong high side driver)

Table 1-1. Pin Description (Continued)

Pin No.	Pin Name	Type	Equivalent Circuit	Description	
27	PB5	Digital		Main Alternate	: AVR Port B5 : PCINT5 / INT1 / NSS
28	PB6	Digital		Main Alternate	: AVR Port B6 : PCINT6 / EVENT (firmware controlled external microcontroller event flag)
29	PB7	Digital		Main Alternate	: AVR Port B7 : NPWRON6/ PCINT7/ RX_ACTIVE (strong high side driver) / LED0 (strong low side driver)
30	AGND	–		Analog ground	
31	ATEST_IO2	–		RF front end test I/O 2 connected to GND in application	
32	ATEST_IO1	–		RF front end test I/O 1 connected to GND in application	
	GND	–		Ground/backplane on exposed die pad	

1.5 Typical Applications

The receiver is designed to be used in the following application areas:

- Remote keyless entry system (RKE)
- Passive entry go system (PEG)
- Tire pressure monitoring system (TPM, TPMS)
- Remote start system (RS)
- Remote control systems, e.g., garage door openers
- Smart RF applications
- Telemetering systems

1.5.1 Typical 5V Application Circuit with External Microcontroller

Figure 1-3. Typical 5V Application Circuit with External Microcontroller

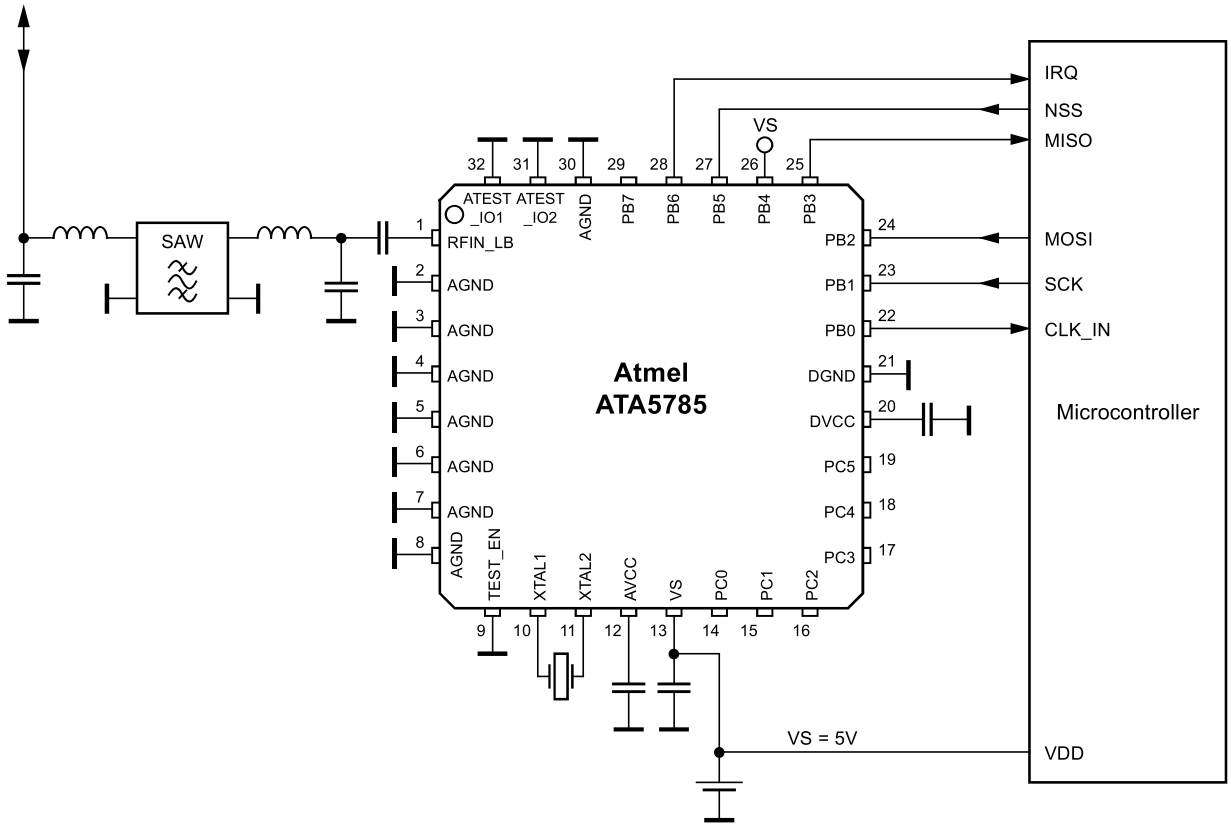


Figure 1-3 shows a typical vehicle side application circuit with an external host microcontroller running from a 5V voltage regulator. The pin PB4 (PWRON) is directly connected to VS and the Atmel® ATA5785 enters the IDLEMode after power-on. In this configuration the Atmel ATA5785 can work autonomously and the µC stays powered down to keep current consumption low while remaining sensitive to RF telegrams and SPI communication.

To achieve a low current in IDLEMode the Atmel ATA5785 can be configured to work with the RC oscillator. The Atmel ATA5785 can also be configured for autonomous multi-channel and multi-application PollingMode. The external µC is notified by an event on pin 28 (EVENT) if an appropriate RF message is received. Until this event, the Atmel ATA5785 periodically switches to RXMode, checks the configured services and channels, and returns to power-down while the external host microcontroller is still in deep sleep mode to keep average current low. Once a valid RF message is detected, it can be buffered inside of the Atmel ATA5785 to enable a µC wake-up and retrieval of buffered data.

The impedance of the SAW filter is transformed with LC matching circuits to the RFIN_LB port and also to the antenna. An external crystal, together with the fractional-N PLL within the Atmel ATA5785 is used to fix the RX frequency. Accurate load capacitors for this crystal are integrated, to reduce system part count and cost. Only three supply blocking capacitors are needed to decouple the different supply voltages AVCC, DVCC and VS of the Atmel ATA5785. The exposed die pad is the RF and analog ground of the Atmel ATA5785. It is directly connected to AGND via a fused lead. The Atmel ATA5785 is controlled using specific SPI commands via the SPI interface for application specific configuration.

2. System Functional Description

2.1 Overview

2.1.1 Service-based Concept

The Atmel® ATA5785 is a highly configurable UHF receiver. General chip-wide settings and service configurations must be written via SPI to the internal SRAM and hardware registers after system initialization. During start-up of a receive mode the specific settings are loaded to the current service in the SRAM and from there to the corresponding hardware registers by the firmware.

A complete configuration set of the receiver is called “service” and includes RF settings, demodulation settings, and telegram handling information. Each service contains three channels which differ in the RF receive frequencies.

The Atmel ATA5785 supports two services which can be configured in various ways to meet customer requirements.

The service configurations are located in the SRAM space and can be modified by an SPI command during IDLEMode.

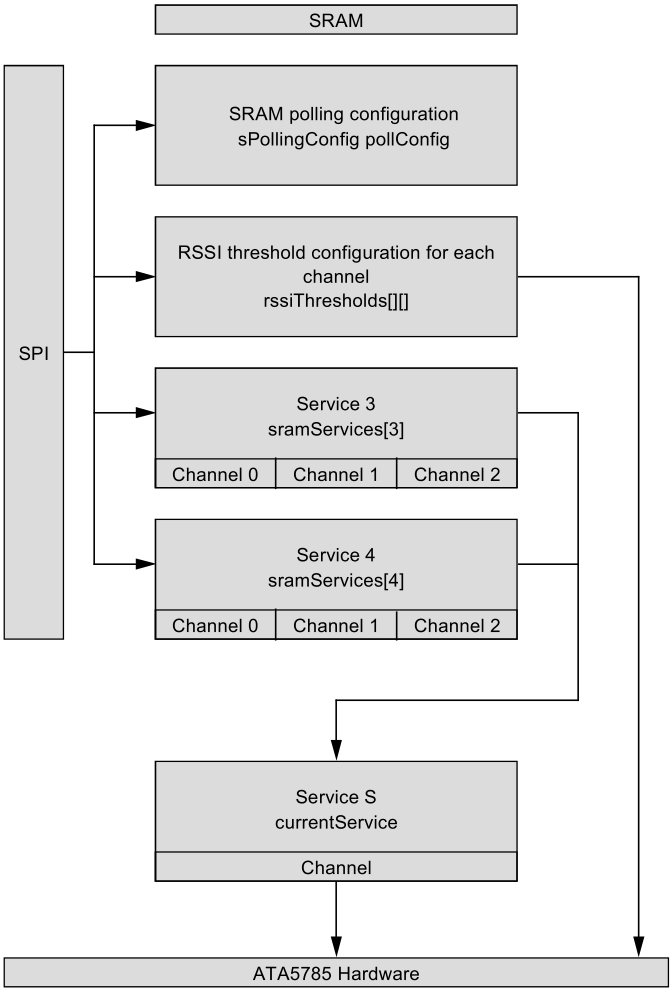
A service consists of

- One service-specific configuration part
- Three channel-specific configuration parts

Further configurations for PollingMode and RSSI are available and can be also modified in IDLEMode via SPI commands.

Figure 2-1 on page 10 gives an overview on the service based-concept.

Figure 2-1. Service-based Concept Overview



2.1.2 Supported Telegrams

2.1.2.1 Telegram Structure

The Atmel® ATA5785 supports the reception of a wide variety of telegrams and protocols. Generally no special structure is required from a telegram to be received by the Atmel ATA5785. However, designated hardware and software features are built in for the blocks that are depicted in Figure 2-2. Using this structure or parts of it can increase the sensitivity and robustness of the broadcast.

Figure 2-2. Telegram Structure

Desync:

The de-synchronization is usually a coding violation with a length of several symbols that should provoke a defined restart of the receiver. The use of a de-synchronization leads to more deterministic receiver behavior, reducing the required preamble length. This can be favorable in timing-critical and energy-critical applications.

Preamble:

The preamble is a pattern that is sent before the actual data payload to synchronize the receiver and provide the starting point of the payload. A very regular pattern (e.g., 1-0-1-0...) is recommended for synchronization (“wake-up pattern, WUP”, sometimes also called “pre-burst”) while a unique, well-defined pattern of up to 32 symbols is required to mark the start of the data payload (“start frame identifier, SFID” or “start bit”). In polling scenarios the WUP can be tens or hundreds of ms long.

Data Payload:

The data payload contains the actual information content of the telegram. It can be NRZ or Manchester-coded. The length of the payload is application dependent, typically 1..64 bytes.

Checksum:

A checksum can be calculated across the data payload to verify that the data have been received correctly. A typical example is an 8-bit CRC checksum. Data bits at the beginning of the payload can be excluded from the CRC calculation.

Stop Sequence:

The stop sequence is a short data pattern (typically 2 to 6 symbols) to mark the end of the telegram. A coding violation can be used to prevent additional (non-deterministic) data from being received.

2.1.2.2 NRZ and Manchester Coding

Within this document the following wording is used:

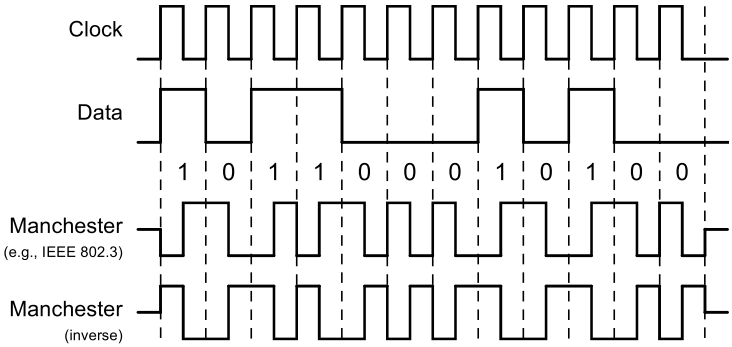
The expression data “bit” describes the real information content that is to be broadcast. This information can be coded in “symbols” (sometimes also called “chips” or “bauds”) which are then physically transmitted from sender to receiver. The receiver has to decode the “symbols” back into data “bits” to access the information. The “symbol rate” is therefore always greater or equal to the “bit rate”.

The Atmel ATA5785 supports two coding modes: Manchester coding and non-return-to-zero (NRZ) coding.

NRZ coding is implemented in a straightforward manner: One bit is represented by one symbol.

Manchester coding implements two symbols per data bit. There is always a transition between the two symbols of one data bit so that one data bit always consists of a “0” and a “1”. The polarity can be either way as shown in Figure 2-3 on page 12.

Figure 2-3. Manchester Code

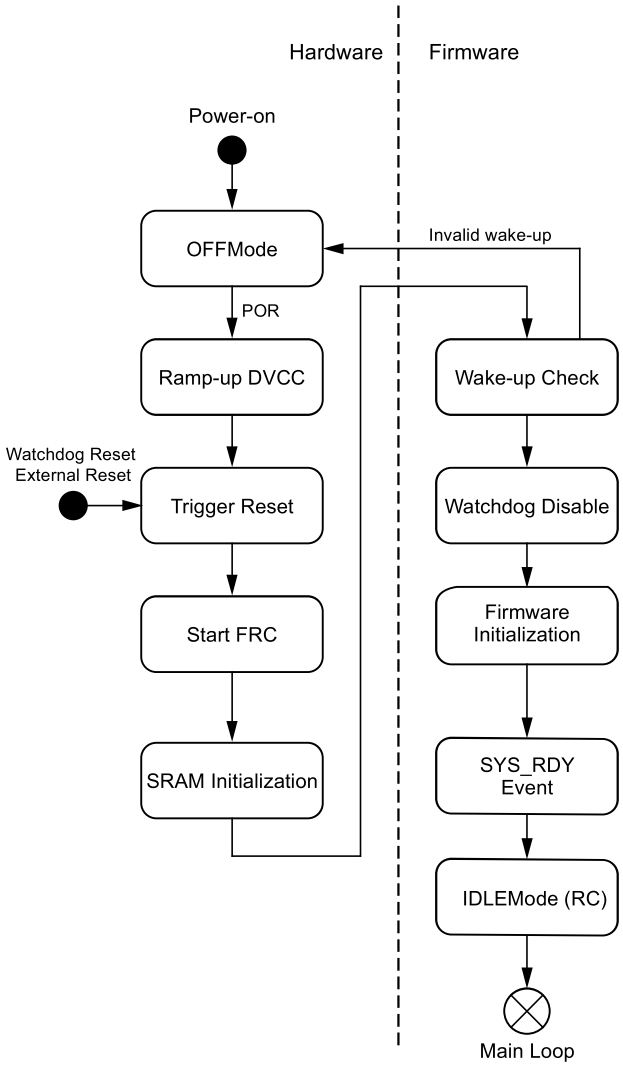


Manchester coding has many advantages such as simple clock recovery, no DC component, and error detection by code violation. Drawbacks are the coding/decoding effort and the increased symbol rate which is twice the data rate.

2.2 System Start-up

Figure 2-4 shows the system start-up flow, consisting of hardware and firmware initialization, after power-on.

Figure 2-4. System Start-up



After connecting the supply voltage to the VS pin, the Atmel® ATA5785 always starts in OFFMode. All internal circuits are disconnected from the power supply. The Atmel ATA5785 can be woken up by one of the following power-on reset (POR) events:

- Setting the PWRON pin to HIGH
- Setting one of the NPWRON1..6 pins to LOW

For more information on the OFFMode refer to Section 2.6 “OFFMode” on page 18. A POR enables the DVCC regulator. The FRC oscillator is started when the DVCC voltage reaches the operating voltage. At the same time the internal AVR® reset is triggered. After reaching stable operation conditions the FRC provides the clock for the hardware initialization state machine. The initialization state machine resets the SRAM content to “0x00”. The internal AVR reset is released when the hardware initialization is done. The system is now controlled by the firmware.

The firmware steps through the following tasks:

- Check for a correct wake-up condition. This is done by reading the pin registers (PINB and PINC) to identify the wake-up source. If a correct wake-up is detected, the wake-up source is signalled as an event. If no wake-up source is detected the wake-up is considered as invalid and the ATA5785 is switched back to OFFMode. Reset sources like watchdog reset, external reset or brown-out reset are not checked. This implies that the chip goes to OFFMode if, e.g., a watchdog reset occurred and no (N) PWRON pin is triggered.
- Disable the watchdog timer to avoid repeated watchdog resets
- Set EVENT pin (pin 28/PB6) to low to signal system ready event to the external host.

The operating mode after system start is the IDLEMode (RC). The firmware is executing the main loop.

The system is now ready to receive the configuration and service data via SPI.

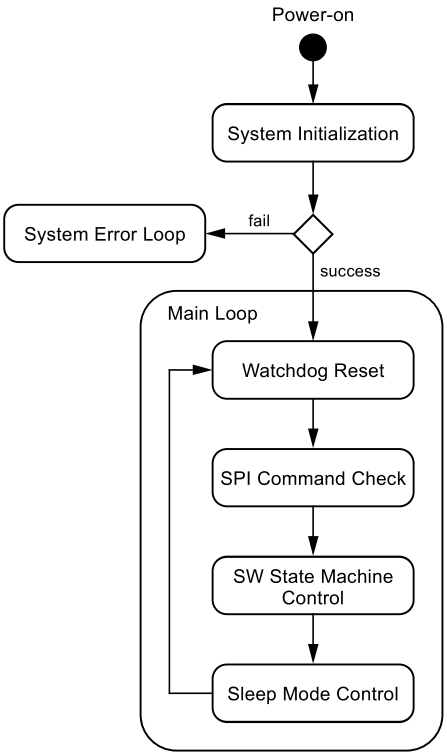
2.3 Main Loop

The main loop is executed periodically after successful system start-up. As shown in Figure 2-5, the main loop executes the following steps:

- Resetting the watchdog timer if activated
- Checking for external requests from SPI interface and configuration of the software state machines
- Controlling the software state machines during switch to operating modes
- Checking for the sleep mode conditions and entering sleep mode if necessary

A main loop cycle takes approximately 25µs at a standard FRC clock.

Figure 2-5. Main Loop Flow



2.4 System Configuration

2.4.1 Clock Configuration

If they are not used by at least one module, the FRC and SRC oscillators are automatically switched off by default. This mechanism can be deactivated, if required. The oscillators are then running permanently, which can decrease the start-up time from sleep mode. The feature can be activated by the FRC always on (FRCAO) and SRC always on (SRCAO) bits in the systemConfig SRAM variable. For more information, see Section 2.12.4.1 “systemConfig” on page 67.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E5	systemConfig	-	-	SRCAO	FRCAO	SFIFO_OFL_UFL_RX_disable	DFIFO_OFL_UFL_RX_disable	Current idle mode selection	-

A system clock can be output on the CLK_OUT pin. Available clock sources are SRC, FRC and XTO. The clock divider must be configured to result in an output clock of less then 4.5MHz (see parameter no. 15.90 in Section 5. “Electrical Characteristics” on page 300). The external clock output can be configured and activated in the CLKOCR and CLKOD hardware registers with the SPI command “Write SRAM/Register”, the following sequence shall be used:

- Disable clock output (CLKOCR.CLKOEN=0)
- Write clock output divider (CLKOD) and source select (CLKOCR.CLKOS[1:0])
- Enable clock output (CLKOCR.CLKOEN=1)

For more information, see Section 3.8.4 “System Clock and Clock Options” on page 201.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00C4	CLKOCR	-	-	-	-	-	CLKOEN	CLKOS[1:0]	
0x00C3	CLKOD	CLKOD[7:0]							

Note: If the clock output is enabled, the I/O port configuration of pin PB0 must be set to output

2.4.2 AVR Sleep Mode Configuration

The AVR® has an integrated sleep mode to reduce power consumption during inactive periods (e.g., the sleep period of a polling cycle). The firmware controlled sleep mode feature can be configured in the SRAM variable sleepModeConfig. One of four different sleep modes can be selected. After start-up the sleep mode IDLE is active. For more information, see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02EB	sleepModeConf	SLEEP_ENA	-	-	-	SM[2:0]			-

2.4.3 Watchdog Configuration

The Atmel® ATA5785 integrates a watchdog timer that prevents the system from stalling. The time-out period can be configured in eight steps between 1ms and 268s. The reference clock for the watchdog timer is the SRC; therefore the watchdog timer also works in all sleep modes.

If enabled, the watchdog timer has to be reset by a watchdog reset (WDR) AVR instruction before a time-out occurs. Otherwise a full system reset is triggered. The WDR is done by the firmware during the main loop. Due to the fact that in PollingMode the main loop has a low priority, the time-out value of the watchdog should be at least twice as long as the active period in PollingMode. If PollingMode is configured with an AVR sleep mode, the WDR is disabled during the sleep period. As a result, the time-out value of the watchdog must be set longer than the sleep period in PollingMode. If the system runs in a receive mode, the WDR is not executed as long as the system is waiting for wake check ok (WCO) signalization and the time-out value of the watchdog has to be set longer than the maximum demodulator/WCO time-out.

The watchdog is disabled after system start-up. It can be enabled by the SPI command “Set Watchdog”, see Section 7.4 “SPI Command Reference” on page 345.

A detailed description of the watchdog timer hardware IP is given in Section 3.8.7.1 “Timer0 – Watchdog/Interval Timer” on page 233.

2.4.4 IO Port Configuration

The initial start-up configuration of port B and port C is done according to Table 2-1.

Table 2-1. I/O Port Configuration after System Start-up

Pin	Definition
Pin14/PC0 (NRESET)	Input pull-up
Pin15/PC1 (NPWRON1)	Input pull-up
Pin16/PC2 (NPWRON2, TRPA)	Output low
Pin17/PC3 (NPWRON3, TMDO)	Output low
Pin18/PC4 (NPWRON4)	Input pull-up
Pin19/PC5 (NPWRON5, TRPB, TMDO_CLK)	Output low
Pin22/PB0 (CLK_OUT)	Output low
Pin23/PB1 (SCK)	Input tri-state
Pin24/PB2 (MOSI)	Input tri-state
Pin25/PB3 (MISO)	Output high
Pin26/PB4 (PWRON)	Input tri-state
Pin27/PB5 (NSS)	Input tri-state
Pin28/PB6 (EVENT)	Output high
Pin29/PB7 (NPWRON6, RX_ACTIVE)	Output low

The port settings can be changed by SPI access to the corresponding hardware registers.

This implies

- Definition of the data direction of each digital I/O pin (input/output)
- Definition of the output value (high/low) if a pin is defined as output
- Enabling/disabling of a pull-up resistor if a pin is defined as input
- Enabling/disabling of the high/low side drivers on PB7 and the high side driver on PB4
- Pin change interrupt configuration

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0006	DDRB	DDRB[7:0]							
0x0009	DDRC	DDRC[7:0]							
0x000E	MCUCR	PB7HS	PB7LS	PB4HS	PUD	ENPS	SPIIO	IVSEL	ÎVCE
0x0026	PCICR	-	-	-	-	-	-	PCIE[1:0]	
0x006C	PCMSK0	PCMSK0[7:0]							
0x006D	PCMSK1	PCMSK1[7:0]							
0x0007	PORTB	PORTB[7:0]							
0x000A	PORTC	PORTC[7:0]							

The I/O port configuration is automatically overwritten by the firmware if certain modes and functions are selected. A detailed description of the I/O ports hardware IP is given in Section 3.8.6 “I/O Ports” on page 215.

2.4.5 Port Debounce Configuration

The ports of the Atmel® ATA5785 feature hardware debouncing that can be used to avoid false events coming from bouncing mechanical switches or bus glitches. A common debounce timer is used for all ports. The debouncing can be activated and deactivated independently on every digital I/O port. The debouncing time is configurable in the range of approximately 2µs to 250ms (see Table 3-61 on page 231), allowing key debouncing as well as LIN bus debouncing, for example. A level change on any port with activated debouncing resets the debounce timer.

There are two modes for debouncing: fast mode and stable mode. In stable mode 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. In fast mode a port event is transmitted to the AVR immediately. All following events are ignored for the configured debouncing time.

If debouncing is selected for any pin the only allowed sleep mode is “idle”. Other combinations are forbidden.

Debouncing can be configured and enabled by SPI access to the corresponding hardware registers. A detailed description is given in Section 3.8.6.2 “Port Debouncing” on page 219.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0153	DBCR	-	-	-	-	-	DBTMS	DBCS	DBMD
0x0154	DBTC	DBTC[7:0]							
0x0155	DBENB	DBENB[7:0]							
0x0156	DBENC	DBENC[7:0]							

2.4.6 FIFO Interrupt Initialization

The support FIFO (SFIFO) and the data FIFO (DFIFO) implement error interrupts for overflow and underflow events. These interrupts can be switched off by setting the SRAM variables systemConfig.SFIFO_OFL_UFL_RX_disable and systemConfig.DFIFO_OFL_UFL_RX_disable to “1”, respectively. For more information, see Section 3.5 “Data and Support FIFOs” on page 162

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E5	systemConfig	-	-	SRCAO	FRCAO	SFIFO_OFL_UFL_RX_disable	DFIFO_OFL_UFL_RX_disable	Current idle mode selection	-

2.4.7 AVCC Low Interrupt Configuration

if the analog voltage (AVCC) drops below a certain threshold voltage (typically 100mV below the nominal AVCC voltage), the event flag “AVCCLOW” in the event byte “system” can be raised and an event to the external host can be triggered. The event flag can be enabled by setting the AVCCLM bit in the SUPCR hardware register to “1”, the external event can additionally be enabled in the sysEventMask SRAM variable.

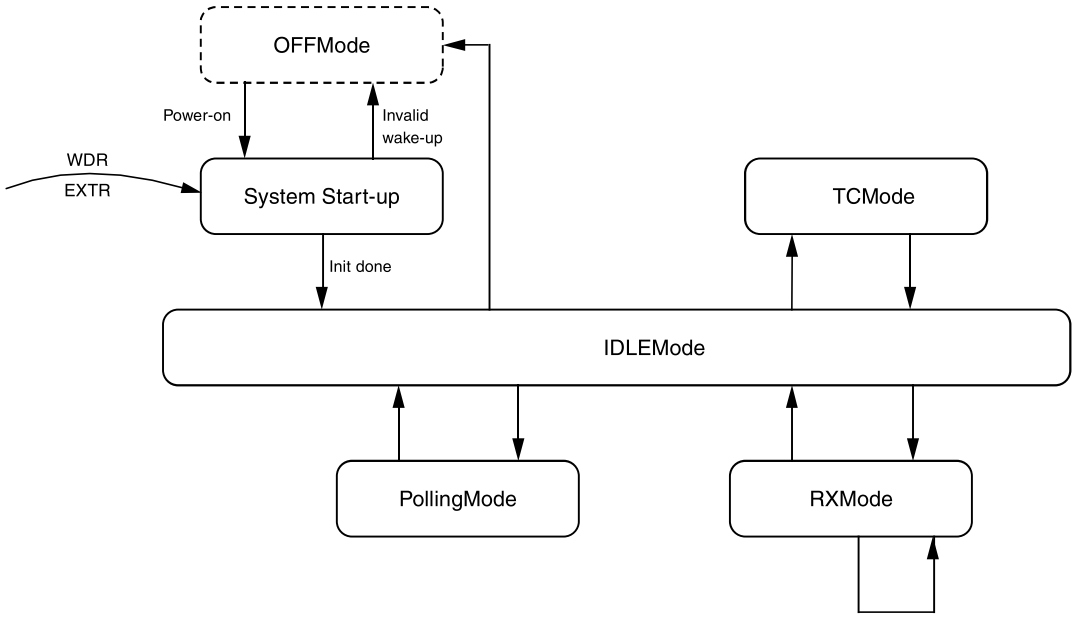
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00CB	SUPCR	-	AVDIC	AVEN	DVDIS			AVCCLM	AVCCRM

Address		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020E	sysEventMask	SYS_ERR	CMD_RDY	SYS_RDY	AVCCLOWM		RX_ACTIVE_EN	RX_ACTIVE_POL	IRQ_POL

2.5 Operating Modes Overview

This section gives an overview of the operating modes supported by the Atmel® ATA5785 as shown in Figure 2-6. For a more detailed description of the operating modes and their transitions, see later sections.

Figure 2-6. Operating Modes Overview



After connecting the supply voltage to the VS pin, the Atmel ATA5785 always starts in OFFMode. All internal circuits are disconnected from the power supply. Therefore, no SPI communication is supported. The Atmel ATA5785 can be woken up by activating the PWRON pin or one of the NPWRONx pins. This triggers the power-on sequence. After the system initialization the Atmel ATA5785 reaches the IDLEMode (RC).

The IDLEMode is the basic system mode supporting SPI communication and transitions to all other operating modes. There are two options of the IDLEMode:

- IDLEMode(RC) with low power consumption using the fast RC (FRC) oscillator for processing
- IDLEMode(XTO) with active crystal oscillator for high accuracy clock output or timing measurements

The receive mode (RXMode) provides data reception on the selected service/channel configuration. The precondition for data reception is a valid preamble. The receiver continuously scans for a valid telegram and receives the data if all pre-configured checks are successful. The RXMode is enabled by the SPI command “Set System Mode” as described in Section 7.4 “SPI Command Reference” on page 345.

In PollingMode the receiver is activated for a short period of time to check for a valid telegram on the selected service/channel configurations. The receiver is deactivated if no valid telegram is found and a sleep period with very low power consumption elapses. This process is repeated periodically in accordance with the polling configuration.

The tune and check mode (TCMode) offers calibration and self-checking functionality for the VCO and FRC oscillators as well as for polling cycle accuracy. This mode is activated via the SPI command “Calibrate and Check” described in Section 7.4 “SPI Command Reference” on page 345. Tune and check tasks can also be activated periodically during PollingMode.

Table 2-2 on page 18 shows the relations between the operating modes and their corresponding power supplies, clock sources, and sleep mode settings.

Table 2-2. Operating Modes versus Power Supplies and Oscillators

Operation Mode	AVR Sleep Mode	DVCC	AVCC	XTO	SRC	FRC
OFFMode	-	off	off	off	off	off
IDLEMode(RC)	Active mode Power-down(1)	on	off off	off off	on on	on off
IDLEMode(XTO)	Active mode Power-down(1)		on on	on on	on on	off off
RXMode	Active mode		on	on	on	off
PollingMode(RC) - Active period - Sleep period	Active mode Power-down(1)		on off	on off	on on	on off
PollingMode(XTO) - Active period - Sleep period	Active mode Power-down(1)		on on	on on	on on	off off

Notes: 1. During IDLEMode(RC) and IDLEMode(XTO) the AVR® microcontroller enters sleep mode to reduce current consumption. The power-down mode is recommended for keeping current consumption low.

2.6 OFFMode

The OFFMode is the mode with the lowest current consumption. Typically 5nA can be expected for an application driven by a 3V battery. In the OFFMode all internal circuitries are disconnected from the power supply. The device is not receptive for RF signals or SPI commands. Furthermore the ports are set to be inputs with or without pull-up resistors.

The Atmel® ATA5785 is in OFFMode after connecting the external power supply under the condition that no NPWRONx pin is set to “0” and the PWRON pin is not set to “1”. During operation a transition to OFFMode is initiated by the SPI command “OFF Command”.

The OFFMode can be left by setting one of the NPWRON1..6 pins to LOW or the PWRON pin to HIGH. This starts the transition to the IDLEMode.

2.7 IDLEMode

The IDLEMode is the basic system mode. From this mode all supported operating modes can be started. Most circuits are deactivated in IDLEMode.

The IDLEMode follows up the OFFMode at system start-up and system reset. It is used as the default mode after leaving other system modes and can be activated at any time by the SPI command “Set System Mode”.

To help attain a system with low power consumption, various AVR sleep modes are possible during the inactive period of the IDLEMode. The sleep mode configuration after start-up is the idle sleep mode. For the lowest current consumption it is recommended to set the AVR core to power-down mode. For more information, see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210.

The Atmel ATA5785 offers two different idle modes: IDLEMode(RC) and IDLEMode(XTO).

In IDLEMode(RC) only the DVCC voltage regulator, the FRC oscillator (active period) and the SRC oscillator (power-down period) are switched on. If enabled, the AVR automatically uses a sleep mode to achieve very low current consumption.

In the IDLEMode(XTO) the AVCC voltage domain and the XTO are additionally activated to provide a high accuracy system clock derived from the crystal oscillator. This increases power consumption compared to the IDLEMode(RC), but the start-up time of receive modes is decreased because the XTO is already running.

During system operation the IDLEMode can be selected by the SPI command “Set System Mode”.

If the IDLEModeSelector parameter is set to “1”, the system switches to IDLEMode(XTO) after the active operating mode has finished, otherwise to IDLEMode(RC). For more information see Section 7.4 “SPI Command Reference” on page 345.

Note: - The IDLEMode neither provides any internal status information nor generates external events.
- The IDLEMode is the only mode that allows reading/writing from/to SRAM, and hardware registers.

2.8 Data Reception

2.8.1 Rx Overview

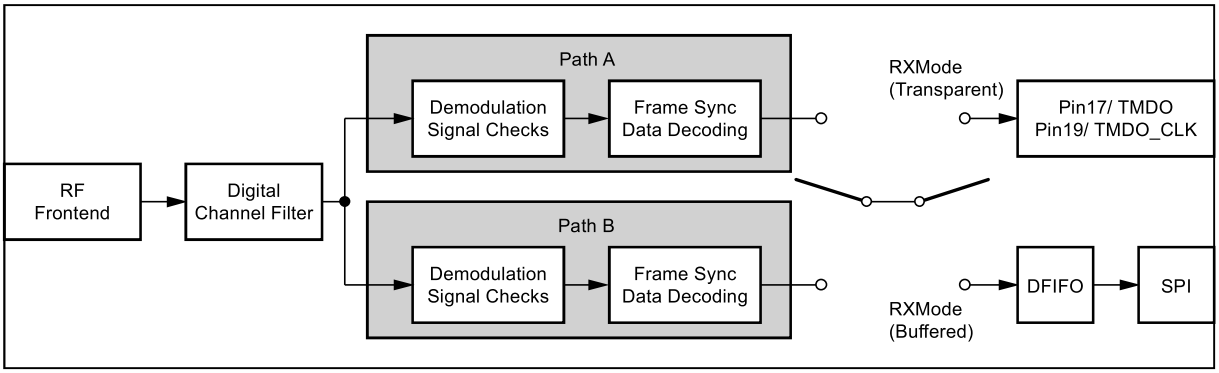
The Atmel® ATA5785 contains two receive paths A and B whose parameters, such as the modulation type, data rate, and telegram settings, can be set separately and are described in more detail in Section 2.8.6 “Rx Telegram Handling” on page 30 and Section 2.8.7 “General Rx Settings” on page 35. If individual settings are possible for the two paths, this is marked with a small “x” instead of an “A” or “B” in the subsequent sections.

The receive unit is split after the digital channel filter and merged again after the data decoding. This implies that:

- It is possible to search in parallel for telegrams with different modulation (ASK/FSK)
- It is possible to search in parallel for telegrams with different data rates
- It is possible to search in parallel for telegrams with different FSK deviations as long as the channel filter setting suits for both
- It is possible to configure signal checks and error handling individually for both paths
- It is possible to configure the preamble settings (WUP, SFID pattern) individually for both paths
- It is NOT possible to search in parallel for telegrams on different RF carrier frequencies
- It’s NOT possible to receive in parallel the data payload on both paths, because one path is switched off after WUP or SFID detections on the other path.

Figure 2-7 gives a high-level functional overview of the receive mode.

Figure 2-7. Receive Mode Overview



The data receive mode (RXMode) can be activated by using the “Set System Mode” SPI command if the OPM bits in the systemModeConfig parameter are set to “0b10”. The required service/channel configuration as well as additional settings such as VCO tuning during mode ramp-up, are part of the SPI command. For more information, see Section 7.4.8 “Set System Mode” on page 349.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
systemModeConfig	RF_CAL	-	VCO_TUNE	IDLEModeSelector	-	TMDEN	OPM[1:0]	
serviceChannelConfig	enaPathB	enaPathA	ch[1:0]		-	ser[2:0]		

The data reception is split into RXMode(transparent) and RXMode(buffered), selectable by the transparent mode data enable (TMDEN) bit. Both modes are described in more detail in the following subsections.

The RXMode can be used for all services without any restrictions. It supports service independent features such as:

- RSSI measurement (see Section 2.8.7.6 “RSSI Measurement” on page 47 for more information)

RXMode generally generates external events on pin 28 (EVENT) if a certain event occurs and the corresponding event mask in rxSysEvent is enabled. Events which are only relevant for the RXMode(transparent) and RXMode(buffered) are described in the associated sections. Commonly used events to indicate internal status information are located in the system (R15) and events (R14) as shown below.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
System (R15) events.system	SYS_ERR	CMD_RDY	SYS_RDY	AVCCLOW	-	SFIFO	DFIFO_RX	-
Events (R14) events.events	IDCHKA	WCOKA	SOTA	EOTA	IDCHKB	WCOKB	SOTB	EOTB

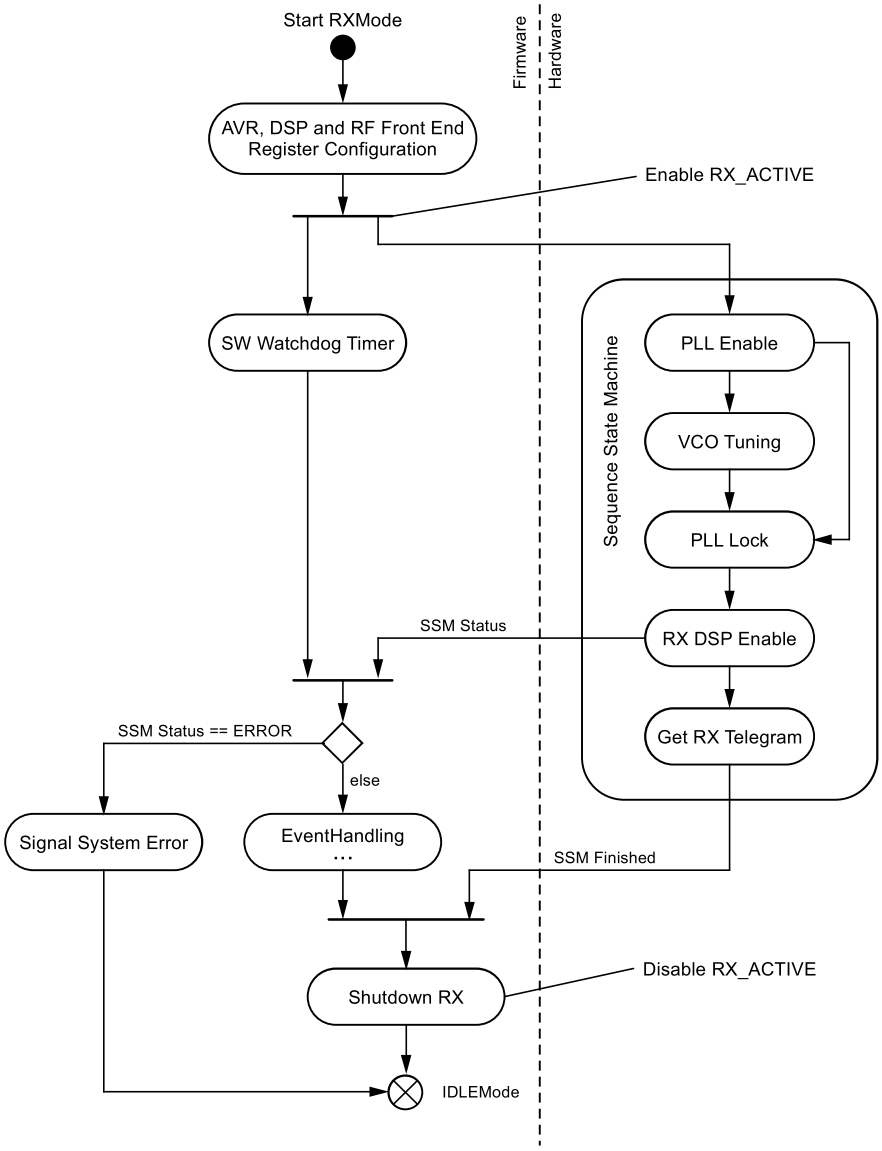
Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0344	rxSysEvent	IDCHKA _Mask	WCOKA _Mask	SOTA _Mask	EOTA _Mask	IDCHKB _Mask	WCOKB _Mask	SOTB _Mask	EOTB _Mask

In RXMode pin 29 (PB7) can be configured as the RX_ACTIVE signal to monitor the status of the RF front end. The RX_ACTIVE pin can be configured in the events.sysEventMask.RX_ACTIVE_EN and RX_ACTIVE_POL SRAM variables. The RX_ACTIVE signal can also be used for biasing an external LNA, see Section 2.8.7.9 “External LNA Control” on page 49 for details.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020E	sysEventMask	SYS_ERR	CMD_RDY	SYS_RDY	AVCC LOWM	-	RX_ACTIVE _EN	RX_ACTIVE _POL	IRQ_POL

The RXMode is controlled mainly by the sequencer state machine (SSM) which is located in the hardware and performs the start-up procedure. The firmware does the configuration and control of the SSM. Figure 2-8 shows the entire RXMode flow.

Figure 2-8. Receive Mode Flow



After the RXMode is activated, the firmware initializes the registers based on the settings in the selected service/channel configuration and configures the sequencer state machine for RXMode. If an RF calibration is selected, a temperature calibration of the XTO is carried out based on the latest available temperature value. If enabled, the RX_ACTIVE pin is set after the service/channel configuration is done and before the RF frontend is activated.

Start-up of the RF front end is done by the SSM and controlled by the firmware via a software watchdog (Timer2 is used). If there is no hardware response or an error message from the SSM, the firmware sets the SYS_ERR bit in the events.system event byte. The error code is stored in the SRAM variable debug, which contains the firmware error code (debug.errorCode) and the SSM error code (debug.ssmErrorCode). For more information, see Section 7.5 “Error Codes” on page 359.

It is possible to start-up the RXMode with or without VCO tuning at RXMode start-up. For more information see the “Set System Mode” SPI command.

After a successful start-up of the RF front end the Rx DSP is enabled and the Get RX Telegram state machine is started. This state machine controls the entire telegram reception process including synchronization, signal checking, data reception and decoding. In RXMode(transparent) the received payload data are routed to pins PC3 (TMDO) and PC5 (TMDO_CLK), in RXMode(buffered) the data are written to the data FIFO (DFIFO) for later retrieval. A detailed description of the Get RX Telegram state machine is given in section Section 2.8.6 “Rx Telegram Handling” on page 30.

The event handling is performed by the firmware in parallel to the telegram reception.

During shutdown of RXMode the RX_ACTIVE pin is disabled and the system returns to IDLEMode.

2.8.2 RXMode(transparent)

In the RXMode(transparent) the receiver provides the received data stream on pin 17 (TMDO) after a wake check OK (WCO) and a corresponding data clock on pin 19 (TMDO_CLK) after a valid start of telegram (SOT).

Figure 2-9 shows a timing diagram of TMDO and TMDO_CLK in NRZ mode. TMDO_CLK has a 50% duty cycle, therefore, the data can be sampled either at the rising or the falling edge. Before a WCO the TMDO line is LOW.

Figure 2-9. Timing Diagram for TMDO and TMDO_CLK in NRZ Mode

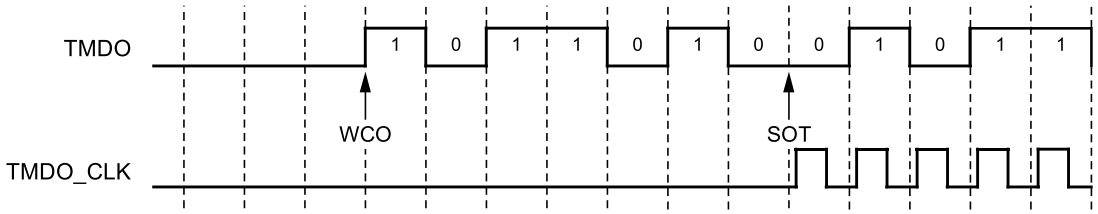
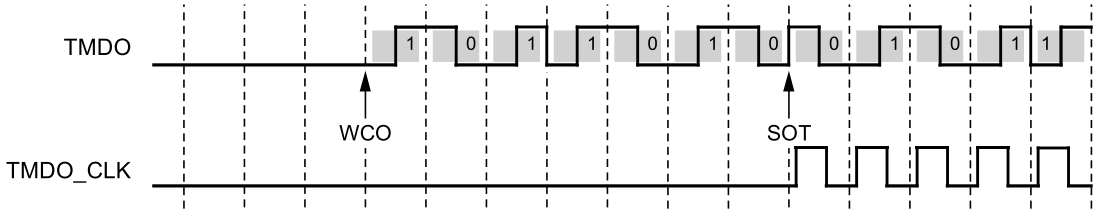


Figure 2-10 shows a timing diagram for TMDO and TMDO_CLK in Manchester mode. TMDO_CLK has a 50% duty cycle, therefore, the data can be sampled either at the rising or the falling edge.

If default polarity is selected (see Section “Data Polarity” on page 38), inverted data are received at the rising edge and correct data are received on the falling edge of TMDO_CLK. If inverted polarity is selected, correct data are received at the rising edge and inverted data are received on the falling edge of TMDO_CLK. Before WCO the TMDO line is LOW.

Figure 2-10. Timing Diagram for TMDO and TMDO_CLK in Manchester Mode



If both demodulator receive paths are activated, the signals generated on pins TMDO and TMDO_CLK come from the path which has recognized the first WCO.

Exception:
If the sramServices.rxSysSet.PathValidAfterSOT_ENA SRAM variable is set to “1”, TMDO and TMDO_CLK are provided simultaneously for the path which recognized the first SOT. A WCO detections does not provoke a TMDO output in this case.

On detection of valid data on path A and path B at the same time, path A is used for data reception.

RXMode(transparent) provides internal status information using the event flags of the event.system (R15) and event.events (R14) as shown in the general part of RXMode.

2.8.3 RXMode(buffered)

In RXMode(buffered) the received data stream is stored in the 32 bytes deep data FIFO (DFIFO). The fill level can be read out by using the “Read Fill Level Rx FIFO” SPI command.

Note: The content of the DFIFO is overwritten if the incoming data stream exceeds 32 bytes in length and has not been read out by using the “Read Rx FIFO” SPI command as described in Section 7.4.5 “Read Rx FIFO” on page 347.

In the buffered data reception mode, the current DFIFO fill level is compared to a configurable buffer threshold for path A or path B. These thresholds can be set up independently for path A and path B of each service using the sramServices.rxSetPathx.RXbufx[5:0] variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0340	rxSetPathA[0]	-	rxBufEv MaskA	RXbufA[5:0]					
0x0342	rxSetPathB[0]	-	rxBufEv MaskB	RXbufB[5:0]					

Any time the fill level reaches the threshold of the current receive path, a DFIFO fill level match condition becomes true. In that case the firmware indicates its occurrence by setting the event flag DFIFO_RX in events.system (R15).

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
System (R15) events.system	SYS_ERR	CMD_RDY	SYS_RDY	AVCCLOW	-	SFIFO	DFIFO_RX	-

Note: Every time the RXMode(buffered) starts, a reset of the DFIFO and its related pointers takes place.

Depending on the telegram requirements, the receiver can be configured to swap the incoming bytes (MSB-first or LSB-first) before storing them to the DFIFO. The associated settings for path A and path B are located in the sramServices.RXBC1 variable.

The ID check feature is supported in RXMode(buffered). This feature is described in more detail in Section 2.8.7.5 “ID Check” on page 46.

The RXMode(buffered) provides internal status information using the event flags of events.system (R15) and events.events (R14) as shown in the general part of the RXMode. In addition, the event flags DFIFO_RX and IDCHKA/B are supported, which might be used to generate an external event on pin 28 (EVENT), if the mask bits rxBufEvMaskA/B and IDCHKA/B_Mask in rxSetPathA/B[0] and rxSysEvent are enabled and the associated event occurs.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
System (R15) events.system	SYS_ERR	CMD_RDY	SYS_RDY	AVCCLOW	-	SFIFO	DFIFO_RX	-
Events (R14) events.events	IDCHKA	WCOKA	SOTA	EOTA	IDCHKB	WCOKB	SOTB	EOTB

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0340	rxSetPathA[0]	-	rxBufEv MaskA	RXbufA[5:0]					
0x0342	rxSetPathB[0]	-	rxBufEv MaskB	RXbufB[5:0]					
0x0344	rxSysEvent	IDCHKA _Mask	WCOKA _Mask	SOTA _Mask	EOTA _Mask	IDCHKB _Mask	WCOKB _Mask	SOTB _Mask	EOTB _Mask

The demodulated data can additionally be output on pin 17/TMDO and pin 19/TMDO_CLOCK as reshaped transparent output. If sramServices.rxSysSet.PathValidAfterSOT_ENA is set to “0”, the data of the active path is output on the TMDO pin after a WCO and the corresponding clock is output on pin TMDO_CLOCK after an SOT.

If sramServices.rxSysSet.PathValidAfterSOT_ENA is set to “1”, both the data and clock of the active path are output on the pins together after an SOT.

The reshaped transparent output can be activated independently for each service by setting at least one of the sramServices.RDOCR.TMDS[1:0] bits to “1”.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x033E	RDOCR	-	0	0	ETRPB	ETRPA	TMDS[1:0]		-

2.8.4 PollingMode

The Atmel® ATA5785 offers a special receive mode that includes all the features of RXMode. It is called PollingMode.

In PollingMode the receiver path is switched so that it alternates between an active period and a sleep period. In fastPollingMode, described in more detail in Section 2.8.4.3 “fastPollingMode” on page 27, the sleep time is zero. During the active period the device seeks valid telegrams on the participating service/channel configurations. The internal wake check logic verifies the incoming signal by checking the telegram for a valid wake check (WCOx) condition and then for a valid start of telegram condition (SOTx). If all participating service/channel configurations are checked and no valid telegram could be detected, the receiver re-enters the sleep period or otherwise remains active in the selected configuration.

PollingMode can be activated by using the “Set System Mode” SPI command, if the OPM bits in the systemModeConfig parameter are set to “0b11”. It also includes additional settings, such as VCO tuning during the ramp-up of the mode, IDLEMode selection and transparent mode enable. The service/channel configuration to start PollingMode is defined by startPollingIndex located in the serviceChannelConfig parameter.

Switching from PollingMode to IDLEMode during initialization of a new polling channel can produce a switching delay of ~200µs. This delay is required to guarantee correct service/channel configuration of the system. For more information, see Section 7.4.8 “Set System Mode” on page 349.

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]	
serviceChannelConfig	-	-	-	-	startPollingIndex			

In PollingMode, the data rate, channel filter bandwidth, RF modulation, frequency, and signal check severity can be set independently for each service/channel configuration and are described in more detail in Section 2.8.7 “General Rx Settings” on page 35.

2.8.4.1 Polling Service and Channel Configuration

The polling service and channel configuration defines the different service/channel combinations checked during the active polling period. Up to 16 different configurations can be set up using the SRAM address range from 0x02F8 to 0x0317.

Table 2-3. Polling Array

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F8	pollChanConf[0].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP
0x02F9	pollChanConf[0].svcChConfig	enaPathB	enaPathA	Ch[1..0]		-	Ser[2..0]		
...									
0x0316	pollChanConf[15].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP
0x0317	pollChanConf[15].svcChConfig	enaPathB	enaPathA	Ch[1..0]		-	Ser[2..0]		

Each configuration consists of two bytes as shown in Table 2-3. The first byte contains general settings, such as tuning and RF calibration, which are executed during ramp-up of the associated service/channel configuration if enabled. The first byte also includes control information which is described in more detail in Section 2.8.4.2 “Polling Cycle” on page 25. The second byte defines the service/channel configuration directly, including the path information. For more information, see Section 2.12.7 “Polling Configuration Section” on page 68. The polling array can be changed by writing to this SRAM location using an SPI command during IDLEMode.

2.8.4.2 Polling Cycle

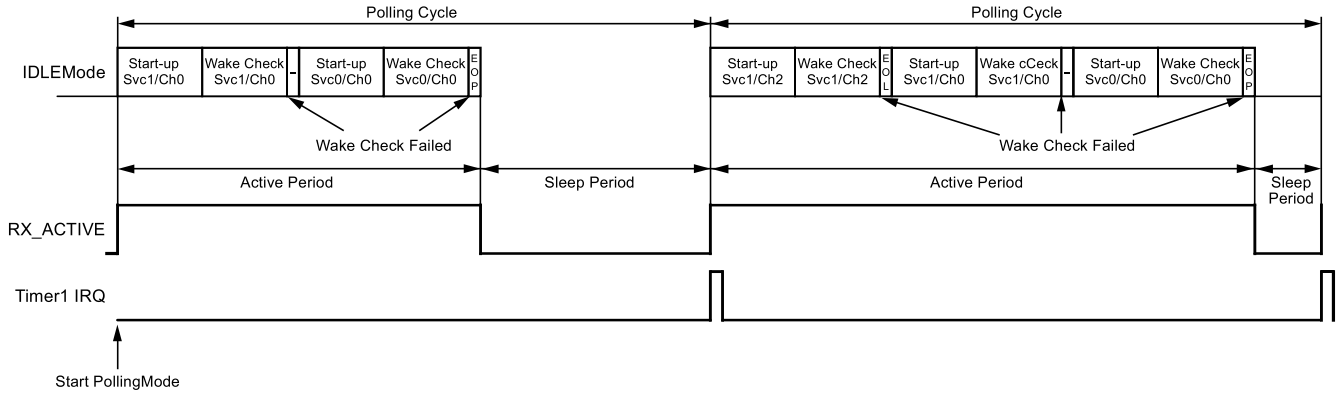
The polling cycle consists of the active polling period and the sleep period, as illustrated in Figure 2-11 on page 25. The timing interval to start the next polling cycle is programmable between 0ms and 4.000ms and controlled by Timer1.

The clock source for the polling timer is defined in an SRAM variable. To select the clock source for Timer1 as well as to adjust a proper polling cycle, the PollConfig.confT1MR, PollConfig.confT1COR, calib.srcCorVal and calib.srcRes variables must be modified (see Section 2.12.7 “Polling Configuration Section” on page 68).

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F6	confT1COR	T1COR[7:0]							
0x02F7	confT1MR	T1DC[1:0]		T1PS[3:0]				T1CS[1:0]	
0x02C4	srcRes	T1COR[7:0]							
0x02C5	srcCorVal	T1COR[7:0]							

The duration of the active polling period depends on several factors such as the number of participated service/channel configurations, telegram time-outs, RF conditions, etc. The duration may vary for each polling cycle.

Figure 2-11. Polling Cycle Definition

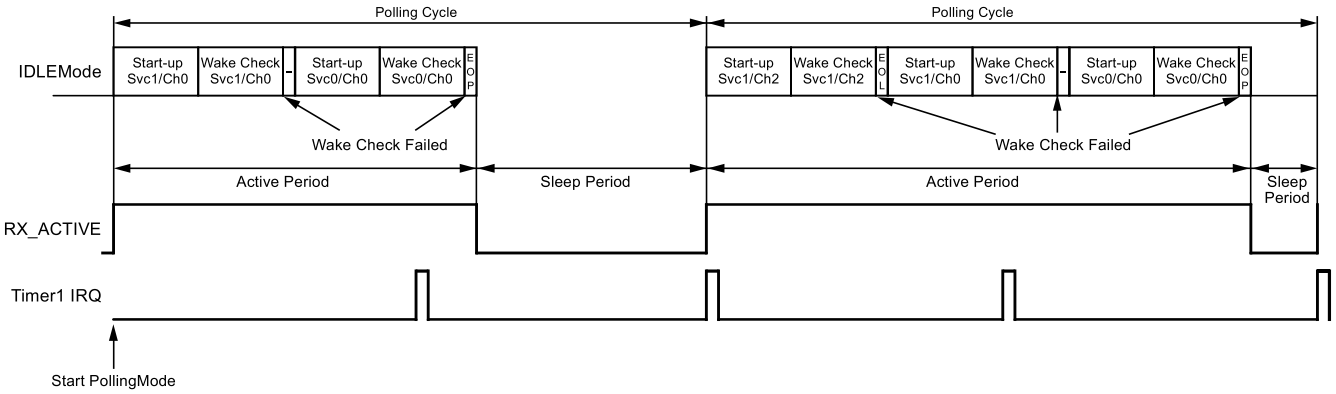


The RX_ACTIVE pin (PB7) can be configured to monitor the status of the RF front end during receive mode. The polarity of the pin can be adjusted in the SRAM setting as described in Section 2.12.1.2 “sysEventMask” on page 63.

The sleep period is primarily intended to keep current consumption at the lowest possible level. The Atmel® ATA5785 turns the receiver path off during the sleep period for this reason. Only the polling timer together with the selected oscillator remain active. Using the SRC oscillator to clock the polling timer is recommended due to its low power consumption and cyclical calibration of the polling cycle.

If the configured timing interval of the polling timer is shorter than the required time to check all the participated polling service/channel configurations, the effective polling cycle is extended until the next Timer1 IRQ, as shown in Figure 2-12.

Figure 2-12. Effective Polling Cycle Definition



Note: If the polling period is set to 0ms the fastPollingMode is activated.

End Of Polling Cycle (EOP)

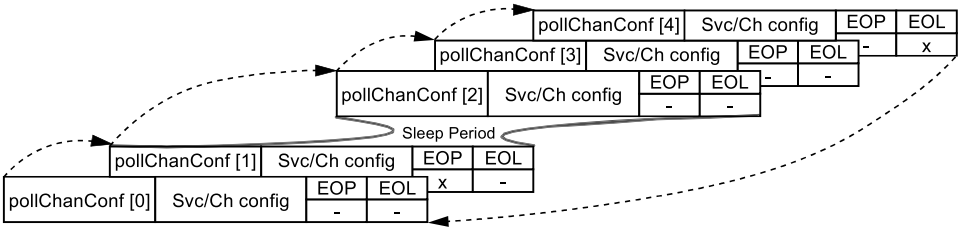
A service/channel configuration with an activated EOP label defines the last configuration in the active polling period. When detecting this label, the firmware enters the polling sleep period. Otherwise the next configuration is triggered.

Note: In fastPollingMode the EOP labels are ignored.

End Of Polling Loop (EOL)

A service/channel configuration with an activated EOL label defines the last valid configuration used for polling. All subsequent configurations are ignored. After detecting this label, the firmware triggers the first configuration of the pollChanConf array as the next applicable service/channel configuration. Figure 2-13 shows an example of a polling configuration.

Figure 2-13. Polling Configuration Example

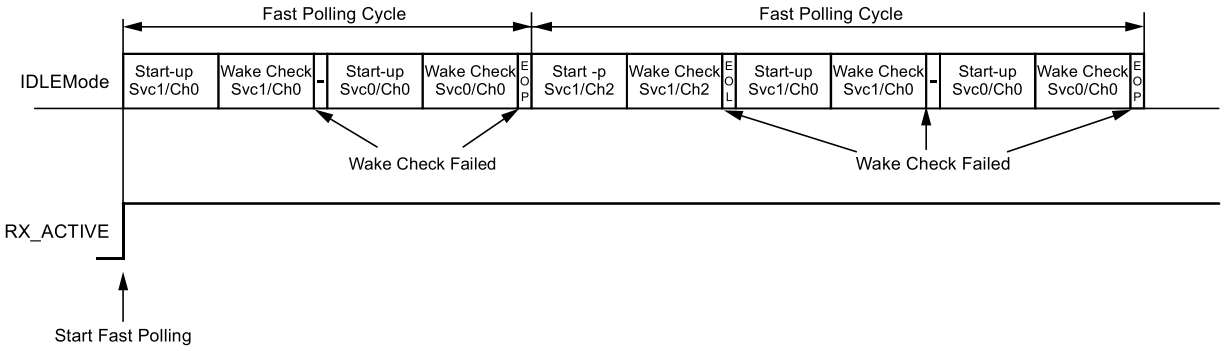


Note: If both the EOP and EOL labels are activated for the current service/channel configuration, first the polling sleep period is entered and then the next polling cycle uses the first array setting.

2.8.4.3 fastPollingMode

The purpose of fastPollingMode is to continuously check all participating service/channel configurations for valid telegrams without any interruption. This mode is very similar to the PollingMode but there is no support of the self check functionality and no sleep period is executed as shown in Figure 2-14. Fast polling is also known as “scanning”.

Figure 2-14. Fast Polling Cycle Definition



If the polling period is set to 0ms in the confT1COR SRAM setting, fastPollingMode is activated. During operation fastPollingMode can be enabled by overwriting the confT1COR SRAM variable manually with 0x00 by using the “Write SRAM/Register” SPI command if the system is in IDLEMode. For more information, see Section 7.4.6 “Write SRAM / Register” on page 348.

Note: EOP signals are ignored in this mode. Furthermore, VCO tuning on the initial service/channel configuration is not supported. For more information, see Section 2.8.4.6 “VCO Tuning” on page 28.

2.8.4.4 Self Check

During PollingMode the firmware is able to periodically start the system self check and calibration procedure. The number of polling cycles between two self check procedures can be modified via the trxCalibConfiguration.selfChk variable in the SRAM settings, as described in more detail in Section 2.12.3.5 “trxCalibConfiguration” on page 65.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E0	selfChk	SC[7:0]							

The self check functionality is always executed at the end of a polling cycle. PollingMode is interrupted and the tune and check task is triggered. After the task has finished, PollingMode resumes.

The following four options are supported and can be activated independently in the trxCalibConfiguration.calConf1 SRAM variable.

- Polling cycle calibration/slow RC calibration
- Fast RC calibration
- Service refresh

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DE	calConf1	-	-	EN_ SRCCAL	EN_ FRCCAL	-	-	-	EN_REG REFRESH

For more information about the system self check and calibration as well as the individual options, see Section 2.9 “Tune and Check” on page 50.

Notes: 1. The self check functionality is NOT supported for fastPollingMode.
2. The self check functionality is also executed once at the end of the first polling cycle after starting PollingMode.

2.8.4.5 Polling Cycle Calibration

The Atmel® ATA5785 offers a calibration procedure to ensure high accuracy of the adjusted polling cycle. The polling cycle is calibrated with the SPI command “Calibrate and Check” or periodically as a sub-task of system self check and calibration. This applies only if the clock source of Timer1 is set to SRC. The EN_SRCCAL bit in the calConf1 SRAM variable must be set in order to enable polling cycle calibration during the system self check and calibration procedure.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DE	calConf1	-	-	EN_ SRCCAL	EN_ FRCCAL	-	-	-	EN_REG REFRESH

The polling cycle calibration requires additional time. For more information about the polling cycle calibration, see Section 2.9 “Tune and Check” on page 50.

2.8.4.6 VCO Tuning

The VCO can be tuned during the PollingMode and is optionally selectable for each service/channel configuration. To enable the VCO tuning for a specific service/channel configuration, set the VCOtune bit in the associated pollChanConf[x].config SRAM configuration.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
pollChanConf[x]. config	RfCalib	-	VCOtune	-	-	-	EOL	EOP

To force VCO tuning on the first used service/channel configuration regardless of its pollChanConf setting, activate the VCO_TUNE bit in the systemModeConfig parameter of the “Set System Mode” SPI command.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
systemModeConfig	RF_CAL	-	VCO_TUNE	IDLEModeS elector	-	TMDEN	OPM[1:0]	

Table 2-4 shows the relationship between the bit VCO_TUNE and VCO tuning on the first used service/channel configuration.

Table 2-4. Activation of VCO Tuning in Polling Mode

	Bit VCO_TUNE		VCO Tuning During	
	Polling Array Configuration	SPI Command	First Polling Cycle	Second - X^{th} Polling Cycle
First Used Service/Channel Configuration	0	0	-	-
	0	1	x	-
	1	0	x	x
	1	1	x	x

Note: Activating the VCO_TUNE bit only affects the mode ramp-up behavior of the first used service/channel configuration during the first polling cycle. This feature is not supported in fastPollingMode.

VCO tuning requires additional time. For more information, see Section 2.9 “Tune and Check” on page 50.

2.8.4.7 RF Calibration

The temperature-dependent drift of the XTAL-based RF frequency can be optionally corrected for each service/channel configuration. A periodical temperature measurement by the host and update of the calib.tempMeas variable is recommended when the RF calibration feature is used. To enable the RF calibration for a specific service/channel configuration, set the RfCalib bit in the associated pollLoopConf[x].config configuration.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
pollLoopConf[x].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP

The RF calibration requires additional time. For more information about the RF calibration, see Section 2.9 “Tune and Check” on page 50.

2.8.5 Transitions to RXMode

The Atmel® ATA5785 firmware supports switching to RXMode(buffered) and RXMode(transparent) from

- IDLEMode(RC) and IDLEMode(XTO)
- RXMode (direct switch, not via IDLEMode)

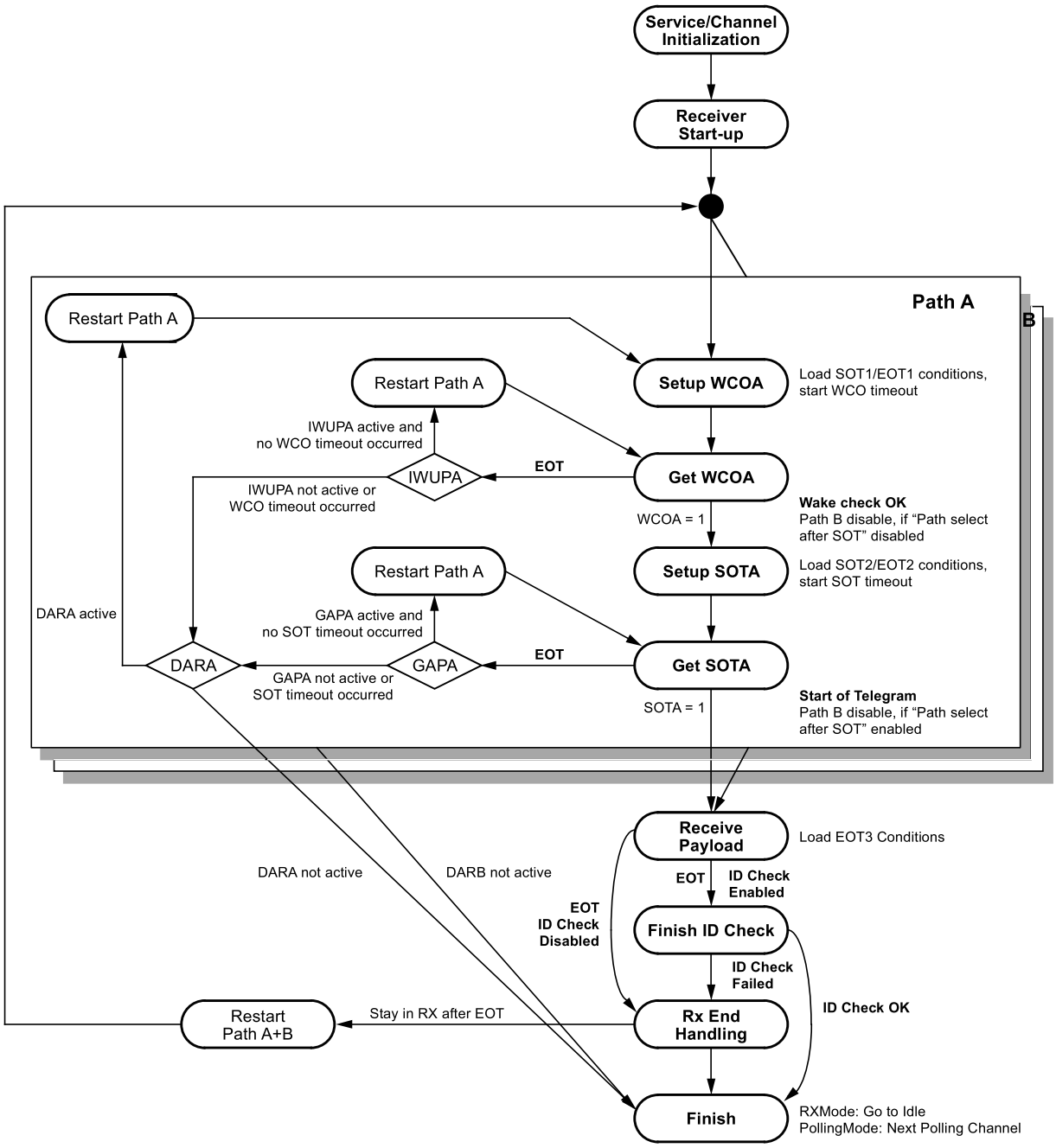
The mode switching is completely controlled by firmware. The SFIFO and DFIFO are cleared during all direct switch operations.

A transition to PollingMode is only available from IDLEMode(RC) or IDLEMode(XTO).

2.8.6 Rx Telegram Handling

The reception of a telegram is performed in several phases within the Atmel® ATA5785. After the initialization and start-up phase the receiver checks for a valid RF signal which results in a wake check OK (WCO). The next step is to search the start frame ID (SFID) pattern that marks the beginning of the payload (start of telegram, SOT). The payload reception is terminated if an end of telegram (EOT) condition becomes true. Figure 2-15 on page 30 gives an overview of the receive telegram handling process.

Figure 2-15. Receive Telegram Handling



Note: Figure 2-15 shows the logical sequence of the telegram handling and does not reflect the implementation in firmware functions.

2.8.6.1 Initialization and Start-up

The initialization and start-up phase is used to

- Configure the internal hardware register according to the selected service and channel settings
- Configure the receive state machine according to the selected receive mode
- Enable the internal modules and monitor the PLL lock and filter settling times

2.8.6.2 Wake Check OK (WCO)

After the receiver is initialized and running, the RF input is checked for a valid signal. The detection of such a signal is called wake check OK (WCO). Several check conditions can be enabled independently for path A and path B of each service in the sramServices.SOT1x variables:

- Manchester coding check (MANOEx)
- Symbol timing check (SYTOEx)
- Modulation amplitude check (AMPOEx)
- Carrier check (CAROEx)
- RSSI range check (RROEx)
- Wake-up pattern check (WUPEx)
- Wake check OK from other path (WCOxOE)

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0352	SOT1A	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x0353	SOT1B	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB

By default the Manchester, symbol timing, modulation amplitude, carrier (FSK only), and wake-up pattern checks are activated as a WCO condition. The SFID check cannot be activated as a WCO condition, but still the SFIDEx bit must always be set to “1” to avoid erroneous behavior of the internal state machine of the Atmel® ATA5785.

The functionality and configuration of the checks is described in detail in Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39.

If any of the following checks is enabled as a WCO check in the SOT1x variable, the corresponding EOT1x check (compare Section 2.8.6.5 “End of Telegram (EOT)” on page 33) must be activated: Manchester coding, symbol timing, modulation amplitude, carrier check, or RSSI range check.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0327	EOT1A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x0328	EOT1B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

A WCO is triggered as soon as all enabled checks have passed; therefore, the time period from receiver start-up until a WCO is determined by the longest check.

If a WCO is received on one path, the other path is switched off unless the sramServices.rxSysSet.PathValidAfterSOT_ENA SRAM variable is set to “1”. Then the other path is switched off not before the SOT event. If both paths receive the WCO/SOT event at the same clock cycle, path A wins.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0345	rxSysSet	-	IdScan _ENA	IFAmplifier_ ENA	PathValid AfterSOT _ENA	-	-	-	-

2.8.6.3 Start of Telegram (SOT)

A WCO indicates that a valid RF signal is present. Now the precise start of the data payload must be determined. This event is called start of telegram (SOT). The SOT is usually determined by a well-defined start frame ID (SFID) pattern. The following SOT check conditions can be enabled independently for path A and path B of each service in the sramServices.SOT2x variables:

- Manchester coding check (MANOEx)
- Symbol timing check (SYTOEx)
- Modulation amplitude check (AMPOEx)
- Carrier check (CAROEx)
- RSSI range check (RROEx)
- Wake-up pattern check (WUPEx)
- Start frame ID pattern check (SFIDEx)
- Wake check OK from other path (WCOxOE)

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0354	SOT2A	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x0355	SOT2B	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB

By default the Manchester, symbol timing, modulation amplitude, carrier (FSK only), and SFID pattern checks are activated as SOT condition.

The functionality and configuration of the checks is described in detail in Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39.

If any of the following checks is enabled as SOT check in the SOT2x variable, the corresponding EOT2x check (compare Section 2.8.6.5 “End of Telegram (EOT)” on page 33) must be activated: Manchester coding, symbol timing, modulation amplitude, carrier check, or RSSI range check.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0329	EOT2A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x032A	EOT2B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

An SOT is triggered as soon as all enabled checks have passed. A check that has already been passed for WCO is stored as successful and does not have to pass again for SOT unless there is a path restart between WCO and SOT. This is relevant if the gap mode is enabled (compare Figure 2-15 on page 30 for the occurrence of path restarts).

As a consequence, in standard applications the SFID pattern check is the only check that has to pass between WCO and SOT as all other checks have already been successful for WCO.

2.8.6.4 Receive Payload

After the occurrence of an SOT the data payload is received on the active path until an EOT event is triggered. In buffered mode the received data are written to the internal 32-byte data FIFO, in transparent mode they are routed to the PC3/TMDO and PC5/TMDO_CLOCK pins.

After an EOT the Atmel® ATA5785 goes back to IDLEMode or restarts the receiver depending on the setting of sramServices.rxSetPathx[1].RXTEHx variables. The value can be set independently for path A and path B of each service.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0341	rxSetPathA[1]	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x0343	rxSetPathB[1]	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB

2.8.6.5 End of Telegram (EOT)

End of telegram (EOT) is used as a general term for a receive error that causes a change of the current receive state. An EOT can happen in every phase of the telegram reception as shown in Figure 2-15 on page 30. Three phases with different EOT behavior can be distinguished:

- Before WCO
- Between WCO and SOT
- After SOT

EOT Before WCO

Before WCO the receiver checks for a valid RF signal. During this period the following EOT conditions can be enabled independently for path A and path B of each service in the sramServices.EOT1x variables:

- Manchester coding fail (MANFEx)
- Symbol timing fail (SYTFEx)
- Modulation amplitude fail (AMPFEx)
- Carrier fail (CARFEx)
- WCO time-out fail (TMOFEA)
- RSSI range fail (RRFEx)
- EOT from other path (EOTxFE)

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0327	EOT1A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x0328	EOT1B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

By default the Manchester, symbol timing, modulation amplitude, carrier (FSK only), and WCO time-out are activated as an EOT1 condition.

An EOT event is triggered as soon as one of the enabled checks fails. The functionality and configuration of the checks is described in detail in Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39.

The further flow depends on the intermittent WUP (sramServices.rxSetPathx.IWUPx) and demodulator auto restart (sramServices.rxSetPathx.DARx and sramServices.DMCRx.DMARx) configurations. If IWUPx is set, the path that received an EOT is restarted and continues searching for a WCO until a WCO time-out occurs. An active IWUPx setting can be useful in PollingMode if a channel has to be checked for a given time before switching to the next channel.

If IWUPx is not active or a WCO time-out occurs, the DARx/DMARx setting decides if the path (including the WCO time-out timer) is restarted or the reception is finished. The DARx/DMARx variables must be set in RXMode for standard applications. In PollingMode these settings are ignored because the function is automatically treated by the firmware.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0341	rxSetPathA[1]	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x0343	rxSetPathB[1]	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				

EOT Between WCO and SOT

Between WCO and SOT the receiver checks for the start of the data payload. During this period the following EOT conditions can be enabled independently for path A and path B of each service in the sramServices.EOT2x variables:

- Manchester coding fail (MANFEx)
- Symbol timing fail (SYTFEx)
- Modulation amplitude fail (AMPFEx)
- Carrier fail (CARFEx)
- SOT time-out fail (TMOFEA)
- RSSI range fail (RRFEx)
- EOT from other path (EOTxFE)

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0329	EOT2A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x032A	EOT2B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

The default setting is to activate the Manchester, symbol timing, modulation amplitude, carrier (FSK only), and SOT time-out as an EOT2 condition.

An EOT event is triggered as soon as one of the enabled checks fails. The functionality and configuration of the checks are described in detail in Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39.

The further flow depends on the gap mode (sramServices.rxSetPathx.GAPMx) and demodulator auto restart (sramServices.rxSetPathx.DARx and sramServices.DMCRx.DMARx) configurations. If GAPMx is set, the path that received an EOT is restarted with active WCO and continues searching for an SOT until an SOT time-out occurs. An active GAPMx setting can be useful in either RXMode or PollingMode for telegrams which have a gap with inactive carrier between the wake-up and start of frame pattern.

If GAPMx is not active or an SOT time-out occurs, the DARx/DMARx setting decides if the path (including the WCO and SOT time-out timer) is restarted or the reception is finished. The DARx and DMARx variables must be set in RXMode for standard applications. In PollingMode these settings are ignored because the function is automatically treated by the firmware.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0341	rxSetPathA[1]	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x0343	rxSetPathB[1]	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				

EOT after SOT

After SOT the actual payload data is received. During this period the following EOT conditions can be enabled independently for path A and path B of each service in the sramServices.EOT3x variables:

- Manchester coding fail (MANFEx)
- Symbol timing fail (SYTFEx)
- Modulation amplitude fail (AMPFEx)
- Carrier fail (CARFEx)
- RSSI range fail (RRFEx)
- Telegram length

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x032B	EOT3A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x032C	EOT3B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

By default the Manchester and symbol timing checks are activated as an EOT3 condition.

An EOT event is triggered as soon as one of the enabled checks fails. The functionality and configuration of the checks is described in detail in Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39.

After an EOT the ID check calculation is finished if activated. The further flow depends on the Rx telegram end handling (sramServices.rxSetPathx.RXTEHx) configuration. If the variable is set, the receiver stays in RXMode after the EOT by restarting all active paths, otherwise IDLEMode is entered. In PollingMode the RXTEHx setting is only evaluated after a telegram has been successfully received.

If a telegram is received after a successful ID check, the firmware returns to IDLEMode independent of the RXTEHx setting. This is necessary to avoid corrupt FIFO content due to subsequent receptions with wrong IDs.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0341	rxSetPathA[1]	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x0343	rxSetPathB[1]	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB

2.8.7 General Rx Settings

The receiver of the Atmel® ATA5785 implements a very flexible concept to allow a wide range of target applications. Therefore, numerous configuration settings are available for the user to optimize the system.

2.8.7.1 Channel Filter

The channel filter determines the receiver bandwidth. Its output is used for both receive paths, it therefore has to be configured to be suitable for path A and path B simultaneously. The channel filter bandwidth is set in the sramServices.CHCR.BWM[3:0] and sramServices.CHDN.BBDN[4:0] variables in the range of 25kHz to 366kHz (–3dB bandwidth).

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x031C	CHCR	-	-	-	-	BWM[3:0]			
0x031D	CHDN	-	-	ADCDN	BBDN[4:0]				

To determine the needed channel filter bandwidth, the frequency error between receiver and transmitter and the occupied bandwidth of the used modulation signal has to be taken into account. A detailed consideration of the trade-offs can be found in Section 4.1.1 “Frequency Accuracy in System Design” on page 286.

The settling time of the channel filter is highly dependent upon its bandwidth. As a result, the setting has considerable influence on the channel start-up time. A wide channel filter has a short settling time (approximately 45µs at 360kHz), a narrow filter has a long settling time (approximately 380µs at 25kHz). The corresponding settings are done for each service separately in the sramServices.CHSTARTFILTER SRAM variable.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x031E	CHSTART FILTER	-	-	PLDT	0	DFDT	FID[2:0]		

The channel filter bandwidth should always be configured by the Atmel configuration tool because it determines the baseband clock frequency which affects several other settings such as the correct data rate configuration. Subsequent manipulations in the SRAM or hardware registers are not recommended.

2.8.7.2 Demodulation Settings

RF Frequency

The Atmel® ATA5785 supports reception in the following RF frequency bands:

- Low-Band 310MHz to 318MHz
- Low-Band 418MHz to 477MHz

For each service there are three channels which can have different RF frequency settings. All channels within one service must be configured to the same frequency band. If the channels differ by more than a certain amount a VCO tuning process is required during start-up of this service/channel configuration (see Section 2.9.4 “VCO Tuning” on page 53 for more information).

The RF frequency can be configured independently for each channel of each service in the sramServices.FFREQ[2:0], sramServices.FEMS, and sramServices.FECT SRAM variables (see Section 2.12.8.2 “Channel-Specific Configuration” on page 85 for more information).

Address Ser3/Ch0	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x039A	FFREQ[0]	FFREQL[7:0]							
0x039B	FFREQ[1]	FFREQM[7:0]							
0x039C	FFREQ[2]	FFREQH[7:0]							
0x039D	FEMS	PLLM[3:0]				PLLS[3:0]			
0x039E	FECR	-	-	ANPS	PLCKG	ADHS	0	S4N3	1

RF Modulation

The Atmel ATA5785 supports the typical modulation types ASK and FSK for short-range device applications. The ASK mode is implemented as pure on-off keying (OOK), the FSK mode allows deviations from approximately ±0.375kHz to ±93kHz.

The modulation type can be configured independently for path A and path B of each service in the sramServices.DMCRx.SASKx and sramServices.rxSetPathx[1].RXMODx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				
0x0341	rxSetPathA[1]	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x0343	rxSetPathB[1]	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB

FSK Deviation

The demodulator can handle FSK deviations from approximately ±0.375kHz to ±93kHz. The actual allowed range varies depending on the configured channel filter bandwidth according to parameter no. 4.40 in Section 5. “Electrical Characteristics” on page 300. The FSK deviation can be configured independently for path A and path B of each service in the sramServices.DMCRx.DMPGx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				

Data Rate

The demodulator can handle symbol rates from 0.5Ksym/s to 160Ksym/s. The actual allowed range varies depending on the configured channel filter bandwidth and modulation type according to parameters no. 4.40 and 4.60 in Section 5. “Electrical Characteristics” on page 300. The data rate can be configured independently for path A and path B of each service in the sramServices.DMCRx.DMPGx and sramServices.DMDRx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				
0x0323	DMDRA	DMDNA[3:0]				DMAA[3:0]			
0x0324	DMDRB	DMDNB[3:0]				DMAB[3:0]			

Data Coding

The demodulator supports NRZ and Manchester decoding. General hints on Manchester decoding can be found in Section 2.1.2.1 “Telegram Structure” on page 11. NRZ decoding is supported for up to eight identical symbols in sequence before a signal transition must occur.

There are some rules to be respected for good NRZ reception:

- The telegram preamble shall not contain more than two consecutive high or low symbols. Best results are obtained with an alternating 0101.. symbol pattern as a preamble.
- A proper modulation amplitude threshold (see Section “Modulation Amplitude Check” on page 41) has to be selected to enable the correct operation of the DC removal feedback loop.
- The hold mode has to be activated for telegrams where more than two consecutive high or low symbols can occur. For more details, see Section “Hold Mode” on page 37.

The coding type can be configured independently for path A and path B of each service in the sramServices.DMMx.DMNEx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0325	DMMA	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x0326	DMMB	DMNEB	DMHB	DMPB	DMATB[4:0]				

Hold Mode

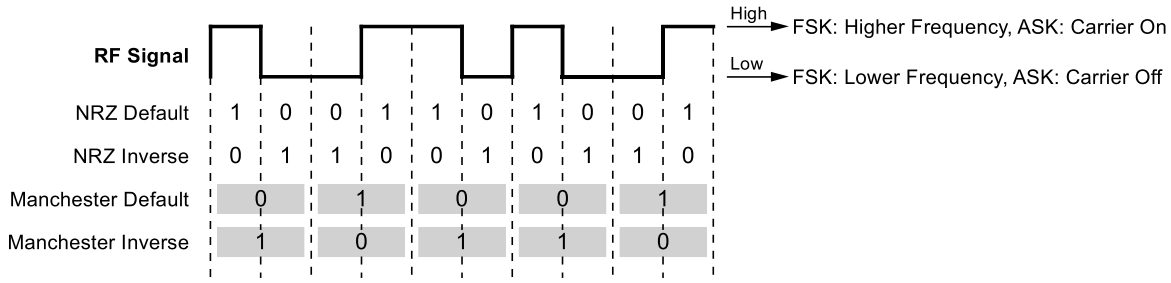
The hold mode provokes a freeze of the DC compensation if no signal transitions occur in the demodulated signal. It must be activated for NRZ coding if more than two identical consecutive symbols are possible. In Manchester mode the hold mode should be switched off. The hold mode can be activated for path A and path B of each service independently in the sramServices.DMMx.DMHx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0325	DMMA	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x0326	DMMB	DMNEB	DMHB	DMPB	DMATB[4:0]				

Data Polarity

The demodulator supports both polarities for the received data. Figure 2-16 shows the demodulated data dependent on the RF signal, the data coding setting, and the data polarity setting.

Figure 2-16. Receive Data Polarity



In NRZ mode the demodulated data correspond to the RF signal if the data polarity is set to default; otherwise the demodulated data are inverted. In Manchester mode a rising edge in the middle of the bit is interpreted as “1” by default.

If the data polarity is set to “1”, the received data are inverted before the correlation of the WUP and SFID pattern; the WUP and SFID pattern must therefore be symbol-wise inverted compared to the RF signal.

The data polarity setting has no influence on the raw data output on the TRPA and TRPB pins.

The data polarity can be configured independently for path A and path B of each service in the sramServices.DMMx.DMPx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0325	DMMA	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x0326	DMMB	DMNEB	DMHB	DMPB	DMATB[4:0]				

Data Order

The received data payload can be written byte-wise LSB-first or MSB-first to the data FIFO. The default value for the FIFO is LSB-first. The data alignment in the FIFO is visualized in Figure 2-17 in dependency to the data order setting. Incomplete bytes are filled up with zeros.

Figure 2-17. Receive Data Order

Received Data Bits

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

LSB First Alignment

Data FIFO

MSB							LSB
x	x	x	x	x	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

MSB First Alignment

Data FIFO

MSB							LSB
16	17	18	x	x	x	x	x
8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7

The data order setting is implemented in the Rx buffer and therefore has no effect on

- WUP and SFID correlation
- Transparent data output on PC3/TMDO, PC2/TRPA, or PC5/TRPB
- CRC calculation

The data order can be configured independently for path A and path B of each service in the sramServices.RXBC1.RXMSBx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0386	RXBC1	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA

Raw Transparent Output

The RXMode offers a raw signal output at the transparent path A (pin 16/TRPA) and transparent path B (pin 19/TRPB) pins for RXMode(transparent) and RXMode(buffered). The signal becomes visible as soon as the demodulator of the corresponding path is enabled, regardless of any signal and pattern checks. The demodulated signal is delivered unprocessed and without a corresponding clock.

The corresponding pins must be configured to output if this feature is used.

The raw transparent output can be activated independently for path A and path B of each service in the sramServices.RDOCR.ETRPx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x033E	RDOCR	-	0	0	ETRPB	ETRPA	TMDS[1:0]		-

2.8.7.3 Telegram Settings and Signal Checks

The Atmel® ATA5785 implements several different signal checks that can be configured as SOT and EOT conditions. These are:

- Manchester coding check (MAN)
- Symbol timing check (SYT)
- Modulation amplitude check (AMP)
- Carrier check (CAR)
- RSSI range check (RR)
- Wake-up pattern check (WUP)
- Start frame ID pattern check (SFID)
- WCO/SOT time-outs (TMO)
- WCO/EOT from other path (WCOx, EOTx)
- Telegram length check (only in buffered mode)

Signal Check Size

The signal check size defines the number of symbols that have to pass without an error before a check is considered successful. The default value for the signal check size is 12 symbols. The counter is reset with every demodulator restart. The signal check size is only valid for the following checks:

- Manchester coding check
- Symbol timing check
- Modulation amplitude check
- Carrier check

The signal check size can be configured independently for path A and path B of each service in the sramServices.SYCx.SYCSx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0358	SYCA	SYTLA[3:0]				SYCSA[3:0]			
0x0359	SYCB	SYTLB[3:0]				SYCSB[3:0]			

Note: The register value represents only half of the number of symbols to be checked.

Manchester Coding Check

The Manchester coding check verifies the Manchester coding conformity of the incoming modulation. The check ensures that there are not more than two identical consecutive symbols and that the signal edges are in the middle of the Manchester bit and not at the bit boundaries. It is one of the slower checks to respond, but it can be used for all Manchester-coded telegrams during data reception without losing sensitivity. It is particularly useful together with the symbol timing check for the end of telegram detection.

The only setting for this check is the activation of the alternating preamble feature. If enabled, a more severe mode is active until the signal check size is reached. Only alternating high and low symbols are accepted. This might reduce the amount of on-time if no valid signal is available. The feature can be configured independently for path A and path B of each service in the sramServices.DMCRx.SY1Tx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				

Note: The Manchester checker is continuously running during reception and is not reset at a WCO or an SOT event. Therefore, it should not be activated as a later EOT condition if it was not activated before. The only valid settings are listed in Table 2-5.

Table 2-5. Valid Manchester-Coding Check EOT Conditions

	EOT1	EOT2	EOT3
Manchester coding check	x	x	x
	x	x	
	x		

Symbol Timing Check

The symbol timing check is concerned with the timing of the signal transition from one symbol to the next. The edges of the symbols are expected to stay within some boundaries relative to the clock recovery in the demodulator. There are two settings for this check:

- Severity: The severity defines a validity window for the symbol edge in relation to its expected location as a percentage of symbol time. The value can be configured independently for path A and path B of each service in the sramServices.SYCx.SYTLx variables. Allowed values are 8 to 15, corresponding to an allowed symbol edge tolerance of ±25% to ±47%. The default value is 11 (34%), lower values lead to a more severe check.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0358	SYCA	SYTLA[3:0]				SYCSA[3:0]			
0x0359	SYCB	SYTLB[3:0]				SYCSB[3:0]			

- Alternating preamble: The symbol timing check generally allows high and low times of the demodulated data with once or twice the selected symbol period. If the alternating preamble feature is enabled, only alternating ones and zeros (0-1-0-1-0-... pattern) are allowed until the signal check size is reached. This represents a tougher criterion and reduces the amount of on-time if no valid signal is available. The feature can be configured independently for path A and path B of each service in the sramServices.DMCRx.SY1Tx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				

Modulation Amplitude Check

An incoming signal is expected to have a modulation. This check verifies that the modulation amplitude at the output of the data filter is above the configured threshold. The recommended threshold value is calculated by the Atmel configuration tool as it depends on the modulation type, deviation, channel filter bandwidth, and data rate. A higher value makes the check more severe. An error is flagged if the signal fails to exceed the threshold within two data symbols for an alternating preamble pattern, otherwise within three data symbols. The modulation amplitude check is useful for ASK and FSK modulation.

The modulation amplitude check has to be deactivated after WOK if the telegram data contains more than two consecutive high or low symbols.

The threshold can be configured independently for path A and path B of each service in the sramServices.DMMx.DMATx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0325	DMMA	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x0326	DMMB	DMNEB	DMHB	DMPB	DMATB[4:0]				

Carrier Check

The carrier check provides a means to verify if a valid carrier signal or only noise is present at the RF input. The check measures the phase difference at the phase comparator of the PLL for each input sample. The phase difference has to stay within ±90° to be valid. A phase jump outside of these boundaries indicates the dominance of noise or an unexpected signal. An error is flagged when the number of violations within a defined time interval exceeds a predefined threshold.

The severity of the signal check is determined by two settings:

- Window: The time window represents the length of the interval in baseband clock samples that is used to accumulate the phase jump violations. The counter is reset at the end of each time window and the counting is restarted. The default window length corresponds to a period of two data symbols. Using a shorter window length speeds up the check but this can have a negative effect on the sensitivity. The window can be configured independently for path A and path B of each service in the sramServices.DMCDx.DMCTx variables.
- Allowed fails: The number of allowed fails is used as the threshold for phase jump violations not triggering an error. The default value is around one quarter of the window size. Reducing this value makes the check more severe. The threshold can be configured independently for path A and path B of each service in the sramServices.DMCDx.DMCLx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x031F	DMCDA	DMCTA[2:0]			DMCLA[4:0]				
0x0320	DMCDB	DMCTB[2:0]			DMCLB[4:0]				

The carrier check can be used only for FSK modulation because an ASK (OOK) modulation has inherent carrier gaps generating a carrier fail indication. The carrier check is the fastest check to respond with an error and is therefore very useful for achieving low average power consumption in PollingMode. The highest power reduction can be seen at low data rates because the window can be made very short compared to the bit duration.

If a carrier check is activated for ASK modulation it's automatically switched off by the hardware.

RSSI Range Check

The RSSI range check verifies that the received signal strength is within certain limits. The check is considered successful if the measured RSSI value complies with the configured settings within an update period. See Section 2.8.7.6 "RSSI Measurement" on page 47 for more information on how to configure the RSSI measurement correctly. If the RSSI range check is used as an SOT or EOT condition, the update rate should be chosen in the range of the other active checks. The recommended default value corresponds to the length of two data symbols, the maximum recommended value depends on the signal check size.

Example: Data rate: 5Kbit/s Manchester; signal check size: six symbols

- Symbol rate: 10Kbit/s
- Symbol time: 100µs
- Two symbols take 200µs, six symbols take 600µs
- The RSSI update period should be configured between 200µs and 600µs

An error is flagged if the measured RSSI value does not match the configured settings within an update period.

Wake-Up Pattern Check

The match of a wake-up pattern can be configured as a WCO and an SOT condition. The symbol-based serial data stream is continuously scanned by a correlator for the configured pattern. The following settings are available:

- Pattern: The reference pattern consists of 32 symbols, the data direction is MSB-first. The pattern can be an arbitrary combination of "0"s and "1"s. It can be configured independently for path A and path B of each service in the sramServices.WUPx[3:0] variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x036C	WUPA[0]	WUP1A[7:0]							
0x036D	WUPA[1]	WUP2A[7:0]							
...	...	...							
0x0373	WUPB[3]	WUP4B[7:0]							

- Length: The length of the pattern can be configured to be between 1 and 32 symbols. If the length of the pattern is configured to be less than 32, the corresponding number of least significant symbols of the reference pattern is selected and the remainder disregarded. A pattern match is only possible if at least the number of symbols that is configured as the length has been received. The pattern length can be configured independently for path A and path B of each service in the sramServices.WUPLx.WUPLx[5:0] variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0374	WUPLA	-	-	WUPLA[5:0]					
0x0375	WUPLB	-	-	WUPLB[5:0]					

- Allowed fails: If the pattern is not required to match completely, the number of allowed fails can be configured. In the Atmel configuration tool any value between 0 (complete match required) and the length of the pattern (check is always successful) is allowed. Setting the allowed fails to a value >0 increases the sensitivity of the receiver but also the probability of an erroneous WCO. Setting the allowed fails to "–1" has the special meaning that a pattern match will never occur. In the SRAM the allowed fails must be configured as the threshold value independently for path A and path B of each service in the sramServices.WUPTx.WUPTx[4:0] variables. The conversion of allowed fails to the threshold value is done by the Atmel configuration tool automatically.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0376	WUPTA	-	-	-	WUPTA[4:0]				
0x0377	WUPTB	-	-	-	WUPTB[4:0]				

Start Frame ID Pattern Check

The match of a start frame ID pattern can be configured as an SOT condition. The symbol-based serial data stream is continuously scanned by a correlator for the configured pattern. The following settings are available:

- Pattern: The reference pattern consists of 32 symbols, the data direction is MSB-first. The pattern can be an arbitrary combination of “0”s and “1”s and can be configured independently for path A and path B of each service in the sramServices.SFIDx[3:0] variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0346	SFIDA[0]	SFID1A[7:0]							
0x0347	SFIDA[1]	SFID2A[7:0]							
...	...	...							
0x034D	SFIDB[3]	SFID4B[7:0]							

- Length: The length of the pattern can be configured to be between 1 and 32 symbols. If the length of the pattern is configured to be less than 32, the corresponding number of least significant symbols of the reference pattern is selected and the rest is ignored. A pattern match is only possible if at least the number of symbols that is configured as length have been received. The pattern length can be configured independently for path A and path B of each service in the sramServices.SFIDLx.SFIDLx[5:0] variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0350	SFIDLA	-	-	SFIDLA[5:0]					
0x0351	SFIDLB	-	-	SFIDLB[5:0]					

- Allowed fails: If the pattern is not required to match completely, the number of allowed fails can be configured. In the Atmel configuration tool any value between 0 (complete match required) and the length of the pattern (check is always successful) is allowed. Setting the allowed fails to a value >0 increases the sensitivity of the receiver but also the probability of an erroneous SOT. Setting the allowed fails to “–1” has the special meaning that a pattern match will never occur. In the SRAM the allowed fails must be configured as the threshold value independently for path A and path B of each service in the sramServices.SFIDCx.SFIDTx[4:0] variables. The conversion of allowed fails to the threshold value is done by the Atmel configuration tool automatically.
- Serial mode: If the serial mode is activated, a WUP match is required before an SFID pattern match is possible. The complete configured SFID pattern must be received after the WUP match and the WUP and SFID patterns are not allowed to overlap. This feature is implemented in the correlator hardware and it is independent of the activation of the pattern as an SOT1/2 condition. The mimic is reset during every path restart (compare Figure 2-15 on page 30 for the occurrence of path restarts) which requires special considerations in gap mode. It is generally recommended to activate this feature for standard applications.
  The serial mode can be activated independently for path A and path B of each service in the sramServices.SFIDCx.SEMEx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x034E	SFIDCA	SEMEA	-	-	SFIDTA[4:0]				
0x034F	SFIDCB	SEMEB	-	-	SFIDTB[4:0]				

WCO Time-out

The WCO time-out limits the period between the receive mode start-up and a WCO event. The valid configuration range of the time-out is between approximately 5µs and 300ms. The timer starts running when the demodulator is activated and stops/resets when a WCO or an EOT event occurs.

Exception: If the intermittent WUP feature is activated, the timer is not restarted at an EOT event.

If no WCO or EOT occurs until the configured time-out value has been reached, an EOT event is triggered. This occurs, for example, if a permanent RF interferer is present.

The WCO time-out period can be activated independently for path A and path B of each service in the sramServices.WCOtimeOutx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x036A	WCOtimeOutA	WCOTOA[7:0]							
0x036B	WCOtimeOutB	WCOTOB[7:0]							

SOT Time-out

The SOT time-out limits the period between a WCO and an SOT event. The valid configuration range of the time-out is between approximately 5µs and 300ms. The timer starts running at a WCO event and stops/resets when an SOT or an EOT event occurs.

Exception: If the gap mode feature is activated, the timer is not restarted at an EOT event.

If no SOT or EOT occurs until the configured time-out value, an EOT event is triggered. This occurs, for example, if a permanent RF interferer is present.

The SOT time-out period can be activated independently for path A and path B of each service in the sramServices.SOTtimeOutx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0356	SOTtimeOutA	SOTTOA[7:0]							
0x0357	SOTtimeOutB	SOTTOB[7:0]							

WCO from Other Path

A WCO event from the other path of the same service can be used as a WCO or an SOT condition. This can be useful if the preamble of a telegram is sent with ASK modulation and the data payload is sent with FSK modulation.

EOT from Other Path

An EOT event from the other path of the same service can be used as an EOT1 or an EOT2 condition.

Telegram Length

The reception of a certain number of bits can be used as an EOT3 condition. This feature is only available in buffered receive mode and not in transparent receive mode. All bits that are received after the SOT event and written to the data FIFO are counted. An EOT is triggered as soon as the configured number of bits has been received. This means a defined code violation at the end of the telegram is not required. Any value between 1 and 4095 is allowed. If the corresponding variables are written to “0”, the check is switched off.

The telegram length can be configured independently for path A and path B of each service in the sramServices.RXTLx[1:0] variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x037D	RXTLA[0]	RXTLLA[7:0]							
0x037E	RXTLA[1]	-	-	-	-	RXTLHA[3:0]			
0x0384	RXTLB[0]	RXTLLB[7:0]							
0x0385	RXTLB[1]	-	-	-	-	RXTLHB[3:0]			

2.8.7.4 Receive CRC Checker

The Atmel® ATA5785 implements a highly configurable hardware CRC checker for the received data payload. The CRC checker can be activated independently for path A and path B of each service in the sramServices.RXBC1.RXCEx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0386	RXBC1	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA

If enabled, all bits that are received after the SOT event are transferred to the CRC checker with the option to skip up to 255 bits at the beginning.

The success of the CRC check is stored in the TESRA.CRCOA hardware register bit for path A and TESRB.CRCOB for path B until the next receiver start-up or path restart. The bits can be read with the “Read SRAM Register” SPI command from address 0x096 and 0x095 for path A and path B respectively. If CRCOx is “1”, the CRC check was successful for the corresponding path.

The usage of the receive CRC checker is bound to the following conditions:

- The CRC checker is available only in buffered receive mode and not in transparent receive mode
- The data payload of the telegram must be concluded by a well defined code violation to keep arbitrary bits at the end of the telegram from disturbing the checksum. Ideally the telegram length EOT condition (see Section “Telegram Length” on page 44) is used to define the number of received data bits.

The CRC checker cannot be used if the stay in Rx after EOT option is set (sramServices.rxSetPathx[1].RXTEHx=”1”) because the checksum is reset at every receiver start-up phase and every path restart (compare Figure 2-15 on page 30).

CRC Length

The CRC checker can be configured to a 4-bit, 8-bit or 16-bit polynomial. The CRC length can be set independently for path A and path B of each service in the sramServices.RXBC1.RXCBLx[1:0] variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0386	RXBC1	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA

CRC Polynomial

The coefficients of the CRC polynomial can be independently configured for path A and path B of each service in the sramServices.RXCPx[0].RXCPLx (low byte) and sramServices.RXCPx[1].RXCPHx (high byte) variables. The LSB corresponds to the X^0 coefficient and is always set to “1” by the hardware regardless of the SRAM setting.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0378	RXCPA[0]	RXCPLA[7:0]							
0x0379	RXCPA[1]	RXCPHA[7:0]							
0x037F	RXCPB[0]	RXCPLB[7:0]							
0x0380	RXCPB[1]	RXCPHB[7:0]							

If the CRC length is set to 4 or 8, only the corresponding number of LSBs are relevant.

CRC Init Value

The CRC checksum can be preloaded with an initialization value that can be independently configured for path A and path B of each service in the sramServices.RXCIx[0].RXCILx (low byte) and sramServices.RXCIx[1].RXCIHx (high byte) variables. The orientation of the checksum is always MSB-first.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x037A	RXCIA[0]	RXCILA[7:0]							
0x037B	RXCIA[1]	RXCIHA[7:0]							
0x0381	RXCIB[0]	RXCILB[7:0]							
0x0382	RXCIB[1]	RXCIHB[7:0]							

If the CRC length is set to 4 or 8, only the corresponding number of LSBs is relevant.

CRC Skip Bits

Up to 255 bits at the beginning of the data payload can be omitted by the CRC checker. The number can be configured independently for path A and path B of each service in the sramServices.RXCSBx variables.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x037C	RXCSBA	RXCSBA[7:0]							
0x0383	RXCSBB	RXCSBB[7:0]							

2.8.7.5 ID Check

The ID check automatically searches for user-defined ID-patterns at the beginning of the data payload. If none of the defined ID-patterns is found, an EOT is triggered. By using this feature the reception of invalid telegrams can be aborted before they are completely received and transferred to the host controller for a validity check. This can reduce overall system power consumption by shortening the active polling period and increasing the host controller sleep period.

The ID check works only in combination with RXMode(buffered) or PollingMode(buffered). The receiver compares the ID taken from the received data stream with the predefined IDs in the SRAM. Up to 18 IDs can be stored and each ID can be up to a maximum length of 4 bytes. The number of IDs to be checked as well as the ID length and position within the data stream is configurable via the SRAM settings (for more information, see TBD).

The sramID.idEna variable enables the ID scan in general if its value is different from “0”. The bits ID[4:0] define the number of IDs to be used in the system. All values greater than 18 are invalid.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x046C	idEna	-	-	-	ID[4:0]				

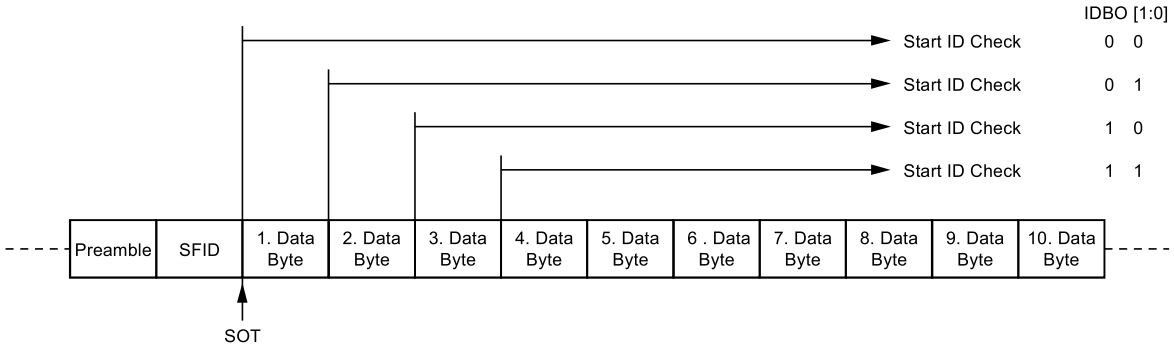
The 18 ID settings occupy the SRAM addresses 0x046D to 0x04C6. A single ID setting consists of a four-byte data field and an associated configuration byte.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
data[x][0]	data_byte0[7:0]							
data[x][1]	data_byte1[7:0]							
data[x][2]	data_byte2[7:0]							
data[x][3]	data_byte3[7:0]							
config[x]	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	

The data[x][0] – data[x][3] (x = 0 to 17) fields contain the ID for scanning the received telegram. Note that data[x][0] is always used first to get a compare match.

The configuration byte config[x] (x = 0 to 17) enables the stored ID setting via the IDCE bit. If the bit is set to “1”, this setting is part of the ID scanning—otherwise it is omitted. The IDBO[1:0] variable defines the start position of the ID within the data stream as illustrated in Figure 2-18. The length of the current ID can be set by the IDL[1:0] variable.

Figure 2-18. Possible Position of the ID in a Data Stream



The ID check can be activated independently for each service in the sramServices.rxSysSet.IDScan_ENA variable.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0345	rxSysSet	-	IdScan _ENA	IFAmplifier_ ENA	PathValid AfterSOT _ENA	-	-	-	-

The ID check is started as soon as seven payload bytes have been received or an EOT is received after an SOT.

If a telegram is received after a successful ID check the firmware always returns to IDLEMode, even if “Stay in RX after EOT” is activated (rxSetPathx[1].RXTEHx=“1”).

The check of one ID takes approximately 11µs which sums up to about 200µs if all 18 IDs are activated. For more details about the ID check hardware refer to Section 3.4.3.5 “ID Check” on page 136.

2.8.7.6 RSSI Measurement

The RSSI measurement feature determines and provides a snapshot of the current RF signal strength. Internal gain compensation automatically removes the IF amplifier from the RSSI value. The result is placed in the 16-byte-wide support FIFO (SFIFO). The fill level and the content of the SFIFO can be read out using the “Read Fill Level RSSI FIFO” and “Read RSSI FIFO” SPI commands, respectively (see Section 7.4 “SPI Command Reference” on page 345 for details).

Note: Failing to read out via SPI when more than 16 sampled RSSI values are stored results in the oldest values being permanently deleted.

The signal power at the matched 50Ω antenna input can be calculated in dBm:

$$\mathrm{RF\_Power_{IN}} = \frac{\mathrm{RSSI}}{2} - \mathrm{RSSI\_OFFSET} \qquad (1)$$

with RSSI_OFFSET = 135dBm

The RSSI measurement can be activated independently for each service in the sramServices.rssiSysConf.RssiEnable variable.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x033F	rssiSysConf	RssiEnable	-	rssiBufEv Mask	RSSIbuf[4:0]				

A new RSSI value is stored in the SFIFO at the end of every RSSI update period. The length of this period can be configured independently for each service in the sramServices.RSSC.RSUP[3:0] variable. The Atmel configuration tool calculates the actual time period based on the configured value. The formula is given in the hardware description of the RSSI buffer (see Section 3.4.3.6 “RSSI Buffer” on page 139).

Depending on the sramServices.RSSC.RSPKF configuration, either the peak or the average value of the last update period is provided.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0399	RSSC	-	RSPKF	0	RSWLH	RSUP[3:0]			

During the RSSI measurement the current SFIFO fill level is compared to a configurable buffer threshold. This threshold can be set up independently for each service using the sramServices.rssiSysConf.RSSIbuf[4:0] variable.

Any time the fill level reaches the threshold of the current receive path, an SFIFO fill level match condition becomes true. In that case the firmware signals the occurrence by setting the SFIFO event flag in the events.system (R15).

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
System (R15) events.system	SYS_ERR	CMD_RDY	SYS_RDY	AVCCLOW	-	SFIFO	DFIFO_RX	-

In addition, the generation of an external event on pin 28 (EVENT) is supported if the rssiBufEvMask mask bit is enabled and the associated event occurs.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0238	rssiSysConf	RssiEnable	-	rssiBufEv Mask	RSSIbuf[4:0]				

The determined RSSI value can also be used as an SOT and EOT condition (compare Section “RSSI Range Check” on page 42).

An RSSI window must be defined and the corners (thresholds) of this window must be configured in the associated rssiThresholds[x][y].RSSH/L SRAM variables. The RSSI window consists of a lower and an upper threshold and is available for each channel in each service.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0448	rssiThresholds[3][0].RSSL	RSSL for service 3 and channel 0							
0x0449	rssiThresholds[3][0].RSSH	RSSH for service 3 and channel 0							
0x044A	rssiThresholds[3][1].RSSL	RSSL for service 3 and channel 1							
0x044B	rssiThresholds[3][1].RSSH	RSSH for service 3 and channel 1							
0x044C	rssiThresholds[3][2].RSSL	RSSL for service 3 and channel 2							
0x044D	rssiThresholds[3][2].RSSH	RSSH for service 3 and channel 2							
0x044E	rssiThresholds[4][0].RSSL	RSSL for service 4 and channel 0							
0x044F	rssiThresholds[4][0].RSSH	RSSH for service 4 and channel 0							
0x0450	rssiThresholds[4][1].RSSL	RSSL for service 4 and channel 1							
0x0451	rssiThresholds[4][1].RSSH	RSSH for service 4 and channel 1							
0x0452	rssiThresholds[4][2].RSSL	RSSL for service 4 and channel 2							
0x0453	rssiThresholds[4][2].RSSH	RSSH for service 4 and channel 2							

The sramServices.RSSC.RSWLH variable defines whether the measured signal strength must be inside or outside of the defined RSSI window for a successful check.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0399	RSSC	-	RSPKF	RSHRX	RSWLH	RSUP[3:0]			

Note: The lower and upper thresholds are initialized to 0x00 and 0xFF during system start-up.

The RSSI measurement can also be activated on demand by using the “Start RSSI Measurement” SPI command. Afterwards the resulting RSSI value can be obtained by using the “Get RSSI Value” SPI command.

2.8.7.7 VCO Tuning

The VCO can be optionally tuned for each service/channel configuration during the RXMode ramp-up. To enable VCO tuning for the selected service/channel configuration, set the VCO_TUNE bit in the systemModeConfig parameter of the “Set System Mode” SPI command.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
systemModeConfig	RF_CAL	-	VCO_TUNE	IDLEModeSelector	-	TMDEN	OPM[1:0]	

VCO tuning requires additional time. For more information about the VCO tuning, see Section 2.9 “Tune and Check” on page 50.

2.8.7.8 RF Calibration

The temperature-dependent drift of the XTAL-based RF frequency can be optionally corrected for each service/channel configuration. To enable the RF calibration for the selected service/channel configuration, set the RF_CAL bit in the systemModeConfig parameter of the “Set System Mode” SPI command.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
systemModeConfig	RF_CAL	-	VCO_TUNE	IDLEModeSelector	-	TMDEN	OPM[1:0]	

The current temperature value must be written to the calib.tempMeas SRAM variable by the host before an RF calibration is started. RF calibration requires additional time. For more information, see Section 2.9 “Tune and Check” on page 50.

2.8.7.9 External LNA Control

Depending on the system requirements an increased sensitivity of the receiving path might be necessary. To boost the sensitivity of the receiver, an external low-noise amplifier can be used. The Atmel® ATA5785 offers an output pin (pin 29 / RX_ACTIVE) which can be used to bias the external LNA. The activation and polarity configuration of the RX_ACTIVE pin can be done in the sysEventMask.RX_ACTIVE_EN and RX_ACTIVE_POL SRAM variables. Application-specific details can be found in Section 4.1.4 “External LNA Support” on page 293.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020E	sysEventMask	SYS_ERR	-	SYS_RDY	AVCCLOW	-	RX_ACTIVE_EN	RX_ACTIVE_POL	IRQ_POL

2.9 Tune and Check

This section provides an overview of the calibration and tuning capabilities of the Atmel® ATA5785 to achieve the best system performance regarding fabrication tolerances as well as temperature drift and voltage drift. Some of the tune and check features influence the behavior of certain operating modes only, while others can be used across the modes. The following tune and check features are available in the Atmel ATA5785:

- Polling cycle/SRC calibration
- FRC calibration
- VCO tuning
- System self check and calibration
- Register refreshing
- RF calibration

A tune and check feature can be executed under the following operating conditions:

- During start-up of a receive mode
- During the self check operation in PollingMode
- During execution of the TCMode activated by the “Calibrate and Check” SPI command

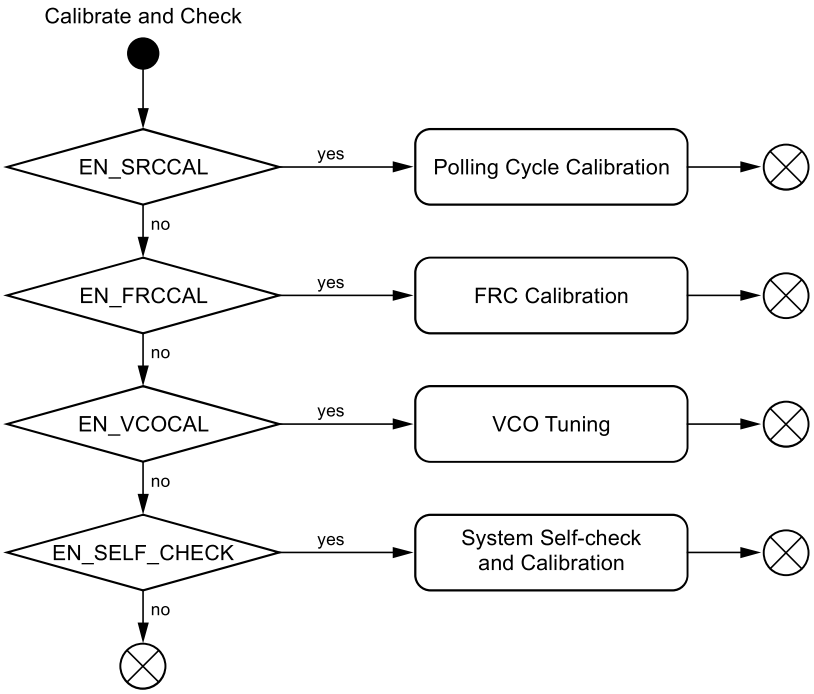
2.9.1 TCMode

The tune and check mode (TCMode) can be activated by the user using a “Calibrate and Check” SPI command (see Section 7.4.9 “Calibrate and Check” on page 353). The system must be operated in IDLEMode for TCMode to be entered via SPI command, otherwise the SYS_ERR event flag in the events.system event byte (R15) is set and the IDLEMode started. In addition, an external event on pin PB6 is generated if the sysEventMask.SYS_ERR variable is set.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020E	sysEventConf	SYS_ERR	-	SYS_RDY	AVCCLOW	-	RX_ACTIVE_EN	RX_ACTIVE_POL	IRQ_POL

Figure 2-19 on page 51 illustrates the supplied TCMode tasks and shows the execution dependencies. Each task can be enabled/disabled separately and is described in more detail in the relevant sections.

Figure 2-19. TCMode Flow Chart



Note: Only one tune and check task can be executed per “Calibrate and Check” command. See the sections below for more information on the various tune and check features.

2.9.2 Polling Cycle/SRC Calibration

Generally, the tolerance of the slow RC (SRC) oscillator is approximately ±10% over temperature drift and voltage range (see parameter no. 14.20 in Section 5. “Electrical Characteristics” on page 300). The polling cycle calibration procedure is intended to ensure a high accuracy of ±2% of the configured polling cycle when the slow RC (SRC) oscillator is used as clock source for Timer1.

During the firmware-driven calibration process the frequency deviation of the SRC oscillator is determined in relation to its nominal frequency of 125kHz. Timer2 and Timer3 with the XTO as reference clock are used to determine a deviation value which serves as the basis for calculating a proper polling cycle calibration value. This value is stored to the calib.srcCorVal SRAM variable.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02C5	srcCorVal	Correction value for T1COR							

The Timer1 Compare Register (T1COR) is corrected based on the determined calibration value, resulting in polling cycle accuracy of ±2% (see parameter no. 14.20 in Section 5. “Electrical Characteristics” on page 300).

A polling cycle calibration process can be triggered at the following time points:

- During the periodic self check in PollingMode or during a self check that is started by using the “Calibrate and Check” SPI command where this is enabled in trxCalibConfiguration.calConf1.EN_SRCCAL.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DE	calConf1	-	-	EN_ SRCCAL	EN_ FRCCAL	-	-	-	EN_REG REFRESH

- From IDLEMode via the “Calibrate and Check” SPI command if the EN_SRCCAL bit in tuneCheckConfig is set to “1” (see Section 7.4.9 “Calibrate and Check” on page 353).

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	-

Setting the SRC_CAL bit within the events.cmdRdyConf command-ready variable to “1” raises a command-ready (CMD_RDY) event on the event pin PB6 when the polling cycle calibration process has finished.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0207	cmdRdyConf	-	-	SRC_CAL	FRC_CAL	VCO_CAL	RF_CAL	SELFCHECK	-

The polling cycle calibration requires additional time, see Section 6. “Timing Characteristics” on page 310.

2.9.3 FRC Calibration

Generally, the tolerance of the fast RC (FRC) oscillator is ±5% over temperature drift and voltage range (see parameter no. 14.30 in Section 5. “Electrical Characteristics” on page 300).

Tolerance of ±2% is required to be able to use the FRC oscillator as a timer/system clock source.

The firmware-driven calibration routine of the FRC is based on the XTO frequency. The frcCalibGate setting located in SRAM defines an XTO-based gate.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E1	frcCalibGate	data[7:0]							

The gate time is calculated so that it always results in a length of 200 periods in relation to a nominal clock frequency of 6.36MHz. During the open gate time, the number of FRC pulses is counted. The result of this counting process serves as the basis for calculating the new FRC oscillator calibration bits of the FRCCAL hardware register as described in Section 3.8.4.4 “System Clock Register Description” on page 205. This register has a direct influence on the FRC frequency.

Note: The FRC is only tested in the production test for FRCCAL values required for achieving 6.36MHz. No other frequencies are used.

An FRC calibration process can be triggered at the following time points:

- During the periodic self check in PollingMode or during a self check that is started by using the “Calibrate and Check” SPI command if enabled in trxCalibConfiguration.calConf1.EN_FRCCAL.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DE	calConf1	-	-	EN_ SRCCAL	EN_ FRCCAL	-	-	-	EN_REG REFRESH

- From IDLEMode via the “Calibrate and Check” SPI command if the EN_FRCCAL bit in tuneCheckConfig is set to “1” (see Section 7.4.9 “Calibrate and Check” on page 353).

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	-

Setting the FRC_CAL bit within the events.cmdRdyConf command-ready variable to “1” raises a command-ready (CMD_RDY) event on the event pin PB6 after the FRC calibration process has finished.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020F	cmdRdyConf	-	-	SRC_CAL	FRC_CAL	VCO_CAL	RF_CAL	SELF CHECK	-

The FRC calibration requires additional time, see Section 6. “Timing Characteristics” on page 310.

2.9.4 VCO Tuning

The voltage-controlled oscillator (VCO) tuning process adjusts the digital frequency setting of the VCO before it is used in the analog PLL loop (see Section 3.3.3 “VCO” on page 93). If the VCO is not tuned, the analog control voltage may be outside the operating range of the charge pump and the fractional-N-PLL does not lock. For this reason the Atmel® ATA5785 does not work reliably without VCO tuning.

During operation a VCO tuning should be executed if the temperature changes by more than 40K or if the VCO frequency changes by more than 25MHz. This change in VCO frequency corresponds to the following changes in the RF frequency:

- Low-Band 310MHz to 318MHz: 3MHz change
- Low-Band 418MHz to 477MHz: 6MHz change

It is generally recommended to activate VCO tuning during every receive service start-up.

As long as no VCO tuning process has been performed after power-on, an initialization value is used from the SRAM. The value can be configured independently for each service in the sramServices.FEVCT.FEVCT[3:0] variables. The default value is 7.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0331	FEVCT	-	-	-	-	FEVCT[3:0]			

A VCO tuning process can be triggered at the following time points:

- During start-up of a receive mode via the “Set System Mode” SPI command (see Section 7.4.8 “Set System Mode” on page 349).

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
systemModeConfig	RF_CAL	-	VCO_TUNE	IDLEModeSelector	-	TMDEN	OPM[1:0]	

- In PollingMode for each service/channel configuration (see Section 2.8.4.6 “VCO Tuning” on page 28 for additional information).

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
pollLoopConf[x].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP

- From IDLEMode via the “Calibrate and Check” SPI command if the EN_VCOCAL bit in tuneCheckConfig is set to “1” (see Section 7.4.9 “Calibrate and Check” on page 353).

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_SELFCHECK	-

Setting the VCO_CAL bit within the events.cmdRdyConf command-ready variable to “1” raises a command-ready (CMD_RDY) event on the event pin PB6 when the VCO tuning process has finished.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020F	cmdRdyConf	-	-	SRC_CAL	FRC_CAL	VCO_CAL	RF_CAL	SELF CHECK	-

VCO tuning requires additional time, see Section 6. “Timing Characteristics” on page 310.

2.9.5 RF Calibration

The temperature-dependent drift of the XTAL-based RF frequency can be corrected for the data receive modes. There are 23 bytes reserved for this in the temperatureCalibration SRAM setting where frequency correction values can be stored in 8K increments (see Section 2.12.3.4 “temperatureCalibration” on page 65 for details). The required data should be provided by the XTAL supplier.

A frequency correction value is calculated together with the service-related sramServices.TMUL multiplier value based on the result of the latest temperature value. This value must be provided by the external host and written to the calib.tempMeas variable via SPI. It is then used by the firmware to correct the FFREQ1/2 settings that determine the output frequency of the fractional-N PLL.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02C7	temperatureCalibration[0]	tempCal_0[7:0]							
...									
0x02DD	temperatureCalibration[22]	tempCal_22[7:0]							

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00FB	TMUL	TMUL[7:0]							

An RF calibration process can be triggered at the following time points:

- During start-up of a receive mode via the “Set System Mode” SPI command (see Section 7.4.8 “Set System Mode” on page 349).

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
systemModeConfig	RF_CAL	-	VCO_TUNE	IDLEModeSelector	-	TMDEN	OPM[1:0]	

- In PollingMode for each service/channel configuration. For more details, see Section 2.8.4.7 “RF Calibration” on page 29.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
pollLoopConf[x].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP

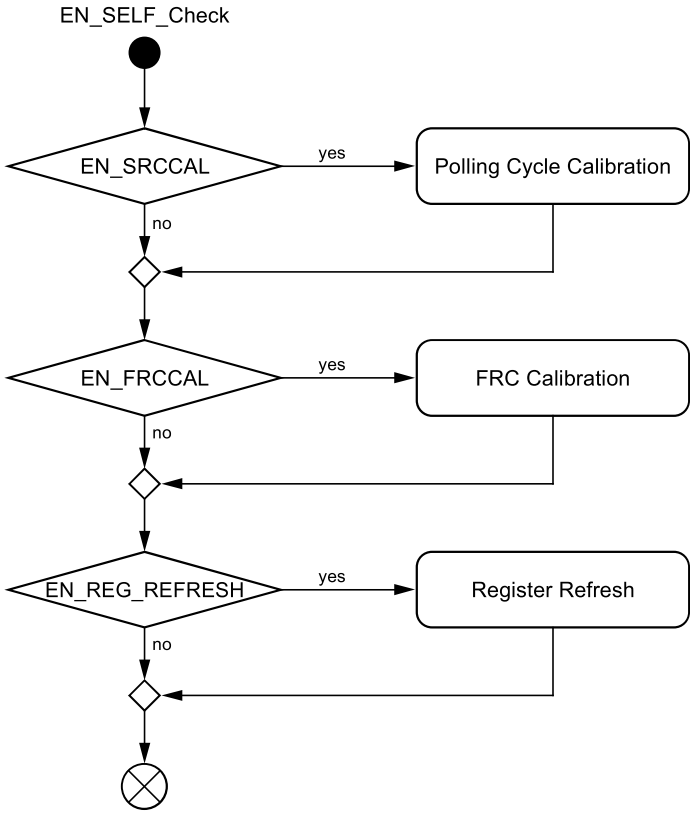
Note: No SPI command supports the RF calibration. It is only available in combination with the start of a data receive mode.

The RF calibration requires additional time, see Section 6. “Timing Characteristics” on page 310.

2.9.6 System Self Check and Calibration

During the system self check and calibration procedure up to four tune and check tasks can be executed in sequence, as shown in Figure 2-20.

Figure 2-20. System Self Check and Calibration Flow



Each task can be separately deactivated or activated by manipulating the associated bit in the trxCalibConfiguration.calConf1 setting.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DE	calConf1	-	-	EN_ SRCCAL	EN_ FRCCAL	-	-	-	EN_REG REFRESH

A system self check and calibration process can be triggered at the following time points:

- During PollingMode at regular intervals if the trxCalibConfiguration.selfChk.SC variable is unequal to “0x00” (see Section 2.8.4.4 “Self Check” on page 27).

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E0	selfChk	SC[7:0]							

- From IDLEMode via the “Calibrate and Check” SPI command if the EN_SELFCHECK bit in tuneCheckConfig is set to “1” (see Section 7.4.9 “Calibrate and Check” on page 353).

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	-

See the relevant sections for more information about the supported tasks.

Setting the SELFCHECK bit to “1” within the events.cmdRdyConf command-ready variable raises a command-ready (CMD_RDY) event on the PB6 event pin when the system self check and calibration process has finished.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020F	cmdRdyConf	-	-	SRC_CAL	FRC_CAL	VCO_CAL	RF_CAL	SELF CHECK	-

2.9.7 Register Refreshing

A register refreshing process can be used to force the reload of the complete service/channel configuration of the SRAM currentService structure using its default SRAM configuration the next time a data receive mode ramps up. This can be useful in these circumstances:

- In a polling scenario where only one service with different channels is used and the user wants a periodic refresh of the currentService data in the SRAM with the data from the SRAM service configuration
- If an SRAM service configuration is updated via SPI and the user wants to transfer these changes to the currentService structure at the subsequent start-up of a receive mode
- If the currentService was manipulated by the user and the default SRAM configuration needs to be reloaded

Note: If a receive mode is started with a service that differs from the service in the currentService structure, the complete service/channel configuration in currentService is updated in any case (see Section 2.1.1 “Service-based Concept” on page 10).

A register refreshing process can be triggered at the following time points:

- During the periodic self check in PollingMode or during a self check that is started by using the “Calibrate and Check” SPI command if enabled in trxCalibConfiguration.calConf1.EN_REGREFRESH.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DE	calConf1	-	-	EN_ SRCCAL	EN_ FRCCAL	-	-	-	EN_REG REFRESH

Note: No SPI command directly supports a register refresh. Refreshing the register is only available in combination with the system self check and calibration procedure.

The register refresh procedure requires additional time, see Section 6. “Timing Characteristics” on page 310.

2.10 Event Handling

This section describes the event handling used in the Atmel® ATA5785. The event handling is used to provide internal status information to an external host controller. For a flexible usage of this mechanism a configurable trigger signal is implemented on pin 28 (EVENT). The polarity of this EVENT pin is LOW-active by default after system start-up but can be changed to HIGH-active in the events.sysEventMask.IRQ_POL variable. However the first system-ready event after start-up from OFFMode will always be LOW-active.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020E	sysEventMask	SYS_ERR	-	SYS_RDY	AVCCLOW	-	RX_ACTIVE_EN	RX_ACTIVE_POL	IRQ_POL

The EVENT pin is set if an internal event occurs, and the corresponding event flag mask is set. The EVENT pin is cleared as soon as the event flags are read out by the “Get Event Bytes” SPI command.

2.10.1 Event Flags

The event flags are located in the R12 to R15 AVR® registers and can be configured in a flexible way. An event flag is always set when the corresponding event occurs. The external EVENT pin is only set, if the flag is configured (masked), see Section 2.10.2 “Event Configuration” on page 59. The event flags can be read out via the “Get Event Bytes” SPI command (see Section 7.4 “SPI Command Reference” on page 345). After a “Get Event Bytes” command, the content of the R13 to R15 registers is cleared. The R12 register (events.config) is not cleared because it contains the information of the currently loaded service and channel configuration in the currentService variable. For more information about SRAM variables and data structures, see Section 7.3 “SRAM Map” on page 325.

Table 2-6 shows an overview of the available event flags.

Table 2-6. Event Flags Overview

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
System (R15) events.system	SYS_ERR	CMD_RDY	SYS_RDY	AVCCLOW	-	SFIFO	DFIFO_RX	-
Events (R14) events.events	IDCHKA	WCOKA	SOTA	EOTA	IDCHKB	WCOKB	SOTB	EOTB
Power (R13) events.power	PWRON	-	NPWRON6	NPWRON5	NPWRON4	NPWRON3	NPWRON2	NPWRON1
Config (R12) events.config	PathB	PathA	ch[1:0]		-	ser[2:0]		

2.10.1.1 “System” Event Byte

The “system” event byte is located in the R15 AVR register.

Table 2-7. “System” Event Overview

Event	Description
SYS_ERR	This flag is set if the firmware detects an error or inconsistency. If the SYS_ERR flag is raised, an error code is stored in the SRAM variable debug. The error codes are listed in Section 7.5 “Error Codes” on page 359.
CMD_RDY	This flag is set every time a tune and check function has finished.
SYS_RDY	This flag is set after system initialization has completed.
AVCCLOW	This flag is set if an AVCCLOW interrupt occurs.
SFIFO	This flag is set in RXMode/PollingMode if the SFIFO is filled with the number of bytes configured in sramServices.rssiSysConf.RSSIbuf.
DFIFO_RX	This flag is set in RXMode if the DFIFO is filled with the number of bytes configured in sramServices.rxSetPathA[0].RxBufA for path A reception. sramServices.rxSetPathB[0].RxBufB for path B reception.

2.10.1.2“Events” Event Byte

The “events” event byte is located in the R14 AVR® register.

Table 2-8. Event “events” Overview

Event	Description
IDCHKA	This flag is set if the ID check was successful during reception on path A.
WCOKA	This flag is set if a wake check OK event was detected on path A.
SOTA	This flag is set if a start of telegram event was detected on path A.
EOTA	This flag is set in RXMode/PollingMode if an end of telegram event was detected on path A.
IDCHKB	This flag is set if the ID check was successful during reception on path B.
WCOKB	This flag is set if a wake check OK event was detected on path B.
SOTB	This flag is set if a start of telegram event was detected on path B.
EOTB	This flag is set in RXMode/PollingMode if an end of telegram event was detected on path B.

2.10.1.3“Power” Event Byte

The “power” event byte is located in the R13 AVR register.

Table 2-9. “Power” Event Overview

Event	Description
PWRON	This flag is set if a wake-up via pin 26 (PWRON) was detected. This flag is also set if a pin change occurs on pin 26.
NPWRON6	This flag is set if a wake-up via pin 29 (NWPRON6) was detected. This flag is also set if a pin change occurs on pin 29.
NPWRON5	This flag is set if a wake-up via pin 19 (NWPRON5) was detected. This flag is also set if a pin change occurs on pin 19.
NPWRON4	This flag is set if a wake-up via pin 18 (NWPRON4) was detected. This flag is also set if a pin change occurs on pin 18.
NPWRON3	This flag is set if a wake-up via pin 17 (NWPRON3) was detected. This flag is also set if a pin change occurs on pin 17.
NPWRON2	This flag is set if a wake-up via pin 16 (NWPRON2) was detected. This flag is also set if a pin change occurs on pin 16.
NPWRON1	This flag is set if a wake-up via pin 15 (NWPRON1) was detected. This flag is also set if a pin change occurs on pin 15.

2.10.1.4“Config” Event Byte

The “config” event byte is located in the R12 AVR register.

Table 2-10. “Config” Event Overview

Event	Description
PathB	This flag is set if the current service uses path B.
PathA	This flag is set if the current service uses path A.
ch[1:0]	These bits show the currently used channel.
ser[2:0]	These bits show the currently used service.

2.10.2 Event Configuration

There are two different types of event flags:

- System-specific event flags
- Service-specific event flags

The system-specific event flags can be configured (masked) in the events SRAM variable.

- The pinEventMask variable contains the configuration masks for the wake-up events (NPWRONx and PWRON). If a wake-up event occurs and the corresponding mask bit is set, the EVENT pin is set by the firmware.
- The sysEventMask variable contains the configuration masks for the system events (SYS_ERR, SYS_RDY, and AVCCLOW). If a system event occurs and the corresponding mask bit is set, the EVENT pin is set by the firmware.
- The cmdRdyConf variable contains the masks for the CMD_RDY event flag. If a mask bit is set, the EVENT pin is set after the corresponding operation is initialized or has completed.

Table 2-11. System-Specific Event Flag Configuration Overview

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020D	pinEventMask	PWRON	-	NPWRON6	NPWRON5	NPWRON4	NPWRON3	NPWRON2	NPWRON1
0x020E	sysEventMask	SYS_ERR	-	SYS_RDY	AVCCLOW	-	RX_ACTIVE_EN	RX_ACTIVE_POL	IRQ_POL
0x020F	cmdRdyConf	-	-	SRC_CAL	FRC_CAL	VCO_CAL	RF_CAL	SELFCHECK	-

The service-specific event flags can be configured for each service. The configuration bytes are located in the sramServices data structure.

- rssiSysConf contains the mask and the threshold for the SFIFO event flag during a receive mode. If the rssiBufEvMask mask is set, the EVENT pin is set as soon as the SFIFO is filled with RSSIbuf bytes.
- rxSetPathA/B[0] contains the masks and the thresholds for the DFIFO_RX event flag. The EVENT pin is set at reception on path A/B if the rxBufEvMaskA/B mask is set and the DFIFO is filled with RxbufA/B bytes.
- rxSysEvent contains the masks for the RF events for path A/B. If a mask bit is set, the EVENT pin is set after the corresponding event is detected by the firmware.

Table 2-12 shows an overview of the service-specific, event-flag configuration.

Table 2-12. Service-Specific Event Flag Configuration Overview

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x033F	rssiSysConf	RssiEnable	-	rssiBufEv Mask	RSSIbuf[4:0]				
0x0340	rxSetPathA[0]	-	rxBufEv MaskA	RXbufA[5:0]					
0x0341	rxSetPathB[0]	-	rxBufEv MaskB	RXbufB[5:0]					
0x0342	rxSysEvent	IDCHKA_Mask	WCOKA_Mask	SOTA_Mask	EOTA_Mask	IDCHKB_Mask	WCOKB_Mask	SOTB_Mask	EOTB_Mask

2.11 SPI Operation

The following sections describe the SPI communication between an external MCU and the Atmel® ATA5785. The external microcontroller is always the master of the SPI bus and the Atmel ATA5785 is the SPI slave.

2.11.1 SPI Command Structure

The SPI command set is optimized for a high data rate. Each communication is started by setting pin 27 (NSS) to LOW level. After a start-up time (see Table 2-13 on page 61) the command byte is transmitted. This byte indicates the desired user action such as reading out the received RF data. The command byte is followed by the bytes needed to form the desired command.

Master	cmd	byte1	…	byteN
ATA5785	events.system	events.events	…	response

The first two response bytes are always the events.system and events.events bytes and contain the information on the current system and event status of the Atmel ATA5785 (see Section 2.10 “Event Handling” on page 57).

See Section 7.4 “SPI Command Reference” on page 345 for an overview of all available SPI commands and their related protocol structure.

2.11.2 SPI Handling

This section provides an overview of the SPI timings and describes the SPI data transfer in general. The timing is independent from the SPI mode used. Each SPI mode results in the same performance. See Section 3.8.8 “SPI – Serial Peripheral Interface” on page 266 for more information about the SPI hardware.

Figure 2-21. SPI Timing Parameters

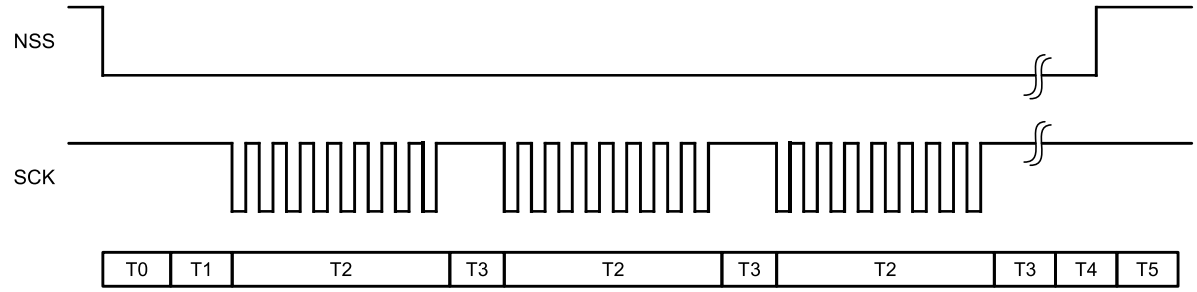


Table 2-13. SPI Timing Example at 500Kbit/s

Time	Timing at 40% Interrupt Load(1)	Description	Depends On	Example
T0	0 or 25µs	Time from NSS LOW to AVR active (in case AVR sleep mode is enabled)	AVR sleep mode (0µs, 1µs, or 25µs)	25µs
T1	17.6µs	Time from AVR active to beginning of first telegram byte	Internal IRQ (falling edge)	45 CPU cycles (ISR) + 15 CPU cycles interrupt response time
T2	16µs	Time to shift in one SPI-byte with SCK	SCK frequency (maximum 500kHz)	8 bit / 500Kbit/s
T3	-	Pause between two bytes. Can be any length including 0µs	External microcontroller timing.	0µs
T4	35.1µs	Time for Atmel ATA5785 to handle last byte	Depends on SPI command used.	Max. 120 CPU cycles(2)
T5	16.1µs	SPI idle time telegram	Internal data processing	40 CPU cycles (ISR) + 15 CPU cycles interrupt response time

Notes: 1. Timing calculation done with AVR® core of 5.7MHz
2. Needed for SPI command “Read RX FIFO” and “Read RSSI FIFO”.

CPU cycles for T4

Read fill level RX FIFO	0
Read fill level RSSI FIFO	0
Get event bytes	0
Read RSSI FIFO	120
Read RX FIFO	120
Write SRAM/register	110
Read SRAM/register	120
Set System/mode	55

Calibrate and check	50
Set watchdog	55
Get version ROM	0
System reset	0
OFF command	0
Start RSSI measurement	55
Get RSSI value	0
Start SRAM CRC calculation	55

SPI Data Transfer Description

After NSS is set to LOW by the external microcontroller, the internal AVR core needs 25µs to wake up from the configured AVR sleep mode. If the FRC is kept alive by setting the FRC always-on (FRCAO) bit in the clock configuration, the time is reduced to 1µs. If the system never enters any of the available sleep modes, T0 can be omitted. The falling edge of the NSS line triggers an internal interrupt. This interrupt resets and initializes the software state machine responsible for handling SPI communication. During the interrupt service routing two event bytes (events.system and events.events) are written to the FIFO SPI output to provide early access to the event status. For this reason the interrupt service routine has to be finished before the transmission of the first telegram byte starts (T1).

After the communication setup is done (T0 + T1), the system is ready to receive the first byte of data. Data bytes can be written continuously or with a pause (T3) between the bytes. The maximum SCK frequency is 500kHz, resulting in a maximum data rate of 500Kbit/s.

The SPI communication is finished by setting NSS back to a logical HIGH level. A certain time T4 must elapse after transmission of the last byte before NSS is allowed to return to HIGH level. T5 is required to internally process the end of SPI transfer before the next SPI command can be received.

2.12 SRAM Configuration

In this section a description of every SRAM variable which holds configuration information is provided. Variables that influence firmware behavior are specified in detail while variables that are only copied to a corresponding hardware register during service/channel start-up are described briefly with a reference to this hardware register. It is generally recommended to use the Atmel configuration tool to define the SRAM configuration and use the generated include files to configure the device.

An overview of the SRAM map in tabular form is provided in Section 7.3 “SRAM Map” on page 325. Modification of the SRAM content is only allowed during IDLEMode.

2.12.1 Event Section

The events variable contains the configuration of the events. See Section 2.10 “Event Handling” on page 57 for a functional description.

2.12.1.1 pinEventMask

The pinEventMask variable contains the configuration masks for the wake-up events.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020D	pinEventMask	PWRON	-	NPWRON6	NPWRON5	NPWRON4	NPWRON3	NPWRON2	NPWRON1

Bit 7: PWRON – Event Mask for pin 26/PWRON
- 0 Pin 28/EVENT is NOT set upon a wake check or pin change on pin 26 (PWRON)
- 1 Pin 28/EVENT is set upon a wake check or pin change on pin 26 (PWRON)

Bit 6: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 5: NPWRON6 – Event Mask for Pin 29/NPWRON6
- 0 Pin 28/EVENT is NOT set upon a wake check or pin change on pin 29 (NPWRON6)
- 1 Pin 28/EVENT is set upon a wake check or pin change on pin 29 (NPWRON6)

Bit 4: NPWRON5 – Event Mask for Pin 19/NPWRON5
- 0 Pin 28/EVENT is NOT set upon a wake check or pin change on pin 19 (NPWRON5)
- 1 Pin 28/EVENT is set upon a wake check or pin change on pin 19 (NPWRON5)

Bit 3: NPWRON4 – Event Mask for Pin 18/NPWRON4
- 0 Pin 28/EVENT is NOT set upon a wake check or pin change on pin 18 (NPWRON4)
- 1 Pin 28/EVENT is set upon a wake check or pin change on pin 18 (NPWRON4)

Bit 2: NPWRON3 – Event Mask for Pin 17/NPWRON3
- 0 Pin 28/EVENT is NOT set upon a wake check or pin change on pin 17 (NPWRON3)
- 1 Pin 28/EVENT is set upon a wake check or pin change on pin 17 (NPWRON3)

Bit 1: NPWRON2 – Event Mask for Pin16/NPWRON2
- 0 Pin 28/EVENT is NOT set upon a wake check or pin change on pin 16 (NPWRON2)
- 1 Pin 28/EVENT is set upon a wake check or pin change on pin 16 (NPWRON2)

Bit 0: NPWRON1 – Event Mask for Pin 15/NPWRON1
- 0 Pin 28/EVENT is NOT set upon a wake check or pin change on pin 15 (NPWRON1)
- 1 Pin 28/EVENT is set upon a wake check or pin change on pin 15 (NPWRON1)

2.12.1.2 sysEventMask

The sysEventMask variable contains the configuration masks for the system events and configuration settings for the RX_ACTIVE and EVENT pins.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020E	sysEventMask	SYS_ERR	-	SYS_RDY	AVCCLOW	-	RX_ACTIVE_EN	RX_ACTIVE_POL	IRQ_POL

Bit 7: SYS_ERR – Event Mask for System Error
- 0 Pin 28/EVENT is NOT set upon detection of a system error
- 1 Pin 28/EVENT is set upon detection of a system error

Bit 6: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 5: SYS_RDY – Event Mask for System Ready
- 0 Pin 28/EVENT is NOT set after system initialization has finished
- 1 Pin 28/EVENT is set after system initialization has finished

Bit 4: AVCCLOW – Event Mask for AVCC Low
- 0 Pin 28/EVENT is NOT set if an AVCCLOW interrupt occurs
- 1 Pin 28/EVENT is set if an AVCCLOW interrupt occurs

Bit 3: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 2: RX_ACTIVE_EN – Enable Rx Active signal on pin 29/RX_ACTIVE
- 0 The receiver activity is not indicated on pin 29/RX_ACTIVE
- 1 The pin 29/RX_ACTIVE is set when the receive path is running and released when the receiver is inactive

Bit 1: RX_ACTIVE_POL – Pin 29/RX_ACTIVE Polarity
- 0 Pin 29/RX_ACTIVE is LOW during an active receive path and HIGH during an inactive receive path
- 1 Pin 29/RX_ACTIVE is HIGH during an active receive path and LOW during an inactive receive path

Bit 0: IRQ_POL – Pin 28/EVENT Polarity
- 0 Pin 28/EVENT is set LOW to indicate an event
- 1 PIn 28/EVENT is set HIGH to indicate an event

2.12.1.3 cmdRdyConf

The cmdRdyConf variable contains the configuration masks for the command ready (CMD_RDY) event.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x020F	cmdRdyConf	-	-	SRC_CAL	FRC_CAL	VCO_CAL	RF_CAL	SELF CHECK	-

Bit 7: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 6: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 5: SRC_CAL – CMD_RDY Event Mask for Polling Cycle/SRC Calibration
- 0 Pin 28/EVENT is NOT set after the polling cycle/SRC calibration process has finished
- 1 Pin 28/EVENT is set after the polling cycle/SRC calibration process has finished

Bit 4: FRC_CAL – CMD_RDY Event Mask for FRC Calibration
- 0 Pin 28/EVENT is NOT set after the FRC calibration process has finished
- 1 Pin 28/EVENT is set after the FRC calibration process has finished

Bit 3: VCO_CAL – CMD_RDY Event Mask for VCO Tuning
- 0 Pin 28/EVENT is NOT set after the VCO tuning process has finished
- 1 Pin 28/EVENT is set after the VCO tuning process has finished

Bit 2: RF_CAL – CMD_RDY Event Mask for RF Calibration
- 0 Pin 28/EVENT is NOT set after the RF calibration process has finished
- 1 Pin 28/EVENT is set after the RF calibration process has finished

Bit 1: SELFCHECK – CMD_RDY Event Mask for System Self Check and Calibration
- 0 Pin 28/EVENT is NOT set after the system self check and calibration process has finished
- 1 Pin 28/EVENT is set after the self check and calibration process has finished

Bit 0: Reserved Bit
This bit is reserved for future use and must be set to zero.

2.12.2 Current Service Section

The current service section contains the configuration of the latest service which was loaded for reception. The structure of the variables is identical to a service in the SRAM service section with one attached channel configuration. An overview of the variables and corresponding addresses can be found in Section 7.3 “SRAM Map” on page 325.

2.12.3 Calibration Section

The calib variable contains settings for receiver calibration.

2.12.3.1 tempMeas

The tempMeas variable contains the current temperature value.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02C3	tempMeas	8-bit temperature measurement result (Range 0..175)							

tempMeas[7:0] can be calculated by the following formula:

tempMeas = T + 50 (2)

Equation parameter: T - Current temperature [°C]

2.12.3.2 srcRes

the srcRes variable contains the result of the SRC oscillator frequency measurement. it shall initially be loaded with the Timer1 compare value T1COR[7:0].

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02C4	srcRes	SRC frequency measurement result							

Bits 7..0: srcRes - SRC Frequency Measurement Result

2.12.3.3 srcCorVal

The srcCorVal variable contains the Timer1 correction value for polling cycle calibration. it shall initially be loaded with the Timer1 compare value T1COR[7:0].

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02C5	srcCorVal	Timer1 correction value							

Bits 7..0: srcCorVal - Timer1 correction value

2.12.3.4 temperatureCalibration

The temperatureCalibration. setting contains a 23-byte table for the XTAL temperature deviation. See Section 2.9.5 “RF Calibration” on page 54 for a functional description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02C7	temperature Calibration	Calibration value for –48°C							
...									
0x02DD	temperature Calibration	Calibration value for +128°C							

The temperature-dependent LO fractional frequency offset (FFRQoffset) caused by the temperature drift of the XTAL is compensated as follows:

The frequency deviation vis-à-vis the crystal temperature must be stored in the SRAM. There are 23 bytes reserved in the variable temperatureCalibration[0:22] for this purpose. This results in a temperature value grid of 8 Kelvin, where temperatureCalibration[0] corresponds to –48°C and temperatureCalibration[22] to +128°C. Each byte contains the frequency deviation of the crystal for the defined temperature in ppm in two’s complement representation. The possible compensation range is thus –128ppm to +127ppm for each temperature value.

2.12.3.5 trxCalibConfiguration

calConf1

The calConf1 variable sets the calibration options during system self check and calibration. See Section 2.9.6 “System Self Check and Calibration” on page 55 for a functional description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DE	calConf1	-	-	EN_ SRCCAL	EN_ FRCCAL	-	-	-	EN_REG REFRESH

Bit 7: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 6: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 5: **EN_SRCCAL** – Enable SRC Calibration
0 Polling cycle/SRC calibration disabled
1 Polling cycle/SRC calibration enabled

Bit 4: **EN_FRCCAL** – Enable FRC Calibration
0 FRC calibration disabled
1 FRC calibration enabled

Bits 3..1: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 0: **EN_REGREFRESH** – Enable Register Refresh
0 Register refreshing disabled
1 Register refreshing enabled

calConf2

The calConf2 variable is used to calibrate the length of the polling cycle to the crystal oscillator frequency.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02DF	calConf2	SRC[7:0]							

Bits 7..0: SRC[7:0] –SRC Calibration Variable

The value SRCCAL_VAR depends on the applied crystal and must be calculated by the customer.

SRCCAL_VAL = [4 × ($f_{XTO/2}$/$f_{SRC\_Nominal}$)] – 256 (3)

SRC7	SRC6	SRC5	SRC4	SRC3	SRC2	SRC1	SRC0	Function
0	0	0	0	0	0	0	0	SRCCAL_VAR = 0
0	0	0	0	0	0	0	1	SRCCAL_VAR = 1
.	.	.	.	.	.	.	.	.
0	1	1	1	1	1	0	1	SRCCAL_VAR = 125 (valid for f_{XTO} = 23.8MHz)
.	.	.	.	.	.	.	.	.
1	0	0	0	0	1	0	1	SRCCAL_VAR = 133 (valid for f_{XTO} = 24.305MHz)
.	.	.	.	.	.	.	.	.
1	0	1	0	0	0	1	1	SRCCAL_VAR = 163 (valid for f_{XTO} = 26.2MHz)
.	.	.	.	.	.	.	.	.

selfChk

The selfChk variable defines the execution period of the self check function. See Section 2.8.4 “PollingMode” on page 24 for a functional description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E0	selfChk	SC[7:0]							

Bits 7..0: SC[7:0] – Self check period

0: No self check
1: Self check every 1*16 polling cycles
...
255: Self check every 255*16 polling cycles

2.12.3.6 frcCalibrationConfiguration

frcCalibGate

The frcCalibGate variable holds the duration of the measured gate (XTO depending). See Section 2.9.3 “FRC Calibration” on page 52 for a functional description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E1	frcCalibGate	data[7:0]							

The data[7:0] variable holds the number of system clock (XTO/4) cycles that are required for 200 FRC cycles:

$$\text{data} = \frac{f_{XTO}/4}{f_{FRC}} \times 200 \qquad (4)$$

This results in data = 191 for a standard crystal with 23.405MHz and the target FRC frequency of 6.36MHz. The default value for data[7:0] is therefore 0xBF.

2.12.4 TRX Configuration Section

2.12.4.1 systemConfig

The systemConfig variable contains power supply and interrupt settings. See Section 2.4.6 “FIFO Interrupt Initialization” on page 16 for a functional description. The hardware descriptions can be found in Section 3.5 “Data and Support FIFOs” on page 162.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E5	systemConfig	-	-	SRCAO	FRCAO	SFIFO_OFL_UFL_RX_disable	DFIFO_OFL_UFL_RX_disable	Current idle mode selector	-

Bits 7..6: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 5: SRCAO – SRC always on
0 SRC “always on” disabled
1 SRC “always on” enabled
This bit is copied to the corresponding bit in the CMOCR hardware register.

Bit 4: FRCAO – FRC always on
0 FRC “always on” disabled
1 FRC “always on” enabled
This bit is copied to the corresponding bit in the CMOCR hardware register.

Bit 3: SFIFO_OFL_UFL_RX_disable – Support FIFO Overflow Underflow Rx Interrupt Disable
0 SFIFO overflow/underflow Rx interrupt enabled
1 SFIFO overflow/underflow Rx interrupt disabled

Bit 2: DFIFO_OFL_UFL_RX_disable – Data FIFO Overflow Underflow Rx Interrupt Disable
0 DFIFO overflow/underflow Rx interrupt enabled
1 DFIFO overflow/underflow Rx interrupt disabled

Bit 1: Current idle mode selector – Firmware internal storage bit.

Bit 0: Reserved Bit
This bit is reserved for future use and must be set to zero.

2.12.5 Sleep Section

The sleep section contains the configuration for the sleep mode behavior. See Section 2.4.2 “AVR Sleep Mode Configuration” on page 15 for a functional description and Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210 for a hardware description.

2.12.5.1 sleepModeConfig

The sleepModeConf setting contains the sleep mode configuration.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02EB	sleepModeConfig	SLEEP_ENA	-	-	-	sleepMode[2:0]			-

Bit 7: SLEEP_ENA – Firmware Sleep Mode Enable Flag
0 Sleep mode disabled
1 Sleep mode enabled

Bits 6..4: Reserved Bits
These bits are reserved for future use and must be set to zero.

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

0: Idle
1: Extended power-save
2: Power-down
3: Power-save

Bit 0: Reserved Bit

This bit is reserved for future use and must be set to zero.

2.12.6 Debug Section

The debug section of the SRAM contains the error code variables in case an internal error occurred. The codes are listed in Section 7.5 “Error Codes” on page 359.

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 last system error							
0x02F5	ssmErrorCode	Contains the detailed error information of SSM error							

2.12.7 Polling Configuration Section

The polling configuration section contains the polling configuration.

2.12.7.1 confT1COR

The confT1COR variable is a copy of the Timer1 compare register (T1COR) and contains the configuration of the Timer1 compare value. Together with the confT1MR variable this setting configures the length of a polling cycle. See Section 2.8.4 “PollingMode” on page 24 for a functional description and Section 3.8.7.2 “Timer1” on page 236 for a hardware description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F6	confT1COR	T1COR[7:0]							

Bits 7..0: T1COR[7:0] – Timer1 Compare Value

Note: To set the correct polling cycle the T1COR[7:0] values must additionally be written to the calib.srcRes and calib.srcCorVal SRAM variables.

2.12.7.2 confT1MR

The confT1MR variable is a copy of the Timer1 mode register (T1MR) and contains the configuration of the Timer1 operating mode. Together with the confT1COR variable this setting configures the length of a polling cycle. See Section 2.8.4 “PollingMode” on page 24 for a functional description and Section 3.8.7.2 “Timer1” on page 236 for a hardware description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F7	confT1MR	T1DC[1:0]		T1PS[3:0]				T1CS[1:0]	

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

T1DC[1:0] must be set to “0b00” when Timer1 is used as the polling timer.

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

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

T1CS[1:0] must be set to “0b00” to select SRC as the polling timer.

2.12.7.3 pollChanConf.config

The pollChanConf.config variable contains the general configuration for the polling channels. See Section 2.8.4 “PollingMode” on page 24 for a functional description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F8	pollChanConf[0].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP
0x02F9	pollChanConf[0].svcChConfig	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
...									
0x0316	pollChanConf[15].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP
0x0317	pollChanConf[15].svcChConfig	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		

Bit 7: RfCalib – RF Calibration
- 0 RF calibration for this polling channel disabled
- 1 RF calibration for this polling channel enabled

Bit 6: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 5: VCOtune – VCO Tuning
- 0 VCO tuning for this polling channel disabled
- 1 VCO tuning for this polling channel enabled

Bits 4..2: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 1: EOL – End of Polling Loop
- 0 End of polling loop not marked
- 1 End of polling loop marked

Bit 0: EOP – End of Polling Cycle
- 0 End of polling cycle not marked
- 1 End of polling cycle marked

2.12.7.4 pollChanConf.svcChConfig

The pollChanConf.svcChConfig variable contains the service/channel configuration for the polling channels. See Section 2.8.4 “PollingMode” on page 24 for a functional description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F8	pollChanConf[0].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP
0x02F9	pollChanConf[0].svcChConfig	enaPathB	enaPathA	Ch[1:0]		-	Ser[2:0]		
...									
0x0316	pollChanConf[15].config	RfCalib	-	VCOtune	-	-	-	EOL	EOP
0x0317	pollChanConf[15].svcChConfig	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]

Ch1	Ch0	Function
0	0	Channel 0
0	1	Channel 1
1	0	Channel 2
1	1	Invalid, set to channel 0

Bit 3: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bits 2..0: Ser[2:0] – Service[2:0]

Ser2	Ser1	Ser0	Function
0	0	0	Invalid
0	0	1	Invalid
0	1	0	Invalid
0	1	1	Service 3
1	0	0	Service 4
1	0	1	Invalid
1	1	0	Invalid
1	1	1	Invalid

2.12.8 SRAM Services Section

The SRAM services section contains the service configuration consisting of one service-specific configuration part and three channel-specific configuration parts.

2.12.8.1 Service-Specific Configuration

This configuration contains the service-specific part of a service configuration.

Note: The used addresses are for service 3. The addresses for service 4 and the current service can be found in Section 7.3 “SRAM Map” on page 325.

CHCR

The CHCR variable is a copy of the channel filter configuration register (CHCR) and contains the channel filter configuration settings. See Section 2.8.7.1 “Channel Filter” on page 35 for a functional description and Section 3.4.3 “Rx Digital Signal Processing (Rx DSP)” on page 97 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x031C	CHCR	-	-	-	-	BWM[3:0]			

Bits 7..4: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bits 3..0: BWM – Bandwidth Mode

CHDN

The CHDN variable is a copy of the channel filter down-sampling register (CHDN) and contains the channel filter down sampling settings. See Section 2.8.7.1 “Channel Filter” on page 35 for a functional description and Section 3.4.3 “Rx Digital Signal Processing (Rx DSP)” on page 97 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x031D	CHDN	-	-	ADCDN	BBDN[4:0]				

Bits 7..6: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 5: ADCDN – ADC Down Sampling Configuration

Bits 4..0: BBDN[4:0] – Baseband Filter Down Sampling Ratio

CHSTARTFILTER

The CHSTARTFILTER variable is a copy of the SSM filter bandwidth register (SSMFBR) and contains the channel filter start-up time settings. See Section 2.8.7.1 “Channel Filter” on page 35 for a functional description and Section 3.7 “Sequencer State Machine” on page 175 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x031E	CHSTART FILTER	-	-	PLDT	-	DFDT	FID[2:0]		

Bits 7..6: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 5: PLDT – PLL Lock Delay Time

Bit 6: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 3: DFDT – Double Filter Delay Time

Bits 2..0: FID – Filter Delay

DMCDA, DMCDB

The DMCDA/B variables are a copy of the demodulator carrier detect registers (DMCDA/B) and contain the demo-dulator carrier detect settings for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.2 “Demodulator and Signal Checks” on page 101 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x031F	DMCDA	DMCTA[2:0]			DMCLA[4:0]				
0x0320	DMCDB	DMCTB[2:0]			DMCLB[4:0]				

Bits 7..5: DMCTA/B[2:0] – Demodulator Carrier Detect Time for Path A/B

Bits 4..0: DMCLA/B[4:0] – Demodulator Carrier Detect Limit for Path A/B

DMCRA, DMCRB

The DMCRA/B variables are a copy of the demodulator control registers (DMCRA/B) and contain the demodulator control settings for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.2 “Demodulator and Signal Checks” on page 101 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0321	DMCRA	DMARA	SY1TA	SASKA	DMPGA[4:0]				
0x0322	DMCRB	DMARB	SY1TB	SASKB	DMPGB[4:0]				

Bit 7: DMARA/B – Demodulator Automatic Restart on Path A/B

Bit 6: SY1TA/B – Symbol Check with 1T only on Path A/B

Bit 5: SASKA/B – Select ASK Input for Path A/B

Bits 4..0: DMPGA/B[4:0] – Demodulator PLL Loop Gain for Path A/B

DMDRA, DMDRB

The DMDRA/B variables are a copy of the demodulator data rate registers (DMDRA/B) and contain the demodulator data rate settings for path A and path B. See Section 2.8.7.2 “Demodulation Settings” on page 36 for a functional description and Section 3.4.3.2 “Demodulator and Signal Checks” on page 101 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0323	DMDRA	DMDNA[3:0]				DMAA[3:0]			
0x0324	DMDRB	DMDNB[3:0]				DMAB[3:0]			

Bits 7..4: DMDNA/B – Demodulator Down Sampling on Path A/B

Bits 3..0: DMAA/B – Demodulator Moving Average Data Rate Factor on Path A/B

DMMA, DMMB

The DMMA/B variables are a copy of the demodulator mode registers (DMMA/B) and contain the demodulator operating mode settings for path A and path B. See Section 2.8.7.2 “Demodulation Settings” on page 36 for a functional description and Section 3.4.3.2 “Demodulator and Signal Checks” on page 101 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0325	DMMA	DMNEA	DMHA	DMPA	DMATA[4:0]				
0x0326	DMMB	DMNEB	DMHB	DMPB	DMATB[4:0]				

Bit 7: DMNEA/B – Demodulator NRZ Enable for Path A/B

Bit 6: DMHA/B – Demodulator Hold Mode for Path A/B

Bit 5: DMPA/B – Demodulator Received Data Polarity for Path A/B

Bits 4..0:DMATA/B[4:0] – Demodulator Amplitude Threshold for Path A/B

EOT1A, EOT1B

The EOT1A/B variables are a copy of the end of telegram conditions 1 registers (EOTC1A/B) and contain the EOT configuration during WUP check for path A and path B. See Section 2.8.6 “Rx Telegram Handling” on page 30 for a functional description and Section 3.4.3.7 “Rx DSP Control” on page 142 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0327	EOT1A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x0328	EOT1B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

Bit 7: EOTB/AFE – End of Telegram on Path B/A Fail Enable

Bit 6: RRFEA/B – RSSI Range Fail Enable on Path A/B

Bit 5: TELREA/B – Telegram Length Reached Enable on Path A/B

Bit 4: TMOFEA/B – Time-out Fail Enable on Path A/B

Bit 3: MANFEA/B – Manchester Coding Fail Enable on Path A/B

Bit 2: SYTFEA/B – Symbol Timing Check Fail Enabled on Path A/B

Bit 1: AMPFEA/B – Demodulation Amplitude Check Fail Enable on Path A/B

Bit 0: CARFA/B – Carrier Check Fail Enable on Path A/B

EOT2A, EOT2B

The EOT2A/B variables are a copy of the end of telegram conditions 2 registers (EOTC2A/B) and contain the EOT configuration during SFID check for path A and path B. See Section 2.8.6 “Rx Telegram Handling” on page 30 for a functional description and Section 3.4.3.7 “Rx DSP Control” on page 142 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0329	EOT2A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x032A	EOT2B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

See EOT1A, EOT1B variables for the bit descriptions.

EOT3A, EOT3B

The EOT3A/B variables are a copy of the end of telegram conditions 3 registers (EOTC3A/B) and contain the EOT configuration during payload reception for path A and path B. See Section 2.8.6 “Rx Telegram Handling” on page 30 for a functional description and Section 3.4.3.7 “Rx DSP Control” on page 142 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x032B	EOT3A	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA
0x032C	EOT3B	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB

See EOT1A, EOT1B variables for the bit descriptions.

FEVCO

The FEVCO variable is a copy of the RF front end VCO and PLL control register (FEVCO) and contains the front end VCO control settings. See Section 3.3.3 “VCO” on page 93 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0330	FEVCO	VCOB[3:0]				CPCC[3:0]			

Bits 7..4: VCOB[3:0] – VCO Bias

Bits 3..0: CPCC[3:0] – Charge Pump Current Control

FEVCT

The FEVCT variable is a copy of the RF front end VCO tuning register (FEVCT) and contains VCO tuning control settings. See Section 2.9.4 “VCO Tuning” on page 53 for a functional description and Section 3.3.3 “VCO” on page 93 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0331	FEVCT	-	-	-	-	FEVCT[3:0]			

Bits 7..4: Reserved Bits

These bits are reserved for future use and must be set to zero.

Bits 3..0: FEVCT[3:0] – Front-end VCO Tuning Control

Initial value for the VCO tuning. The default value is 7.

IF

The IF variable contains the intermediate frequency setting.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x033C	IF[0]	IF[7:0]							
0x033D	IF[1]	IF[15:8]							

The IF value can be calculated by the following equation:

$$IF = \text{round}\left(\frac{F_{RF}}{F_{XTO}} \times \frac{2^{\eta}}{F_{LO\_IF}}\right) \qquad (5)$$

f_{RF}: RF frequency in Hz

f_{XTO}: XTO frequency in Hz

η: 18 for Low-Band (310MHz to 477MHz)

FLO_IF: 1225 for Low-Band 310MHz to 318MHz

1728 for Low-Band 418MHz to 477MHz

RDOCR

The RDOCR variable contains the receive output control settings. See Section 2.8.3 “RXMode(buffered)” on page 23 for a functional description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x033E	RDOCR	-	0	0	ETRPB	ETRPA	TMDS[1:0]		-

Bit 7: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 6: Reserved Bit
This bit must always be set to zero.

Bit 5: Reserved Bit
This bit must always be set to zero**.**

Bit 4: **ETRPB** – Enable Raw Transparent Output Path B to Pin 19/TRPB

Bit 3: **ETRPA** – Enable Raw Transparent Output Path A to Pin 16/TRPA

Bits 2..1: TMDS[1:0] – Transparent Mode Data Select
In transparent receive mode the TMDS bits are irrelevant because the data selection is done by the firmware and the SSM.

In buffered receive mode an additional reshaped transparent output on the 17/TMDO and 19/TMDO_CLOCK pins is activated if at least one of the TMDS bits is set to “1”. The active path is automatically selected by the SSM.

Bit 0: Reserved Bit
This bit is reserved for future use and must be set to zero.

rssiSysConf

The rssiSysConf variable is used for RSSI enabling/disabling and event signaling. See Section 2.8.7.6 “RSSI Measurement” on page 47 for a functional description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x033F	rssiSysConf	RssiEnable	-	rssiBufEv Mask	RSSIbuf[4:0]				

Bit 7: RssiEnable – RSSI Enable
0 RSSI disabled
1 RSSI enabled

Bit 6: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 5: rssiBufEvMask – SFIFO Fill Level Event Mask in Receive Mode
0 No external event on pin 28 (EVENT) is generated if the SFIFO fill level is reached
1 An external event on pin 28 (EVENT) is generated if the SFIFO fill level is reached

Bits 4..0: RSSIbuf[4:0] – SFIFO [4:0] Fill Level Threshold in Receive Mode

rxSetPathA[0], rxSetPathB[0]

The rxSetPathA/B[0] variables are used for Rx event signaling and configuration for path A/B. See Section 2.8.3 “RXMode(buffered)” on page 23 for a functional description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0340	rxSetPathA[0]	-	rxBufEv MaskA	RXbufA[5:0]					
0x0341	rxSetPathB[0]	-	rxBufEv MaskB	RXbufB[5:0]					

Bit 7: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 6: rxBufEvMaskA/B – DFIFO Fill Level Path A/B in Receive Mode
0 No external event on pin 28 (EVENT) is generated if the DFIFO fill level on path A/B is reached in receive mode.
1 An external event on pin 28 (EVENT) is generated if the DFIFO fill level on path A/B is reached in receive mode.

Bits 5..0: RXbufA/B[5:0] – DFIFO A/B [5:0] Fill Level Threshold in Receive Mode

rxSetPathA[1], rxSetPathB[1]

The rxSetPathA/B[1] variables are mainly used for Rx telegram handling control for path A and path B. See Section 2.8.6 “Rx Telegram Handling” on page 30 for a functional description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0342	rxSetPathA[1]	IWUPA	DARA	GAPMA	RXTEHA	-	-	-	RXMODA
0x0343	rxSetPathB[1]	IWUPB	DARB	GAPMB	RXTEHB	-	-	-	RXMODB

Bit 7: IWUPA/B – Intermittent WUP Path A/B
- 0 Intermittent WUP disabled
- 1 Intermittent WUP enabled

Bit 6: DARA/B – Demodulator Automatic Restart on Path A/B
- 0 Demodulator automatic restart disabled
- 1 Demodulator automatic restart enabled

Bit 5: GAPMA/B – Gap Mode Path A/B
- 0 Gap mode disabled
- 1 Gap mode enabled

Bit 4: RXTEHA/B – Rx Telegram End Handling Path A/B
- 0 Return to IDLEMode after EOT
- 1 Stay in RxMode after EOT

Bits 3..1: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 0: RXMODA/B – Rx Modulation Path A/B
- 0 FSK modulation
- 1 ASK modulation

rxSysEvent

The rxSysEvent variable contains the configuration masks for the system events. For a functional description, see Section 2.10 “Event Handling” on page 57.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0344	rxSysEvent	IDCHKA _Mask	WCOKA _Mask	SOTA _Mask	EOTA _Mask	IDCHKB _Mask	WCOKB _Mask	SOTB _Mask	EOTB _Mask

Bit 7: IDCHKA_Mask – ID Check OK on Path A Mask
- 0 No external event on pin 28 (EVENT) is generated if ID check OK on path A
- 1 An external event on pin 28 (EVENT) is generated if ID check OK on path A

Bit 6: WCOKA_Mask – Wake Check OK on Path A Mask
- 0 No external event on pin 28 (EVENT) is generated if wake check OK on path A
- 1 An external event on pin 28 (EVENT) is generated if wake check OK on path A

Bit 5: SOTA_Mask – Start of Telegram on Path A Mask
- 0 No external event on pin 28 (EVENT) is generated if SOT on path A
- 1 An external event on pin 28 (EVENT) is generated if SOT on path A

Bit 4: EOTA_Mask – End of Telegram on Path A Mask
- 0 No external event on pin 28 (EVENT) is generated if EOT on path A
- 1 An external event on pin 28 (EVENT) is generated if EOT on path A

Bit 3: IDCHKB_Mask – ID Check OK on Path B Mask
- 0 No external event on pin 28 (EVENT) is generated if ID check OK on path B
- 1 An external event on pin 28 (EVENT) is generated if ID check OK on path B

Bit 2: WCOKB_Mask – Wake Check OK on Path B Mask
- 0 No external event on pin 28 (EVENT) is generated if wake check OK on path B
- 1 An external event on pin 28 (EVENT) is generated if wake check OK on path B

Bit 1: SOTB_Mask – Start of Telegram on Path B Mask
- 0 No external event on pin 28 (EVENT) is generated if SOT on path B
- 1 An external event on pin 28 (EVENT) is generated if SOT on path B

Bit 0: EOTB_Mask – End of Telegram on Path B Mask
- 0 No external event on pin 28 (EVENT) is generated if EOT on path A
- 1 An external event on pin 28 (EVENT) is generated if EOT on path A

rxSysSet

The rxSysSet variable is used for enabling and disabling various RXMode features. See Section 2.8.7 “General Rx Settings” on page 35 for a functional description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0345	rxSysSet	-	IdScan _ENA	IFAmplifier_ ENA	PathValid AfterSOT _ENA	1	1	-	-

Bit 7: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 6: IdScan_ENA – ID Check Enable
- 0 Disable ID check
- 1 Enable ID check

Bit 5: IFAmplifier_ENA – IF Amplifier Enable
This bit must always be set to “1” during normal operation.

Bit 4: PathValidAfterSOT_ENA – Path Valid after Start of Telegram Enable
- 0 Path valid after SOT disable
- 1 Path valid after SOT enable

Bits 3..2: Reserved Bits
These bits must always be set to 1.

Bits 1..0: Reserved Bits
These bits are reserved for future use and must be set to zero.

SFIDA, SFIDB

The SFIDA/B[3:0] variables are a copy of the start frame ID A/B registers (SFID[4:1]A/B) and contain the start frame ID pattern for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.3 “Frame Synchronizer” on page 121 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0346	SFIDA[0]	SFID1A[7:0]							
0x0347	SFIDA[1]	SFID2A[7:0]							
0x0348	SFIDA[2]	SFID3A[7:0]							
0x0349	SFIDA[3]	SFID4A[7:0]							
0x034A	SFIDB[0]	SFID1B[7:0]							
0x034B	SFIDB[1]	SFID2B[7:0]							
0x034C	SFIDB[2]	SFID3B[7:0]							
0x034D	SFIDB[3]	SFID4B[7:0]							

SFIDCA, SFIDCB

The SFICA/B variables are a copy of the start frame ID configuration registers (SFIDCA/B) and contain the start frame ID configuration for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.3 “Frame Synchronizer” on page 121 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x034E	SFIDCA	SEMEA	-	-	SFIDTA[4:0]				
0x034F	SFIDCB	SEMEB	-	-	SFIDTB[4:0]				

Bit 7: SEMEA/B – Serial Mode Enable for Path A/B

0 Serial Mode Disabled

1 Serial Mode Enabled

Bits 6..5: Reserved Bits

These bits are reserved for future use and must be set to zero.

Bits 4..0: SFIDTA/B[4:0] – SFID Threshold for Path A/B

SFIDLA, SFIDLB

The SFIDLA/B variables are a copy of the start frame ID length registers (SFIDLA/B) and contain the start frame ID length settings for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.3 “Frame Synchronizer” on page 121 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0350	SFIDLA	-	-	SFIDLA[5:0]					
0x0351	SFIDLB	-	-	SFIDLB[5:0]					

Bits 7..6: Reserved Bits

These bits are reserved for future use and must be set to zero.

Bits 5..0: SFIDLA/B[5:0] – SFID Length for Path A/B (0..32)

SOT1A, SOT1B

The SOT1A/B variables are a copy of the start of telegram conditions 1 registers (SOTC1A/B) and contain the SOT configuration during WUP check for path A and path B. See Section 2.8.6 “Rx Telegram Handling” on page 30 for a functional description and Section 3.4.3.7 “Rx DSP Control” on page 142 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0352	SOT1A	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x0353	SOT1B	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB

Bit 7: WCOB/AOE – Wake Check OK from Path B/A OK Enable

Bit 6: RROEA/B – RSSI Range OK Enable for Path A/B

Bit 5: SFIDEA/B – SFID Match Enable for Path A/B

Bit 4: WUPEA/B – Wake Up Pattern Match Enable for Path A/B

Bit 3: MANOEA/B – Manchester Coding OK Enable for Path A/B

Bit 2: SYTOEA/B – Symbol Timing Check OK Enable for Path A/B

Bit 1: AMPOEA/B – Demodulation Amplitude Check OK Enable for Path A/B

Bit 0: CAROEA/B – Carrier Check OK Enable for Path A/B

SOT2A, SOT2B

The SOT2A/B variables are a copy of the start of telegram conditions 2 registers (SOTC2A/B) and contain the SOT configuration during SOT check for path A and path B. See Section 2.8.6 “Rx Telegram Handling” on page 30 for a functional description and Section 3.4.3.7 “Rx DSP Control” on page 142 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0354	SOT2A	WCOBOE	RROEA	SFIDEA	WUPEA	MANOEA	SYTOEA	AMPOEA	CAROEA
0x0355	SOT2B	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB

See SOT1A, SOT1B variables for the bit descriptions.

SOTtimeOutA, SOTtimeOutB

The SOTtimeOutA/B variables are a copy of the SOT OK time-out for path A/B registers (SOTTOA/B) and contain the start of telegram time-out configuration for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.7 “Sequencer State Machine” on page 175 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0356	SOTtimeOutA	SOTTOA[7:0]							
0x0357	SOTtimeOutB	SOTTOB[7:0]							

Bits 7..0: SOTTOA/B[7:0] – SOT Time-out for Path A/B

SOTTOA/B allows to set time-out periods between approximately 5µs and 300ms. A precise formula is given at the corresponding hardware register description.

SYCA, SYCB

The SYCA/B variables are a copy of the symbol check configuration registers (SYCA/B) and contain the symbol check configuration for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.2 “Demodulator and Signal Checks” on page 101 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0358	SYCA	SYTLA[3:0]				SYCSA[3:0]			
0x0359	SYCB	SYTLB[3:0]				SYCSB[3:0]			

Bits 7..4: SYTLA/B[3:0] – Symbol Timing Limit for Path A/B

Bits 3..0: SYCSA/B[3:0] – Symbol Check Size for Path A/B

TMUL

The TMUL variable contains the multiplier to calculate temperature compensation of the crystal. See Section 2.9.5 “RF Calibration” on page 54 for a functional description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x035C	TMUL	TMUL[7:0]							

The TMUL value can be calculated by the following equation:

$TMUL = round(32 \times f_{RF} / (10^6 \times f_{Step}))$ (6)

f_{RF}: RF channel frequency

$f_{Step} = f_{XTO} / (N \times 2^{16})$

f_{XTO}: Crystal oscillator frequency

N = 4 for Low-Band (315MHz/433MHz)

trxSysConf

The trxSysConf variable contains the enable bits for raw transparent data output. See Section 2.8.7 “General Rx Settings” on page 35 for a functional description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x035D	trxSysConf	-	-	-	-	TRPB_ENA	TRPA_ENA	-	-

Bits 7..4: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 3: TRPB_ENA – Transparent Rx Path B Enable
This setting is not used by the Atmel® ATA5785 firmware. Use sramServices.RDOCR.ETRPB to activate the raw transparent output on path B.

Bit 2: TRPA_ENA - Transparent Rx Path A Enable
This setting is not used by the Atmel ATA5785 firmware. Use sramServices.RDOCR.ETRPA to activate the raw transparent output on path A.

Bits 1..0: Reserved Bits
These bits are reserved for future use and must be set to zero.

WCOtimeOutA, WCOtimeOutB

The WCOtimeOutA/B variables are a copy of the WCO time-out for path A/B registers (WCOTOA/B) and contain the wake check OK time-out configuration for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.7 “Sequencer State Machine” on page 175 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x036A	WCOtimeOutA	WCOTOA[7:0]							
0x0036B	WCOtimeOutB	WCOTOB[7:0]							

Bits 7..0: WCOTOA/B[7:0] – WCO Time-out for Path A/B

WCOTOA/B allows time-out periods to be set between approximately 5µs and 300ms. A precise formula is given at the corresponding hardware register description.

WUPA, WUPB

The WUPA/B[3:0] variables are a copy of the wake-up pattern A/B registers (WUP[4:1]A/B) and contain the wake-up pattern for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.3 “Frame Synchronizer” on page 121 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x036C	WUPA[0]	WUP1A[7:0]							
0x036D	WUPA[1]	WUP2A[7:0]							
0x036E	WUPA[2]	WUP3A[7:0]							
0x036F	WUPA[3]	WUP4A[7:0]							
0x0370	WUPB[0]	WUP1B[7:0]							
0x0371	WUPB[1]	WUP2B[7:0]							
0x0372	WUPB[2]	WUP3B[7:0]							
0x0373	WUPB[3]	WUP4B[7:0]							

WUPLA, WUPLB

The WUPLA/B variables are a copy of the wake-up pattern length registers (WUPLA/B) and contain the wake-up pattern length settings for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.3 “Frame Synchronizer” on page 121 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0374	WUPLA	-	-	WUPLA[5:0]					
0x0375	WUPLB	-	-	WUPLB[5:0]					

Bits 7..6: Reserved Bits

These bits are reserved for future use and must be set to zero.

Bits 5..0: WUPLA/B[5:0] – WUP Length for Path A/B (0..32)

WUPTA, WUPTB

The WUPTA/B variables are a copy of the wake-up pattern threshold registers (WUPTA/B) and contain the wake-up pattern threshold for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.3 “Frame Synchronizer” on page 121 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0376	WUPTA	-	-	-	WUPTA[4:0]				
0x0377	WUPTB	-	-	-	WUPTB[4:0]				

Bits 7..5: Reserved Bits

These bits are reserved for future use and must be set to zero.

Bits 4..0: WUPTA/B[4:0] – WUP Threshold for Path A/B

RXCPA[1:0], RXCPB[1:0]

The RXCPA/B[1:0] variables are a copy of the Rx CRC polynomial registers (RXCPHA/B and RXCPLA/B) and contain the Rx CRC polynomial for path A and path B. See Section 2.8.7.4 “Receive CRC Checker ” on page 45 for a functional description and Section 3.4.3.4 “Rx Buffer” on page 129 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0378	RXCPA[0]	RXCPLA[7:0]							
0x0379	RXCPA[1]	RXCPHA[7:0]							
0x037F	RXCPB[0]	RXCPLB[7:0]							
0x0380	RXCPB[1]	RXCPHB[7:0]							

Bits 7..0: RXCPLA/B[7:0] – Rx CRC Polynomial Low Byte for Path A/B

Bits 7..0: RXCPHA/B[7:0] – Rx CRC Polynomial High Byte for Path A/B

RXCIA[1:0], RXCIB[1:0]

The RXCIA/B[1:0] variables are a copy of the Rx CRC initialization registers (RXCIHA/B and RXCILA/B) and contain the Rx CRC initialization value for path A and path B. See Section 2.8.7.4 “Receive CRC Checker ” on page 45 for a functional description and Section 3.4.3.4 “Rx Buffer” on page 129 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x037A	RXCIA[0]	RXCILA[7:0]							
0x037B	RXCIA[1]	RXCIHA[7:0]							
0x0381	RXCIB[0]	RXCILB[7:0]							
0x0382	RXCIB[1]	RXCIHB[7:0]							

Bits 7..0: RXCILA/B[7:0] – Rx CRC Init Value Low Byte for Path A/B

Bits 7..0: RXCIHA/B[7:0] – Rx CRC Init Value High Byte for Path A/B

RXCSBA, RXCSBB

The RXCSBA/B variables are a copy of the Rx CRC skip bits registers (RXCSBA/B) and contain the Rx CRC skip bits value for path A and path B. See Section 2.8.7.4 “Receive CRC Checker ” on page 45 for a functional description and Section 3.4.3.4 “Rx Buffer” on page 129 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x037C	RXCSBA	RXCSBA[7:0]							
0x0383	RXCSBB	RXCSBB[7:0]							

Bits 7..0: RXCSBA/B[7:0] – Rx CRC Skip Bit Number for Path A/B

RXTLA[1:0], RXTLB[1:0]

The RXTLA/B[1:0] variables are a copy of the Rx telegram length registers (RXTLHA/B and RXTLLA/B) and contain the telegram length setting in RXMode(buffered) for path A and path B. See Section 2.8.7.3 “Telegram Settings and Signal Checks” on page 39 for a functional description and Section 3.4.3.4 “Rx Buffer” on page 129 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x037D	RXTLA[0]	RXTLLA[7:0]							
0x037E	RXTLA[1]	-	-	-	-	RXTLHA[3:0]			
0x0384	RXTLB[0]	RXTLLB[7:0]							
0x0385	RXTLB[1]	-	-	-	-	RXTLHB[3:0]			

Bits 7..0: RXTLLA/B[7:0] – Rx Telegram Length Low Byte for Path A/B

Bits 3..0: RXTLHA/B[3:0] – Rx Telegram Length High Byte for Path A/B

RXBC1

The RXBC1 variable is a copy of the Rx buffer control 1 register (RXBC1) and contains the settings for the Rx buffer configuration for path A and path B. See Section 2.8.7 “General Rx Settings” on page 35 for a functional description and Section 3.4.3.4 “Rx Buffer” on page 129 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0386	RXBC1	RXMSBB	RXCBLB[1:0]		RXCEB	RXMSBA	RXCBLA[1:0]		RXCEA

Bit 7: RXMSBB – Rx MSB-First for Path B

Bits 6..5: RXCBLB[1:0] – Rx CRC Bit Length for Path B

Bit 4: RXCEB – Rx CRC Enable for Path B

Bit 3: RXMSBA – Rx MSB-First for Path A

Bits 2..1: RXCBLA[1:0] – Rx CRC Bit Length for Path A

Bit 0: RXCEA – Rx CRC Enable for Path A

RSSC

The RSSC variable is a copy of the receive signal strength configuration register (RSSC) and contains the control settings of the RSSI buffer. See Section 2.8.7.6 “RSSI Measurement” on page 47 for a functional description and Section 3.4.3.6 “RSSI Buffer” on page 139 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0399	RSSC	-	RSPKF	-	RSWLH	RSUP[3:0]			

Bit 7: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 6: RSPKF – RSSI Peak Values to SFIFO

Bit 5: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 4: RSWLH – RSSI Within Low and High Limits

Bits 3..0: RSUP[3:0] – RSSI Update Period

2.12.8.2 Channel-Specific Configuration

This section contains the channel-specific part of a service configuration.

Note: The addresses shown are for service 3/channel 0. The addresses for different services and channels can be found in Section 7.3 “SRAM Map” on page 325.

FFREQ[2:0]

The FFREQ[2:0] variables are a copy of the fractional frequency registers (FFREQ[2:1]L/M/H) and contain the base frequency setting for the fractional-N PLL. See Section 2.8.7.2 “Demodulation Settings” on page 36 for a functional description and Section 3.3 “Fractional-N PLL” on page 92 for a hardware description.

Address Ser3/Ch0	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x039A	FFREQ[0]	FFREQL[7:0]							
0x039B	FFREQ[1]	FFREQM[7:0]							
0x039C	FFREQ[2]	FFREQH[7:0]							

Bits 7..0: FFREQL/M/H – Fractional Frequency Base Setting

FEMS

The FEMS variable is a copy of the RF front end main and swallow control register (FEMS) and contains the main and swallow counter values of the fractional-N PLL. See Section 2.8.7.2 “Demodulation Settings” on page 36 for a functional description and Section 3.3 “Fractional-N PLL” on page 92 for a hardware description.

Address Ser3/Ch0	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x039D	FEMS	PLLM[3:0]				PLLS[3:0]			

Bits 7..4: PLLM[3:0] – PLL M Counter Divider Ratio

Bits 3..0: PLLS[3:0] – PLL S Counter Divider Ratio

FECR

The FECR variable is a copy of the RF front end control register (FECR) and contains control settings of the fractional-N PLL. See Section 2.8.7.2 “Demodulation Settings” on page 36 for a functional description and Section 3.3 “Fractional-N PLL” on page 92 for a hardware description.

Address Ser3/Ch0	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x039E	FECR	-	-	ANPS	PLCKG	ADHS	-	S4N3	-

Bits 7..6: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bit 5: ANPS – ASK not DPSK Switch.
This bit must be set to “0” because the function is controlled by the firmware/SSM.

Bit 4: PLCKG – PLL Lock Detect Gate
This bit must be set to “0” because the function is controlled by the firmware/SSM.

Bit 3: ADHS – ADC High Sample Rate
This bit must be set to “0” because the function is controlled by the firmware/SSM.

Bit 2: Reserved Bit
This bit is reserved for future use and must be set to zero.

Bit 1: S4N3 – Select 433MHz not 315MHz Band

Bit 0: Reserved Bit

This bit is reserved for future use and must always be set to zero.

2.12.9 RSSI Section

The RSSI section contains the low and high corners of the RSSI window. The values can be configured independently for every service /channel combination. See Section 2.8.7.6 “RSSI Measurement” on page 47 for a functional description and Section 3.4.3.6 “RSSI Buffer” on page 139 for a hardware description.

Address Service3	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0448	rssiThresholds[3][0].RSSL	RSSL for service 3 and channel 0							
0x0449	rssiThresholds[3][0].RSSH	RSSH for service 3 and channel 0							
0x044A	rssiThresholds[3][1].RSSL	RSSL for service 3 and channel 1							
0x044B	rssiThresholds[3][1].RSSH	RSSH for service 3 and channel 1							
0x044C	rssiThresholds[3][2].RSSL	RSSL for service 3 and channel 2							
0x044D	rssiThresholds[3][2].RSSH	RSSH for service 3 and channel 2							
0x044E	rssiThresholds[4][0].RSSL	RSSL for service 4 and channel 0							
0x044F	rssiThresholds[4][0].RSSH	RSSH for service 4 and channel 0							
0x0450	rssiThresholds[4][1].RSSL	RSSL for service 4 and channel 1							
0x0451	rssiThresholds[4][1].RSSH	RSSH for service 4 and channel 1							
0x0452	rssiThresholds[4][2].RSSL	RSSL for service 4 and channel 2							
0x0453	rssiThresholds[4][2].RSSH	RSSH for service 4 and channel 2							

2.12.10 ID Check Section

The ID check section contains the settings for the ID check. 18 IDs with up to 4 bytes together with a config variable and a global enable variable can be stored in this structure. See Section 2.8.7.5 “ID Check” on page 46 for a functional description and Section 3.4.3.5 “ID Check” on page 136 for a hardware description.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x046C	idEna	-	-	-	ID[4:0]				
0x046D	data[0][0]	data0_byte0[7:0]							
0x046E	data[0][1]	data0_byte1[7:0]							
0x046F	data[0][2]	data0_byte2[7:0]							
0x0470	data[0][3]	data0_byte3[7:0]							
0x0471	config[0]	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	
...									
0x04C2	data[17][0]	data17_byte0[7:0]							
0x04C3	data[17][1]	data17_byte1[7:0]							
0x04C4	data[17][2]	data17_byte2[7:0]							
0x04C5	data[17][3]	data17_byte3[7:0]							
0x04C6	config[17]	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	

idEna

The idEna variable is used to activate the ID check and define the number of active IDs.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x046C	idEna	-	-	-	ID[4:0]				

Bits 7..5: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bits 4..0: ID[4:0] – Number of Enabled IDs

ID[4]	ID[3]	ID[2]	ID[1]	ID[0]	Description
0	0	0	0	0	ID check function disabled
0	0	0	0	1	ID check function enabled for ID 0 (all other IDs are disabled)
0	0	0	1	0	ID check function enabled for ID 0 to 1 (all other IDs are disabled)
0	0	0	1	1	ID check function enabled for ID 0 to 2 (all other IDs are disabled)
.	.	.	.	.	.
1	0	0	1	0	ID check function enabled for ID 0 to 17
1	0	0	1	1	Not applicable
.	.	.	.	.	.
1	1	1	1	1	Not applicable

data

The data variables store the IDs themselves with a length of 1, 2, 3, or 4 bytes. The IDs are stored MSB-first, starting with the data[x][0] variable (x = 0..17). Each ID can have a unique length.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
data[x][0]	data0_byte0[7:0]							
data[x][1]	data0_byte1[7:0]							
data[x][2]	data0_byte2[7:0]							
data[x][3]	data0_byte3[7:0]							

config

The config variable contains the configuration data of the corresponding ID.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
config[x]	IDCE	-	-	-	IDBO[1:0]		IDL[1:0]	

Bit 7: IDCE – ID Check Execute
This bit must be set to “1”. If set to “0”, the hardware correlator omits the corresponding ID.

Bits 6..4: Reserved Bits
These bits are reserved for future use and must be set to zero.

Bits 3..2: IDBO[1:0] – ID Byte Offset

IDBO[1]	IDBO[0]	Description
0	0	ID check starts with 1st byte in telegram payload
0	1	ID check starts with 2nd byte in telegram payload
1	0	ID check starts with 3rd byte in telegram payload
1	1	ID check starts with 4th byte in telegram payload

Bits 1..0: IDL[1:0] – ID Length

IDL[1]	IDL[0]	Description
0	0	ID length is 1 byte, data_byte0 is used only
0	1	ID length is 2 bytes, data_byte0 and data_byte1 are used only
1	0	ID length is 3 bytes, data_byte0 .. data_byte2 are used only
1	1	ID length is 4 bytes, data_byte0 .. data_byte3 are used

2.12.11 SRAM CRC Section

The SRAM CRC section contains the busy flag and the result of the CRC calculation that was started by the SPI command “Start SRAM CRC Calculation”

result

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x04F9	result	Final calculated CRC value							

status

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x04FA	status	-	-	-	-	-	-	-	CRCstatus

Bits 7..1: Reserved Bits
These bits are reserved for future use and must be set to zero.

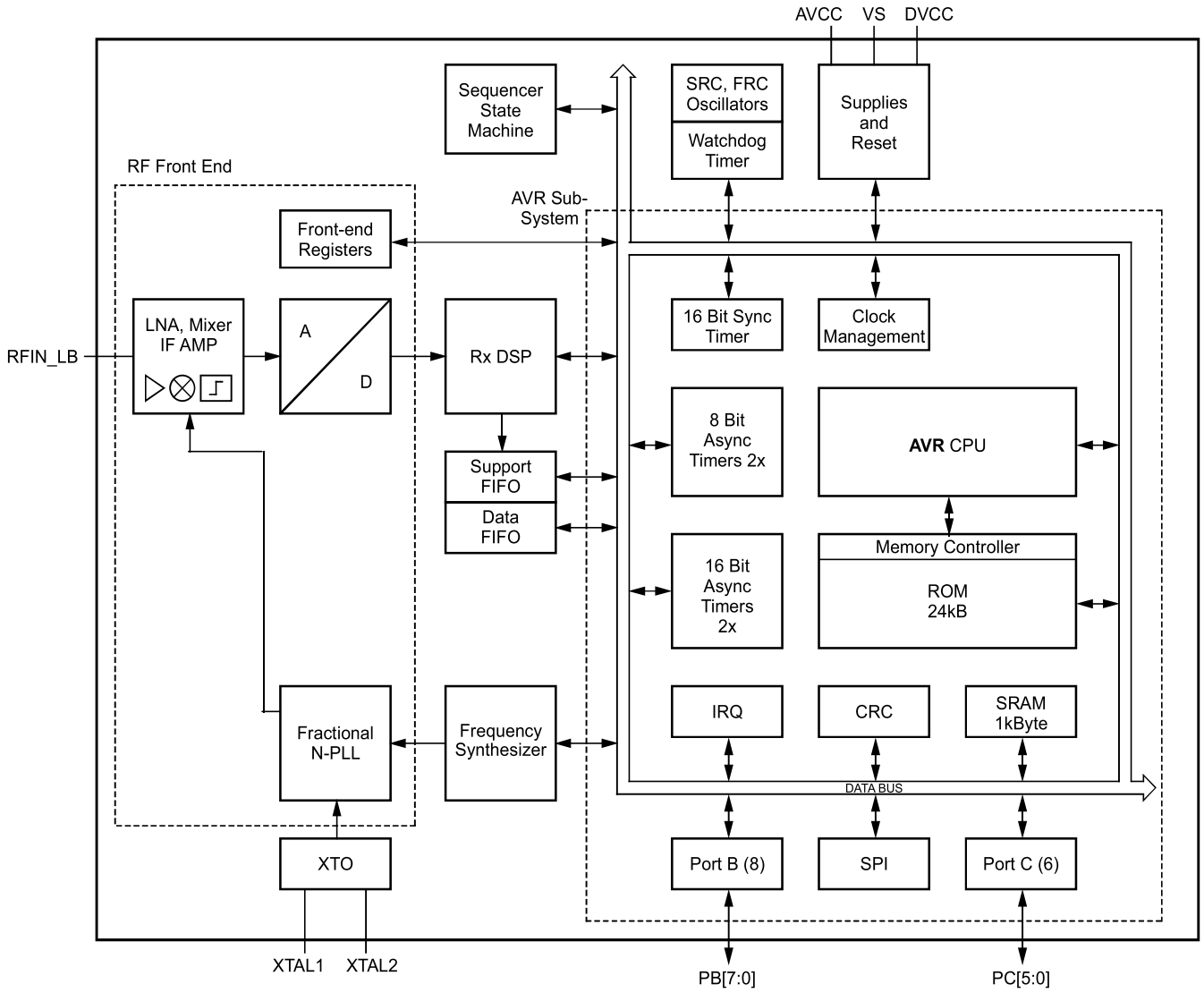
Bit 0: CRCstatus – status of SRAM CRC calculation
0 CRC calculation inactive
1 CRC calculation active

3. Hardware

3.1 Overview

The Atmel® ATA5785 consists of an analog front end, digital signal processing blocks (DSP), an 8-bit AVR® sub-system and various supply modules such as oscillators and power regulators. A hardware block diagram of the Atmel ATA5785 is shown in Figure 3-1.

Figure 3-1. Block Diagram



Together with the fractional-N PLL, the crystal oscillator (XTO) generates the local oscillator (LO) signal for the mixer in RXMode. The RF signal comes from the Low-Band input (RFIN_LB) and is amplified by the low-noise amplifier (LNA) and down-converted by the mixer to the intermediate frequency (IF) using the LO signal. A 10dB IF amplifier with low-pass filter characteristic is used to achieve enhanced system sensitivity without affecting blocking performance.

After the mixer, the IF signal is sampled using a high-resolution analog-to-digital converter (ADC).

Within the Rx digital signal processing (Rx DSP) the received signal from the ADC is filtered by a digital channel filter and demodulated. Two data receive paths, path A and path B, are included in the Rx DSP after the digital channel filter.

The system is controlled by an AVR® CPU with 24KB firmware ROM. A 1024-byte SRAM, and other peripherals are supporting the receiver handling. Two GPIO ports, PB[7:0] and PC[5:0], are available for external digital connections, for example, as an alternate function the SPI interface is connected to port B. The Atmel® ATA5785 is controlled by SPI commands and the functional behavior is mainly determined by firmware in the ROM. The firmware running on the AVR gives access to the hardware functionality of the Atmel ATA5785. The Rx DSP registers are addressed directly and accessible from the AVR via SPI commands. A set of sequencer state machines is included to perform Rx path operations (such as enable, disable, receive) which require a defined timing parallel to the AVR program execution.

The power management contains low-dropout (LDO) regulators and reset circuits for the supply voltages VS, AVCC, and DVCC of the Atmel ATA5785. In OFFMode all the supply voltages AVCC and DVCC are switched off to achieve very low current consumption. The Atmel ATA5785 can be powered up by activating the PWRON pin or one of the NPWRON[6:1] pins because they are still active in OFFMode. The AVCC domain can be switched on and off independently from DVCC. The Atmel ATA5785 includes two idle modes. In IDLEMode(RC) only the DVCC voltage regulator, the FRC and SRC oscillators are active and the AVR uses a power-down mode to achieve low current consumption. The same power-down mode can be used during the inactive phases of the PollingMode. In IDLEMode(XTO) the AVCC voltage domain as well as the XTO are additionally activated leading to short receiver start-up times.

An integrated watchdog timer is available to restart the Atmel ATA5785 when it is not served within the configured time-out period.

3.2 Crystal Oscillator

The XTO is an amplitude-regulated Pierce oscillator with internal load capacitances on the XTAL1 and XTAL2 pins of 14 ± 0.7pF (see parameter no. 14.10 in Section 5. “Electrical Characteristics” on page 300). The capacitance accuracy of ±0.7pF is determined by the FETN4.CTN[3:0] register. Due to additional board parasitics of about 1pF on each side, the total capacitance for C_{L1} and C_{L2} amounts to 15 ± 0.7pF and the load capacitance specification C_L for the crystal is 7.5pF.

The crystal oscillator is enabled if the RF front-end register FEEN1.XTOEN is set to “1”. Note that the AVCC voltage must be switched on with SUPCR.AVEN set to “1” and the starting time of the AVCC supply regulator taken into account.

After the crystal oscillator output amplitude has reached a defined level, the RF front-end status register FESR.XRDY bit is set to “1” by the XTO. The XTO clock is then available for the fractional-N PLL and the AVR.

The XTO oscillation frequency f_{XTO} is the reference frequency for the fractional N PLL. When designing the system in terms of receive and transmit frequency offset, the accuracy of the crystal and the XTO must be considered as well.

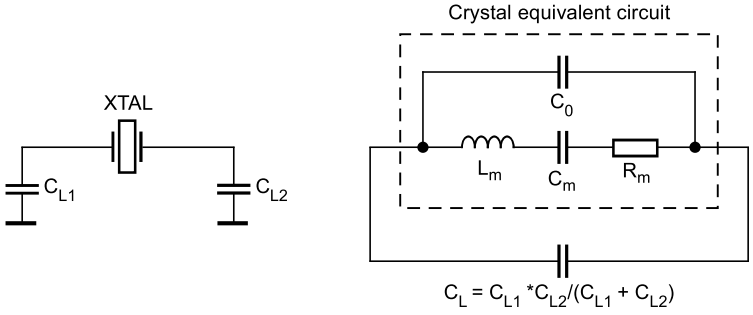
The additional pulling of the XTO is lower than ±10ppm for a motional capacitance of Cm = 4fF including the integrated load capacitors. If the initial tolerance is compensated by the fractional-N PLL, a pulling due to temperature and supply voltage of ±4ppm remains (see parameters no. 13.40 and 13.50 in Section 5. “Electrical Characteristics” on page 300).

The XTO frequency depends on the XTAL properties and the integrated load capacitances $C_{L1,2}$ at the XTAL1 and XTAL2 pins. The pulling P of the f_{XTO} from the nominal frequency f_{XTAL} is calculated using the following formula:

$$P = \frac{C_m}{2} \times \frac{C_{LN} - C_L}{(C_0 + C_{LN}) \times (C_0 + C_{LN})} \times 10^6 \text{ppm} \qquad (7)$$

C_m is the crystal motional capacitance, C_0 the shunt capacitance, and C_{LN} the nominal load capacitance of the XTAL indicated in the datasheet of the applied quartz. C_L is the total actual load capacitance of the crystal in the circuit and consists of C_{L1} and C_{L2} in series connection (for more information, see Figure 3-2 on page 90).

Figure 3-2. XTAL with Load Capacitance



For the Atmel® ATA5785 the pulling P amounts from –9.3ppm to +10.1ppm using the following values:

- C_m = 4fF
- C_0 = 1.0pF
- C_{LN} = 7.5pF
- $C_{L1,2}$ = 15pF ± 0.7pF

To ensure proper start-up behavior of the XTO, the small signal gain and the negative resistance provided by the XTO at start-up is very large. For example, the oscillation starts up even in a worst-case scenario with a crystal series resistance of 950Ω at C_0 ≤ 1pF. An approximation of the negative resistance seen by the crystal is:

$$\mathrm{Re}\{Z_{xtocore}\} = \mathrm{Re}\left\{\frac{Z_1 \times Z_3 + Z_2 \times Z_3 + Z_1 \times Z_2 \times Z_3 \times g_m}{Z_1 + Z_2 + Z_3 + Z_1 \times Z_2 \times g_m}\right\} \qquad (8)$$

Z_1, Z_2 are complex impedances at the XTAL1 and XTAL2 pins, therefore

$Z_1 = –j/(2 \times \pi \times f_{XTO} \times C_{L1}) + 5\Omega$ and $Z_2 = –j/(2 \times \pi \times f_{XTO} \times C_{L2}) + 5\Omega$

Z_3 consists of C_0 in parallel with an internal 180kΩ resistor, therefore

$Z_3 = –j/(2 \times \pi \times f_{XTO} \times C_0) \,//\, 180k\Omega$

gm is the internal transconductance between XTAL1 and XTAL2 with typically 40ms at 25°C during start-up.

With f_{XTO} = 24.305MHz, g_m = 40ms, C_L = 7.5pF, C_0 = 1pF, a typical negative resistance of about –1,300Ω can be reached. The worst case for technological, supply voltage, and temperature variations is for C_0 ≤ 1pF always better than –950Ω at T_{amb} = 105°C and –1,100Ω at T_{amb} = 85°C (see parameters no. 13.80 and 13.90 of the Section 5. “Electrical Characteristics” on page 300).

Due to the large gain at start-up, the XTO is able to achieve very high start-up speed. The oscillation start-up time can be estimated with the small signal time constant τ.

$$\tau = \frac{2}{4 \times \pi^2 \times f_m^{\,2} \times C_m \times (\left|\mathrm{Re}(Z_{xtocore})\right| - R_m)} \qquad (9)$$

After about 11τ an amplitude detector detects the oscillation amplitude and sets FESR.XRDY to “1” if the amplitude has reached a defined value.

Atmel recommends using a crystal with C_m = 4fF to 10fF, C_{LN} = 8pF, R_m < 110Ω and C_0 =1.0pF to 2pF.

Lower values of C_m increase the start-up time and influence the system timings, whereas higher values of C_m (up to 10fF) increase the pulling.

Higher values of C_0 reduce the negative resistance seen by the crystal. Care must therefore be taken on the additional PCB capacitance between XTAL1 and XTAL2 and a crystal with low C_0 should be chosen.

The Atmel ATA5785 crystal oscillator is a low-power design. A large start current is applied to the XTO core transistor enabling a start-up with up to 1100Ω cold-start resistance at T_{amb} = 85°C (see parameter no. 13.90 in Section 5. “Electrical Characteristics” on page 300). After a successful start-up, the current is reduced and the maximum achievable series resistance thus reduced to 110Ω (see parameter no. 14.00 of the Section 5. “Electrical Characteristics” on page 300).

In summary, the following front-end register settings are affected by the XTO:

- FEEN1. XTOEN: Enables/disables the XTO.
- FESR.XRDY: Indicates that the XTO is ready to be used as system clock source.
- FETN4.CTN4[3:0]: Calibration value for internal XTO load capacitance.

3.3 Fractional-N PLL

3.3.1 Overview

Figure 3-3. Fractional-N PLL

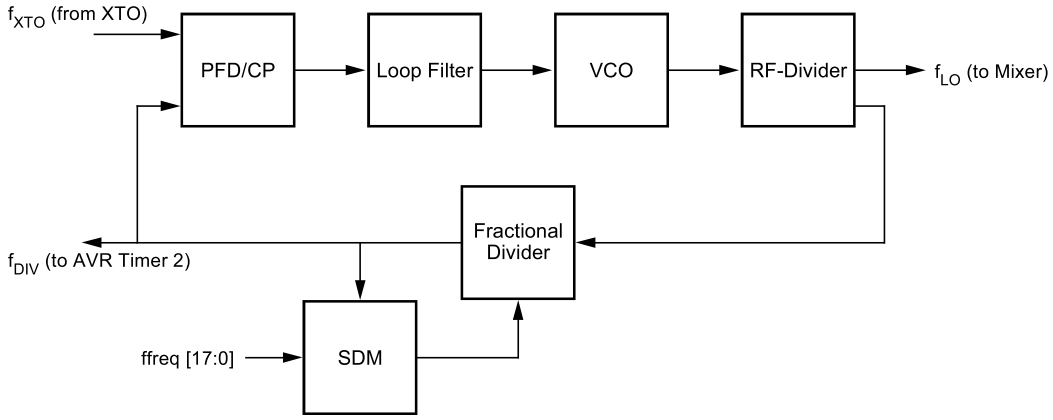


Figure 3-3 shows the block diagram of the fractional-N PLL. It consists of a phase frequency detector (PFD), a charge pump (CP), a loop filter (LF), a voltage controlled oscillator (VCO), an RF divider, a fractional divider, and a sigma delta modulator (SDM). The reference frequency f_{XTO} which is used for the PLL is the output of the XTO.

In RXMode the PLL is used to generate the local oscillator signal with the frequency of f_{LO} for the mixer. The PLL is able to cover the frequency bands of 310MHz to 318MHz and 418MHz to 477MHz (Low-Band).

The tolerances of the PLL are minimized by means of calibration. The dynamic transfer function of the PLL is thus very stable and the locking time of the PLL is very constant. During start-up an additional speedup mode can be activated (register bit FEEN1.PLSP1) to achieve shorter locking times.

A lock detect circuit, enabled with the register bit FECR.PLCKG, delivers a lock status in the FESR.PLCK register bit. Based on this information the software can ensure that the Atmel ATA5785 does not start up a receive mode if the PLL is not locked.

A sigma delta modulator (SDM) is used to dither the fractional spurious in a way that the resulting pseudo random noise is below the phase noise of the VCO. The frequency resolution of the output frequencies (f_{LO}) is $f_{xto} / 2^{18}$ = 92.7Hz when using a 24.3MHz crystal.

3.3.2 PFD/CP and Loop Filter

The PFD/CP and the loop filter compare the output frequency of the fractional divider f_{DIV} with the reference frequency f_{XTO} in phase domain and adjust the control voltage of the VCO resulting in $f_{DIV} = f_{XTO}$.

To obtain a stable dynamic behavior of the PFD/CP and loop filter over production the following preprogrammed calibration values must be set before the analog PLL loop is used: FEBT.RTN2[1:0], FEBT.CTN2[1:0], FETN4.CTN4[3:0], and FETN4.RTN4[3:0], which control the resistors and capacitors in the loop filter. In addition, the complete loop is controlled by the charge pump current using FEVCO.CPCC[3:0] (for more information, see Table 3-2 on page 94).

Two parameters can be used to calculate the FEVCO.CPCC[3:0] value. The first parameter is an internal calibration value which compensates for the process tolerances. The second one is calculated based on the actual output frequency for each service. The Atmel configuration tool should be used to calculate the necessary settings for each service in use.

3.3.3 VCO

The VCO is designed to cover a frequency range of 1,672MHz to 1,912MHz to accommodate all operating frequency bands of the Atmel® ATA5785.

To support this high frequency range two digital settings of the VCO have to be set before the oscillator can be used in the analog PLL loop. These settings are the amplitude control FEVCO.VCOB[3:0] and the frequency control FEVCT.FEVCT[3:0].

The VCO amplitude is determined by FEVCO.VCOB[3:0]. Two parameters are used to calculate the FEVCO.VCOB[3:0] value. The first parameter is an internal calibration value which compensates the process tolerances of the VCO. The second one is dependent on the actual output frequency for each service. The Atmel configuration tool should be used to calculate the required settings.

The content of the FEVCT[3:0] register is determined in an automated frequency tuning loop. The activation of the VCO tuning process is described in Section 2.9.4 “VCO Tuning” on page 53. The processing time for the calibration routine is about 150µs. This calibration needs to be executed if the temperature changes by more than 40K or if the VCO frequency changes by more than 25MHz.

3.3.4 RF Divider

The RF divider delivers the local oscillator signal to the mixer in RXMode by dividing the VCO output frequency by 2, 4, or 6. It divides by 6 for 310MHz to 318MHz and by 4 for 418MHz to 477MHz. The RF divider is controlled by the FECR.S4N3 control bit.

3.3.5 Fractional-N Divider and SDM

A detailed description of the fractional-N divider and SDM is found in Section 4.1.2 “Fractional-N Divider and SDM” on page 291.

3.3.6 Fractional-N PLL Register Overview

The following registers are used by the PLL. The RF front-end registers from Table 3-1 on page 93 should not be changed by application software. They are only used by firmware to control the start-up, power-down and calibration of the fractional-N PLL.

Table 3-1. RF Front-End Registers Used by Firmware

Register	Function
FESR.PLCK	PLL locked
FEEN1.PLSP1	PLL speedup
FEEN1.PLCAL	PLL calibration mode
FEEN1.PLEN	PLL enable
FEEN2.PLPEN	PLL post enable
FECR.PLCKG	PLL lock detect gate
FEVCT.FEVCT[3:0]	RF front-end VCO tuning register

The RF front-end values, summarized in Table 3-2 are calibration values which are transferred to the RF front end by firmware before using the PLL.

Table 3-2. RF Front-End Registers for Calibration

Register	Function
FEBT.RTN2[1:0]	Resistor tuning
FEBT.CTN2[1:0]	Capacitor tuning
FETN4.CTN4[3:0]	Capacitor tuning
FETN4.RTN4[3:0]	Resistor tuning
FEVCO.VCOB[3:0]	VCO bias 4-bit value [1)]
FEVCO.CPCC[3:0]	Charge pump current control [1)]

Note: 1. For this register information a calibration value and an application-specific value are used. The register information must be copied to the RF front-end registers before using the PLL.

The RF front-end registers from Table 3-3 are changed due to application-specific settings. The Atmel configuration tool should be used to calculate these settings. The FFREQ settings can be modified there to compensate both the initial XTO and the XTAL frequency tolerances. The exact frequency steps of the PLL ($f_{xto}/2^{18}$ = 92.7Hz in Low-Band frequency range when using a 24.3MHz crystal) have to be considered.

Table 3-3. RF Front-End Registers Used for Application Settings

Register	Function
FEMS.M[3:0]	Main counter
FEMS.S[3:0]	Swallow counter
FECR.S4N3	Select 433MHz/315MHz band
FEVCO.VCOB[3:0]	VCO bias [1)]
FEVCO.CPCC[3:0]	Charge pump current control [1)]

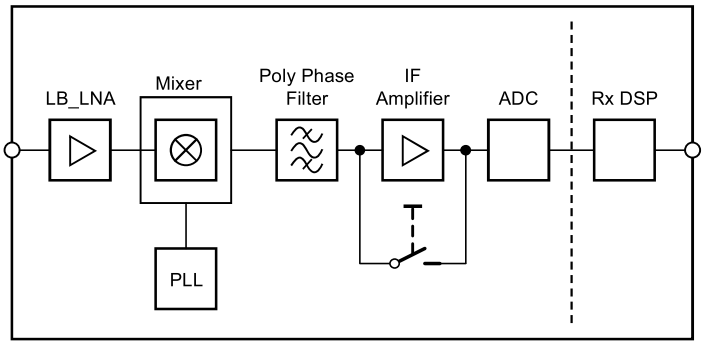
Note: 1. For this register information a calibration value and an application-specific value are use. The register information must be copied to the RF front-end registers before using the PLL.

3.4 Receive Path

3.4.1 Overview

The receive path consists of a low-noise amplifier (LNA), mixer, IF amplifier, analog-to-digital converter (ADC), and an Rx digital signal processor (Rx DSP) as shown in Figure 3-4. The fractional-N PLL and the XTO deliver the local oscillator frequency in RXMode. The receive path is controlled by the RF front-end registers.

Figure 3-4. Rx Path Overview



A highly optimized low-band LNA is provided to obtain optimum performance matching for the frequency ranges 310MHz to 318MHz and 418MHz to 477MHz.

The mixer converts the received RF signal to a low intermediate frequency (IF) of about 250kHz. A double-quadrature architecture is used for the mixer to achieve high image rejection. Additionally, the third-order suppression of the local oscillator (LO) harmonics makes receiving without a front-end SAW filter less critical.

An IF amplifier provides additional gain and improves the receiver sensitivity by 2-3dB. Because of built-in filter function, the in-band compression is degraded by 10dB, while the out-of-band compression remains unchanged.

The ADC converts the IF signal into the digital domain. Due to the high effective resolution of the ADC, the channel filter and received signal strength indicator (RSSI) can be realized in the digital signal domain. Therefore, no analog gain control (AGC) potentially leading to critical timing issues or analog filtering is required in front of the ADC. This leads to a receiver front end with excellent blocking performance up to the 1dB compression point of the LNA and mixer, and a steep digital channel filter can be used.

The Rx DSP performs the channel filtering and converts the digital output signal of the ADC to the baseband for demodulation. Due to the digital realization of these functions the Rx DSP can be adapted to the needs of many different applications. Channel bandwidth, data rate, modulation type, wake-up criteria, signal checks, clock recovery, and many other properties are configurable. The RSSI value is realized completely in the digital signal domain, enabling very high relative and absolute accuracy that is only deteriorated by the gain errors of the LNA, mixer, and ADC.

Two independent receive paths A and B are integrated in the Rx DSP after the channel filter and allow the use of different data rates, modulation types, and protocols without the need to power up the receive path more than once to decide which signal should be received. This results in a reduced polling current in several applications.

The integration of remote keyless entry (RKE), passive entry and go (PEG) and tire-pressure monitoring systems (TPM) into one module is simplified because completely different protocols can be supported and a low polling current is achieved.

3.4.2 Analog Receive Path

3.4.2.1 LNA and Mixer

An optimized LNA is integrated in the Atmel® ATA5785 for Low-Band ranging from 310MHz to 477MHz which is connected to the 1/RFIN_LB pin. Table 3-4 shows the setting of the RF front-end registers used for the different RF frequency bands.

The combined LNA and mixer gain for the Low-Band is calibrated using the FELNA.LBL[3:0] RF front-end register. The calibration data compensate for the process tolerances in the gain of the LNA and the mixer. The values must be written to the RF front-end registers before powering up the receive path.

Table 3-4. Used Bias for the LNAs

Frequency Band [MHz]	FECR.S4N3	Used LNA Bias
310.00 … 318.00	0	FELNA_2.LBL_315[3:0]
418.00 … 477.00	1	FELNA_1.LBL_433[3:0]

3.4.2.2 IF Amplifier

A 10dB IF amplifier between the RF front end and the ADC input results in a 2-3dB improvement of the system sensitivity. The low-pass filter characteristic of this amplifier separates in-band and out-of-band compression. This results in enhanced system sensitivity without affecting blocking performance.

3.4.2.3 ADC

The sigma delta ADC converts the IF signals into the digital signal domain by using a clock coming from the fractional-N PLL.

The ADC time constants are calibrated using the RF front-end registers FEBT.RTN2[1:0] and FEBT.CTN2[1:0]. These parameters compensate for the process tolerances of the ADC. The values must be written to the registers before powering up the receive path.

Depending on the RF frequency applied, different sampling rates are used within the ADC and the RF front-end register FECR.ADHS can be enabled for higher sampling rates. The register must be set before activating the receive path. The Atmel configuration tool should be used to calculate the necessary settings.

3.4.2.4 Registers Involved In Analog Receive Path

The following registers are used by the analog receive path consisting of LNA, mixer, and ADC. See also Section 3.6 “RF Front-End Register Description” on page 171.

The RF front-end registers from Table 3-5 should not be changed by application software. They are used by firmware to control the power-up and power-down of the receive path.

Table 3-5. RF Front-End Registers Used by Firmware

Register	Function
FEEN1.ADEN	Analog-to-digital converter enable
FEEN1.ADCLK	Analog-to-digital converter clock enable
FEEN1.LNAEN	Low-noise amplifier enable

The RF front-end values summarized in Table 3-5 are transferred to the RF front end by firmware before powering-up the receive path.

Table 3-6. RF Front-End Registers for Calibration

Register	Function
FELNA.LBL[3:0]	LNA bias Low-Band
FEBT.RTN2[1:0]	Resistor tuning
FEBT.CTN2[1:0]	Capacitor tuning

The RF front-end registers from Table 3-7 include application-specific settings. The Atmel configuration tool should be used to calculate these settings.

Table 3-7. RF Front-End Registers Used for Application Settings

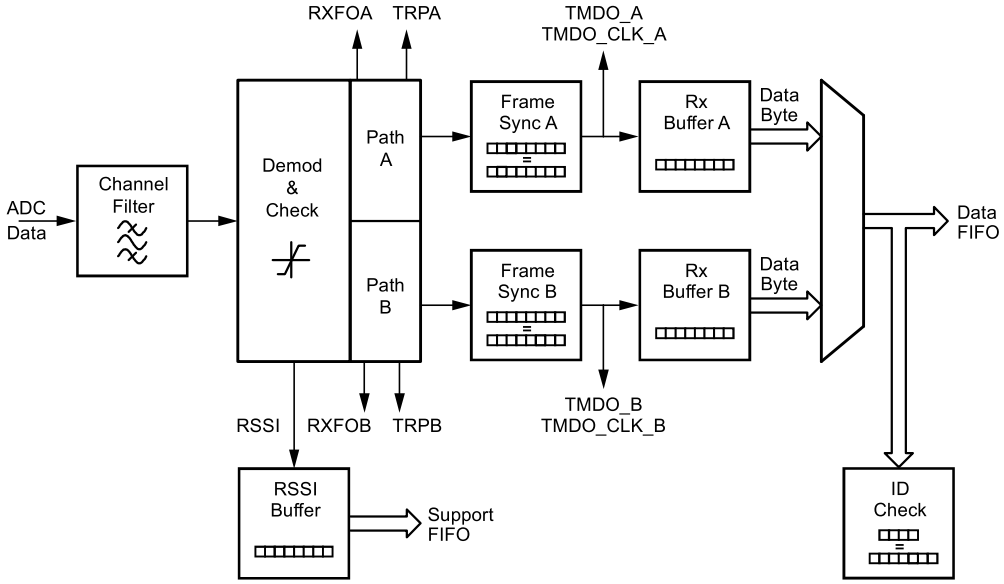
Register	Function
FECR.ADHS	Analog-to-digital converter high sample rate
FECR.S4N3	Select 433MHz not 315MHz band

3.4.3 Rx Digital Signal Processing (Rx DSP)

The Rx digital signal processing (DSP) block performs the digital filtering, decoding, checking, and byte-wise buffering of the Rx samples that are derived from the ADC as shown in Figure 3-5. The Rx DSP provides the following outputs:

- Raw demodulated data at the TRPA/B pins
- Decoded data at the TMDO and TMDO_CLK pins
- Buffered data bytes toward the data FIFO and ID check block
- Auxiliary information about the signal such as the received signal strength indication (RSSI) and the frequency offset of the received signal from the selected center frequency (RXFOA/B)

Figure 3-5. Rx DSP Overview



The channel filter determines the receiver bandwidth. Its output is used for both receiving paths A and B, making it necessary to configure the filter to match both paths. The receiving paths A and B are identical and consist of an ASK/FSK demodulator with attached signal checks, a frame synchronizer which supports pattern-based searches for the telegram start and a 1-byte hardware buffer with integrated CRC checker for the received data.

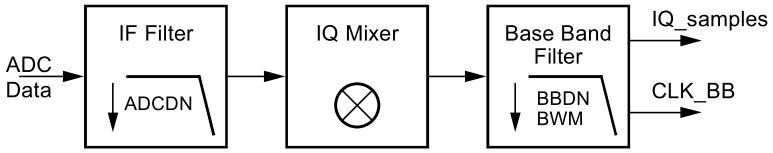
Depending on the signal checks, one path is selected which writes the received data to the data FIFO and optionally to the ID check block.

The RSSI values are determined by the demodulator and written via the RSSI buffer to the support FIFO where the latest 16 values are stored for further processing.

3.4.3.1 Channel Filter

The channel filter performs a decimation filtering (IF filter) of the ADC samples, mixing of the intermediate frequency signal (IQ Mixer) to baseband and finally low-pass filtering (baseband filter) of the result, providing a high resolution output (IQ_samples) and a processing clock (CLK_BB) for the demodulator. A corresponding block diagram is shown in Figure 3-6.

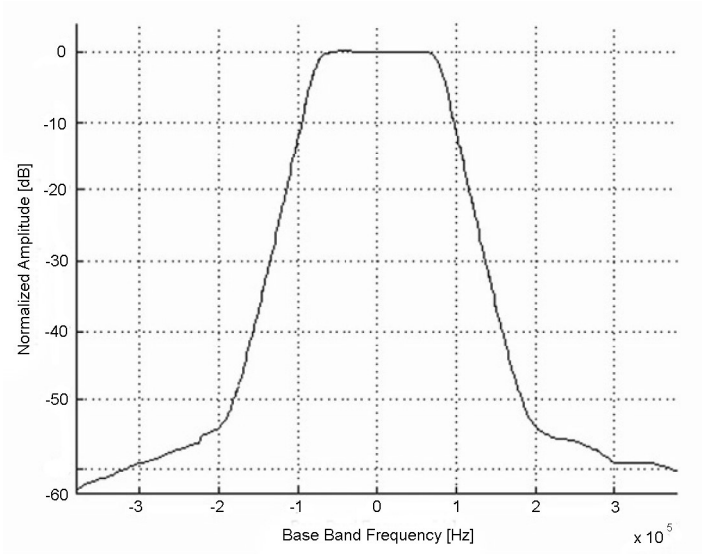
Figure 3-6. Channel Filter Block Diagram



The correct decimation for each RF frequency band is selected by the CHDN.ADCDN register. The receiving bandwidth is selected by configuring the CHDN.BBDN and CHCR.BWM registers. Table 3-8 on page 99 describes the correct settings for these registers for each bandwidth.

A typical frequency response of the channel filter including the phase noise effect is shown in Figure 3-7.

Figure 3-7. Typical Channel Filter Frequency Response for the 165kHz Bandwidth



Channel Filter Bandwidths

The register bits CHDN.BBDN and CHCR.BWM are used to select the desired bandwidth. Table 3-8 shows the typical bandwidths for an RF signal (f_{RF}) of 433.92MHz with the corresponding register settings.

Table 3-8. Filter Settings for Each Supported Bandwidth

No	$BW_{-3dB,\ 433.92}$ –3dB Bandwidth (f_{RF} = 433.92MHz)	–30dB Bandwidth (f_{RF} = 433.92MHz)	BBDN	BWM	Resulting $CLK\_BB_{433.92}$ Frequency
1	25kHz	41kHz	20	0	75.3kHz
2	27kHz	43kHz	19	0	79.3kHz
3	29kHz	49kHz	17	0	88.6kHz
4	33kHz	55kHz	15	0	100.4kHz
5	35kHz	59kHz	14	0	107.6kHz
6	41kHz	69kHz	12	0	125.5kHz
7	45kHz	75kHz	11	0	136.9kHz
8	50kHz	83kHz	10	0	150.6kHz
9	55kHz	92kHz	9	0	167.3kHz
10	61kHz	103kHz	8	0	188.2kHz
11	71kHz	117kHz	7	0	215.1kHz
12	80kHz	139kHz	7	4	215.1kHz
13	93kHz	151kHz	7	8	215.1kHz
14	99kHz	167kHz	5	1	301.2kHz
15	110kHz	193kHz	5	5	301.2kHz
16	123kHz	205kHz	4	1	376.5kHz
17	134kHz	234kHz	4	5	376.5kHz
18	146kHz	177kHz	5	12	301.2kHz
19	165kHz	270kHz	3	2	502.0kHz
20	185kHz	328kHz	3	6	502.0kHz
21	219kHz	351kHz	3	9	502.0kHz
22	237kHz	368kHz	2	3	753.0kHz
23	243kHz	295kHz	3	13	502.0kHz
24	276kHz	474kHz	2	7	753.0kHz
25	325kHz	494kHz	2	10	753.0kHz
26	366kHz	446kHz	2	14	753.0kHz

Bandwidth Scaling

The channel filter is realized as a digital filter and the sampling rate depends on the RF signal. It is therefore subject to scaling with the RF signal and the corresponding sampling clock (CLK_BB). This scaling effect can modify the actual –3dB and –30dB bandwidth in the range of –5.6% to +3.6% versus the typical bandwidths given for f_{RF} = 433.92MHz in Table 3-8.

If the target application frequency f_{RF} differs from 433.92MHz in Table 3-8, the following procedure can be used to calculate the resulting application bandwidth $BW_{-3dB,APPL}$:

1. The target division factor (DIV_IF) is selected from Table 3-9 on page 100.

Table 3-9. Frequency Dependent Division Factor (DIV_IF)

No.	f_{RF}	DIV_IF
1	310MHz to 318MHz	204
2	418MHz to 440MHz	288
3	441MHz to 477MHz	306
4	Reserved	Reserved
5	Reserved	Reserved

2. The target (–3dB) bandwidth from Table 3-8 on page 99 is selected and the corresponding BBDN and $CLK\_BB_{433.92}$ are remembered.
3. The application sampling clock CLK_BB_{APPL} for the target application frequency $f_{RF,APPL,}$ is calculated using

$$CLK\_BB_{APPL} = \frac{f_{RF}}{DIV\_IF \times BBDN} \quad (10)$$

4. The resulting frequency-dependent scaling factor is calculated using

$$SFRQ_{APPL} = \frac{CLK\_BB_{APPL}}{CLK\_BB_{433.92}} \quad (11)$$

5. Apply the scaling to the typical bandwidth

$$BW_{-3dB,APPL} = BW_{-3dB,433.92} \times SFRQ_{APPL} \quad (12)$$

6. If the resulting BW_{APPL} is too narrow for the target data rate, deviation and crystal tolerances, the procedure should be restarted at the next higher bandwidth.

Usage Example

The application has a target application frequency of f_{RF} = 316MHz and requires a bandwidth of 366kHz.

1. DIV_IF = 205 is taken from line 1 of Table 3-9
2. BBDN = 2 and $CLK\_BB_{433.92}$ = 753.0kHz are taken from line 26 of Table 3-8 on page 99
3. The application sampling clock is calculated using

$$CLK\_BB_{APPL} = \frac{316MHz}{204 \times 2} = 774.5kHz$$

4. The resulting scaling is calculated using

$$SFRQ_{APPL} = \frac{774.5kHz}{753.0kHz} = 1.029$$

5. Apply the scaling to the typical bandwidth

$BW_{-3dB,APPL}$ = 366kHz ×1.029 = 376.6kHz

Channel Filter Register Description

The channel filter bandwidth is configured by selecting the appropriate down-sampling register (CHDN) and operating mode register (CHCR).

CHDN – Channel Filter Down-Sampling Register

This register should only be modified if the channel filter is disabled (RDPR.PRFLT=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	-	0	ADCDN	BBDN[4:0]					CHDN
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: Reserved for future use, read as 0

Bit 6: Reserved. Has to be 0 during operation. Otherwise the RSSI value is corrupted.

Bit 5: ADCDN - ADC down-sampling configuration

Table 3-10. ADCDN Configuration

ADCDN	Description
0	Use for RF bands: 418MHz to 440MHz
1	Use for RF bands: 310MHz to 318MHz, 441MHz to 477MHz

Bits 4..0: BBDN - Baseband Filter Down-Sampling Ratio

The required setting for each supported bandwidth is found in Table 3-8 on page 99. Valid settings for BBDN are 2 to 20. All other values are forbidden and might result in flawed circuit behavior.

CHCR – Channel Filter Configuration Register

This register should be modified only if the channel filter is disabled (RDPR.PRFLT=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	BWM[3:0]				CHCR
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..4: Reserved for future use, read as 0

Bits 3..0: BWM – Bandwidth mode. Adapts the filter frequency transfer function to the application. The required setting for each supported bandwidth is found in Table 3-8 on page 99.

3.4.3.2 Demodulator and Signal Checks

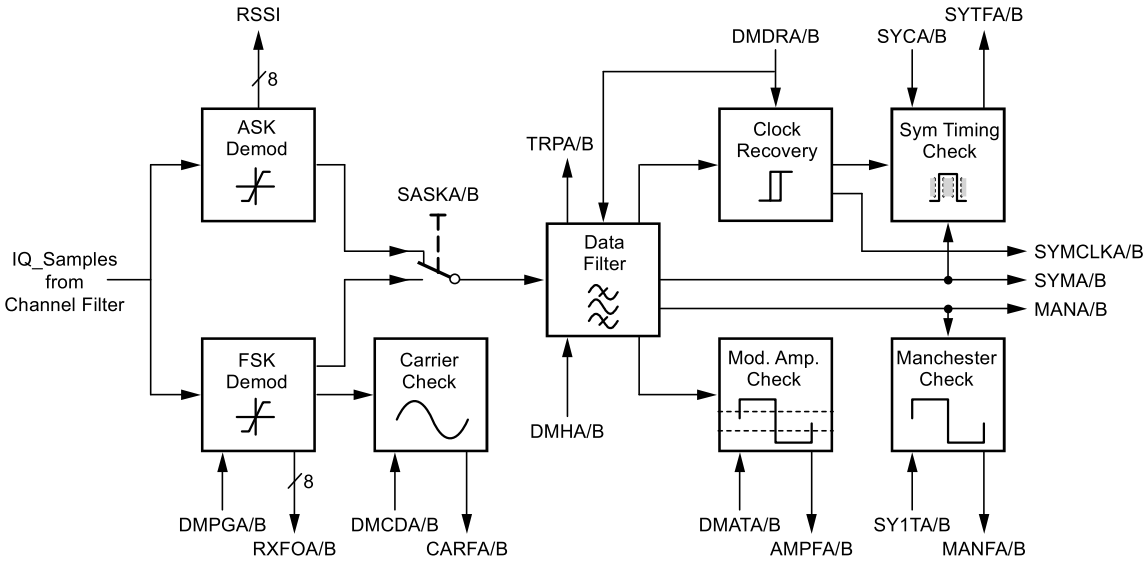
This section describes the demodulators and the signal checks for the receiving path A and path B. Path B is an identical copy of path A. A/B at the ending of a signal name indicates this signal is present in both paths. For example, RXFOA/B means there is an RXFOA signal in path A and an RXFOB signal in path B.

The internal states of the receiving paths A/B are reset by disabling the complete Rx DSP (RDCR.RDEN=0), by activating the power reduction for the channel filter (RDPR.PRFLT=1) or by activating only the power reduction for the receiving A/B path (RDPR.PRPTA/B=1).

An ASK and an FSK demodulator are available in each path. Only one of them is used for further processing. The relevant demodulator is chosen by setting the DMCRA/B.SASKA/B bit accordingly.

If not otherwise specified, a detailed description of the required register bits is found in Section "Demodulator Register Description" on page 105.

Figure 3-8. Demodulator and Clock Recovery for Path A or Path B



ASK Demodulation

ASK demodulation is achieved by calculating the magnitude of the complex IQ_samples from the channel filter in a logarithmic scale. Before the signal goes to the filters it is limited to an amplitude range of 23dB from the peak value. The ASK demodulator tolerates fading of up to 1dB per bit.

The ASK demodulator additionally provides the raw RSSI signal which is further processed in the RSSI buffer (see Section 3.4.3.6 “RSSI Buffer” on page 139).

FSK Demodulation

The demodulator is based on a digital PLL. The demodulating process starts with the carrier frequency detection of the useful signal, which is (assumed to be) the dominant frequency within the spectrum in the channel filter.

The demodulator is centered using the measured frequency offset to the desired frequency. To do this, the time constant of the regulation loop is changed in stepwise increments. The required final effective bandwidth depends on the frequency deviation and data rate of the signal to be received and is set in the DMCRA/B.DMPGA/B register bits. The effective noise bandwidth for the demodulation process is therefore less than the receiver bandwidth, resulting in a higher CNR.

The maximum deviation supported by the FSK demodulator is limited by the internal sampling rate. See Table 3-11 on page 103 for limit values.

Received Frequency Offset

The offset of the actually received RF signal frequency versus the expected RF signal frequency is provided by the FSK demodulators for path A and path B. It can be read out during reception from the RXFOA/B register.

Note: This function is only available if the related path has been set to FSK.

Carrier Check

The carrier check provides a means for verifying if a carrier signal or only noise is present at the RF input. The block checks the phase difference at the phase comparator of the PLL for each input sample. The phase difference has to stay within ±90° to be valid. A phase jump outside of these boundaries indicates the dominance of noise or an unexpected signal. An error is flagged when the number of violations within a defined time interval exceeds a predefined threshold.

The severity of the signal check is determined by two settings:

- The length of the time interval where the phase jump violations are accumulated is set by the DMCDA/B.DMCTA/B register bits. The counter is reset at the end of each time window and the counting restarted.
- The threshold for phase jump violations not triggering an error is set by the DMCDA/B.DMCLA/B register bits.

The resulting EOTSA/B.CARFA/B flag indicates the failure of the carrier check, see Section "Rx DSP Control Register Description" on page 146. The SOTSA/B.CAROA/B success flag is set if the number of symbols defined by SYCA/B.SYCSA/B is received at the beginning of a telegram without the occurrence of this error.

Data Filter

The data filter contains two matched filters. One for symbol-based NRZ reception and one for Manchester-coded data. One of them can be selected for data reception by setting the DMMA/B.DMNEA/B bits.

The data filters require target data rate information for proper operation. The data rate is set by configuring the DMDRA/B registers accordingly. The data rates can be set independently for path A and path B.

The matched symbol filter generates the TRPA/B outputs and the corresponding SYMA/B signals (see Figure 3-8 on page 102) for internal use. The symbol filter contains a feedback loop for DC removal.

A matched filter for Manchester-coded data is provided for improved sensitivity. The filter generates the internal MANA/B signals.

The maximum symbol rate and deviation is limited by the sampling rate of the demodulator (CLK_BB) and the data filter. The sampling rate is determined by the selected channel filter bandwidth. The typical limits for an RF signal of 433.92MHz are shown in Table 3-11. The limits for a certain application are subject to RF signal-dependent scaling and can differ from these values by –5.6% to +3.6%. The actual limits can be calculated by multiplying the maximum symbol rate and the maximum deviation from Table 3-11 with the scaling factor ($SFRQ_{APPL}$) derived for the bandwidth scaling (see Section "Channel Filter Bandwidths" on page 99, calculation step 4).

Table 3-11. Maximum Symbol Rate and Deviation versus Bandwidth for f_{RF} = 433.92MHz

No	$BW_{-3dB,\ 433.92}$ –3dB Bandwidth (f_{RF} = 433.92MHz)	Resulting $CLK\_BB_{433.92}$	Maximum Symbol Rate (Kilo Baud)	Maximum Deviation
1	25kHz	75.3kHz	16.5	9.4kHz
2	27kHz	79.3kHz	17.3	9.9kHz
3	29kHz	88.6kHz	19.4	11.1kHz
4	33kHz	100.4kHz	22.0	12.5kHz
5	35kHz	107.6kHz	23.5	13.4kHz
6	41kHz	125.5kHz	27.4	15.7kHz
7	45kHz	136.9kHz	29.9	17.1kHz
8	50kHz	150.6kHz	32.9	18.8kHz
9	55kHz	167.3kHz	36.6	20.9kHz
10	61kHz	188.2kHz	41.2	23.5kHz
11	71kHz	215.1kHz	47.1	26.9kHz
12	80kHz	215.1kHz	47.1	26.9kHz
13	93kHz	215.1kHz	47.1	26.9kHz
14	99kHz	301.2kHz	65.9	37.6kHz
15	110kHz	301.2kHz	65.9	37.6kHz
16	123kHz	376.5kHz	82.3	47.1kHz
17	134kHz	376.5kHz	82.3	47.1kHz
18	146kHz	301.2kHz	65.9	37.6kHz
19	165kHz	502.0kHz	109.8	62.7kHz
20	185kHz	502.0kHz	109.8	62.7kHz
21	219kHz	502.0kHz	109.8	62.7kHz
22	237kHz	753.0kHz	160.0	94.1kHz
23	243kHz	502.0kHz	109.8	62.7kHz
24	276kHz	753.0kHz	160.0	94.1kHz
25	325kHz	753.0kHz	160.0	94.1kHz
26	366kHz	753.0kHz	160.0	94.1kHz

Modulation Amplitude Check

An incoming signal is expected to have a modulation. This check verifies the modulation amplitude at the output of the data filter. The minimum for the expected signal modulation amplitude can be set in the DMMA/B.DMATA/B register. An error is flagged if the signal fails to exceed the threshold within two data symbols for alternating preamble pattern (DMCRA.SY1TA/B=”1”) or three symbols for flexible preamble pattern (DMCRA.SY1TA/B=”0”).

The EOTSA/B.AMPFA/B register bit is set to indicate the failure of this check, see Section “Rx DSP Control Register Description” on page 146.

The SOTSA/B.AMPOA/B success flag is set if the number of symbols defined by SYCA/B.SYCSA/B passes at the beginning of a telegram without the occurrence of this error.

Clock Recovery

The clock recovery provides a sampling clock for the incoming data and the clock output for the TMDO_CLK pin. It is a DPLL-based circuit capable of being adapted to data rate tolerances and to ignore individual bit errors. The DPLL is more tolerant at power-up (telegram preamble) and thus allows for a wider data rate range. Within a few signal edges the DPLL locks to the actual data rate and reduces the flexibility, enhancing noise immunity.

The initial data rate setting is obtained from the DMDRA/B register.

Symbol Timing Check

The symbol timing check is concerned with the timing of the signal transition from one symbol to the next. The edges of the symbols are expected to stay within some boundaries relative to the internal clock recovery. There are two settings for this check:

- The timing limit (SYCA/B.SYTLA/B) defines a validity window for the symbol edge relative to its expected location.
- The symbol check size (SYCA/B.SYCSA/B) defines the number of symbols that have to pass without an error before OK flags indicate the success of several signal checks. This is the precondition for marking an incoming signal as a valid telegram by setting the RDSIFR.SOTA/B flags to high (see Section “Rx DSP Control Register Description” on page 146).

The OK flags affected by this setting are:

- Carrier OK (SOTSA/B.CAROA/B)
- Modulation amplitude OK (SOTSA/B.AMPOA/B)
- Symbol timing OK (SOTSA/B.SYTOA/B)
- Manchester check OK (SOTSA/B.MANOA/B)

The EOTS.SYTFA/B bit is set to indicate the failure of this check, see Section “Rx DSP Control Register Description” on page 146.

The SOTSA/B.SYTOA/B success flag is set if the number of symbols defined by SYCA/B.SYCSA/B (see Section “SYCA – Symbol Check Configuration for Path A” on page 118) passes at the beginning of a telegram without the occurrence of this error.

Manchester Check

The Manchester check verifies that the incoming modulation conforms with Manchester coding.

The only setting for this check is the activation of a more severe mode for the telegram start where only alternating high and low symbols are accepted as valid by the (DMCRA/B.SY1TA/B) register.

The EOTS.MANFA/B bit is set to indicate the failure of this check (for more information, see Section “Rx DSP Control Register Description” on page 146).

The SOTSA/B.MANOA/B success flag is set if the number of symbols defined by SYCA/B.SYCSA/B passes at the beginning of a telegram without the occurrence of this error.

Demodulator Register Description

RXFOA – Received Frequency Offset versus Intermediate Frequency on Path A (FSK Only)

The RXFOA register provides the frequency difference between the selected receive frequency and the currently received signal on path A.

This register is updated continuously if the Rx path A is working. The register is held in the reset state if path A is disabled (RDPR.PRFLT=1 or RDPR.PRPTA=1) or the complete Rx DSP is turned off (RDCR.RDEN=0).

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

Bits 7..0: RXFOA – Received Frequency Offset.

The values are two's complement signed integers.

The carrier frequency offset versus intermediate frequency can be calculated as:

$$f\_offset\_A = \frac{twos\_complement2dec(RXFOA) \times CLK\_BB}{256} \qquad (13)$$

RXFOB – Received Frequency Offset versus Intermediate Frequency on Path B (FSK Only)

The RXFOB register provides the frequency difference between the selected receive frequency and the currently received signal on path B.

This register is updated continuously if the Rx path B is working. It is held in the reset state if path B is disabled (RDPR.PRFLT=1 or RDPR.PRPTB=1) or the complete Rx DSP is turned off (RDCR.RDEN=0).

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

Bits 7..0: RXFOB – Received Frequency Offset.

The values are two's complement signed integers.

The carrier frequency offset versus intermediate frequency can be calculated as:

$$f\_offset\_B = \frac{twos\_complement2dec(RXFOB) \times CLK\_BB}{256} \qquad (14)$$

DMDRA – Demodulator Data Rate on Path A

The target symbol rate for the receiving path A is configured in this register.

This register should only be modified if the block receiving the settings is disabled (RDPR.PRPTA=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMDNA[3:0]				DMAA[3:0]				DMDRA
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..4: DMDNA – Demodulator Down-Sampling on Path A

Selects the down-sampling and cut-off frequency of the first stage of the data filter.

Table 3-12. DMDNA Setting versus Amplitude Threshold Scaling on Path A (ATSCAL_A)

DMDNA	ATSCAL_A
0	1
1	1
2	2
3	2
4	4
5	4
6	8
7	8
8	16

The recommended down-sampling for a target symbol rate is:

$$DMDNA = floor\left[ld(20) + ld\left(\frac{CLK\_BB}{SymbolRatePathA \times 128}\right)\right] \qquad (15)$$

If the result is –1 set DMDNA=0

Equation parameters:

SymbolRatePathA	Maximum expected symbol rate in Hz on the useful signal for path A
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section "Bandwidth Scaling" on page 99.
ld()	Logarithm to the base 2

Bits 3..0: DMAA – Demodulator Moving Average Data Rate Factor on Path A

Selects the moving average filter length for the data filter and initial data rate for the clock recovery on path A.

Table 3-13. DMAA Setting versus Moving Average Filter Length and Symbol Rate Factor on Path A (MAVFL_A, SR_F_A)

DMAA	MAVFL_A	SR_F_A
0	12	10
1	11	11
2	10	12
3	9	13
4	9	14
5	8	15
6	8	16
7	7	17
8	7	18
9	6	19
10	6	20
11	6	21
12	5	22
13	5	24
14	4	26
15	4	28

The recommended settings for this register are provided by the Atmel configuration tool after selecting the desired target data rate for path A.

They can also be calculated by using this procedure:

Calculation of the symbol rate factor

$$SR\_F\_A = \mathrm{round}\left(\mathrm{SymbolRatePathA} \times 2^{DMDNA} \times \frac{128}{CLK\_BB}\right) \qquad (16)$$

Equation parameters:

SymbolRatePathA	Maximum expected symbol rate in Hz on the useful signal for path A
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section "Bandwidth Scaling" on page 99.
SR_F_A	Must be in the range of 10 to 28, if possible 10 to 19.

Find the corresponding DMAA setting in Table 3-13 on page 106.

DMDRB – Demodulator Data Rate on Path B

The target symbol rate for the receiving path B is configured in this register.

This register should only be modified if the block receiving the settings is disabled (RDPR.PRPTB=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMDNB[3:0]				DMAB[3:0]				DMDRB
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..4: DMDNB – Demodulator Down-Sampling on Path B

Selects the down-sampling and cut-off frequency of the first stage of the data filter.

Table 3-14. DMDNB Setting versus Amplitude Threshold Scaling on Path B (ATSCAL_B)

DMDNB	ATSCAL_B
0	1
1	1
2	2
3	2
4	4
5	4
6	8
7	8
8	16

The recommended down-sampling for a target symbol rate is:

$$DMDNB = floor\left[ld(20) + ld\left(\frac{CLK\_BB}{SymbolRatePathB \times 128}\right)\right] \qquad (17)$$

If the result is –1 set DMDNB=0

Equation parameters:

SymbolRatePathB	Maximum expected symbol rate in Hz on the useful signal for path B
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section "Bandwidth Scaling" on page 99.
ld()	Logarithm to the base 2

Bits 3..0: DMAB – Demodulator Moving Average Data Rate Factor on Path B

Selects the moving average filter length for the data filter and initial data rate for the clock recovery on path B.

Table 3-15. DMAB Setting versus Moving Average Filter Length and Symbol Rate Factor on Path B (MAVFL_B, SR_F_B)

DMAB	MAVFL_B	SR_F_B
0	12	10
1	11	11
2	10	12
3	9	13
4	9	14
5	8	15
6	8	16
7	7	17
8	7	18
9	6	19
10	6	20
11	6	21
12	5	22
13	5	24
14	4	26
15	4	28

The recommended settings for this register are provided by the Atmel configuration tool after selecting the desired target data rate for path B.

These settings can also be calculated using this procedure:

1. Calculation of the symbol rate factor

$$SR\_F\_B = round\left(SymbolRatePathB \times 2^{DMDNB} \times \frac{128}{CLK\_BB}\right) \qquad (18)$$

Equation parameters:

SymbolRatePathB	Maximum expected symbol rate in Hz on the useful signal for path B
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section "Bandwidth Scaling" on page 99.
SR_F_B	Must be in the range of 10 to 28, if possible 10 to 19

Find the corresponding DMAB setting in Table 3-15 on page 108.

DMCRA - Demodulator Control Register for Path A

This register should only be modified if the block receiving the settings is disabled (RDPR.PRPTA=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMARA	SY1TA	SASKA	DMPGA[4:0]					DMCRA
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: DMARA – Demodulator Automatic Restart on Path A

This bit enables a hardware-controlled automatic restart of the demodulator and the subsequent receiving path A.

Bit 7	Description
0	The reception and demodulation is continued regardless of error events.
1	Reception and demodulation are restarted if any activated condition for the EOTA interrupt is true. The EOTA condition status flags (EOTS[3:0]) and the SOTA condition status flags (SOTSA) are cleared at every restart. The restart is automatically performed by toggling the RDPR.PRPTA bit. The PRPTA bit is not writable while automatic mode is active.

Bit 6: SY1TA – Symbol Check with 1T only on Path A

Activates a more rigorous check during symbol timing check.

Bit 6	Description
0	The symbol timing check accepts HIGH and LOW times with the selected symbol period (1T) and the double symbol period (2T). Random Manchester data can be used as the preamble in this mode.
1	The symbol timing check allows HIGH and LOW times with the selected symbol rate only. This corresponds to an alternating preamble of 1 and 0 symbols. It is a more rigorous criterion and therefore leads to shorter average on-times. The 1T check is performed for the number of bits specified in the SYCSA register. Afterwards 1T and 2T are acceptable (corresponding to random Manchester).

Bit 5: SASKA – Select ASK Input for Path A

Bit 5	Description
0	FSK input is selected for receiving path A
1	ASK input is selected for receiving path A

Bits 4..0: DMPGA – FSK Demodulator PLL Loop Gain for Path A

The correct settings for this register are provided by the Atmel configuration tool after selecting the target deviation and data rate for path A. They can also be calculated by using the following procedure:

- Calculation of the target PLL loop gain for the maximum symbol rate

$$SYM\_GAIN\_A = 5 \times \frac{SymbolRatePathA}{CLK\_BB} \qquad (19)$$

Equation parameters:

SymbolRatePathA	Maximum expected symbol rate in Hz on the useful signal for path A
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section "Bandwidth Scaling" on page 99.

- Calculation of the target PLL gain for the maximum deviation

$$DEV\_GAIN\_A = 10 \times \frac{DeviationPathA}{CLK\_BB} \quad (20)$$

Equation parameters:

DeviationPathA	Maximum expected frequency deviation in Hz of the useful signal for path A
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section "Bandwidth Scaling" on page 99.

- Selection of the required PLL loop gain which is the larger one of the two gains

REQ_PLL_GAIN = max(SYM_GAIN_A,DEV_GAIN_A) (21)

- Selection of the DMPGA value according to the required PLL loop gain

The PLL_GAIN_A should be selected from the table below to be greater than or equal to the REQ_PLL_GAIN.

If REQ_PLL_GAIN ≥ 1 set DMPGA=16.

The corresponding DMPGA value is the correct setting for this register.

Table 3-16. PLL Gain on Path A versus DMPGA Setting

PLL_GAIN_A	DMPGA (Dec)
0.06	0
0.08	1
0.09	2
0.11	3
0.13	4
0.16	5
0.19	6
0.22	7
0.25	8
0.31	9
0.38	10
0.44	11
0.50	12
0.63	13
0.75	14
0.88	15
≥ 1.00	16

DMCRB - Demodulator Control Register

This register should only be modified if the block receiving the settings is disabled (RDPR.PRPTB=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMARB	SY1TB	SASKB	DMPGB[4:0]					DMCRB
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: DMARB – Demodulator Automatic Restart on Path B

This bit enables a hardware-controlled automatic restart of the demodulator and the subsequent receiving path B.

Bit 7	Description
0	The reception and demodulation is continued regardless of error events.
1	Reception and demodulation are restarted if any activated condition for the EOTB interrupt is true. The EOTB condition status flags (EOTS[3:0]) and the SOTB condition status flags (SOTSB) are cleared at every restart. The restart is automatically performed by toggling the RDPR.PRPTB bit. The PRPTB bit is not writable while automatic mode is active.

Bit 6: SY1TB – Symbol Check with 1T only on Path B

Activates a more rigorous check during symbol timing check.

Bit 6	Description
0	The symbol timing check accepts HIGH and LOW times with the selected symbol period (1T) and the double symbol period (2T). Random Manchester data can be used as preamble in this mode.
1	The symbol timing check allows HIGH and LOW times with the selected symbol rate only. This corresponds to an alternating preamble of 1 and 0 symbols. It is a more rigorous criterion and therefore leads to shorter average on-times. The 1T check is performed for the number of bits specified in the SYCSB register. Afterwards 1T and 2T are acceptable (corresponding to random Manchester).

Bit 5: SASKB – Select ASK Input for Path B

Bit 5	Description
0	FSK input is selected for receiving path B
1	ASK input is selected for receiving path B

Bits 4..0: DMPGB – Demodulator PLL Loop Gain for Path B

The correct settings for this register are provided by the Atmel configuration tool after selecting the target deviation and data rate for path B. They can also be calculated by using the following procedure:

1. Calculation of the target PLL loop gain for the maximum symbol rate

$$SYM\_GAIN\_B = 5 \times \frac{SymbolRatePathB}{CLK\_BB} \qquad (22)$$

Equation parameters:

SymbolRatePathB	Maximum expected symbol rate in Hz on the useful signal for path B
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section “Bandwidth Scaling” on page 99.

2. Calculation of the target PLL loop gain for the maximum deviation

$$DEV\_GAIN\_B = 10 \times \frac{DeviationPathB}{CLK\_BB} \qquad (23)$$

Equation parameters:

DeviationPathB	Maximum expected frequency deviation in Hz of the useful signal for path B
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section “Bandwidth Scaling” on page 99.

3. Selection of the required PLL gain which is the larger one of the two gains

   REQ_PLL_GAIN = max(SYM_GAIN_B,DEV_GAIN_B) (24)
4. Selection of the DMPGB value according to the required PLL loop gain

The PLL_GAIN_B should be selected from Table 3-17 on page 112 to be greater than or equal to the REQ_PLL_GAIN. If REQ_PLL_GAIN ≥ 1 set DMPGB=16.

The corresponding DMPGB value is the correct setting for this register.

Table 3-17. PLL Gain on Path B versus DMPGB Setting

PLL_GAIN_B	DMPGB (Dec)
0.06	0
0.08	1
0.09	2
0.11	3
0.13	4
0.16	5
0.19	6
0.22	7
0.25	8
0.31	9
0.38	10
0.44	11
0.50	12
0.63	13
0.75	14
0.88	15
≥1.00	16

DMCDA – Demodulator Carrier Detect for Path A

The DMCDA register contains the settings for the carrier detection of path A. The number of checked samples (DMCDA.DMCTA) and the maximum number of allowed errors (DMCDA.DMCLA) within this time window can be configured.

This register should only be modified if the block receiving the settings is disabled (RDPR.PRPTA=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMCTA[2:0]			DMCLA[4:0]					DMCDA
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: DMCTA – Demodulator Carrier Detect Time for Path A

This setting defines the time window for carrier detection as a number of checked samples. A counter accumulates the carrier errors for the number of samples defined in this register. The counter is reset after this period for the next check cycle. Selecting DMCTA=0 turns the carrier check on path A off.

The recommendation is to select a DMCTA value covering two symbols (one Manchester bit).

1. Calculation of the number of samples to span over two symbols

$$SAMPLES\_2SYNA = 2 \times \frac{CLK\_BB}{SymbolRatePathA} \quad (25)$$

Equation parameters:

CLK_BB Baseband clock frequency in Hz.
See equation (10) in the Section "Bandwidth Scaling" on page 99.

Choose a number of samples greater or equal to SAMPLES_2SYNA from Table 3-18. The corresponding carrier detect time DMCTA[2:0] value is the preferred setting.

$$SAMPLES\_CTA \geq SAMPLES\_2SYNA \quad (26)$$

Table 3-18. Number of Samples for Carrier Detect Time versus DMCTA Setting

SAMPLES_CTA	DMCTA
Carrier Detect OFF	0
16	1
32	2
64	3
128	4
256	5
512	6
1024	7

Bits 4..0: DMCLA - Demodulator Carrier Detect Limit for Path A

Sets the number of allowed carrier errors within the time window defined by DMCTA. The recommended value for this limit is 20% of the samples within the carrier detect time window (see SAMPLES_CTA value above). Smaller values, e.g., 10%, lead to more rigorous checking and thus to more rapid error detection. More rigorous checking may cause reduced sensitivity.

Calculation procedure example:

1. Select the number of allowed errors within the carrier detect time window (20%)

$$AllowedCarrierCheckErrors = \frac{SAMPLES\_CTA}{5} \quad (27)$$

Equation parameters:

SAMPLES_CTB Number of samples for carrier check. See Table 3-18.

2. Calculate the register setting

$$DMCLA = round\left(\frac{AllowedCarrierCheckErrors}{4}\right) \quad (28)$$

DMCDB – Demodulator Carrier Detect for Path B

The DMCDB register contains the settings for the carrier detection of path B. The number of checked samples (DMCDB.DMCTB) and the maximum number of allowed errors (DMCDB.DMCLB) within this time window can be configured.

This register should only be modified if the block receiving the settings is disabled (RDPR.PRPTB=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMCTB[2:0]			DMCLB[4:0]					DMCDB
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: DMCTB – Demodulator Carrier Detect Time for Path B

This setting defines the time window for carrier detection as a number of checked samples. A counter accumulates the carrier errors for the number of samples defined in this register. The counter is reset after this period for the next check cycle. Selecting DMCTB=0 turns the carrier check on path B off.

Atmel recommends selecting a DMCTB value covering 2 symbols (one Manchester bit).

1. Calculation of the number of samples to span over 2 symbols

$$\mathrm{SAMPLES\_2SYNB} = 2 \times \frac{\mathrm{CLK\_BB}}{\mathrm{SymbolRatePathB}} \qquad (29)$$

Equation parameters:

CLK_BB Baseband clock frequency in Hz.
See equation (10) in the Section "Bandwidth Scaling" on page 99.

Choose a number of samples greater or equal to SAMPLES_2SYNB from the table below. The corresponding carrier detect time DMCTB[2:0] value is the preferred setting.

$$\mathrm{SAMPLES\_CTB} \geq \mathrm{SAMPLES\_2SYNB} \qquad (30)$$

Table 3-19. Number of Samples for Carrier Detect Time versus DMCTB Setting

SAMPLES_CTB	DMCTB
Carrier Detect OFF	0
16	1
32	2
64	3
128	4
256	5
512	6
1024	7

Bit 4..0: DMCLB - Demodulator Carrier Detect Limit for Path B

Sets the number of allowed carrier errors within the time window defined by DMCTB. The recommended value for this limit is 20% of the samples within the carrier detect time window (see SAMPLES_CTB value above). Smaller values, e.g., 10% lead to more rigorous checking, resulting in more rapid error detection. More rigorous checking may cause reduced sensitivity.

Calculation procedure example:

1. Select the number of allowed errors within the carrier detect time window (20%)

$$\mathrm{AllowedCarrierCheckErrorsB} = \frac{\mathrm{SAMPLES\_CTB}}{5} \qquad (31)$$

Equation parameters:

SAMPLES_CTB Number of samples for carrier check. See Table 3-19.

2. Calculate the register setting

$$\mathrm{DMCLB} = \mathrm{round}\left(\frac{\mathrm{AllowedCarrierCheckErrorsB}}{4}\right) \qquad (32)$$

DMMA – Demodulator Mode for Path A

This register selects the main operating modes of the demodulator and the data filter.

It should only be modified if the block receiving the settings is disabled (RDPR.PRPTA =1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMNEA	DMHA	DMPA	DMATA[4:0]					DMMA
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: DMNEA - Demodulator NRZ Enable for Path A

Enables NRZ reception on data path A.

Table 3-20. DMNEA Bit Details

Bit 7	Description
0	Matched Manchester filter is used for data reception and TMDO output.
1	Symbol-based (NRZ) filter is used for data reception and TMDO output. Atmel recommends activating hold mode (DMHA=1) and setting a reasonable amplitude threshold (DMATA) because this is used for noise suppression in the DC removal loop.

Bit 6: DMHA - Demodulator Hold Mode for Path A

This register activates the freezing of the DC compensation if no signal transitions are detected.

Table 3-21. DMHA Bit Details

Bit 6	Description
0	Hold mode deactivated – To be used for telegrams with up to two identical consecutive symbols. Adapts well to changing signal conditions. Not suitable for longer consecutive high or low symbol sequences because the DC threshold fades over time.
1	Hold mode activated – To be used for NRZ or other reception with up to eight identical symbols in sequence. This mode freezes the DC compensation loop after a symbol change until the next signal edge is detected.

Bit 5: DMPA - Demodulator Received Data Polarity Selection for Path A

This register selects the data polarity at TMDO and the Rx buffer for path A. There is no effect on TRPA.

Table 3-22. DMPA Bit Details

Bit 5	Description
0	For Manchester coding: Rising edge in the middle of a bit is considered a “1”. For NRZ coding: Higher frequency (FSK) and active carrier (ASK) is considered a “1”.
1	For Manchester coding: Falling edge in the middle of a bit is considered a “1”. For NRZ coding: Lower frequency (FSK) and no carrier (ASK) is considered a “1”.

Bits 4..0: DMATA – Demodulator Amplitude Threshold for Path A

Sets a minimum required modulation amplitude for the modulation amplitude check.

An EOTS.AMPFA error is flagged if the selected modulation threshold is not exceeded by the incoming signal within two symbols (during WOK check if SY1TA is set) or within three symbols under all other circumstances. This check can also be used as an additional start of telegram condition (SOTSA.AMPOA). It is set to “1” if no error occurred during the symbol check time.

The recommended threshold is calculated differently for ASK and FSK modulation.

Recommended DMATA Setting for FSK

The maximum useful amplitude threshold for a given FSK signal depends on the expected deviation and the data rate settings:

$$\mathrm{DMATA\_MAX\_FSK} = \mathrm{round}\left(\frac{\mathrm{MIN\_DEVIATION\_A} \times \mathrm{ATSCAL\_A} \times \mathrm{MAVFL\_A} \times 10}{\mathrm{CLK\_BB} \times \mathrm{PLL\_GAIN\_A}}\right) \quad (33)$$

Equation parameters:

MIN_DEVIATION_A	The lowest possible frequency deviation in Hz of the useful signal on path A.
ATSCAL_A	Amplitude threshold scaling on path A. See Table 3-12 on page 106.
MAVFL_A	Moving average filter length on path A. See Table 3-13 on page 106.
CLK_BB	Baseband clock frequency in Hz. See equation (10) in the Section “Bandwidth Scaling” on page 99.
PLL_GAIN_A	FSK pll gain on path A. See Table 3-16 on page 110.

Using a higher value results in very frequent or a continuous error indication because the modulation amplitude threshold is similar or higher than the minimum deviation (MIN_DEVIATION_A) of the useful signal.

The recommended setting for DMATA is half of the calculated maximum value (DMATA_MAX_FSK)

$$\mathrm{DMATA} = \mathrm{round}\left(\frac{\mathrm{DMATA\_MAX\_FSK}}{2}\right) \quad (34)$$

Raising this value increases the severity of this check.

Recommended DMATA Setting for ASK

$$\mathrm{DMATA} = \mathrm{round}\left(\frac{3 \times \mathrm{ATSCAL\_A} \times \sqrt{\mathrm{MAVFL\_A}}}{2^{\frac{\mathrm{DMDNA}}{2}}}\right) \quad (35)$$

Equation parameters:

ATSCAL_A	Amplitude threshold scaling on path A. See Table 3-12 on page 106.
MAVFL_A	Moving average filter length on path A. See Table 3-13 on page 106.
DMDNA	Down-sampling on path A. See equation (15) on 106.

DMMB – Demodulator Mode for Path B

This register selects the main operating modes of the demodulator and the data filter.

It should only be modified if the block receiving the settings is disabled (RDPR.PRPTB=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	DMNEB	DMHB	DMPB	DMATB[4:0]					DMMB
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: DMNEB - Demodulator NRZ Enable for Path B

Table 3-23. DMNEB Bit Details

Bit 7	Description
0	Matched Manchester filter is used for data reception and TMDO output.
1	Symbol-based (NRZ) filter is used for data reception and TMDO output.

Bit 6: DMHB - Demodulator Hold Mode for Path B

Controls the operation of the DC compensation loop for the symbol-based data filter.

Table 3-24. DMHB Bit Details

Bit 6	Description
0	Hold mode deactivated – To be used for telegrams with up to two identical consecutive symbols. Adapts better to changing signal conditions. Not suitable for longer consecutive high or low periods.
1	Hold mode activated – To be used for NRZ or other reception with up to eight identical symbols in sequence. This mode freezes the DC compensation loop after a symbol change until the next signal edge is detected.

Bit 5: DMPB - Demodulator Received Data Polarity Selection for Path B

This register selects the data polarity at TMDO and Rx buffer for path B. There is no effect on TRPB.

Table 3-25. DMPB Bit Details

Bit 5	Description
0	For Manchester coding: Rising edge in the middle of a bit is considered a “1”. For NRZ coding: Higher frequency (FSK) and active carrier (ASK) is considered a “1”.
1	For Manchester coding: Falling edge in the middle of a bit is considered a “1”. For NRZ coding: Lower frequency (FSK) and no carrier (ASK) is considered a “1”.

Bits 4..0: DMATB – Demodulator Amplitude Threshold for Path B

Sets a minimum required modulation amplitude for the modulation amplitude check.

An EOTS.AMPFB error is flagged if the selected modulation threshold is not exceeded by the incoming signal within two symbols (during WOK check if SY1TB is set) or within three symbols under all other circumstances. This check can also be used as an additional start of telegram condition (SOTSB.AMPOB). It is set to “1” if no error occurred during the symbol check time.

The recommended threshold is calculated differently for ASK and FSK modulation.

Recommended DMATB Setting for FSK

The maximum useful amplitude threshold for a given FSK signal depends on the expected deviation and the data rate settings:

$$\mathrm{DMATB\_MAX\_FSK} = \mathrm{round}\left(\frac{\mathrm{MIN\_DEVIATION\_B} \times \mathrm{ATSCAL\_B} \times \mathrm{MAVFL\_B} \times 10}{\mathrm{CLK\_BB} \times \mathrm{PLL\_GAIN\_B}}\right) \quad (36)$$

Equation parameters:

MIN_DEVIATION_B	The lowest possible frequency deviation in Hz of the useful signal on path B.
ATSCAL_B	Amplitude threshold on path B. See Table 3-14 on page 107.
MAVFL_B	Moving average filter length on path B. See Table 3-15 on page 108.
CLK_BB	Baseband clock frequency in Hz. See equation (10) in Section "Bandwidth Scaling" on page 99.
PLL_GAIN_B	FSK pll gain on path B. See Table 3-17 on page 112.

Using a higher value results in very frequent or permanent error indication because the modulation amplitude threshold is similar to or higher than the minimum deviation (MIN_DEVIATION_B) of the useful signal.

The recommended setting for DMATB is half of the calculated maximum value (DMATB_MAX_FSK)

$$\mathrm{DMATB} = \mathrm{round}\left(\frac{\mathrm{DMATB\_MAX\_FSK}}{2}\right) \quad (37)$$

Raising this value increases the severity of this check.

Recommended DMATB Setting for ASK

$$\mathrm{DMATB} = \mathrm{round}\left(\frac{3 \times \mathrm{ATSCAL\_B} \times \sqrt{MAVFL\_B}}{2^{\frac{\mathrm{DMDNB}}{2}}}\right) \quad (38)$$

Equation parameters:

ATSCAL_B	Amplitude threshold scaling on path B. See Table 3-14 on page 107.
MAVFL_B	Moving average filter length on path B. See Table 3-15 on page 108.
DMDNB	Down-sampling on path B. See equation (17) on 108.

SYCA – Symbol Check Configuration for Path A

The symbol check duration and timing is configured by the settings in this register for path A.

This register should only be modified if the block receiving the settings is disabled (RDPR.PRPTA=1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	SYTLA[3:0]				SYCSA[3:0]				SYCA
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..4: SYTLA – Symbol Timing Limit for Path A

SYTLA sets the timing limits for the allowed tolerance of the actual symbol edge transition versus the expected position. The expected symbol edge position is derived from the clock recovery circuit.

The table below shows the allowed tolerance as a percentage of the symbol duration. Values below 25% are not recommended because the edge to edge jitter can be in this range and therefore might lead to false error indication.

Table 3-26. Allowed Symbol Edge Tolerance versus SYTLA

Allowed Symbol Edge Tolerance ±	SYTLA
0%	0
3%	1
6%	2
9%	3
13%	4
16%	5
19%	6
22%	7
Tight – 25%	8
28%	9
31%	10
Default – 34%	11
38%	12
41%	13
44%	14
Lax – 47%	15

Bits 3..0: SYCSA – Symbol Check Size for Data Path A.

The SYCSA bits configure the number of symbols checked for the symbol check OK signals (SOTSA.MANOA, SOTSA.SYTOA, SOTSA.AMPOA, SOTSA.CAROA).

Table 3-27. Number of Symbols Checked versus SYCSA

Number of Symbols Checked	SYCSA
Check Disabled	0
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
17	9
19	10
21	11
23	12
25	13
27	14
29	15

A SYCSA value of 6 is a good setting for avoiding unwanted wake-ups. Lower values lead to laxer checking, higher values subjects the incoming signal to a more severe check.

SYCB – Symbol Check Configuration for Path B

The symbol check duration and timing is configured by the settings in this register for path B.

This register should only be modified if the block receiving the settings is disabled (RDPR. PRPTB =1). Modifying the settings during operation may lead to unstable operation.

Bit	7	6	5	4	3	2	1	0	
	SYTLB[3:0]				SYCSB[3:0]				SYCB
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..4: SYTLB – Symbol Timing Limit for Path B

SYTLB sets the timing limits for the allowed tolerance of the actual symbol edge transition versus the expected position. The expected symbol edge position is derived from the clock recovery circuit.

Table 3-28 shows the allowed tolerance as a percentage of the symbol duration. Values below 25% are not recommended because the edge to edge jitter can be in this range and therefore might lead to false error indication.

Table 3-28. Allowed Symbol Edge Tolerance versus SYTLB

Allowed Symbol Edge Tolerance +/-	SYTLB
0%	0
3%	1
6%	2
9%	3
13%	4
16%	5
19%	6
22%	7
Tight – 25%	8
28%	9
31%	10
Default – 34%	11
38%	12
41%	13
44%	14
Lax – 47%	15

Bits 3..0: SYCSB – Symbol Check Size for Data Path B.

The SYCSB bits configure the number of symbols checked for the symbol check OK signals (SOTSB.MANOB, SOTSB.SYTOB, SOTSB.AMPOB, SOTSB.CAROB).

Table 3-29. Number of Symbols Checked versus SYCSB

Number of Symbols Checked	SYCSB
Check Disabled	0
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
17	9
19	10
21	11
23	12
25	13
27	14
29	15

A SYCSB value of 6 is a good setting for avoiding unwanted wake-ups. Lower values lead to laxer checking, higher values subjects the incoming signal to a more severe check.

3.4.3.3 Frame Synchronizer

The frame synchronizer block analyzes the demodulated data and compares it to the configured pattern in order to detect a valid preamble and the start point of a telegram within the data stream.

The frame synchronizer works on a symbol basis. All registers configure and refer to data symbols, not data bits. Please refer to Section 2.1.2.2 “NRZ and Manchester Coding” on page 11 for hints on “bits” and “symbols.”

The complete hardware is available twice to allow separate configuration and processing of path A and path B. In the following sections a small x indicates that the description is valid for both paths.

The frame synchronizer is reset at every demodulator restart.

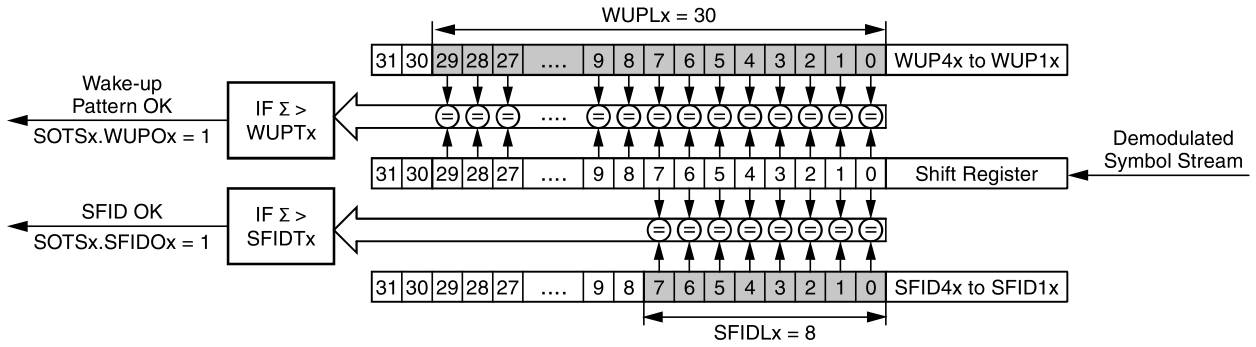
If not stated otherwise, a detailed description of the required register bits is found in Section “Frame Synchronizer Register Description” on page 123.

Pattern Correlation

Two different patterns, each with a length of up to 32 symbols, can be defined to implement a customized wake check OK (WCO) and start of telegram (SOT) detection procedure.

An example for the correlation process and the pattern match generation for one path is visualized in Figure 3-9 on page 122.

Figure 3-9. WUP and SFID Pattern Correlator



The demodulated symbol stream is checked in parallel for the configured wake-up pattern (WUP) and start frame identification (SFID). A pattern match is signaled by setting the corresponding SOTSx.WUPOx, or SOTSx.SFIDOx bit to 1 (for more information, see Section "Rx DSP Control Register Description" on page 146).

The WUP can be configured on a symbol basis in the WUP4x, WUP3x, WUP2x, and WUP1x registers. The SIFD pattern can be configured on a symbol basis in the SFID4x, SFID3x, SFID2x, and SFID1x registers.

The length of each pattern is configurable between 1 and 32 symbols in WUPLx and SFIDLx. If the length of the pattern is configured to be less than 32, the corresponding number of least significant symbols are selected and the rest is ignored.

To create a pattern match, the number of matching symbols has to exceed the threshold values in WUPTx and SFIDCx.SFIDTx. Thus, if the complete pattern is supposed to match, the threshold value has to be one less than the pattern length value.

To avoid an unintended early WUP or SOT, a pattern match will only be indicated if at least the number of symbols configured in the length register has been received. Any earlier pattern match is suppressed.

The SIFDCx.SEMEx serial mode enable bit can be used to activate a hardware suppression of an SFID match until a WUP match has occurred and a full SFID pattern has been received after the WUP.

The data polarity can be configured in DMMx.DMPx (see Section "Demodulator Register Description" on page 105). If DMPx = "1", the received data are inverted in the demodulator before they are transferred to the correlator. The configured WUP and SFID pattern must therefore be the inverse of the RF signal.

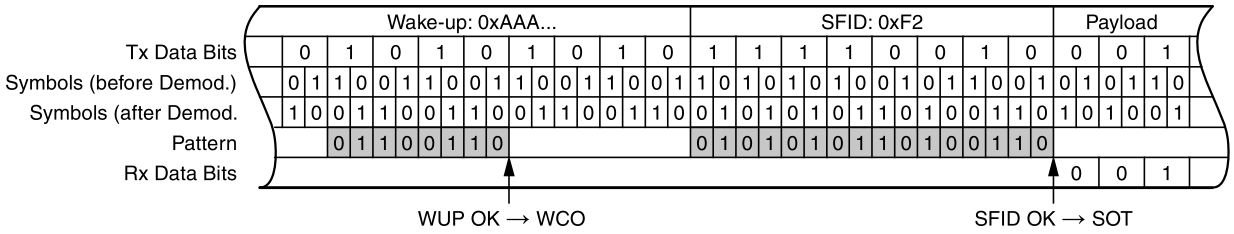
Configuration example for a wake-up and SFID pattern on path A:

DMMA.DMNEA = 0 Data are Manchester-coded
DMMA.DMPA = 1 Manchester polarity: Falling edge in the middle of a bit is "1", data are inverted in the demodulator

WUP4A = 0x80
WUP3A = 0x00
WUP2A = 0x66
WUP1A = 0x66 -> WUP= 1000 0000 0000 0000 0110 0110 0110 0110
WUPLA = 8 -> Only eight LSBs of WUP are considered (0x66)
WUPTA = 7 -> Threshold = WUPTA+1 -> All eight symbols must match

SFID4A = 0x00
SFID3A = 0x00
SFID2A = 0x55
SFID1A = 0xA6 -> SFID = 0000 0000 0000 0000 0101 0101 1010 0110
SFIDLA = 16 -> Only 16 LSBs of WUP are considered (0x55A6)
SFIDTA = 15 -> Threshold = SFIDTA+1 -> All 16 symbols must match

Figure 3-10. WUP and SFID Pattern Example



Frame Synchronizer Register Description

WUP4A – Wake-Up Pattern Byte 4 on Path A

Bit	7	6	5	4	3	2	1	0	
	WUP4A[7:0]								WUP4A
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 31.. 24 of the 32-bit wake-up pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

WUP3A – Wake-Up Pattern Byte 3 on Path A

Bit	7	6	5	4	3	2	1	0	
	WUP3A[7:0]								WUP3A
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 23..16 of the 32-bit wake-up pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

WUP2A – Wake-Up Pattern Byte 2 on Path A

Bit	7	6	5	4	3	2	1	0	
	WUP2A[7:0]								WUP2A
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 15.. 8 of the 32-bit wake-up pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

WUP1A – Wake-Up Pattern Byte 1 on Path A

Bit	7	6	5	4	3	2	1	0	
	WUP1A[7:0]								WUP1A
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 of the 32-bit wake-up pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

SFID4A – Start Frame ID Byte 4 on Path A

Bit	7	6	5	4	3	2	1	0	
	SFID4A[7:0]								SFID4A
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 31.. 24 of the 32-bit start frame identification pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

SFID3A – Start Frame ID Byte 3 on Path A

Bit	7	6	5	4	3	2	1	0	
	SFID3A[7:0]								SFID3A
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 23.. 16 of the 32-bit start frame identification pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

SFID2A – Start Frame ID Byte 2 on Path A

Bit	7	6	5	4	3	2	1	0	
	SFID2A[7:0]								SFID2A
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 15.. 8 of the 32-bit start frame identification pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

SFID1A – Start Frame ID Byte 1 on Path A

Bit	7	6	5	4	3	2	1	0	
	SFID1A[7:0]								SFID1A
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 of the 32-bit start frame identification pattern for receiving path A. The bits of this register represent data symbols. The coding must comply with the configured polarity in DMMA.DMPA.

WUPLA – Wake-Up Pattern Length for Path A

Bit	7	6	5	4	3	2	1	0	
	-	-	WUPLA[5:0]						WUPLA
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 return zero when read.

Bits 5..0: WUPLA – Wake-Up Pattern Length for Path A

The wake-up pattern length has a valid range of 1 to 32 symbols. Other values should not be used and lead to erroneous behavior.

Starting with the LSB, the number of bits is compared with the incoming data stream to generate a wake-up OK event (SOTSA.WUPOA) if the specified threshold from the WUPTA register is exceeded.

WUPTA – Wake-Up Pattern Threshold for Path A

Bit	7	6	5	4	3	2	1	0	
	-	-	-	WUPTA[4:0]					WUPTA
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 return zero when read.

Bits 4..0: WUPTA – Wake-Up Pattern Threshold for Path A

The number of symbols matching the expected wake-up pattern has to exceed the number specified in this register to generate a wake-up OK event (SOTSA.WUPOA).

SFIDLA – Start Frame ID Length for Path A

Bit	7	6	5	4	3	2	1	0	
	-	-	SFIDLA[5:0]						SFIDLA
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 return zero when read.

Bits 5..0: SFIDLA – Start Frame ID Length for Path A

The start frame ID length has a valid range of 1 to 32 symbols. Other values should not be used and lead to erroneous behavior.

The number of bits is compared with the incoming data stream to generate a start frame ID OK event (SOTSA.SFIDOA) if the specified threshold from the SFIDCA.SFIDTA register is exceeded.

SFIDCA – Start Frame ID Configuration for Path A

Bit	7	6	5	4	3	2	1	0	
	SEMEA	-	-	SFIDTA[4:0]					SFIDCA
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: SEMEA – Serial Mode Enable for Path A

Setting this bit to “1” enables a hardware-controlled serial mode for the path A correlator.

Value	Description
0	Parallel search for wake-up pattern (WUP) and start frame ID (SFID) is enabled.
1	Serial search for wake-up pattern (WUP) and start frame ID (SFID) is enabled. An SFID match is accepted only after a successful wake-up pattern match.

Bits 6..5: Reserved Bits

These bits are reserved and return zero when read.

Bits 4..0: SFIDTA – Start Frame ID Threshold for Path A

The number of symbols matching the expected start frame ID has to exceed the number specified in this register to generate a start frame ID OK event (SOTSA.SFIDOA).

WUP4B – Wake-Up Pattern Byte 4 on Path B

Bit	7	6	5	4	3	2	1	0	
	WUP4B[7:0]								WUP4B
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 31.. 24 of the 32-bit wake-up pattern for receiving path B. The coding must be according to the configured polarity in DMMB.DMPB.

WUP3B – Wake-Up Pattern Byte 3 on Path B

Bit	7	6	5	4	3	2	1	0	
	WUP3B[7:0]								WUP3B
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 23.. 16 of the 32-bit wake-up pattern for receiving path B. The bits of this register represent data symbols. The coding must be according to the configured polarity in DMMB.DMPB.

WUP2B – Wake-Up Pattern Byte 2 on Path B

Bit	7	6	5	4	3	2	1	0	
	WUP2B[7:0]								WUP2B
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 15 .. 8 of the 32-bit wake-up pattern for receiving path B. The bits of this register represent data symbols. The coding must be according to the configured polarity in DMMB.DMPB.

WUP1B – Wake-Up Pattern Byte 1 on Path B

Bit	7	6	5	4	3	2	1	0	
	WUP1B[7:0]								WUP1B
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 of the 32-bit wake-up pattern for receiving path B. The bits of this register represent data symbols. The coding must be according to the configured polarity in DMMB.DMPB.

SFID4B – Start Frame ID Byte 4 on Path B

Bit	7	6	5	4	3	2	1	0	
	SFID4B[7:0]								SFID4B
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 31.. 24 of the 32-bit start frame identification pattern for receiving path B. The bits of this register represent data symbols. The coding must be according to the configured polarity in DMMB.DMPB.

SFID3B – Start Frame ID Byte 3 on Path B

Bit	7	6	5	4	3	2	1	0	
	SFID3B[7:0]								SFID3B
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 23.. 16 of the 32-bit start frame identification pattern for receiving path B. The bits of this register represent data symbols. The coding must be according to the configured polarity in DMMB.DMPB.

SFID2B – Start Frame ID Byte 2 on Path B

Bit	7	6	5	4	3	2	1	0	
	SFID2B[7:0]								SFID2B
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 15.. 8 of the 32-bit start frame identification pattern for receiving path B. The bits of this register represent data symbols. The coding must be according to the configured polarity in DMMB.DMPB.

SFID1B – Start Frame ID Byte 1 on Path B

Bit	7	6	5	4	3	2	1	0	
	SFID1B[7:0]								SFID1B
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 of the 32-bit start frame identification pattern for receiving path B. The bits of this register represent data symbols. The coding must be according to the configured polarity in DMMB.DMPB.

WUPLB – Wake-Up Pattern Length for Path B

Bit	7	6	5	4	3	2	1	0	
	-	-	WUPLB[5:0]						WUPLB
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 return zero when read.

Bits 5..0: WUPLB – Wake-Up Pattern Length for Path B

The wake-up pattern length has a valid range of 1 to 32 symbols. Other values should not be used and lead to erroneous behavior.

This amount of bits is compared, starting with the LSB, versus the incoming data stream to generate a wake-up OK event (SOTSB.WUPOB) if the specified threshold from the WUPTB register is exceeded.

WUPTB – Wake-Up Pattern Threshold for Path B

Bit	7	6	5	4	3	2	1	0	
	-	-	-	WUPTB[4:0]					WUPTB
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 return zero when read.

Bits 4..0: WUPTB – Wake-Up Pattern Threshold for Path B

The number of symbols matching the expected wake-up pattern has to exceed the number specified in this register to generate a wake-up OK event (SOTSB.WUPOB).

SFIDLB – Start Frame ID Length for Path B

Bit	7	6	5	4	3	2	1	0	
	-	-	SFIDLB[5:0]						SFIDLB
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 return zero when read.

Bits 5..0: SFIDLB – Start Frame ID Length for Path B

The start frame ID length has a valid range of 1 to 32 symbols. Other values should not be used and lead to erroneous behavior.

This amount of bits is compared versus the incoming data stream to generate a start frame ID OK event (SOTSB.SFIDOB) if the specified threshold from the SFIDCB.SFIDTB register is exceeded.

SFIDCB – Start Frame ID Configuration for Path B

Bit	7	6	5	4	3	2	1	0	
	SEMEB	-	-	SFIDTB[4:0]					SFIDCB
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: SEMEB – Serial Mode Enable for Path B

Setting this bit to “1” enables a hardware-controlled serial mode for the path B correlator.

Value	Description
0	Parallel search for wake-up pattern (WUP) and start frame ID (SFID) is enabled.
1	Serial search for wake-up pattern (WUP) and start frame ID (SFID) is enabled. An SFID match is accepted only after a successful wake-up pattern match.

Bits 6..5: Reserved Bits

These bits are reserved and return zero when read.

Bits 4..0: SFIDTB – Start Frame ID Threshold for Path B

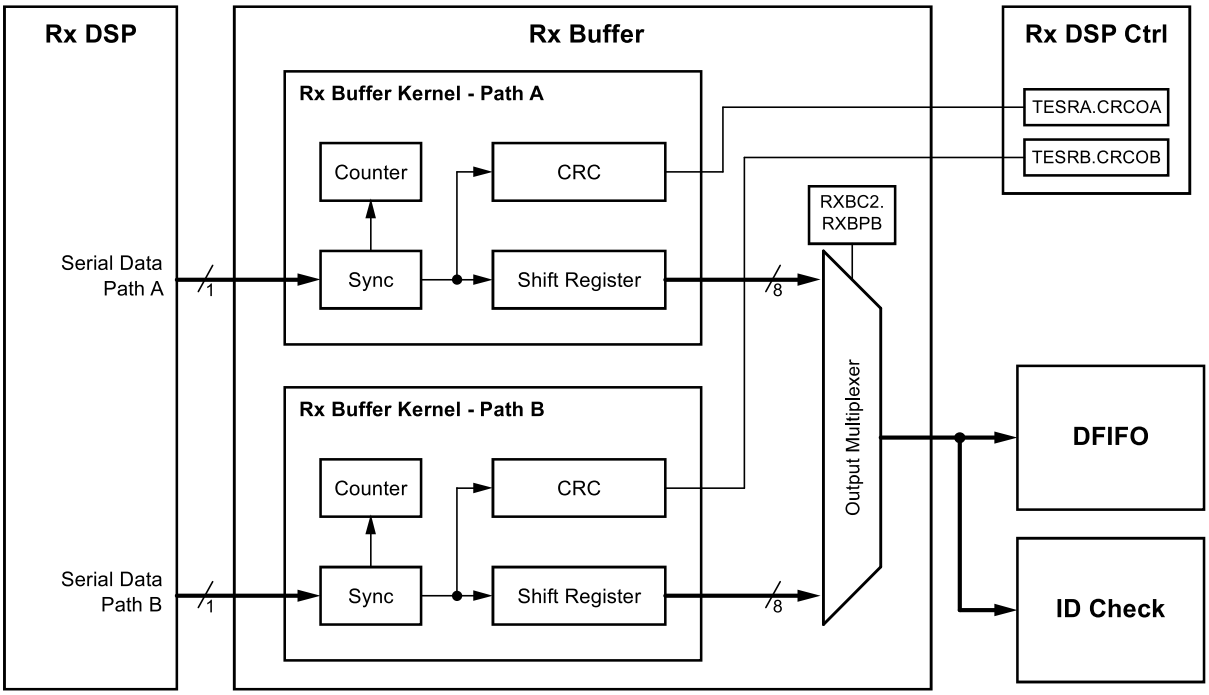
The number of symbols matching the expected start frame ID has to exceed the number specified in this register to generate a start frame ID OK event (SOTSB.SFIDOB).

3.4.3.4 Rx Buffer

Rx Buffer Overview

The Rx buffer uses the demodulated serial data stream from the Rx-DSP as input. The input data are synchronized to the AVR® clock domain and strobed to an 8-bit shift register. The data is then automatically written byte-wise from the shift register to the DFIFO and the ID check module (if enabled). The data order can be selected to be MSB-first or LSB-first in the RXBC1.RXMSBA/B register. Additionally, a CRC checksum can be calculated on the received data. Figure 3-11 shows a block diagram of the Rx buffer.

Figure 3-11. Rx Buffer Block Diagram



There are two identical Rx buffer kernels available to allow a separate configuration and processing of path A and path B. In the following sections a small x indicates that the description is valid for both paths. Upon detection of an SOT the active path must be selected in RXBC2.RXBPB before the first byte is written to the DFIFO via the output multiplexer. In the Atmel® ATA5785 the process is controlled by the sequencer state machine.

After SOT, any EOT event on the active path causes the sequencer state machine to set the RXBC2.RXBF finish bit. The data currently available in the shift register are immediately sent to the DFIFO correctly aligned and afterwards the Rx buffer stops operation until it is reset. The same procedure is carried out when the telegram length is reached which can be configured in the RXTLHx/RXTLLx registers or when the Rx buffer finish bit is set in RXBC2.RXBF via the AVR bus.

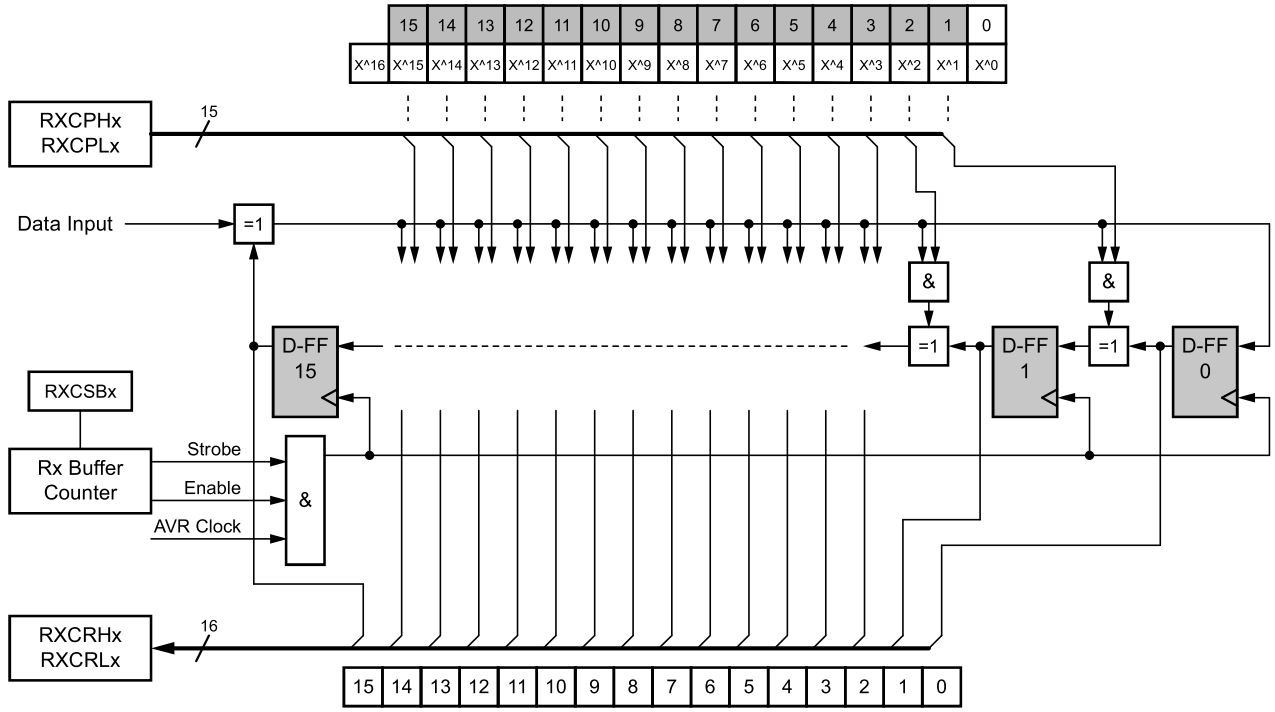
The Rx buffer is reset when activated in the corresponding PRR2.PRXx power reduction register (see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210) or by writing a “1” to the RXBC2.RXBCLR bit.

If not stated otherwise, a description of the required register bits is found in Section “Rx Buffer Register Description” on page 131.

Receive CRC Calculation

A cyclic redundancy check (CRC) calculation of the received payload data can be enabled independently for every path by setting RXBC1.RXCEx to “1”. Figure 3-12 shows a block diagram of the CRC module in 16-bit mode.

Figure 3-12. Receive CRC Block Diagram

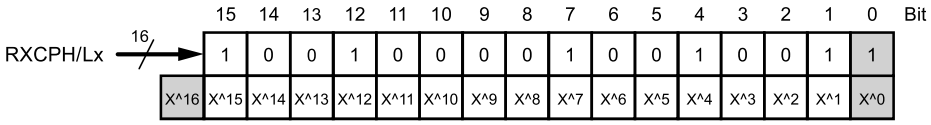


The CRC block is implemented as a 16-bit shift register with configurable feedback loops. When the Rx buffer is enabled, the CRC shift register is initialized with RXCIHx/RXCILx (default value: 0x00). The length of the shift register can be reduced to eight bits or four bits in the RXCR2.RXCBLx register for CRC8 or CRC4 calculations, respectively. The corresponding logic is not shown in Figure 3-12.

The coefficients of the CRC polynomial can be configured in the RXCPHx/RXCPLx registers. The position in the RXCPx word corresponds to the degree of the term within the polynomial as illustrated for a 16-bit polynomial in Figure 3-13. The coefficient of the x^0 term of a CRC polynomial must always be 1 and bit 0 of RXCPLx is thus hard-wired to “1”.

Figure 3-13. Receive CRC Polynomial

Example RXCPHx/RXCPLx Setting: 0x9093 = 0b1001000010010011



Polynomial: $1 \cdot x^{16}+1 \cdot x^{15}+0 \cdot x^{14}+0 \cdot x^{13}+1 \cdot x^{12}+0 \cdot x^{11}+0 \cdot x^{10}+0 \cdot x^{9}+0 \cdot x^{8}+1 \cdot x^{7}+0 \cdot x^{6}+0 \cdot x^{5}+1 \cdot x^{4}+0 \cdot x^{3}+0 \cdot x^{2}+1 \cdot x^{1}+1 \cdot x^{0}$

$= x^{16}+x^{15}+x^{12}+x^{7}+x^{4}+x^{1}+1$

A skip period can be configured in RXCSBx. If this value is greater than 0x00, the Rx buffer omits the first RXCSBx bits of the payload data stream before enabling the CRC.

The current CRC result can be read in the RXCRHx/RXCRLx registers. If the CRC result is “0x0000” for an Rx buffer finish event, the CRC check is considered successful and the TESRx.CRCOx register bit is written to “1” (see Section “Rx Buffer Register Description” on page 131).

Rx Buffer Register Description

RXBC1 – Rx Buffer Configuration Register 1

Bit	7	6	5	4	3	2	1	0	
	RXMSBB	RXCBLB		RXCEB	RXMSBA	RXCBLA		RXCEA	RXBC1
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: RXMSBB – Receive Data MSB-First for Data Path B

1: Data is shifted into the shift register MSB-first

0: Data is shifted into the shift register LSB-first

Bits 6..5: RXCBLB – Receive CRC Bit Lengths Setting for Data Path B

RXCBLB	Description
00	CRC 4-bit configuration
01	CRC 8-bit configuration
10	Not used (4-bit configuration default)
11	CRC 16-bit configuration

Bit 4: RXCEB – RX CRC Enable Data Path B

1: CRC is enabled

0: CRC is disabled

Bit 3: RXMSBA – Receive Data MSB-First for Data Path A

1: Data is shifted into the shift register MSB-first

0: Data is shifted into the shift register LSB-first

Bits 2..1: RXCBLA – Receive CRC Bit Lengths Setting for Data Path A

RXCBLAth	Description
00	CRC 4-bit configuration
01	CRC 8-bit configuration
10	Not used (4-bit configuration default)
11	CRC 16-bit configuration

Bit 0: RXCEA – RX CRC Enable Data Path A

1: CRC is enabled

0: CRC is disabled

RXBC2 – Rx Buffer Configuration Register 2

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	RXBCLR	RXBF	RXBPB	RXBC2
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 return zero when read.

Bit 2: RXBCLR – RX Buffer Clear

If a logic “1” is written to this bit, the RX buffer aborts both channels with an internal disable immediately and restarts the receive operation (soft reset). The actual content of the shift register is not written to the DFIFO. The RXBCLR bit returns the internal “clear” state of the Rx buffer when read. If RXBCLR and RXBF are written at the same time, RXBCLR is executed.

Bit 1: RXBF – RX Buffer Finish

If a logic “1” is written to this bit, the RX buffer stops receiving immediately and sends the received bits which are available in the shift register correctly aligned to the DFIFO. Reception can be restarted by writing the RXBC bit or by disabling and enabling the buffer in the corresponding power reduction register. This bit returns the internal “finish” state when read.

Bit 0: RXBPB – RX Buffer Path B Select Signal

1: Path B is selected as output to the DFIFO

0: Path A is selected as output to the DFIFO

RXDSA – Rx Data Shift Register for Data Path A

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

Bits 7..0: RXDSA – Receive Data Shift Register for Data Path A

RXDSA holds the current content of the receive data shift register. This register is read-only. Writing to this register has no effect.

RXCPHA – Rx CRC Polynomial High Byte for Data Path A

Bit	7	6	5	4	3	2	1	0	
	RXCPHA[7:0]								RXCPHA
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: RXCPHA – Rx CRC Polynomial High Byte for Data Path A

This register defines the upper eight bits of the 15-bit Rx CRC polynomial value. This register is unused for CRC4 and CRC8 polynomials.

RXCPLA – Rx CRC Polynomial Low Byte for Data Path A

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

Bits 7..1: RXCPLA – Rx CRC Polynomial Low Byte for Data Path A

This register defines the lower seven bits of the 15-bit Rx CRC polynomial value. Bits 7..4 are unused for CRC4 polynomials.

Bit 0: This bit is hard-wired to “1” and therefore not configurable.

RXCIHA – Rx CRC Init Value High Byte for Data Path A

Bit	7	6	5	4	3	2	1	0	
	RXCIHA[7:0]								RXCIHA
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: RXCIHA – Rx CRC Init Value High Byte for Data Path A

This register defines the upper eight bits of the 16-bit Rx CRC initialization value which is the start value of the CRC result register. This register is unused for CRC4 and CRC8 polynomials.

RXCILA – Rx CRC Init Value Low Byte for Data Path A

Bit	7	6	5	4	3	2	1	0	
	RXCILA[7:0]								RXCILA
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: RXCILA – Rx CRC Init Value Low Byte for Data Path A

This register defines the lower eight bits of the 16- bit Rx CRC initialization value which is the start value of the CRC result register. Bits 7..4 are unused for CRC4 polynomials.

RXCSBA – Rx CRC Skip Bit Number for Data Path A

Bit	7	6	5	4	3	2	1	0	
	RXCSBA[7:0]								RXCSBA
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: RXCSBA Rx CRC Skip Bit Number for Data Path A

Number of bits after start of telegrams which are not strobed to the CRC shift register. Any value between 0 and 255 is valid.

RXCRHA – Rx CRC Result Register High Byte for Data Path A

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

Bits 7..0: RXCRHA Rx CRC Result Register High Byte for Data Path A

The upper eight bits of the actual value in the CRC shift register are readable in RXCRHA, such as for checking the calculated value after reception is complete. This register reads as 0x00 in CRC4 and CRC8 mode.

RXCRLA – Rx CRC Result Register Low Byte for Data Path A

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

Bits 7..0: RXCRLA Rx CRC Result Register Low Byte for Data Path A

The lower eight bits of the actual value in the CRC shift register are readable in RXCRLA, such as for checking the calculated value after reception is complete. Bits 7..4 read as 0x0 in CRC4 mode.

RXTLHA – Rx Telegram Length High Byte for Data Path A

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	RXTLHA[3:0]				RXTLHA
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 return zero when read.

Bits 3..0: RXTLHA – Receive Telegram Length High Byte for Data Path A

RXTLHA contains the upper four bits of the 12-bit telegram length register. When the configured number of bits has been received, reception is terminated and a corresponding signal is sent to the receive state machine. If the telegram length is set to “0x000”, this feature is switched off and the reception must be terminated by another receive error condition.

RXTLLA – Rx Telegram Length Low Byte for Data Path A

Bit	7	6	5	4	3	2	1	0	
	RXTLLA[7:0]								RXTLLA
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: RXTLLA – Receive Telegram Length Low Byte for Data Path A

RXTLLA contains the lower eight bits of the 12-bit telegram length register (see the RXTLHA register above for details).

RXDSB – Rx Data Shift Register for Data Path B

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

Bits 7..0: RXDSB – Receive Data Shift Register for Data Path B

RXDSB holds the current content of the receive data shift register. This register is read-only. Writing to this register has no effect.

RXCPHB – Rx CRC Polynomial High Byte for Data Path B

Bit	7	6	5	4	3	2	1	0	
	RXCPHB[7:0]								RXCPHB
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: RXCPHB – Rx CRC Polynomial High Byte for Data Path B

This register defines the upper eight bits of the 15-bit Rx CRC polynomial value. This register is unused for CRC4 and CRC8 polynomials.

RXCPLB – Rx CRC Polynomial Low Byte for Data Path B

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

Bits 7..1: RXCPLB – Rx CRC Polynomial Low Byte for Data Path B

This register defines the lower seven bits of the 15-bit Rx CRC polynomial value. Bits 7..4 are unused for CRC4 polynomials.

Bit 0: This bit is hard-wired to “1” and therefore not configurable.

RXCIHB – Rx CRC Init Value High Byte for Data Path B

Bit	7	6	5	4	3	2	1	0	
	RXCIHB[7:0]								RXCIHB
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: RXCIHB – Rx CRC Init Value High Byte for Data Path B

This register defines the upper eight bits of the 16-bit Rx CRC initialization value, which is the start value of the CRC result register. This register is unused for CRC4 and CRC8 polynomials.

RXCILB – Rx CRC Init Value Low Byte for Data Path B

Bit	7	6	5	4	3	2	1	0	
	RXCILB[7:0]								RXCILB
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: RXCILB – Rx CRC Init Value Low Byte for Data Path B

This register defines the lower eight bits of the 16- bit Rx CRC initialization value which is the start value of the CRC result register. Bits 7..4 are unused for CRC4 polynomials.

RXCSBB – Rx CRC Skip Bit Number for Data Path B

Bit	7	6	5	4	3	2	1	0	
	RXCSBB[7:0]								RXCSBB
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: RXCSBB Rx CRC Skip Bit Number for Data Path B

Number of bits after start of telegram which are not strobed to the CRC shift register. Any value between 0 and 255 is valid.

RXCRHB – Rx CRC Result Register High Byte for Data Path B

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

Bits 7..0: RXCRHB Rx CRC Result Register High Byte for Data Path B

The upper eight bits of the actual value in the CRC shift register are readable in RXCRHB, such as for checking the calculated value after reception is complete. This register reads as 0x00 in CRC4 and CRC8 mode.

RXCRLB – Rx CRC Result Register Low Byte for Data Path B

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

Bits 7..0: RXCRLB Rx CRC Result Register Low Byte for Data Path B

The lower eight bits of the actual value in the CRC shift register are readable in RXCRLB, such as for checking the calculated value after the transmission is complete. Bits 7..4 read as 0x0 in CRC4 mode.

RXTLHB – Rx Telegram Length High Byte for Data Path B

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	RXTLHB[3:0]				RXTLHB
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 return zero when read.

Bits 3..0: RXTLHB – Receive Telegram Length High Byte for Data Path B

RXTLHB contains the upper four bits of the 12-bit telegram length register. When the configured number of bits has been received, the reception is terminated and a corresponding signal is sent to the receive state machine. If the telegram length is set to “0x000”, this feature is switched off and the reception must be terminated by another receive error condition.

RXTLLB – Rx Telegram Length Low Byte for Data Path B

Bit	7	6	5	4	3	2	1	0	
	RXTLLB[7:0]								RXTLLB
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: RXTLLB – Receive Telegram Length Low Byte for Data Path B

RXTLLB contains the lower eight bits of the 12-bit telegram length register (see the RXTLHB register above for details).

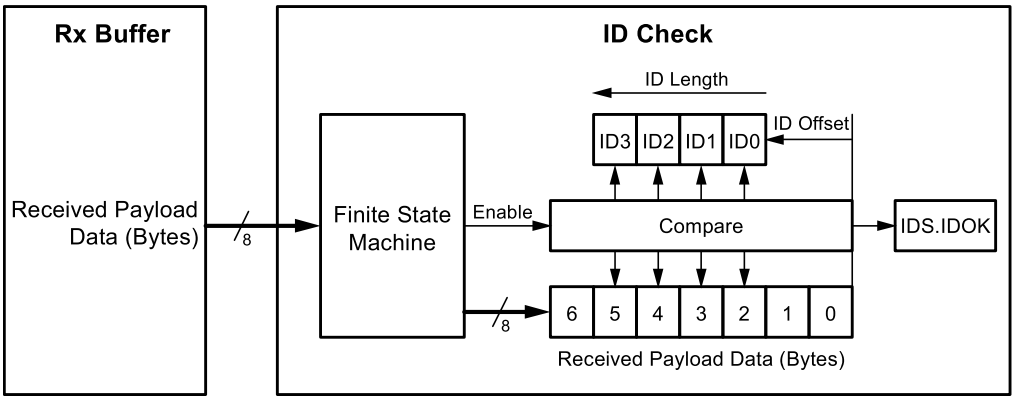
3.4.3.5 ID Check

ID Check Overview

The ID check module can compare predefined IDs with the first bytes of the received data payload. If a match occurs, this is indicated in a single register bit. The ID can have a length of 1, 2, 3, or 4 bytes and an offset from the beginning of the data payload of 0, 1, 2, or 3 bytes. The block is controlled by an internal finite state machine.

Figure 3-14 shows a block diagram of the ID check module.

Figure 3-14. ID Check Block Diagram



To initiate an ID check the required ID must first be written to the ID byte registers IDB0, IDB1, IDB2, and IDB3. An ID can therefore have a length of 1, 2, 3, or 4 bytes which must be configured in the IDC.IDL register. If the ID is not located at the very beginning of the data payload, a byte-wise offset of up to 3 bytes can be configured in IDC.IDBO.

Before the ID check is started, the user must ensure that at least the IDC.IDBO + IDC.IDL payload bytes have been received, because otherwise the ID check will fail.

The user can enable an ID buffer full interrupt by setting IDC.IDFIM to “1”. This triggers an interrupt to the AVR® after 7 bytes have been received, which is the worst case scenario (ID length of 4 bytes and 3 bytes offset). The interrupt can be used by the software to start an ID check with any configuration regardless of the length and offset values of the actual ID. Instead of using the interrupt the IDS.IDFULL flag can be polled.

After configuration is complete, the ID check can be started by setting IDC.IDCE to “1”. It is permitted to write IDCE together with IDBO and IDL in one write access to the IDC register. The result is available within one clock cycle and written to IDS.IDOK. A “1” marks a successful ID check. The bit is only cleared by a module reset or by writing a “1” to it, but not by a successive negative ID check. This functionality allows an arbitrary number of IDs to be processed in a row and checking afterwards if at least one match has occurred.

The ID check is reset when activated in the corresponding PRR2.PRIDS power reduction register (see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210) or by writing a “1” to the IDC.IDCLR bit.

If not stated otherwise, a detailed description of the required register bits is found in Section “ID Check Register Description” on page 137.

ID Check Register Description

IDC – ID Configuration

Bit	7	6	5	4	3	2	1	0	
	IDCE	IDCLR	IDFIM	–	IDBO[1:0]		IDL[1:0]		IDC
Read/Write	R/W	W	RW	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: IDCE – ID Check Execute

If IDCE is set to “1”, the ID check is executed with the ID[3:0] bytes. After the check is complete, this bit is automatically reset to “0”.

Bit 6: IDCLR – ID Clear

Setting this bit to “1” clears the ID check block (soft reset). All internal registers and the status register are reset. The configuration registers IDC and IDB[3:0] are not cleared.

Bit 5: IDFIM – ID Full Interrupt Mask

Writing this bit to “1” enables the ID full interrupt. An interrupt is generated after 7 bytes have been received. The corresponding flag register is IDS.IDFULL.

Bit 4: Reserved Bit

This bit is reserved and returns zero when read.

Bits 3..2: IDBO – ID Byte Offset

0: Offset 0 bytes

1: Offset 1 byte

2: Offset 2 bytes

3: Offset 3 bytes

Bits 1..0: IDL – ID Length

0: ID byte length 1 byte

1: ID byte length 2 bytes

2: ID byte length 3 bytes

3: ID byte length 4 bytes

IDS – ID Status

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	IDFULL	IDOK	IDS
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 return zero when read.

Bit 1: IDFULL – ID Full Flag

This bit indicates if the IDSCAN buffer has already received 7 bytes.

0: fewer than 7 bytes have been received

1: 7 bytes have been received (no more bytes will be accepted)

The IDFULL flag can generate an interrupt when masked in IDC.IDFIM. The IDFULL flag is automatically cleared when the interrupt is executed or by writing a “1” to its bit location.

Bit 0: IDOK – ID Check OK

The first time the ID check produces a match, this bit is set to “1”. The bit stays at “1” even if a subsequent ID check does not match. Only writing a “1” to this bit or re-enabling the complete block resets it to “0”.

IDB0 – ID Byte 0

Bit	7	6	5	4	3	2	1	0	
	IDB0[7:0]								IDB0
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: IDB0 – ID Byte 0

Configuration for ID byte 0.

IDB1 – ID Byte 1

Bit	7	6	5	4	3	2	1	0	
	IDB1[7:0]								IDB1
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: IDB1 – ID Byte 1

Configuration for ID byte 1.

IDB2 – ID Byte 2

Bit	7	6	5	4	3	2	1	0	
	IDB2[7:0]								IDB2
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: IDB2 – ID Byte 2

Configuration for ID byte 2.

IDB3 – ID Byte 3

Bit	7	6	5	4	3	2	1	0	
	IDB3[7:0]								IDB3
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: IDB3 – ID Byte 3

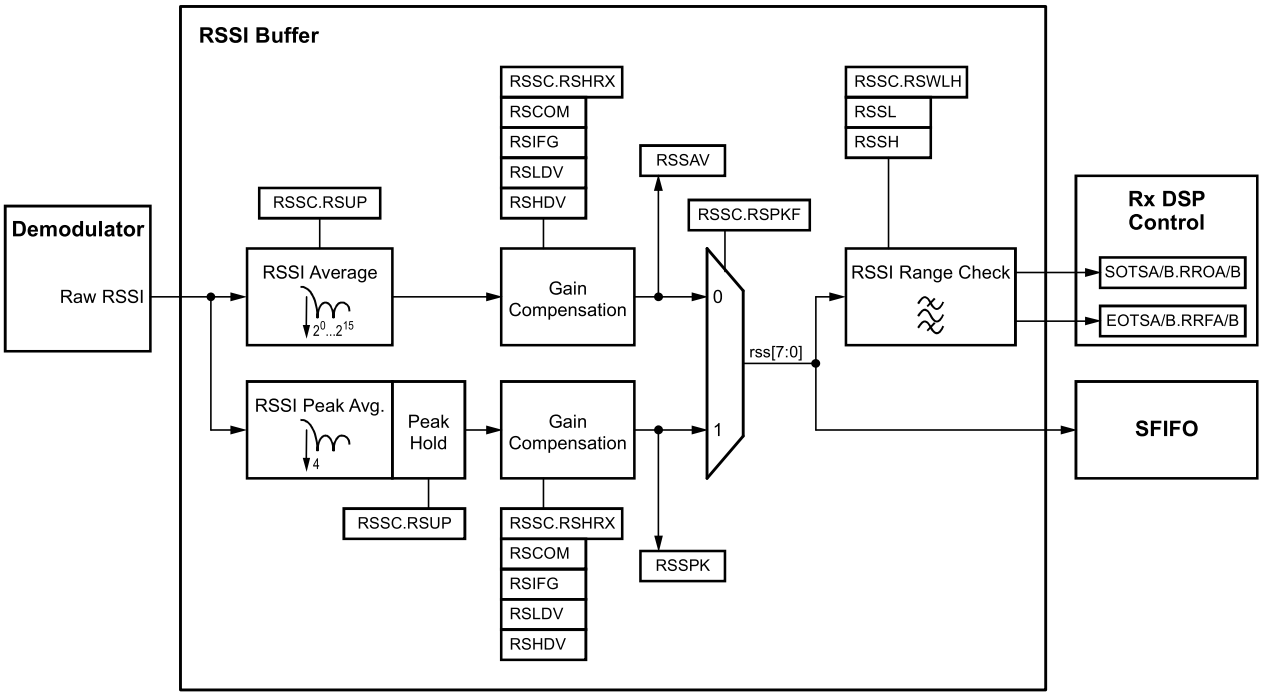
Configuration for ID byte 3.

3.4.3.6 RSSI Buffer

RSSI Buffer Overview

The RSSI buffer module provides an averaged RSSI (received signal strength indication) value, a peak RSSI value and two flags indicating a failure or a success of the RSSI range check. Peak or average values can be collected in the SFIFO for bulk readout. A block diagram of the RSSI buffer is shown in Figure 3-15.

Figure 3-15. RSSI Buffer Block Diagram



The ASK demodulator provides a raw RSSI signal which is used as input for the average and peak value calculations. The averaging period can be set by the RSSC.RSUP register. A single RSSI value is forwarded after every averaging period. The peak detection forwards the maximum value within one averaging period. A four-sample smoothening of the raw RSSI data stream is used during the peak value detection.

The RSSI values are finally written to the 16-byte deep SFIFO. RSSC.RSPKF selects whether the average or the peak values are used. The values can be read from the SFIFO via SPI.

The RSSI buffer contains an additional RSSI range check which generates an RSSI range OK signal and an RSSI range fail signal when the RSSI measurement is used as a signal check. A window can be defined by the RSSL and RSSH register and, depending on the RSSC.RSWLH setting, the current RSSI value must be inside or outside the defined window for a successful check. The RSSI buffer is reset when activated in the corresponding PRR2.PRRS power reduction register (see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210).

If not stated otherwise, a detailed description of the required register bits is found in Section “RSSI Buffer Register Description” on page 140.

RSSI Buffer Register Description

RSSC – RSSI Configuration Register

Bit	7	6	5	4	3	2	1	0	
	-	RSPKF	0	RSWLH	RSUP[3:0]				RSSC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: Reserved Bit

This bit is reserved and returns zero when read.

Bit 6: RSPKF – RSSI Peak Values to SFIFO

RSIFC	Description
0	RSSI average values are forwarded to the SFIFO.
1	RSSI peak values are forwarded to the SFIFO.

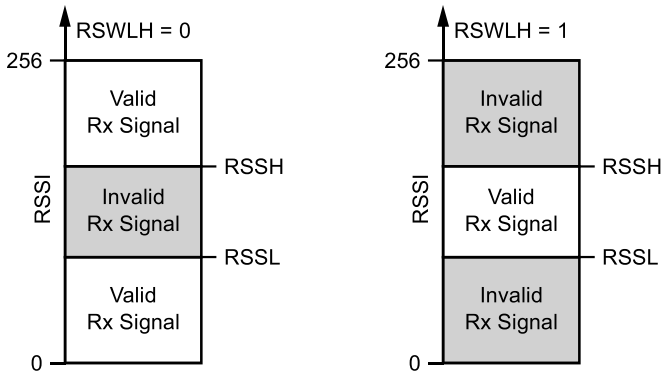
Bit 5: This bit must always be set to “0”

Bit 4: RSWLH – RSSI within Low and High Limits

Selects the valid received signal range in relation to the limit registers.

RSWLH	Description
0	A valid signal is expected to be outside the low and high limits set by the RSSL and RSSH registers. This mode can be used to reject signals within a certain RSSI range.
1	A valid signal is expected to be within the limits set by the RSSIL and RSSIH registers. This mode can be used to only accept signals within a certain RSSI range.

Figure 3-16. RSWLH Working Principle



Bit 3..0: RSUP[3:0] – RSSI Update Period

Selects the update period for RSSI averaging and peak calculation.

The register setting for a desired update period can be calculated:

$RSUP = round\ (\log_2\ (update\_period \times f_{CLK\_BB}))$ (39)

update_period: Required update period in s

f_{CLK_BB}: Baseband clock frequency in Hz. See equation (10) in Section “Bandwidth Scaling” on page 99.

RSCOM – RSSI Compensation Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	RSIFC	0	RSCOM
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 return zero when read.

Bit 1: RSIFC – RSSI IF Amplifier Compensation

Activates the RSSI compensation for the IF amplifier.

RSIFC	Description
0	IF amplifier gain RSSI compensation is deactivated.
1	IF amplifier gain RSSI compensation is activated. The RSIFG is subtracted from the RSSI value.

Bit 0: This bit must always be set to “0”

RSSL – RSSI Low Threshold for Signal Check

Lower threshold for signal validity check. See the RSSC.RSWLH description for details.

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

RSSH – RSSI High Threshold for Signal Check

Upper threshold for signal validity check. See the RSSC.RSWLH description for details.

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

RSIFG – This register is loaded by firmware and must not be changed.

RSLDV – This register is loaded by firmware and must not be changed**.**

RSHDV – This register is loaded by firmware and must not be changed**.**

RSSPK – RSSI Peak Value

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

This register provides the peak RSSI value from the most recent update period. The peak calculation is done block-wise and is updated after each RSSI update period defined by the RSUP[3:0] registers. The value is updated while the Rx DSP is working and the RSSI buffer is enabled in the corresponding power reduction register. The signal power at the watched 50Ω antenna input can be calculated by formula (1) on 47.

Note: The RSSI peak and RSSI average value updates are synchronized so they can be used to compare the peak and average value for an update period.

RSSAV – RSSI Average Value

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

This register provides the average RSSI value of the last update period. The averaging is done block-wise and is updated after each RSSI update period defined by the RSUP[3:0] registers. The value is updated while the Rx DSP is working and the RSSI buffer is enabled in the corresponding power reduction register. The signal power at the watched 50Ω antenna input can be calculated by formula (1) on page 47.

Note: The RSSI peak and RSSI average value updates are synchronized so they can be used to compare the peak and the average value for an update period.

3.4.3.7 Rx DSP Control

The Rx DSP control block contains the high level control and status signals of the Rx DSP.

A fixed sequence has to be observed when the Rx path is enabled. This is normally done by the Atmel firmware.

The procedure to enable the Rx DSP path from the reset state is:

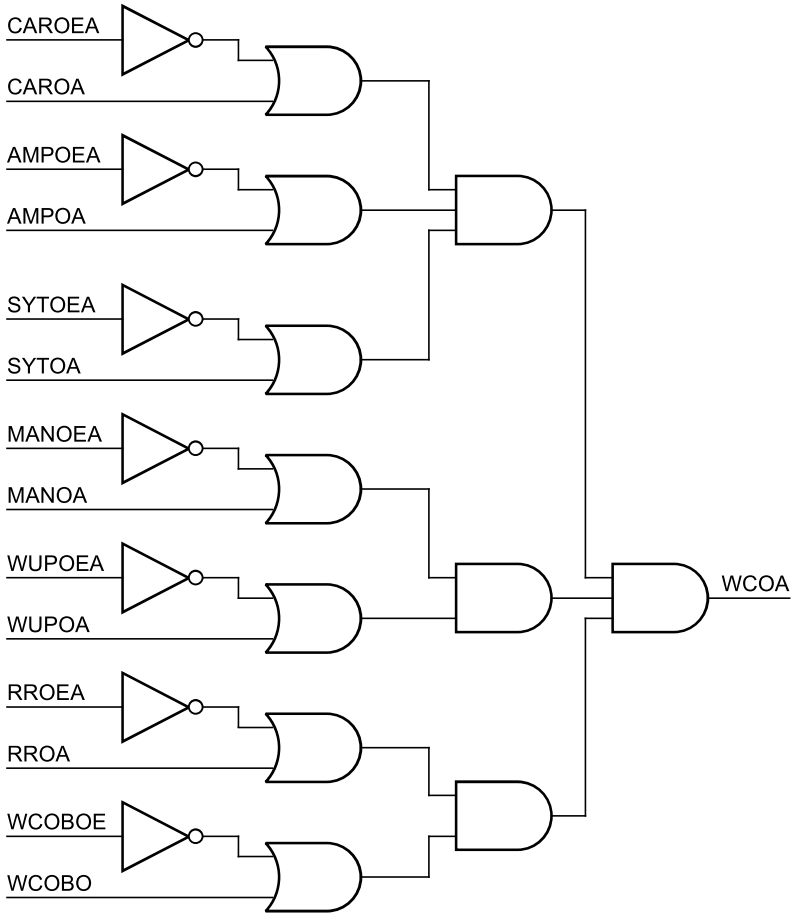
1. Start the channel filter.
   a) Set the channel filter configuration
      i. CHDN(ADCDN, BBDN) register
      ii. CHCR(BWM) register
   b) Enable CLK_{ADIV} if required RDCR.ADIVEN= [1|0].
   c) RDCR.RDPU = 1; enable the Rx DSP power
   d) Wait 200ns for the power supply settling
   e) RDCR.RDEN=1; enable the Rx DSP receive path
   f) RDPR.PRFLT=0; enable channel filter operation
   g) Wait for the channel filter settling time
2. Start the demodulator.
   a) Set the demodulator configuration (can also be set at 1.a)
   b) RDPR.PRPTA/B =0; enable demodulator
3. Rx DSP is operational.

Wake Check OK

The “Wake Check OK” (WCO) signal provides a flexible means for detecting a valid preamble signal. Up to seven checks can be activated as a condition for a successful wake check (WCOA/B).

Figure 3-17 shows the logic structure of the wake check for receiving path A.
Path B is identical except for the signal name endings which must be replaced by a “B”.

Figure 3-17. WCOA Logic



Each enable signal (ending with ..EA) activates the corresponding condition. For example, CAROEA = 1 activates the CAROA (“Carrier Check OK on Path A”) signal as a necessary condition for WCOA to become 1. If a check is not activated (e.g., CAROEA = 0), it is considered successful from the beginning. All activated checks have to be true to set the WCOA register (not shown in Figure 3-17).

For further details about the activation of the conditions, see the SOTCA/B register description. A detailed explanation of the actual signal checks is found at the register description of the SOTSA/B registers. The handling of the WCOA/B flag is described in the RDSIFR register. For all register descriptions, see Section “Rx DSP Control Register Description” on page 146.

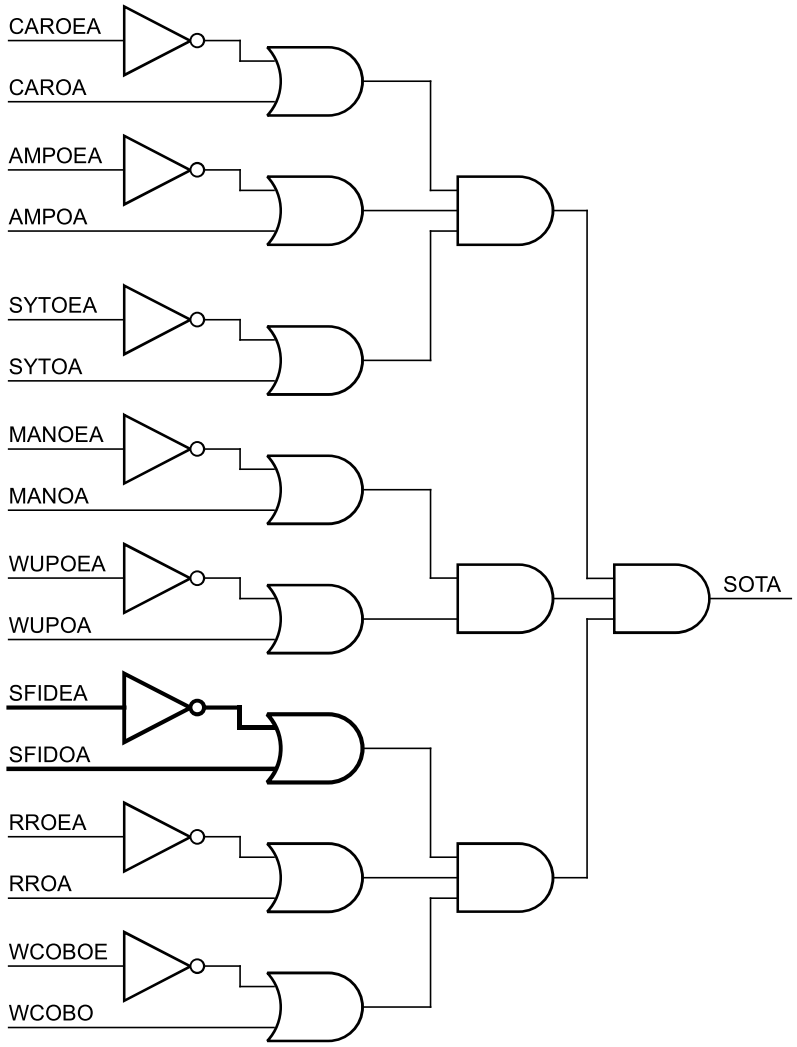
Start of Telegram

The “Start of Telegram” signal provides a flexible means of detecting a valid preamble signal. Up to eight checks can be activated as a condition for a successful start of telegram check (SOTA/B). The start of telegram conditions are the same as the wake check OK conditions with the additional SFIDOA check.

Figure 3-18 shows the logic structure of the start of telegram check for the receiving path A.

Path B is identical except for the signal name endings which must be replaced by a “B”.

Figure 3-18. SOTA Logic



Each enable signal (ending with ..EA) activates the corresponding condition. For example, SFIDOEA = 1 activates the SFIDOA (“Start Frame ID OK on Path A”) signal as a necessary condition for SOTA to become 1. If a check is not activated (e.g., SFIDOEA = 0), it is considered successful from the beginning. All activated checks have to be true to set the SOTA register (not shown in Figure 3-18).

For further details about the activation of the conditions, see the SOTCA/B register description. A detailed explanation of the actual signal checks is found at the register description of the SOTSA/B registers. The handling of the SOTA/B flag is described in the RDSIFR register. For all register descriptions, see Section “Rx DSP Control Register Description” on page 146.

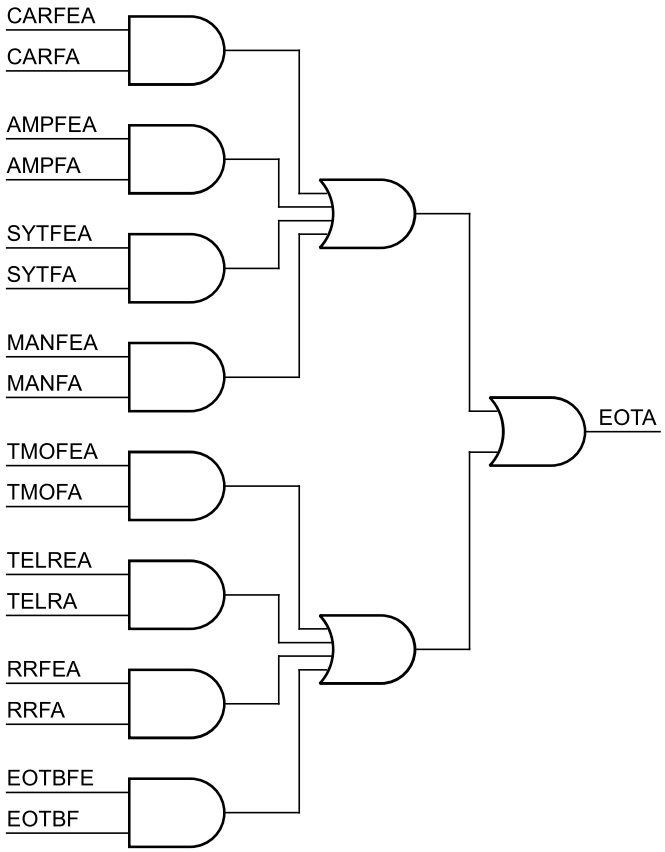
End of Telegram

The “End of Telegram” signal provides a flexible means of detecting an invalid signal or the end of a telegram. Up to eight checks can be activated to indicate an invalid signal (EOTA/B).

Figure 3-19 shows the logic structure of the end of telegram check for the receiving path A.

Path B is identical except for the signal name endings which must be replaced by a “B”.

Figure 3-19. EOTA Logic



Each enable signal (ending with ..EA) activates the corresponding condition. For example, MANFEA = 1 activates the MANFA (“Manchester Coding Conformity Check Failure on Path A”) signal as a necessary condition for EOTA to become 1. If a check is not activated (e.g., MANFEA = 0), it is ignored from the beginning. Each activated check alone can trigger the EOTA register (not shown in Figure 3-19).

For further details about the activation of the conditions, see the EOTCA register. A detailed explanation of the actual signal checks is found at the register description of the EOTSA registers. The handling of the EOTA/B flag is described in the RDSIFR register. For all register descriptions, see Section “Rx DSP Control Register Description” on page 146.

Rx DSP Control Register Description

RDCR – Rx DSP Control Register

Controls the operation of the Rx DSP block.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	RDEN	ADIVEN	RDPU	RDCR
Read/Write	R	R	R	R	R	R/W	R/W(1)	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. Writable only if RDCR.RDEN=0

Bits 7..3: Reserved Bits

These bits are reserved and return zero when read.

Bit 2: RDEN – Rx DSP Enable

Table 3-30. RDEN Bit Details

Value	Description
0	Rx DSP is disabled, all Rx DSP registers are in the asynchronous reset state. The power reduction bits for the Rx DSP are set to “1” (RDPR.PRFLT, RDPR.PRPTA/B, PRR2.PRXA/B).
1	Rx DSP is enabled with some delay to assure synchronous operation. Some clock dividers are running to allow the setting of configuration data. The channel filter and the demodulator have to be enabled separately by clearing the corresponding power reduction registers (RDPR.PRFLT, RDPR.PRPTA/B, PRR2.PRXA/B).

Bit 1: ADIVEN - Divided ADC Clock Enable

Enables CLK_{ADIV} generator.

Caution! This bit can only be modified while RDCR.RDEN=0. A clock is generated if the ADC is running, and the Rx DSP is enabled (RDCR.RDEN=1) because this is the clock source for the generator.

Table 3-31. ADIVEN Bit Details

Value	Description
0	Divided ADC clock output is disabled.
1	Divided ADC clock output is enabled.

Bit 0: RDPU – Rx DSP Power Up

Enables the power supply for the decimation filter at the beginning of the Rx DSP path.

RDPR – Rx DSP Power Reduction Register

This register is used to disable the clocks for several parts of the Rx DSP path. Each block with a disabled clock is automatically put into reset state.

A gradual power reduction scheme allows a step-by-step enabling of the blocks and a fast restart of the required blocks. A disabled block has no active clock source and a reset is executed in the internal registers. The external settings of the block are preserved. Enabling the power reduction at the beginning of the Rx path disables and resets all subsequent circuits to assure consistent operation.

A push synchronizer is used to transfer the settings from the AVR® clock domain to the asynchronous receive path. The user has to carry out the following procedure to modify the registers:

Check that the RDPR register is not busy (RDPR.RDPRF=0).

Write a new configuration to the RDPR register. The data is taken by the register and transferred to the Rx_DSP block. The RDPR.RDPRF flag is HIGH while this operation is running, indicating that further writing is blocked.

The settings are active and can be modified again when the operation is complete (RDPR.RDPRF = 0).

Note: Data can be written to this register only if RDPR.RDPRF=0. This is because a handshake is activated whenever data is written to RDPR and RDPR.RDPRF=1 indicates that the procedure is not completed yet. It is required to wait until RDPR.RDPRF becomes 0. The only way to abort the procedure is to disable and reset the complete Rx DSP block by setting RDCR.RDEN=0.

Bit	7	6	5	4	3	2	1	0	
	RDPRF	**ARDPRF**	**APRPTA**	**APRPTB**	**1**	**PRFLT**	**PRPTA**	**PRPTB**	**RDPR**
Read/Write	R	R	R	R	R/W	R/W	R/W [1)]	R/W [2)]	
Initial Value	0	0	0	0	1	1	1	1	

Notes: 1. Writable only if DMCRA.DMARA=0

2. Writable only if DMCRB.DMARB=0

The bits in this register are reset (in addition to the normal resets) if RDCR.RDEN=0.

Bit 7: RDPRF – Rx DSP Power Reduction Register (RDPR) Busy Flag

Value	Description
0	IDLEMode: The circuit is ready to start an update cycle. Writing new settings to the RDPR register is possible.
1	The RDPR is performing an update cycle. Writing to RDPR is blocked until the setting is transferred to the Rx DSP block.

Bit 6: ARDPRF – Automatic Rx DSP Power Reduction Flag

This is a read-only flag for debugging purposes. It indicates an automatic update cycle of the RDPR.ARPTA/B bits. This works independently of the RDPRF flag.

Bit 5: APRPTA – Automatic Power Reduction of Path A

This is a read-only flag for debugging purposes. It displays the current power reduction status of the automatic demodulator restart feature controlled by the DMCRA.DMARA bit. This bit is kept LOW while the automatic restart mode is disabled, giving full control to the PRPTA flag. The receiving path A is enabled and working only if ARPTA=0 and PRPTA=0. ARPTA is temporarily set to HIGH and back to LOW if the automatic mode is enabled (DMCRA.DMARA=1) and an EOTA condition becomes true.

Bit 4: APRPTB – Automatic Power Reduction of Path B

This is a read-only flag for debugging purposes. It displays the current power reduction status of the automatic demodulator restart feature controlled by the DMCRB.DMARB bit. This bit is kept LOW while the automatic restart mode is disabled, giving full control to the PRPTB flag. The receiving path B is enabled and working only if ARPTB=0 and PRPTB=0. ARPTB is temporarily set to HIGH and back to LOW if the automatic mode is enabled (DMCRB.DMARB=1) and an EOTB condition becomes true.

Bit 3: Reserved, must always be set to 1

Bit 2: PRFLT – Power Reduction Register for the Digital Channel Filter

Value	Description
0	Channel filter is enabled. The clocks are running.
1	Channel filter and the subsequent blocks (demodulator and receiving path A and path B) are disabled by stopping the clocks. A reset of the internal registers and the corresponding clock dividers is performed.

Bit 1: PRPTA – Power Reduction Register for Receiving Path A

This bit activates the power reduction mode for the entire receiving path A. Path A is enabled if PRPTA=0 and APRPTA=0. The behavior of the ARPTA bit depends on the DMCRA.DMARA configuration.

PRPTA	DMARA	Description
0	0	The receiving path A is enabled. The demodulator, frame synchronizer, and Rx buffer are running.
0	1	Receiving path A is in automatic restart mode. Path A is receiving until an enabled EOT condition occurs. A hardware-controlled reset with subsequent restart is performed. This mode is useful for repeated attempts to find a signal on the same frequency.
1	x	Receiving path A is disabled. The demodulator, clock recovery, symbol timing check, frame synchronizer, and Rx buffer are disabled by stopping their clocks. A reset of the internal registers and the corresponding clock dividers is performed. The external settings for the circuits are preserved to allow a fast restart.

Bit 0: PRPTB – Power Reduction Register for Receiving Path B

This bit activates the power reduction mode for the entire receiving path B. Path B is enabled if PRPTB=0 and APRPTB=0. The behavior of the ARPTB bit depends on the DMCRB.DMARB configuration.

PRPTB	DMARB	Description
0	0	The receiving path B is enabled. The demodulator, frame synchronizer, and Rx buffer are running.
0	1	Receiving path B is in automatic restart mode. Path B is receiving until an enabled EOT condition occurs. A hardware-controlled reset with subsequent restart is performed. This mode is useful for repeated attempts to find a signal on the same frequency.
1	x	Receiving path B is disabled. The demodulator, clock recovery, symbol timing check, frame synchronizer, and Rx buffer are disabled by stopping their clocks. A reset of the internal registers and the corresponding clock dividers is performed. The external settings for the circuits are preserved to allow a fast restart.

RDSIFR – Rx DSP Status Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	WCOB	**WCOA**	**SOTB**	**SOTA**	**EOTB**	**EOTA**	**NBITB**	**NBITA**	**RDSIFR**
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: WCOB – Wake Check OK on Path B

Set if the receiving path B is enabled (RDCR.RDEN=1, RDPR.PRFLT=0, RDPR.PRPTB=0) and all start of telegram conditions that are configured in the SOTCB[7:6] and SOTCB[4:0] register are true. The SFID condition SOTCB[5] is ignored for the WCOB interrupt.

It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to its bit location. This interrupt is a reduced version of the SOTB interrupt allowing some actions to be started before the SFIDB match occurs. Its conditions can also be used to enable the TMDO output if configured accordingly.

Bit 6: WCOA – Wake Check OK on Path A

Set if the receiving path A is enabled (RDCR.RDEN=1, RDPR.PRFLT=0, RDPR.PRPTA=0) and all start of telegram conditions that are configured in the SOTCA[7:6] and SOTCA[4:0] register are true. The SFID condition SOTCA[5] is ignored for the WCOA interrupt.

It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to its bit location. This interrupt is a reduced version of the SOTA interrupt allowing some actions to be started before the SFIDA match occurs. Its conditions can also be used to enable the TMDO output if configured accordingly.

Bit 5: SOTB – Start Of Telegram on Path B

Set if the receiving path B is enabled (RDCR.RDEN=1, RDPR.PRFLT=0, RDPR.PRPTB=0) and all start of telegram conditions that are configured in the STOCB register are true. It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to its bit location.

Bit 4: SOTA – Start Of Telegram on Path A

Set if the receiving path A is enabled (RDCR.RDEN=1, RDPR.PRFLT=0, RDPR.PRPTA=0) and all start of telegram conditions that are configured in the SOTCA register are true. It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to its bit location.

Bit 3: EOTB – End of Telegram on Path B

Set if at least one end of telegram condition configured in EOTCB is true. It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to the EOTB bit location.Writing a “1” to the EOTB bit location also clears all EOTSB flags.

Bit 2: EOTA – End of Telegram on Path A

Set if at least one end of telegram condition configured in EOTCA is true. It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to the EOTA bit location.Writing a “1” to the EOTA bit location also clears all EOTSA flags.

Bit 1: NBITB – New Bit on Rx Pat B

Set if a new bit is received on data path B. It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to its bit location.

Bit 0: NBITA – New Bit on Rx Path A

Set if a new bit is received on data path A. It is automatically cleared when the corresponding interrupt is executed or it can be cleared by writing a “1” to its bit location.

RDSIMR – Rx DSP Status Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	WCOBM	**WCOAM**	**SOTBM**	**SOTAM**	**EOTBM**	**EOTAM**	**NBITBM**	**NBITAM**	**RDSIMR**
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: WCOBM – Wake Check OK on Path B Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.WCOB flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the WCOB bit in RDSIFR is set.

Bit 6: WCOAM – Wake Check OK on Path A Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.WCOA flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the WCOA bit in RDSIFR is set.

Bit 5: SOTBM – Start Of Telegram on Rx Path B Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.SOTB flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the SOTB bit in RDSIFR is set.

Bit 4: SOTAM – Start Of Telegram on Rx Path A Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.SOTA flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the SOTA bit in RDSIFR is set.

Bit 3: EOTBM – End Of Telegram on Rx Path B Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.EOTB flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the EOTB bit in RDSIFR is set.

Bit 2: EOTAM – End Of Telegram on Rx Path A Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.EOTA flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the EOTA bit in RDSIFR is set.

Bit 1: NBITBM – New Bit on Rx Path B Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.NBITB flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the NBITB bit in RDSIFR is set.

Bit 0: NBITAM – New Bit on Rx Path A Interrupt Mask Register

Writing this bit to “1” enables the interrupt on the RDSIFR.NBITA flag. An interrupt is generated only if this bit is HIGH, the global interrupt flag in the SREG is activated and the NBITA bit in RDSIFR is set.

RDOCR – Rx DSP Output Control

This register controls the direct signal outputs from the Rx DSP to the receiver pins.

Bit	7	6	5	4	3	2	1	0	
	-	**0**	**0**	**ETRPB**	**ETRPA**	**TMDS[1:0]**		**-**	**RDOCR**
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: Reserved Bit

This bit is reserved and returns zero when read.

Bits 6..5: Reserved Bits

These bits must always be zero.

Bit 4: ETRPB – Enable Transparent Path B

Enables the raw data output of receiving path B on pin 19/TRPB. The corresponding data direction register of the port has to be set as output to enable the driver.

Bit 3: ETRPA - Enable Transparent Path A

Enables the raw data output of receiving path A on pin 16/TRPA. The corresponding data direction register of the port has to be set as output to enable the driver.

Bits 2..1: TMDS1, TMDS0 – Transparent Mode Data Select

Select the data to be routed to the TMDO pin. The corresponding data direction registers of the TMDO and TMDO_CLK ports have to be set as output to enable the driver.

TMDS1	TMDS0	
Bit 2	Bit 1	Description
0	0	No transparent demodulator output on TMDO. Transparent output function is deactivated and has no influence on the pin behavior.
0	1	Receiver path A data is selected for transparent data output. The data signal is visible at TMDO (pin 17) after a successful wake check (WCOA). The corresponding data clock is visible at the TMDO_CLK (pin 19) pin after a successful start of telegram (SOTA) detection.
1	0	Receiver path B data is selected for transparent data output. The data signal is visible at TMDO (pin 17) after a successful wake check (WCOB). The corresponding data clock is visible at the TMDO_CLK (pin 19) pin after a successful start of telegram (SOTB) detection.
1	1	No transparent demodulator output on TMDO. Transparent output function is deactivated and has no influence on the pin behavior.

Bit 0: Reserved Bit

This bit is reserved and returns zero when read.

SOTCA – Start Of Telegram Conditions for Path A

This is the configuration register for the wake check OK on path A interrupt (RDSIFR.WCOA) and start of telegram on path A interrupt (RDSIFR.SOTA). A condition is activated by setting the corresponding bit to high. The interrupts are set if all activated conditions are met. The current status of the conditions can be checked by reading the SOTSA register.

Bit	7	6	5	4	3	2	1	0	
	WCOBOE	**RROEA**	**SFIDEA**	**WUPEA**	**MANOEA**	**SYTOEA**	**AMPOEA**	**CAROEA**	**SOTCA**
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: WCOBOE – Wake Check OK from Path B OK Enable

Writing this bit to “1” selects the wake check OK condition for path B (RDSIFR.WCOB) as a precondition for a successful wake check OK on path A (RDSIFR.WCOA) and start of telegram on path A (RDSIFR.SOTA).

Bit 6: RROEA – RSSI Range OK Enable for Path A

Writing this bit to “1” enables the RSSI range OK on path A (SOTSA.RROA) as a condition for the wake check OK on path A (RDSIFR.WCOA) and start of telegram on path A (RDSIFR.SOTA) interrupts. The SOTSA.RROA is ignored if this bit is cleared.

Bit 5: SFIDEA – Start of Frame Identifier Match Enabled for Path A

Writing this bit to “1” enables the start frame ID match on path A as a condition for the start of telegram on path A (RDSIFR.SOTA) interrupt. The start frame ID result is ignored if this bit is cleared.

Bit 4: WUPEA – Wake-Up Pattern Match Enabled for Path A

Writing this bit to “1” enables the wake-up pattern match on path A as a condition for the wake check OK on path A (RDSIFR.WCOA) and start of telegram on path A (RDSIFR.SOTA) interrupts. The wake-up pattern result is ignored if this bit is cleared.

Bit 3: MANOEA – Manchester Coding OK Enabled for Path A

Writing this bit to “1” enables the Manchester coding check OK output from path A as a condition for the wake check OK on path A (RDSIFR.WCOA) and start of telegram on path A (RDSIFR.SOTA) interrupts. The Manchester coding check result is ignored if this bit is cleared.

Bit 2: SYTOEA – Symbol Timing OK Enabled for Path A

Writing this bit to “1” enables the symbol timing check OK output from path A as a condition for the wake check OK on path A (RDSIFR.WCOA) and start of telegram on path A (RDSIFR.SOTA) interrupts. The symbol timing check result is ignored if this bit is cleared.

Bit 1: AMPOEA – Amplitude OK Enabled for Path A

Writing this bit to “1” enables the demodulator amplitude check OK output from path A as a condition for the wake check OK on path A (RDSIFR.WCOA) and start of telegram on path A (RDSIFR.SOTA) interrupts. The demodulator amplitude check result is ignored if this bit is cleared.

Bit 0: CAROEA – Carrier Check OK Enabled for Path A

Writing this bit to “1” enables the carrier check OK output from path A as a condition for the wake check OK on path A (RDSIFR.WCOA) and start of telegram on path A (RDSIFR.SOTA) interrupts. The carrier check result is ignored if this bit is cleared.

SOTC1A – Start of Telegram Conditions 1 for Path A

This register is used only for hardware-controlled automatic telegram reception. It stores the SOTCA settings that are valid from the start of the reception until a wake check OK (RDSIFR.WCOA) is detected. The sequencer state machine copies its content to the SOTCA register at the beginning of reception.

Bit	7	6	5	4	3	2	1	0	
	WCOBOE	**RROEA**	**SFIDEA**	**WUPEA**	**MANOEA**	**SYTOEA**	**AMPOEA**	**CAROEA**	SOTC1A
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	

The bit descriptions are found at the SOTCA target register.

SOTC2A – Start of Telegram Conditions 2 for Path A

This register is used only for hardware-controlled automatic telegram reception. It stores the SOTCA settings that are valid from the wake check OK (RDSIFR.WCOA) to start of telegram OK (RDSIFR.SOTA). The sequencer state machine copies its content immediately after RDSIFR.WCOA detection to the SOTCA register.

Bit	7	6	5	4	3	2	1	0	
	WCOBOE	**RROEA**	**SFIDEA**	**WUPEA**	**MANOEA**	**SYTOEA**	**AMPOEA**	**CAROEA**	**SOTC2A**
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	

The bit descriptions are found at the SOTCA target register.

SOTCB – Start Of Telegram Conditions for Path B

This is the configuration register for the wake check OK on path B interrupt (RDSIFR.WCOB) and start of telegram on path B interrupt (RDSIFR.SOTB). A condition is activated by setting the corresponding bit to high. The interrupts are set if all activated conditions are met. The current status of the conditions can be seen in the SOTSB register.

Bit	7	6	5	4	3	2	1	0	
	WCOAOE	**RROEB**	**SFIDEB**	**WUPEB**	**MANOEB**	**SYTOEB**	**AMPOEB**	**CAROEB**	**SOTCB**
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: WCOAOE – Wake Check OK from Path A OK Enable

Writing this bit to “1” selects the wake check OK on path A condition (RDSIFR.WCOA) as a precondition for a successful wake check OK on path B (RDSIFR.WCOB) and start of telegram on path B (RDSIFR.SOTB).

Bit 6: RROEB – RSSI Range OK Enable for Path B

Writing this bit to “1” enables the RSSI range OK on path B (SOTSB.RROB) as a condition for the wake check OK on path B (RDSIFR.WCOB) and start of telegram on path B (RDSIFR.SOTB) interrupts. The SOTSB.RROB is ignored if this bit is cleared.

Bit 5: SFIDEB – Start of Frame Identifier Match Enabled for Path B

Writing this bit to “1” enables the start frame ID match on path B as a condition for the start of telegram on path B (RDSIFR.SOTB). The start frame ID result is ignored if this bit is cleared.

Bit 4: WUPEB – Wake-Up Pattern Match Enabled for Path B

Writing this bit to “1” enables the wake-up pattern match on path B as a condition for the wake check OK on path B (RDSIFR.WCOB) and start of telegram on path B (RDSIFR.SOTB) interrupts. The wake-up pattern result is ignored if this bit is cleared.

Bit 3: MANOEB – Manchester Coding OK Enabled for Path B

Writing this bit to “1” enables the Manchester coding check OK output from path B as a condition for the wake check OK on path B (RDSIFR.WCOB) and start of telegram on path B (RDSIFR.SOTB) interrupts. The Manchester coding check result is ignored if this bit is cleared.

Bit 2: SYTOEB – Symbol Timing OK Enabled for Path B

Writing this bit to “1” enables the symbol timing check OK output from path B as a condition for the wake check OK on path B (RDSIFR.WCOB) and start of telegram on path B (RDSIFR.SOTB) interrupts. The symbol timing check result is ignored if this bit is cleared.

Bit 1: AMPOEB – Amplitude OK Enabled for Path B

Writing this bit to “1” enables the demodulator amplitude check OK output from path B as a condition for the wake check OK on path B (RDSIFR.WCOB) and start of telegram on path B (RDSIFR.SOTB) interrupts. The demodulator amplitude check result is ignored if this bit is cleared.

Bit 0: CAROEB – Carrier Check OK Enabled for Path B

Writing this bit to “1” enables the carrier check OK output from path B as a condition for the wake check OK on path B (RDSIFR.WCOB) and start of telegram on path B (RDSIFR.SOTB) interrupts. The carrier check result is ignored if this bit is cleared.

SOTC1B – Start of Telegram Conditions 1 for Path B

This register is used only for hardware-controlled automatic telegram reception. It stores the SOTCB settings that are valid from the start of the reception until a wake check OK (RDSIFR.WCOB) is detected. The sequencer state machine copies its content at the beginning of the reception to the SOTCB register.

Bit	7	6	5	4	3	2	1	0	
	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB	SOTC1B
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	

The bit descriptions are found at the SOTCB target register.

SOTC2B – Start of Telegram Conditions 2 for Path B

This register is used only for hardware-controlled automatic telegram reception. It stores the SOTCB settings that are valid from the wake check OK (RDSIFR.WCOB) to start of telegram OK (RDSIFR.SOTB). The sequencer state machine copies its content immediately after RDSIFR.WCOB detection to the SOTCB register.

Bit	7	6	5	4	3	2	1	0	
	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB	SOTC2B
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	

The bit descriptions are found at the SOTCB target register.

SOTSA – Start Of Telegram Status for Path A

This register displays the status of all start of telegram conditions. The corresponding bit is set to “1” if the condition has matched. It is cleared if a “1” is written to its position or the receiving path A is disabled (RDPR.PRFLT=1 or RDPR.PRPTA=1).

The flags are handled by the hardware if the get_rx_telegram state machine is activated (SSMRR.SSMR). In this case the flags are cleared if an activated error condition for RDSIFR.EOTA occurs. This is done automatically by toggling the RDPR.APRPTA bit.

Bit	7	6	5	4	3	2	1	0	
	WCOBO	RROA	SFIDOA	WUPOA	MANOA	SYTOA	AMPOA	CAROA	SOTSA
Read/Write	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. Set to hardware-controlled read-only mode if the get_rx_telegram state machine is active.

Bit 7: WCOBO – Wake Check OK on Path B OK

The bit is set at a successful wake check OK from path B (RDSIFR.WCOB) that can be used as a precondition for a successful wake check (RDSIFR.WCOA) and start of telegram (RDSIFR.SOTA) detection on path A.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

Bit 6: RROA – RSSI Range OK on Path A

This bit is set if the received signal strength on path A is within the expected range. The RSSI comparison is based on the first averaged RSSI sample after Rx start-up. A failure of this check is indicated by the setting of the RSSFA flag in the EOTSA register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

Bit 5: SFIDOA – Start of Frame Identifier Matched on Path A

This bit indicates a successful correlator-based start of frame ID check.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

Bit 4: WUPOA – Wake-Up Pattern Matched on Path A

This bit is set if the wake-up pattern match occurred.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

Bit 3: MANOA – Manchester Coding OK on Path A

This bit is set if the Manchester coding check is OK. The Manchester coding is verified for the duration specified in the SYCSA register. If it was right for this duration, it is considered OK. A failure of this check is indicated by the setting of the MANFEA flag in the EOTS register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

Bit 2: SYTOA – Symbol Timing OK on Path A

This bit is set if the symbol timing check is OK. The symbol timing is verified for the duration specified in the SYCSA register. If it was right for this duration, it is considered OK. A failure of this check is indicated by the setting of the SYTFEA flag in the EOTS register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

Bit 1: AMPOA – Amplitude OK on Path A

This bit is set if the signal amplitude inside the demodulator is above a specified threshold (DMMA.DMATA). A failure of this check is indicated by the setting of the AMPFA flag in the EOTSA register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

Bit 0: CAROA – Carrier Check OK on Path A

This bit is set by a successful signal carrier check. A failure of this check is indicated by the setting of the CARFA flag in the EOTSA register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTA interrupt flag, or the receiving path A is disabled.

SOTSB – Start Of Telegram Status for Path B

This register displays the status of all start of telegram conditions. The corresponding bit is set to “1” if the condition has matched. It is cleared if a “1” is written to its position or the receiving path B is disabled (RDPR.PRFLT=1 or RDPR.PRPTB=1).

The flags are handled by the hardware if the get_rx_telegram state machine (SSMRR.SSMR) is activated. In this case the flags are cleared if an activated error condition for RDSIFR.EOTB occurs. This is done automatically by toggling the RDPR.APRPTB bit.

Bit	7	6	5	4	3	2	1	0	
	WCOAO	**RROB**	**SFIDOB**	**WUPOB**	**MANOB**	**SYTOB**	**AMPOB**	**CAROB**	**SOTSB**
Read/Write	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1))	R/W(1)	R/W(1)	R/W(1)	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. Set to hardware-controlled read-only mode if the get_rx_telegram state machine is active

Bit 7: WCOAO – Wake Check OK on Path A OK

The bit is set at a successful wake check OK from path A (RDSIFR.WCOA) that can be used as a precondition for a successful wake check (RDSIFR.WCOB) and start of telegram (RDSIFR.SOTB) detection on path B.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

Bit 6: RROB – RSSI Range OK on Path B

This bit is set if the received signal strength on path B is within the expected range. The RSSI comparison is based on the first averaged RSSI sample after Rx start-up. A failure of this check is indicated by the setting of the RSSFB flag in the EOTSB register. It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

Bit 5: SFIDOB – Start of Frame Identifier Matched on Path B

This bit indicates a successful correlator-based start of frame ID check.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

Bit 4: WUPOB – Wake-Up Pattern Matched on Path B

This bit is set if the wake-up pattern match occurred.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

Bit 3: MANOB – Manchester Coding OK on Path B

This bit is set if the manchester coding check is OK. The manchester coding is verified for the duration specified in the SYCSB register. If it was right for this duration, it is considered OK. A failure of this check is indicated by the setting of the MANFEB flag in the EOTSB register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

Bit 2: SYTOB – Symbol Timing OK on Path B

This bit is set if the symbol timing check is OK. The symbol timing is verified for the duration specified in the SYCSB register. If it was right for this duration, it is considered OK. A failure of this check is indicated by the setting of the SYTFEB flag in the EOTSB register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

Bit 1: AMPOB – Amplitude OK on Path B

This bit is set if the signal amplitude inside the demodulator is above a specified threshold (DMMB.DMATB). A failure of this check is indicated by the setting of the AMPFB flag in the EOTSB register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

Bit 0: CAROB – Carrier Check OK on Path B

This bit is set by a successful signal carrier check. A failure of this check is indicated by the setting of the CARFB flag in the EOTSB register.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.SOTB interrupt flag, or the receiving path B is disabled.

EOTCA – End of Telegram Conditions for Path A

This is the configuration register for the end of telegram (EOT) interrupt on path A (RDSIFR.EOTA). A condition is activated by setting the corresponding bit to “1”. The EOTA interrupt flag in the RDSIFR register is set if any of the activated conditions fails. Inactive conditions are disabled by setting the corresponding bit to “0”.

Bit	7	6	5	4	3	2	1	0	
	EOTBFE	**RRFEA**	**TELREA**	**TMOFEA**	**MANFEA**	**SYTFEA**	**AMPFEA**	**CARFEA**	**EOTCA**
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: EOTBFE – End Of Telegram on Path B Fail Enable

Writing this bit to “1” selects the EOT on path B failed signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared.

Setting this bit to “1” can be useful when the corresponding wake check OK (RDSIFR.WCOB) from path B is selected as a wake-up or start of telegram condition. It allows for a restart of path A when an error occurred on path B.

Bit 6 RRFEA – RSSI Range Fail Enable on Path A

Writing this bit to “1” selects the RSSI range failed signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared.

Bit 5: TELREA – Telegram Length Reached Enable on Path A

Writing this bit to “1” selects the telegram length reached signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared.

Bit 4: TMOFEA – Time-out Fail Enable on Path A

Writing this bit to “1” selects the time-out fail signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared. See EOTSA.TMOFA bit for details on the functionality.

Bit 3: MANFEA – Manchester Coding Failed Enable for Path A

Writing this bit to “1” selects the Manchester coding failed signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared.

Bit 2: SYTFEA – Symbol Timing Check Failed Enable for Path A

Writing this bit to “1” selects the symbol timing check failed signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared.

Bit 1: AMPFEA – Amplitude Check Failed Enable for Path A

Writing this bit to “1” selects the demodulator amplitude check failed signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared.

Bit 0: CARFEA – Carrier Check Failed Enable for Path A

Writing this bit to “1” selects the carrier check failed signal as one condition for setting the RDSIFR.EOTA flag. This condition is ignored if the bit is cleared.

EOTC1A – End of Telegram Conditions 1 for Path A

This register is used only for hardware-controlled automatic telegram reception. It stores the EOTCA settings that are valid from the start of the reception until a wake check OK (RDSIFR.WCOA) is detected. The sequencer state machine copies its content at the beginning of the reception to the EOTCA register.

Bit	7	6	5	4	3	2	1	0	
	EOTBFE	**RRFEA**	**TELREA**	**TMOFEA**	**MANFEA**	**SYTFEA**	**AMPFEA**	**CARFEA**	**EOTC1A**
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	

The bit descriptions are found at the EOTCA target register.

EOTC2A – End of Telegram Conditions 2 for Path A

This register is used only for hardware-controlled automatic telegram reception. It stores the EOTCA settings that are valid from a valid wake check OK (RDSIFR.WCOA) up to the successful start of telegram (RDSIFR.SOTA) detection. The sequencer state machine copies its content at a successful RDSIFR.WCOA to the EOTCA register.

Bit	7	6	5	4	3	2	1	0	
	EOTBFE	**RRFEA**	**TELREA**	**TMOFEA**	**MANFEA**	**SYTFEA**	**AMPFEA**	**CARFEA**	**EOTC2A**
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	

The bit descriptions are found at the EOTCA target register.

EOTC3A – End of Telegram Conditions 3 for Path A

This register is used only for hardware-controlled automatic telegram reception. It stores the EOTCA settings that are valid from a valid start of telegram detection (RDSIFR.SOTA) until the end of the telegram. The sequencer state machine copies its content at a successful RDSIFR.SOTA check to the EOTCA register.

Bit	7	6	5	4	3	2	1	0	
	EOTBFE	**RRFEA**	**TELREA**	**TMOFEA**	**MANFEA**	**SYTFEA**	**AMPFEA**	**CARFEA**	**EOTC3A**
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	

The bit descriptions are found at the EOTCA target register.

EOTCB – End of Telegram Conditions for Path B

This is the configuration register for the EOT interrupt on path B (RDSIFR.EOTB). A condition is activated by setting the corresponding bit to “1”. The EOTB interrupt flag in the RDSIFR register is set if any of the activated conditions fails. Inactive conditions are disabled by setting the corresponding bit to “0”.

Bit	7	6	5	4	3	2	1	0	
	EOTAFE	**RRFEB**	**TELREB**	**TMOFEB**	**MANFEB**	**SYTFEB**	**AMPFEB**	**CARFEB**	**EOTCB**
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: EOTAFE – End Of Telegram on Path A Fail Enable

Writing this bit to “1” selects the EOT on path A failed signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared.

Setting this bit to “1” can be useful when the corresponding wake check OK (RDSIFR.WCOA) from path A is selected as a wake-up or start of telegram condition. It allows a restart of path B when an error occurred on path A.

Bit 6: RRFEB – RSSI Range Fail Enable on Path B

Writing this bit to “1” selects the RSSI range failed signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared.

Bit 5: TELREB – Telegram Length Reached Enable on Path B

Writing this bit to “1” selects the telegram length reached signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared.

Bit 4: TMOFEB – Time-out Fail Enable on Path B

Writing this bit to “1” selects the time-out fail signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared. See EOTSB.TMOFB bit for details on the functionality.

Bit 3: MANFEB – Manchester Coding Failed Enable for Path B

Writing this bit to “1” selects the Manchester coding failed signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared.

Bit 2: SYTFEB – Symbol Timing Check Failed Enable for Path B

Writing this bit to “1” selects the symbol timing check failed signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared.

Bit 1: AMPFEB – Amplitude Check Failed Enable for Path B

Writing this bit to “1” selects the demodulator amplitude check failed signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared.

Bit 0: CARFEB – Carrier Check Failed Enable for Path B

Writing this bit to “1” selects the carrier check failed signal as one condition for setting the RDSIFR.EOTB flag. This condition is ignored if the bit is cleared.

EOTC1B – End of Telegram Conditions 1 for Path B

This register is used only for hardware-controlled automatic telegram reception. It stores the EOTCB settings that are valid from the start of the reception until a wake check OK (RDSIFR.WCOB) is detected. The sequencer state machine copies its content at the beginning of the reception to the EOTCB register.

Bit	7	6	5	4	3	2	1	0	
	EOTAFE	**RRFEB**	**TELREB**	**TMOFEB**	**MANFEB**	**SYTFEB**	**AMPFEB**	**CARFEB**	**EOTC1B**
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	

The bit descriptions are found at the EOTCB target register.

EOTC2B – End of Telegram Conditions 2 for Path B

This register is used only for hardware-controlled automatic telegram reception. It stores the EOTCB settings that are valid from a valid wake check OK (RDSIFR.WCOB) up to the successful start of telegram (RDSIFR.SOTB) detection. The sequencer state machine copies its content at a successful RDSIFR.WCOB to the EOTCB register.

Bit	7	6	5	4	3	2	1	0	
	EOTAFE	**RRFEB**	**TELREB**	**TMOFEB**	**MANFEB**	**SYTFEB**	**AMPFEB**	**CARFEB**	**EOTC2B**
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	

The bit descriptions are found at the EOTCB target register.

EOTC3B – End of Telegram Conditions 3 for Path B

This register is used only for hardware-controlled automatic telegram reception. It stores the EOTCB settings that are valid from a valid start of telegram detection (RDSIFR.SOTB) until the end of the telegram. The sequencer state machine copies its content at a successful RDSIFR.SOTB check to the EOTCB register.

Bit	7	6	5	4	3	2	1	0	
	EOTAFE	**RRFEB**	**TELREB**	**TMOFEB**	**MANFEB**	**SYTFEB**	**AMPFEB**	**CARFEB**	**EOTC3B**
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	

The bit descriptions are found at the EOTCB target register.

EOTSA – End of Telegram Status on Path A

This is the status register for the EOT conditions on path A. The RDSIFR.EOTA interrupt flag is set if one of the fail conditions selected in the EOTCA register becomes true in the EOTSA register.

The bits in the EOTSA register are set if the corresponding fail condition is detected on the Rx path A. The flags are handled by the hardware if the get_rx_telegram state machine is activated (SSMRR.SSMR). In this case the flags are cleared if an activated error condition for RDSIFR.EOTA occurs. This is done automatically by toggling the RDPR.APRPTA bit.

Bit	7	6	5	4	3	2	1	0	
	EOTBF	**RRFA**	**TELRA**	**TMOFA**	**MANFA**	**SYTFA**	**AMPFA**	**CARFA**	**EOTSA**
Read/Write	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. Set to hardware-controlled read-only mode if the get_rx_telegram state machine is active.

Bit 7: EOTBF – End Of Telegram on Path B Flag

The bit is set to “1” simultaneously with the RDSIFR.EOTB interrupt flag register. It can be used to set the RDSIFR.EOTA flag based on RDSIFR.EOTB, allowing both paths to be restarted simultaneously. This is useful when the start of telegram condition on path A depends on the wake check OK (RDSIFR.WCOB) of path B.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

Bit 6: RRFA – RSSI Range Fail on Path A

This bit is set if the received signal strength on path A is within the expected range.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

Bit 5: TELRA – Telegram Length Reached on Path A

This bit is set if the target telegram length is reached on path A.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

Bit 4: TMOFA – Time-out Fail on Path A

This bit is set to “1” if a time-out occurs during wake check OK search on path A or during start of telegram search on path A. For the time-out duration see the description of the corresponding sequencer state machine registers. Each time-out can be deactivated by setting the duration to “0”.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

Bit 3: MANFA – Manchester Coding Failed on Path A

This bit is set if the Manchester coding check failed on path A.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

Bit 2: SYTFA – Symbol Timing Check Failed on Path A

This bit is set if the symbol timing check failed on path A.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

Bit 1: AMPFA – Amplitude Check Failed on Path A

This bit is set if the demodulator amplitude check failed on path A.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

Bit 0: CARFA – Carrier Check Failed on Path A

This bit is set if the carrier check failed on path A.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTA interrupt flag, a new value is written to the EOTCA register, or the receiving path A is disabled.

TESRA – Telegram Status Register on Path A

This is the status register for path A. It provides additional information compared to the EOTSA register but it cannot be used for an automatic restart of path A.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	EOTLA[1:0]		CRCOA	TESRA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bits 7..3: Reserved Bits

These bits are reserved and return zero when read.

Bits 2..1: EOTLA – End of Telegram Location on Path A

If the get_rx_telegram state machine is activated (SSMRR.SSMR = 1), these two bits indicate the location of the EOTA event within the telegram. These bits are not cleared when the get_rx_telegram state machine is disabled (SSMRR.SSMR = 0).

Table 3-32. EOTLA Bit Setting versus EOTA Location

EOTLA1	EOTLA0	Description
0	0	No EOTA
0	1	Before WCOA
1	0	Between WCOA and SOTA
1	1	After SOTA

Bit 0: CRCOA – Cyclic Redundancy Check OK on Path A

This bit is set when a telegram is received without CRC error on path A.

This bit is cleared when the Rx buffer is disabled (PRR2.PRXB=1) and enabled (PRR2.PRXB=0) again or when the Rx buffer is cleared (RXBC2.RXBCLR = 1).

EOTSB – End of Telegram Status on Path B

This is the status register for the EOT conditions on path B. The RDSIFR.EOTB interrupt flag is set if one of the fail conditions selected in the EOTCB register becomes true in the EOTSB register.

The bits in the EOTSB register are set if the corresponding fail condition is detected on the Rx path B. The flags are handled by the hardware if the get_rx_telegram state machine is activated (SSMRR.SSMR). In this case the flags are cleared if an activated error condition for RDSIFR.EOTB occurs. This is done automatically by toggling the RDPR.APRPTB bit.

Bit	7	6	5	4	3	2	1	0	
	EOTAF	**RRFB**	**TELRB**	**TMOFB**	**MANFB**	**SYTFB**	**AMPFB**	**CARFB**	**EOTSB**
Read/Write	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. Set to hardware-controlled read-only mode if the get_rx_telegram state machine is active.

Bit 7: EOTAF – End Of Telegram on Path A Flag

The bit is set to “1” simultaneously with the RDSIFR.EOTA interrupt flag register. It can be used to set the RDSIFR.EOTB flag based on RDSIFR.EOTA, allowing both paths to be restarted simultaneously. This is useful when the start of telegram condition on path B depends on the wake check OK (RDSIFR.WCOA) of path A.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

Bit 6: RRFB – RSSI Range Fail on Path B

This bit is set if the received signal strength on path B is within the expected range.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

Bit 5: TELRB – Telegram Length Reached on Path B

This bit is set if the target telegram length is reached on path B.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

Bit 4: TMOFB – Time-out Fail on Path B

This bit is set to “1” if a time-out occurs during a wake check OK search on path B or during start of telegram search on path B. For the time-out duration, see the description of the corresponding sequencer state machine registers. Each time-out can be deactivated by setting the duration to “0”.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

Bit 3: MANFB – Manchester Coding Failed on Path B

This bit is set if the Manchester coding check failed on path B.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

Bit 2: SYTFB – Symbol Timing Check Failed on Path B

This bit is set if the symbol timing check failed on path B.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

Bit 1: AMPFB – Amplitude Check Failed on Path B

This bit is set if the demodulator amplitude check failed on path B.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

Bit 0: CARFB – Carrier Check Failed on Path B

This bit is set if the carrier check failed on path B.

It is cleared if a “1” is written to its position, a “1” is written to the RDSIFR.EOTB interrupt flag, a new value is written to the EOTCB register, or the receiving path B is disabled.

TESRB – Telegram Status Register on Path B

This is the status register for path B. It provides additional information compared to the EOTSB register but it cannot be used for an automatic restart of the path B.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	EOTLB[1:0]		CRCOB	TESRB
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 return zero when read.

Bits 2..1: EOTLB – End of Telegram Location on Path B

If the get_rx_telegram state machine is activated (SSMRR.SSMR = 1), these two bits indicate the location of the EOTB event within the telegram. These bits are not cleared when the get_rx_telegram state machine is disabled (SSMRR.SSMR = 0).

Table 3-33. EOTLB Bit Setting versus EOTB Location

EOTLB1	EOTLB0	Description
0	0	No EOTB
0	1	Before WCOB
1	0	Between WCOB and SOTB
1	1	After SOTB

Bit 0: CRCOB – Cyclic Redundancy Check OK on Path B

This bit is set when a telegram is received without a CRC error on path B.

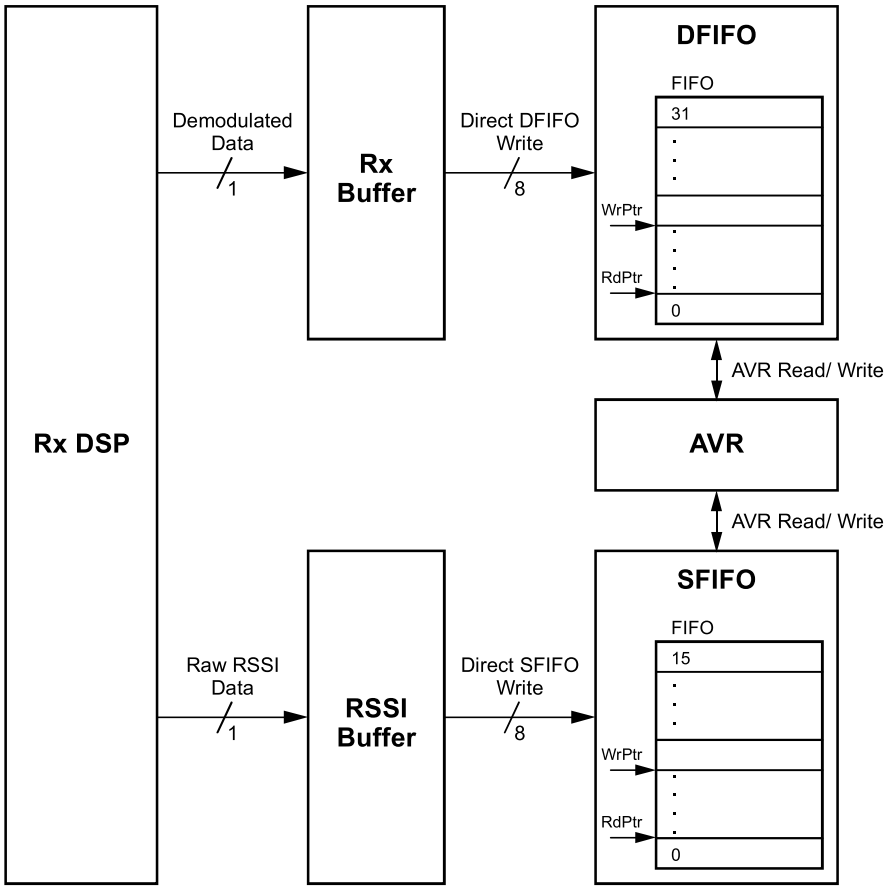
This bit is cleared when the Rx buffer is disabled (PRR2.PRXB=1) and enabled (PRR2.PRXB=0) again or when the Rx buffer is cleared (RXBC2.RXBCLR = 1).

3.5 Data and Support FIFOs

The Atmel® ATA5785 integrates two hardware buffers, a 32-byte deep data FIFO (DFIFO) and a 16-byte deep support FIFO (SFIFO). The DFIFO is used as a data buffer in buffered receive modes while the SFIFO serves as a buffer for the RSSI values.

Figure 3-20 shows a diagram of the DFIFO and SFIFO block-level interconnections.

Figure 3-20. DFIFO and SFIFO System View



3.5.1 Data FIFO

3.5.1.1 Data FIFO Overview

The data FIFO (DFIFO) is a 32-byte deep register bank which stores data with first-in-first-out functionality. The DFIFO has two read and two write interfaces:

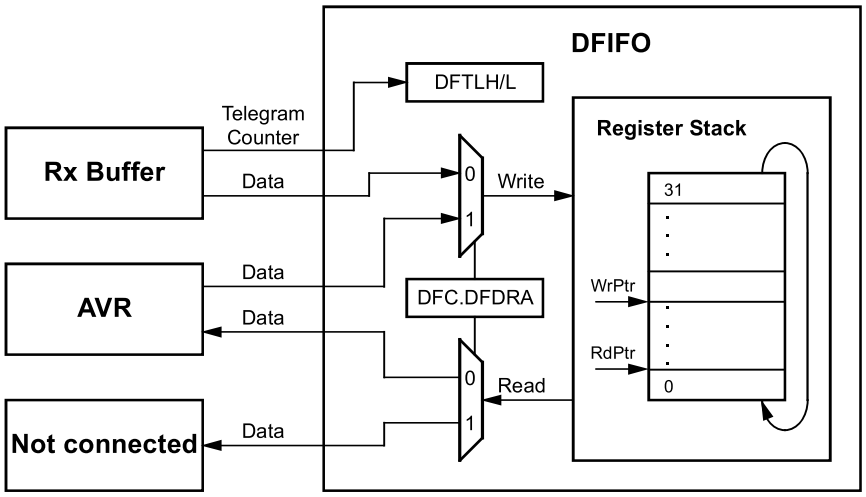
- AVR® bus read and write access
- Direct write access from the Rx buffer
- Direct read access not connected in Atmel® ATA5785

The DFIFO additionally implements a configurable fill level status bit with maskable interrupt signaling. Read attempts to an empty DFIFO (underflow) and write attempts a full DFIFO (overflow) are indicated by error flags (DFS.DFUFL and DFS.DFOFL) in the user interface. An error IRQ is triggered if masked in DFI.DFERIM.

A special feature of the DFIFO are the telegram length registers (DFTLH/DFTLL). These registers can be used in DFC.DFDRA=0 mode to read the number of received bits of the last telegram. This is helpful for unknown telegram lengths, especially if a telegram is not byte-aligned (number of bits is not divisible by 8). The telegram length is written to the registers by the Rx buffer once after an EOT has been received. The registers are not continuously updated during reception.

Figure 3-21 shows a block diagram of the DFIFO.

Figure 3-21. DFIFO Block Diagram



The access mode of the DFIFO is configured by the DFC.DFDRA register bit. If the bit is set to “0”, the Rx buffer can write to the DFIFO via the direct write interface while the AVR can only read from the DFIFO. This mode should be selected if the Atmel ATA5785 is receiving data. If the DFC.DFDRA bit is set to “1”, the AVR can only write to the DFIFO.

The DFIFO can trigger an interrupt if a certain fill level is reached. The interrupt can be switched on by setting the data FIFO fill level interrupt mask bit (DFI.DFFLIM) to “1”. The required fill level can be configured in the data FIFO fill level configuration register (DFC.DFFLC) to any value between 0 and 32. Once the configured fill level is reached (in DFDRA=0 mode after a byte was written to the DFIFO, in DFDRA=1 mode after a byte was read from the DFIFO), the data FIFO fill level reached status bit (DFS.DFFLRF) is set and the interrupt is triggered if enabled.

The current fill level can be read at any time in the data FIFO fill level register (DFL.DFFLS).

The AVR can access the data in the DFIFO by writing or reading the DFIFO data register (DFD). The user has to set the DFC.DFDRA correctly to allow either write or read access by the AVR.

The current read and write pointer addresses can be read from the DFRP and DFWP registers, respectively. The AVR also has write access to these registers, but manual pointer manipulations should be handled with care and are mainly implemented for the purpose of debugging.

The DFIFO is reset when activated in the corresponding power reduction register PRR2.PRDF (see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210) or by writing a “1” to the DFL.DFCLR bit.

A detailed description of the required register bits is found in Section 3.5.1.2 “Data FIFO Register Description” on page 164 if not stated otherwise.

3.5.1.2 Data FIFO Register Description

DFC – Data FIFO Configuration Register

Bit	7	6	5	4	3	2	1	0	
	DFDRA	-	DFFLC[5:0]						DFC
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: DFDRA – Data FIFO Direct Read Access Operational Mode

This bit defines the operational mode of the data FIFO.

DFDRA	Description
0	Direct write access: The FIFO is configured for data reception. The AVR can only read from the FIFO. Write access is allowed via the direct write access interface from the RX buffer.
1	Direct read access: The AVR can only write to the FIFO.

Bit 6: Reserved Bit

This bit is reserved and always returns zero when read.

Bits 5..0: **DFFLC** – Data FIFO Fill Level Configuration

The DFFLC bits define the required fill level for setting the DFS.DFFLRF status flag and triggering an interrupt. Precise conditions for this event are described at the DFS.DFFLRF bit. Valid DFFLC values are 0..32. Values exceeding this do not set the status flag or trigger an interrupt.

DFI – Data FIFO Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	-		-	-	-	-	DFERIM	DFFLIM	DFI
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 always return zero when read.

Bit 1: DFERIM – Data FIFO Error Interrupt Mask

Writing this bit to “1” enables the error interrupt. An interrupt is generated if a FIFO overflow or underflow occurs or if read and write pointer are set to invalid addresses.

Bit 0: DFFLIM – Data FIFO Fill Level Interrupt Mask

Writing this bit to “1” enables the fill level interrupt. Precise conditions for this event are described at the DFS.DFFLRF bit.

DFD – Data FIFO Data Register

Bit	7	6	5	4	3	2	1	0	
	DFD[7:0]								DFD
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: DFD – Data FIFO Data

The DFD is not a physical register. It merely defines the I/O address for the AVR® to write or read the FIFO data. A write access to this address transfers a data byte to the FIFO. A read access to this address fetches a data byte from the FIFO.

DFRP – Data FIFO Read Pointer Register

Bit	7	6	5	4	3	2	1	0	
	-	-	DFRP[5:0]						DFRP
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 always return zero when read.

Bit 5..0: DFRP – Data FIFO Read Pointer Register

This register gives read and write access to the data FIFO read pointer (RdPtr). The manipulation of the RdPtr requires thorough knowledge of the internal functionality of the FIFO. To avoid unpredictable behavior, the RdPtr should only be changed in IDLEMode. The RdPtr can have values between 0 and 63. The pointer cycles through these values in a round-robin manner. The difference between WrPtr and RdPtr (WrPtr – RdPtr) should never exceed 32, because otherwise the FIFO will be in an undefined state. This can be checked by reading the fill level in the DFL.DFFLS register. If the value is larger than 32, the RdPtr or WrPtr value is invalid.

Writing to this register overrides any other access.

DFWP – Data FIFO Write Pointer Register

Bit	7	6	5	4	3	2	1	0	
	-	-	DFWP[5:0]						DFWP
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 always return zero when read.

Bit 5..0: DFWP – Data FIFO Write Pointer Register

This register gives read and write access to the data FIFO write pointer (WrPtr). The manipulation of the WrPtr requires thorough knowledge of the internal functionality of the FIFO. To avoid unpredictable behavior, the WrPtr should only be changed in IDLEMode. The WrPtr can have values between 0 and 63. The pointer cycles through these values in a round-robin manner. The difference between WrPtr and RdPtr (WrPtr – RdPtr) should never exceed the FIFO depth, because otherwise the FIFO will be in an undefined state. This can be checked by reading the fill level in the DFL.DFFLS register. If the value is larger than 32, the RdPtr or WrPtr value is invalid.

Writing to this register overrides any other access.

DFL – Data FIFO Fill Level Register

Bit	7	6	5	4	3	2	1	0	
	DFCLR	-	DFFLS[5:0]						DFL
Read/Write	W	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: DFCLR – Data FIFO Clear

Writing a “1” to this bit location clears the data FIFO. RdPtr, WrPtr and fill level are reset to zero. The data FIFO status (DFS) and telegram length (DFTLH/DFTLL) registers are cleared. The data FIFO configuration registers (DFC and DFI) keep their settings. The data content of the FIFO is also retained. It can be recovered by RdPtr and WrPtr manipulations.

Reading this bit always returns zero.

Bit 6: Reserved Bit

This bit is reserved and always returns zero when read.

Bits 5..0: DFFLS – Data FIFO Fill Level Status

The current fill level of the data FIFO is stored in DFFLS. These bits are read-only. A DFFLS value that is larger than 32 indicates incorrect programming of the read (DFRP) or write (DFWP) pointer.

The fill level is updated one system clock cycle after a read/write access or pointer manipulation.

DFTLH – Data FIFO Telegram Length High Byte

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

Bits 7..4: Reserved Bits

These bits are reserved and always return zero when read.

Bits 3..0: DFTLH – Data FIFO Telegram Length High Byte

DFTLH stores the upper four bits of the 12-bit DFTL word. DFTL shows the number of bits in the most recent telegram received continuously by the data FIFO in DFC.DFDRA=0 mode. If DFTLH and DFTLL are both “0x00”, a full telegram has not been received since the last reset. DFTLH is a read-only register.

DFTLL – Data FIFO Telegram Length Low Byte

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

Bits 7..0: DFTLL – Data FIFO Telegram Length Low Byte

DFTLL stores the lower eight bits of the 12-bit DFTL word. DFTL shows the number of bits in the latest telegram which has been received continuously by the data FIFO in DFC.DFDRA=0 mode. If DFTLH and DFTLL are both “0x00”, a full telegram has not been received since the last reset. DFTLL is a read-only register.

DFS – Data FIFO Status Register

Bit	7	6	5	4	3	2	1	0	
	-		-	-	-	DFOFL	DFUFL	DFFLRF	DFS
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 always return zero when read.

Bit 2: DFOFL – Data FIFO Overflow Flag

This error flag bit is set if there is a write access to a full FIFO or if the fill level is higher than the FIFO depth, which can happen due to faulty pointer manipulations in the DFRP or DFWP registers. The DFOFL flag can generate an interrupt when masked in DFI.DFERIM. The DFOFL flag is automatically cleared when the interrupt is executed or by writing a “1” to its bit location.

Bit 1: DFUFL – Data FIFO Underflow Flag

This error flag bit is set if there is a read access to an empty FIFO. The DFUFL flag can generate an interrupt when masked in DFI.DFERIM. The DFUFL flag is automatically cleared when the interrupt is executed or by writing a “1” to its bit location.

Bit 0: DFFLRF – Data FIFO Fill Level Reached Status Flag

If DFC.DFDRA=0, this flag is set if the current fill level of the data FIFO matches the fill level configuration (DFC.DFFLC) after the reception of a byte.

The DFFLRF flag generates an interrupt when masked in DFI.DFFLIM. The DFFLRF flag is automatically cleared when the interrupt is executed or by writing a “1” to its bit location.

3.5.2 Support FIFO

3.5.2.1 Support FIFO Overview

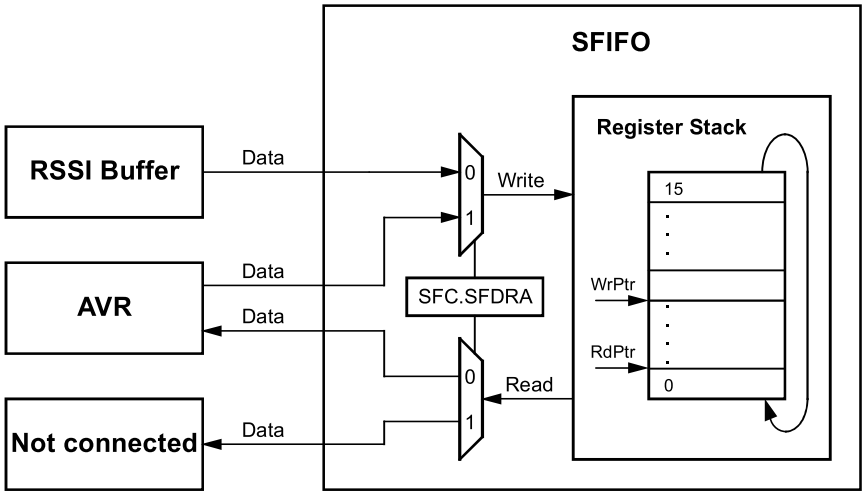
The support FIFO (SFIFO) is a 16-byte deep register bank which stores data with first-in-first-out functionality. The SFIFO has two read and two write interfaces:

- AVR® bus read and write access
- Direct write access from the RSSI buffer
- Direct read access is not connected in Atmel® ATA5785

Furthermore, the SFIFO implements a configurable fill level status bit with maskable interrupt signaling. Read attempts to an empty SFIFO (underflow) and write attempts to a full SFIFO (overflow) are indicated by error flags (SFS.SFUFL and SFS.SFOFL) in the user interface. An error IRQ is triggered if masked in SFI.SFERIM.

Figure 3-22 shows a block diagram of the SFIFO.

Figure 3-22. SFIFO Block Diagram



The access mode of the SFIFO is configured by the SFC.SFDRA register bit. If the bit is set to “0”, the RSSI buffer can write to the SFIFO via the direct write interface while the AVR can only read from the SFIFO. This mode should be selected if the Atmel ATA5785 is receiving data. If the SFC.SFDRA bit is set to “1”, the AVR can only write to the SFIFO.

The SFIFO can trigger an interrupt if a certain fill level is reached. The interrupt can be switched on by setting the support FIFO fill level interrupt mask bit (SFI.SFFLIM) to “1”. The required fill level can be configured in the support FIFO fill level configuration register (SFC.SFFLC) to any value between 0 and 16. Once the configured fill level is reached (in SFDRA=0 mode after a byte was written to the SFIFO, in SFDRA=1 mode after a byte was read from the SFIFO), the support FIFO fill level reached status bit (SFS.SFFLRF) is set and the interrupt is triggered if enabled.

The current fill level can be read at any time in the support FIFO fill level register (SFL.SFFLS).

The AVR can access the data in the SFIFO by writing or reading the SFIFO data register (SFD). The user has to set the SFC.SFDRA correctly to allow either a write or a read access by the AVR.

The current read and write pointer addresses can be read from the SFRP and SFWP registers, respectively. The AVR also has write access to these registers, but manual pointer manipulations should be handled with care and are mainly implemented for debugging purposes.

The SFIFO is reset when activated in the corresponding power reduction register PRR2.PRSF (see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210) or by writing a “1” to the SFL.SFCLR bit.

If not stated otherwise, a detailed description of the required register bits is found in Section 3.5.2.2 “Support FIFO Register Description” on page 168.

3.5.2.2 Support FIFO Register Description

SFC – Support FIFO Configuration Register

Bit	7	6	5	4	3	2	1	0	
	SFDRA	-	-	SFFLC[4:0]					SFC
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: SFDRA – Support FIFO Direct Read Access Operational Mode

This bit defines the operational mode of the support FIFO.

SFDRA	Description
0	Direct write access: The FIFO is configured for data reception and serves as RSSI value storage. The AVR can only read from the FIFO. Write access is allowed via the direct write access interface from the RSSI buffer.
1	Direct read access: The AVR can only write to the FIFO.

Bits 6..5: Reserved Bits

These bits are reserved and always return zero when read.

Bits 4..0: **SFFLC** – Support FIFO Fill Level Configuration

The SFFLC bits define the required fill level for setting the SFS.SFFLRF status flag and triggering an interrupt. Precise conditions for this event are described at the SFS.SFFLRF bit. Valid SFFLC values are 0..16. Values exceeding this do not set the status flag or trigger an interrupt.

SFI – Support FIFO Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	-		-	-	-	-	SFERIM	SFFLIM	SFI
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 always return zero when read.

Bit 1: SFERIM – Support FIFO Error Interrupt Mask

Writing this bit to “1” enables the error interrupt. An interrupt is generated if a FIFO overflow or underflow occurs or if read and write pointer are set to invalid addresses.

Bit 0: SFFLIM – Support FIFO Fill Level Interrupt Mask

Writing this bit to “1” enables the fill level interrupt. Precise conditions for this event are described at the SFS.SFFLRF bit.

SFD – Support FIFO Data Register

Bit	7	6	5	4	3	2	1	0	
	SFD[7:0]								SFD
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: SFD – Support FIFO Data

The SFD is not a physical register. It merely defines the I/O address for the AVR® to write or read the FIFO data. A write access to this address transfers a data byte to the FIFO. A read access to this address fetches a data byte from the FIFO.

SFRP – Support FIFO Read Pointer Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	SFRP[4:0]					SFRP
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 always return zero when read.

Bit 4..0: SFRP – Support FIFO Read Pointer Register

This register gives read and write access to the support FIFO read pointer (RdPtr). The manipulation of the RdPtr requires thorough knowledge of the internal functionality of the FIFO. To avoid unpredictable behavior, the RdPtr should only be changed in IDLEMode. The RdPtr can have values between 0 and 31. The pointer cycles through these values in a round-robin manner. The difference between WrPtr and RdPtr (WrPtr – RdPtr) should never exceed 16, because otherwise the FIFO will be in an undefined state. This can be checked by reading the fill level in the SFL.SFFLS register. If the value is larger than 16, the RdPtr or WrPtr value is invalid.

Writing to this register overrides any other access.

SFWP – Support FIFO Write Pointer Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	SFWP[4:0]					SFWP
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 always return zero when read.

Bit 4..0: SFWP – Support FIFO Write Pointer Register

This register gives read and write access to the support FIFO write pointer (WrPtr). The manipulation of the WrPtr requires thorough knowledge of the internal functionality of the FIFO. To avoid unpredictable behavior, the WrPtr should only be changed in IDLEMode. The WrPtr can have values between 0 and 31. The pointer cycles through these values in a round-robin manner. The difference between WrPtr and RdPtr (WrPtr – RdPtr) should never exceed 16, because otherwise the FIFO will be in an undefined state. This can be checked by reading the fill level in the SFL.SFFLS register. If the value is larger than 16, the RdPtr or WrPtr value is invalid.

Writing to this register overrides any other access.

SFL – Support FIFO Fill Level Register

Bit	7	6	5	4	3	2	1	0	
	SFCLR	-	-	SFFLS[4:0]					SFL
Read/Write	W	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: SFCLR – Support FIFO Clear

Writing a “1” to this bit location clears the support FIFO. RdPtr, WrPtr, and fill level are reset to zero. The support FIFO status register is cleared. The support FIFO configuration registers (SFC and SFI) keep their settings. The data content of the FIFO is also retained. It can be recovered by RdPtr and WrPtr manipulations.

Reading this bit always returns zero.

Bits 6..5: Reserved Bits

These bits are reserved and always return zero when read.

Bits 4..0: SFFLS – Support FIFO Fill Level Status

The current fill level of the support FIFO is stored in SFFLS. These bits are read-only. An SFFLS value that is larger than 16 indicates incorrect programming of the read (SFRP) or write (SFWP) pointer.

The fill level is updated one system clock cycle after a read/write access or pointer manipulation.

SFS – Support FIFO Status Register

Bit	7	6	5	4	3	2	1	0	
	-		-	-	-	SFOFL	SFUFL	SFFLRF	SFS
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 always return zero when read.

Bit 2: SFOFL – Support FIFO Overflow Flag

This error flag bit is set if there is a write access to a full FIFO or if the fill level exceeds the FIFO depth, which can happen due to faulty pointer manipulations in the SFRP or SFWP registers. The SFOFL flag can generate an interrupt when masked in SFI.SFERIM. The SFOFL flag is automatically cleared when the interrupt is executed or by writing a “1” to its bit location.

Bit 1: SFUFL – Support FIFO Underflow Flag

This error flag bit is set if there is a read access to an empty FIFO. The SFUFL flag can generate an interrupt when masked in SFI.SFERIM. The SFUFL flag is automatically cleared when the interrupt is executed or by writing a “1” to its bit location.

Bit 0: SFFLRF – Support FIFO Fill Level Reached Status Flag

If SFC.SFDRA=0, this flag is set if the current fill level of the support FIFO matches the fill level configuration (SFC.SFFLC) after the reception of a byte.

The SFFLRF flag generates an interrupt when masked in SFI.SFFLIM. The SFFLRF flag is automatically cleared when the interrupt is executed or by writing a “1” to its bit location.

3.6 RF Front-End Register Description

The registers described in this section are located in the RF front end and are accessible by an on-chip bridge interface. They are mapped to the extended I/O memory space. A read access to a front-end register generates three wait states to complete.

In standard applications the RF front-end registers are controlled exclusively by firmware and the sequencer state machine. The registers are changed according to the application-specific settings. Manual access to these registers by the user is not recommended.

The description in this section is intended to serve as an overview. The complete register map is shown in Section 7.2 “Register Map” on page 317.

The RF front-end registers are all reset to “0” with RST_AVCC_N=”0”. This occurs automatically when the RF front end is completely disabled by setting AVEN=”0”. Reading one of the registers while the front end is disabled always returns “0”.

FESR – RF Front-End Status Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	PLCK	XRDY	-	-	FESR
Read/Write	R	R	R	R	R	R	R	R	
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: PLCK – PLL Locked

This status bit is set to “1” if the PLL is locked within the time window specified by FECR.PLCKG. It shows that a stable frequency is available from the fractional-N PLL.

Bit 2: XRDY – XTO Ready

This status bit is set to “1” if the XTO amplitude is settled and provides a reliable clock output. If this bit is set to “1”, the XTO frequency is available to the AVR® and the fractional-N PLL.

Bits 1..0: Reserved Bits

These bits are reserved and always return zero when read.

FEEN1– RF Front-End Enable Register 1

Bit	7	6	5	4	3	2	1	0	
	–	PLSP1	ADCLK	ADEN	LNAEN	XTOEN	PLCAL	PLEN	FEEN1
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: Reserved Bit

This bit is reserved and reads as zero.

Bit 6: PLSP1 – PLL Speed-Up

If this bit is written to “1”, the fractional-N PLL speed-up mode is activated. This bit should be set to “0” after the FESR.PLCK is locked.

Bit 5: ADCLK – ADC Clock Enable

If this bit is written to “1”, the ADC clock is activated. It has to be set >50µs after ADEN is set because the internal biasing of the ADC must settle before the clock is activated.

Bit 4: ADEN – Analog Digital Converter Enable

If this bit is written to “1”, the DC biasing of the ADC is activated.

Bit 3: LNAEN – Low Noise Amplifier Enable

If this bit is written to “1”, the Low-Band is enabled.

Bit 2: XTOEN – Crystal Oscillator Enable

If this bit is written to “1”, the crystal oscillator is started.

This bit enables the XTO, which operates in IDLEMode(XTO) and RXMode. The XTO frequency is available as soon as FESR.XRDY is set by the XTO.

Bit 1: PLCAL – PLL Calibration Mode

If this bit is written to “1”, the PLL is set to calibration mode. In this mode the loop filter is clamped to nominal control voltage AVCC/2 and the charge pump is deactivated.

During the calibration mode, the frequency is measured with a counter and the correct FEVCT.FEVCT[3:0] frequency setting for the VCO is determined by the sequencer state machine.

Bit 0: PLEN – PLL Enable

If this bit is written to “1”, the fractional-N PLL is activated.

FEEN2 – RF Front-End Enable Register 2

Bit	7	6	5	4	3	2	1	0	
	–	–	0	PLPEN	0	–	0	0	FEEN2
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..6: Reserved Bits

These bits are reserved and read as zero.

Bit 5: This bit must always be set to “0”

Bit 4: PLPEN – PLL Post Enable

This bit shall be set 10µs after the PLL was enabled in FEEN1.PLEN.

Bit 3: This bit must always be set to “0”

Bit 2: Reserved Bit

This bit is reserved and reads as zero.

Bit 1: This bit must always be set to “0”

Bit 0: This bit must always be set to “0”

FELNA – RF Front-End LNA Bias Register

Bit	7	6	5	4	3	2	1	0	
	LBL[3:0]				0				FELNA
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..4: LBL[3:0] – LNA Bias Low-Band

LBL[3:0] sets the current of the Low-Band LNA.

Bits 3..0: These bits must always be set to “0”

FEVCT – RF Front-End VCO Tuning Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	FEVCT[3:0]				FEVCT
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..0: FEVCT[3:0] – Front-End VCO Tuning Register

FEVCT[3:0] is used for the coarse tuning of the VCO. These bits are set by means of a calibration state machine, which is part of the sequencer state machine. For further information, see Section 3.7.9 “VCO Tuning State Machine” on page 182.

FEBT – RF Front-End RC Tuning 2 Bit Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	RTN2[1:0]		CTN2[1:0]		FEBT
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..2: RTN2[1:0] – Resistor Tuning 2 Bit

RTN2[1:0] is used for 2-bit resistor calibration.

Bits 1..0: CTN2[1:0] – Capacitor Tuning 2 Bit

CTN2[1:0] is used for the 2-bit capacitor calibration.

FEMS – RF Front-End Main and Swallow Control Register

Bit	7	6	5	4	3	2	1	0	
	PLLM[3:0]				PLLS[3:0]				FEMS
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..4: PLLM[3:0] – Fractional-N PLL M Value

PLLM[3:0] sets the M counter divider ratio used for the fractional-N PLL.

Bits 3..0: PLLS[3:0] – Fractional-N PLL S Value

PLLS[3:0] sets the S counter divider ratio used for the fractional-N PLL.

The values must be copied to the PLL register before using the RF front end. Firmware does this by default.

FETN4 – RF Front-End RC Tuning 4-Bit Register

Bit	7	6	5	4	3	2	1	0	
	RTN4[3:0]				CTN4[3:0]				FETN4
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..4: RTN4[3:0] – Resistor Tuning 4-Bit

RTN4[3:0] is used for the 4-bit resistor calibration.

Bits 3..0: CTN4[3:0] – Capacitor Tuning 4-Bit

CTN4[3:0] is used for the 4-bit capacitor calibration.

FECR – RF Front-End Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	ANPS	PLCKG	ADHS	0	S4N3	1	FECR
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.

Bit 5: ANPS – ASK Not DPSK Switch

This bit should always be set to “1”.

Bit 4: PLCKG – PLL Lock Detect Gate

Setting this bit to “1” opens a time window during which the fractional-N PLL detects frequency locking. If the PLL frequency is close to the wanted frequency, the status bit FESR.PLCK is set to “1”.

Bit 3: ADHS – ADC High Sample Rate

This bit should be set to “1” if the ADC is used with a high sample rate.

Bit 2: This bit must always be set to “0”

Bit 1: S4N3 – Select 433 MHz Not 315 MHz Band

If this bit is written to “1”, the 418MHz to 477MHz band is selected. Otherwise the 310MHz to 318MHz band is selected.

Bit 0: This bit must always be set to “1”

FEVCO – RF Front-End VCO and PLL Control

Bit	7	6	5	4	3	2	1	0	
	VCOB[3:0]				CPCC[3:0]				FEVCO
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..4: VCOB[3:0] – VCO Bias

The VCOB[3:0] setting controls the amplitude of the VCO. In firmware this frequency-dependent value is stored in the SRAM configuration for each service. In addition, it is corrected with a factory calibration value.

Bits 3..0: CPCC[3:0] – Charge Pump Current Control

The CPCC[3:0] setting controls the charge pump current in the fractional-N PLL and, as a result, the loop bandwidth of the fractional-N PLL. In firmware this frequency-dependent value is stored for each service. In addition, it is corrected with a factory calibration value.

FEBIA – RF Front-End Amplifier Bias

Bit	7	6	5	4	3	2	1	0	
	IFAEN	–	–	–	–	–	–	–	FEBIA
Read/Write	R/W	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: IFAEN – IF Amplifier Enable

If this bit is set to “1”, the additional IF amplifier is added to the Rx chain. This enhances the sensitivity by approximately 2-3dB. The in-band compression is degraded by 10dB while the out-band compression remains nearly unchanged.

Bits 6..0: Reserved Bits

These bits are reserved and read as zero.

3.7 Sequencer State Machine

3.7.1 Overview

The sequencer state machine implements major control parts to set up and run the receive operations. It makes the start-up and shutdown process more robust and predictable because it is not susceptible to timing deviations as if controlled by the AVR® core with a varying work load.

To set up a reception chain the software initially needs to load the configuration, such as the filter settings and frequency parameters to the corresponding hardware registers. Afterwards it hands over the control to the SSM that enables all required blocks with an appropriate timing, taking into account the settling times of filters and analog components.

The configuration of the SSM offers some flexibility such as deciding whether VCO tuning is needed or not. The start-up flows are broken up into smaller elements, allowing parts of the sequencer state machine functionality to be used only. In addition, the start-up flows augment software-controlled sequences.

The integrated dedicated VCO tuning timer frees other on-chip resources for application use.

Figure 3-23 shows a diagram of useful sub-state-machine sequences. The system needs to be in the IDLEMode(XTO) with the RF front end powered up and the system running on clk_{XTO}.

Several state machines are needed to transition between the three shown states of IDLEMode(XTO), and RXMode. The flow may be interrupted between every sub-state machine to intersperse some user software code. Other orders of sub-state machines than those shown in the diagram are prohibited.

Figure 3-23. SSM Sub-State-Machine Sequence Diagram

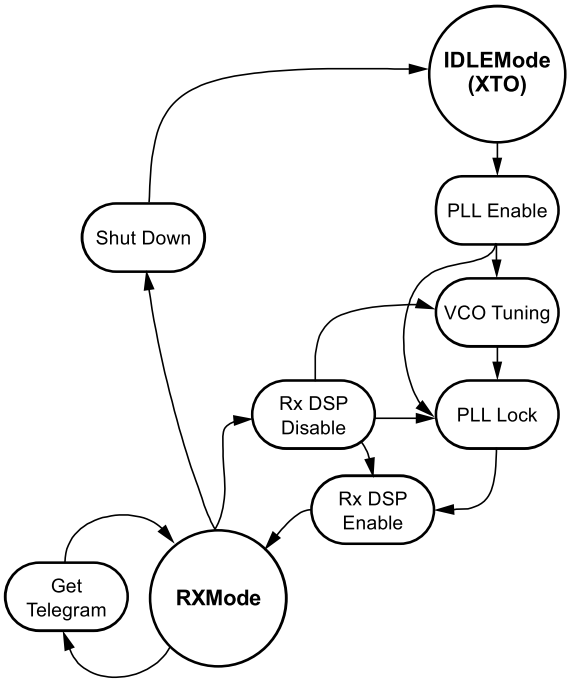
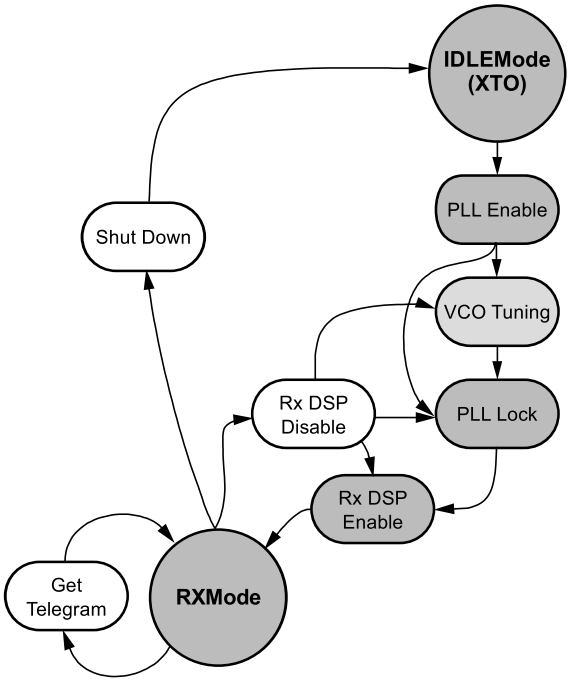


Figure 3-24 shows the standard sequence for initializing the system for RF reception. The optional VCO tuning sub-state machine is marked in light gray.

Figure 3-24. SSM Start Rx Sequence



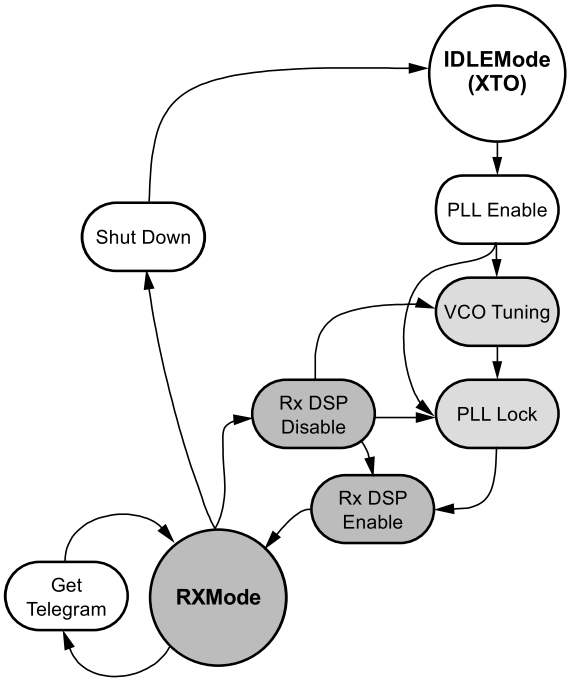
When the RF channel is switched during reception, it needs to be ensured that the PLL has settled to the new frequency.

A new PLL lock sequence is thus required if one of the following conditions is true:

- A VCO tuning occurred
- Modification of the FEMS.PLLM or FEMS.PLLS values
- Modification of the MSB in the FFREQ1H or FFREQ2H register
- Modification of the PLL frequency by more than 1MHz

The Rx to Rx switching sequence is realized as shown in Figure 3-25. Once the Rx DSP is stopped, the PLL lock sub-state machine waits for the lock signal of the PLL and the Rx DSP is enabled again while running on the new frequency.

Figure 3-25. SSM Rx to Rx Switching Sequence



The SSM must be activated in the corresponding power reduction register PRR2.PRSSM (see Section 3.8.5 “Sleep Modes and Active Power Reduction” on page 210).

3.7.2 Master State Machine

The master state machine is the main controlling block of the SSM, which starts the different sub-state machines in the order configured in the MSMCRx registers. It waits for the completion of a sub-state machine before starting the next one.

The configuration of the master state machine is done via the MSMCRx registers where a sequence of up to eight sub-state machines can be programmed. The sequence is then started by writing a “1” to the run bit (SSMR) in the run register (SSMRR). The master state machine iterates through the programmed sub-state machines starting with the entry in MSMSM0. When the first sub-state machine is completed, the master state machine continues with the next one in MSMSM1 and so on. If it has reached either the end of the eighth sub-state machine or starts an entry with an empty MSMSMx, the master state machine is done and stops. The SSMIF flag in SSMIFR is set to “1”. If the interrupt mask in SSMIMR is set and if global interrupts are enabled, an interrupt is requested by the sequencer state machine. Firmware should check SSMERR in SSMSR if the state machine has completed successfully to identify any errors that may have occurred during operation.

If, for any reason the sequencer state machine needs to be stopped before it has completed its operation, this can be done by writing a “1” to the SSMST stop bit in the SSMRR run register. The master state machine as well as the currently running sub-state machine are stopped immediately and no interrupt is issued. As the state of the receiver is unpredictable after a user stop, Atmel recommends restarting it with only the shut down state machine programmed to MSMSM0, which brings the system back into a known state.

All configuration settings such as filter bandwidth, enabled paths, or other options need to be written before the master state machine is started. In addition, none of these settings should be changed while the state machine is running.

3.7.3 PLL Enable State Machine

The PLL enable state machine starts the PLL for Rx.

As a plausibility check the XRDY flag in the FESR is expected to be set to “1” to ensure that the XTO is up and running. If this check is not successful, the sequencer state machine ends with the error flag set.

Usually the PLL enable state machine should be the first entry in a setup sequence to start Rx. It can be followed by the VCO tuning or immediately followed by the PLL lock state machine if no tuning is required.

3.7.4 PLL Lock State Machine

The PLL lock state machine activates the PLL speed-up mode for faster locking and then waits for the PLL lock signal. The necessary sequence to read the lock flag is passed a maximum of three times. If no lock is reached by that point, the state machine ends with the error flag set.

Depending on the settings gathered in Table 3-34, this state machine enables the ADC and the IF amplifier.

After having successfully passed the PLL lock state machine, the RF frequency should be up and running and the design can be put into RxMode by starting the Rx DSP enable state machine.

Table 3-34. Settings Influencing the PLL Lock State Machine

Setting	Description
SSMRCR.SSMIFA	Switches IFA on only if configured.
SSMFBR.SSMPLDT	Adjusts the settling wait time for internal filters.

3.7.5 Rx DSP Enable State Machine

The Rx DSP enable state machine enables the modules that are needed for reception only. It expects a successful run of PLL lock as a prerequisite. The state machine switches the Rx DSP on including the channel filter and demodulator paths A/B as configured. The IF amplifier circuitry is set up if configured in the SSMRCR register. The Rx DSP enable state machine turns on the ADCLK.

In addition, the SOT and the EOT condition flags are copied from the SOT1A/B and the EOTC1A/B registers to the demodulator registers SOTCA/B and EOTCA/B.

After having successfully passed the Rx DSP enable state machine, the design is set up for reception and the flow control can be passed to the get telegram state machine.

Table 3-35. Settings Influencing the Rx DSP Enable State Machine

Setting	Description
SSMRCR.SSMIFA	Switches the IF amplifier on only if configured.
SSMRCR.SSMPA/B	Enables the demodulator path A/B in Rx DSP.
SSMCR.SETRPA/B	Enables the transparent output.
SSMFBR.SSMFID	Adjusts the settling wait time for internal filters.
SSMFBR.SSMDFDT	
SOTC1A/B	Copies these settings to the demodulator registers as initial conditions.
EOTC1A/B	

3.7.6 Rx DSP Disable State Machine

The Rx DSP disable state machine is complementary to the Rx DSP enable state machine and switches all the blocks off that have been enabled. It disables the Rx DSP with the demodulator as well as the analog components such as the IF amplifier, the ADC clock and, if configured, the transparent outputs TRPA/B (see Table 3-36).

After having passed the Rx DSP disable state machine, the system is now ready to run either the VCO tuning state machine or run the PLL lock state machine directly (see Figure 3-25 on page 177).

Table 3-36. Settings Influencing the Rx DSP Disable State Machine

Setting	Description
SSMCR.SETRPA/B	Disables the transparent output.

3.7.7 Get Telegram State Machine

The get telegram state machine handles the reception of a telegram after the Rx path is enabled. It is the most complex state machine. Several options can modify the flow to support a wide range of different protocols.

After the first initialization, the state machine can be spawned into two paths, one for path A and one for path B. Both paths can be active at the same time waiting for the start of a transmission. Depending on the SSMRCR.SSMPVS configuration, one path is halted either after a wake check OK or after a start of telegram event on the other path. Only one telegram can be received at a time. After the end of a telegram, both paths can be restarted, with both paths again waiting for a wake event.

Figure 3-26 on page 181 shows the state diagram of the get telegram state machine. For better readability, the states for path B are hidden because they are equivalent to path A. Each part is associated with the corresponding demodulator path. The two paths of the state machine are thus not necessarily in the same state. For example, path A can be waiting for a wake check OK event in the get WCOA state whereas path B may have already gotten a wake check OK and is then waiting for the start of telegram event in the get SOTB state.

Telegram Phases/Demodulator Condition Flags

The get telegram state machine supports the three phases of telegram reception.

During the first phase the state machine is waiting for a wake check OK (WCO) condition. For this phase the demodulator uses the settings from SOTC1A/B and EOTC1A/B. These are copied into SOTCA/B and EOTCA/B in the Rx DSP enable state machine and in the setup WCO state to obtain the settings restored after any restart.

During the second phase the state machine is waiting for a start of telegram (SOT). In this phase the demodulator uses the settings from SOTC2A/B and EOTC2A/B. These are copied into SOTCA/B and EOTCA/B in the setup SOT state.

During the third phase the state machine is waiting for the EOT. In this phase the demodulator uses the settings from EOTC3A/B. These are copied into EOTCA/B during the disable state.

Loop Control

For each path the get telegram state machine has four configuration registers which control the loop and restart behavior. First there is the GTCR.DARA/B setting which enables or disables the automatic demodulator restart. If this bit is set to “1”, the demodulator gets restarted automatically when an EOT condition occurs while the system is waiting for a wake check OK or for a start of telegram.

Then there is the intermittent wake-up GTCR.IWUPA/B setting which is intended for automatic polling cycles. If it is set to “1”, the state machine waits for a wake check OK as long as this is defined by the wake check OK time-out WCOTOA/B. If an EOT condition occurs before the time-out, the demodulator is restarted and is again waiting for the wake check OK condition.

The GTCR.GAPA/B setting is for supporting protocols that have a pause after the wake-up pattern. If enabled, the state machine waits for a start of telegram even if an EOT condition has occurred after detecting a wake check OK, for example, due to a missing carrier or Manchester error. The maximum wait time and thus the maximum length of the gap can be defined by the start of telegram time-out SOTTOA/B.

The setting of the Rx telegram end handling GTCR.RXTEHA/B decides whether the state machine should stop or restart after the telegram has been received successfully.

ID Check Handling

The get telegram state machine can process ID check results. The ID check feature needs software support to write the results of an ID check into the SSMFCR register. Due to the implementation nature of the ID check hardware (see Section 3.4.3.5 “ID Check” on page 136) which itself requires software support, this mechanism avoids overlapping results from different telegrams.

If the ID check handling is enabled by the SSMRCR.SSMIDSE register and the get telegram state machine is in the get EOT state, a “1” in the SSMIDSF bit ends this state and transitions to the next state. The telegram currently being received is incomplete and can be discarded.

When the ID check is enabled, the get telegram state machine is in the wait ID scan finish state waiting for either an SSMIDSF or an SSMIDSO bit to be set.

Additional Controls

The get telegram state machine controls the RSSI buffer if it is enabled in the PRR2.PRRS register.
If the get telegram state machine is started with no path enabled in the SSMRCR, it stops immediately and sets the error flag in SSMSR.

Figure 3-26. Get Telegram State Diagram

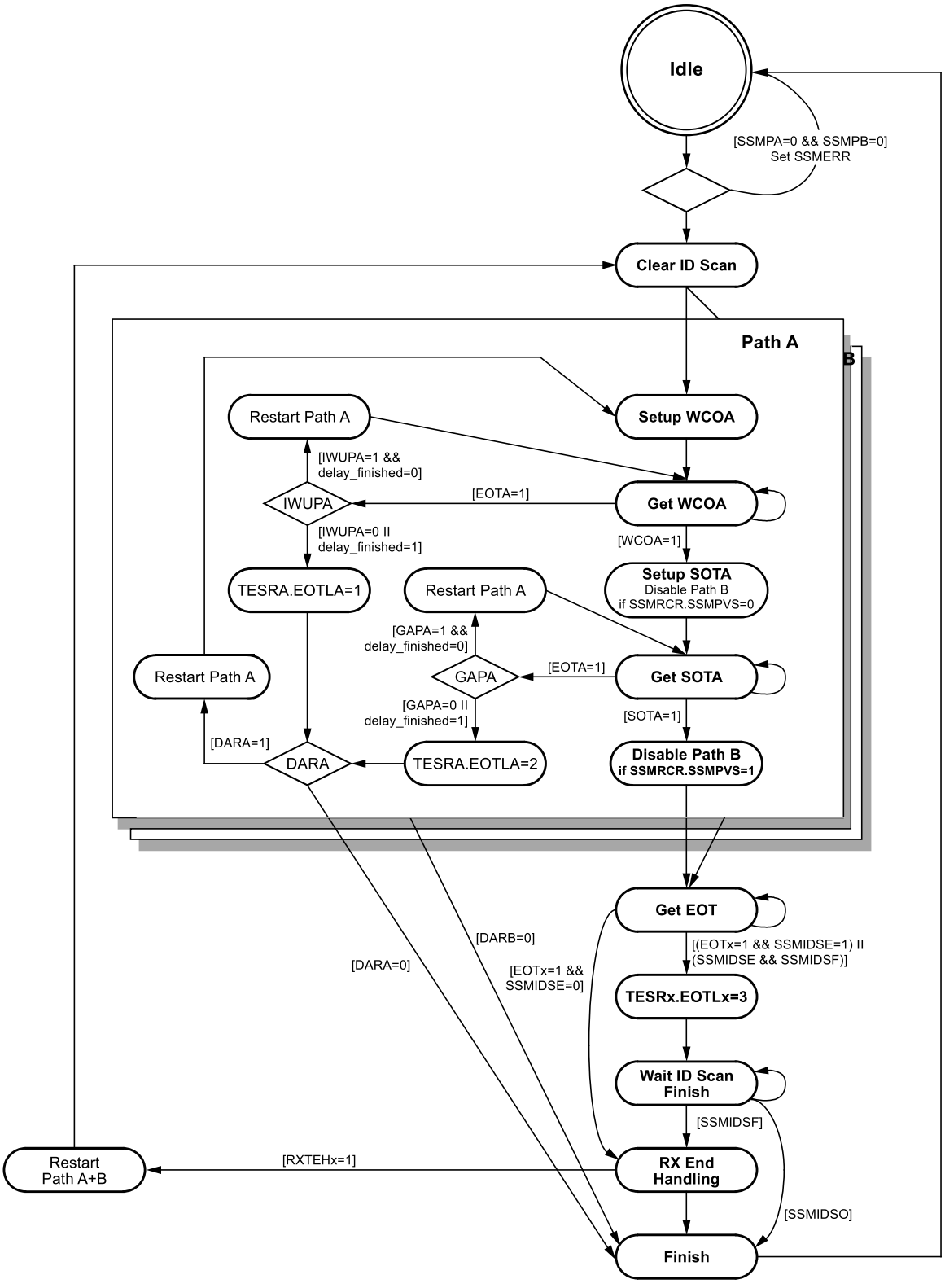


Table 3-37. Settings Influencing the Get Telegram State Machine

Setting	Description
SSMRCR.SSMPA/B	Enable path A/B of the sub-state machine.
SSMRCR.SSMIFA	Enable IF amplifier
SSMCR.SETRPA/B	Enable/disable transparent outputs.
SSMRCR.SSMTMOE	
SSMRCR.SSMIDSE	Check ID check results if enabled.
SSMFBR.SSMFID	Adjusts the settling wait time for internal filters
SSMFBR.SSMDFDT	
SSMRCR.SSMPVS	Flow control of telegram reception: Path valid after SOT Gap mode Intermittent WUP Automatic restart Rx end handling → see Figure 3-26 on page 181
GTCR.GAPMA/B	
GTCR.IWUPA/B	
GTCR.DARA/B	
GTCR.RXTEHA/B	
SOTC1A/B	Copy these settings to demodulator registers before waiting for wake check OK.
EOTC1A/B	
SOTC2A/B	Copy these settings to demodulator registers before waiting for start of telegram.
EOTC2A/B	
EOTC3A/B	Copy these settings to demodulator registers before waiting for end of telegram.
WCOTOA/B	Wake check OK time-out.
SOTTOA/B	Start of telegram time-out.

3.7.8 Shut Down State Machine

The shut down state machine switches off all blocks that might have been previously enabled by another state machine. This includes the DSP blocks as the demodulator, channel filter, as well as the analog blocks in the RF front end such as LNA, IF amplifier, PLL, and ADC. Only the XTO is kept on to allow the system to go back to IDLEMode(XTO).

If configured in SETRPA/B, the transparent outputs get disabled, too.

It is strongly recommended to start the shut down state machine after a software-induced state machine stop via the SSMRR.SSMST signal to revert to most of the changes made previously by the sequencer state machine. Neither the changes to setting registers as EOTC and SOTC nor changes of tuning registers as FEVCT are undone.

Table 3-38. Settings Influencing the Shut Down State Machine

Setting	Description
SSMCR.SETRPA/B	Disable the transparent output.

3.7.9 VCO Tuning State Machine

The VCO tuning state machine tunes the VCO frequency using the XTO as reference. It changes the FEVCT value in a successive approximation algorithm until the LSB has been adjusted. The best value during the tuning process is programmed to FEVCT. This value may differ from the most recent value.

There are no user settings which influence the behavior of this state machine.

3.7.10 Interaction with IO Registers

Table 3-27 on page 184 shows all IO registers that are read or modified by the sequencer state machine. While the state machine is running, it is forbidden to write to any of these registers. Failure to observe this may cause unpredictable or faulty behavior of the state machine.

Reading is always allowed, but in the case of RF front-end registers, doing so can lead to timing deviations (delay the register access of the state machine).

The table also marks the different types of access performed by the corresponding sub-state machine. *Set* means the bit is set to “1”. *Clear* means the bit is set to “0”. *Set & clear* means both settings are done throughout the sub-state machine sequence, for example, set at the beginning and cleared later on or at the end. *Value* means this register is accessed as a whole, writing a value into it. No single bit operations are done on these registers. *Read* means the state machine is only reading the indicated bit.

Figure 3-27. Control Register Access from State Machines

Setting	pll_en	pll_lock	rx_dsp_en	rx_dsp_dis	get_tel	shut_down	vcot
1 FEEN1.PLEN	set					clear	
2 FEEN1.LNAEN	set					clear	
3 Reserved							
4 FEEN1.PLCAL						clear	set&clear
5 FEEN1.PLSP1		set&clear				clear	
6 FEEN1.ADEN		set				clear	
7 FEEN1.ADCLK			set	clear		clear	
8 FEEN1.XTOEN						set	
9 Reserved							
10 Reserved							
11 Reserved							
12 FEEN2.PLPEN	set					clear	
13 Reserved							
14 FECR.PLCKG		set&clear				clear	
15 Reserved							
16 FEBIA.IFAEN		set	set&clear	clear	set&clear	clear	
17 FSEN.SDPU	set					clear	
18 FSEN.SDEN	set					clear	
19 Reserved							
20 Reserved							
21 Reserved							
22 Reserved							
23 SUPCR.PVEN						clear	
24 RDCR.RDPU			set	clear		clear	
25 RDCR.RDEN			set	clear		clear	
26 RDOCR.ETRPA			set	clear		clear	
27 RDOCR.ETRPB			set	clear		clear	
28 RSCOM.RSIFC		set	set&clear	clear	set&clear	clear	
29 RSCOM.RSDC			set&clear	clear	set&clear	clear	
30 RSEN (PRR2.PRRS)	set	set	set&clear	set	set&clear	set	set
31 RDPR.PRTMP			clear				
32 RDPR.PRFLT			clear				
33 RDPR.PRPTA			clear		set&clear		
34 RDPR.PRPTB			clear		set&clear		
35 RDPR.APRPTA					set		
36 RDPR.APRPTB					set		
37 (RDPR)ARDPRFA					read		
38 (RDPR)ARDPRFB					read		
39 RXBC2.RXBPB					set&clear		
40 RXBC2.RXBF					set		
41 RXBC2.RXBCLR			set		set		
42 Reserved							
43 Reserved							
44 Reserved							
45 FEVCT							value
46 EOTCA			value		value		
47 EOTCB			value		value		
48 SOTCA			value		value		
49 SOTCB			value		value		
50 TESRA.EOTLA					value		
51 TESRB.EOTLB					value		
52 FESR.XRDY	read						
53 FESR.PLCK		read					
54 Reserved							
55 Reserved							
56 Reserved							
57 SSMFCR.IDSO					clear		
58 SSMFCR.IDSF					clear		
59 RDSIFR.EOTA					clear		
60 RDSIFR.EOTB					clear		
61 RDSIFR.SOTA					clear		
62 RDSIFR.SOTB					clear		
63 RDSIFR.WCOA					clear		
64 RDSIFR.WCOB					clear		

Legend

clear: This bit can be cleared by the state machine.
set: This bit can be set to “1” by the state machine.
set&clear: This bit can be set to “1” and/or cleared by the state machine.
value A multi-bit value is written to this register.
read This value gets read by the state machine.

3.7.11 Sequencer State Machine Register Description

SSMCR – SSM Control Register

Several options can be selected in the control register for how the chip is initialized by the state machines.

Bit	7	6	5	4	3	2	1	0	
	SETRPB	**SETRPA**	**-**	**SSMPVE**	**-**	**-**	**0**	**0**	**SSMCR**
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: SETRPB – Sequencer State Machine Enable Transparent Path B

If this bit is set, the transparent receive path B output is automatically enabled and disabled by the sequencer state machine.

Bit 6: SETRPA – Sequencer State Machine Enable Transparent Path A

If this bit is set, the transparent receive path A output is automatically enabled and disabled by the sequencer state machine.

Bit 5: Reserved Bit

This bit is reserved and reads as zero.

Bit 4: SSMPVE – Sequencer State Machine PV Enable

This bit must be set to “1” if a 5V application is in use.

Bits 3..2: Reserved Bits

These bits are reserved and read as zero.

Bit 1: Reserved Bit

This bit must always be set to zero.

Bit 0: Reserved Bit

This bit must always be set to zero.

SSMRCR – SSM Rx Control Register

Bit	7	6	5	4	3	2	1	0	
	SSMTMOE	**SSMIDSE**	**SSMIFA**	**SSMPVS**	**1**	**1**	**SSMPB**	**SSMPA**	**SSMRCR**
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: SSMTMOE – Sequencer State Machine Transparent Mode Output Enable

This bit specifies whether the transparent mode output is enabled by the sequencer state machine.

Bit 6: SSMIDSE – Sequencer State Machine ID Check Enable

This bit specifies whether the ID check is enabled.

Bit 5: SSMIFA – Sequencer State Machine IF Amplifier Enable

This bit specifies whether the IF amplifier should be used and needs to be turned on during start-up.

Bit 4: SSMPVS – Sequencer State Machine Path Valid After SOT

This bit specifies when the get telegram state machine selects the path (A or B) for further reception if both paths are activated. If SSMPVS is set to “0”, the valid receiving path is selected after a successful wake check (WCO). If it is set to “1”, the receiving path is selected after a detected start of telegram (SOT).

Bit 3: Reserved Bit

This bit must always be set to “1”.

Bit 2: Reserved Bit

This bit must always be set to “1”.

Bit 1: SSMPB – Sequencer State Machine Path B

Enables receive path B for Rx reception (Rx DSP enable and get telegram state machines).

Bit 0: SSMPA – Sequencer State Machine Path A

Enables receive path A for Rx reception (Rx DSP enable and get telegram state machines).

SSMFCR – SSM Flow Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	SSMIDSF	SSMIDSO	SSMFCR
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: SSMIDSF – Sequencer State Machine ID Check Fail

By writing a “1” to this bit, the software indicates that the ID check failed. This bit is automatically cleared by the sequencer state machine. It is not possible to write “0” to it.

Bit 0: SSMIDSO – Sequencer State Machine ID Check OK

By writing a “1” to this bit, the software indicates that the ID check was successful. This bit is automatically cleared by the sequencer state machine. It is not possible to write “0” to it.

SSMFBR – SSM Filter Bandwidth Register

Several options can be selected to control and optimize the settling wait times depending on the filter bandwidth chosen.

Bit	7	6	5	4	3	2	1	0	
	–	–	SSMPLDT	–	SSMDFDT	SSMFID[2:0]			SSMFBR
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.

Bit 5: SSMPLDT – Sequencer State Machine PLL Lock Delay Time

0 = 13µs delay time in the PLL lock state machine

1 = 63µs delay time in the PLL lock state machine

The longer setting (SSMPLDT=1) has to be used if the filter bandwidth is less than 125kHz.

Bit 4: Reserved Bit

This bit is reserved and reads as zero.

Bit 3: SSMDFDT – Sequencer State Machine Double Filter Delay Time

Writing a “1” to this bit doubles the channel filter delay time in the Rx DSP enable state machine.

Bits 2..0: SSMFID – Sequencer State Machine Filter Delay

This value defines the general channel filter delay times for the SSM after enabling the channel filter. If the used bandwidth is not in the table, the next smaller bandwidth value (longer delay time) should be used.

Table 3-39 shows the resulting delay times in the Rx DSP enable and the get telegram state machine depending on the SSMFID and SSMDFDT settings.

Table 3-39. SSMFIB Filter Bandwidth Settings

SSMFID	Delay Time in Rx DSP Enable		Bandwidth of Channel Filter
	SSMDFDT=0	SSMDFDT=1	
0x0	380µs	760µs	25kHz
0x1	202.µs	405µs	50kHz
0x2	135µs	270µs	80kHz
0x3	75µ	150µs	165kHz
0x4	58.5µs	117µs	235kHz
0x5	45µs	90µs	360kHz
0x6	380µs	760µs	25kHz
0x7	380µs	760µs	25kHz

SSMRR – SSM Run Register

The master state machine can be started and stopped by writing to the corresponding configuration bits. If started, the state machine calls the sub-state machines as configured in the MSMCRx registers and – if enabled in SSMIMR register – requests an interrupt at the end of operation.

If necessary, all state machine operation can be stopped immediately by setting the SSMST bit.

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	SSMST	SSMR	SSMRR
Read/Write	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: SSMST – Sequencer State Machine Stop

Writing a “1” to this bit immediately stops the master state machine and the currently running sub-state machine and resets the SSMR bit. No interrupt is given by this software-driven stop. SSMST is reset automatically after one clock cycle and therefore always reads “0”.

After giving the stop signal to the state machine, it is not defined in what state the design is in. Starting the shut down state machine after the software-driven stop should be considered in order to put the design back into a safe and known state.

Bit 0: SSMR – Sequencer State Machine Run

Writing a “1” to this bit starts the master state machine. The bit is reset automatically when the master state machine has completed all configured sub-state machines or the software writes a “1” to the SSMST bit. Reading this bit is an indicator of whether the state machine is running or has completed its operation.

SSMSR – SSM Status Register

Bit	7	6	5	4	3	2	1	0	
	SSMERR	**–**	**–**	**–**	**SSMESM[3:0]**				**SSMSR**
Read/Write	R/W	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7: SSMERR – Sequencer State Machine Error

This bit is set to “1” if the sequencer state machine stops operation with an error condition, such as when a sub-state machine is waiting for a condition and the predefined time is over.

The flag can be reset by writing a “1” to this bit.

Bits 6..4: Reserved Bits

These bits are reserved and read as zero.

Bits 3..0: SSMESM – Sequencer State Machine Error State Machine

This value holds the number (see Table 3-40 on page 190) of the state machine that raised the most recent error. Possible candidates for error are PLL enable, PLL lock, and get telegram state machines.

SSMIFR – SSM Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	–	**–**	**–**	**–**	**–**	**–**	**–**	**SSMIF**	**SSMIFR**
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: SSMIF – Sequencer State Machine Interrupt Flag

This flag is set to “1” if the master state machine has reached the end of its programmed sequence or has encountered an error and has completed operation. It is cleared automatically if the SSM interrupt routine is executed or it can be reset manually by writing a “1” to this bit.

SSMIMR – SSM Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	–	**–**	**–**	**–**	**–**	**–**	**–**	**SSMIM**	**SSMIMR**
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: SSMIM – Sequencer State Machine Interrupt Mask

If this bit is set to “1” and if global interrupts are enabled, the sequencer state machine issues an interrupt when the SSMIF flag in SSMIFR is set. If SSMIM is “0”, no interrupt is requested when the master state machine completes.

MSMSTR – Master State Machine State Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	SSMMST[4:0]					MSMSTR
Read/Write	R	R	R	R	R	R	R	R	
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..0: SSMMST – Sequencer State Machine Master State

This value gives the current state of the master state machine. Reading this register triggers sampling of SSMSTR and SSMXSR to get a consistent state of the entire SSM block. This value is intended mainly for debugging. A read access to this register automatically updates the SSMSTR and SSMXSR registers.

SSMSTR – SSM State Register

Bit	7	6	5	4	3	2	1	0	
	–	–	SSMSTA[5:0]						SSMSTR
Read/Write	R	R	R	R	R	R	R	R	
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: SSMSTA – Sequencer State Machine State A

This value holds the state of path A of the get telegram state machine or the state of any other sub-state machine currently running. It is sampled when a read access to MSMSTR takes place to have the state value and the master state at the same time. This value is intended mainly for debugging.

SSMXSR – SSM Extended State Register

Bit	7	6	5	4	3	2	1	0	
	–	–	SSMSTB[5:0]						SSMXSR
Read/Write	R	R	R	R	R	R	R	R	
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: SSMSTB – Sequencer State Machine State B

This value holds the state of path B of the get telegram state machine. It is sampled when a read access to MSMSTR takes place to have the state values of both paths and the master state at the same time. If any other state machine than the get telegram is running, then this value reads as zero. This value is intended mainly for debugging.

MSMCR1 – Master State Machine Control Register 1

The SSM module has eight places to configure and store the master state machine run order. MSMSM0-MSMSM7 holds a maximum of eight numbers defining the sub-state machines as shown in Table 3-40 on page 190.

Using these fields allows the definition of the exact initialization or run process in the right order. By inserting appropriate state machine numbers, it is possible to, for example, configure whether VCO tuning should take place or not.

The first entry in this series of registers that is set to zero defines the end of the configuration. The master state machine stops operation upon reaching this zero value. Undefined values for MSMSMx are treated as “0” and lead to a completed state machine sequence.

Bit	7	6	5	4	3	2	1	0	
	MSMSM1[3:0]				MSMSM0[3:0]				MSMCR1
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..4: MSMSM1 – Master State Machine (Sub)State Machine Select 1

This value defines the second sub-state machine that should be run by the SSM master state machine.

Bits 3..0: MSMSM0 – Master State Machine (Sub)State Machine Select 0

This value defines the first sub-state machine that should be run by the SSM master state machine.

See Table 3-40 for configuration.

Table 3-40. Sub-State Machine Selection

MSMSMx	Description
0	No sub-state machine → end state
1	PLL enable (see Section 3.7.3 “PLL Enable State Machine” on page 178)
2	PLL lock (see Section 3.7.4 “PLL Lock State Machine” on page 178)
3	Rx DSP enable (see Section 3.7.5 “Rx DSP Enable State Machine” on page 178)
4	Rx DSP disable (see Section 3.7.6 “Rx DSP Disable State Machine” on page 179)
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Get telegram (see Section 3.7.7 “Get Telegram State Machine” on page 179)
10	Reserved
11	Shut down (see Section 3.7.8 “Shut Down State Machine” on page 182)
12	VCO tuning (see Section 3.7.9 “VCO Tuning State Machine” on page 182)
13	Reserved

MSMCR2 – Master State Machine Control Register 2

Bit	7	6	5	4	3	2	1	0	
	MSMSM3[3:0]				MSMSM2[3:0]				MSMCR2
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..4: MSMSM3 – Master State Machine (Sub)State Machine Select 3

This value defines the fourth sub-state machine that should be run by the SSM master state machine.

Bits 3..0: MSMSM2 – Master State Machine (Sub)State Machine Select 2

This value defines the third sub-state machine that should be run by the SSM master state machine.

See Table 3-40 on page 190 for configuration.

MSMCR3 – Master State Machine Control Register 3

Bit	7	6	5	4	3	2	1	0	
	MSMSM5[3:0]				MSMSM4[3:0]				MSMCR3
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..4: MSMSM5 – Master State Machine (Sub)State Machine Select 5

This value defines the sixth sub-state machine that should be run by the SSM master state machine.

Bits 3..0: MSMSM4 – Master State Machine (Sub)State Machine Select 4

This value defines the fifth sub-state machine that should be run by the SSM master state machine.

See Table 3-40 on page 190 for configuration.

MSMCR4 – Master State Machine Control Register 4

Bit	7	6	5	4	3	2	1	0	
	MSMSM7[3:0]				MSMSM6[3:0]				MSMCR4
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..4: MSMSM7 – Master State Machine (Sub)State Machine Select 7

This value defines the eighth sub-state machine that should be run by the SSM master state machine.

Bits 3..0: MSMSM6 – Master State Machine (Sub)State Machine Select 6

This value defines the seventh sub-state machine that should be run by the SSM master state machine.

See Table 3-40 on page 190 for configuration.

GTCR – Get Telegram Control Register

Bit	7	6	5	4	3	2	1	0	
	IWUPB	DARB	GAPMB	RXTEHB	IWUPA	DARA	GAPMA	RXTEHA	GTCR
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: IWUPB – Intermittent WUP Mode Path B

If this bit is set to “1”, path B is restarted when an end of telegram (EOT) is detected while waiting for a wait check OK (WCO). Wake check OK time-out remains untouched and continues to run.

If this bit is cleared, receiving an EOT while waiting for WCO is considered an error and the get telegram state machine is either stopped or the path B is restarted, depending on the DARB setting. Wake check OK time-out is restarted, too.

Bit 6: DARB – Demodulator Automatic Restart on Path B

If this bit is set to “1”, do an automatic restart of demodulator path B if an EOTB has occurred while waiting for WCOB or SOTB. If this bit is set to “0”, reception is stopped after an EOTB or time-out has occurred on path B and the state machine path B stops its operation.

Bit 5: GAPMB – Gap Mode Path B

If this bit is set to “1”, path B is restarted when an end of telegram (EOTB) is detected while waiting for a start of telegram (SOTB). The SOT time-out remains untouched and continues to run.

If this bit is cleared, receiving an EOTB while waiting for SOTB is considered an error and the get telegram state machine is either stopped or the path B is restarted, depending on the DARB setting. All time-outs are restarted, too.

Bit 4: RXTEHB – Rx Telegram End Handling Path B

This bit selects the behavior of the state machine after successfully receiving an end of telegram.

0 = stop immediate

1 = restart Rx after EOT

Bit 3: IWUPA – Intermittent WUP Mode Path A

If this bit is set to “1”, path A is restarted when an end of telegram (EOT) is detected while waiting for a wait check OK (WCO). Wake check OK time-out remains untouched and continues to run.

If this bit is cleared, receiving an EOT while waiting for WCO is considered an error and the get telegram state machine is either stopped or path A is restarted, depending on the DARA setting. Wake check OK time-out is restarted, too.

Bit 2: DARA – Demodulator Automatic Restart on Path A

If this bit is set to “1”, do an automatic restart of demodulator path A if an EOTA has occurred while waiting for WCOA or SOTA. If this bit is set to “0”, reception is stopped after an EOTA or time-out has occurred on path A and the state machine path A stops its operation.

Bit 1: GAPMA – Gap Mode Path A

If this bit is set to “1”, path A is restarted when an end of telegram (EOTA) is detected while waiting for a start of telegram (SOTA). The SOT time-out remains untouched and continues to run.

If this bit is cleared, receiving an EOTA while waiting for SOTA is considered an error and the get telegram state machine is either stopped or the path A is restarted completely depending on DARA setting. All time-outs are restarted, too.

Bit 0: RXTEHA – Rx Telegram End Handling Path A

This bit selects the behavior of the state machine after the successful reception of an end of telegram.

0 = stop immediate

1 = restart Rx after EOT

SOTC1A – Start of Telegram Conditions 1 for Path A

This register stores the SOTCA settings that are valid from the start of the reception until a wake check OK (WCOA) is detected. The sequencer state machine copies its content at the beginning of the reception to the SOTCA register.

Bit	7	6	5	4	3	2	1	0	
	WCOBOE	**RROEA**	**SFIDEA**	**WUPEA**	**MANOEA**	**SYTOEA**	**AMPOEA**	**CAROEA**	**SOTC1A**
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	

SOTC2A – Start of Telegram Conditions 2 for Path A

This register stores the SOTCA settings that are valid from the wake check OK (WCOA) to start of telegram OK (SOTA). The sequencer state machine copies its content immediately after WCOA detection to the SOTCA register.

Bit	7	6	5	4	3	2	1	0	
	WCOBOE	**RROEA**	**SFIDEA**	**WUPEA**	**MANOEA**	**SYTOEA**	**AMPOEA**	**CAROEA**	**SOTC2A**
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	

SOTC1B – Start of Telegram Conditions 1 for Path B

This register stores the SOTCB settings that are valid from the start of the reception until a wake check OK (WCOB) is detected. The sequencer state machine copies its content at the beginning of the reception to the SOTCB register.

Bit	7	6	5	4	3	2	1	0	
	WCOAOE	**RROEB**	**SFIDEB**	**WUPEB**	**MANOEB**	**SYTOEB**	**AMPOEB**	**CAROEB**	**SOTC1B**
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	

SOTC2B – Start of Telegram Conditions 2 for Path B

This register stores the SOTCB settings that are valid from the wake check OK (WCOB) to start of telegram OK (SOTB). The sequencer state machine copies its content immediately after WCOB detection to the SOTCB register.

Bit	7	6	5	4	3	2	1	0	
	WCOAOE	RROEB	SFIDEB	WUPEB	MANOEB	SYTOEB	AMPOEB	CAROEB	SOTC2B
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	

EOTC1A – End of Telegram Conditions 1 for Path A

This register stores the EOTCA settings that are valid from the start of the reception until a wake check OK (WCOA) is detected. The sequencer state machine copies its content at the beginning of the reception to the EOTCA register.

Bit	7	6	5	4	3	2	1	0	
	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA	EOTC1A
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	

EOTC2A – End of Telegram Conditions 2 for Path A

This register stores the EOTCA settings that are valid from a valid wake check OK (WCOA) up to the successful start of telegram (SOTA) detection. The sequencer state machine copies its content after a successful WCOA to the EOTCA register.

Bit	7	6	5	4	3	2	1	0	
	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA	EOTC2A
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	

EOTC3A – End of Telegram Conditions 3 for Path A

This register stores the EOTCA settings that are valid from a valid start of telegram detection (SOTA) until the end of the telegram. The sequencer state machine copies its content after a successful SOTA check to the EOTCA register.

Bit	7	6	5	4	3	2	1	0	
	EOTBFE	RRFEA	TELREA	TMOFEA	MANFEA	SYTFEA	AMPFEA	CARFEA	EOTC3A
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	

EOTC1B – End of Telegram Conditions 1 for Path B

This register stores the EOTCB settings that are valid from the start of the reception until a wake check OK (WCOB) is detected. The sequencer state machine copies its content at the beginning of the reception to the EOTCB register.

Bit	7	6	5	4	3	2	1	0	
	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB	EOTC1B
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	

EOTC2B – End of Telegram Conditions 2 for Path B

This register stores the EOTCB settings that are valid from a valid wake check OK (WCOB) up to the successful start of telegram (SOTB) detection. The sequencer state machine copies its content after a successful WCOB to the EOTCB register.

Bit	7	6	5	4	3	2	1	0	
	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB	EOTC2B
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	

EOTC3B – End of Telegram Conditions 3 for Path B

This register stores the EOTCB settings that are valid from a valid start of telegram detection (SOTB) until the end of the telegram. The sequencer state machine copies its content after a successful SOTB check to the EOTCB register.

Bit	7	6	5	4	3	2	1	0	
	EOTAFE	RRFEB	TELREB	TMOFEB	MANFEB	SYTFEB	AMPFEB	CARFEB	EOTC3B
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	

WCOTOA – Wake Check OK Time-out for Path A

This value defines the time before the get telegram state machine stops waiting for a wake check OK signal on path A. The state machine either stops itself or triggers a restart of path A depending on the DARA configuration in the GTCR register.

The wait time is a piece-wise linear approximation of an exponential curve. The delay time in XTO/4 cycles can be calculated by the following formula:

$$delaycycles = (WCOTOA[3:0] \times 2 + 32) \times (2^{WCOTOA[7:4]}) \quad (40)$$

A few example values are given in Table 3-41 on page 194.

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

Table 3-41. Time-out Delay Time Examples

WCOTOA[7:0]	Delay Time	Example Time at 26MHz XTO Frequency
0	infinite	infinite
1	34 cycles	5.23µs
2	36 cycles	5.54µs
32	128 cycles	19.69µs
33	136 cycles	20.92µs
64	512 cycles	78.77µs
128	8192 cycles	1.26ms
192	131072 cycles	20.16ms
255	2031616 cycles	312.56ms

WCOTOB – Wake Check OK Time-out for Path B

This value defines the time to wait before the get telegram state machine stops waiting for a wake check OK signal on path B. The state machine either stops itself or triggers a restart of path B depending on the DARB configuration in the GTCR register.

For calculation of the value, see “WCOTOA – Wake Check OK Time-out for Path A” on page 194.

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

SOTTOA – Start of Telegram Time-out for Path A

This value defines the time to wait before the get telegram state machine stops waiting for a start of telegram signal on path A. The state machine either stops itself or triggers a restart of path A depending on the DARA configuration in the GTCR register.

For calculation of the value, see “WCOTOA – Wake Check OK Time-out for Path A” on page 194.

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

SOTTOB – Start of Telegram Time-out for Path B

This value defines the time to wait before the get telegram state machine stops waiting for a start of telegram signal on path B. The state machine either stops itself or triggers a restart of path B depending on the DARB configuration in the GTCR register. For calculation of the value, see “WCOTOA – Wake Check OK Time-out for Path A” on page 194.

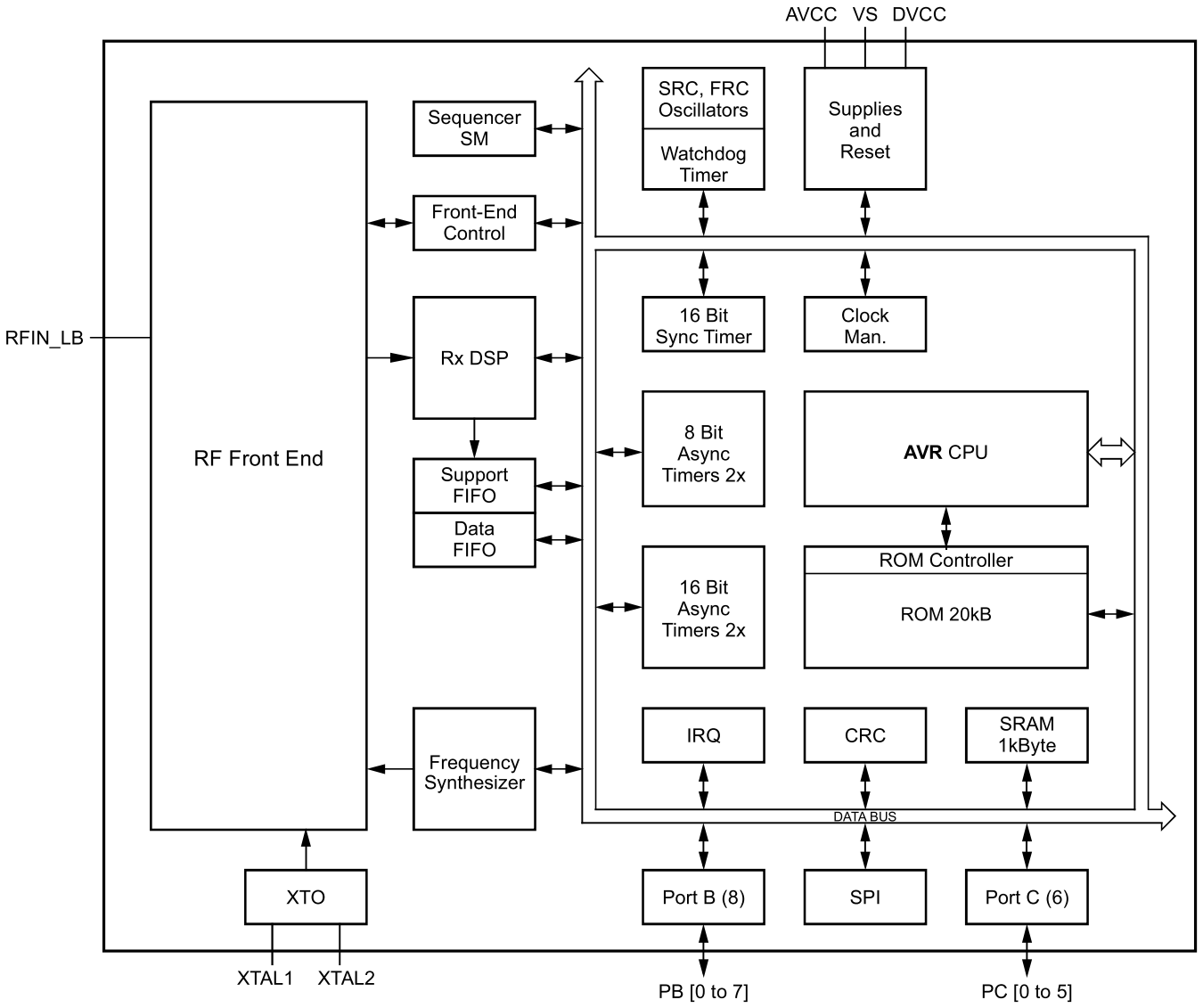
Bit	7	6	5	4	3	2	1	0	
	SOTTOB[7:4]				SOTTOB[3:0]				SOTTOB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.8 AVR Controller

3.8.1 AVR Controller Sub-System Overview

Figure 3-28 shows a block diagram of the Atmel® ATA5785. The AVR® controller sub-system consists of the AVR CPU core, its program memory, and a data bus with data memory and peripheral blocks attached. The receive path also has its user interfaces connected to the data bus.

Figure 3-28. AVR Controller Sub-System Overview

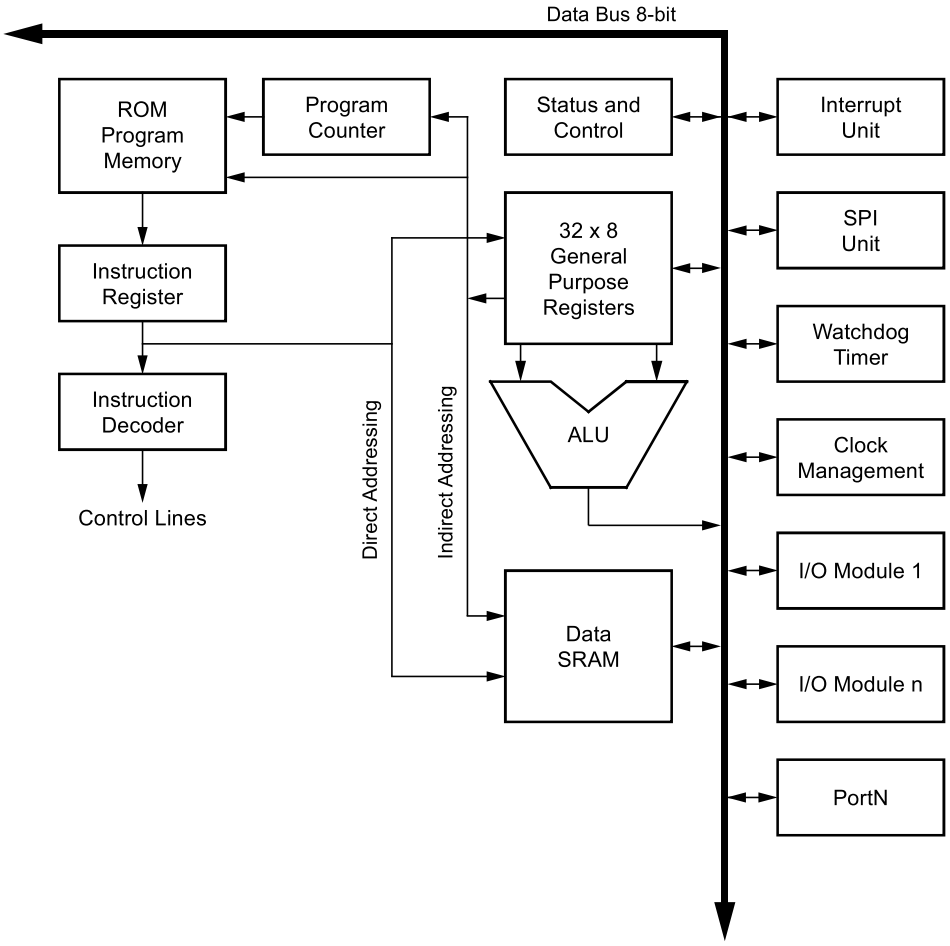


3.8.2 CPU Core

3.8.2.1 Architectural Overview

This section discusses the AVR® core architecture in general. The main function of the CPU core is to ensure correct program execution. For this reason, the CPU core must be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 3-29. Overview of Architecture



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

The ATA5785 is a ROM-based firmware-only receiver IC. Software customization by the end user is not possible and hence no detailed description of the AVR functions and registers is given in this user manual.

3.8.3 Memories

This section describes the different memories of the Atmel® ATA5785. The AVR® architecture has two memory spaces. The data memory for volatile data and the ROM program memory. All memory spaces are linear and regular.

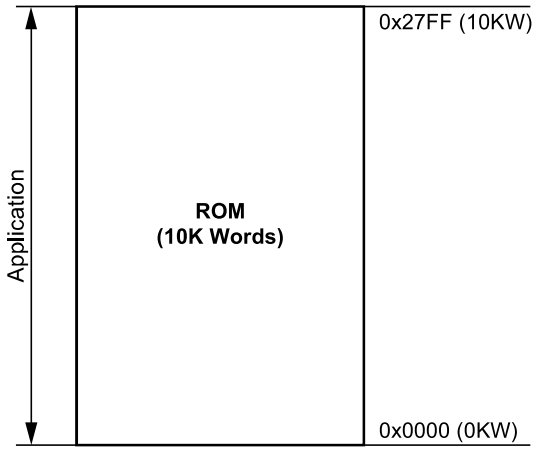
Table 3-42. Memory Overview

Type	Details	Size	Start	Stop
Data	General Purpose Registers	32byte	0x0000	0x001F
	I/O Registers 1	32byte	0x0020	0x003F
	I/O Registers 2	32byte	0x0040	0x005F
	Extended I/O Registers	416byte	0x0060	0x01FF
	Internal SRAM	1024byte	0x0200	0x05FF
Program Memory	ROM Boot Memory	10Kwords (20KB)	0x0000	0x27FF (Words)

3.8.3.1 Program Memory

The program memory of the Atmel ATA5785 consists of a 10K-words firmware ROM as shown in Figure 3-30 on page 198. User-defined software extensions are not possible.

Figure 3-30. Program Memory Map of the Atmel ATA5785



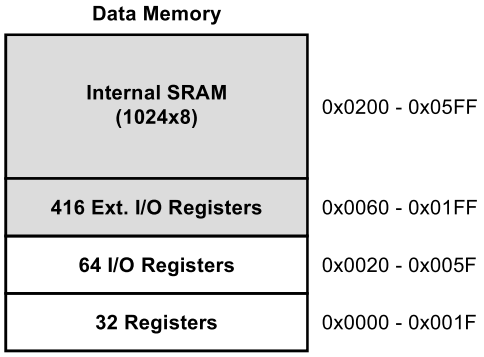
The 20KB ROM provides the Atmel firmware functionality of the Atmel ATA5785, including initialization, self calibration and automatic receiving of multiple channels with configurable parameters. The firmware features are described in Section 2. “System Functional Description” on page 10.

3.8.3.2 Data Memory

The AVR® data memory consists of 32 general purpose working registers, a decent number of hardware i/O registers and 1Kbyte of internal SRAM.

The first 32 locations address the register file, the next 64 locations the standard I/O memory, then 416 locations of extended I/O memory followed by 1024 locations which address the internal data SRAM as shown in Figure 3-31.

Figure 3-31. Data Memory Map



3.8.3.3 General Purpose I/O Register

The AVR contains seven general purpose I/O registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and status flags.

Note: The registers GPIOR0, GPIOR3 and GPIOR4 are used by the Atmel® ATA5785 firmware and are therefore not available for the user.

General Purpose I/O Register Description

GPIOR0 – General Purpose I/O Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	GPIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

GPIOR1 – General Purpose I/O Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	GPIOR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

GPIOR2 – General Purpose I/O Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	GPIOR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	