



# **Programming Manual**

## **AX5043**

**Advanced high performance  
ASK and FSK narrow-band  
transceiver for 27-1050 MHz  
range**

## TABLE OF CONTENTS

1. Overview.....	7
1.1. Connecting the AX5043 to an AX8052F100 or other Microcontroller.....	8
1.2. Pin Function Descriptions.....	9
1.3. SPI Register Access.....	10
1.3.1. Deep Sleep.....	12
1.3.2. Address Space.....	12
2. FIFO Operation.....	14
2.1. FIFO Chunk Encoding.....	15
2.1.1. NOP Command.....	16
2.1.2. RSSI Command.....	16
2.1.3. TXCTRL Command.....	17
2.1.4. FREQOFFS Command.....	17
2.1.5. ANTRSSI2 Command.....	17
2.1.6. REPEATDATA Command.....	18
2.1.7. TIMER Command.....	18
2.1.8. RFFREQOFFS Command.....	18
2.1.9. DATARATE Command.....	19
2.1.10. ANTRSSI3 Command.....	19
2.1.11. DATA Command.....	19
2.1.11.1. Transmit Data Format.....	19
2.1.11.2. Receive Data Format.....	21
2.1.12. TXPWR Command.....	22
3. Programming the Chip.....	23
3.1. Power Modes.....	23
3.1.1. FIFO Power Management.....	24
3.2. Autoranging.....	24
3.3. Choosing the Fundamental Communication Characteristics.....	26
3.4. Framing.....	27
3.5. Transmitter.....	29
3.5.1. Recommended Preamble.....	31
3.6. Receiver.....	32
3.6.1. Receiver State Machine.....	34
3.7. Low Power Oscillator Calibration.....	36
3.8. Auxiliary DAC.....	37
4. Register Overview.....	40
5. Register Details.....	58
5.1. Revision and Interface Probing.....	58
5.1.1. Register: REVISION.....	58
5.1.2. Register: SCRATCH.....	58
5.2. Operating Mode.....	58
5.2.1. Register: PWRMODE.....	58
5.3. Power Management.....	59
5.3.1. Register: POWSTAT.....	59
5.3.2. Register: POWSTICKYSTAT.....	59
5.3.3. Register: POWIRQMASK.....	60
5.4. Interrupt Control.....	60
5.4.1. Register: IRQMASK1, IRQMASK0.....	60

5.4.2.Register: RADIOEVENTMASK1, RADIOEVENTMASK0.....	61
5.4.3.Register: IRQINVERSION1, IRQINVERSION0.....	61
5.4.4.Register: IRQREQUEST1, IRQREQUEST0.....	61
5.4.5.Register: RADIOEVENTREQ1, RADIOEVENTREQ0.....	62
5.5.Modulation and Framing.....	62
5.5.1.Register: MODULATION.....	62
5.5.2.Register: ENCODING.....	63
5.5.3.Register: FRAMING.....	65
5.5.4.Register: CRCINIT3, CRCINIT2, CRCINIT1, CRCINIT0.....	66
5.6.Forward Error Correction.....	66
5.6.1.Register: FEC.....	66
5.6.2.Register: FECSYNC.....	67
5.6.3.Register: FECSTATUS.....	67
5.7.Status.....	67
5.7.1.Register: RADIOSTATE.....	67
5.7.2.Register: XTALSTATUS.....	68
5.8.Pin Configuration.....	68
5.8.1.Register: PINSTATE.....	68
5.8.2.Register: PINFUNCSYSCCLK.....	68
5.8.3.Register: PINFUNCDCLK.....	69
5.8.4.Register: PINFUNCDDATA.....	70
5.8.5.Register: PINFUNCIRQ.....	70
5.8.6.Register: PINFUNCANTSSEL.....	70
5.8.7.Register: PINFUNCPWRAMP.....	71
5.8.8.Register: PWRAMP.....	71
5.9.FIFO Registers.....	72
5.9.1.Register: FIFOSTAT.....	72
5.9.2.Register: FIFODATA.....	73
5.9.3.Register: FIFOCOUNT1, FIFOCOUNT0.....	73
5.9.4.Register: FIFOFREE1, FIFOFREE0.....	73
5.9.5.Register: FIFOTHRESH1, FIFOTHRESH0.....	73
5.10.Synthesizer.....	73
5.10.1.Register: PLLLOOP, PLLLOOPBOOST.....	73
5.10.2.Register: PLLCPI, PLLCPIBOOST.....	74
5.10.3.Register: PLLVCODIV.....	74
5.10.4.Register: PLLRANGINGA, PLLRANGINGB.....	74
5.10.5.Register: FREQA3, FREQA2, FREQA1, FREQA0.....	75
5.10.6.Register: FREQB3, FREQB2, FREQB1, FREQB0.....	75
5.11.Signal Strength.....	75
5.11.1.Register: RSSI.....	75
5.11.2.Register: BGNDRSSI.....	76
5.11.3.Register: DIVERSITY.....	76
5.11.4.Register: AGCCOUNTER.....	76
5.12.Receiver Tracking.....	76
5.12.1.Register: TRKDATARATE2, TRKDATARATE1, TRKDATARATE0.....	76
5.12.2.Register: TRKAMPL1, TRKAMPL0.....	76

5.12.3.Register: TRKPHASE1, TRKPHASE0.....	76
5.12.4.Register: TRKRFFREQ2, TRKRFFREQ1, TRKRFFREQ0.....	77
5.12.5.Register: TRKFREQ1, TRKFREQ0.....	77
5.12.6.Register: TRKFSKDEMOD1, TRKFSKDEMOD0.....	77
5.12.7.Register: TRKAFSKDEMOD1, TRKAFSKDEMOD0.....	77
5.12.8.Tracking Register Resets.....	77
5.13.Timer.....	78
5.13.1.Register: TIMER2, TIMER1, TIMER0.....	78
5.14.Wakeup Timer.....	78
5.14.1.Register: WAKEUPTIMER1, WAKEUPTIMER0.....	79
5.14.2.Register: WAKEUP1, WAKEUP0.....	79
5.14.3.Register: WAKEUPFREQ1, WAKEUPFREQ0.....	79
5.14.4.Register: WAKEUPXOEARLY.....	79
5.15.Receiver Parameters.....	79
5.15.1.Register: IFFREQ1, IFFREQ0.....	79
5.15.2.Register: DECIMATION.....	79
5.15.3.Register: RXDATARATE2, RXDATARATE1, RXDATARATE0.....	80
5.15.4.Register: MAXDROFFSET2, MAXDROFFSET1, MAXDROFFSET0.....	80
5.15.5.Register: MAXRFOFFSET2, MAXRFOFFSET1, MAXRFOFFSET0.....	80
5.15.6.Register: FSKDMAX1, FSKDMAX0.....	81
5.15.7.Register: FSKDMIN1, FSKDMIN0.....	81
5.15.8.Register: AFSKSPACE1, AFSKSPACE0.....	81
5.15.9.Register: AFSKMARK1, AFSKMARK0.....	81
5.15.10.Register: AFSKCTRL.....	82
5.15.11.Register: AMPLFILTER.....	82
5.15.12.Register: FREQUENCYLEAK.....	82
5.15.13.Register: RXPARAMSETS.....	82
5.15.14.Register: RXPARAMCURSET.....	83
5.15.15.Register: AGCGAIN0, AGCGAIN1, AGCGAIN2, AGCGAIN3.....	83
5.15.16.Register: AGCTARGET0, AGCTARGET1, AGCTARGET2, AGCTARGET3.....	84
5.15.17.Register: AGCAHYST0, AGCAHYST1, AGCAHYST2, AGCAHYST3.....	84
5.15.18.Register: AGCMINMAX0, AGCMINMAX1, AGCMINMAX2, AGCMINMAX3.....	85
5.15.19.Register: TIMEGAIN0, TIMEGAIN1, TIMEGAIN2, TIMEGAIN3.....	85
5.15.20.Register: DRGAIN0, DRGAIN1, DRGAIN2, DRGAIN3.....	86
5.15.21.Register: PHASEGAIN0, PHASEGAIN1, PHASEGAIN2, PHASEGAIN3.....	86
5.15.22.Register: FREQGAINA0, FREQGAINA1, FREQGAINA2, FREQGAINA3.....	87
5.15.23.Register: FREQGAINB0, FREQGAINB1, FREQGAINB2, FREQGAINB3.....	88
5.15.24.Register: FREQGAINC0, FREQGAINC1, FREQGAINC2, FREQGAINC3.....	88
5.15.25.Register: FREQGAIND0, FREQGAIND1, FREQGAIND2, FREQGAIND3.....	89
5.15.26.Register: AMPLGAIN0, AMPLGAIN1, AMPLGAIN2, AMPLGAIN3.....	89
5.15.27.Register: FREQDEV10, FREQDEV00, FREQDEV11, FREQDEV01, FREQDEV12, FREQDEV02, FREQDEV13, FREQDEV03.....	90
5.15.28.Register: FOURFSK0, FOURFSK1, FOURFSK2, FOURFSK3.....	90
5.15.29.Register: BBOFFSRES0, BBOFFSRES1, BBOFFSRES2, BBOFFSRES3.....	92
5.16.Transmitter Parameters.....	93
5.16.1.Register: MODCFGF.....	93
5.16.2.Register: FSKDEV2, FSKDEV1, FSKDEV0.....	93

5.16.3.Register: MODCFG	94
5.16.4.Register: TXRATE2, TXRATE1, TXRATE0	95
5.16.5.Register: TXPWRCOEFFA1, TXPWRCOEFFA0	95
5.16.6.Register: TXPWRCOEFFB1, TXPWRCOEFFB0	95
5.16.7.Register: TXPWRCOEFFC1, TXPWRCOEFFC0	96
5.16.8.Register: TXPWRCOEFFD1, TXPWRCOEFFD0	96
5.16.9.Register: TXPWRCOEFFE1, TXPWRCOEFFE0	96
5.17.PLL Parameters	97
5.17.1.Register: PLLVCOI	97
5.17.2.Register: PLLVCOIR	97
5.17.3.Register: PLLLOCKDET	97
5.17.4.Register: PLLRNGCLK	98
5.18.Crystal Oscillator	98
5.18.1.Register: XTALCAP	98
5.19.Baseband	99
5.19.1.Register: BBTUNE	99
5.19.2.Register: BBOFFSCAP	99
5.20.Packet Format	99
5.20.1.Register: PKTADDRCFG	99
5.20.2.Register: PKTLENCFG	99
5.20.3.Register: PKTLENOFFSET	100
5.20.4.Register: PKTMAXLEN	100
5.20.5.Register: PKTADDR3, PKTADDR2, PKTADDR1, PKTADDR0	100
5.20.6.Register: PKTADDRMASK3, PKTADDRMASK2, PKTADDRMASK1, PKTADDRMASK0	101
5.21.Pattern Match	101
5.21.1.Register: MATCHOPAT3, MATCHOPAT2, MATCHOPAT1, MATCHOPAT0	101
5.21.2.Register: MATCHOLEN	101
5.21.3.Register: MATCH0MIN	101
5.21.4.Register: MATCH0MAX	101
5.21.5.Register: MATCH1PAT1, MATCH1PAT0	102
5.21.6.Register: MATCH1LEN	102
5.21.7.Register: MATCH1MIN	102
5.21.8.Register: MATCH1MAX	102
5.22.Packet Controller	102
5.22.1.Register: TMGTXB00ST	102
5.22.2.Register: TMGTXSETTLE	103
5.22.3.Register: TMGRXB00ST	103
5.22.4.Register: TMGRXSETTLE	103
5.22.5.Register: TMGRXOFFSACQ	103
5.22.6.Register: TMGRXCOARSEAGC	103
5.22.7.Register: TMGRXAGC	104
5.22.8.Register: TMGRXRSSI	104
5.22.9.Register: TMGRXPREAMBLE1	104
5.22.10.Register: TMGRXPREAMBLE2	104
5.22.11.Register: TMGRXPREAMBLE3	105

5.22.12.Register: RSSIREFERENCE.....	105
5.22.13.Register: RSSIABSTHR.....	105
5.22.14.Register: BGNDRSSIGAIN.....	105
5.22.15.Register: BGNDRSSITHR.....	106
5.22.16.Register: PKTCHUNKSIZE.....	106
5.22.17.Register: PKTMISCFLAGS.....	106
5.22.18.Register: PKTSTOREFLAGS.....	107
5.22.19.Register: PKTACCEPTFLAGS.....	107
5.23.General Purpose ADC.....	108
5.23.1.Register: GPADCCTRL.....	108
5.23.2.Register: GPADCPERIOD.....	108
5.23.3.Register: GPADC13VALUE1, GPADC13VALUE0.....	108
5.24.Low Power Oscillator Calibration.....	108
5.24.1.Register: LPOSCCONFIG.....	108
5.24.2.Register: LPOSCSTATUS.....	109
5.24.3.Register: LPOSCFILT1, LPOSCFILT0.....	109
5.24.4.Register: LPOSCREF1, LPOSCREF0.....	109
5.24.5.Register: LPOSCFREQ1, LPOSCFREQ0.....	110
5.24.6.Register: LPOSCPER1, LPOSCPER0.....	110
5.25.DAC.....	110
5.25.1.Register: DACVALUE1, DACVALUE0.....	110
5.25.2.Register: DACCONFIG.....	110
5.26.Performance Tuning Registers.....	111
6.References.....	113
7.Contact Information.....	114

## 1. OVERVIEW

**AX5043** is a true single chip low-power CMOS transceiver for narrow band applications. A fully integrated VCO supports carrier frequencies in the 433MHz, 868MHz and 915MHz ISM band. An external VCO inductor enables carrier frequencies from 27MHz to 1050MHz. The on-chip transceiver consists of a fully integrated RF front-end with modulator, and demodulator. Base band data processing is implemented in an advanced and flexible communication controller that enables user friendly communication via the SPI interface. An on-chip low power oscillator as well as Wake-on-radio enable very low power standby applications. The **AX5043** is also available with the **AX8052F100** microcontroller in a single integrated circuit as the **AX8052F143**. Figure 1 shows the block diagram of the **AX5043**.

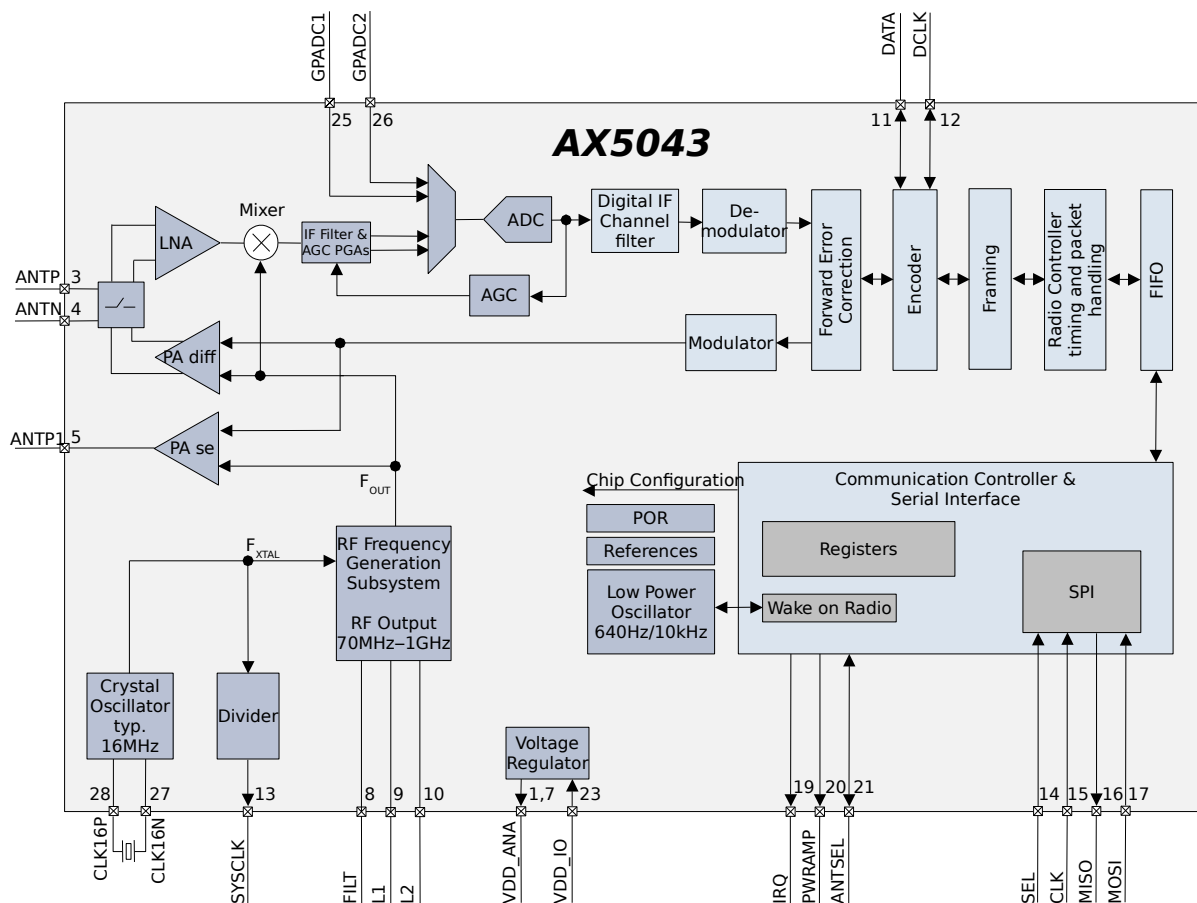


Figure 1: Block Diagram

### 1.1. CONNECTING THE **AX5043** TO AN **AX8052F100** OR OTHER MICROCONTROLLER

The **AX5043** can easily be connected to an **AX8052F100** or any other microcontroller. The microcontroller communicates with the **AX5043** via a register file that is implemented in the **AX5043** and that can be accessed serially via an industry standard Serial Peripheral Interface (SPI) protocol.

Reset is performed by the integrated power-on-reset (POR) block and can be performed manually via the register file.

The **AX5043** sends and receives data via the SPI port in frames. This standard operation mode is called frame mode.

In frame mode, the internal communication controller performs frame delimiting, and data is received and transmitted via a 256 Byte FIFO, accessible via the register file. The FIFO is shared between receive and transmit. Figure 2 shows the corresponding diagram. Connecting the interrupt line is highly recommended, though not strictly required. With the **AX8052F100**, it is also recommended to connect the SYSCLK line. This allows the Microcontroller to run from the precise crystal clock of the **AX5043**, or to calibrate its internal oscillators from against this clock.

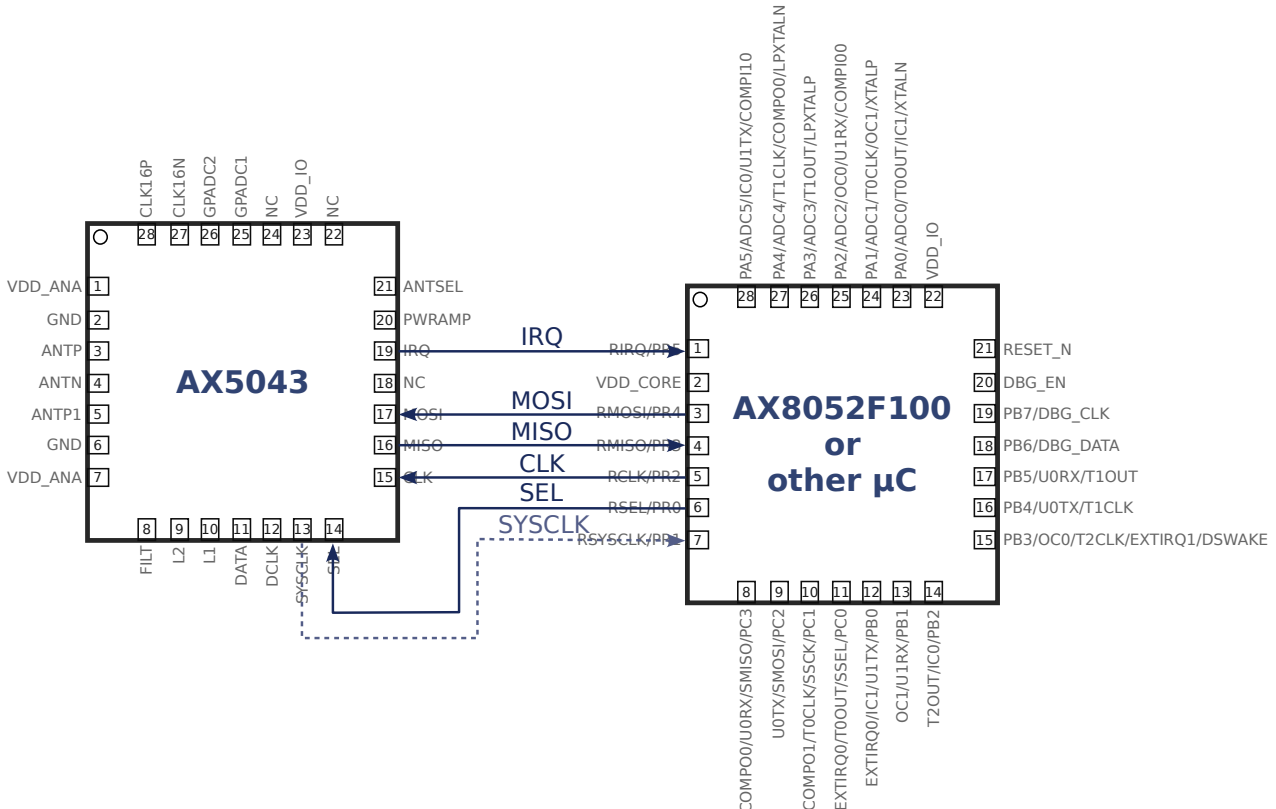


Figure 2: Connecting AX5043 to AX8052F100 or other  $\mu$ C



## 1.2. PIN FUNCTION DESCRIPTIONS

Symbol	Pin(s)	Type	Description
VDD_ANA	1	P	Analog power output, decouple to neighboring GND
GND	2	P	Ground, decouple to neighboring VDD_ANA
ANTP	3	A	Differential antenna input/output
ANTN	4	A	Differential antenna input/output
ANTP1	5	A	Single-ended antenna output
GND	6	P	Ground, decouple to neighboring VDD_ANA
VDD_ANA	7	P	Analog power output, decouple to neighboring GND
FILT	8	A	Optional synthesizer filter
L2	9	A	Optional synthesizer inductor
L1	10	A	Optional synthesizer inductor
DATA	11	I/O	In wire mode: Data in-out/output Can be programmed to be used as a general purpose I/O pin Selectable internal 65 kΩ pull-up resistor
DCLK	12	I/O	In wire mode: Clock output Can be programmed to be used as a general purpose I/O pin Selectable internal 65 kΩ pull-up resistor
SYSCLK	13	I/O	Default functionality: Crystal oscillator (or divided) clock output Can be programmed to be used as a general purpose I/O pin Selectable internal 65 kΩ pull-up resistor
SEL	14	I	Serial peripheral interface select
CLK	15	I	Serial peripheral interface clock
MISO	16	O	Serial peripheral interface data output
MOSI	17	I	Serial peripheral interface data input
NC	18	N	Must be left unconnected

Symbol	Pin(s)	Type	Description
IRQ	19	O	Default functionality: Transmit and receive interrupt Can be programmed to be used as a general purpose I/O pin Selectable internal 65 kΩ pull-up resistor
PWRAMP	20	I/O	Default functionality: Power amplifier control output Can be programmed to be used as a general purpose I/O pin Selectable internal 65 kΩ pull-up resistor
ANTSEL	21	I/O	Default functionality: Diversity antenna selection output Can be programmed to be used as a general purpose I/O pin Selectable internal 65 kΩ pull-up resistor
NC	22	N	Must be left unconnected
VDD_IO	23	P	Power supply 1.8V - 3.6V
NC	24	N	Must be left unconnected
GPADC1	25	A	GPADC input
GPADC2	26	A	GPADC input
CLK16N	27	A	Crystal oscillator input/output
CLK16P	28	A	Crystal oscillator input/output
GND	Center Pad	P	Ground on center pad of QFN, must be connected

A = analog signal

I = digital input signal

O = digital output signal

I/O = digital input/output signal

N = not to be connected

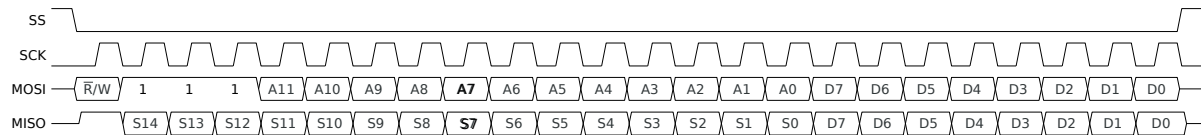
P = power or ground

All digital inputs are Schmitt trigger inputs, digital input and output levels are LVCMOS/LVTTL compatible and 5V tolerant.

### 1.3. SPI REGISTER ACCESS

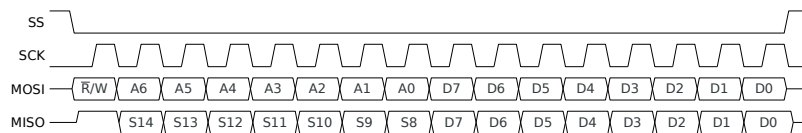
Registers are accessed via a synchronous Serial Peripheral Interface (SPI). Most Registers are 8 bits wide and accessed using the waveforms as detailed in Figure 3. These waveforms are compatible to most hardware SPI master controllers, and can easily be generated in

software. MISO changes on the falling edge of CLK, while MOSI is latched on the rising edge of CLK.



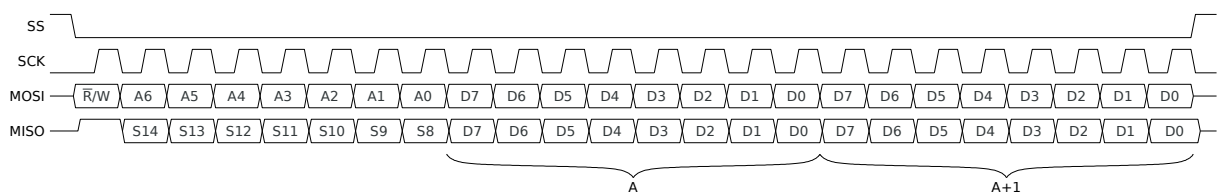
**Figure 3: SPI 8bit Long Address Read/Write Access**

The most important registers are at the beginning of the address space, i.e. at addresses less than 0x70. These registers can be accessed more efficiently using the short address form, which is detailed in Figure 4.



**Figure 4: SPI 8bit Read/Write Access**

Some registers are longer than 8 bits. These registers can be accessed more quickly than by reading and writing individual 8 bit parts. This is illustrated in Figure 5. Accesses are not limited by 16 bits either, reading and writing data bytes can be continued as long as desired. After each byte, the address counter is incremented by one. Also, this access form also works with long addresses.



**Figure 5: SPI 16bit Read/Write Access**

During the address phase of the access, the chip outputs the most important status bits. This feature is designed to speed up software decision on what to do in an interrupt handler. The table below shows which register bit is transmitted during the status timeslots.

SPI Bit Cell	Status	Register Bit
0	–	1 (when transitioning out of deep sleep, this bit transitions from 0→1 when the power becomes ready)
1	S14	PLL LOCK
2	S13	FIFO OVER
3	S12	FIFO UNDER
4	S11	THRESHOLD FREE ( <a href="#">FIFO free</a> > <a href="#">FIFO threshold</a> )
5	S10	THRESHOLD COUNT ( <a href="#">FIFO count</a> > <a href="#">FIFO threshold</a> )
6	S9	FIFO FULL
7	S8	FIFO EMPTY
8	S7	PWRGOOD (not BROWNOUT)
9	S6	PWR INTERRUPT PENDING
10	S5	RADIO EVENT PENDING
11	S4	XTAL OSCILLATOR RUNNING
12	S3	WAKEUP INTERRUPT PENDING
13	S2	LPOSC INTERRUPT PENDING
14	S1	GPADC INTERRUPT PENDING
15	S0	undefined

Note that bit cells 8-15 (S7...S0) are only available in two address byte SPI access formats.

### 1.3.1. DEEP SLEEP

The chip can be programmed into deep sleep mode. In deep sleep mode, the chip is completely switched off, which results in very low leakage power. All registers lose their programming.

To enter deep sleep mode, write the deep sleep encoding into bits 3:0 of [PWRMODE](#). At the rising edge of the SEL line, the chip will enter deep sleep mode.

To exit deep sleep mode, lower the SEL line. This will initiate startup and reset of the chip. Then poll the MISO line. The MISO line will be held low during initialization, and will rise to high at the end of the initialization, when the chip becomes ready for further operation.

### 1.3.2. ADDRESS SPACE

The address space has been allocated as follows. Addresses from 0x000 to 0x06F are reserved for “dynamic registers”, i.e. registers that are expected to be frequently accessed during normal operation, as they can be efficiently accessed using single address byte SPI

accesses. Addresses from 0x070 to 0x0FF have been left unused (they could only be accessed using the two address byte SPI format). Addresses from 0x100 to 0x1FF have been reserved for physical layer parameter registers, for example receiver, transmitter, PLL, crystal oscillator. Addresses from 0x200 to 0x2FF have been reserved for medium access parameters, such as framing, packet handling. Addresses from 0x300 to 0x3FF have been reserved for special functions, such as GPADC.

## 2. FIFO OPERATION

The **AX5043** features a 256 Byte FIFO. The same FIFO is used for both reception and transmission. During transmit, only the write port is accessible by the microcontroller. During receive, only the read port is accessible by the microcontroller. Otherwise, both ports are accessible through the register file.

In order to prevent transmitting premature data, the FIFO contains three pointers. Data is read at the read pointer, up to the write pointer. Data is written to the write ahead pointer. The write pointer is not updated when data is written, therefore, new data is not immediately visible to the consumer. Writing the COMMIT command to the [FIFOSTAT](#) register copies the write ahead pointer to the write pointer, thus making the written data visible to the receiver. Writing the ROLLBACK command to the [FIFOSTAT](#) register sets the write ahead pointer to the write pointer, thus discarding data written to the FIFO. During transmit, this means that the transmitter will only consider data written to the FIFO after the commit command. During receive, this feature is used by the receiver to store packet data before it is known whether the CRC check passes. [FIFOCOUNT](#) reports the number of bytes that can be read without causing an underflow. [FIFOFREE](#) reports the number of bytes that can be written without causing an overflow. [FIFOCOUNT](#) and [FIFOFREE](#) do not add up to 256 Bytes whenever there are uncommitted bytes in the FIFO. Figure 6 Illustrates this.

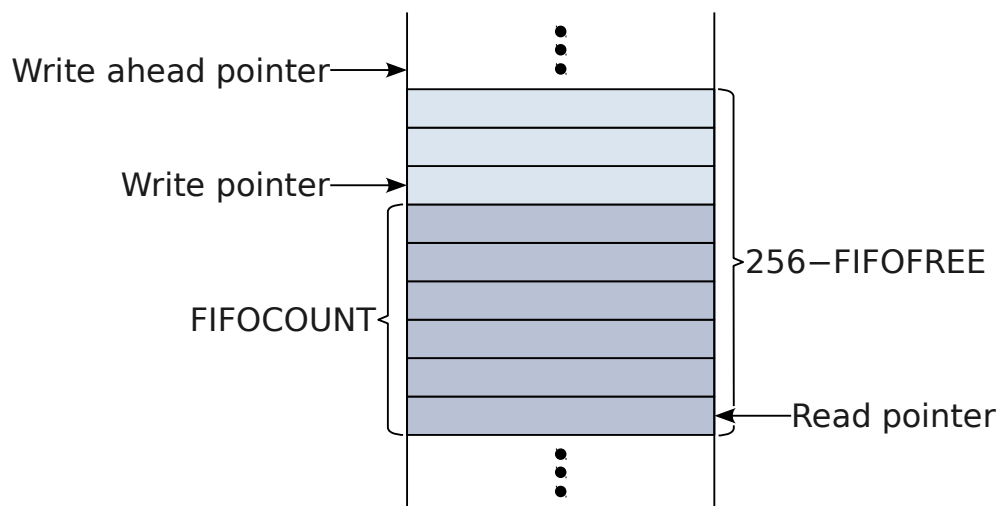
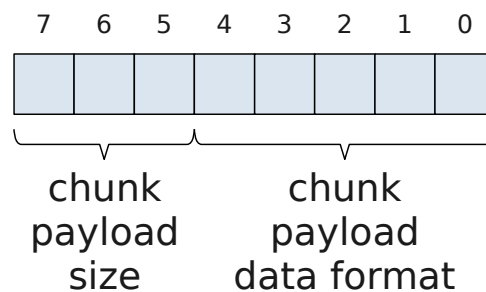


Figure 6: FIFO Pointer

## 2.1. FIFO CHUNK ENCODING

In order to distinguish meta-data (such as RSSI) from receive or transmit data, FIFO contents are organized as chunks. Chunks consist of a header that encodes the chunk length as well as the payload data format.

Each chunk starts with a single byte header. The header encodes the length of a chunk, and indicates the data it contains. The top 3 bits encode the length (or optionally refer to an additional length byte after the header byte), and the bottom 5 bits indicate what payload data the chunk contains. The following table lists the encoding of the length bits (top 3 bits of the first chunk header byte). Figure 7 Shows the chunk header byte encoding.



**Figure 7: FIFO Header Byte Format**

The following table lists the chunk payload size encoding:

Top Bits	Chunk Payload Size
000	No payload
001	Single byte payload
010	Two byte payload
011	Three byte payload
100	Invalid
101	Invalid
110	Invalid
111	Variable length payload; payload size is encoded in the following length byte the length byte is part of the header (and not included in length), everything after the length byte is included in the length

The following table lists the chunk types and their encodings. The Hdr Byte column lists the complete FIFO Chunk Header Byte, consisting of the length and data format encodings.

Name	Dir	Hdr Byte	Description
		7-0	
No payload commands			
<a href="#">NOP</a>	T	00000000	No Operation
One byte payload commands			
<a href="#">RSSI</a>	R	00110001	RSSI
<a href="#">TXCTRL</a>	T	00111100	Transmit Control (Antenna, Power Amp)
Two byte payload commands			
<a href="#">FREQOFFS</a>	R	01010010	Frequency Offset
<a href="#">ANTRSSI2</a>	R	01010101	Background Noise Calculation RSSI
Three byte payload commands			
<a href="#">REPEATDATA</a>	T	01100010	Repeat Data
<a href="#">TIMER</a>	R	01110000	Timer
<a href="#">RFFREQOFFS</a>	R	01110011	RF Frequency Offset
<a href="#">DATARATE</a>	R	01110100	Datarate
<a href="#">ANTRSSI3</a>	R	01110101	Antenna Selection RSSI
Variable length payload commands			
<a href="#">DATA</a>	TR	11100001	Data
<a href="#">TXPWR</a>	T	11111101	Transmit Power

Direction: T = Transmit, R = Receive

#### 2.1.1. NOP COMMAND

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

The NOP command will be discarded without effect by the transmitter. The receiver will not generate NOP commands.

#### 2.1.2. RSSI COMMAND

7	6	5	4	3	2	1	0
0	0	1	1	0	0	0	1
RSSI							



The RSSI command will only be generated by the receiver at the end of a packet if bit STRSSI is set in register [PKTSTOREFLAGS](#). The encoding is the same as that of the [RSSI](#) register.

### 2.1.3. TXCTRL COMMAND

7	6	5	4	3	2	1	0
0	0	1	1	1	1	0	0
0	SETTX	TXSE	TXDIFF	SETANT	ANTSTATE	SETPA	PASTATE

The TXCTRL command allows certain aspects of the transmitter to be changed on the fly. If SETTX is set, TXSE and TXDIFF are copied into the register [MODCFG4](#). If SETANT is set, ANTSTATE is copied into register [DIVERSITY](#). If SETPA is set, PASTATE is copied into register [PWRAMP](#).

### 2.1.4. FREQOFFS COMMAND

7	6	5	4	3	2	1	0
0	1	0	1	0	0	1	0
FREQOFFS1							
FREQOFFS0							

The FREQOFFS command will only be generated by the receiver at the end of a packet if bit STFOFFS is set in register [PKTSTOREFLAGS](#). The encoding is the same as that of the [TRKFREQ](#) register.

### 2.1.5. ANTRSSI2 COMMAND

7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	1
RSSI							
BGNDNOISE							

The ANTRSSI2 command will be generated by the receiver when it is idle if bit STANTRSSI is set in register [PKTSTOREFLAGS](#). If DIVENA is set in register [DIVERSITY](#), the [ANTRSSI3](#) command is generated instead. The encoding of the RSSI field is the same as that of the [RSSI](#) register. The BGNDNOISE field contains an estimate of the background noise.

## 2.1.6. REPEATDATA COMMAND

7	6	5	4	3	2	1	0
0	1	1	0	0	0	1	0
0	0	UNENC	RAW	NOCRC	RESIDUE	PKTEND	PKTSTART
REPEATCNT							
DATA							

The REPEATDATA command allows the efficient transmission of repetitive data bytes. The DATA byte given in the payload is repeated REPEATCNT times. See [DATA](#) command for a description of the flag byte. This command is especially handy for constructing preambles.

## 2.1.7. TIMER COMMAND

7	6	5	4	3	2	1	0
0	1	1	1	0	0	0	0
TIMER2							
TIMER1							
TIMER0							

The TIMER command will only be generated by the receiver at the start of a packet if bit STTIMER is set in register [PKTSTOREFLAGS](#). The payload is a copy of the  $\mu$ s timer [TIMER](#) register. This command enables exact packet timing for example for frequency hopping systems.

## 2.1.8. RFFREQOFFS COMMAND

7	6	5	4	3	2	1	0
0	1	1	1	0	0	1	1
RFFREQOFFS2							
RFFREQOFFS1							
RFFREQOFFS0							

The RFFREQOFFS command will only be generated by the receiver at the end of a packet if bit STRFOFFS is set in register [PKTSTOREFLAGS](#). The encoding is the same as that of the [TRKRFFREQ](#) register.

### 2.1.9. DATARATE COMMAND

7	6	5	4	3	2	1	0
0	1	1	1	0	1	0	0
DATARATE2							
DATARATE1							
DATARATE0							

The DATARATE command will only be generated by the receiver at the end of a packet if bit STDR is set in register [PKTSTOREFLAGS](#). The encoding is the same as that of the [TRKDATARATE](#) register.

### 2.1.10. ANTRSSI3 COMMAND

7	6	5	4	3	2	1	0
0	1	1	1	0	1	0	1
ANT0RSSI							
ANT1RSSI							
BGNDNOISE							

The ANTRSSI3 command will be generated by the receiver when it is idle if bit STANTRSSI is set in register [PKTSTOREFLAGS](#). If DIVENA is not set in register [DIVERSITY](#), the [ANTRSSI2](#) command is generated instead. The encoding of the ANT0RSSI and ANT1RSSI fields are the same as that of the [RSSI](#) register. The BGNDNOISE field contains an estimate of the background noise.

### 2.1.11. DATA COMMAND

The DATA command transports actual transmit and receive data. While the basic format is the same for transmit and receive, the semantics of the flag byte differs.

#### 2.1.11.1. TRANSMIT DATA FORMAT

7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	1
LENGTH							
0	0	UNENC	RAW	NOCRC	RESIDUE	PKTEND	PKTSTART
DATA							
⋮							

LENGTH includes the flags byte as well as all DATA bytes.

Setting RAW to one causes the DATA to bypass the framing mode, but still pass through the encoder.

Setting UNENC to one causes the DATA to bypass the framing mode, as well as the encoder, except for inversion. UNENC has priority over RAW.

Setting NOCRC suppresses the generation of the CRC bytes.

Setting RESIDUE allows the transmission of a number of data bits that is not a multiple of eight. All but the last data byte are transmitted as if RESIDUE was not set. The last byte however contains only 7 bits or less. The transmitter looks for the highest bit set. This is considered the stop bit. Only bits below the stop bit are transmitted. If the MSBFIRST in register [PKTADDRCFG](#) is set, the algorithm is reversed, i.e. the lowest bit set is considered the stop bit and bits above the stop bit are transmitted.

PKTSTART and PKTEND bits enable the transmission of packets that are larger than the FIFO size. If PKTSTART is set, the radio packet starts at the beginning of the DATA command payload. If PKTEND is set, the radio packet ends at the end of the DATA command payload. If PKTSTART is not set, this command is the continuation of a previous DATA command. If PKTEND is not set, the packet is continued with the next DATA command.

PKTSTART in RAW mode causes the DATA bytes to be aligned to DiBit boundaries in 4-FSK mode.

For example, to transmit 20 bits of an alternating 0-1 pattern as a preamble, the following bytes should be written to the FIFO (MSBFIRST=0 in register [PKTADDRCFG](#) is assumed):

0xE1	FIFO Command
0x04	Length Byte
0x24	Flag Byte: Unencoded, to ensure 0-1 remains 0-1, and Residue set, because the number of bits transmitted is not a multiple of 8
0xAA	Alternating 0-1 bits
0xAA	Alternating 0-1 bits
0x1A	Alternating 0-1 bits; Bit 4 is the "Stop" bit

#### 2.1.11.2. RECEIVE DATA FORMAT

7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	1
LENGTH							
0	ABORT	SIZEFAIL	ADDRFAIL	CRCFAIL	RESIDUE	PKTEND	PKTSTART
DATA							
:							

ABORT is set if the packet has been aborted. An ABORT sequence is a sequence of seven or more consecutive one bits when HDLC [\[1\]](#) framing is used. Note that if ACCPTABRT is not set in register [PKTACCEPTFLAGS](#), then aborted packets are silently dropped.

SIZEFAIL is set if the packet does not pass the size checks. Size checks are implemented using the [PKTLENCFG](#), [PKTLENOFFSET](#) and [PKTMAXLEN](#) registers. Note that if ACCPTSZF is not set in register [PKTACCEPTFLAGS](#), then packets with an invalid size are silently dropped.

ADDRFAIL is set if the packet does not pass the address checks. Address checks are implemented using the [PKTADDRCFG](#), [PKTADDR](#) and [PKTADDRMASK](#) registers. Note that if ACCEPTADDRF is not set in register [PKTACCEPTFLAGS](#), then packets which do not match the programmed address are silently dropped.

CRCFAIL is set if the packet does not pass the CRC check. Note that if ACCEPTCRCF is not set in register [PKTACCEPTFLAGS](#), then packets which fail the CRC check are silently dropped.

RESIDUE, PKTEND and PKTSTART work identical as in transmit mode, see above.

The receiver generates chunks up to [PKTCHUNKSIZE](#) bytes. If [PKTMAXLEN](#) is larger than [PKTCHUNKSIZE](#), multiple chunks may be generated for one packet. Since CRC and size checks may only be performed at the end of the packet, only the last chunk can be dropped at failure of one of those tests. It is therefore important that the microcontroller receiver routine clears its receive buffer at the beginning of DATA commands whose PKTSTART bit is set, as the buffer may still contain bytes from erroneous packets.

## 2.1.12. TXPWR COMMAND

7	6	5	4	3	2	1	0
1	1	1	1	0	0	1	0
LENGTH = 10							
TXPWRCOEFFA(7:0)							
TXPWRCOEFFA(15:8)							
TXPWRCOEFFB(7:0)							
TXPWRCOEFFB(15:8)							
TXPWRCOEFFC(7:0)							
TXPWRCOEFFC(15:8)							
TXPWRCOEFFD(7:0)							
TXPWRCOEFFD(15:8)							
TXPWRCOEFFE(7:0)							
TXPWRCOEFFE(15:8)							

The TXPWR command allows the transmit power to be changed on the fly. This command updates the [TXPWRCOEFFA](#), [TXPWRCOEFFB](#), [TXPWRCOEFFC](#), [TXPWRCOEFFD](#) and [TXPWRCOEFFE](#) registers.

### 3. PROGRAMMING THE CHIP

#### 3.1. POWER MODES

To enable the lowest possible application power consumption, the **AX5043** allows to shut down its circuits when not needed. This is controlled by the [PWRMODE](#) register. Idd values are typical; for exact values, please refer to the **AX5043** datasheet [\[2\]](#).

PWRMODE register	Name	Description	Typical Idd
0000	POWERDOWN	Powerdown; all circuits powered down except for the register file	400nA
0001	DEEPSLEEP	Deep Sleep Mode; Chip is fully powered down until SEL is lowered again; loses all register contents	50nA
0101	STANDBY	Crystal Oscillator enabled	230µA
0111	FIFOON	FIFO and Crystal Oscillator enabled	310µA
1000	SYNTHRX	Synthesizer running, Receive Mode	5mA
1001	FULLRX	Receiver Running	7–11mA
1011	WORRX	Receiver Wake-on-Radio Mode	500nA
1100	SYNTHTX	Synthesizer running, Transmit Mode	5mA
1101	FULLTX	Transmitter Running	6–70mA

The following list explains the typical programming flow.

Preparation:

1. Reset the Chip. Set SEL to high for at least 1µs, then low. Wait until MISO goes high. Set, and then clear, the RST bit of register [PWRMODE](#).
2. Set the [PWRMODE](#) register to POWERDOWN.
3. Program parameters. It is recommended that suitable parameters are calculated using the AX\_RadioLab tool available from Axsem.

4. Perform auto-ranging, to ensure the correct VCO range setting.

The chip is now ready for transmit and receive operations.

### 3.1.1. FIFO POWER MANAGEMENT

The FIFO is powered down during POWERDOWN and DEEPSLEEP modes (Register [PWRMODE](#)). The FIFO EMPTY and FIFO FULL bits (Register [FIFOSTAT](#)), as well as the [FIFOCOUNT](#) and [FIFOFREE](#) registers read zero. Reads from the FIFO will return undefined data, and writes to the FIFO will be lost.

In the receive case, the FIFO is automatically powered on when the chip [PWRMODE](#) is set to FULLRX. The FIFO should be emptied before the [PWRMODE](#) is set to POWERDOWN. In Wake-on-radio or POWERDOWN mode, the FIFO is automatically kept powered until it is emptied by the microprocessor.

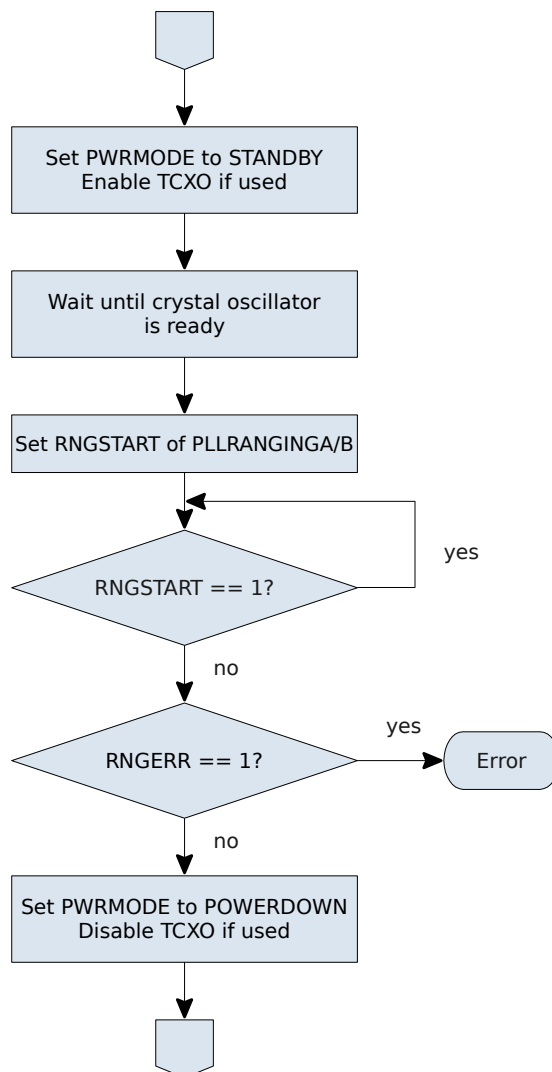
In the transmit case, [PWRMODE](#) should first be set to FULLTX. *Before writing to the FIFO, the microprocessor must ensure that the SVMODEM bit is high in Register [POWSTAT](#)*, to ensure that the on-chip voltage regulator supplying the FIFO has finished starting up. The transmitter remains idle until the contents of the FIFO are committed (unless the FIFO AUTO COMMIT bit is set in Register [FIFOSTAT](#)).

## 3.2. AUTORANGING

Whenever the frequency changes, the synthesizer VCO should be set to the correct range using the built-in auto-ranging. A re-ranging of the VCO is required if the frequency change required is larger than 5 MHz in the 868/915 MHz band or 2.5 MHz in the 433 MHz band. Each individual chip must be auto-ranged. If both frequency register sets [FREQA](#) and [FREQB](#) are used, then both frequencies must be auto-ranged by first starting auto-ranging in [PLLRANGINGA](#), waiting for its completion, followed by starting auto-ranging in [PLLRANGINGB](#) and waiting for its completion.



Figure 8 shows the flow chart of the auto-ranging process.



**Figure 8: Autoranging Flow Chart**

Before starting the auto-ranging, the appropriate frequency registers ([FREQA3](#), [FREQA2](#), [FREQA1](#) and [FREQA0](#) or [FREQB3](#), [FREQB2](#), [FREQB1](#) and [FREQB0](#)) need to be programmed. Auto-ranging starts at the VCOR (register [PLLRANGINGA](#) or [PLLRANGINGB](#)) setting; if you already know the approximately correct synthesizer VCO range, you should set VCORA/VCORB to this value prior to starting auto-ranging; this can speed up the ranging process considerably. If you have no prior knowledge about the correct range, set VCORA/VCORB to 8. Starting with VCORA/VCORB < 6 should be avoided, as the initial synthesizer frequency can exceed the maximum frequency specification.

Hardware clears the RNG START bit automatically as soon as the ranging is finished; the device may be programmed to deliver an interrupt on resetting of the RNG START bit.

Waiting until auto-ranging terminates can be performed by either polling the register [PLLRANGINGA](#) or [PLLRANGINGB](#) for RNG START to go low, or by enabling the IRQMPLLRNGDONE interrupt in register [IRQMASK1](#).

### 3.3. CHOOSING THE FUNDAMENTAL COMMUNICATION CHARACTERISTICS

The following table lists the fundamental communication characteristics that need to be chosen before the device can be programmed.

Parameter	Description
$f_{XTAL}$	Frequency of the connected crystal in Hz
modulation	FSK, MSK, OQPSK, 4-FSK or AFSK (for recommendations see below)
$f_{CARRIER}$	Carrier frequency (i.e. center frequency of the signal) in Hz
BITRATE	Desired bit rate in bit/s
h	Modulation index, determines the frequency deviation for FSK $32 > h \geq 0.5$ for FSK, 4-FSK or AFSK, $f_{deviation} = 0.5 * h * BITRATE$ $h = 0.5$ for MSK and OQPSK (For AFSK, $f_{deviation}$ is usually set according to the FM channel specification. For 25kHz channels, it is often approximately 3kHz)
encoding	Inversion, differential, manchester, scrambled, for recommendations see the description of the register <a href="#">ENCODING</a> .

The following table gives an overview of the trade-offs between the different modulations that AX5043 offers, they should be considered when making a choice.

Modulation	Trade-offs
FSK	For bit rates up to 125 kbit/s Frequency deviation is a free parameter

Modulation	Trade-offs
MSK	<p>For bit rates up to 125 kbit/s</p> <p>Robust and spectrally efficient form of FSK (Modulation is the same as FSK with <math>h=0.5</math>)</p> <p>Frequency deviation given by bit rate</p> <p>The advantage of MSK over FSK is that it can be demodulated with higher sensitivity.</p> <p>Slightly longer preambles required than for FSK</p>
OQPSK	<p>For bit rates up to 125 kbit/s</p> <p>Very similar to MSK, with added precoding / postdecoding</p> <p>For new designs, use MSK instead</p>
PSK	<p>For bit rates up to 125 kBit/s</p> <p>Spectrally efficient and high sensitivity</p> <p>Very accurate frequency reference (maximum carrier frequency deviation <math>\pm \frac{1}{4} \cdot \text{BITRATE}</math> ) and long preambles required</p>
4-FSK	<p>For bit rates up to 100 kSymbols/s, or 200 kbit/s</p> <p>Similar to FSK, but four frequencies are used to transmit 2 bits simultaneously</p> <p>Very slightly more spectrally efficient compared to FSK ( <math>\frac{1+3 \cdot h}{2} \cdot \text{BITRATE}</math> versus <math>(1+h) \cdot \text{BITRATE}</math> ) for small <math>h</math>.</p> <p>Longer preambles required as frequency offset estimation needs to be more precise to successfully demodulate</p> <p>For new designs, use FSK instead</p>
AFSK	<p>For bit rates up to 25 kbit/s</p> <p>Bits are FSK modulated in the audio band, then frequency modulated on the carrier frequency.</p> <p>For legacy compatibility applications only.</p>

Given these fundamental physical layer parameters, AX\_RadioLab should be used to compute the register settings of the **AX5043**.

### 3.4. FRAMING

Figure 1 shows the block diagram of the **AX5043**. After the user writes a transmit packet into the FIFO, the Radio Controller sequences the transmitter start-up, and signals the Packet Controller to read the packet from the FIFO and add framing bits, allowing the receiver to lock to the transmit waveform, and to detect packet and byte boundaries. If MSB

first is selected (register [PKTADDRCFG](#)), then the bits within each byte are swapped when the data is read out from the FIFO.

The Packet Controller also (optionally) adds cyclic redundancy check bits at the end of the packet, to enable the receiver to detect transmission errors. Both 16 and 32 Bit CRC can be selected, as well as different generator polynomials. The CRC polynomial can be selected in register [FRAMING](#). The following polynomials are supported:

- CRC-CCITT (16bit):  $x^{16} + x^{12} + x^5 + 1$  (hexadecimal: 0x1021)
- CRC-16 (16bit):  $x^{16} + x^{15} + x^2 + 1$  (hexadecimal: 0x8005)
- CRC-DNP (16bit):  $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$  (hexadecimal: 0x3D65)  
This polynomial is used for Wireless M-Bus.
- CRC-32 (32bit):  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$   
(hexadecimal: 0x04C11DB7)

The CRC is always transmitted MSB first regardless of the MSB first setting of register [PKTADDRCFG](#), to enable the receiver to process CRC bits as they arrive (otherwise, they would have to be stored and reordered). For an in-depth guide on how CRC's are computed, see [3].

Finally, the encoder is able to perform certain bit-wise operations on the bit-stream:

- Manchester: Manchester transmits a one bit as 10 and a zero bit as 01, i.e. it doubles the data rate on the radio channel. Its advantage is that the resulting bit-stream has many transitions and thus simplifies synchronizing to the transmission on the receiver side. The downside is that it now requires twice the amount of energy for the transmission. Manchester is not recommended, except for compatibility with legacy systems.
- Scrambler: The scrambler ensures that even highly regular transmit data results in a seemingly random transmitted bit-stream. This avoids discrete tones in the spectrum. Do not confuse the scrambler with encryption – it does not provide any secrecy, its actions are easily reversed. Its use is recommended.
- Differential: Differential transmits zero bits as constant level, and one bits as level change. This allows to accomodate modulations that can invert the bit-stream, such as PSK. It is available for compatibility with other Axsem transceivers, but usually not used on the **AX5043**.

- Inversion: If on, the bit-stream is inverted. Useful for example for compatibility with legacy systems, such as POCSAG, which differ from the usual convention that the higher FSK frequency signifies a one.

The encoder is controlled using the register [ENCODING](#). It may be temporarily bypassed *except for the inversion* by setting the UNENC bit of the FIFO chunks [DATA](#) or [REPEATDATA](#). This is useful for synthesizing preambles.

The receiver performs these tasks in reverse order.

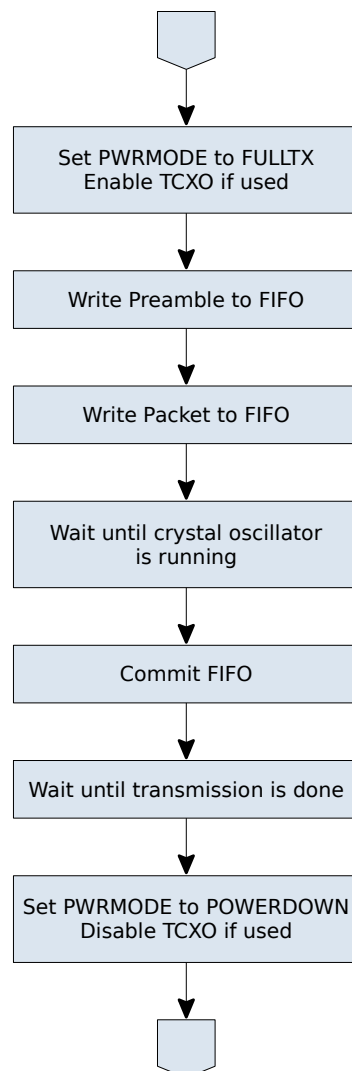
### 3.5. TRANSMITTER

Figure 9 shows the transmitter flow chart. The microprocessor first places the chip into FULLTX mode. This prepares the chip for a future transmission, enables the FIFO in transmit direction, but does not yet power-up the synthesizer or any other transmit circuitry.

The microprocessor can now write the preamble and the actual packet to the FIFO. The preamble is programmable to allow standards to be implemented that specify a specific preamble to be used. Otherwise, the recommendations for preambles can be found below.

Waiting for the crystal oscillator to start up may be performed by polling the register [XTALSTATUS](#), or by enabling the IRQMXTALREADY interrupt in register [IRQMASK1](#).

After the FIFO contents are committed (writing the Commit command to the [FIFOSTAT](#) register), the transmitter notices that the FIFO is no longer empty. It then powers up the synthesizer and settles it (registers [TMGTXBOOST](#) and [TMGTXSETTLE](#) determine the timing). The Preamble and the Packet(s) are then transmitted, followed by the transmitter and synthesizer shut-down.



**Figure 9: Transmitter Flowchart**

The transmitter is automatically ramped up and down smoothly, to prevent unwanted spurious emissions. The ramp time is normally one bit time, but may be longer by changing the SLOWRAMP field of register [MODCFG0](#).

The [PWRMODE](#) register should stay at FULLTX until the transmission is fully completed. The end of the transmission may be determined by polling the register [RADIOSTATE](#) until it indicates idle, or by enabling the radio controller interrupt (bit IRQMRADIOCTRL) in register [IRQMASK0](#) and setting the radio controller to signal an interrupt at the end of transmission (bit REVM DONE of register [RADIOEVENTMASK0](#)).

### 3.5.1. RECOMMENDED PREAMBLE

The main purpose of the preamble is to allow for the receiver to acquire vital transmission parameters before the actual packet data starts. The minimum duration of the preamble is dependent on how much time the receiver needs to acquire these parameters to sufficient precision. More specifically, it depends on:

- the time needed for the receiver adaptive gain control (AGC) to acquire the signal strength
- the time needed for the receiver to acquire the maximum possible frequency offset (registers [MAXROFFSET0](#), [MAXROFFSET1](#) and [MAXROFFSET2](#))
- the time needed for the receiver to acquire the maximum possible data rate offset (registers [MAXDROFFSET0](#), [MAXDROFFSET1](#) and [MAXDROFFSET2](#))
- the time needed for the receiver to acquire the exact bit sampling time (registers [TIMEGAIN0](#), [TIMEGAIN1](#), [TIMEGAIN2](#) and [TIMEGAIN3](#))
- the time needed to acquire the actual frequency deviation in 4-FSK mode (registers [FSKDMAX0](#), [FSKDMAX1](#), [FSKDMIN0](#) and [FSKDMAX0](#))

On the **AX5043**, these loops run in parallel. An AGC that is significantly off however causes the received signal to fall outside the IF strip dynamic range, and thus prevents the other loops from working. And a frequency offset that is compensated insufficiently causes the received signal to fall (partially) outside the IF filter, thus also preventing the timing and 4-FSK loops from working.

The minimum possible preamble duration can be achieved under the following conditions:

- Use a transmitter with a sufficiently precise bit timing. If the maximum deviation of the transmitter data rate from the receiver data rate is less than approximately 0.1%, then the data rate acquisition loop should be switched off completely (setting registers [MAXDROFFSET0](#), [MAXDROFFSET1](#) and [MAXDROFFSET2](#) to zero). The **AX5043** is able to track the remaining small offset without the data rate offset loop. All Axsem transmitters derive the bit rate timing from the crystal reference and can therefore easily meet this requirement.
- Use an FSK frequency deviation that is larger than the maximum frequency offset between transmitter and receiver. In this case, receiver frequency offset acquisition is not needed. Do not use 4-FSK.

- Use the **AX5043** receiver parameter set feature, below.

Finally, the frame synchronization word achieves byte synchronization.

The recommended preamble bit pattern is now discussed.

If the standard to be implemented requires a specific preamble, use it.

In **FEC** mode, HDLC [\[1\]](#) flags (pattern 01111110) must be transmitted. The convolutional encoder ensures enough bit transitions, and the **AX5043** receiver needs flags to synchronize its interleaver.

If the **scrambler** or **manchester** is enabled, send RAW bytes 00010001. The scrambler or manchester encoder ensure enough transitions to acquire the bit timing.

In **4-FSK** mode, send UNENCODED bytes 00010001. This ensures that the preamble toggles between the highest and the lowest frequency. The frequent transitions ensure the bit timing is acquired as quickly as possible, and the maximum and minimum frequencies allow the deviation to be acquired.

Otherwise, use UNENCODED 01010101. This preamble ensures the maximum number of transitions for bit timing synchronization. This preamble could also be used with the scrambler enabled; the main purpose of the scrambler is however to ensure no spectral lines (tones), this would be defeated by this preamble.

If MSBFIRST in register [PKTADDRCFG](#) is set, then the preamble sequences should be reversed.

### 3.6. RECEIVER

Figure 10 shows the receiver flow chart. When the microprocessor places the chip into FULLRX mode, the **AX5043** immediately powers up the synthesizer, settles it (registers [TMGRXBOOST](#) and [TMGRXSETTLE](#) determine the timing) and starts receiving. The reception continues until the microprocessor changes the [PWRMODE](#) register.



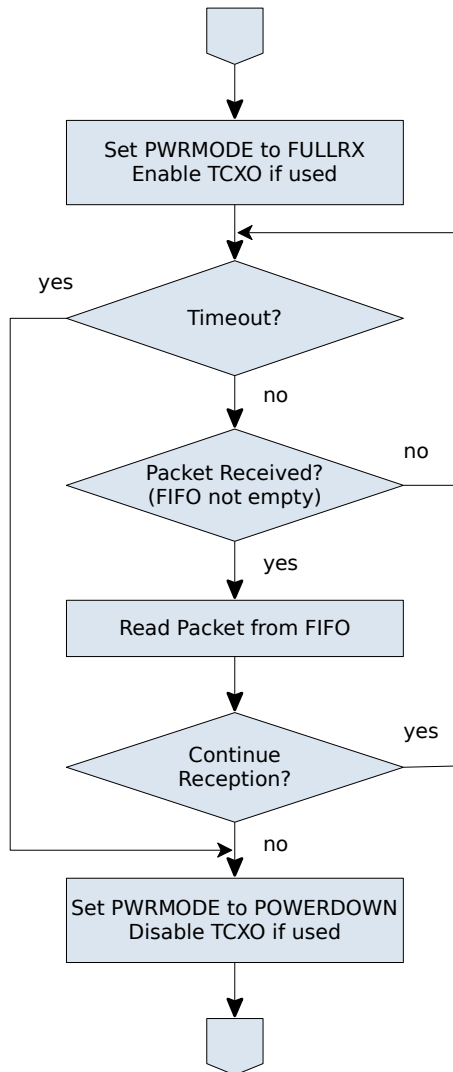


Figure 10: Receiver Flowchart

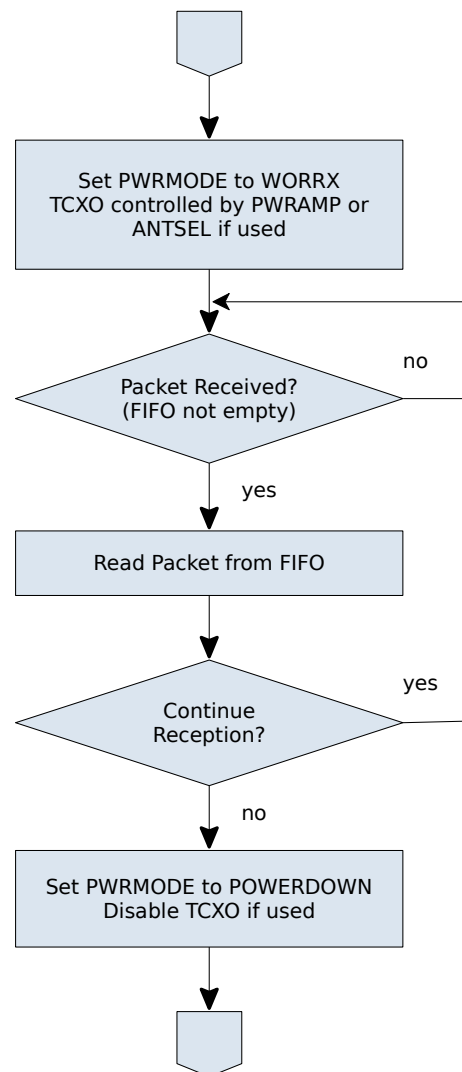


Figure 11: Wake-on-Radio Receiver Flowchart

If antenna diversity is enabled, the **AX5043** continuously switches between the antennas (controlled by the ANTSEL pin) to find the antenna with the better signal strength, until a valid preamble is detected. Antenna scanning is resumed after a packet is completed.

Actual packet data in the FIFO may be preceded and followed by meta-data. Meta-data may be a time stamp at the beginning of the packet, and signal strength, frequency offset and data rate offset at the end of the packet. Which meta-data is written to the FIFO is controlled by the register [PKTSTOREFLAGS](#).

Wake-on-Radio mode allows the **AX5043** to periodically poll the radio channel for a transmission while using only very little power. Figure 11 shows the wake-on-radio flow

chart. The **AX5043** periodically wakes up. The wake-up is controlled by the on-chip low-power 640Hz/10kHz RC oscillator and the period is programmed using the [WAKEUPFREQ1](#) and [WAKEUPFREQ0](#) registers.

After waking up, the **AX5043** quickly settles the AGC and computes the channel RSSI. If it is below an absolute threshold (register [RSSIABSTHR](#)) and a dynamic threshold (register [BGNDRSSITHR](#)), it is switched off immediately. Otherwise, it looks for a valid preamble. If none is found within a preprogrammed time (registers [TMGRXPREAMBLE1](#) and [TMGRXPREAMBLE2](#)), the receiver is powered down. Otherwise, it continues to receive the packet.

If a packet is successfully received, the receiver may either be shut down again, or continue to run if WORMULTIPKT is set in register [PKTMISCFLAGS](#).

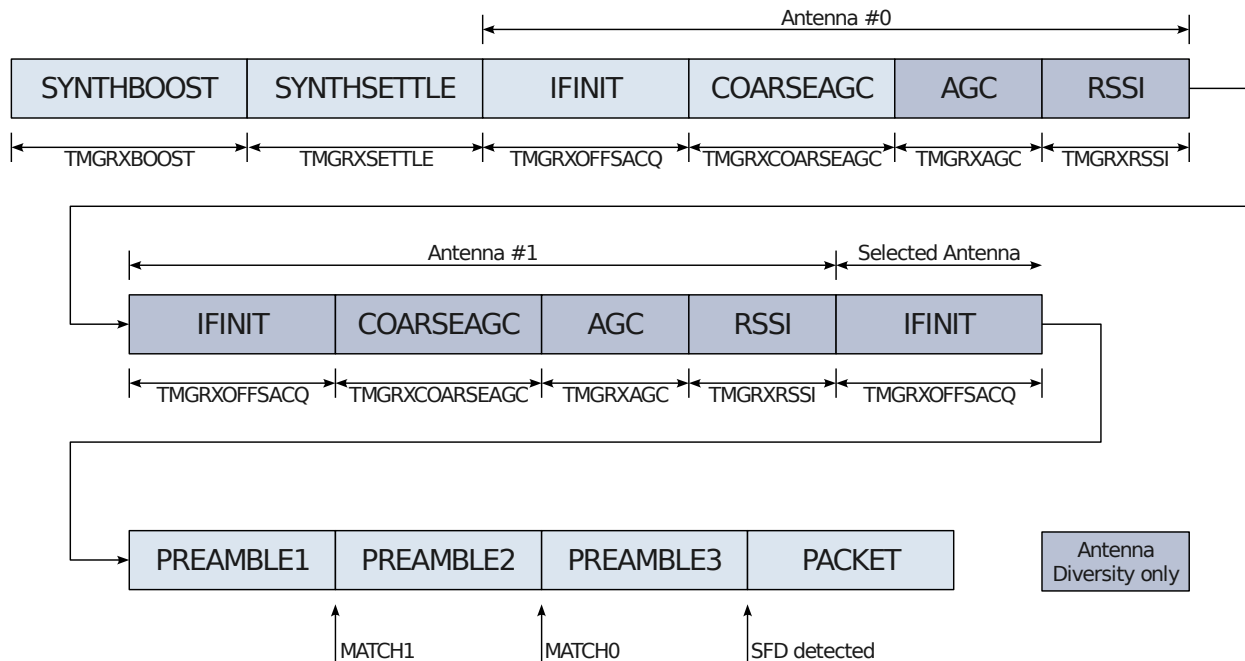
In Wake-on-Radio mode, the **AX5043** is completely autonomous until a packet is received. The microprocessor may be shut down and only wake up once the FIFO is no longer empty (IRQMFIFONOTEMPTY interrupt in register [IRQMASK0](#)).

### 3.6.1. RECEIVER STATE MACHINE

Figure 12 shows the receiver timing diagram. The actions in the first two lines are time controlled. The arrows below indicate which register controls the timing. The actions colored in a darker shade of blue are only performed when diversity mode is enabled (DIVENA is set in register [DIVERSITY](#)). The actions in the last line are detailed in the state diagram Figure 13.

*SYNTHBOOST* and *SYNTHSETTLE* form the two stage procedure to settle the synthesizer on the first LO frequency. During *SYNTHBOOST*, the synthesizer is operated at a higher loop bandwidth (register [PLLLOOPBOOST](#)), while during *SYNTHSETTLE*, the final settling is done at the nominal, lower noise, loop bandwidth (register [PLLLOOP](#)).

*IFINIT* settles the IF strip. *COARSEAGC* uses a fast AGC time constant to quickly settle the AGC to a value close to the correct one. This is especially important during wake-on-radio, as it is desirable to keep the receiver powered the shortest possible time to save power. *AGC* settles the AGC using a slower time constant. *RSSI* measures the received signal strength. This value is then used to determine whether the receiver should be kept running in wake-on-radio, or to select the antenna with the stronger signal in diversity mode.

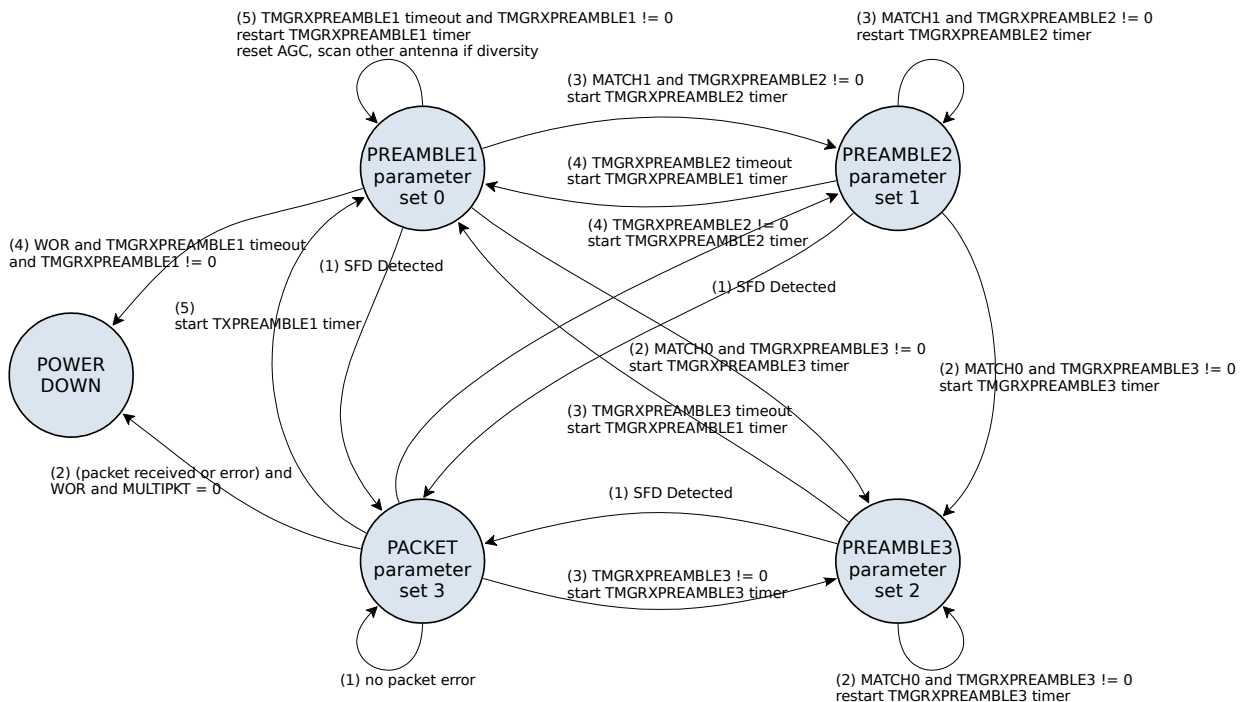


**Figure 12: Receiver Timing Diagram**

Once the receiver is initialized, *PREAMBLE1*, *PREAMBLE2*, *PREAMBLE3*, and *PACKET* coordinate the reception of packets. The receiver contains several loops that acquire and track transmission parameters the receiver needs to know in order to correctly receive a packet.

- The AGC acquires and tracks the signal strength
- The frequency tracking loop acquires and tracks the frequency offset
- The timing and data rate tracking loop acquires and tracks the sampling time and the data rate offset

The bandwidth of these loops is programmable. The bandwidth controls the acquisition time as well as the noisiness of the parameter estimates. In order to allow both fast acquisition to enable short preambles and low steady state noise performance to enable high receiver sensitivity, the receiver supports multiple acquisition and tracking loop parameter sets. When the receiver searches for a transmission signal, it uses wide loop bandwidths. Once it detects a preamble with sufficient probability, it switches to a lower loop bandwidth. Once a frame start is detected, it switches to an even lower loop bandwidth. Figure 13 shows the state diagram that controls which receiver parameter set is used.



**Figure 13: Receiver State Diagram**

Conditions are evaluated in priority order. The priority number is given in parentheses at the beginning of arrow labels.

In order to reduce the number of registers that need to be programmed if not all parameter sets are different, the parameter set number of Figure 13 is not directly used to address the parameter set. Instead, it indexes into register [RXPARAMSETS](#), where the actual parameter set number is read out.

### 3.7. LOW POWER OSCILLATOR CALIBRATION

The low power oscillator is used to control the wake-up frequency, or polling period, during wake-on-radio mode. In order to increase the precision of the wake-up frequency, calibration logic allows the low power oscillator to be calibrated against the crystal oscillator or TCXO.

Figure 14 shows a block diagram of the calibration logic. It works similarly to a PLL. The reference frequency from the crystal or TCXO is divided by the value of the [LPOSCREF](#) register. This signal is then compared to the actual frequency of the Low Power Oscillator. The frequency difference is then low pass filtered ([LPOSCFILT](#) register) and used to adjust the Low Power Oscillator frequency ([LPOSCFREQ](#) register).

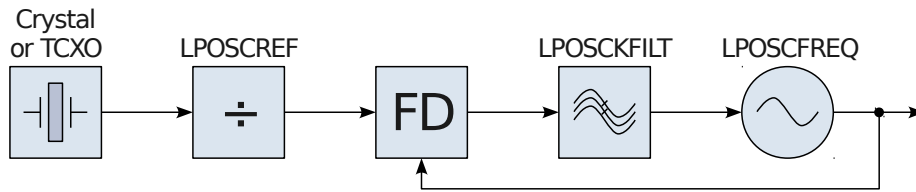


Figure 14: Low Power Oscillator Calibration Logic

When enabled (LPOSCCALIBR or LPOSCCALIBF enabled in register [LPOSCCONFIG](#)), the calibration logic is only activated when the crystal oscillator or TCXO is enabled as well. This allows “opportunistic” calibration – the Low Power Oscillator is calibrated whenever the reference frequency is enabled.

### 3.8. AUXILIARY DAC

The **AX5043** contains an auxiliary DAC. It can be used to output various receiver signals, such as RSSI or Frequency Offset, or just a value under program control. The DAC signal can be output either on the PWRAMP or ANTSEL pad.

The DAC may be operated in two modes.  $\Sigma\Delta$  mode employs a digital modulator to output a high resolution signal. Its output voltage range is  $\frac{1}{4}$  VDDIO to  $\frac{3}{4}$  VDDIO for a DACVALUE range from  $-2048$  to  $2047$ .

PWM mode outputs a pulse width modulated signal. It is only suitable for low frequency signals. Its output voltage range is  $0$  to  $VDDIO$  for a DACVALUE range from  $-2048$  to  $2047$ .

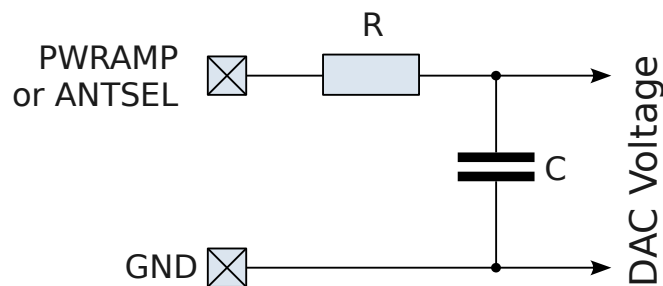
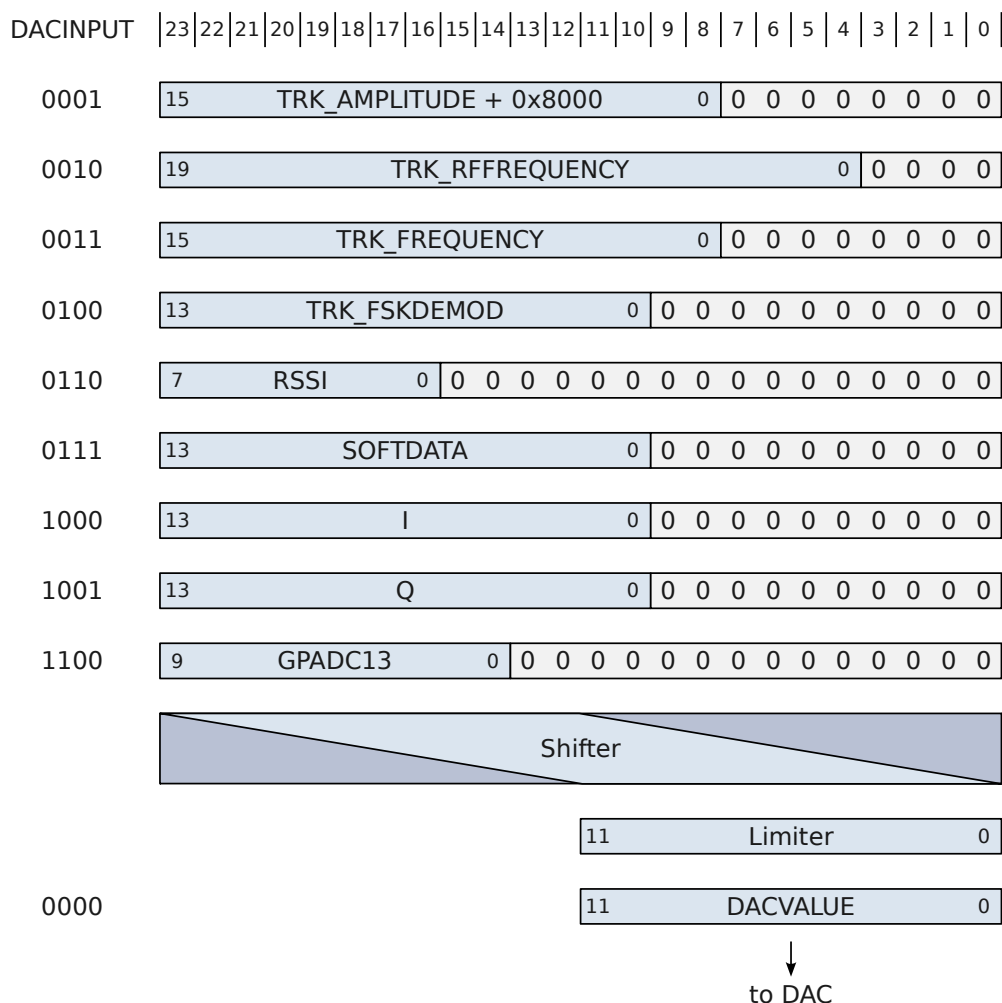


Figure 15: DAC RC Filter

A low pass filter, such as a simple R-C filter as shown in Figure 15, must be used to obtain the analog voltage.



**Figure 16: DAC Signal Scaling**

Figure 16 shows the DAC Signal scaling. DACINPUT in register [DACCONFIG](#) selects the source signal. The input signals are left aligned to 24 bits and padded with zeros. A signed shifter then shifts the selected value to the right by 0 to 15 digits as selected by the lower four bits of the [DACVALUE](#) register. The signal is then limited to the DAC value range of  $-2^{11} \dots 2^{11} - 1$ . This signal is then sent to the DAC core. Note that if [DACVALUE](#) is selected as input, the register value is directly sent to the DAC, the shifter is not used. In fact, DACVALUE and DACSHIFT share the same register bits.

## 4. REGISTER OVERVIEW

Addr Hex	Name	Dir	R	Reset	Bit								Description
					7	6	5	4	3	2	1	0	
Revision & Interface Probing													
000	<a href="#">REVISION</a>	R	R	01010001	SILICONREV(7:0)								Silicon Revision
001	<a href="#">SCRATCH</a>	RW	R	11000101	SCRATCH(7:0)								Scratch Register
Operating Mode													
002	<a href="#">PWRMODE</a>	RW	R	011-0000	RST	REFEN	XOEN	WDS	PWRMODE(3:0)				Power Mode
Voltage Regulator													
003	<a href="#">POWSTAT</a>	R	R	-----	SSUM	SREF	SVREF	SVANA	SVMODE M	SBEVANA	SBEVMO DEM	SVIO	Power Management Status
004	<a href="#">POWSTICKYSTAT</a>	R	R	-----	SSSUM	SSREF	SSVREF	SSVANA	SSVMOD EM	SSBEVAN A	SSBEVMO DEM	SSVIO	Power Management Sticky Status
005	<a href="#">POWIRQMASK</a>	RW	R	00000000	MPWR GOOD	MSREF	MSVREF	MS VANA	MS VMODEM	MSBE VANA	MSBE VMODEM	MSVIO	Power Management Interrupt Mask
Interrupt Control													
006	<a href="#">IRQMASK1</a>	RW	R	---00000	–	–	–	IRQMASK(12:8)				IRQ Mask	
007	<a href="#">IRQMASK0</a>	RW	R	00000000	IRQMASK(7:0)								IRQ Mask
008	<a href="#">RADIOEVENTMASK1</a>	RW	R	-----0	–	–	–	–	–	–	–	RADIO EVENT MASK(8)	Radio Event Mask
009	<a href="#">RADIOEVENTMASK0</a>	RW	R	00000000	RADIO EVENT MASK(7:0)								Radio Event Mask
00A	<a href="#">IRQINVERSION1</a>	RW	R	---00000	–	–	–	IRQINVERSION(12:8)				IRQ Inversion	

00B	<a href="#">IRQINVERSION0</a>	RW	R	00000000	IRQINVERSION(7:0)								IRQ Inversion
00C	<a href="#">IRQREQUEST1</a>	R	R	-----	-	-	-	IRQREQUEST(12:8)				IRQ Request	
00D	<a href="#">IRQREQUEST0</a>	R	R	-----	IRQREQUEST(7:0)								IRQ Request
00E	<a href="#">RADIOEVENTREQ1</a>	R		-----	-	-	-	-	-	-	-	RADIO EVENT REQ(8)	Radio Event Request
00F	<a href="#">RADIOEVENTREQ0</a>	R		-----	RADIO EVENT REQ(7:0)								Radio Event Request
Modulation & Framing													
010	<a href="#">MODULATION</a>	RW	R	---01000	-	-	-	RX HALF SPEED	MODULATION(3:0)				Modulation
011	<a href="#">ENCODING</a>	RW	R	---00010	-	-	-	ENC NOSYNC	ENC MANCH	ENC SCRAM	ENC DIFF	ENC INV	Encoder/Decoder Settings
012	<a href="#">FRAMING</a>	RW	R	-0000000	FRMRX	CRCMODE(2:0)			FRMMODE(2:0)			FABORT	Framing settings
014	<a href="#">CRCINIT3</a>	RW	R	11111111	CRCINIT(31:24)								CRC Initialisation Data
015	<a href="#">CRCINIT2</a>	RW	R	11111111	CRCINIT(23:16)								CRC Initialisation Data
016	<a href="#">CRCINIT1</a>	RW	R	11111111	CRCINIT(15:8)								CRC Initialisation Data
017	<a href="#">CRCINIT0</a>	RW	R	11111111	CRCINIT(7:0)								CRC Initialisation Data
Forward Error Correction													
018	<a href="#">FEC</a>	RW	R	00000000	SHORT MEM	RSTVI TERBI	FEC NEG	FEC POS	FECINPSHIFT(2:0)			FEC ENA	FEC (Viterbi) Configuration
019	<a href="#">FECSYNC</a>	RW	R	01100010	FECSYNC(7:0)								Interleaver Synchronisation Threshold
01A	<a href="#">FECSTATUS</a>	R	R	-----	FEC INV	MAXMETRIC(6:0)							FEC Status
Status													
01C	<a href="#">RADIOSTATE</a>	R	-	----0000	-	-	-	-	RADIOSTATE(3:0)				Radio Controller State



01D	<a href="#">XTALSTATUS</a>	R	R	-----	-	-	-	-	-	-	-	XTAL RUN	Crystal Oscillator Status
Pin Configuration													
020	<a href="#">PINSTATE</a>	R	R	-----	-	-	PS PWR AMP	PS ANT SEL	PS IRQ	PS DATA	PS DCLK	PS SYS CLK	Pinstate
021	<a href="#">PINFUNCSYSCLK</a>	RW	R	0--01000	PU SYSCLK	-	-	PFSYSCLK(4:0)					SYSCLK Pin Function
022	<a href="#">PINFUNCDCLK</a>	RW	R	00---100	PU DCLK	PI DCLK	-	-	-	PFDCLK(2:0)		DCLK Pin Function	
023	<a href="#">PINFUNCDATA</a>	RW	R	10---111	PU DATA	PI DATA	-	-	-	PFDATA(2:0)		DATA Pin Function	
024	<a href="#">PINFUNCIRQ</a>	RW	R	00---011	PU IRQ	PI IRQ	-	-	-	PFIRQ(2:0)		IRQ Pin Function	
025	<a href="#">PINFUNCANTSEL</a>	RW	R	00---110	PU ANTSEL	PI ANTSEL	-	-	-	PFANTSEL(2:0)		ANTSEL Pin Function	
026	<a href="#">PINFUNCPWRAMP</a>	RW	R	00--0110	PU PWRAMP	PI PWRAMP	-	-	PFPWRAMP(3:0)			PWRAMP Pin Function	
027	<a href="#">PWRAMP</a>	RW	R	-----0	-	-	-	-	-	-	-	PWRAMP	PWRAMP Control
FIFO													
028	<a href="#">FIFOSTAT</a>	R	R	0-----	FIFO AUTO COMMIT	-	FIFO FREE THR	FIFO CNT THR	FIFO OVER	FIFO UNDER	FIFO FULL	FIFO EMPTY	FIFO Control
		W	-			FIFOCMD(5:0)							
029	<a href="#">FIFODATA</a>	RW		-----	FIFODATA(7:0)			FIFO Data					
02A	<a href="#">FIFOCOUNT1</a>	R	R	-----0	-	-	-	-	-	-	-	FIFO COUNT(8)	Number of Words currently in FIFO
02B	<a href="#">FIFOCOUNT0</a>	R	R	00000000	FIFOCOUNT(7:0)								Number of Words currently in FIFO
02C	<a href="#">FIFOFREE1</a>	R	R	-----1	-	-	-	-	-	-	-	FIFO FREE(8)	Number of Words that can be written to FIFO
02D	<a href="#">FIFOFREE0</a>	R	R	00000000	FIFOFREE(7:0)								Number of Words that can be written to FIFO

02E	<a href="#">FIFOTHRESH1</a>	RW	R	-----0	–	–	–	–	–	–	–	FIFO THRESH(8)	FIFO Threshold	
02F	<a href="#">FIFOTHRESH0</a>	RW	R	00000000	FIFOTHRESH(7:0)								FIFO Threshold	
Synthesizer														
030	<a href="#">PLLLOOP</a>	RW	R	0---1001	FREQB	–	–	–	DIRECT	FILT EN	FLT(1:0)	PLL Loop Filter Settings		
031	<a href="#">PLLCPI</a>	RW	R	00001000	PLLCPI								PLL Charge Pump Current (Boosted)	
032	<a href="#">PLLVCODIV</a>	RW	R	–000–000	–	VCOI MAN	VCO2INT	VCOSSEL	–	RFDIV	REFDIV(1:0)	PLL Divider Settings		
033	<a href="#">PLLRANGINGA</a>	RW	R	00001000	STICKY LOCK	PLL LOCK	RNGERR	RNG START	VCORA(3:0)				PLL Autoranging	
034	<a href="#">FREQA3</a>	RW	R	00111001	FREQA(31:24)								Synthesizer Frequency	
035	<a href="#">FREQA2</a>	RW	R	00110100	FREQA(23:16)								Synthesizer Frequency	
036	<a href="#">FREQA1</a>	RW	R	11001100	FREQA(15:8)								Synthesizer Frequency	
037	<a href="#">FREQA0</a>	RW	R	11001101	FREQA(7:0)								Synthesizer Frequency	
038	<a href="#">PLLLOOPBOOST</a>	RW	R	0---1011	FREQB	–	–	–	DIRECT	FILT EN	FLT(1:0)	PLL Loop Filter Settings (Boosted)		
039	<a href="#">PLLCPIBOOST</a>	RW	R	11001000	PLLCPI								PLL Charge Pump Current	
03B	<a href="#">PLLRANGINGB</a>	RW	R	00001000	STICKY LOCK	PLL LOCK	RNGERR	RNG START	VCORB(3:0)				PLL Autoranging	
03C	<a href="#">FREQB3</a>	RW	R	00111001	FREQB(31:24)								Synthesizer Frequency	
03D	<a href="#">FREQB2</a>	RW	R	00110100	FREQB(23:16)								Synthesizer Frequency	
03E	<a href="#">FREQB1</a>	RW	R	11001100	FREQB(15:8)								Synthesizer Frequency	
03F	<a href="#">FREQB0</a>	RW	R	11001101	FREQB(7:0)								Synthesizer Frequency	

Signal Strength													
040	<a href="#">RSSI</a>	R	R	-----	RSSI(7:0)							Received Signal Strength Indicator	
041	<a href="#">BGNDRSSI</a>	RW	R	00000000	BGNDRSSI(7:0)							Background RSSI	
042	<a href="#">DIVERSITY</a>	RW	R	-----00	-	-	-	-	-	-	ANT SEL	DIV ENA	Antenna Diversity Configuration
043	<a href="#">AGCCOUNTER</a>	RW	R	-----	AGCCOUNTER(7:0)							AGC Current Value	
Receiver Tracking													
045	<a href="#">TRKDATARATE2</a>	R	R	-----	TRKDATARATE(23:16)							Datarate Tracking	
046	<a href="#">TRKDATARATE1</a>	R	R	-----	TRKDATARATE(15:8)							Datarate Tracking	
047	<a href="#">TRKDATARATE0</a>	R	R	-----	TRKDATARATE(7:0)							Datarate Tracking	
048	<a href="#">TRKAMPL1</a>	R	R	-----	TRKAMPL(15:8)							Amplitude Tracking	
049	<a href="#">TRKAMPL0</a>	R	R	-----	TRKAMPL(7:0)							Amplitude Tracking	
04A	<a href="#">TRKPHASE1</a>	R	R	-----	-	-	-	-	TRKPHASE(11:8)			Phase Tracking	
04B	<a href="#">TRKPHASE0</a>	R	R	-----	TRKPHASE(7:0)							Phase Tracking	
04D	<a href="#">TRKRFFREQ2</a>	RW	R	-----	-	-	-	-	TRRKFREQ(19:16)			RF Frequency Tracking	
04E	<a href="#">TRKRFFREQ1</a>	RW	R	-----	TRRKFREQ(15:8)							RF Frequency Tracking	
04F	<a href="#">TRKRFFREQ0</a>	RW	R	-----	TRRKFREQ(7:0)							RF Frequency Tracking	
050	<a href="#">TRKFREQ1</a>	RW	R	-----	TRKFREQ(15:8)							Frequency Tracking	
051	<a href="#">TRKFREQ0</a>	RW	R	-----	TRKFREQ(7:0)							Frequency Tracking	
052	<a href="#">TRKFSKDEMOD1</a>	R	R	-----	-	-	TRKFSKDEMOD(13:8)					FSK Demodulator Tracking	
053	<a href="#">TRKFSKDEMOD0</a>	R	R	-----	TRKFSKDEMOD(7:0)							FSK Demodulator Tracking	
Timer													
059	<a href="#">TIMER2</a>	R	-	-----	TIMER(23:16)							1MHz Timer	
05A	<a href="#">TIMER1</a>	R	-	-----	TIMER(15:8)							1MHz Timer	

05B	<a href="#">TIMER0</a>	R	–	-----	TIMER(7:0)	1MHz Timer
Wakeup Timer						
068	<a href="#">WAKEUPTIMER1</a>	R	R	-----	WAKEUPTIMER(15:8)	Wakeup Timer
069	<a href="#">WAKEUPTIMER0</a>	R	R	-----	WAKEUPTIMER(7:0)	Wakeup Timer
06A	<a href="#">WAKEUP1</a>	RW	R	00000000	WAKEUP(15:8)	Wakeup Time
06B	<a href="#">WAKEUP0</a>	RW	R	00000000	WAKEUP(7:0)	Wakeup Time
06C	<a href="#">WAKEUPFREQ1</a>	RW	R	00000000	WAKEUPFREQ(15:8)	Wakeup Frequency
06D	<a href="#">WAKEUPFREQ0</a>	RW	R	00000000	WAKEUPFREQ(7:0)	Wakeup Frequency
06E	<a href="#">WAKEUPXOEARLY</a>	RW	R	00000000	WAKEUPXOEARLY(7:0)	Wakeup Crystal Oscillator Early
Physical Layer Parameters						
Receiver Parameters						
100	<a href="#">IFFREQ1</a>	RW	R	00010011	IFFREQ(15:8)	2nd LO / IF Frequency
101	<a href="#">IFFREQ0</a>	RW	R	00100111	IFFREQ(7:0)	2nd LO / IF Frequency
102	<a href="#">DECIMATION</a>	RW	R	–0001101	–DECIMATION(6:0)	Decimation Factor
103	<a href="#">RXDATARATE2</a>	RW	R	00000000	RXDATARATE(23:16)	Receiver Datarate
104	<a href="#">RXDATARATE1</a>	RW	R	00111101	RXDATARATE(15:8)	Receiver Datarate
105	<a href="#">RXDATARATE0</a>	RW	R	10001010	RXDATARATE(7:0)	Receiver Datarate
106	<a href="#">MAXDROFFSET2</a>	RW	R	00000000	MAXDROFFSET(23:16)	Maximum Receiver Datarate Offset
107	<a href="#">MAXDROFFSET1</a>	RW	R	00000000	MAXDROFFSET(15:8)	Maximum Receiver Datarate Offset
108	<a href="#">MAXDROFFSET0</a>	RW	R	10011110	MAXDROFFSET(7:0)	Maximum Receiver Datarate Offset

109	<a href="#">MAXRFOFFSET2</a>	RW	R	0---0000	FREQ OFFS CORR	-	-	-	MAXRFOFFSET(19:16)	Maximum Receiver RF Offset	
10A	<a href="#">MAXRFOFFSET1</a>	RW	R	00010110	MAXRFOFFSET(15:8)					Maximum Receiver RF Offset	
10B	<a href="#">MAXRFOFFSET0</a>	RW	R	10000111	MAXRFOFFSET(7:0)					Maximum Receiver RF Offset	
10C	<a href="#">FSKDEVMAX1</a>	RW	R	00000000	FSKDEVMAX(15:8)					Four FSK Rx Deviation	
10D	<a href="#">FSKDEVMAX0</a>	RW	R	10000000	FSKDEVMAX(7:0)					Four FSK Rx Deviation	
10E	<a href="#">FSKDEVMIN1</a>	RW	R	11111111	FSKDEVMIN(15:8)					Four FSK Rx Deviation	
10F	<a href="#">FSKDEVMIN0</a>	RW	R	10000000	FSKDEVMIN(7:0)					Four FSK Rx Deviation	
110	<a href="#">AFSKSPACE1</a>	RW	R	----0000	-	-	-	-	AFSKSPACE(11:8)	AFSK Space (0) Frequency	
111	<a href="#">AFSKSPACE0</a>	RW	R	01000000	AFSKSPACE(7:0)					AFSK Space (0) Frequency	
112	<a href="#">AFSKMARK1</a>	RW	R	----0000	-	-	-	-	AFSKMARK(11:8)	AFSK Mark (1) Frequency	
113	<a href="#">AFSKMARK0</a>	RW	R	01110101	AFSKMARK(7:0)					AFSK Mark (1) Frequency	
114	<a href="#">AFSKCTRL</a>	RW	R	---00100	-	-	-	AFSKSHIFT0(4:0)		AFSK Control	
115	<a href="#">AMPLFILTER</a>	RW	R	----0000	-	-	-	-	AMPLFILTER(3:0)	Amplitude Filter	
116	<a href="#">FREQUENCYLEAK</a>	RW	R	----0000	-	-	-	-	FREQUENCYLEAK(3:0)	Baseband Frequency Recovery Loop Leakiness	
117	<a href="#">RXPARAMSETS</a>	RW	R	00000000	RXPS3(1:0)		RXPS2(1:0)		RXPS1(1:0)	RXPS0(1:0)	Receiver Parameter Set Indirection
118	<a href="#">RXPARAMCURSET</a>	R	R	-----	-	-	-	RXSI(2)	RXSN(1:0)	RXSI(1:0)	Receiver Parameter Current Set
Receiver Parameter Set 0											
120	<a href="#">AGCGAIN0</a>	RW	R	10110100	AGCDECAY0(3:0)				AGCATTACK0(3:0)		AGC Speed
121	<a href="#">AGCTARGET0</a>	RW	R	01110110	AGCTARGET0(7:0)					AGC Target	
122	<a href="#">AGCAHYST0</a>	RW	R	-----000	-	-	-	-	-	AGCAHYST0(2:0)	AGC Digital Threshold Range
123	<a href="#">AGCMINMAX0</a>	RW	R	-000-000	-	AGCMAXDA0(2:0)			-	AGCMINDA0(2:0)	AGC Digital Minimum/Maximum Set Points

124	<a href="#">TIMEGAIN0</a>	RW	R	11111000	TIMEGAIN0M(3:0)				TIMEGAIN0E(3:0)	Timing Gain
125	<a href="#">DRGAIN0</a>	RW	R	11110010	DRGAIN0M(3:0)				DRGAIN0E(3:0)	Data Rate Gain
126	<a href="#">PHASEGAIN0</a>	RW	R	11--0011	FILTERIDX0(1:0)	–	–	PHASEGAIN0(3:0)	Filter Index, Phase Gain	
127	<a href="#">FREQGAINA0</a>	RW	R	00001111	FREQ LIM0	FREQ MODULO0	FREQ HALFMOD0	FREQ AMPL GATE0	FREQGAINA0(3:0)	Frequency Gain A
128	<a href="#">FREQGAINB0</a>	RW	R	00-11111	FREQ FREEZE0	FREQ AVG0	–	FREQGAINB0(4:0)	Frequency Gain B	
129	<a href="#">FREQGAINC0</a>	RW	R	---01010	–	–	–	FREQGAINC0(4:0)	Frequency Gain C	
12A	<a href="#">FREQGAIND0</a>	RW	R	0--01010	RFFREQ FREEZE0	–	–	FREQGAIND0(4:0)	Frequency Gain D	
12B	<a href="#">AMPLGAIN0</a>	RW	R	01--0110	AMPL AVG0	AMPL AGC0	–	–	AMPLGAIN0(3:0)	Amplitude Gain
12C	<a href="#">FREQDEV10</a>	RW	R	----0000	–	–	–	–	FREQDEV0(11:8)	Receiver Frequency Deviation
12D	<a href="#">FREQDEV00</a>	RW	R	00100000	FREQDEV0(7:0)					Receiver Frequency Deviation
12E	<a href="#">FOURFSK0</a>	RW	R	---10110	–	–	–	DEV UPDATE0	DEVDECAY0(3:0)	Four FSK Control
12F	<a href="#">BBOFFSRES0</a>	RW	R	10001000	RESINTB0(3:0)				RESINTA0(3:0)	Baseband Offset Compensation Resistors
Receiver Parameter Set 1										
130	<a href="#">AGCGAIN1</a>	RW	R	10110100	AGCDECAY1(3:0)				AGCATTACK1(3:0)	AGC Speed
131	<a href="#">AGCTARGET1</a>	RW	R	01110110	AGCTARGET1(7:0)					AGC Target

132	<a href="#">AGCAHYST1</a>	RW	R	-----000	–	–	–	–	–	AGCAHYST1(2:0)	AGC Digital Threshold Range
133	<a href="#">AGCMINMAX1</a>	RW	R	–000–000	–	AGCMAXDA1(2:0)			–	AGCMINDA1(2:0)	AGC Digital Minimum/Maximum Set Points
134	<a href="#">TIMEGAIN1</a>	RW	R	11110110	TIMEGAIN1M(3:0)				TIMEGAIN1E(3:0)		Timing Gain
135	<a href="#">DRGAIN1</a>	RW	R	11110001	DRGAIN1M(3:0)				DRGAIN1E(3:0)		Data Rate Gain
136	<a href="#">PHASEGAIN1</a>	RW	R	11--0011	FILTERIDX1(1:0)		–	–	PHASEGAIN1(3:0)		Filter Index, Phase Gain
137	<a href="#">FREQGAINA1</a>	RW	R	00001111	FREQ LIM1	FREQ MODULO 1	FREQ HALFMOD 1	FREQ AMPL GATE1	FREQGAINA1(3:0)		Frequency Gain A
138	<a href="#">FREQGAINB1</a>	RW	R	00–11111	FREQ FREEZE1	FREQ AVG1	–	FREQGAINB1(4:0)			Frequency Gain B
139	<a href="#">FREQGAINC1</a>	RW	R	---01011	–	–	–	FREQGAINC1(4:0)			Frequency Gain C
13A	<a href="#">FREQGAIND1</a>	RW	R	0--01011	RFFREQ FREEZE1	–	–	FREQGAIND1(4:0)			Frequency Gain D
13B	<a href="#">AMPLGAIN1</a>	RW	R	01--0110	AMPL AVG1	AMPL1 AGC1	–	–	AMPLGAIN1(3:0)		Amplitude Gain
13C	<a href="#">FREQDEV11</a>	RW	R	-----0000	–	–	–	–	FREQDEV1(11:8)		Receiver Frequency Deviation
13D	<a href="#">FREQDEV01</a>	RW	R	00100000	FREQDEV1(7:0)						Receiver Frequency Deviation
13E	<a href="#">FOURFSK1</a>	RW	R	---11000	–	–	–	DEV UPDATE1	DEVDECAY1(3:0)		Four FSK Control
13F	<a href="#">BBOFFSRES1</a>	RW	R	10001000	RESINTB1(3:0)				RESINTA1(3:0)		Baseband Offset Compensation Resistors
Receiver Parameter Set 2											
140	<a href="#">AGCGAIN2</a>	RW	R	11111111	AGCDECAY2(3:0)				AGCATTACK2(3:0)		AGC Speed
141	<a href="#">AGCTARGET2</a>	RW	R	01110110	AGCTARGET2(7:0)						AGC Target
142	<a href="#">AGCAHYST2</a>	RW	R	-----000	–	–	–	–	–	AGCAHYST2(2:0)	AGC Digital Threshold Range

143	<a href="#">AGCMINMAX2</a>	RW	R	−000−000	−	AGCMAXDA2(2:0)		−	AGCMINDA2(2:0)	AGC Digital Minimum/Maximum Set Points	
144	<a href="#">TIMEGAIN2</a>	RW	R	11110101	TIMEGAIN2M(3:0)				TIMEGAIN2E(3:0)	Timing Gain	
145	<a href="#">DRGAIN2</a>	RW	R	11110000	DRGAIN2M(3:0)				DRGAIN2E(3:0)	Data Rate Gain	
146	<a href="#">PHASEGAIN2</a>	RW	R	11−−0011	FILTERIDX2(1:0)		−	−	PHASEGAIN2(3:0)	Filter Index, Phase Gain	
147	<a href="#">FREQGAINA2</a>	RW	R	00001111	FREQ LIM2	FREQ MODULO 2	FREQ HALFMOD 2	FREQ AMPL GATE2	FREQGAINA2(3:0)	Frequency Gain A	
148	<a href="#">FREQGAINB2</a>	RW	R	00−11111	FREQ FREEZE2	FREQ AVG2	−	FREQGAINB2(4:0)		Frequency Gain B	
149	<a href="#">FREQGAINC2</a>	RW	R	−−−01101	−	−	−	FREQGAINC2(4:0)		Frequency Gain C	
14A	<a href="#">FREQGAIND2</a>	RW	R	0−−01101	RFFREQ FREEZE2	−	−	FREQGAIND2(4:0)		Frequency Gain D	
14B	<a href="#">AMPLGAIN2</a>	RW	R	01−−0110	AMPL AVG2	AMPL AGC2	−	−	AMPLGAIN2(3:0)	Amplitude Gain	
14C	<a href="#">FREQDEV12</a>	RW	R	−−−−0000	−	−	−	−	FREQDEV2(11:8)	Receiver Frequency Deviation	
14D	<a href="#">FREQDEV02</a>	RW	R	00100000	FREQDEV2(7:0)					Receiver Frequency Deviation	
14E	<a href="#">FOURFSK2</a>	RW	R	−−−11010	−	−	−	DEV UPDATE2	DEVDECAY2(3:0)	Four FSK Control	
14F	<a href="#">BBOFFSRES2</a>	RW	R	10001000	RESINTB2(3:0)				RESINTA2(3:0)		Baseband Offset Compensation Resistors
Receiver Parameter Set 3											
150	<a href="#">AGCGAIN3</a>	RW	R	11111111	AGCDECAY3(3:0)				AGCATTACK3(3:0)		AGC Speed
151	<a href="#">AGCTARGET3</a>	RW	R	01110110	AGCTARGET3(7:0)					AGC Target	
152	<a href="#">AGCAHYST3</a>	RW	R	−−−−−000	−	−	−	−	−	AGCAHYST3(2:0)	AGC Digital Threshold Range



153	<a href="#">AGCMINMAX3</a>	RW	R	−000−000	−	AGCMAXDA3(2:0)			−	AGCMINDA3(2:0)		AGC Digital Minimum/Maximum Set Points
154	<a href="#">TIMEGAIN3</a>	RW	R	11110101	TIMEGAIN3M(3:0)					TIMEGAIN3E(3:0)		Timing Gain
155	<a href="#">DRGAIN3</a>	RW	R	11110000	DRGAIN3M(3:0)					DRGAIN3E(3:0)		Data Rate Gain
156	<a href="#">PHASEGAIN3</a>	RW	R	11−−0011	FILTERIDX3(1:0)		−	−	PHASEGAIN3(3:0)		Filter Index, Phase Gain	
157	<a href="#">FREQGAINA3</a>	RW	R	00001111	FREQ LIM3	FREQ MODULO 3	FREQ HALFMOD 3	FREQ AMPL GATE3	FREQGAINA3(3:0)		Frequency Gain A	
158	<a href="#">FREQGAINB3</a>	RW	R	00−11111	FREQ FREEZE3	FREQ AVG3	−	FREQGAINB3(4:0)		Frequency Gain B		
159	<a href="#">FREQGAINC3</a>	RW	R	−−−01101	−	−	−	FREQGAINC3(4:0)		Frequency Gain C		
15A	<a href="#">FREQGAIND3</a>	RW	R	0−−01101	RFFREQ FREEZE3	−	−	FREQGAIND3(4:0)		Frequency Gain D		
15B	<a href="#">AMPLGAIN3</a>	RW	R	01−−0110	AMPL AVG3	AMPL AGC3	−	−	AMPLGAIN3(3:0)		Amplitude Gain	
15C	<a href="#">FREQDEV13</a>	RW	R	−−−−0000	−	−	−	−	FREQDEV3(11:8)		Receiver Frequency Deviation	
15D	<a href="#">FREQDEV03</a>	RW	R	00100000	FREQDEV3(7:0)					Receiver Frequency Deviation		
15E	<a href="#">FOURFSK3</a>	RW	R	−−−11010	−	−	−	DEV UPDATE3	DEVDECAY3(3:0)		Four FSK Control	
15F	<a href="#">BBOFFSRES3</a>	RW	R	10001000	RESINTB3(3:0)					RESINTA3(3:0)		Baseband Offset Compensation Resistors
Transmitter Parameters												
160	<a href="#">MODCFGF</a>	RW	R	−−−−−00	−	−	−	−	−	−	FREQ SHAPE(1:0)	Modulator Configuration F
161	<a href="#">FSKDEV2</a>	RW	R	00000000	FSKDEV(23:16)					FSK Frequency Deviation		
162	<a href="#">FSKDEV1</a>	RW	R	00001010	FSKDEV(15:8)					FSK Frequency Deviation		
163	<a href="#">FSKDEV0</a>	RW	R	00111101	FSKDEV(7:0)					FSK Frequency Deviation		

164	<a href="#">MODCFG</a>	RW	R	0000–101	BROWN GATE	PTTLCK GATE	SLOW RAMP(1:0)	–	AMPL SHAPE	TX SE	TX DIFF	Modulator Configuration A
165	<a href="#">TXRATE2</a>	RW	R	00000000	TXRATE(23:16)							Transmitter Bitrate
166	<a href="#">TXRATE1</a>	RW	R	00101000	TXRATE(15:8)							Transmitter Bitrate
167	<a href="#">TXRATE0</a>	RW	R	11110110	TXRATE(7:0)							Transmitter Bitrate
168	<a href="#">TXPWRCOEFFA1</a>	RW	R	00000000	TXPWRCOEFFA(15:8)							Transmitter Predistortion Coefficient A
169	<a href="#">TXPWRCOEFFA0</a>	RW	R	00000000	TXPWRCOEFFA(7:0)							Transmitter Predistortion Coefficient A
16A	<a href="#">TXPWRCOEFFB1</a>	RW	R	00001111	TXPWRCOEFFB(15:8)							Transmitter Predistortion Coefficient B
16B	<a href="#">TXPWRCOEFFB0</a>	RW	R	11111111	TXPWRCOEFFB(7:0)							Transmitter Predistortion Coefficient B
16C	<a href="#">TXPWRCOEFFC1</a>	RW	R	00000000	TXPWRCOEFFC(15:8)							Transmitter Predistortion Coefficient C
16D	<a href="#">TXPWRCOEFFC0</a>	RW	R	00000000	TXPWRCOEFFC(7:0)							Transmitter Predistortion Coefficient C
16E	<a href="#">TXPWRCOEFFD1</a>	RW	R	00000000	TXPWRCOEFFD(15:8)							Transmitter Predistortion Coefficient D
16F	<a href="#">TXPWRCOEFFD0</a>	RW	R	00000000	TXPWRCOEFFD(7:0)							Transmitter Predistortion Coefficient D
170	<a href="#">TXPWRCOEFFE1</a>	RW	R	00000000	TXPWRCOEFFE(15:8)							Transmitter Predistortion Coefficient E
171	<a href="#">TXPWRCOEFFE0</a>	RW	R	00000000	TXPWRCOEFFE(7:0)							Transmitter Predistortion Coefficient E
PLL Parameters												
180	<a href="#">PLLVCOI</a>	RW	R	0–010010	VCOIE	–	VCOI(5:0)					VCO Current

181	<a href="#">PLLVOIR</a>	RW	R	-----	-	-	VCOIR(5:0)				VCO Current Readback	
182	<a href="#">PLLLOCKDET</a>	RW	R	-----011	LOCKDETDLYR(1:0)		-	-	-	LOCK DET DLYM	LOCKDETDLY(1:0)	PLL Lock Detect Delay
183	<a href="#">PLLRNGCLK</a>	RW	R	-----011	-	-	-	-	-	PLLRNGCLK(2:0)		PLL Ranging Clock
Crystal Oscillator												
184	<a href="#">XTALCAP</a>	RW	R	00000000	XTALCAP(7:0)						Crystal Oscillator Load Capacitance Configuration	
Baseband												
188	<a href="#">BBTUNE</a>	RW	R	---01001	-	-	-	BB TUNE RUN	BBTUNE(3:0)		Baseband Tuning	
189	<a href="#">BBOFFSCAP</a>	RW	R	-111-111	-	CAP INT B(2:0)		-	CAP INT A(2:0)		Baseband Offset Compensation Capacitors	
MAC Layer Parameters												
Packet Format												
200	<a href="#">PKTADDRCFG</a>	RW	R	001-0000	MSB FIRST	CRC SKIP FIRST	FEC SYNC DIS	-	ADDR POS(3:0)		Packet Address Config	
201	<a href="#">PKTLENCFG</a>	RW	R	00000000	LEN BITS(3:0)				LEN POS(3:0)		Packet Length Config	
202	<a href="#">PKTLENOFFSET</a>	RW	R	00000000	LEN OFFSET(7:0)						Packet Length Offset	
203	<a href="#">PKTMAXLEN</a>	RW	R	00000000	MAX LEN(7:0)						Packet Maximum Length	
204	<a href="#">PKTADDR3</a>	RW	R	00000000	ADDR(31:24)						Packet Address 3	
205	<a href="#">PKTADDR2</a>	RW	R	00000000	ADDR(23:16)						Packet Address 2	
206	<a href="#">PKTADDR1</a>	RW	R	00000000	ADDR(15:8)						Packet Address 1	
207	<a href="#">PKTADDR0</a>	RW	R	00000000	ADDR(7:0)						Packet Address 0	
208	<a href="#">PKTADDRMASK3</a>	RW	R	00000000	ADDRMASK(31:24)						Packet Address Mask 3	
209	<a href="#">PKTADDRMASK2</a>	RW	R	00000000	ADDRMASK(23:16)						Packet Address Mask 2	

20A	<a href="#">PKTADDRMASK1</a>	RW	R	00000000	ADDRMASK(15:8)				Packet Address Mask 1	
20B	<a href="#">PKTADDRMASK0</a>	RW	R	00000000	ADDRMASK(7:0)				Packet Address Mask 0	
Pattern Match										
210	<a href="#">MATCH0PAT3</a>	RW	R	00000000	MATCH0PAT(31:24)				Pattern Match Unit 0, Pattern	
211	<a href="#">MATCH0PAT2</a>	RW	R	00000000	MATCH0PAT(23:16)				Pattern Match Unit 0, Pattern	
212	<a href="#">MATCH0PAT1</a>	RW	R	00000000	MATCH0PAT(15:8)				Pattern Match Unit 0, Pattern	
213	<a href="#">MATCH0PAT0</a>	RW	R	00000000	MATCH0PAT(7:0)				Pattern Match Unit 0, Pattern	
214	<a href="#">MATCH0LEN</a>	RW	R	0---00000	MATCH0 RAW	-	-	MATCH0LEN(4:0)	Pattern Match Unit 0, Pattern Length	
215	<a href="#">MATCH0MIN</a>	RW	R	----00000	-	-	-	MATCH0MIN(4:0)	Pattern Match Unit 0, Minimum Match	
216	<a href="#">MATCH0MAX</a>	RW	R	----11111	-	-	-	MATCH0MAX(4:0)	Pattern Match Unit 0, Maximum Match	
218	<a href="#">MATCH1PAT1</a>	RW	R	00000000	MATCH1PAT(15:8)				Pattern Match Unit 1, Pattern	
219	<a href="#">MATCH1PAT0</a>	RW	R	00000000	MATCH1PAT(7:0)				Pattern Match Unit 1, Pattern	
21C	<a href="#">MATCH1LEN</a>	RW	R	0---0000	MATCH1 RAW	-	-	-	MATCH1LEN(3:0)	Pattern Match Unit 1, Pattern Length
21D	<a href="#">MATCH1MIN</a>	RW	R	-----0000	-	-	-	-	MATCH1MIN(3:0)	Pattern Match Unit 1, Minimum Match
21E	<a href="#">MATCH1MAX</a>	RW	R	-----1111	-	-	-	-	MATCH1MAX(3:0)	Pattern Match Unit 1, Maximum Match
Packet Controller										
220	<a href="#">TMGTXBOOST</a>	RW	R	00110010	TMGTXBOOSTE(2:0)			TMGTXBOOSTM(4:0)	Transmit PLL Boost Time	
221	<a href="#">TMGTXSETTLE</a>	RW	R	00001010	TMGTXSETTLEE(2:0)			TMGTXSETTLEM(4:0)	Transmit PLL (post Boost) Settling Time	

223	<a href="#">TMGRXBOOST</a>	RW	R	00110010	TMGRXBOOSTE(2:0)		TMGRXBOOSTM(4:0)				Receive PLL Boost Time		
224	<a href="#">TMGRXSETTLE</a>	RW	R	00010100	TMGRXSETTLEE(2:0)		TMGRXSETTLEM(4:0)				Receive PLL (post Boost) Settling Time		
225	<a href="#">TMGRXOFFSACQ</a>	RW	R	01110011	TMGRXOFFSACQE(2:0)		TMGRXOFFSACQM(4:0)				Receive Baseband DC Offset Acquisition Time		
226	<a href="#">TMGRXCOARSEAGC</a>	RW	R	00111001	TMGRXCOARSEAGCE(2:0)		TMGRXCOARSEAGCM(4:0)				Receive Coarse AGC Time		
227	<a href="#">TMGRXAGC</a>	RW	R	00000000	TMGRXAGCE(2:0)		TMGRXAGCM(4:0)				Receiver AGC Settling Time		
228	<a href="#">TMGRXRSSI</a>	RW	R	00000000	TMGRXRSSIE(2:0)		TMGRXRSSIM(4:0)				Receiver RSSI Settling Time		
229	<a href="#">TMGRXPREAMBLE1</a>	RW	R	00000000	TMGRXPREAMBLE1E(2:0)		TMGRXPREAMBLE1M(4:0)				Receiver Preamble 1 Timeout		
22A	<a href="#">TMGRXPREAMBLE2</a>	RW	R	00000000	TMGRXPREAMBLE2E(2:0)		TMGRXPREAMBLE2M(4:0)				Receiver Preamble 2 Timeout		
22B	<a href="#">TMGRXPREAMBLE3</a>	RW	R	00000000	TMGRXPREAMBLE3E(2:0)		TMGRXPREAMBLE3M(4:0)				Receiver Preamble 3 Timeout		
22C	<a href="#">RSSIREFERENCE</a>	RW	R	00000000	RSSIREFERENCE(7:0)						RSSI Offset		
22D	<a href="#">RSSIABSTHR</a>	RW	R	00000000	RSSIABSTHR(7:0)						RSSI Absolute Threshold		
22E	<a href="#">BGNDRSSIGAIN</a>	RW	R	----0000	–	–	–	–	BGNDRSSIGAIN(3:0)			Background RSSI Averaging Time Constant	
22F	<a href="#">BGNDRSSITHR</a>	RW	R	--000000	–	–	BGNDRSSITHR(5:0)			Background RSSI Relative Threshold			
230	<a href="#">PKTCHUNKSIZE</a>	RW	R	----0000	–	–	–	–	PKTCHUNKSIZE(3:0)			Packet Chunk Size	
231	<a href="#">PKTMISCFLAGS</a>	RW	R	----00000	–	–	–	WOR MULTI PKT	AGC SETTL DET	BGND RSSI	RXAGC CLK	RXRSSI CLK	Packet Controller Miscellaneous Flags
232	<a href="#">PKTSTOREFLAGS</a>	RW	R	–0000000	–	ST ANT RSSI	ST CRCB	ST RSSI	ST DR	ST RFOFFS	ST FOFFS	ST TIMER	Packet Controller Store Flags
233	<a href="#">PKTACCEPTFLAGS</a>	RW	R	--000000	–	–	ACCPT LRGP	ACCPT SZF	ACCPT ADDRF	ACCPT CRCF	ACCPT ABRT	ACCPT RESIDUE	Packet Controller Accept Flags

Special Functions													
General Purpose ADC													
300	<a href="#">GPADCCTRL</a>	RW	R	--000000	BUSY	–	0	0	0	GPADC13	CONT	CH ISOL	General Purpose ADC Control
301	<a href="#">GPADCPERIOD</a>	RW	R	00111111	GPADCPERIOD(7:0)								GPADC Sampling Period
308	<a href="#">GPADC13VALUE1</a>	R		-----	–	–	–	–	–	–	GPADC13VALUE(9:8)		GPADC13 Value
309	<a href="#">GPADC13VALUE0</a>	R		-----	GPADC13VALUE(7:0)								GPADC13 Value
Low Power Oscillator Calibration													
310	<a href="#">LPOSCCONFIG</a>	RW	R	00000000	LPOSC OSC INVERT	LPOSC OSC DOUBLE	LPOSC CALIBR	LPOSC CALIBF	LPOSC IRQR	LPOSC IRQF	LPOSC FAST	LPOSC ENA	Low Power Oscillator Configuration
311	<a href="#">LPOSCSTATUS</a>	R	R	-----	–	–	–	–	–	–	LPOSC IRQ	LPOSC EDGE	Low Power Oscillator Status
312	<a href="#">LPOSCFILT1</a>	RW	R	00100000	LPOSCFILT(15:8)								Low Power Oscillator Calibration Filter Constant
313	<a href="#">LPOSCFILT0</a>	RW	R	11000100	LPOSCFILT(7:0)								Low Power Oscillator Calibration Filter Constant
314	<a href="#">LPOSCREF1</a>	RW	R	01100001	LPOSCREF(15:8)								Low Power Oscillator Calibration Reference
315	<a href="#">LPOSCREF0</a>	RW	R	10101000	LPOSCREF(7:0)								Low Power Oscillator Calibration Reference
316	<a href="#">LPOSCFREQ1</a>	RW	R	00000000	LPOSCFREQ(9:2)								Low Power Oscillator Calibration Frequency
317	<a href="#">LPOSCFREQ0</a>	RW	R	0000----	LPOSCFREQ(1:-2)				–	–	–	–	Low Power Oscillator Calibration Frequency

318	<a href="#">LPOSCPER1</a>	RW		-----	LPOSCPER(15:8)					Low Power Oscillator Calibration Period
319	<a href="#">LPOSCPER0</a>	RW		-----	LPOSCPER(7:0)					Low Power Oscillator Calibration Period
DAC										
330	<a href="#">DACVALUE1</a>	RW	R	----0000	–	–	–	–	DACVALUE(11:8)	DAC Value
331	<a href="#">DACVALUE0</a>	RW	R	00000000	DACVALUE(7:0)					DAC Value
332	<a href="#">DACCONFIG</a>	RW	R	00--0000	DAC PWM	DAC CLK X2	–	–	DACINPUT(3:0)	DAC Configuration
Performance Tuning Registers										
F00–FFF	<a href="#">PERFTUNE</a>	RW		-----						Performance Tuning Registers

Two Columns, 2 spaltig

## 5. REGISTER DETAILS

### 5.1. REVISION AND INTERFACE PROBING

#### 5.1.1. REGISTER: REVISION

Name	Bits	R/W	Reset	Description
REVISION	7:0	R	01010001	Silicon Revision

#### 5.1.2. REGISTER: SCRATCH

Name	Bits	R/W	Reset	Description
SCRATCH	7:0	R	11000101	Scratch Register

The SCRATCH register does not affect the function of the chip in any way. It is intended for the Microcontroller to test communication to the AX5043.

### 5.2. OPERATING MODE

#### 5.2.1. REGISTER: PWRMODE

Name	Bits	R/W	Reset	Description
PWRMODE	3:0	RW	0000	Bits    Meaning 0000 Powerdown; all circuits powered down 0001 Deep Sleep Mode; Chip is fully powered down until SEL is lowered again; loses all register contents 0101 Crystal Oscillator enabled 0111 FIFO enabled 1000 Synthesizer running, Receive Mode 1001 Receiver Running 1011 Receiver Wake-on-Radio Mode 1100 Synthesizer running, Transmit Mode 1101 Transmitter Running
WDS	4	R	–	Wakeup from Deep Sleep
REFEN	5	RW	1	Reference Enable; set to 1 to power the internal reference circuitry
XOEN	6	RW	1	Crystal Oscillator Enable



RST	7	RW	0	Reset; setting this bit to 1 resets the whole chip. This bit does not auto-reset – the chip remains in reset state until this bit is cleared.
-----	---	----	---	---

### 5.3. POWER MANAGEMENT

#### 5.3.1. REGISTER: POWSTAT

Name	Bits	R/W	Reset	Description
SVIO	0	R	–	IO Voltage Large Enough (not Brownout)
SBEVMODEM	1	R	–	Modem Domain Voltage Brownout Error (Inverted; 0=Brownout, 1=Power OK)
SBEVANA	2	R	–	Analog Domain Voltage Brownout Error (Inverted; 0=Brownout, 1=Power OK)
SVMODEM	3	R	–	Modem Domain Voltage Regulator Ready
SVANA	4	R	–	Analog Domain Voltage Regulator Ready
SVREF	5	R	–	Reference Voltage Regulator Ready
SREF	6	R	–	Reference Ready
SSUM	7	R	–	Summary Ready Status (one when all unmasked <a href="#">POWIRQMASK</a> power sources are ready)

#### 5.3.2. REGISTER: POWSTICKYSTAT

Name	Bits	R/W	Reset	Description
SSVIO	0	R	–	Sticky IO Voltage Large Enough (not Brownout)
SSBEVMODEM	1	R	–	Sticky Modem Domain Voltage Brownout Error (Inverted; 0=Brownout detected, 1=Power OK)
SSBEVANA	2	R	–	Sticky Analog Domain Voltage Brownout Error (Inverted; 0=Brownout detected, 1=Power OK)
SSVMODEM	3	R	–	Sticky Modem Domain Voltage Regulator Ready
SSVANA	4	R	–	Sticky Analog Domain Voltage Regulator Ready
SSVREF	5	R	–	Sticky Reference Voltage Regulator Ready
SSREF	6	R	–	Sticky Reference Ready
SSSUM	7	R	–	Sticky Summary Ready Status (zero when any unmasked <a href="#">POWIRQMASK</a> power sources is not ready)

## 5.3.3. REGISTER: POWIRQMASK

Name	Bits	R/W	Reset	Description
MSVIO	0	RW	0	IO Voltage Large Enough (not Brownout) Interrupt Mask
MSBEVMODEM	1	RW	0	Modem Domain Voltage Brownout Error Interrupt Mask
MSBEVANA	2	RW	0	Analog Domain Voltage Brownout Error Interrupt Mask
MSVMODEM	3	RW	0	Modem Domain Voltage Regulator Ready Interrupt Mask
MSVANA	4	RW	0	Analog Domain Voltage Regulator Ready Interrupt Mask
MSVREF	5	RW	0	Reference Voltage Regulator Ready Interrupt Mask
MSREF	6	RW	0	Reference Ready Interrupt Mask
MPWRGOOD	7	RW	0	If 0, interrupt whenever one of the unmasked power sources fail (clear interrupt by reading <a href="#">POWSTICKYSTAT</a> ); if 1, interrupt when all unmasked power sources are good

## 5.4. INTERRUPT CONTROL

## 5.4.1. REGISTER: IRQMASK1, IRQMASK0

Name	Bits	R/W	Reset	Description
IRQMFIFONOTEMPTY	0	RW	0	FIFO not empty interrupt enable
IRQMFIFONOTFULL	1	RW	0	FIFO not full interrupt enable
IRQMFIFOTHRCNT	2	RW	0	FIFO count > threshold interrupt enable
IRQMFIFOTHRFREE	3	RW	0	FIFO free > threshold interrupt enable
IRQMFIFOERROR	4	RW	0	FIFO error interrupt enable
IRQMPLLUNLOCK	5	RW	0	PLL lock lost interrupt enable
IRQMRADIOCTRL	6	RW	0	Radio Controller interrupt enable
IRQMPOWER	7	RW	0	Power interrupt enable
IRQMXTALREADY	8	RW	0	Crystal Oscillator Ready interrupt enable
IRQMWAKEUPTIMER	9	RW	0	Wakeup Timer interrupt enable
IRQMLPOSC	10	RW	0	Low Power Oscillator interrupt enable
IRQMGPADC	11	RW	0	GPADC interrupt enable
IRQMPLLNRNGDONE	12	RW	0	PLL autoranging done interrupt enable

--	--	--	--	--

Zero disables the corresponding interrupt, while one enables it.

#### 5.4.2. REGISTER: RADIOEVENTMASK1, RADIOEVENTMASK0

Name	Bits	R/W	Reset	Description
REVM DONE	0	RW	0	Transmit or Receive Done Radio Event Enable
REVM SETTLED	1	RW	0	PLL Settled Radio Event Enable
REVM RADIO STATE CHG	2	RW	0	Radio State Changed Event Enable
REVM RX PARAM SET CHG	3	RW	0	Receiver Parameter Set Changed Event Enable
REVM FRAME CLK	4	RW	0	Frame Clock Event Enable

#### 5.4.3. REGISTER: IRQINVERSION1, IRQINVERSION0

Name	Bits	R/W	Reset	Description
IRQINV FIFONOTEMPTY	0	RW	0	FIFO not empty interrupt inversion
IRQINV FIFONOTFULL	1	RW	0	FIFO not full interrupt inversion
IRQINV FIFOTHRCNT	2	RW	0	FIFO count > threshold interrupt inversion
IRQINV FIFOTHRFREE	3	RW	0	FIFO free > threshold interrupt inversion
IRQINV FIFOERROR	4	RW	0	FIFO error interrupt inversion
IRQINV PLLUNLOCK	5	RW	0	PLL lock lost interrupt inversion
IRQINV RADIOCTRL	6	RW	0	Radio Controller interrupt inversion
IRQINV POWER	7	RW	0	Power interrupt inversion
IRQINV XTALREADY	8	RW	0	Crystal Oscillator Ready interrupt inversion
IRQINV WAKEUP TIMER	9	RW	0	Wakeup Timer interrupt inversion
IRQINV LPOSC	10	RW	0	Low Power Oscillator interrupt inversion
IRQINV GPADC	11	RW	0	GPADC interrupt inversion
IRQINV PLLRNGDONE	12	RW	0	PLL autoranging done interrupt inversion

#### 5.4.4. REGISTER: IRQREQUEST1, IRQREQUEST0

Name	Bits	R/W	Reset	Description
IRQRQ FIFONOTEMPTY	0	R	–	FIFO not empty interrupt pending

Y				
IRQRQFIFONOTFULL	1	R	–	FIFO not full interrupt pending
IRQRFIFOTHRCNT	2	R	–	FIFO count > threshold interrupt pending
IRQRFIFOTHRFREE	3	R	–	FIFO free > threshold interrupt pending
IRQRFIFOERROR	4	R	–	FIFO error interrupt pending
IRQRQPLLUNLOCK	5	R	–	PLL lock lost interrupt pending
IRQRRADIOCTRL	6	R	–	Radio Controller interrupt pending
IRQRPOWER	7	R	–	Power interrupt pending
IRQRXTALREADY	8	R	–	Crystal Oscillator Ready interrupt pending
IRQRWAKEUPTIMER	9	R	–	Wakeup Timer interrupt pending
IRQRLPOSC	10	R	–	Low Power Oscillator interrupt pending
IRQRGPADC	11	R	–	GPADC interrupt pending
IRQRQPLLNRNGDONE	12	R	–	PLL autoranging done interrupt pending

#### 5.4.5. REGISTER: RADIOEVENTREQ1, RADIOEVENTREQ0

Name	Bits	R/W	Reset	Description
REVRDONE	0	RC	–	Transmit or Receive Done Radio Event Pending
REVRSETTLED	1	RC	–	PLL Settled Radio Event Pending
REVRRADIOSTATECHG	2	RC	–	Radio State Changed Event Pending
REVRRXPARAMSETCHG	3	RC	–	Receiver Parameter Set Changed Event Pending
REVRFRAMECLK	4	RC	–	Frame Clock Event Pending

The bits in this register are cleared upon reading this register.

### 5.5. MODULATION AND FRAMING

#### 5.5.1. REGISTER: MODULATION

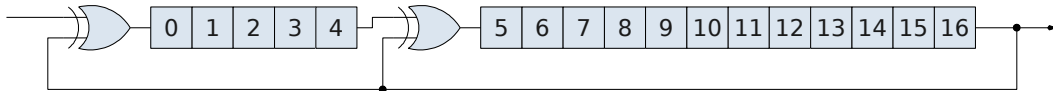
Name	Bits	R/W	Reset	Description
MODULATION	3:0	RW	1000	<div>Bits</div> <div>Meaning</div> <div>0000 ASK</div> <div>0001 ASK Coherent</div> <div>0100 PSK</div> <div>0110 OQSK</div> <div>0111 MSK</div>

				1000 FSK 1001 4-FSK 1010 AFSK 1011 FM
RX HALFSPEED	4	RW	0	if set, halves the receive bitrate

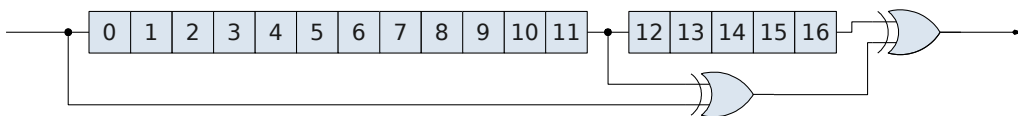
Transmitter amplitude shaping is set using the [MODCFGA](#) register, and frequency shaping is set using the [MODCFGF](#) register.

### 5.5.2. REGISTER: ENCODING

Name	Bits	R/W	Reset	Description
ENC INV	0	RW	0	Invert data if set to 1
ENC DIFF	1	RW	1	Differential Encode/Decode data if set to 1
ENC SCRAM	2	RW	0	Enable Scrambler / Descrambler if set to 1
ENC MANCH	3	RW	0	Enable manchester encoding/decoding. FM0/FM1 may be achieved by also appropriately setting ENC DIFF and ENC INV
ENC NOSYNC	4	RW	0	Disable Dibit synchronisation in 4-FSK mode



**Figure 17: Scrambler Schematic Diagram**



**Figure 18: Descrambler Schematic Diagram**

The intention of the scrambler is the removal of tones contained in the transmit data, i.e. to randomize the transmit spectrum. The scrambler polynomial is  $1 + X^{12} + X^{17}$ , it is therefore compatible to the K9NG/G3RUH Satellite Modems.

Figure 17 and Figure 18 show schematic diagrams of the scrambler and the descrambler operation. The numbered boxes represent delays by one bit.

ENC NOSYNC should normally be set to zero, unless the chip is either in the RXFRAMING or TXFRAMING mode and PWRUP is not used as a synchronisation signal.

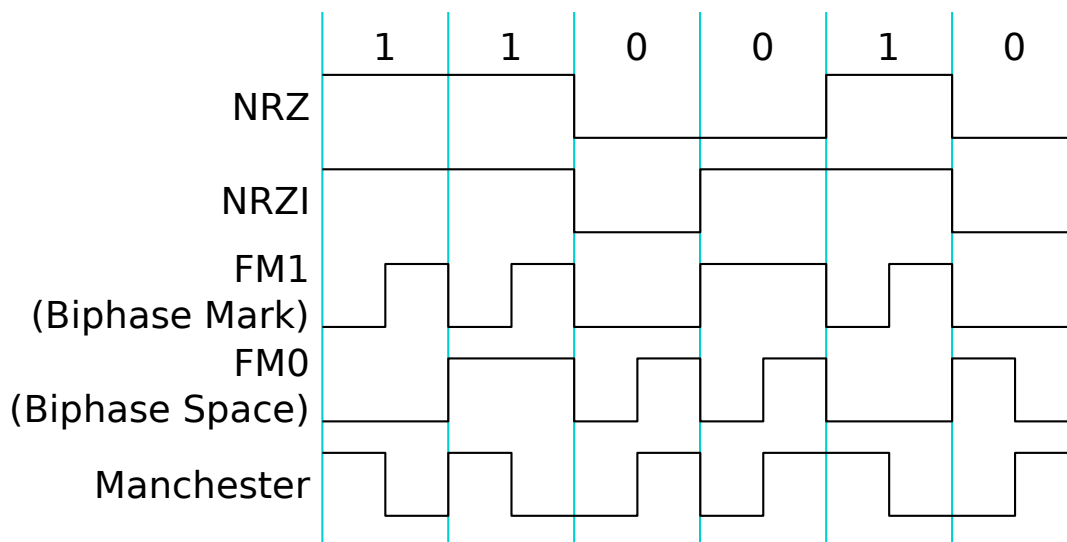


Figure 19: Customary Encodings

Figure 19 shows a few well known encoding formats used in telecom.

Name	Bits	Description
NRZ	INV=0, DIFF=0, SCRAM=0, MANCH=0	NRZ represents 1 as a high signal level, 0 as a low signal level. NRZ performs no change
NRZI	INV=1, DIFF=1, SCRAM=0, MANCH=0	NRZI represents 1 as no change in the signal level, and 0 as a change in the signal level. NRZI is recommended for HDLC <a href="#">[1]</a> . The HDLC bit stuffing ensures that there are periodic zeros and thus transitions, and the encoding is inversion invariant.
FM1	INV=1, DIFF=1, SCRAM=0, MANCH=1	FM1 (Biphase Mark) always ensures transitions at bit edges. It encodes 1 as a transition at the bit center, and 0 as no transition at the bit center.
FM0	INV=0, DIFF=1, SCRAM=0, MANCH=1	FM0 (Biphase Space) always ensures transitions at bit edges. It encodes 1 as no transition at the bit center, and 0 as a transition at the bit center.
Manchester	INV=0, DIFF=0, SCRAM=0, MANCH=1	Manchester encodes 1 as a 10 pattern, and 0 as a 01 pattern. Manchester is not inversion invariant.

Guidelines:

- Manchester, FM0, and FM1 are not recommended for new systems, as they double the bitrate
- In HDLC [\[1\]](#) mode, use NRZI, NRZI+Scrambler, or NRZ+Scrambler.
- In Raw modes, the choice depends on the legacy system to be implemented.

### 5.5.3. REGISTER: FRAMING

Name	Bits	R/W	Reset	Description
FABORT	0	S	0	Write 1 to abort current HDLC <a href="#">[1]</a> packet / pattern match
FRMMODE	3:1	RW	000	Bits Meaning 000 Raw 001 Raw, Soft Bits 010 HDLC <a href="#">[1]</a> 011 Raw, Pattern Match 100 Wireless M-Bus 101 Wireless M-Bus, 4-to-6 encoding
CRCMODE	6:4	RW	000	Bits Meaning 000 Off 001 CCITT (16bit) 010 CRC-16 011 DNP (16bit) 110 CRC-32
FRMRX	7	R	–	Packet start detected, receiver running; this bit is set when a flag is detected in HDLC <a href="#">[1]</a> mode or when the preamble matches in Raw Pattern Match mode. Cleared by writing 1 to FABORT

Note: The wireless M-Bus definition of “Manchester” is inverse to the definition used by the AX5043. AX5043 defines “Manchester” as the transmission of the data bit followed by the transmission of the inverted data bit. Wireless M-Bus defines it the other way around. In order to avoid having to enable inversion in the [ENCODING](#) register, the AX5043 inverts normal data bits when FRMMODE is set to Wireless M-Bus.

Note: If FRMMODE is set to Raw, Soft Bits, register F72 must be set to 0x06. Otherwise, it should be left or set to 0x00.

## 5.5.4. REGISTER: CRCINIT3, CRCINIT2, CRCINIT1, CRCINIT0

Name	Bits	R/W	Reset	Description
CRCINIT	31:0	RW	0xFFFFFFFF	CRC Reset Value; normally all ones

## 5.6. FORWARD ERROR CORRECTION

## 5.6.1. REGISTER: FEC

Name	Bits	R/W	Reset	Description
FECENA	0	RW	0	Enable FEC (Convolutional Encoder)
FECINPSHIFT	3:1	RW	000	Attenuate soft Rx Data by $2^{-\text{FECINPSHIFT}}$
FECPOS	4	RW	0	Enable noninverted Interleaver Synchronisation
FECNEG	5	RW	0	Enable inverted Interleaver Synchronisation
RSTVITERBI	6	RW	0	Reset Viterbi Decoder
SHORTMEM	7	RW	0	Shorten Backtrack Memory

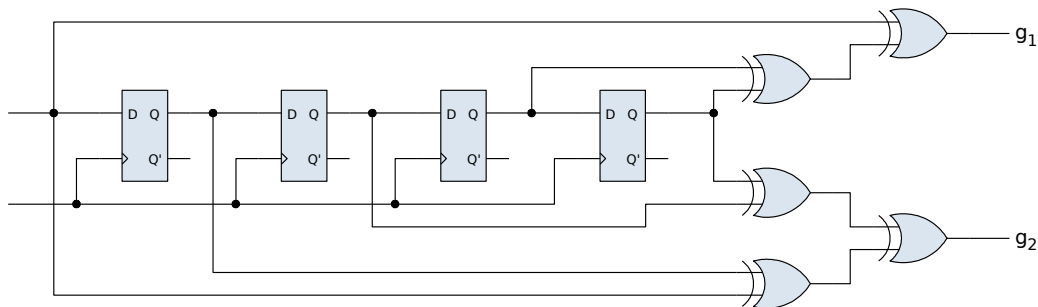


Figure 20: Schematic Diagram of the Convolutional Encoder

FECENA enables the Forward Error Correction and the Interleaver.

The Interleaver is a  $4 \times 4$  matrix interleaver, i.e. transmit bits are filled in row-wise and read out column-wise.

The Convolutional Code is a nonsystematic Rate  $\frac{1}{2}$  code with the generators  $g_1 = 1 + D^3 + D^4$  and  $g_2 = 1 + D + D^2 + D^4$ . It has a minimum free distance of  $d_{\text{free}} = 7$ . Figure 20 shows a schematic diagram of the convolutional encoder.



In the Transmitter, HDLC [1] flags are aligned (by inserting zero bits) to the interleaver. In the Receiver, a convolver to the encoded / interleaved flag sequence establishes deinterleaver synchronisation and inversion detection. That means, that FEC only works with HDLC framing.

The Viterbi decoder uses soft metric.

#### 5.6.2. REGISTER: FECSYNC

Name	Bits	R/W	Reset	Description
FECSYNC	7:0	RW	01100010	Interleaver Synchronisation Threshold

#### 5.6.3. REGISTER: FECSTATUS

Name	Bits	R/W	Reset	Description
MAXMETRIC	6:0	R	–	Metric increment of the survivor path
FEC INV	7	R	–	Inverted Synchronisation Sequence received

### 5.7. STATUS

#### 5.7.1. REGISTER: RADIOSTATE

Name	Bits	R/W	Reset	Description
RADIO STATE	3:0	R	0000	Radio Controller State Bits    Meaning 0000 Idle 0001 Powerdown 0100 Tx PLL Settling 0110 Tx 0111 Tx Tail 1000 Rx PLL Settling 1001 Rx Antenna Selection 1100 Rx Preamble 1 1101 Rx Preamble 2 1110 Rx Preamble 3 1111 Rx

## 5.7.2. REGISTER: XTALSTATUS

Name	Bits	R/W	Reset	Description
XTAL RUN	0	R	–	1 indicates crystal oscillator running and stable

## 5.8. PIN CONFIGURATION

## 5.8.1. REGISTER: PINSTATE

Name	Bits	R/W	Reset	Description
PSSYSCLK	0	R	–	Signal Level on Pin SYSCLK
PSDCLK	1	R	–	Signal Level on Pin DCLK
PSDATA	2	R	–	Signal Level on Pin DATA
PSIRQ	3	R	–	Signal Level on Pin IRQ
PSANTSEL	4	R	–	Signal Level on Pin ANTSEL
PSPWRAMP	5	R	–	Signal Level on Pin PWRAMP

## 5.8.2. REGISTER: PINFUNCSYSCLK

Name	Bits	R/W	Reset	Description
PFSYSCLK	4:0	RW	01000	<div> <div>Bits</div> <div>Meaning</div> </div> <div> 00000 SYSCLK Output '0'  00001 SYSCLK Output '1'  00010 SYSCLK Output 'Z'  00011 SYSCLK Output inverted <math>f_{XTAL}</math>  00100 SYSCLK Output <math>f_{XTAL}</math>  00101 SYSCLK Output <math>\frac{f_{XTAL}}{2}</math>  00110 SYSCLK Output <math>\frac{f_{XTAL}}{4}</math>  00111 SYSCLK Output <math>\frac{f_{XTAL}}{8}</math>  01000 SYSCLK Output <math>\frac{f_{XTAL}}{16}</math>  01001 SYSCLK Output <math>\frac{f_{XTAL}}{32}</math>  01010 SYSCLK Output <math>\frac{f_{XTAL}}{64}</math>  01011 SYSCLK Output <math>\frac{f_{XTAL}}{128}</math> </div>

				01100	SYSCLK Output $\frac{f_{XTAL}}{256}$
				01101	SYSCLK Output $\frac{f_{XTAL}}{512}$
				01110	SYSCLK Output $\frac{f_{XTAL}}{1024}$
				01111	SYSCLK Output Low Power (LP) Oscillator
				11111	SYSCLK Output Test Observation
PUSYSCLK	7	RW	0	SYSCLK weak Pullup enable	

### 5.8.3. REGISTER: PINFUNCCLK

Name	Bits	R/W	Reset	Description
PFDCLK	2:0	RW	100	Bits Meaning 000 DCLK Output '0' 001 DCLK Output '1' 010 DCLK Output 'Z' 011 DCLK Output Modem Data Clock Input; use when inputting/outputting framing data on DATA 100 DCLK Output Modem Data Clock Output; use when observing modem data on DATA 101 DCLK Output Modem Data Clock Output; use when inputting/outputting framing data on DATA, and you do not want to generate a clock yourself 110 invalid 111 DCLK Output Test Observation
PIDCLK	6	RW	0	DCLK inversion
PUDCLK	7	RW	0	DCLK weak Pullup enable

## 5.8.4. REGISTER: PINFUNCDATA

Name	Bits	R/W	Reset	Description
PFDATA	3:0	RW	0111	Bits Meaning 0000 DATA Output '0' 0001 DATA Output '1' 0010 DATA Output 'Z' 0011 DATA Input/Output Framing Data 0100 DATA Input/Output Modem Data 0101 DATA Input/Output Async Modem Data 0110 invalid 0111 DATA Output Modem Data 1111 DATA Output Test Observation
PIDATA	6	RW	0	DATA inversion
PUDATA	7	RW	1	DATA weak Pullup enable

In Asynchronous Wire Mode, the maximum bitrate is limited to  $\frac{f_{XTAL}}{32}$ .

## 5.8.5. REGISTER: PINFUNCIRQ

Name	Bits	R/W	Reset	Description
PFIRQ	2:0	RW	011	Bits Meaning 000 IRQ Output '0' 001 IRQ Output '1' 010 IRQ Output 'Z' 011 IRQ Output Interrupt Request 111 IRQ Output Test Observation
PIIRQ	6	RW	0	IRQ inversion
PUIRQ	7	RW	0	IRQ weak Pullup enable

## 5.8.6. REGISTER: PINFUNCANTSEL

Name	Bits	R/W	Reset	Description
PFANTSEL	2:0	RW	110	Bits Meaning 000 ANTSEL Output '0' 001 ANTSEL Output '1' 010 ANTSEL Output 'Z' 011 ANTSEL Output Baseband Tune Clock 100 ANTSEL Output External TCXO Enable

				101 ANTSEL Output DAC 110 ANTSEL Output Diversity Antenna Select 111 ANTSEL Output Test Observation
PIANTSEL	6	RW	0	ANTSEL inversion
PUANTSEL	7	RW	0	ANTSEL weak Pullup enable

#### 5.8.7. REGISTER: PINFUNCPWAMP

Name	Bits	R/W	Reset	Description
PFPWRAMP	3:0	RW	0110	Bits    Meaning 0000 PWRAMP Output '0' 0001 PWRAMP Output '1' 0010 PWRAMP Output 'Z' 0011 PWRAMP Input DiBit Synchronisation (4-FSK); use when inputting/outputting 4-FSK framing data on DATA 0100 PWRAMP Output DiBit Synchronisation (4-FSK); use when observing 4-FSK modem data on DATA 0101 PWRAMP Output DAC 0110 PWRAMP Output Power Amplifier Control 0111 PWRAMP Output External TCXO Enable 1111 PWRAMP Output Test Observation
PIPWRAMP	6	RW	0	PWRAMP inversion
PUPWRAMP	7	RW	0	PWRAMP weak Pullup enable

#### 5.8.8. REGISTER: PWRAMP

Name	Bits	R/W	Reset	Description
PWRAMP	0	RW	0	Power Amplifier Control

The PWRAMP bit may be output on the PWRAMP pin. This signal may be used to control an external power amplifier.

## 5.9. FIFO REGISTERS

## 5.9.1. REGISTER: FIFOSTAT

Name	Bits	R/W	Reset	Description
FIFO EMPTY	0	R	1	FIFO is empty if 1. This bit is dangerous to use when <a href="#">PWRMODE</a> is set to Receiver Wake-on-Radio mode. In this mode, the FIFO and thus the <a href="#">FIFOSTAT</a> register is only powered up while the FIFO is not empty, and powered down immediately when the FIFO becomes empty. When powered down, reading <a href="#">FIFOSTAT</a> returns zero, indicating a non-empty FIFO while in reality the FIFO is empty. In Wake-on-Radio mode, it is recommended to use the <a href="#">IRQRFIFONOTEMPTY</a> bit of Register <a href="#">IRQREQUEST0</a> . This bit will work in all cases, even when the interrupt is masked.
FIFO FULL	1	R	0	FIFO is full if 1
FIFO UNDER	2	R	0	FIFO underrun occurred since last read of <a href="#">FIFOSTAT</a> when 1
FIFO OVER	3	R	0	FIFO overrun occurred since last read of <a href="#">FIFOSTAT</a> when 1
FIFO CNT THR	4	R	0	1 if the FIFO count is > FIFOTHRESH
FIFO FREE THR	5	R	0	1 if the FIFO free space is > FIFOTHRESH
FIFOCMD	5:0	W	–	FIFO Command Bits      Meaning 000000    No Operation 000001    Clear FIFO Data Clear FIFO Error (OVER and UNDER) 000010    Flags 000011    Clear FIFO Data and Flags 000100    Commit 000101    Rollback 000110    Invalid 000111    Invalid 001XXX    Invalid 01XXXX    Invalid 1XXXXX    Invalid
FIFO AUTO COMMIT	7	RW	0	If one, FIFO write bytes are automatically committed on every write

### 5.9.2. REGISTER: FIFODATA

Name	Bits	R/W	Reset	Description
FIFODATA	7:0	RW	–	FIFO access register

Note that when accessing this register, the SPI address pointer is not incremented, allowing for efficient burst accesses

### 5.9.3. REGISTER: FIFOCOUNT1, FIFOCOUNT0

Name	Bits	R/W	Reset	Description
FIFOCOUNT	8:0	R	-----	Current number of committed FIFO Words

### 5.9.4. REGISTER: FIFOFREE1, FIFOFREE0

Name	Bits	R/W	Reset	Description
FIFOFREE	8:0	R	-----	Current number of empty FIFO Words

### 5.9.5. REGISTER: FIFOTHRESH1, FIFOTHRESH0

Name	Bits	R/W	Reset	Description
FIFOTHRESH	8:0	RW	000000000	FIFO Threshold

## 5.10. SYNTHESIZER

### 5.10.1. REGISTER: PLLLOOP, PLLLOOPBOOST

The PLLLOOP and PLLLOOPBOOST select PLL Loop Filter configuration for both normal mode and boosted mode. All fields in this register are separate, except for FREQSEL, which is common to both registers.

Name	Bits	R/W	Reset	Description
FLT FLTBOOST	1:0	RW	01 11	Bits Meaning 00 External Loop Filter 01 Internal Loop Filter, BW=100kHz for $I_{CP} = 68\mu A$ 10 Internal Loop Filter $\times 2$ , BW=200kHz for $I_{CP} = 272\mu A$ 11 Internal Loop Filter $\times 5$ , BW=500kHz for $I_{CP} = 1.7mA$
FILTEN FILTENBOOST	2	RW	0 0	Enable External Filter Pin
DIRECT DIRECTBOOST	3	RW	1 1	Bypass External Filter Pin
FREQSEL	7	RW	0	Frequency Register Selection; 0=use FREQA, 1=use FREQB

## 5.10.2. REGISTER: PLLCPI, PLLCPIBOOST

Name	Bits	R/W	Reset	Description
PLLCPI PLLCPIBOOST	7:0	RW	00001000 11001000	Charge pump current in multiples of 8.5 $\mu A$

## 5.10.3. REGISTER: PLLVCODIV

Name	Bits	R/W	Reset	Description
REFDIV	1:0	RW	00	Reference Divider Bits Meaning 00 $f_{PD} = f_{XTAL}$ 01 $f_{PD} = \frac{f_{XTAL}}{2}$ 10 $f_{PD} = \frac{f_{XTAL}}{4}$ 11 $f_{PD} = \frac{f_{XTAL}}{8}$
RFDIV	2	RW	0	RF divider: 0=no RF divider, 1=divide RF by 2
VCOSEL	4	RW	0	0=fully internal VCO1, 1=internal VCO2 with external inductor or external VCO, depending on VCO2INT
VCO2INT	5	RW	0	1=internal VCO2 with external Inductor, 0=external VCO



#### 5.10.4. REGISTER: PLLRANGINGA, PLLRANGINGB

Name	Bits	R/W	Reset	Description
VCORA VCORB	3:0	RW	1000 1000	VCO Range; depending on bit FREQSEL of <a href="#">PLLLOOP</a> , VCORA or VCORB is used
RNG START	4	RS	0	PLL Autoranging; Write 1 to start autoranging, bit clears when autoranging done. Autoranging always applies to the VCOR selected by FREQSEL of <a href="#">PLLLOOP</a> .
RNGERR	5	R	–	Ranging Error; Set when RNG START transitions from 1 to 0 and the programmed frequency cannot be achieved
PLL LOCK	6	R	–	PLL is locked if 1
STICKY LOCK	7	R	–	if 0, PLL lost lock after last read of PLLRANGINGA or PLLRANGINGB register

#### 5.10.5. REGISTER: FREQA3, FREQA2, FREQA1, FREQA0

Name	Bits	R/W	Reset	Description
FREQA	31:0	RW	0x3934CCCD	Frequency; $FREQA = \left\lfloor \frac{f_{CARRIER}}{f_{XTAL}} \cdot 2^{24} + \frac{1}{2} \right\rfloor$

It is not recommended to use an RF frequency that is an integer multiple of the reference frequency, due to stray RF desensitizing the receiver.

It is strongly recommended to always set bit 0 to avoid spectral tones.

#### 5.10.6. REGISTER: FREQB3, FREQB2, FREQB1, FREQB0

Name	Bits	R/W	Reset	Description
FREQB	31:0	RW	0x3934CCCD	Frequency; $FREQB = \left\lfloor \frac{f_{CARRIER}}{f_{XTAL}} \cdot 2^{24} + \frac{1}{2} \right\rfloor$

See notes of [FREQA](#) register.

### 5.11. SIGNAL STRENGTH

#### 5.11.1. REGISTER: RSSI

Name	Bits	R/W	Reset	Description
RSSI	7:0	R	–	Received Signal Strength, in dB

#### 5.11.2. REGISTER: BGNDRSSI

Name	Bits	R/W	Reset	Description
BGNDRSSI	7:0	RW	00000000	Background Noise (RSSI)

#### 5.11.3. REGISTER: DIVERSITY

Name	Bits	R/W	Reset	Description
DIVENA	0	RW	0	Antenna Diversity Enable
ANTSEL	1	RW	0	Antenna Select

DIVENA enables the internal antenna diversity logic.

The ANTSEL bit may be output on pin ANTSEL, and this signal may be used to control an external antenna switch.

#### 5.11.4. REGISTER: AGCCOUNTER

Name	Bits	R/W	Reset	Description
AGCCOUNTER	7:0	R	–	Current AGC Gain, in 0.75dB steps

### 5.12. RECEIVER TRACKING

#### 5.12.1. REGISTER: TRKDATARATE2, TRKDATARATE1, TRKDATARATE0

Name	Bits	R/W	Reset	Description
TRKDATARATE	23:0	R	–	Current datarate tracking value

#### 5.12.2. REGISTER: TRKAMPL1, TRKAMPL0

Name	Bits	R/W	Reset	Description
TRKAMPL	15:0	R	–	Current amplitude tracking value

### 5.12.3. REGISTER: TRKPHASE1, TRKPHASE0

Name	Bits	R/W	Reset	Description
TRKPHASE	11:0	R	–	Current phase tracking value

### 5.12.4. REGISTER: TRKRFFREQ2, TRKRFFREQ1, TRKRFFREQ0

Name	Bits	R/W	Reset	Description
TRKRFFREQ	19:0	RW	–	Current RF frequency tracking value

This Register is reset to zero when the demodulator is not running. In order to avoid write collisions between the demodulator and the microcontroller with undefined results, TRKFREQ should be frozen before attempting to write to. To freeze, set the RFFREQFREEZE bit in the appropriate [FREQGAIND0](#), [FREQGAIND1](#), [FREQGAIND2](#), or [FREQGAIND3](#) register, then wait for

$\frac{1}{4 \cdot \text{BAUDRATE}}$  for the freeze to take effect.

### 5.12.5. REGISTER: TRKFREQ1, TRKFREQ0

Name	Bits	R/W	Reset	Description
TRKFREQ	15:0	RW	–	Current frequency tracking value

The current frequency offset estimate is  $\Delta f = \frac{\text{TRKFREQ}}{2^{16}} \cdot \text{BITRATE}$

This Register is reset to zero when the demodulator is not running. In order to avoid write collisions between the demodulator and the microcontroller with undefined results, TRKFREQ should be frozen before attempting to write to. To freeze, set the FREQFREEZE bit in the appropriate [FREQGAINB0](#), [FREQGAINB1](#), [FREQGAINB2](#), or [FREQGAINB3](#) register, then wait for

$\frac{1}{4 \cdot \text{BAUDRATE}}$  for the freeze to take effect.

### 5.12.6. REGISTER: TRKFSKDEMOM1, TRKFSKDEMOM0

Name	Bits	R/W	Reset	Description
TRKFSKDEMOM	13:0	R	–	Current FSK demodulator value

### 5.12.7. REGISTER: TRKAFSKDEMOM1, TRKAFSKDEMOM0

Name	Bits	R/W	Reset	Description
TRKAFSKDEMOM	15:0	R	–	Current AFSK demodulator value

### 5.12.8. TRACKING REGISTER RESETS

Writes to [TRKAMPL1](#), [TRKAMPL0](#), [TRKPHASE1](#), [TRKPHASE0](#), [TRKDATARATE2](#), [TRKDATARATE1](#), [TRKDATARATE0](#) cause the following action:

Name	Bits	R/W	Reset	Description
DTRKRESET	3	W	–	Writing 1 clears the Datarate Tracking Register
ATRKRESET	4	W	–	Writing 1 clears the Amplitude Tracking Register
PTRKRESET	5	W	–	Writing 1 clears the Phase Tracking Register
RTRKRESET	6	W	–	Writing 1 clears the RF Frequency Tracking Register
FTRKRESET	7	W	–	Writing 1 clears the Frequency Tracking Register

## 5.13. TIMER

### 5.13.1. REGISTER: TIMER2, TIMER1, TIMER0

The main purpose of the fast  $\mu$ s Timer is to enable the microcontroller to exactly determine the packet start time. A snapshot of this timer at packet start can be written to the FIFO.

Name	Bits	R/W	Reset	Description
TIMER	23:0	R	–	1MHz ( $\frac{f_{XTAL}}{16}$ ) Counter; starts counting as soon as modem voltage regulator and Crystal Oscillator running

## 5.14. WAKEUP TIMER

The wakeup timer is a low power timer that can generate periodic events. It can generate a microcontroller interrupt (register [IROMASK1](#)) or start the receiver in wake-on-radio mode (register [PWRMODE](#)). The interrupt can be cleared by reading or writing any wakeup timer register.

The wakeup timer is driven by the low power oscillator. At every low power oscillator clock edge, the [WAKEUPTIMER](#) register is incremented by 1. The counting frequency can be set to 640Hz or 10.24kHz (register [LPOSCCONFIG](#)).

Whenever the [WAKEUPTIMER](#) register matches the [WAKEUP](#) register, an event is signalled, and the [WAKEUPFREQ](#) register is added to the [WAKEUP](#) register, to prepare for the next wakeup event.

Since crystals often take a significant amount of time to start up, the crystal oscillator may be started early using the [WAKEUPXOEARLY](#) register.

#### 5.14.1. REGISTER: WAKEUPTIMER1, WAKEUPTIMER0

Name	Bits	R/W	Reset	Description
WAKEUPTIMER	15:0	R	----	Wakeup Timer

#### 5.14.2. REGISTER: WAKEUP1, WAKEUP0

Name	Bits	R/W	Reset	Description
WAKEUP	15:0	RW	0x0000	Wakeup Time

#### 5.14.3. REGISTER: WAKEUPFREQ1, WAKEUPFREQ0

Name	Bits	R/W	Reset	Description
WAKEUPFREQ	15:0	RW	0x0000	Wakeup Frequency; Zero disables Wakeup

#### 5.14.4. REGISTER: WAKEUPXOEARLY

Name	Bits	R/W	Reset	Description
WAKEUPXOEARLY	7:0	RW	0x00	Number of LPOSC clock cycles by which the Crystal Oscillator is woken up before the main receiver

### 5.15. RECEIVER PARAMETERS

#### 5.15.1. REGISTER: IFFREQ1, IFFREQ0

Name	Bits	R/W	Reset	Description
IFFREQ	15:0	RW	0x1327	IF Frequency; $IFFREQ = \left\lfloor \frac{f_{IF} \cdot f_{XTALDIV}}{f_{XTAL}} \cdot 2^{20} + \frac{1}{2} \right\rfloor$

Please use the AX\_RadioLab software to calculate the optimum IF frequency for given physical layer parameters.

## 5.15.2. REGISTER: DECIMATION

Name	Bits	R/W	Reset	Description
DECIMATION	6:0	RW	0001101	Filter Decimation factor; Filter Output runs at $f_{\text{BASEBAND}} = \frac{f_{\text{XTAL}}}{2^4 \cdot f_{\text{XTALDIV}} \cdot \text{DECIMATION}}$

The value 0 is illegal.

## 5.15.3. REGISTER: RXDATARATE2, RXDATARATE1, RXDATARATE0

Name	Bits	R/W	Reset	Description
RXDATARATE	23:0	RW	0x003D8A	RXDATARATE = $\left\lfloor \frac{2^7 \cdot f_{\text{XTAL}}}{f_{\text{XTALDIV}} \cdot \text{BITRATE} \cdot \text{DECIMATION}} + \frac{1}{2} \right\rfloor$

RXDATARATE - TIMEGAINx  $\geq 2^{12}$  should be ensured when programming. Otherwise, the hardware does it, but this may cause instability due to asymmetric timing correction.

## 5.15.4. REGISTER: MAXDROFFSET2, MAXDROFFSET1, MAXDROFFSET0

Name	Bits	R/W	Reset	Description
MAXDROFFSET	23:0	RW	0x00009E	MAXDROFFSET = $\left\lfloor \frac{2^7 \cdot f_{\text{XTAL}} \cdot \Delta \text{BITRATE}}{f_{\text{XTALDIV}} \cdot \text{BITRATE}^2 \cdot \text{DECIMATION}} + \frac{1}{2} \right\rfloor$

The maximum bitrate offset the receiver is able to tolerate can be specified by the parameter  $\Delta \text{BITRATE}$ . The receiver will be able to tolerate a data rate within the range  $\text{BITRATE} \pm \Delta \text{BITRATE}$ . The downside of increasing  $\Delta \text{BITRATE}$  is that the required preamble length increases. Therefore,  $\Delta \text{BITRATE}$  should only be chosen as large as the transmitters require. If the bitrate offset is less than approximately  $\pm 1\%$ , receiver bitrate tracking should be switched off completely by setting MAXDROFFSET to zero, to ensure minimum preamble length.

## 5.15.5. REGISTER: MAXRFOFFSET2, MAXRFOFFSET1, MAXRFOFFSET0

Name	Bits	R/W	Reset	Description
MAXRFOFFSET	19:0	RW	0x01687	$\text{MAXRFOFFSET} = \left\lfloor \frac{\Delta f_{\text{CARRIER}}}{f_{\text{XTAL}}} \cdot 2^{24} + \frac{1}{2} \right\rfloor$
FREQOFFSCORR	23	RW	0	Correct frequency offset at the first LO if this bit is one; at the second LO if this bit is zero

This register sets the maximum frequency offset the built-in Automatic Frequency Correction (AFC) should handle. Set it to the maximum frequency offset between Transmitter and Receiver. Enlarging this register increases the time needed for the AFC to achieve lock. The AFC can only achieve lock if the transmit signal partially passes through the receiver channel filter. This limits the practically usable range for the AFC circuit to approximately  $\pm\frac{1}{4}$  of the Filter Bandwidth. The acquisition and tracking range can be increased by increasing the Receiver Channel Filter Bandwidth, at the expense of slightly reducing the Sensitivity.

#### 5.15.6. REGISTER: FSKD MAX1, FSKD MAX0

Name	Bits	R/W	Reset	Description
FSKDEV MAX	15:0	RW	0x0080	Current FSK Demodulator Max Deviation

In manual mode, it should be set to  $3 \cdot 512 \cdot \frac{f_{DEVIAION}}{BAUDRATE}$ .

#### 5.15.7. REGISTER: FSKD MIN1, FSKD MIN0

Name	Bits	R/W	Reset	Description
FSKDEV MIN	15:0	RW	0xFF80	Current FSK Demodulator Min Deviation

In manual mode, it should be set to  $-3 \cdot 512 \cdot \frac{f_{DEVIAION}}{BAUDRATE}$ .

#### 5.15.8. REGISTER: AFSKSPACE1, AFSKSPACE0

Name	Bits	R/W	Reset	Description
AFSKSPACE	15:0	RW	0x0040	AFSK Space (0-Bit encoding) Frequency

For receive, the register should be computed as follows:

$$AFSKSPACE = \left\lfloor \frac{f_{AFSKSPACE} \cdot DECIMATION \cdot f_{XTALDIV} \cdot 2^{16}}{f_{XTAL}} + \frac{1}{2} \right\rfloor$$

For transmit, the register has a slightly different definition:

$$AFSKSPACE = \left\lfloor \frac{f_{AFSKSPACE} \cdot 2^{18}}{f_{XTAL}} + \frac{1}{2} \right\rfloor$$

## 5.15.9. REGISTER: AFSKMARK1, AFSKMARK0

Name	Bits	R/W	Reset	Description
AFSKMARK	15:0	RW	0x0075	AFSK Mark (1-Bit encoding) Frequency

For receive, the register should be computed as follows:

$$AFSKMARK = \left\lfloor \frac{f_{AFSKMARK} \cdot DECIMATION \cdot f_{XTALDIV} \cdot 2^{16}}{f_{XTAL}} + \frac{1}{2} \right\rfloor$$

For transmit, the register has a slightly different definition:

$$AFSKMARK = \left\lfloor \frac{f_{AFSKMARK} \cdot 2^{18}}{f_{XTAL}} + \frac{1}{2} \right\rfloor$$

## 5.15.10. REGISTER: AFSKCTRL

Name	Bits	R/W	Reset	Description
AFSKSHIFT	4:0	RW	00100	<p>AFSK Detector Bandwidth; AFSKSHIFT =</p> $2 \cdot \left\lceil \log_2 \left( \frac{f_{XTAL}}{2^5 \cdot BITRATE \cdot f_{XTALDIV} \cdot DECIMATION} \right) \right\rceil$ <p>3dB corner frequency of the AFSK detector filter is:</p> $f_c = \frac{f_{XTAL}}{2^5 \cdot \pi \cdot f_{XTALDIV} \cdot DECIMATION} \cdot \arccos \left( \frac{k^2 + 2 \cdot k - 2}{2 \cdot (k - 1)} \right)$ <p>with <math>k = 2^{\lfloor \frac{AFSKSHIFT}{2} \rfloor}</math></p>

## 5.15.11. REGISTER: AMPLFILTER

Name	Bits	R/W	Reset	Description
AMPLFILTER	3:0	RW	0000	<p>3dB corner frequency of the Amplitude (Magnitude) Lowpass Filter;</p> $f_c = \frac{f_{XTAL}}{2^5 \cdot \pi \cdot f_{XTALDIV} \cdot DECIMATION} \cdot \arccos \left( \frac{k^2 + 2 \cdot k - 2}{2 \cdot (k - 1)} \right)$ <p>with <math>k = 2^{\lfloor \frac{AMPLFILTER}{2} \rfloor}</math></p> <p>0000: Filter bypassed</p>



#### 5.15.12. REGISTER: FREQUENCYLEAK

Name	Bits	R/W	Reset	Description
FREQUENCYLEAK	3:0	RW	0000	Leakiness of the Baseband Frequency Recovery Loop (0000 = off)

#### 5.15.13. REGISTER: RXPARAMSETS

Name	Bits	R/W	Reset	Description
RXPS0	1:0	RW	00	RX Parameter Set Number to be used for initial settling
RXPS1	3:2	RW	00	RX Parameter Set Number to be used after Pattern 1 matched and before Pattern 0 match
RXPS2	5:4	RW	00	RX Parameter Set Number to be used after Pattern 0 matched
RXPS3	7:6	RW	00	RX Parameter Set Number to be used after a packet start has been detected

#### 5.15.14. REGISTER: RXPARAMCURSET

Name	Bits	R/W	Reset	Description
RXSI	1:0	R	--	RX Parameter Set Index (determines which RXPS is used)
RXSN	3:2	R	--	RX Parameter Set Number (=RXPS[RXSI(1:0)])
RXSI	4	R	-	RX Parameter Set Index (special function bit) RXSI Bits    Meaning 0XX        Normal Function (indirection via RXPS) 1X0        Coarse AGC 1X1        Baseband Offset Acquisition

#### 5.15.15. REGISTER: AGCGAIN0, AGCGAIN1, AGCGAIN2, AGCGAIN3

Name	Bits	R/W	Reset	Description
AGCATTACK0	3:0	RW	0100	AGC gain reduction speed
AGCATTACK1			0100	
AGCATTACK2			1111	
AGCATTACK3			1111	

AGCDECAY0	7:4	RW	1011	AGC gain increase speed
AGCDECAY1			1011	
AGCDECAY2			1111	
AGCDECAY3			1111	

The 3dB corner frequency of the AGC loop is:

$$f_{3dB} = \frac{f_{XTAL}}{2^5 \cdot \pi \cdot f_{XTALDIV}} \cdot \arccos \left( \frac{2 + 2^{1-AGC\{ATTACK|DECAY\}x} - 2^{-2 \cdot AGC\{ATTACK|DECAY\}x}}{2 + 2^{1-AGC\{ATTACK|DECAY\}x}} \right)$$

$$\approx \frac{f_{XTAL}}{2^5 \cdot \pi \cdot f_{XTALDIV}} \cdot (2^{-AGC\{ATTACK|DECAY\}x} - 2^{-1-2 \cdot AGC\{ATTACK|DECAY\}x})$$

The AGC{ATTACK|DECAY}x values can be computed from the 3dB corner frequency  $f_{3dB}$  as follows:

$$c = \cos \left( \frac{2^5 \cdot \pi \cdot f_{XTALDIV} \cdot f_{3dB}}{f_{XTAL}} \right)$$

$$AGC\{ATTACK|DECAY\}x = -\log_2(1 - c + \sqrt{c^2 - 4 \cdot c + 3}) \approx -\log_2 \left( 1 - \sqrt{1 - \frac{2^6 \cdot \pi \cdot f_{XTALDIV} \cdot f_{3dB}}{f_{XTAL}}} \right)$$

The recommended AGCATTACK setting is  $f_{3dB} \approx \frac{BITRATE}{10}$  for ASK, and  $f_{3dB} \approx BITRATE$  for (G)FSK.

The recommended AGCDECAY setting is  $f_{3dB} \approx \frac{BITRATE}{100}$  for ASK, and  $f_{3dB} \approx \frac{BITRATE}{10}$  for (G)FSK.

A value of 0xF in the AGC{ATTACK|DECAY}x disables AGC update. Thus, setting the [AGCGAIN0/AGCGAIN1/AGCGAIN2/AGCGAIN3](#) register to 0xFF completely freezes the AGC.

#### 5.15.16. REGISTER: AGCTARGET0, AGCTARGET1, AGCTARGET2, AGCTARGET3

Name	Bits	R/W	Reset	Description
AGCTARGET0	7:0	RW	01110110	The target ADC output average magnitude is $\frac{AGCTARGETx}{2^{\frac{16}{16}}}$ . Note that the ADC can produce magnitudes from 0...2 <sup>9</sup> -1.
AGCTARGET1			01110110	
AGCTARGET2			01110110	

AGCTARGET3			01110110	
------------	--	--	----------	--

#### 5.15.17. REGISTER: AGCAHYST0, AGCAHYST1, AGCAHYST2, AGCAHYST3

Name	Bits	R/W	Reset	Description
AGCAHYST0	2:0	RW	000	This field specifies Digital Threshold Range. It is (AGCAHYSTx+1)·3dB; If set to zero, the analog AGC always follows immediately. Increasing this value gives the AGC controller more leeway delay analog AGC following.
AGCAHYST1			000	
AGCAHYST2			000	
AGCAHYST3			000	

#### 5.15.18. REGISTER: AGCMINMAX0, AGCMINMAX1, AGCMINMAX2, AGCMINMAX3

Name	Bits	R/W	Reset	Description
AGCMAXDA0	6:4	RW	000	When the digital AGC attenuation exceeds its maximum value, it is reset to the value given in AGCMAXDAx, and the analog AGC gain is recomputed accordingly. This value is given in 3dB steps. Setting it to AGCAHYSTx causes “drag” AGC behaviour with minimum analog AGC steps (probably desirable); decreasing it causes less frequent but larger analog AGC steps
AGCMAXDA1			000	
AGCMAXDA2			000	
AGCMAXDA3			000	
AGCMINDA0	2:0	RW	000	When the digital AGC attenuation exceeds its minimum value, it is reset to the value given in AGCMINDAx, and the analog AGC gain is recomputed accordingly. This value is given in 3dB steps. Setting it to 000 causes “drag” AGC behaviour with minimum analog AGC steps (probably desirable); increasing it causes less frequent but larger analog AGC steps
AGCMINDA1			000	
AGCMINDA2			000	
AGCMINDA3			000	

#### 5.15.19. REGISTER: TIMEGAIN0, TIMEGAIN1, TIMEGAIN2, TIMEGAIN3

Name	Bits	R/W	Reset	Description
TIMEGAIN0E	3:0	RW	1000	Gain of the timing recovery loop; this is the exponent
TIMEGAIN1E			0110	
TIMEGAIN2E			0101	
TIMEGAIN3E			0101	

TIMEGAIN0M	7:4	RW	1111	Gain of the timing recovery loop; this is the mantissa
TIMEGAIN1M			1111	
TIMEGAIN2M			1111	
TIMEGAIN3M			1111	

$$TIMEGAIN_{xM}, TIMEGAIN_{xE} = \arg \min_{TIMEGAIN_{xM}, E} \left| \frac{RXDatarate}{TMGCorrFrac_x} - TIMEGAIN_{xM} \cdot 2^{TIMEGAIN_{xE}} \right|$$

TMGCorrFrac should be chosen at least 4. Larger values result in less sampling time jitter, but slower timing lock-in.

#### 5.15.20. REGISTER: DRGAIN0, DRGAIN1, DRGAIN2, DRGAIN3

Name	Bits	R/W	Reset	Description
DRGAIN0E	3:0	RW	0010	Gain of the datarate recovery loop; this is the exponent
DRGAIN1E			0001	
DRGAIN2E			0000	
DRGAIN3E			0000	
DRGAIN0M	7:4	RW	1111	Gain of the datarate recovery loop; this is the mantissa
DRGAIN1M			1111	
DRGAIN2M			1111	
DRGAIN3M			1111	

$$DRGAIN_{xM}, DRGAIN_{xE} = \arg \min_{DRGAIN_{xM}, E} \left| \frac{RXDatarate}{DRGCorrFrac_x} - DRGAIN_{xM} \cdot 2^{DRGAIN_{xE}} \right|$$

DRGCorrFrac should be chosen at least 64. Larger values result in less estimated datarate jitter, but slower datarate acquisition.

#### 5.15.21. REGISTER: PHASEGAIN0, PHASEGAIN1, PHASEGAIN2, PHASEGAIN3

Name	Bits	R/W	Reset	Description
PHASEGAIN0	3:0	RW	0011	Gain of the phase recovery loop
PHASEGAIN1			0011	
PHASEGAIN2			0011	
PHASEGAIN3			0011	

FILTERIDX0	7:6	RW	11	Decimation Filter Fractional Bandwidth, see the table below
FILTERIDX1			11	
FILTERIDX2			11	
FILTERIDX3			11	

This register does not normally need to be changed.

	<b>Relative Bandwidth</b> $\frac{f_{XTAL}}{2^{16} \cdot f_{XTALDIV} \cdot DECIMATION}$ <b>Hz</b>			
FILTERIDXx	-3dB BW	nominal BW	-10dB BW	-40dB BW
00	0.121399	0.150000	0.174805	0.256653
01	0.149475	0.177845	0.202759	0.284729
10	0.182373	0.210858	0.235718	0.317566
11	0.221497	0.250000	0.274780	0.356812

1. Fractional Filter Bandwidth

The relative bandwidths in the table above need to be multiplied with

$$\frac{f_{XTAL}}{2^{16} \cdot f_{XTALDIV} \cdot DECIMATION} \text{ to get the bandwidth in Hz.}$$

#### 5.15.22. REGISTER: FREQGAINA0, FREQGAINA1, FREQGAINA2, FREQGAINA3

Name	Bits	R/W	Reset	Description
FREQGAINA0	3:0	RW	1111	Gain of the baseband frequency recovery loop; the frequency error is measured with the phase detector
FREQGAINA1			1111	
FREQGAINA2			1111	
FREQGAINA3			1111	
FREQAMPLGATE0	4	RW	0	If set to 1, only update the frequency offset recovery loops if the amplitude of the signal is larger than half the maximum (or larger than the average amplitude)
FREQAMPLGATE1			0	
FREQAMPLGATE2			0	
FREQAMPLGATE3			0	

FREQHALFMODE0	5	RW	0	If 1, the Frequency offset wraps around from 0x1fff to -0x2000, and vice versa.
FREQHALFMODE1			0	
FREQHALFMODE2			0	
FREQHALFMODE3			0	
FREQMODULO0	6	RW	0	If 1, the Frequency offset wraps around from 0x3fff to -0x4000, and vice versa.
FREQMODULO1			0	
FREQMODULO2			0	
FREQMODULO3			0	
FREQLIM0	7	RW	0	If 1, limit Frequency Offset to -0x4000...0x3fff
FREQLIM1			0	
FREQLIM2			0	
FREQLIM3			0	

Set FREQGAINA0 = 15 and FREQGAINB0 = 31 to completely disable the baseband frequency recovery loop, setting its output to zero.

#### 5.15.23. REGISTER: FREQGAINB0, FREQGAINB1, FREQGAINB2, FREQGAINB3

Name	Bits	R/W	Reset	Description
FREQGAINB0	4:0	RW	11111	Gain of the baseband frequency recovery loop; the frequency error is measured with the frequency detector
FREQGAINB1			11111	
FREQGAINB2			11111	
FREQGAINB3			11111	
FREQAVG0	6	RW	0	Average the frequency offset of two consecutive bits; this is useful for 0101 preambles in FSK mode
FREQAVG1			0	
FREQAVG2			0	
FREQAVG3			0	
FREQFREEZE0	7	RW	0	Freeze the baseband frequency recovery loop if set
FREQFREEZE1			0	
FREQFREEZE2			0	
FREQFREEZE3			0	

Set FREQGAINA0 = 15 and FREQGAINB0 = 31 to completely disable the baseband frequency recovery loop, setting its output to zero.

#### 5.15.24. REGISTER: FREQGAINC0, FREQGAINC1, FREQGAINC2, FREQGAINC3

Name	Bits	R/W	Reset	Description
FREQGAINC0	4:0	RW	01010	Gain of the RF frequency recovery loop; the frequency error is measured with the phase detector
FREQGAINC1			01011	
FREQGAINC2			01101	
FREQGAINC3			01101	

Set FREQGAINC0 = 31 and FREQGAIND0 = 31 to completely disable the RF frequency recovery loop, setting its output to zero.

#### 5.15.25. REGISTER: FREQGAIND0, FREQGAIND1, FREQGAIND2, FREQGAIND3

Name	Bits	R/W	Reset	Description
FREQGAIND0	4:0	RW	01010	Gain of the RF frequency recovery loop; the frequency error is measured with the frequency detector
FREQGAIND1			01011	
FREQGAIND2			01101	
FREQGAIND3			01101	
RFFREQFREEZE0	7	RW	0	Freeze the RF frequency recovery loop if set
RFFREQFREEZE1			0	
RFFREQFREEZE2			0	
RFFREQFREEZE3			0	

Set FREQGAINC0 = 31 and FREQGAIND0 = 31 to completely disable the RF frequency recovery loop, setting its output to zero.

#### 5.15.26. REGISTER: AMPLGAIN0, AMPLGAIN1, AMPLGAIN2, AMPLGAIN3

Name	Bits	R/W	Reset	Description
AMPLGAIN0	3:0	RW	0110	Gain of the amplitude recovery loop
AMPLGAIN1			0110	
AMPLGAIN2			0110	
AMPLGAIN3			0110	

AMPLAGC0	6	RW	1	if 1, try to correct the amplitude register when AGC jumps. This is not perfect, though
AMPLAGC1			1	
AMPLAGC2			1	
AMPLAGC3			1	
AMPLAVG0	7	RW	0	if 0, the amplitude is recovered by a peak detector with decay; if 1, the amplitude is recovered by averaging
AMPLAVG1			0	
AMPLAVG2			0	
AMPLAVG3			0	

This register does not normally need to be changed.

#### 5.15.27. REGISTER: FREQDEV10, FREQDEV00, FREQDEV11, FREQDEV01, FREQDEV12, FREQDEV02, FREQDEV13, FREQDEV03

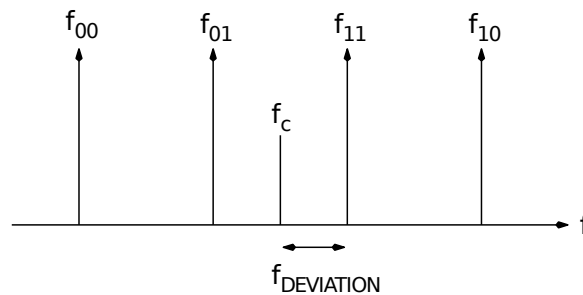
Name	Bits	R/W	Reset	Description
FREQDEV0	11:0	RW	0x020	Receiver Frequency Deviation; $FREQDEV_X = \left\lfloor \frac{f_{DEVIAION} \cdot 2^8 \cdot k_{SF}}{BITRATE} + \frac{1}{2} \right\rfloor$ ; $k_{SF}$ is a transmitter shaping and receiver filtering dependent constant. It is usually around $k_{SF} \approx 0.8$
FREQDEV1			0x020	
FREQDEV2			0x020	
FREQDEV3			0x020	

Enabling this feature ( $FREQDEV_X \neq 0$ ) can lead the frequency offset estimator to lock at the wrong offset. It is therefore recommended to enable it only after the frequency offset estimator is close to the correct offset (i.e.  $FREQDEV0 = 0$ ).

#### 5.15.28. REGISTER: FOURFSK0, FOURFSK1, FOURFSK2, FOURFSK3

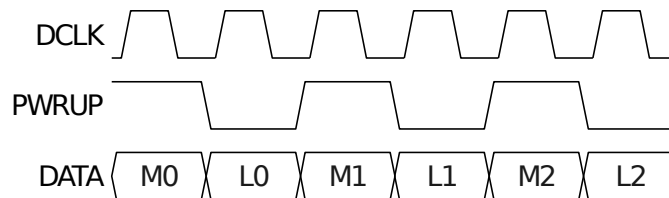
Name	Bits	R/W	Reset	Description
DEVDECAY0	3:0	RW	0110	Deviation Decay
DEVDECAY1			1000	
DEVDECAY2			1010	
DEVDECAY3			1010	
DEVUPDATE0	4	RW	1	Enable Deviation Update
DEVUPDATE1			1	
DEVUPDATE2			1	
DEVUPDATE3			1	





**Figure 21: 4-FSK Frequency Diagram**

In 4-FSK mode, two bits are transmitted together during each symbol, by using four frequencies instead of two. Figure 21 depicts the frequencies used.



**Figure 22: Wiremode Timing Diagram**

Wiremode is also available in 4-FSK mode, see Figure 22. The two bits that encode one symbol are serialized on the DATA pin. The PWRUP pin can be used as a synchronisation pin to allow symbol (dibit) boundaries to be reconstructed. DCLK is approximately but not exactly square. Gray encoding is used to reduce the number of bit errors in case of a wrong decision. The two bits encode the following frequencies:

$M_x$	$L_x$	Frequency
0	0	$f_{\text{CARRIER}} - 3 \cdot f_{\text{DEVIATION}}$
0	1	$f_{\text{CARRIER}} - f_{\text{DEVIATION}}$
1	1	$f_{\text{CARRIER}} + f_{\text{DEVIATION}}$
1	0	$f_{\text{CARRIER}} + 3 \cdot f_{\text{DEVIATION}}$

In framing mode, unless ENC NOSYNC in the [ENCODING](#) register is set, the shift register is synchronized to the dibit boundaries, and the pattern matches only at dibit boundaries. The shift register shifts right, so the bits end up in the FIFO word as follows:

7	6	5	4	3	2	1	0
$L_{n+3}$	$M_{n+3}$	$L_{n+2}$	$M_{n+2}$	$L_{n+1}$	$M_{n+1}$	$L_n$	$M_n$

In 4-FSK mode, it is no longer sufficient to compare the actual frequency with the center frequency and just record the sign. The frequency deviation of the transmitter must be known in order to choose the correct decision thresholds. This is the purpose of the [FSKMAX1](#), [FSKMAX0](#), [FSKMIN1](#) and [FSKMIN0](#) registers. These registers can either be set manually or recover the frequency deviation automatically. DEVUPDATE selects automatic mode if set to one, and manual mode if set to zero. Normally, automatic mode can be selected, but if the frequency deviation of the transmitter is exactly known at the receiver, manual mode can result in slightly better performance.

In automatic mode, [FSKMAX1](#), [FSKMAX0](#), [FSKMIN1](#) and [FSKMIN0](#) record the maximal and the minimal frequency seen at the receiver. “Leakage” or “gravity to zero” is added such that if these registers are disturbed by noise spikes, the effect decays. The amount of leakage is controlled by DEVDECAY.

DEVDECAY	# Samples to Decay to 0.5
0000	0
0001	1
0010	2
0011	5
0100	11
0101	22
0110	44
0111	88
1000	177
1001	355
1010	709
1011	1419
1100	2839
1101	5678
1110	11356
1111	22713

### 5.15.29. REGISTER: BBOFFSRES0, BBOFFSRES1, BBOFFSRES2, BBOFFSRES3

Name	Bits	R/W	Reset	Description
RESINTA0	3:0	RW	1000	Baseband Gain Block A Offset Compensation Resistors
RESINTA1	3:0	RW	1000	
RESINTA2	3:0	RW	1000	
RESINTA3	3:0	RW	1000	
RESINTB0	7:4	RW	1000	Baseband Gain Block B Offset Compensation Resistors
RESINTB1	7:4	RW	1000	
RESINTB2	7:4	RW	1000	
RESINTB3	7:4	RW	1000	

## 5.16. TRANSMITTER PARAMETERS

### 5.16.1. REGISTER: MODCFGF

This register selects the frequency shaping mode of the transmitter.

Name	Bits	R/W	Reset	Description
FREQSHAPE	1:0	RW	00	Bits Meaning 00 Unshaped (Rectangular) 01 Invalid 10 Gaussian BT=0.3 11 Gaussian BT=0.5

### 5.16.2. REGISTER: FSKDEV2, FSKDEV1, FSKDEV0

Name	Bits	R/W	Reset	Description
FSKDEV	23:0	RW	0x000A3D	(G)FSK Frequency Deviation; $FSKDEV = \left\lfloor \frac{f_{DEV IATION}}{f_{XTAL}} \cdot 2^{24} + \frac{1}{2} \right\rfloor$

Note that  $f_{DEV IATION}$  is actually half the deviation. The mark frequency is  $f_{CARRIER} + f_{DEV IATION}$ , the space frequency is  $f_{CARRIER} - f_{DEV IATION}$ .

$$f_{DEV IATION} = \frac{h}{2} \cdot BITRATE .$$

In AFSK mode, the register has a slightly different definition:

$$FSKDEV = \left\lfloor \frac{0.858785 \cdot f_{DEVATION}}{f_{XTAL}} \cdot 2^{24} + \frac{1}{2} \right\rfloor$$

In FM mode, the register has a different definition. It defines the conditioning of the ADC values prior to applying them to the transmit amplitude or the frequency deviation.

Name	Bits	R/W	Reset	Description
FMSHIFT	2:0	RW	101	<p>These Bits scale the ADC value</p> <p>Bits Meaning</p> <p>000 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^{15}}</math></p> <p>001 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^{14}}</math></p> <p>010 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^{13}}</math></p> <p>011 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^{12}}</math></p> <p>100 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^{11}}</math></p> <p>101 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^{10}}</math></p> <p>110 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^9}</math></p> <p>111 FM: <math>f_{DEVATION} = \frac{\pm ADCFS \cdot f_{XTAL}}{2^8}</math></p>
FMINPUT	9:8	RW	10	<p>Input Selection</p> <p>Bits Meaning</p> <p>00 GPADC13</p> <p>01 GPADC1</p> <p>10 GPADC2</p> <p>11 GPADC3</p>
FMSEXT	14	RW	0	ADC Sign Extension
FMOFFS	15	RW	0	ADC Offset Subtract

### 5.16.3. REGISTER: MODCFG0

This register selects the amplitude shaping mode of the transmitter. Amplitude shaping is used even for constant modulus modulation such as FSK, to ramp up and down the transmitter at the beginning and the end of the transmission.

Name	Bits	R/W	Reset	Description
TXDIFF	0	RW	1	Enable Differential Transmitter
TXSE	1	RW	0	Enable Single Ended Transmitter
AMPLSHAPE	2	RW	1	Bits Meaning 0 Unshaped 1 Raised Cosine
SLOWRAMP	5:4	RW	00	Bits Meaning 00 Normal Startup (1 Bit Time) 01 2 Bit Time Startup 10 4 Bit Time Startup 11 8 Bit Time Startup
PTTLCK GATE	6	RW	0	If 1, disable transmitter if PLL loses lock
BROWN GATE	7	RW	0	If 1, disable transmitter if Brown Out is detected

If BROWN GATE is set, the transmitter is disabled whenever one (or more) of the SSVIO, SSBEVMODEM or SSBEVANA bits of the [POWSTICKYSTAT](#) register is zero. In order for this to work, the user must read the [POWSTICKYSTAT](#) after setting the [PWRMODE](#) register for transmission.

### 5.16.4. REGISTER: TXRATE2, TXRATE1, TXRATE0

Name	Bits	R/W	Reset	Description
TXRATE	23:0	RW	0x0028F6	Transmit Bitrate $TXRATE = \left\lfloor \frac{BITRATE}{f_{XTAL}} \cdot 2^{24} + \frac{1}{2} \right\rfloor$

In asynchronous wire mode,  $BITRATE < \frac{f_{XTAL}}{32}$ .

### 5.16.5. REGISTER: TXPWRCOEFFA1, TXPWRCOEFFA0

Name	Bits	R/W	Reset	Description
TXPWRCOEFFA	15:0	RW	0x0000	Transmit Predistortion $TXPWRCOEFFA = \left\lfloor \alpha_0 \cdot 2^{12} + \frac{1}{2} \right\rfloor$

See [TXPWRCOEFFB0](#) for an explanation.

#### 5.16.6. REGISTER: TXPWRCOEFFB1, TXPWRCOEFFB0

Name	Bits	R/W	Reset	Description
TXPWRCOEFFB	15:0	RW	0x0FFF	Transmit Predistortion $TXPWRCOEFFB = \left\lfloor \alpha_1 \cdot 2^{12} + \frac{1}{2} \right\rfloor$

The transmit predistortion circuit applies the following function to the output of the raised cosine amplitude shaping:

$$f(x) = \alpha_4 \cdot x^4 + \alpha_3 \cdot x^3 + \alpha_2 \cdot x^2 + \alpha_1 \cdot x + \alpha_0$$

x is the input from the raised cosine shaping circuit ( $0 \leq x \leq 1$ ), and the output f(x) drives the power amplifier (0 means no output power, 1 means maximum output power).

For conventional (non-predistorted output),  $\alpha_0 = \alpha_2 = \alpha_3 = \alpha_4 = 0$  and  $0 \leq \alpha_1 \leq 1$  controls the output power. If hard amplitude shaping is selected, both the raised cosine amplitude shaper and the predistortion is bypassed, and  $\alpha_1$  used.

#### 5.16.7. REGISTER: TXPWRCOEFFC1, TXPWRCOEFFC0

Name	Bits	R/W	Reset	Description
TXPWRCOEFFC	15:0	RW	0x0000	Transmit Predistortion $TXPWRCOEFFC = \left\lfloor \alpha_2 \cdot 2^{12} + \frac{1}{2} \right\rfloor$

See [TXPWRCOEFFB0](#) for an explanation.

#### 5.16.8. REGISTER: TXPWRCOEFFD1, TXPWRCOEFFD0

Name	Bits	R/W	Reset	Description
TXPWRCOEFFD	15:0	RW	0x0000	Transmit Predistortion $TXPWRCOEFFD = \left\lfloor \alpha_3 \cdot 2^{12} + \frac{1}{2} \right\rfloor$

See [TXPWRCOEFFB0](#) for an explanation.

#### 5.16.9. REGISTER: TXPWRCOEFFE1, TXPWRCOEFFE0

Name	Bits	R/W	Reset	Description
TXPWRCOEFFE	15:0	RW	0x0000	Transmit Predistortion $TXPWRCOEFFE = \left\lfloor \alpha_4 \cdot 2^{12} + \frac{1}{2} \right\rfloor$

See [TXPWRCOEFFB0](#) for an explanation.

### 5.17. PLL PARAMETERS

#### 5.17.1. REGISTER: PLLVCOI

Name	Bits	R/W	Reset	Description
VCOI	5:0	RW	010010	This field sets the bias current for both VCOs. The increment is 50μA for VCO1 and 10μA for VCO2.
VCOIE	7	RW	0	Enable manual VCOI

#### 5.17.2. REGISTER: PLLVCOIR

Name	Bits	R/W	Reset	Description
VCOIR	5:0	R	–	This field reflects the actual VCO current selected. If VCOIE (Register <a href="#">PLLVCOI</a> ) is selected, this field reads the same as VCOI (also Register <a href="#">PLLVCOI</a> ). Otherwise, the value reflects the automatic setting.

#### 5.17.3. REGISTER: PLLLOCKDET

Name	Bits	R/W	Reset	Description
LOCKDETDLY	1:0	RW	11	Bits Meaning 00 Lock Detector Delay 6ns 01 Lock Detector Delay 9ns 10 Lock Detector Delay 12ns 11 Lock Detector Delay 14ns
LOCKDETDLYM	2	RW	0	0=Automatic Lock Delay (determined by the currently active frequency register); 1=Manual Lock Delay (Bits LOCKDETDLY)
LOCKDETDLYR	7:6	R	–	Lock Detect Read Back (not valid in power down mode)

## 5.17.4. REGISTER: PLLRNGCLK

Name	Bits	R/W	Reset	Description
PLLRNGCLK	2:0	RW	011	Bits Meaning 000 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^8}$ 001 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^9}$ 010 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^8}$ 011 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^{11}}$ 100 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^{12}}$ 101 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^{13}}$ 110 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^{14}}$ 111 PLL Ranging Clock $f_{PLLNG} = \frac{f_{XTAL}}{2^{15}}$

$f_{PLLNG}$  should be less than one tenth of the loop filter bandwidth, to allow enough settling time.

## 5.18. CRYSTAL OSCILLATOR

## 5.18.1. REGISTER: XTALCAP

Name	Bits	R/W	Reset	Description
XTALCAP	7:0	RW	00000000	Load Capacitance Configuration Bits (5:0) Meaning 000000 3pF 000001 8.5pF 000010 9pF : : 110111 36pF : : 111111 40pF

For values XTALCAP(5:0) ≠ 0,  $C_L = 8\text{pF} + 0.5\text{pF} \cdot \text{XTALCAP}(5:0)$ .



## 5.19. BASEBAND

### 5.19.1. REGISTER: BBTUNE

Name	Bits	R/W	Reset	Description
BBTUNE	3:0	RW	1001	Baseband Tuning Value
BBTUNERUN	4	RW	0	Baseband Tuning Start

### 5.19.2. REGISTER: BBOFFSCAP

Name	Bits	R/W	Reset	Description
CAPINTA	2:0	RW	111	Baseband Gain Block A Offset Compensation Capacitors
CAPINTB	6:4	RW	111	Baseband Gain Block B Offset Compensation Capacitors

## 5.20. PACKET FORMAT

### 5.20.1. REGISTER: PKTADDRCFG

Name	Bits	R/W	Reset	Description
ADDR POS	3:0	RW	0000	Position of the address bytes
FEC SYNC DIS	5	RW	1	When set, disable FEC sync search during packet reception
CRC SKIP FIRST	6	RW	0	When set, the first byte of the packet is not included in the CRC calculation
MSB FIRST	7	RW	0	When set, each byte is sent MSB first; when cleared, each byte is sent LSB first

### 5.20.2. REGISTER: PKTLENCFG

Name	Bits	R/W	Reset	Description
LEN POS	3:0	RW	0000	Position of the length byte
LEN BITS	7:4	RW	0000	Number of significant bits in the length byte

The built-in packet length logic can support up to 255 byte packets. It is still possible to receive larger packets if packet length and, unless using HDLC, CRC is handled in the microprocessor firmware. In order to enable reception of arbitrary length packets, the following settings must be made:

- Register [PKTLENCFG](#) LEN BITS (bits 7:4) = 1111
- Register [PKTMAXLEN](#) = 0xFF
- Register [PKTACCEPTFLAGS](#) ACCPT LRGP (bit 5) = 1

#### 5.20.3. REGISTER: PKTLENOFFSET

Name	Bits	R/W	Reset	Description
LEN OFFSET	7:0	RW	0x00	Packet Length Offset

The receiver adds LEN OFFSET to the length byte. The value of (length byte + LEN OFFSET) counts every byte in the packet after the synchronization pattern, up to and excluding the CRC bytes, but including the length byte.

For example with PKTLENCFG = 0x80 and PKTLENOFFSET = 0x00 the receiver will correctly receive the following packet (b1, b2 and b3 being data bytes)

Mode specific Framing	0x04	B1	B2	B3	CRC
-----------------------	------	----	----	----	-----

With PKTLENCFG = 0x80 and PKTLENOFFSET = 0x01 the receiver will correctly receive the following packet

Mode specific Framing	0x03	B1	B2	B3	CRC
-----------------------	------	----	----	----	-----

With PKTLENCFG = 0x00 and PKTLENOFFSET = 0x03 the receiver will correctly receive the following packet without length byte

Mode specific Framing	B1	B2	B3	CRC
-----------------------	----	----	----	-----

The length offset is treated as a signed value; LEN OFFSET 0xff means the length offset is -1.

#### 5.20.4. REGISTER: PKTMAXLEN

Name	Bits	R/W	Reset	Description
MAX LEN	7:0	RW	0x00	Packet Maximum Length

#### 5.20.5. REGISTER: PKTADDR3, PKTADDR2, PKTADDR1, PKTADDR0

Name	Bits	R/W	Reset	Description
ADDR	31:0	RW	0x00000000	Packet Address

#### 5.20.6. REGISTER: PKTADDRMASK3, PKTADDRMASK2, PKTADDRMASK1, PKTADDRMASK0

Name	Bits	R/W	Reset	Description
ADDRMASK	31:0	RW	0x00000000	Packet Address Mask

### 5.21. PATTERN MATCH

#### 5.21.1. REGISTER: MATCH0PAT3, MATCH0PAT2, MATCH0PAT1, MATCH0PAT0

Name	Bits	R/W	Reset	Description
MATCH0PAT	31:0	RW	0x00000000	Pattern for Match Unit 0; LSB is received first; patterns of length less than 32 must be MSB aligned

#### 5.21.2. REGISTER: MATCH0LEN

Name	Bits	R/W	Reset	Description
MATCH0LEN	4:0	RW	00000	Pattern Length for Match Unit 0; The length in bits of the pattern is MATCH0LEN + 1
MATCH0RAW	7	RW	0	Select whether Match Unit 0 operates on decoded (after Manchester, Descrambler etc.) (if 0), or on raw received bits (if 1)

#### 5.21.3. REGISTER: MATCH0MIN

Name	Bits	R/W	Reset	Description
MATCH0MIN	4:0	RW	00000	A match is signalled if the received bitstream matches the pattern in less than MATCH0MIN positions. This can be used to detect inverted sequences.

#### 5.21.4. REGISTER: MATCH0MAX

Name	Bits	R/W	Reset	Description
MATCH0MAX	4:0	RW	11111	A match is signalled if the received bitstream matches the pattern in more than MATCH0MAX positions.

## 5.21.5. REGISTER: MATCH1PAT1, MATCH1PAT0

Name	Bits	R/W	Reset	Description
MATCH1PAT	15:0	RW	0x0000	Pattern for Match Unit 1; LSB is received first; patterns of length less than 16 must be MSB aligned

## 5.21.6. REGISTER: MATCH1LEN

Name	Bits	R/W	Reset	Description
MATCH1LEN	3:0	RW	0000	Pattern Length for Match Unit 1; The length in bits of the pattern is MATCH1LEN + 1
MATCH1RAW	7	RW	0	Select whether Match Unit 1 operates on decoded (after Manchester, Descrambler etc.) (if 0), or on raw received bits (if 1)

## 5.21.7. REGISTER: MATCH1MIN

Name	Bits	R/W	Reset	Description
MATCH1MIN	3:0	RW	0000	A match is signalled if the received bitstream matches the pattern in less than MATCH1MIN positions. This can be used to detect inverted sequences.

## 5.21.8. REGISTER: MATCH1MAX

Name	Bits	R/W	Reset	Description
MATCH1MAX	3:0	RW	1111	A match is signalled if the received bitstream matches the pattern in more than MATCH1MAX positions.

## 5.22. PACKET CONTROLLER

## 5.22.1. REGISTER: TMGTXBBOOST

Name	Bits	R/W	Reset	Description
TMGTXBBOOSTM	4:0	RW	10010	Transmit PLL Boost Time Mantissa
TMGTXBBOOSTE	7:5	RW	001	Transmit PLL Boost Time Exponent

The Transmit PLL Boost Time is  $TMGTXBBOOSTM \cdot 2^{TMGTXBBOOSTE} \mu s$ .

### 5.22.2. REGISTER: TMGTXSETTLE

Name	Bits	R/W	Reset	Description
TMGTXSETTLEM	4:0	RW	01010	Transmit PLL (post Boost) Settling Time Mantissa
TMGTXSETTLEE	7:5	RW	000	Transmit PLL (post Boost) Settling Time Exponent

The Transmit PLL (post Boost) Settling Time is  $TMGTXSETTLEM \cdot 2^{TMGTXSETTLEE} \mu s$ .

### 5.22.3. REGISTER: TMGRXBOOST

Name	Bits	R/W	Reset	Description
TMGRXBOOSTM	4:0	RW	10010	Receive PLL Boost Time Mantissa
TMGRXBOOSTE	7:5	RW	001	Receive PLL Boost Time Exponent

The Receive PLL Boost Time is  $TMGRXBOOSTM \cdot 2^{TMGRXBOOSTE} \mu s$ .

### 5.22.4. REGISTER: TMGRXSETTLE

Name	Bits	R/W	Reset	Description
TMGRXSETTLEM	4:0	RW	10100	Receive PLL (post Boost) Settling Time Mantissa
TMGRXSETTLEE	7:5	RW	000	Receive PLL (post Boost) Settling Time Exponent

The Receive PLL (post Boost) Settling Time is  $TMGRXSETTLEM \cdot 2^{TMGRXSETTLEE} \mu s$ .

### 5.22.5. REGISTER: TMGRXOFFSACQ

Name	Bits	R/W	Reset	Description
TMGRXOFFSACQM	4:0	RW	10011	Baseband DC Offset Acquisition Time Mantissa
TMGRXOFFSACQE	7:5	RW	011	Baseband DC Offset Acquisition Time Exponent

The Baseband DC Offset Acquisition Time is  $TMGRXOFFSACQM \cdot 2^{TMGRXOFFSACQE} \mu s$ .

### 5.22.6. REGISTER: TMGRXCOARSEAGC

Name	Bits	R/W	Reset	Description
TMGRXCOARSEAGC M	4:0	RW	11001	Receive Coarse AGC Time Mantissa
TMGRXCOARSEAGC E	7:5	RW	001	Receive Coarse AGC Time Exponent

The Receive Coarse AGC Time is  $TMGRXCOARSEAGCM \cdot 2^{TMGRXCOARSEAGCE} \mu s$ .

#### 5.22.7. REGISTER: TMGRXAGC

Name	Bits	R/W	Reset	Description
TMGRXAGCM	4:0	RW	00000	Receiver AGC Settling Time Mantissa
TMGRXAGCE	7:5	RW	000	Receiver AGC Settling Time Exponent

The Receiver AGC Settling Time is  $TMGRXAGCM \cdot 2^{TMGRXAGCE}$ . Whether this time is measured in Bits or  $\mu s$  is determined by bit RXAGC CLK in register [PKTMISCFLAGS](#).

#### 5.22.8. REGISTER: TMGRXRSSI

Name	Bits	R/W	Reset	Description
TMGRXRSSIM	4:0	RW	00000	Receiver RSSI Settling Time Mantissa
TMGRXRSSIE	7:5	RW	000	Receiver RSSI Settling Time Exponent

The Receiver RSSI Settling Time is  $TMGRXRSSIM \cdot 2^{TMGRXRSSIE}$ . Whether this time is measured in Bits or  $\mu s$  is determined by bit RXRSSI CLK in register [PKTMISCFLAGS](#).

#### 5.22.9. REGISTER: TMGRXPREAMBLE1

Name	Bits	R/W	Reset	Description
TMGRXPREAMBLE1M	4:0	RW	00000	Receiver Preamble 1 Timeout Mantissa
TMGRXPREAMBLE1E	7:5	RW	000	Receiver Preamble 1 Timeout Exponent

The Receiver Preamble 1 Timeout is  $TMGRXPREAMBLE1M \cdot 2^{TMGRXPREAMBLE1E}$  Bits.

#### 5.22.10. REGISTER: TMGRXPREAMBLE2

Name	Bits	R/W	Reset	Description
TMGRXPREAMBLE2M	4:0	RW	00000	Receiver Preamble 2 Timeout Mantissa
TMGRXPREAMBLE2E	7:5	RW	000	Receiver Preamble 2 Timeout Exponent

The Receiver Preamble 2 Timeout is  $TMGRXPREAMBLE2M \cdot 2^{TMGRXPREAMBLE2E}$  Bits.

#### 5.22.11. REGISTER: TMGRXPREAMBLE3

Name	Bits	R/W	Reset	Description
TMGRXPREAMBLE3M	4:0	RW	00000	Receiver Preamble 3 Timeout Mantissa
TMGRXPREAMBLE3E	7:5	RW	000	Receiver Preamble 3 Timeout Exponent

The Receiver Preamble 3 Timeout is  $TMGRXPREAMBLE3M \cdot 2^{TMGRXPREAMBLE3E}$  Bits.

#### 5.22.12. REGISTER: RSSIREFERENCE

Name	Bits	R/W	Reset	Description
RSSIREFERENCE	7:0	RW	0x00	RSSI Offset

This register adds a constant offset to the computed RSSI value. It is used to compensate for board effects.

#### 5.22.13. REGISTER: RSSIABSTHR

Name	Bits	R/W	Reset	Description
RSSIABSTHR	7:0	RW	0x00	RSSI Absolute Threshold

RSSI levels above this threshold indicate a busy channel.

#### 5.22.14. REGISTER: BGNDRSSIGAIN

Name	Bits	R/W	Reset	Description
BGNDRSSIGAIN	3:0	RW	0000	Background RSSI Averaging Time Constant

The background RSSI estimate [BGNDRSSI](#) is updated after antenna RSSI measurement. Antenna RSSI measurement is performed in state RSSI in the Receiver Timing Diagram Figure 12. The background RSSI estimate is updated only once if antenna selection is performed.

The update is performed as follows:

$$BGNDRSSI := BGNDRSSI + (RSSI - BGNDRSSI) \cdot 2^{-BGNDRSSIGAIN}$$

## 5.22.15. REGISTER: BGNDRSSITHR

Name	Bits	R/W	Reset	Description
BGNDRSSITHR	5:0	RW	000000	Background RSSI Relative Threshold

RSSI levels more than BGNDRSSITHR above the background RSSI level indicate a busy channel.

## 5.22.16. REGISTER: PKTCHUNKSIZE

Name	Bits	R/W	Reset	Description
PKTCHUNKSIZE	3:0	RW	0000	Maximum Packet Chunk Size Bits    Meaning 0000    invalid 0001    1 0010    2 0011    4 0100    8 0101    16 0110    32 0111    64 1000    96 1001    128 1010    160 1011    192 1100    224 1101    240 1110    invalid 1111    invalid

The PKTCHUNKSIZE limits the maximum chunk size in the FIFO. This number includes the flags byte and all data bytes, but not the chunk header and the chunk length byte. Packets larger than PKTCHUNKSIZE - 1 are split into multiple chunks.

## 5.22.17. REGISTER: PKTMISCFLAGS

Name	Bits	R/W	Reset	Description
RXRSSI CLK	0	RW	0	Clock source for RSSI settling timeout: 0=1μs, 1=Bit clock
RXAGC CLK	1	RW	0	Clock source for AGC settling timeout: 0=1μs, 1=Bit clock



BGND RSSI	2	RW	0	If 1, enable the calculation of the background noise/RSSI level
AGC SETTLE DET	3	RW	0	If 1, if AGC settling is detected, terminate settling before timeout
WOR MULTI PKT	4	RW	0	If 1, the receiver continues to be on after a packet is received in wake-on-radio mode; otherwise, it is shut down

#### 5.22.18. REGISTER: PKTSTOREFLAGS

Name	Bits	R/W	Reset	Description
ST TIMER	0	RW	0	Store Timer value when a delimiter is detected
ST FOFSS	1	RW	0	Store Frequency offset at end of packet
ST RFOFSS	2	RW	0	Store RF Frequency offset at end of packet
ST DR	3	RW	0	Store Datarate offset at end of packet
ST RSSI	4	RW	0	Store RSSI at end of packet
ST CRCB	5	RW	0	Store CRC Bytes. Normally, CRC bytes are discarded after checking. In HDLC <a href="#">[1]</a> mode, CRC bytes are always stored, regardless of this bit.
ST ANT RSSI	6	RW	0	Store RSSI and Background Noise Estimate at antenna selection time

#### 5.22.19. REGISTER: PKTACCEPTFLAGS

Name	Bits	R/W	Reset	Description
ACCPT RESIDUE	0	RW	0	Accept Packets with a nonintegral number of Bytes (HDLC <a href="#">[1]</a> only)
ACCPT ABRT	1	RW	0	Accept aborted Packets
ACCPT CRCF	2	RW	0	Accept Packets that fail CRC check
ACCPT ADDRFR	3	RW	0	Accept Packets that fail Address check
ACCPT SZF	4	RW	0	Accept Packets that are too long
ACCPT LRGP	5	RW	0	Accept Packets that span multiple FIFO chunks

### 5.23. GENERAL PURPOSE ADC

#### 5.23.1. REGISTER: GPADCCTRL

Name	Bits	R/W	Reset	Description
CH ISOL	0	RW	0	Isolate Channels by sampling common mode between channels
CONT	1	RW	0	Enable Continuous Sampling (period according to <a href="#">GPADCPERIOD</a> )
GPADC13	2	RW	0	Enable Sampling GPADC1-GPADC3
BUSY	7	RS	0	Conversion ongoing when 1; when writing 1, a single conversion is started

#### 5.23.2. REGISTER: GPADCPERIOD

Name	Bits	R/W	Reset	Description
GPADCPERIOD	7:0	RW	00111111	GPADC Sampling Period $f_{SR} = \frac{f_{XTAL}}{32 \cdot GPADCPERIOD}$

#### 5.23.3. REGISTER: GPADC13VALUE1, GPADC13VALUE0

Name	Bits	R/W	Reset	Description
GPADC13VALUE	9:0	R	---	GPADC13 Value

Reading this register clears the GPADC Interrupt.

### 5.24. LOW POWER OSCILLATOR CALIBRATION

#### 5.24.1. REGISTER: LPOSCCONFIG

Name	Bits	R/W	Reset	Description
LPOSC ENA	0	RW	0	Enable the Low Power Oscillator. If 0, it is disabled.
LPOSC FAST	1	RW	0	Select the Frequency of the Low Power Oscillator. 0=640Hz, 1=10.24kHz
LPOSC IRQR	2	RW	0	Enable LP Oscillator Interrupt on the Rising Edge
LPOSC IRQF	3	RW	0	Enable LP Oscillator Interrupt on the Falling Edge
LPOSC CALIBF	4	RW	0	Enable LP Oscillator Calibration on the Falling Edge

LPOSC CALIBR	5	RW	0	Enable LP Oscillator Calibration on the Rising Edge
LPOSC OSC DOUBLE	6	RW	0	Enable LP Oscillator Calibration Reference Oscillator Doubling
LPOSC OSC INVERT	7	RW	0	Invert LP Oscillator Clock

#### 5.24.2. REGISTER: LPOSCSTATUS

Name	Bits	R/W	Reset	Description
LPOSC EDGE	0	R	–	Enabled Low Power Oscillator Edge detected
LPOSC IRQ	1	R	–	Low Power Oscillator Interrupt Active

The EDGE and IRQ flags can be cleared by reading either the [LPOSCCONFIG](#), [LPOSCSTATUS](#), [LPOSCPER1](#) or [LPOSCPER0](#) register.

#### 5.24.3. REGISTER: LPOSCKFILT1, LPOSCKFILT0

Name	Bits	R/W	Reset	Description
LPOSCKFILT	15:0	RW	0x20C4	$k_{FILT}$ (Low Power Oscillator Calibration Filter Constant)

The maximum value of  $k_{FILT}$ , that results in quickest calibration (single cycle), but no jitter suppression, is:

$$k_{FILT} = \left\lceil \frac{21333\text{Hz} \cdot 2^{20}}{f_{XTAL}} \right\rceil$$

Smaller values of  $k_{FILT}$  result in longer calibration, but increased jitter suppression.

#### 5.24.4. REGISTER: LPOSCREF1, LPOSCREF0

Name	Bits	R/W	Reset	Description
LPOSCREF	15:0	RW	0x61A8	LP Oscillator Reference Frequency Divider; set to $\frac{f_{XTAL}}{640\text{Hz}}$

## 5.24.5. REGISTER: LPOSCFREQ1, LPOSCFREQ0

Name	Bits	R/W	Reset	Description
LPOSCFREQ	9:-2	RW	0x000	LP Oscillator Frequency Tune Value; in $\frac{1}{32}$ %.

## 5.24.6. REGISTER: LPOSCPER1, LPOSCPER0

Name	Bits	R/W	Reset	Description
LPOSCPER	15:0	R	–	Last measured LP Oscillator Period

## 5.25. DAC

## 5.25.1. REGISTER: DACVALUE1, DACVALUE0

Name	Bits	R/W	Reset	Description
DACVALUE	11:0	RW	0x000	DAC Value (signed) (if DACINPUT = 0000)
DACSHIFT	3:0	RW	0x0	DAC Input Shift (if DACINPUT != 0000)

## 5.25.2. REGISTER: DACCONFIG

Name	Bits	R/W	Reset	Description
DACINPUT	3:0	RW	0000	DAC Input Multiplexer Bits    Meaning 0000    DACVALUE 0001    TRKAMPLITUDE 0010    TRKRFFFREQUENCY 0011    TRKFREQUENCY 0100    FSKDEMOD 0101    AFSKDEMOD 0110    RXSOFTDATA 0111    RSSI 1000    SAMPLE_ROT_I 1001    SAMPLE_ROT_Q 1100    GPADC13 1101    invalid 1110    invalid 1111    invalid

DACCLKX2	6	RW	0	Enable DAC Clock Doubler if set to 1
DACPWM	7	RW	0	Select PWM mode if 1, otherwise $\Sigma\Delta$ mode

Note that in  $\Sigma\Delta$  mode, the output range is limited to the range  $\frac{1}{4} \dots \frac{3}{4} \cdot VDDIO$ , to ensure modulator stability. The input value  $-2^{11}$  results in  $\frac{1}{4} \cdot VDDIO$ , the input value  $2^{11}-1$  results in  $\frac{3}{4} \cdot VDDIO$ . In PWM mode, the output voltage range is  $0 \dots VDDIO$ .

## 5.26. PERFORMANCE TUNING REGISTERS

Registers with Addresses from 0xF00 to 0xFFF are performance tuning registers. Their optimum values are computed by AX\_RadioLab; this section only gives a rough overview of how they should be set. Do not read or write addresses not listed in the table below.

Addr	RX/TX	Description
F00	RX/TX	Set to 0x0F
F0C	RX/TX	Keep the default 0x00
F0D	RX/TX	Set to 0x03
F10	RX/TX	Set to 0x04 if a TCXO is used. If a crystal is used, set to 0x0D if the reference frequency (crystal or TCXO) is more than 43MHz, or to 0x03 otherwise
F11	RX/TX	Set to 0x07 if a crystal is connected to CLK16P/CLK16N, or 0x00 if a TCXO is used
F1C	RX/TX	Set to 0x07
F21	RX	Set to 0x5C
F22	RX	Set to 0x53
F23	RX	Set to 0x76
F26	RX	Set to 0x92
F30	RX	This register should be reset between WOR wake-ups. The reset value is the value read after successful packet reception or 0x3F if no packet has been received yet.
F31	RX	This register should be reset between WOR wake-ups. The reset value is the value read after successful packet reception or 0xF0 if no packet has been received yet.
F32	RX	This register should be reset between WOR wake-ups. The reset value is the value read after successful packet reception or 0x3F if no packet has been received yet.
F33	RX	This register should be reset between WOR wake-ups. The reset value is the

Addr	RX/TX	Description
		value read after successful packet reception or 0xF0 if no packet has been received yet.
F34	RX/TX	Set to 0x28 if RFDIV in register <a href="#">PLLVCODIV</a> is set, or to 0x08 otherwise
F35	RX/TX	Set to 0x10 for reference frequencies (crystal or TCXO) less than 24.8MHz ( $\rightarrow f_{XTALDIV}=1$ ), or to 0x11 otherwise ( $\rightarrow f_{XTALDIV}=2$ )
F44	RX/TX	Set to 0x24
F72	RX	Set to 0x06 if the framing mode is set to "Raw, Soft Bits" (register <a href="#">FRAMING</a> ), or to 0x00 otherwise

## 6. REFERENCES

- [1] Wikipedia. *High-Level Data Link Control*. <http://en.wikipedia.org/wiki/HDLC>.
- [2] Axsem AG. *AX5043 Datasheet*. <http://www.axsem.com>
- [3] Ross N. Williams. *A Painless Guide to CRC Error Detection Algorithms*.  
[http://www.ross.net/crc/download/crc\\_v3.txt](http://www.ross.net/crc/download/crc_v3.txt)



## 7. CONTACT INFORMATION

**AXSEM AG**

Oskar-Bider-Strasse 1  
CH-8600 Dübendorf  
SWITZERLAND

Phone +41 44 882 17 07

Fax +41 44 882 17 09

Email [sales@axsem.com](mailto:sales@axsem.com)

[www.axsem.com](http://www.axsem.com)

For further product related or sales information please visit our website or contact your local representative.

The specifications in this document are subject to change at AXSEM's discretion. AXSEM assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of products based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights. AXSEM makes no warranties, either expressed or implied with respect to the information and specifications contained in this document. AXSEM does not support any applications in connection with life support and commercial aircraft. Performance characteristics listed in this document are estimates only and do not constitute a warranty or guarantee of product performance. The copying, distribution and utilization of this document as well as the communication of its contents to others without expressed authorization is prohibited. Offenders will be held liable for the payment of damages. All rights reserved. Copyright © 2011, 2012, 2013, 2014 AXSEM AG