AND9354/D

AX5042 Programming Manual

ON Semiconductor®

www.onsemi.com

APPLICATION NOTE

OVERVIEW

AX5042 is true single chip low−power CMOS transceiver for use in SRD bands. The on−chip transceiver consists of a fully integrated RF front−end with modulator and demodulator and flexible communication controller. Base band data processing is implemented in an advanced and flexible communication controller that enables user friendly communication either via SPI interface or in direct wire mode.

Connecting the AX5042 to a Micro−Controller

The AX5042 can easily be connected to any micro−controller. The micro−controller communicates with the AX5042 via a register file that is implemented in the AX5042 and that can be accessed serially via an industry standard Serial Peripheral Interface (SPI) protocol. There are also a few dedicated signalling lines.

Power−up, Reset and Receive/Transmit switching can be performed via these dedicated lines or via the register file. Therefore, connecting these dedicated signals is optional.

Reset can be performed via a dedicated signalling line or via the register file. It is also safe to perform power−on reset using the SPI reset bit in the PWRMODE register, so the RESET_N line is strictly optional. If RESET_N is not used, it should be tied to VDD, and the micro−controller should perform a device reset using SPI as soon as it leaves reset.

The AX5042 supports three different modes:

Frame Mode

In Frame mode, the internal communication controller performs frame delimiting, and data is received and transmitted via a 3 level x 10 bit FIFO accessible via the register file. Figure 1 shows the corresponding diagram for frame mode. In frame mode, connecting the interrupt line is highly recommended, though not strictly required.

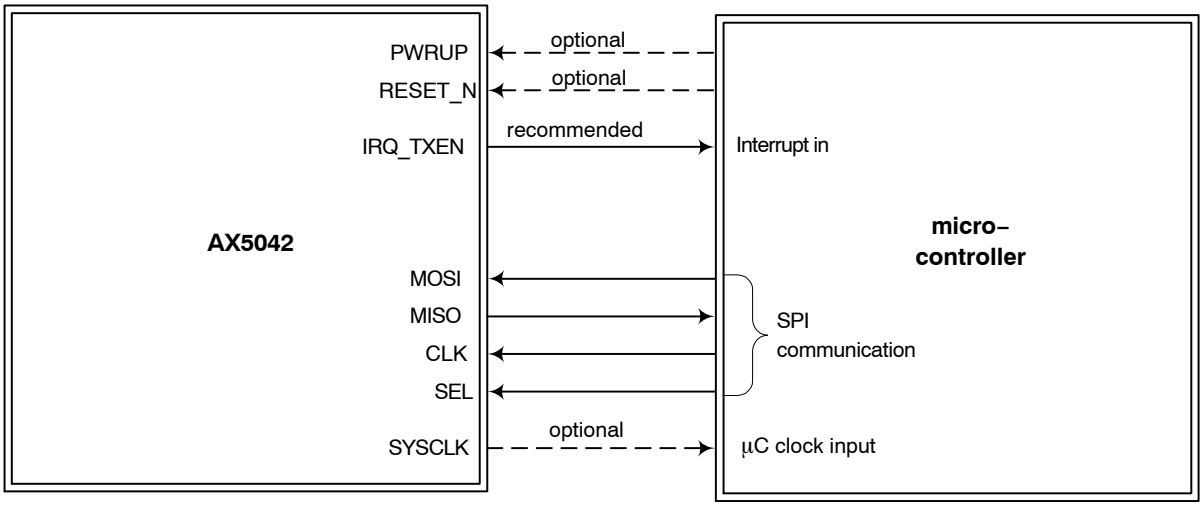

Figure 1. Frame Mode Connection Diagram

Synchronous Wire Mode

In Synchronous Wire mode, the internal communication controller is disabled, and the modem data is directly available on a dedicated pin (DATA). The modem also outputs the bit clock on a dedicated pin (DCLK), both during receive and transmit.

In synchronous wire mode, the AX5042 generates the bit clock both in receive and in transmit mode. Therefore, it is important that the micro−controller generates and receives data on the DATA pin synchronous to the clock on the DCLK pin.

Asynchronous Wire Mode

Asynchronous Wire mode works similar to synchronous wire mode, but in addition it performs RS232 start bit recognition and resynchronisation for transmit. It is therefore intended to be directly connected to an RS232 interface. In Asynchronous Wire mode the maximum bit rate is limited to $f_{xtal}/32$.

Figure 2 shows the wiring diagram between the AX5042 and the micro−controller in wire mode. Power−up, Reset and Receive/Transmit switching can be performed via dedicated lines or via the register file. Therefore, these dedicated signals are optional.

The SYSCLK pin may be used to clock the micro−controller, but otherwise is not required. In wire mode, transmit/receive data is available on the DATA line, so it must be connected. In asynchronous wire mode, the receive / transmit clock is available on the DCLK pin, but its usage is optional.

The AX5042 receive bit rate, the transmit bit rate, and the micro−controller RS232 interface baud rate must all be programmed to the same value. In transmit mode, the micro−controller must be programmed to transmit two stop bits (e.g. by setting the format to 8N2). In receive mode, the micro−controller must be programmed to accept only one stop bit (e.g. by setting the format to 8N1). The AX5042 synchronizer synchronizes the micro−controller RS232 interface clock to its bit clock by inserting or omitting stop bits.

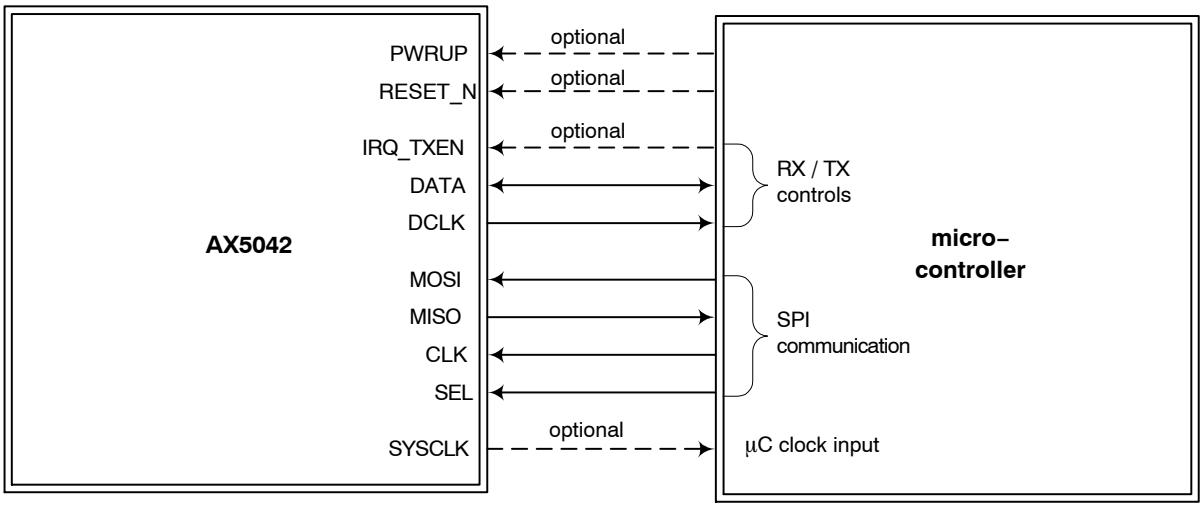

Figure 2. Wire Mode Connection Diagram

Pin Function Descriptions

Table 1. PIN FUNCTION DESCRIPTIONS

Symbol	Pin(s)	Type	Description
NC	1	N	Not to be connected
VDD	2	P	Power supply
GND	3	G	Ground
ANTP	4	A	Antenna input/output
ANTN	5	A	Antenna input/output
GND	6	P	Ground
VDD	7	P	Power supply
NC	8	N	Not to be connected
LPFILT	9	A	Pin for optional external synthesizer loop filter; leave unconnected if not used It is recommended to use the internal loop filter
NC	10	N	Not to be connected
GND	11	P	Ground
RESET_N	12	I	Optional reset input. If not used this pin must be connected to VDD.
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
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
DATA	18	I/O	In wire mode: Data input/output Can be programmed to be used as a general purpose I/O pin
IRQ_TXEN	19	I/O	In frame mode: Interrupt request output In wire mode: Transmit enable input Can be programmed to be used as a general purpose I/O pin
VDD	20	P	Power supply
DCLK	21	I/O	In wire mode: Clock output Can be programmed to be used as a general purpose I/O pin
GND	22	P	Ground
PWRUP	23	I/O	Power−up/−down input; activates/deactivates analog blocks Can be programmed to be used as a general purpose I/O pin If the power−up/−down functionality is handled in software and no usage as general purpose I/O pin is planned then this pin should be tied to VDD
NC	24	N	Not to be connected
NC	25	N	Not to be connected
VDD	26	P	Power supply
CLK16P	27	A	Crystal oscillator input/output
CLK16N	28	A	Crystal oscillator input/output

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

The center pad of the QFN28 package should be connected to GND.

SPI Register Access

Registers are accessed via a synchronous Serial Peripheral Interface (SPI). Most registers are 8 bit wide and accessed using the waveforms 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.

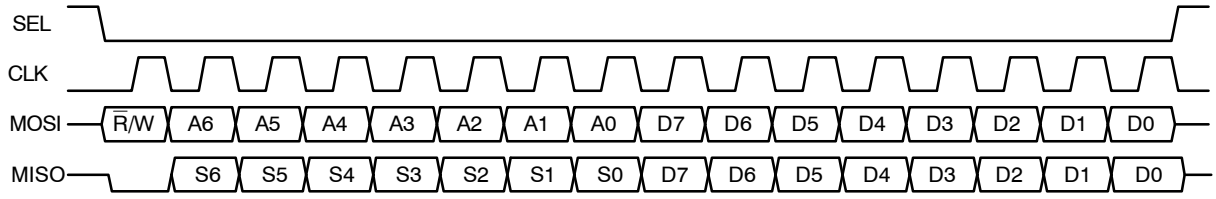

Figure 3. SPI 8 Bit Read/Write Access

It is necessary to deactivate and reactivate SEL between register accesses. Some registers perform preparatory actions on the falling edge of SEL and perform cleanup actions on the rising edge of SEL, so if SEL is left active between register accesses, some registers may fail.

Some device registers (TRKAMPL, TRKPHASE, TRKFREQ) are 16 bit registers that are continuously updated by the chip. These registers should not be accessed by two individual 8 bit accesses, as both halves may be inconsistent if the chip updates the register between the two accesses.

The chip therefore supports atomic 16 bit register read accesses. Figure 4 shows the 16 bit read waveform if the address of the high byte is supplied, and Figure 5 shows the waveform if the address of the low byte is supplied.

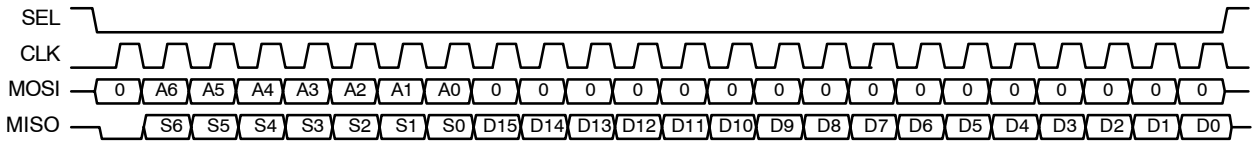

Figure 4. SPI 16 Bit Read Access, Most Significant Byte First

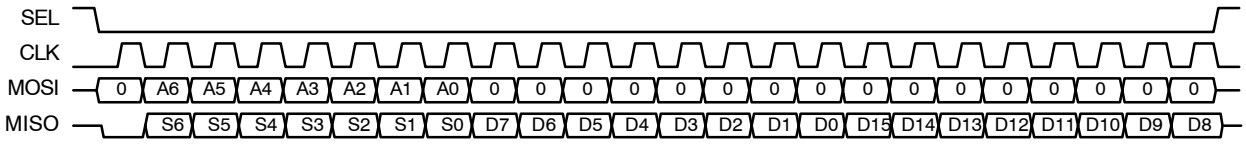

Figure 5. SPI 16 Bit Read Access, Least Significant Byte First

16 bit write accesses are not supported.

Status Bits

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. Table 2 shows which register bit is transmitted during the status timeslots.

Table 2. STATUS REGISTER BITS

SPI Bit Cell	Status	Register Bit
0	–	0
1	S6	PLL LOCK
2	S5	FIFO OVER
3	S4	FIFO UNDER
4	S3	FIFO FULL
5	S2	FIFO EMPTY
6	S1	FIFOSTAT(1)
7	S0	FIFOSTAT(0)

For information on the meaning of the status bits see the Transmit section of the next chapter as well as the description of the register FIFOCTRL in the Register Description section.

PROGRAMMING THE CHIP

The operation sequences of the chip can be controlled using the PWRMODE and APEOVER registers.

Table 3. PWRMODE AND APEOVER REGISTER STATES

PWRMODE Register	APEOVER Register	Name	Description	Typical Idd
0x00	0x80	POWERDOWN	All digital and analog functions, except the register file, are disabled. SPI registers are still accessible.	0.5 μA
0x60	0x00	STANDBY	The crystal oscillator is powered on; receiver and transmitter are off.	650 μA
0x00				
0x61	0x00	PWRUPPIN	The mode is determined by the state of the PWRUP and IRQ_TXEN pins. PWRUP = 0: Same function as POWERDOWN PWRUP = 1, IRQ_TXEN = 0: Same function as FULLRX PWRUP = 1, IRQ_TXEN = 1: Same function as FULLTX	0.5 μA 17 – 23 mA 13 – 37 mA
0x01				
0x68	0x00	SYNTHRX	The synthesizer is running on the receive frequency. Transmitter and receiver are still off. This mode is used to let the synthesizer settle on the correct frequency for receive.	12 mA
0x69	0x00	FULLRX	Synthesizer and receiver are running	17 – 23 mA
0x6C	0x00	SYNTHTX	The synthesizer is running on the transmit frequency. Transmitter and receiver are still off. This mode is used to let the synthesizer settle on the correct frequency for transmit.	11 mA
0x6D	0x00	FULLTX	Synthesizer and transmitter are running. Do not switch into this mode before the synthesizer has completely settled on the transmit frequency (in SYNTHTX mode), otherwise spurious spectral transmissions will occur.	13 – 37 mA

Alternatively the operation sequences of the chip can be controlled using the pins PWRUP and IRQ_TXEN if PWRMODE = 0x01. The use of the use of PWRUP and IRQ_TXEN pins to control transmission is however not recommended. Since there is no way to enter the SYNTHTX mode, the transmitter is switched on before the synthesizer is fully settled, thus producing spurious signals at various frequencies. To mitigate this, it is possible to first set TXPWR to 0, then PWRUP = 1 and IRQ_TXEN = 1 to turn the transmitter on, and then after the synthesizer settling time of 5 – 50 μs program the desired transmit power into TXPWR.

Table 4. PWRUP AND IRQ_TXEN PIN STATES

PWRUP Pin	IRQ_TXEN Pin	Name	Description	Typical Idd
0	X	POWERDOWN	All digital and analog functions, except the register file, are disabled. SPI registers are still accessible, but at a slower speed.	0.5 μA
1	0	FULLRX	Synthesizer and Receiver are running.	17 – 23 mA
1	1	FULLTX	Synthesizer and Transmitter are running. Do not switch on transmitter power (register TXPWR) before the synthesizer has settled, otherwise spurious spectral transmissions will occur.	13 – 37 mA

Figure 6 hows the basic programming flow chart of the device for transmitting, and Figure 7 for receiving.

1. Power up reference and oscillators: Set PWRMODE to STANDBY
   First, the on−chip references and the crystal oscillator are powered up, but the synthesizer is still powered down. Settling time of this phase is dominated by the crystal oscillator start−up time, which depends on the specific crystal used but is typically 3 ms.
2. Program parameters
   Then the desired modulation, carrier frequency and encoding is set (see section “Parameter Programming”). This can be done while the crystal oscillator is settling.

3. Power up synthesizer: Set PWRMODE to SYNTHTX (transmit mode) or SYNTHRX (receive mode)
   The settling time of the synthesizer is 5 – 50 μs depending on settings (see section AC Characteristics in the AX5042 Datasheet).
4. Auto−ranging
   After all the modulation parameters are set, the VCO in the synthesizer needs to be auto−ranged to the correct range setting. This is done using the auto−ranging procedure, for details see section: Synthesizer VCO Auto−Ranging. The auto−ranging needs to be performed if it has not been done in a previous RX/TX session, if the temperature or VDD have changed or if the frequency has changed.
5. Start transmitter/receiver: Set PWRMODE to FULLTX (transmit mode) or FULLRX (receive mode)
6. Power down: Set PWRMODE to POWERDOWN
   When transmission or reception is finished, the chip can be powered down.

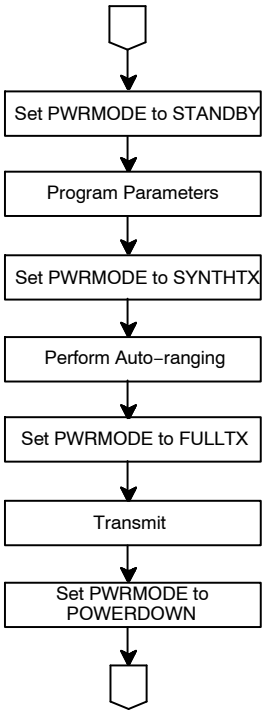

Figure 6. Transmit Flow Chart

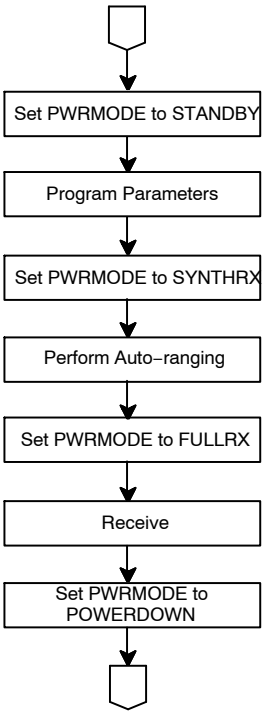

Figure 7. Receive Flow Chart

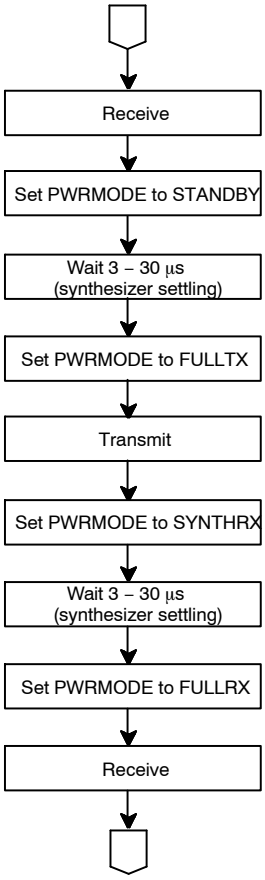

Figure 8. Receive Interrupted by Transmit Flow Chart

The register contents are preserved as long as the chip is powered, therefore, registers that do not change between receiving and transmitting do not need to be reprogrammed. Figure 8 shows the recommended sequence for transmitting packets during packet reception. This sequence avoids powering down the crystal oscillator and reference, thereby avoiding the start−up delays. The synthesizer VCO does not need to be re−auto−ranged, but, since this is not a zero IF receiver, the synthesizer needs 3 – 30 μs to settle on the correct frequency. The value depends on the synthesizer settings, see section AC Characteristics in the AX5042 Datasheet.

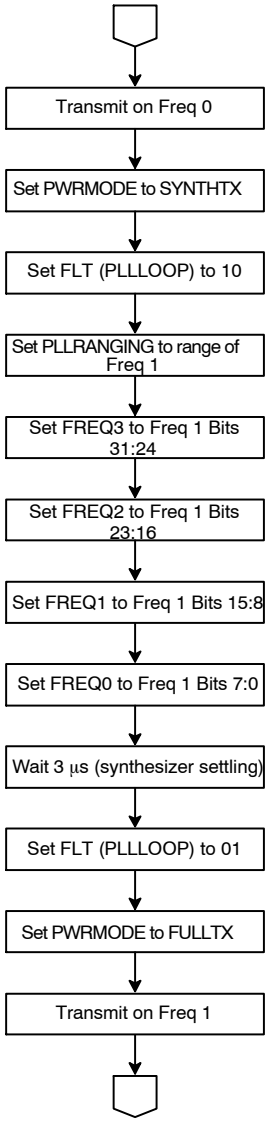

Figure 9. Transmit Frequency Change Flow Chart

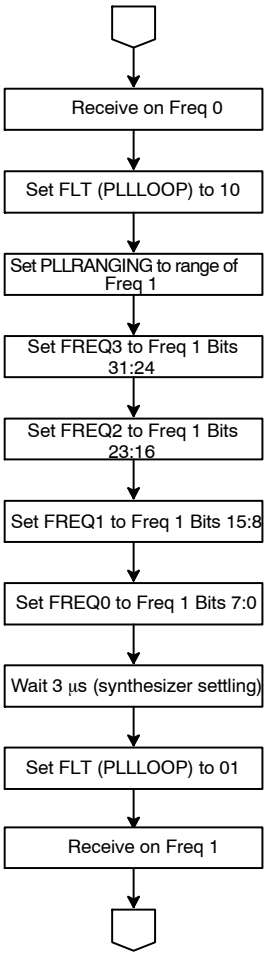

Figure 10. Receive Frequency Change Flow Chart

In Frequency Hopping systems, it is important to perform fast frequency changes. Figure 10 shows the recommended frequency change flow chart for frequency hopping receivers, while Figure 9 shows the recommended frequency change flow chart for frequency hopping transmitters.

These flow charts detail the recommended sequence to change the transmit/receive frequency. They do not detail the synchronization necessary to keep transmitter and receiver hopping schedules synchronous.

It is assumed that auto−ranging has been performed offline for all frequencies of the hopping schedule, and the auto−ranging results (VCOR bits of register PLLRANGING) have been stored in the micro−controller. For a detailed description of the synthesizer VCO auto−ranging see section: Synthesizer VCO Auto−Ranging.

In the transmit case, the transmitter must be disabled before starting the frequency change and must only be re−enabled once the synthesizer has settled on the new frequency, in order to avoid spurious transmissions. In the receive case, this is not necessary, the receiver can be left running.

Parameter Programming

Choosing the Fundamental Communication Characteristics

Table 5 lists the fundamental communication characteristics that need to be chosen before the device can be programmed.

Table 5. FUNDAMENTAL COMMUNICATION CHARACTERISTICS

Parameter	Description
f_{XTAL}	Frequency of the connected crystal in Hz
modulation	GFSK, FSK, MSK, GMSK, ASK, PSK or OQPSK (for recommendations see Table 6)
$f_{CARRIER}$	Carrier frequency (i.e. center frequency of the signal) in Hz
f_{IF}	Intermediate frequency in Hz, nominally 1 MHz
BITRATE	Desired bit rate, in bits/s
h	Modulation index, determines the frequency deviation for FSK and GFSK. $4 > h \geq 0.5$ for FSK, $f_{deviation} = 0.5 * h *$ BITRATE h = 0.5 for MSK, GMSK and OQPSK h = 0 for all other modulations
TMGCORRFRAC	Determines the timing recovery speed and the preamble length required The relationship between TMGCORRFRAC and the preamble length and is preamble length in bits = 3*TMGCORRFRAC, for details see section: Choosing the Preamble Duration Choose TMGCORRFRAC=32 for best noise performance at the expense of long synchronization time Choose TMGCORRFRAC=8 for faster synchronization time at the expense of noise performance Note that there is a lower bound for this value given in point 9 of section: Setting−up the Chip.
encoding	Inversion, differential, Manchester, scrambled, for recommendations see the description of the register ENCODING in the section: Overview and Table 16: Customary telecom modes description.

Table 6 gives an overview of the trade−offs between the different modulations that AX5042 offers, they should be considered when making a choice.

Table 6. MODULATION TRADE−OFFS

Modulation	Trade−offs
ASK	For bit rates up to 600 kbit/s The sensitivity for equivalent peak output power is 3 dB lower than for the other modulation types, as the average transmit power is only half the maximum transmit power. It is recommended to use shaped ASK for data transmissions, as the spectral efficiency is greatly improved vs. non−shaped ASK. For receive operation there is no difference between shaped and non−shaped.
FSK	For bit rates up to 200 kbit/s Frequency deviation is a free parameter
GFSK	For bit rates up to 200 kbit/s Gaussian shaped FSK, spectrally more efficient than FSK; GFSK with h = 0.5 is spectrally more efficient than MSK (which is FSK with h = 0.5). Frequency deviation is a free parameter
MSK	For bit rates up to 200 kbit/s Robust and spectrally efficient form of FSK (Modulation is the same as FSK with h = 0.5) Frequency deviation given by bit rate Slightly longer pre−ambles required than for FSK
GMSK	For bit rates up to 200 kbit/s Robust and spectrally efficient form of FSK (Modulation is the same as GFSK with h = 0.5) Frequency deviation given by bit rate Slightly longer pre−ambles required than for GFSK
PSK	For bit rates up to 600 kbit/s Slightly longer pre−ambles required than for FSK It is recommended to use shaped PSK for data transmissions, as the spectral efficiency is greatly improved vs. non−shaped PSK. For receive operation there is no difference between shaped and non−shaped.
OQPSK	For bit rates up to 200 kbit/s Very similar to MSK, with added precoding / postdecoding For new designs, use MSK instead

Table 7. PWRMODE REGISTER STATES

PWRMODE Register	Name	Description	Typical Idd
0000	POWERDOWN	All digital and analog functions, except the register file, are disabled. The core supply voltage is reduced to conserve leakage power. SPI registers are still accessible, but at a slower speed. FIFO access is possible.	0.25 μA
0100	VREGON	All digital and analog functions, except the register file, are disabled. The core voltage, however is at its nominal value for operation, and all SPI registers are accessible at the maximum speed.	140 μA
0101	STANDBY	The crystal oscillator is powered on; the transmitter is off.	500 μA
1100	SYNTHTX	The synthesizer is running on the transmit frequency. The transmitter is still off. This mode is used to let the synthesizer settle on the correct frequency for transmit.	10 mA
1101	FULLTX	Synthesizer and transmitter are running. Do not switch into this mode before the synthesizer has completely settled on the transmit frequency (in SYNTHTX mode), otherwise spurious spectral transmissions will occur.	11 – 45 mA

Figure 11 shows the basic programming flow chart of the device for transmitting.

7. Power up references and oscillators:
   Set PWRMODE to STANDBY
   First, the on−chip references and the crystal oscillator are powered up, but the synthesizer is still powered down. Settling time of this phase is dominated by the crystal oscillator start−up time, which depends on the specific crystal used but is typically 3 ms.
8. Program parameters
   Then the desired modulation, carrier frequency and encoding is set (see section “Parameter Programming”). This can be done while the crystal oscillator is settling.
9. Power up synthesizer: Set PWRMODE to SYNTHTX
   After all the modulation parameters are set, the synthesizer can be powered up. The settling time of the synthesizer is 5 – 50 μs depending on settings (see section AC Characteristics in the AX5031 Datasheet)
10. Auto−ranging
    After powering up, the VCO in the synthesizer needs to be auto−ranged to the correct range setting. This is done using the auto−ranging procedure, for details see section: Synthesizer VCO Auto−Ranging. The auto−ranging needs to be performed, if it has not been done in a previous TX session, if the temperature or VDD have changed or if the frequency has changed.
11. Start transmitter: Set PWRMODE to FULLTX
12. Power down: Set PWRMODE to POWERDOWN
    When transmission is finished, the chip can be powered down.

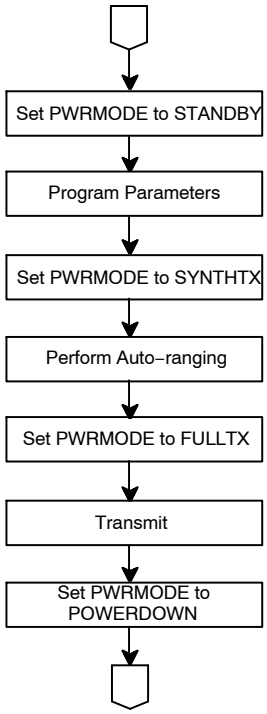

Figure 11. Transmit Flow Chart

The register contents are preserved as long as the chip is powered, therefore, registers that do not change between different transmit cycles do not need to be reprogrammed.

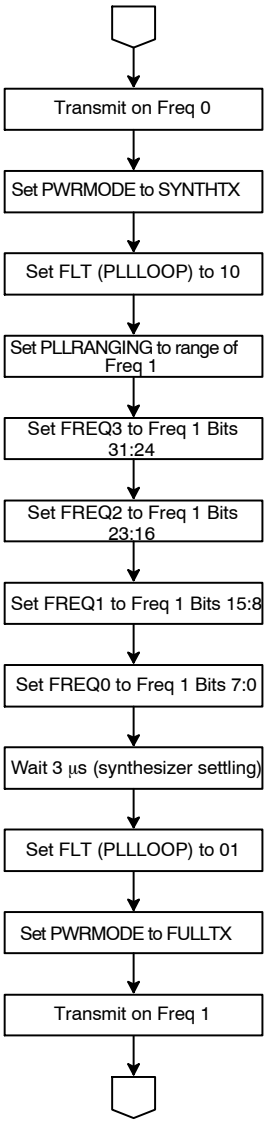

Figure 12. Transmit Frequency Change Flow Chart

In Frequency Hopping systems, it is important to perform fast frequency changes. Figure 12 shows the recommended frequency change flow chart for frequency hopping transmitters.

This flow chart details the recommended sequence to change the transmit frequency. It does not detail the synchronization necessary to keep transmitter and receiver hopping schedules synchronous.

It is assumed that auto−ranging has been performed offline for all frequencies of the hopping schedule, and the auto−ranging results (VCOR bits of register REGPLLRANGING) have been stored in the micro−controller.

The transmitter must be disabled before starting the frequency change and must only be re−enabled once the synthesizer has settled on the new frequency, in order to avoid spurious transmissions.

Parameter Programming

Choosing the Fundamental Communication Characteristics

Table 8 lists the fundamental communication characteristics that need to be chosen before the device can be programmed.

Table 9 gives an overview of the trade−offs between the different modulations that AX5031 offers, they should be considered when making a choice.

Table 8. FUNDAMENTAL COMMUNICATION CHARACTERISTICS

Parameter	Description
f_{XTAL}	Frequency of the connected crystal in Hz
modulation	FSK, MSK, ASK, PSK or OQPSK (for recommendations see Table 9: Modulation Trade−offs)
$f_{CARRIER}$	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, $f_{deviation} = 0.5 * h *$ BITRATE h = 0.5 for MSK and OQPSK h = 0 for all other modulations
encoding	Inversion, differential, manchester, scrambled, for recommendations see the description of the register ENCODING and Table 13: Customary telecom modes description.

Table 9. MODULATION TRADE−OFFS

Modulation	Trade−offs
ASK	For bit rates up to 2000 kbit/s The sensitivity for equivalent peak output power is 3 dB lower than for other modulation types, as the average transmit power is only half the maximum transmit power. It is recommended to use shaped ASK for data transmissions, as the spectral efficiency is greatly improved vs. non−shaped ASK.
FSK	For bit rates up to 350 kbit/s Frequency deviation is a free parameter
MSK	For bit rates up to 350 kbit/s Robust and spectrally efficient form of FSK (Modulation is the same as FSK with h=0.5) Frequency deviation given by bit rate The advantage of MSK over FSK is that it can be demodulated with higher sensitivity. Slightly longer pre−ambles required than for FSK.
PSK	For bit rates up to 2000 kbit/s Slightly longer pre−ambles required than for FSK. It is recommended to use shaped PSK for data transmissions, as the spectral efficiency is greatly improved vs. non−shaped PSK.
OQPSK	For bit rates up to 350 kbit/s Very similar to MSK, with added precoding / postdecoding For new designs, use MSK instead

Setting−up the Chip

The AX5042 should be programmed according to the following guide−line, for more detailed recommendations and descriptions see the corresponding register descriptions in the section Register Bank Description:

1. General set−up registers
   Set register AGCTARGET = 0x0E
   Set bit PLLARNG=1 in register PLLRNG, otherwise auto−ranging will not work correctly under all circumstances Set bits RXIMIX = 01 in register RXMISC These settings are mandatory for optimal performance of AX5042
2. Program the PLLLOOP register
   Bits FLT and PLLCPI must be set to program the synthesizer loop bandwidth. Recommended settings are given in Table 10. Bit BANDSEL is programmed to select the appropriate frequency band for $f_{carrier}$, set to 0 for 868/915 MHz band, set to 1 for 433 MHz band.

Table 10. RECOMMENDED SYNTHESIZER LOOP BANDWIDTH SETTINGS

Register Settings		Characteristics			
FLT	PLLCPI	Loop Bandwidth	Start−up Time	RX/TX Switch Time	Usage
01	010	100 kHz	25 μs	15 μs	Recommended setting for all modulations, all values of BITRATE, RX and TX Mandatory for FSK, GFSK, GMSK, MSK, OQPSK with BITRATE > 50 kHz
01	001	50 kHz	50 μs	30 μs	Use for TX if phase noise between 300 kHz and 1 MHz from carrier is critical Cannot be used for FSK, GFSK, GMSK, MSK, OQPSK with BITRATE > 50 kHz
11	010	200 kHz	12 μs	7 μs	Use to speed up start−up or switching Do not use for RX or TX Note that this setting will not work if an external loop filter is connected to LPFILT
10	010	500 kHz	5 μs	3 μs	Use to speed up start−up or switching Do not use for TX Note that this setting will not work if an external loop filter is connected to LPFILT

3. Program the frequency registers FREQ3, FREQ2, FREQ1 and FREQ0;
   $FREQ = [\ f_{CARRIER}/f_{XTAL} \cdot 2^{24} + ½\ ]$
   Ensure that the bit 0 of FREQ0 is set to one; this ensures that the built−in ΣΔ modulator does not exhibit tonal behaviour.
   NOTE: [x] denotes the floor function of the real number x. It returns the highest integer less than or equal x.
   For coding details and frequencies that are not selectable in 433 MHz band see the FREQ3, FREQ2, FREQ1, FREQ0 register description in section: Register Descriptions.
   Note that to program frequencies in the 433 MHz band registers FREQ3, FREQ2, FREQ1 and FREQ0 must be programmed to appropriate values and the bit BANDSEL in the PLLLOOP register must be set to 1.
4. Program the TXPWR register according to the desired output power
5. Program the IF frequency registers IFFREQHI and IFFREQLO
   $IFFREQLO = [\ f_{IF}/f_{XTAL} \cdot 2^{17} + ½\ ]$
6. Program the frequency deviation registers FSKDEV2, FSKDEV1 and FSKDEV0;
   $f_{DEVIATION} = h/2 \cdot BITRATE$
   $FSKDEV = [\ f_{DEVIATION}/f_{XTAL} \cdot 2^{24} + ½\ ]$
7. Program the transmit bit rate registers TXRATEHI, TXRATEMID and TXRATELO;
   $TXRATE = [\ BITRATE/f_{XTAL} \cdot 2^{24} + ½\ ]$
8. Program the receiver IF bandwidth registers CICDECHI and CICDECLO
   $CICDEC = [\ (1.5 \cdot f_{XTAL}) / (8 \cdot 1.2 \cdot BW)\ ]$, if $TMGCORRFRAC > 16$, or
   $CICDEC = [\ (1.5 \cdot f_{XTAL}) / (8 \cdot 1.4 \cdot BW)\ ]$, if $TMGCORRFRAC \le 16$,
   with $BW = (1 + h)\ BITRATE$
   Note that CICDEC must lie between $2 \le CICDEC \le 512$. If the above formulas result in a CICDEC less than 2, the chosen bandwidth is too high. Reduce the bit rate, or in the case of FSK, the modulation factor h. If the resulting CICDEC value is larger than 512, the chosen bandwidth is too narrow and not supported by the channel filter. Increase the bandwidth (set CICDEC to 512). The chip will work with BW > (1+h) BITRATE, at somewhat reduced sensitivity.
9. Determine the FSK over−sampling factor FSKMUL
   For modulations other than FSK, GFSK and GMSK, FSKMUL=1. For FSK, GFSK and GMSK, first, make sure TMGCORRFRAC fulfils the following inequality:
   $TMGCORRFRAC \ge f_{XTAL} / (4 \cdot BITRATE \cdot CICDEC)$
   Then compute FSKMUL:

$$FSKMUL = \left[\frac{1}{\frac{4 \cdot BITRATE \cdot CICDEC}{f_{XTAL}} + \frac{1}{TMGCORRFRAC}} \right]$$

   The resulting FSKMUL value must lie between 1 and 4 (inclusive). If FSKMUL > 4, then h is larger than the supported maximum value, i.e. the deviation is too large compared to the given bit rate. In this case h and thus also the deviation must be reduced.
10. Program the modulation register MODULATION according to Table 11.
    For FSK and GFSK use the calculation of FSKMUL to determine the correct FSK or GFSK over−sampling mode. Note that for RX operation there is no difference between shaped and non−shaped modulations. For GMSK chose GFSK and use h = 0.5.

Table 11. MODULATION REGISTER PROGRAMMING

MODULATION bits	FSKMUL	Meaning
0000	1	ASK
0010	1	ASK Shaped
0100	1	PSK
0101	1	PSK Shaped
0110	1	OQSK
0111	1	MSK
1000	1	FSK
1001	2	
1010	3	
1011	4	
1100	1	GFSK
1101	2	
1110	3	
1111	4	

11. Program the receiver bit rate registers DATARATEHI and DATARATELO

$$DATARATE = \left[\frac{2^{10} f_{XTAL}}{BITRATE \cdot CICDEC \cdot FSKMUL} + \frac{1}{2} \right]$$

12. Program the timing recovery dynamics registers TMGGAINHI and TMGGAINLO

$$TMGGAIN = \left[\frac{FSKMUL \cdot DATARATE}{TMGCORRFRAC} + \frac{1}{2} \right]$$

    DATARATE and TMGGAIN must fulfil the following inequality in order to function correctly:
    $DATARATE \ge TMGGAIN + 2^{12}$
    The bandwidth computation in point 8 above and the condition on TMGCORRFRAC in point 9 above ensure that this inequality holds.

13. Program the tracking loop dynamics registers PHASEGAIN, FREQGAIN, FREQGAIN2 and AMPLGAIN according to Table 12:

Table 12. TRACKING LOOP DYNAMICS REGISTER VALUES

Modulation	PHASEGAIN	FREQGAIN	FREQGAIN2	AMPLGAIN
ASK	0	6	6	6
PSK, MSK, OQPSK	3	6	6	6
(G)FSK, GMSK	3	3	6	6

14. Program the AGC dynamics registers AGCATTACK and AGCDECAY according to Table 13.

Table 13. AGC DYNAMICS REGISTER VALUES

Modulation	Register	Recommended Setting
ASK	AGCATTACK	$AGCATTACK = 27 + [\log_2 (BITRATE / (10 \cdot f_{XTAL}))]$
ASK	AGCDECAY	$AGCDECAY = 27 + [\log_2 (BITRATE / (100 \cdot f_{XTAL}))]$
(G)FSK, (G)MSK, (OQ)PSK	AGCATTACK	$AGCATTACK = 27 + [\log_2 (BITRATE / f_{XTAL})]$
(G)FSK, (G)MSK, (OQ)PSK	AGCDECAY	$AGCDECAY = 27 + [\log_2 (BITRATE / (10 \cdot f_{XTAL}))]$

15. Program the ENCODING register according to the desired bit encoding
16. Program the FRAMING register according to the desired framing mode
17. Program the IFMODE register according to the desired interfacing mode
18. Program the PINCFG1, PINCFG2, PINCFG3 according to the desired pin usage

Synthesizer VCO Auto−Ranging

Whenever the frequency or the environment (e.g. temperature, voltage) of the chip 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.

Figure 13 shows the flow chart of the auto−ranging process.

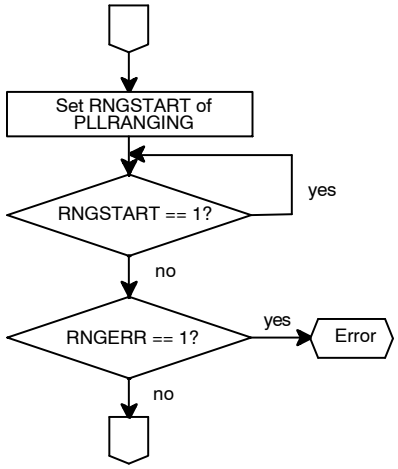

Figure 13. Synthesizer VCO Auto−Ranging Flow Chart

Before starting the auto−ranging, the frequency registers (FREQ3, FREQ2, FREQ1 and FREQ0) need to be programmed, and the chip should be in SYNTHRX or SYNTHTX mode.

Auto−ranging starts at the VCOR (register PLLRANGING) setting; if you already know the approximately correct synthesizer VCO range, you should set VCOR 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 VCOR to 8. Starting with VCOR < 6 should be avoided, as the initial synthesizer frequency can exceed the maximum frequency specification.

Furthermore, make sure that before starting the auto−ranging, the bit PLLARNG=1 in register PLLRNG,

otherwise auto−ranging will not work correctly under all circumstances. This setting can be done once at chip power−on.

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.

AFC

Commercial crystals only have a limited accuracy. Furthermore, since the crystal runs at a fraction of the RF carrier frequency, any crystal frequency offset is multiplied by the synthesizer by approximately a factor of 25 or 50, depending on the RF frequency band. While the receiver does have automatic frequency tracking, it can only track incoming signals that fall within its digital channel filter pass−band. It is therefore important to transmit and receive on the correct frequency. The smaller the bit rate, the higher the accuracy requirements for the reference crystal.

There are three primary methods to deal with frequency offset:

- Frequency Tracking
- Frequency Acquisition
- Factory Calibration

Frequency Tracking is automatically performed by the chip.

Whenever the frequency uncertainties are larger than the maximum tracking range of the frequency tracking logic, Frequency Acquisition and/or Factory Calibration may be used to augment Frequency Tracking.

As an example, consider a 433 MHz communication system utilizing a 16 MHz reference frequency with ±10 ppm frequency uncertainty. This translates into a RF carrier frequency uncertainty of 4.33 kHz. Since both the receiver and the transmitter will exhibit this uncertainty, the maximum frequency offset is ±8.66 kHz. For bit rates ≥ 40 kbit/s, the built in frequency tracking circuit is enough (the following section lists the maximum frequency offsets for frequency tracking). For lower bit rates, Frequency Acquisition, Factory Calibration or a better reference frequency accuracy must be used in addition to Frequency Tracking.

Frequency Tracking

The receiver contains circuitry to compensate for transmitter frequency offsets. This circuitry is fully automatic. The current frequency offset can be read out from the TRKFREQLO and TRKFREQHI registers. These registers are valid whenever the receiver is locked to a transmit signal.

The frequency tracking circuitry can compensate offsets up to approximately ±½ BITRATE in FSK mode, and up to approximately ±¼·BITRATE in PSK mode. In ASK mode, the frequency tracking circuitry is not used, the received signal must simply pass the receiver filter.

The frequency tracking logic can also be used to compensate for environmental conditions and crystal aging. To do this, the receiver should monitor frequency offsets over long timeframes. To make sure that a valid transmit signal is present, the receiver should read the tracking registers immediately after receiving a correct packet. If the observed frequency is consistently off the expected frequency over a longer timeframe, the micro−controller can assume that its crystal has drifted off and should compensate for the frequency change. Compensation should be performed by changing the frequency registers (FREQ3, FREQ2, FREQ1 and FREQ0) accordingly.

The exact algorithm for the frequency compensation varies widely with MAC protocol and other system considerations, but the following guidelines are recommended:

- In a peer−to−peer scenario with two stations, both stations should adjust only their receive frequency, to avoid instability of the whole system.
- In a master−slave system with higher quality masters, only the slaves should adjust both their receive and transmit frequencies.

Frequency Acquisition

Frequency Acquisition makes use of the on−chip Frequency Tracking hardware with progressively narrower bandwidths to widen the range of initial frequency offsets that can be dealt with.

One side transmits a long preamble (or even just an unmodulated carrier), during which the other side measures the frequency of this transmit signal. This frequency acquisition step can be performed during system setup, upon user interaction, or before each transmission.

On the receiver side, the frequency tracking circuit is used to measure the signal frequency. This is possible because the frequency tracking circuit works at approximately 10 dB lower signal levels than where data reception is possible. So the receiver should perform the following actions:

1. Set the receiver to FSK, bandwidth approximately 10 times the desired modulation bandwidth. FSK should be used during acquisition irrespective of the data transmission modulation. Also, DATARATE should be set to 0x1000, which results in a datarate being tied to the filter bandwidth, and having no relationship to the actual transmission datarate. Furthermore, TMGGAIN should be set to 0 to disable timing acquisition.
2. Wait until TRKFREQHI, TRKFREQLO is settled (see section: “Preamble” for the time required for TRKFREQHI, TRKFREQLO settling)
3. Read TRKFREQHI, TRKFREQLO, and compute the offset that needs to be applied to FREQ3, FREQ2, FREQ1, FREQ0
4. Repeat steps 1−3 with approximately 3 times the modulation bandwidth
5. Set the receiver to the modulation parameters, and start receiving

6. Check the received data for plausibility – start over if only garbage is received, as there may not have been a carrier transmitted during acquisition.

The AX5042−RNG Range Evaluation Kit contains example software to perform Frequency Acquisition.

Factory Calibration

Factory calibration cannot be done in−system as it involves the use of external measurement equipment.

In order to be able to calibrate the crystal, one needs to measure its frequency. The recommended method to measure its frequency is to use the SYSCLK pin, which can be programmed to output the crystal clock frequency (or a fraction of it). Directly probing the CLK16P or CLK16N pins is not recommended, as the load capacitance of the measuring equipment will change the frequency of the crystal.

An alternative method to measure the actual crystal frequency is to transmit an RF signal on a nominal frequency, and then measure the deviation of the actual transmit signal frequency from the nominal one with an RF counter. Measurements with a spectrum analyzer are generally not accurate enough.

Once the actual crystal frequency is known, it is recommended to correct for the crystal frequency deviation by changing the frequency registers (FREQ3, FREQ2, FREQ1 and FREQ0) accordingly.

Receive and Transmit

The chip offers two basic modes for receiving and transmitting data:

- Wire Mode can be seen as a UART interface and transmits all bits received over the air
- Frame Mode communicates over a SPI interface, and sends framed data from a dedicated FIFO

Wire Mode

In both the synchronous and the asynchronous wire mode, no registers need to be accessed during receive and transmit, once the FULLRX / FULLTX mode has been entered. Data is exchanged with the micro−controller or other circuitry using the dedicated pins DATA and DCLK.

Frame Mode

During receive and transmit, the software communicates with the receiver and the transmitter through a 10 bit wide and 3 levels deep FIFO.

Figure 14 shows the FIFO write process and Figure 15 shows the FIFO read process.

FIFO full, empty, overrun and underrun flags are also transmitted during the status phase of SPI transfers. See section: SPI Register Access and Table 2: Status Register Bits for details. FIFO flags may also be used to generate interrupts.

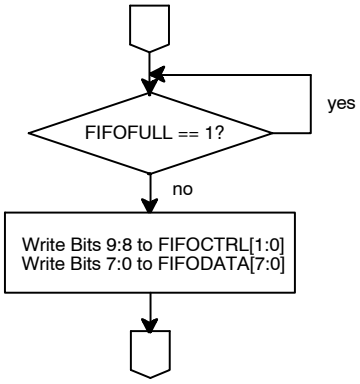

Figure 14. Write FIFO Flow Chart

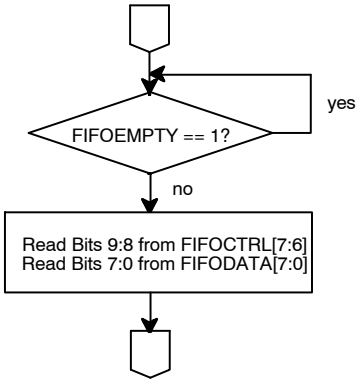

Figure 15. Read FIFO Flow Chart

Bits [7:0] are data information in both read and write. During a write access to the FIFO, Bits 9 and 8 hold the FIFOCMD[1:0] bits of the FIFOCTRL register. During a read access to the FIFO, Bits 9 and 8 are read from FIFOSTAT[1:0] of the FIFOCTRL register bits[7:6]. The function of these bits depends on the framing mode (for more information see following sections). The device offers two different framing modes, namely HDLC and 802.15.4 (ZigBee). Additionally, Raw Mode allows the implementation of legacy protocols in software. FIFO operation differs slightly depending on the framing mode.

Write Access:
Bits 9 and 8 hold the bits FIFOCMD[1:0] of the FIFOCTRL register during a write access to the FIFO.

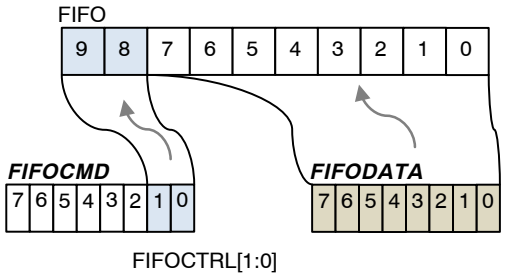

Figure 16.

Read Access:
During a read access to the FIFO Bits 9 and 8 are read from FIFOSTAT[1:0] of the FIFOCTRL register Bits[7:6].

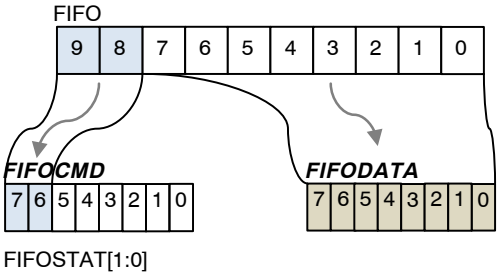

Figure 17.

HDLC

In HDLC mode, frames start and end with the bit pattern 01111110.

HDLC uses bit−stuffing: In order to ensure that no bit pattern inside the frame can be erroneously detected as a frame end, the transmitter inserts a 0 bit after five consecutive one bits; the receiver automatically removes those inserted 0 bits, making the process transparent to the user.

At the end of a HDLC frame, a checksum is transmitted. Seven or more consecutive one bits are treated as an ABORT, causing the current packet to be discarded. See [4] for a more elaborate description of HDLC.

In HDLC mode the meaning of the additional 2 bits in the 10 bit FIFO describe the content of FIFODATA[7:0]:

Table 14. HDLC MODE BITS

Bit [9:8]	Transmit FIFOCTRL[1:0]	Receive FIFOSTAT[1:0]
00	Data Byte (bit stuffed)	Data Byte
01	CRC Byte	Packet End (Data holds status information) Packet End is also an indication for Packet Start Status Information Bit[3]=1: CRC ok Bit[2:0]=110: full byte transfers only
10	Not used	Abort detected
11	RAW Byte (not bit−stuffing, CRC is initialized) Used for flags (e.g. EOF)	Abort detected

In transmit the bits [9:8] describe the type of data in the FIFODATA[7:0] to be transmitted. This controls the internal framing block and enables or disables bit stuffing for data or flags, respectively. It also initiates CRC calculation. However the flag content and the CRC bytes have to be written by the host processor according to the sequence shown in Figure 19. The number of CRC bytes has to be chosen according to the type of CRC chosen in the FRAMING register (16 bit or 32 bit). For CRC insertion it does not matter what is written in the CRC bytes, as the chip will calculate the CRC value and will change the values.

In receive the bits [9:8] describe the type of data received. If an end of packet delimiter flag is detected, the chip automatically evaluates the CRC and sets the bits [3:0] of the data in the flag to signal the result of the CRC.

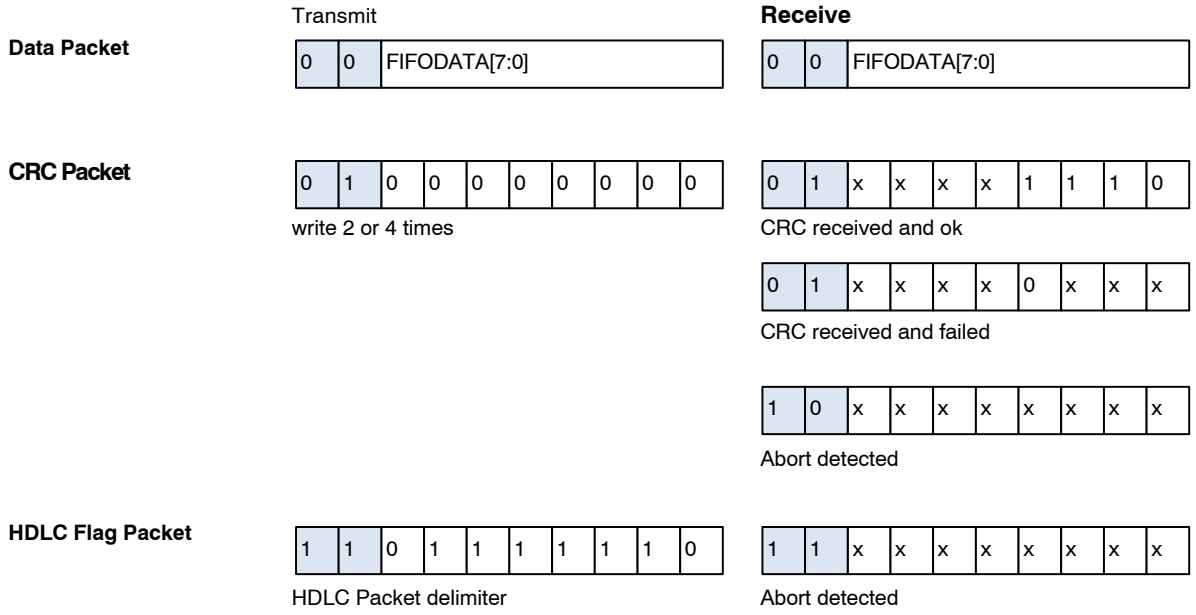

Figure 18.

Figure 19 shows the HDLC transmit process, while Figure 20 shows the HDLC receive process.

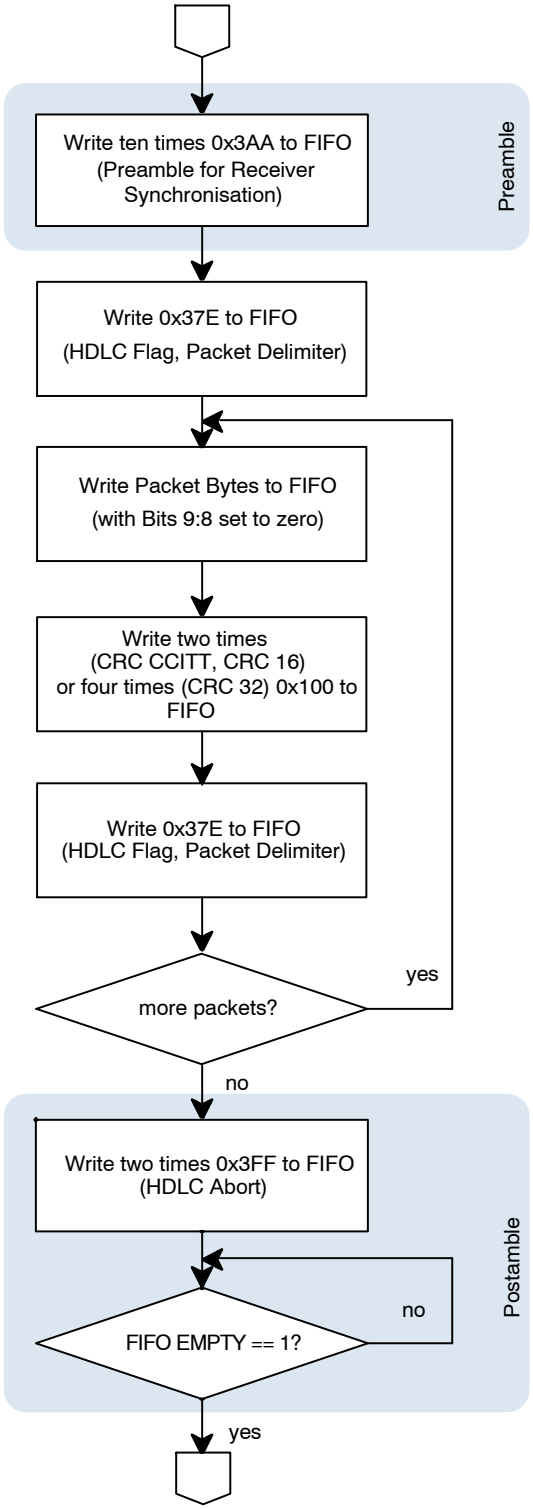

Figure 19. HDLC Transmit Flow Chart

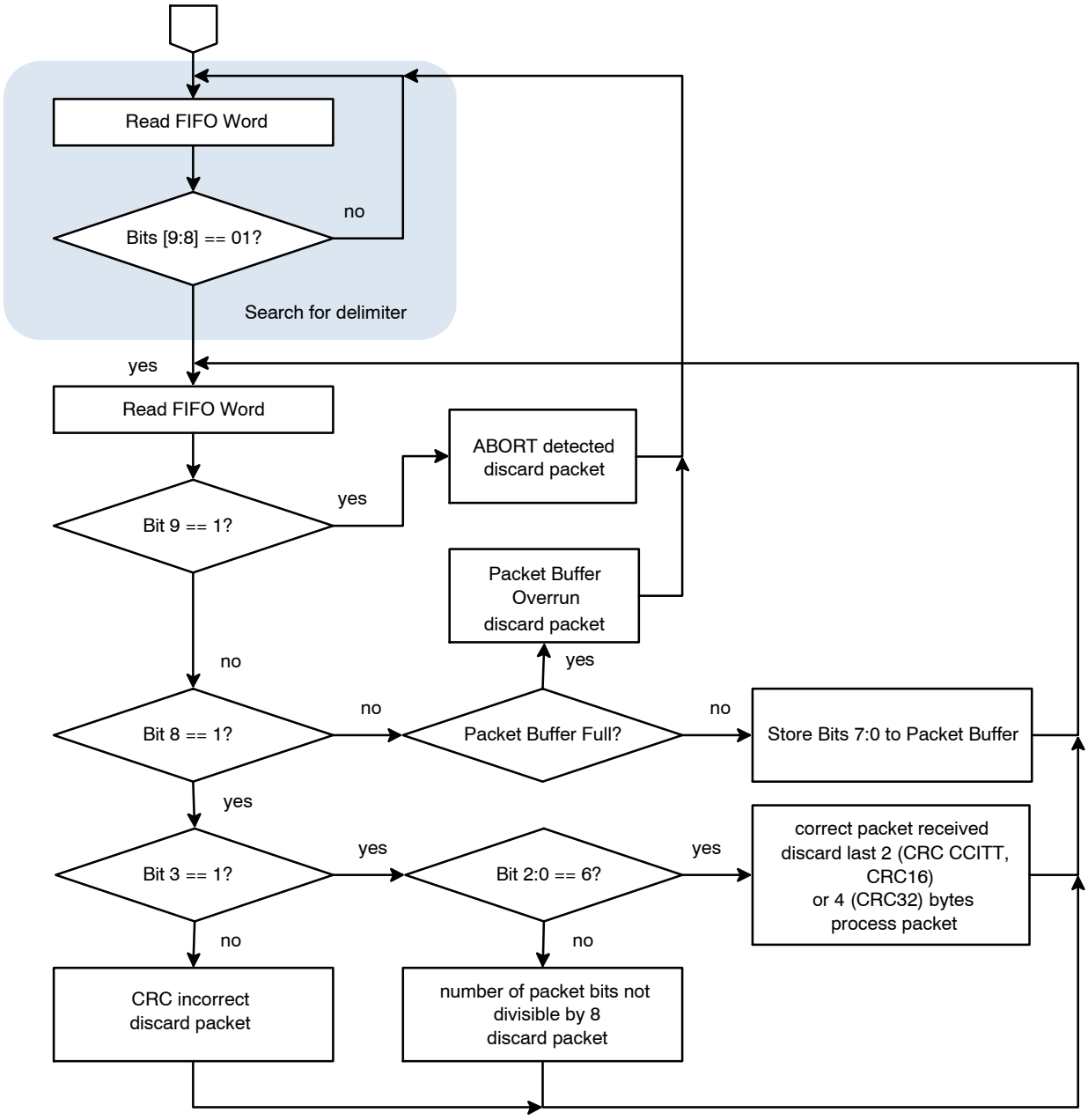

Figure 20. HDLC Receive Flow Chart

Raw Mode

In Raw Mode, no framing is performed. Received bits are grouped into 8 bit bytes and stored in the FIFO. Transmit bits are retrieved from the FIFO as 8 bit bytes and then serialized. The bits are received and transmitted LSB first, that means that bit 0 was received first or will be transmitted first. No byte synchronisation is performed.

Raw Mode is useful to implement legacy protocols in software on the micro−controller.

Raw Soft−decision Mode

In Raw Soft−Decision Mode, no framing is performed. During receive, for each received bit, a 10−bit signed value is written into the FIFO. The sign of the value determines the received bit value, and the magnitude indicates the likelihood of the value being correct.

This mode can be used to improve the performance of error correcting codes implemented in software on the micro−controller.

Transmission works exactly the same as in Raw Mode.

802.15.4

Receiver and transmitter operation differs slightly in 802.15.4 mode versus HDLC mode, due to IEEE 802.15.4 not having a PHY CRC, and 802.15.4 determining packet length from the first byte transmitted. See [3] for a description of the 802.15.4 PHY.

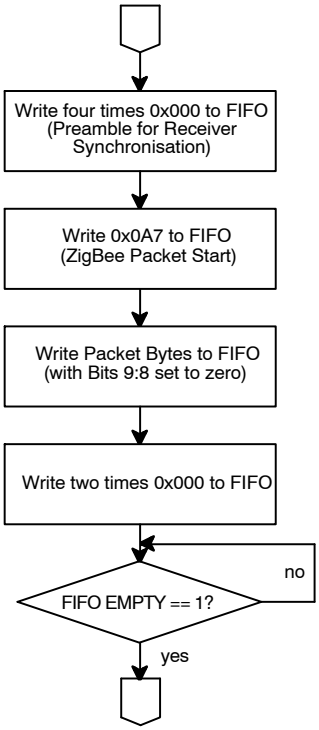

Figure 21. 802.15.4 Transmit Flow Chart

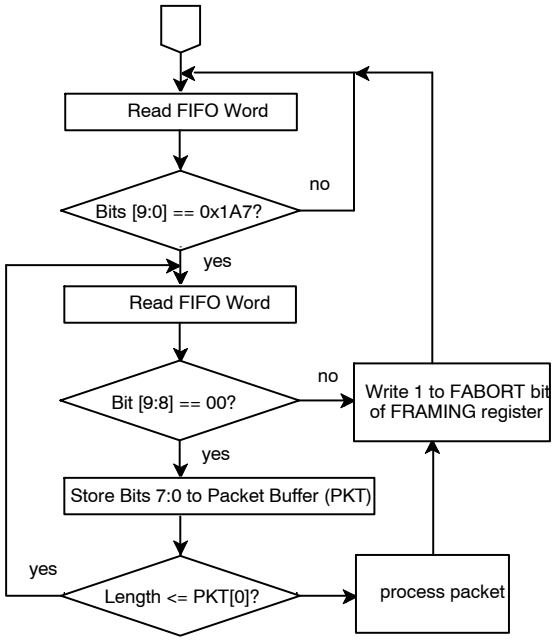

Figure 22. 802.15.4 Receive Flow Chart

Figure 21 details the 802.15.4 transmit operation, while Figure 22 details the 802.15.4 receive operation.

Interrupts

The AX5042 supports interrupts for all non−immediate actions. Interrupts, while not strictly necessary, allow the micro−controller to perform other tasks instead of waiting for the AX5042.

The AX5042 supports level triggered interrupts.

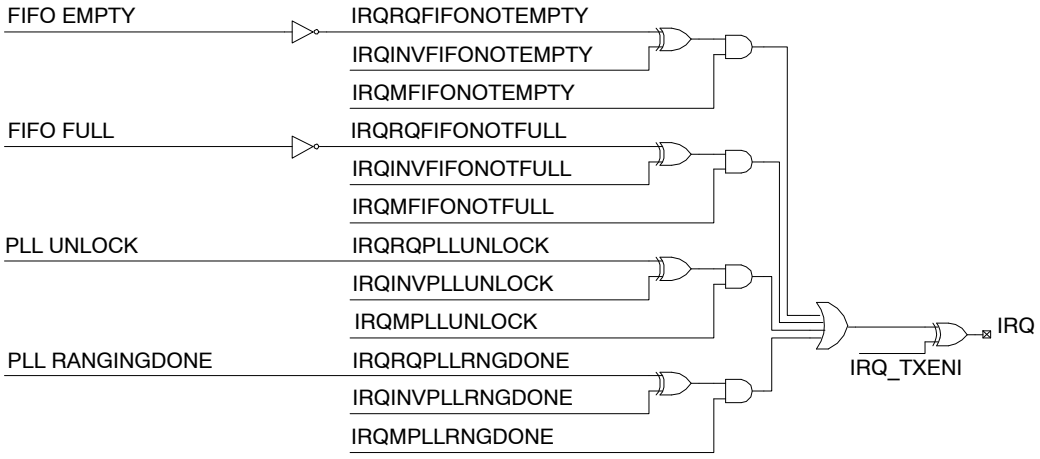

Figure 23. Interrupt Logic Diagram

Figure 23 shows the interrupt logic. The AX5042 supports 4 interrupt sources. Each source may be individually inverted and masked. The final interrupt pin may also be inverted, to support both level active high and level active low interrupts. Inverting and masking is configured using registers IRQMASK, IRQREQUEST, IRQINVERSION. The bit IRQ_TXENI that is used to invert the final interrupt can be found in register PINCFG2.

Table 15 lists all interrupt sources, and how they can be cleared.

Table 15. INTERRUPT SOURCES

Source	When Active	How to Clear
FIFO Not Full	The FIFO contains less than 3 words. At least one word can be written without causing an overrun.	Write words into the FIFO until it is full. Be careful not to cause overruns.
FIFO Not Empty	The FIFO contains at least one word. At least one word can be read without causing an underrun.	Read words from the FIFO until it is empty. Be careful not to cause underruns.
PLL Unlock	The synthesizer has lost lock	This interrupt can be cleared by reading the PLLRANGING register. After switching the synthesizer on, and after frequency changes (including receive ↔ transmit switches), the synthesizer requires some time to settle on the correct frequency and to achieve phase lock with the reference crystal. After that, it should remain locked. The synthesizer losing lock after that point indicates a severe problem. Check the following: • Synthesizer programming (esp. frequency, loop filter settings, charge pump settings, VCO settings) are correct • Synthesizer VCO has been auto−ranged properly • VDD is within spec and not too noisy • Temperature is within spec • Synthesizer is enabled
PLL Ranging Done	The synthesizer has finished auto−ranging its VCO	PLL Ranging Done can be cleared only by restarting a new auto−ranging process. If no more ranging processes are needed, mask the interrupt.

Edge triggered interrupts are not directly supported. In the unlikely event that the chosen micro−controller does not support level triggered interrupts and only supports edge triggered interrupts, they need to be emulated in software. The following C pseudo code illustrates how this can be done:

```
void interrupt_handler(void)
{
     acknowledge_interrupt();
     do {
     handle_interrupt();
     } while (IRQ);
}
```

The first line, acknowledge_interrupt(), acknowledges the interrupt in the interrupt controller of the micro−controller. How this is done is specific to the micro−controller in question, and may even be implicit. The following loop handles interrupts as long as the IRQ line is still active. It is important that the interrupt handler is not terminated before IRQ goes inactive, because otherwise no new edges will be produced by the AX5042, and the interrupt becomes stuck.

Interrupt Strategies

The AX5042 supports two interrupt strategies:

1. The default strategy is to assert IRQ_TXEN as soon as there is one word in the FIFO (receive, using the FIFONOTEMPTY interrupt) or there is one word empty space in the FIFO (transmit, using the FIFONOTFULL interrupt). The micro−controller is required to service the interrupt within 24 bit times (24/BITRATE) to prevent a FIFO overrun or underrun. The micro−controller will receive one interrupt per received FIFO word (message byte). This strategy is recommended for micro−controllers with low interrupt overhead (which is true for most micro−controllers).
2. The second strategy is to assert IRQ_TXEN only when absolutely necessary, i.e. when the FIFO is full (receive, using the inverted FIFONOTEMPTY interrupt) or when the FIFO is empty (transmit, using the inverted FIFONOTEMPTY interrupt). The micro−controller will receive one interrupt every three FIFO words (message bytes). This strategy is useful for micro−controllers with a very high interrupt overhead. Care must be taken to avoid FIFO overruns and underruns.

Preamble

At the beginning of a data transfer, a preamble must be transmitted, before the actual data can be transmitted. The preamble has sveral purposes:

- The preamble allows the power amplifier to ramp up to operational power levels. This is not an issue with the built−in amplifier of the AX5042, which is nearly instantaneous, but may be an issue if external amplifiers are used.
- The preamble allows the various parts of the receiver to achieve lock
- The preamble allows the encoder (transmitter) and the decoder (receiver) to initialise

The AX5042 /AX5051 Preamble Calculator [5] summarizes the rest of this chapter and allows to calculate recommended preamble lengths.

Choosing the Preamble Bit Pattern

In 802.15.4, the preamble bit pattern is specified by the standards committee. This specification, which is four bytes of 0x00, should be followed.

In all other modes, the preamble bit pattern as it enters the modulator should be chosen such that:

- It is DC−free, to ensure that frequency offset estimation works correctly
- It contains as many transitions as possible

Now the transmitter cannot directly control the modulator bits, only the bits that enter the encoder. Thus, the bytes transmitted during the preamble should be chosen according to the selected encoder mode:

Table 16. RECOMMENDED PREAMBLE VALUES

Encoder Settings	Preamble Byte
INV=X, DIFF=0, SCRAM=0, MANCH=0	0x55 or 0xAA
INV=0, DIFF=1, SCRAM=0, MANCH=0	0xFF
INV=1, DIFF=1, SCRAM=0, MANCH=0	0x00
INV=X, DIFF=X, SCRAM=1, MANCH=X	0x55 or 0xAA.
INV=X, DIFF=0, SCRAM=0, MANCH=1	0x00 or 0xFF
INV=0, DIFF=1, SCRAM=0, MANCH=1	0x00
INV=1, DIFF=1, SCRAM=0, MANCH=1	0xFF

Choosing the Preamble Duration

A recommended choice for the preamble duration for FSK is 32 bytes (TMGCORRFRAC = 32) for full frequency offset compensation capabilities. The receiver can work with preambles as short as 3 byte, TMGCORRFRAC must be set to 8 accordingly. With a 3 byte preamble it may not be possible to reach optimal sensitivities and to correct for the full frequency offset range. All sensitivities quoted in the AX5042 Data Sheet refer to TMGCORRFRAC = 32 and a preamble length long enough to correct for the full frequency offset range.

The following section gives some details for a more complete understanding of the factors affecting the preamble duration choice:

The preamble duration should be chosen to be the sum of the following components, rounded up the next higher integral number of bytes (numbers below are given in bits)

- Power amplifier startup time. Zero for the built−in amplifier of the AX5042. Consult documentation in case an external amplifier is used.
- The decoder needs 18−19 bits to synchronize the descrambler, if the descrambler is used, otherwise 1−2 bits
- The time the receiver needs to achieve bit lock is a probabilistic process, and depends on the bit recovery speed settings, the frequency of transitions in the transmit signal, as well as on the signal−to−noise ratio of the received signal. A reasonable estimate would be 3·TMGCORRFRAC if the preamble values detailed above are used and the scrambler is disabled, or 4·TMGCORRFRAC if the scrambler is used.
- The time the receiver needs to achieve frequency offset and phase lock is again a probabilistic process that depends on the initial frequency offset, the signal−to−noise ratio of the received signal, the modulation, and the bandwidth (speed) setting of the frequency recovery loop.
  - ASK: For ASK, achieving frequency lock is not required for demodulation, so no additional preamble for achieving frequency lock needs to be used. TRKFREQ is valid after approximately 600−800 bits.
  - FSK: FSK frequency lock is achieved within 160 bits with FREQGAIN= 3 for the full supported frequency offset range (±½ ·BITRATE). Frequency lock time is approximately proportional to the frequency offset, so if the frequency offset can be guaranteed to be lower than the maximum supported range, correspondingly shorter preambles can be used. Setting FREQGAIN=2 halves the number of bits required for frequency lock at the expense of a slightly worse BER performance (< 1 dB). For small h (h ≤ 1), FREQGAIN=1 or FREQGAIN=0 can be used to further shorten the required number of preamble bits, at the expense of a larger BER performance penalty.
  - PSK: PSK frequency lock is required for demodulation, and is achieved within 140 bits over the full supported offset range (±¼ ·BITRATE). Guaranteeing lower frequency offset does not shorten the required preamble. Additional time may be needed to ensure lock at the correct offset, see the next section for more information.

PSK Frequency Lock

PSK transmits information bits by using a carrier phase angle of 0 or π. The transmit waveform is therefore periodic. The frequency tracking circuit can therefore lock at either the correct offset, or the correct offset ±½ ·DATARATE. In the latter case, every second bit at the input of the decoder will be inverted, because the receiver applies an additional π rotation per received bit. Differential encoding is usually used together with PSK, so after differential decoding, the bitstream will look inverted if the frequency acquisition circuitry is locked to the correct offset ±½ ·DATARATE.

In order to prevent false lock of the frequency acquistion, the microcontroller should periodically check whether the current frequency offset is outside the range −¼ DATARATE…+¼ DATARATE, and restart frequency acquisition if this is the case.

The following C code fragment performs this task and should be called periodically:

```
if (abs((int8_t)spi_read(TRKFREQHI)) >= 0x40) {
      spi_write(TRKFREQHI, 0xC0);
}
```

Furthermore, the preamble duration should be prolonged by the periodicity of executing this code fragment, to ensure that the receiver is fully synchronized before packet data is transmitted.

For example, if 100 kbit/s PSK is used and this fragment is executed once per millisecond (ms), the preamble should be prolonged by 1ms or 100 bits.

Postamble

After the data is transmitted, the micro−controller must write two additional postamble bytes to the FIFO. These bytes are used to clear the transmit pipeline. Their contents do not matter; HDLC flags can be used in HDLC mode.

After these preamble bytes are written to the FIFO, the micro−controller must wait until the FIFO is fully drained (empty). Only then can the transmitter be turned off.

RSSI

A first order approximation of the received signal strength (RSSI) is

$$RSSI1 = -AGCCOUNTER \cdot 0.625\ dB - C1$$

C1 is a constant that is hardware specific. For the AX5042−DVK it is 38 dBm.

The first order approximation degrades for low S/N and low bit rates.

A more accurate RSSI formula is

$$\begin{aligned} RSSI2 &= -AGCCOUNTER[7:1] \cdot 1.25\ dB + \\ &\quad + 20 \cdot \log10(TRKAMPL/0x8000) + \\ &\quad + \{C2 - 80 \cdot \log10(CICDEC) + 6 \cdot CICSHIFT\} \\ &= -AGCCOUNTER[7:1] \cdot 1.25\ dB + \\ &\quad + 20 \cdot \log10(TRKAMPL) + \{C2 - \\ &\quad - 80 \cdot \log10(CICDEC) + 6 \cdot CICSHIFT - \\ &\quad - 20 \cdot \log10(0x8000)\} \end{aligned}$$

As soon as the device has been set−up according to section 2: Programming the Chip the register CICSHIFT can be read and the term in {} can be pre−computed. C2 is a hardware specific constant, for the AX5042−DVK it is 18 dBm for 433 MHz, 16 dBm for 868 MHz, and 16 dBm for 915 MHz. The formula for RSSI2 does not need to be computed with double precision floating point. It can be approximated precisely using a few integer shifts and adds. The following code is used in the AX5042−DVK firmware:

```
/** \brief RSSI correction table
 *
 * This table contains the values of 80*log10(x) for 0.5<=x<1.0
 */
static const int8_t yf_rssilogtable[32] = {
           -24, -23, -22, -21, -20, -19, -18, -17,
           -16, -15, -15, -14, -13, -12, -11, -11,
           -10,  -9,  -9,  -8,  -7,  -7,  -6,  -5,
            -5,  -4,  -3,  -3,  -2,  -2,  -1,  -1
};
/**
 * \brief Return bandwidth specific AGC reference level
 *
 * Due to the gain of internal filters, the reference level for the
 * AGC routine is bandwidth specific. This routine computes the AGC
 * reference level. It is 16dBm-80*log10(CICDEC)+6*CICSHIFT
 * \returns the bandwidth specific AGC reference level in dBm
 */
static int8_t yellowfoot_get_agcref(void)
{
           // 240: 80*log10(1024)
           // 16: board specific reference level
           int8_t r = 16-240;
           uint16_t t = spi_read16(AX5042_REG_CICDECHI);
           if (!t)
                return -128;
           while (t < 512) {
                t <<= 1;
                r += 24;
           }
           t >>= 4;
           r -= yf_rssilogtable[((uint8_t)t) - 32];
           r += 6 * (spi_read(AX5042_REG_CICSHIFT) & 0x1F);
           return r;
}
/**
 * \brief Return current AGC value
 *
 * \returns the current AGC value in dBm
 */
int8_t yellowfoot_get_agc(void)
{
           int16_t agc = spi_read(AX5042_REG_AGCCOUNTER) & 0xfe;
           uint16_t trkampl = spi_read16(AX5042_REG_TRKAMPLITUDEHI);
           if (!trkampl)
                return -128;
           agc <<= 1;
           agc += (agc >> 2);
           agc = -agc;
           while (trkampl < 0x4000) {
                trkampl <<= 1;
                agc -= 24;
           }
           trkampl >>= 9;
           agc += yf_rssilogtable[((uint8_t)trkampl) - 32];
           return (int8_t)(agc >> 2) + yellowfoot_get_agcref();
}
```

REGISTER BANK DESCRIPTION

This section describes the bits of the register bank in detail. The registers are grouped by functional block to facilitate programming.

No checks are made whether the programmed combination of bits makes sense! Bit 0 is always the LSB.

NOTES: Whole registers or register bits marked as reserved should be kept at their default values. All addresses not documented here must not be accessed, neither in reading nor in writing.

Table 17. CONTROL REGISTER MAP

Addr	Name	Dir	Reset	Bit 7	6	5	4	3	2	1	0	Description
Revision & Interface Probing												
0	REVISION	R	00000010	SILICONREV(7:0)								Silicon Revision
1	SCRATCH	RW	11000101	SCRATCH(7:0)								Scratch Register
Operating Mode												
2	PWRMODE	RW	011–0000	RST	REFEN	XOEN	–	PWRMODE(3:0)				Power Mode
3	XTALOSC	RW	––––0010	–	–	–	–	XTALOSCGM(3:0)				GM of Crystal Oscillator
FIFO												
4	FIFOCTRL	RW	––––––11	FIFOSTAT(1:0)		FIFO OVER	FIFO UNDER	FIFO FULL	FIFO EMPTY	FIFOCMD(1:0)		FIFO Control
5	FIFODATA	RW	––––––––	FIFODATA(7:0)								FIFO Data
Interrupt Control												
6	IRQMASK	RW	––––0000	–	–	–	–	IRQMASK(3:0)				IRQ Mask
7	IRQREQUEST	R	––––––––	–	–	–	–	IRQREQUEST(3:0)				IRQ Request
Interface & Pin Control												
8	IFMODE	RW	––––0011	–	–	–	–	IFMODE(3:0)				Interface Mode
0C	PINCFG1	RW	11111000	DATAZ	DCLKZ	IRQ_TXENZ	PWRUPZ	SYSCLK(3:0)				Pin Configuration 1
0D	PINCFG2	RW	00000000	DATAE	DCLKE	PWRUP_IRQ_TXENE		DATAI	DCLKI	IRQPTTI	PWRUPI	Pin Configuration 2
0E	PINCFG3	R	––––––––	–	–	–	SYSCLKR	DATAR	DCLKR	IRQPTTR	PWRUPR	Pin Configuration 3
0F	IRQINVERSION	RW	––––0000	–	–	–	–	IRQINVERSION(3:0)				IRQ Inversion
Modulation & Framing												
10	MODULATION	RW	––––0010	–	–	–	–	MODULATION(3:0)				Modulation
11	ENCODING	RW	––––0010	–	–	–	–	ENC MANCH	ENC SCRAM	ENC DIFF	ENC INV	Encoder/Decoder Settings
12	FRAMING	RW	–0000000	–	HSUPP	CRCMODE(1:0)		FRMMODE(2:0)			FABORT	Framing settings
14	CRCINIT3	RW	11111111	CRCINIT(31:24)								CRC Initialisation Data
15	CRCINIT2	RW	11111111	CRCINIT(23:16)								CRC Initialisation Data
16	CRCINIT1	RW	11111111	CRCINIT(15:8)								CRC Initialisation Data
17	CRCINIT0	RW	11111111	CRCINIT(7:0)								CRC Initialisation Data
Synthesizer												
20	FREQ3	RW	00111001	FREQ(31:24)								Synthesizer Frequency
21	FREQ2	RW	00110100	FREQ(23:16)								Synthesizer Frequency
22	FREQ1	RW	11001100	FREQ(15:8)								Synthesizer Frequency

Table 17. CONTROL REGISTER MAP

Addr	Name	Dir	Reset	Bit 7	6	5	4	3	2	1	0	Description
23	FREQ0	RW	11001101	FREQ(7:0)								Synthesizer Frequency
25	FSKDEV2	RW	00000010	FSKDEV(23:16)								FSK Frequency Deviation
26	FSKDEV1	RW	01100110	FSKDEV(15:8)								FSK Frequency Deviation
27	FSKDEV0	RW	01100110	FSKDEV(7:0)								FSK Frequency Deviation
28	IFFREQHI	RW	00100000	IFFREQ(15:8)								2nd LO / IF Frequency
29	IFFREQLO	RW	00000000	IFFREQ(7:0)								2nd LO / IF Frequency
2C	PLLLOOP	RW	–0011101	–	Reserved	BANDSEL	PLLCPI(2:0)			FLT(1:0)		Synthesizer Loop Filter Settings
2D	PLLRANGING	RW	00001000	STICKY LOCK	PLL LOCK	RNGERR	RNG START	VCOR(3:0)				Synthesizer VCO Auto–Ranging
Transmitter												
30	TXPWR	RW	––––1000	–	–	–	–	TXRNG(3:0)				Transmit Power
31	TXRATEHI	RW	00001001	TXRATE(23:16)								Transmitter Bit Rate
32	TXRATEMID	RW	10011001	TXRATE(15:8)								Transmitter Bit Rate
33	TXRATELO	RW	10011010	TXRATE(7:0)								Transmitter Bit Rate
34	MODMISC	RW	––––––11	–	–	–	–	–	–	reserved	PTTCLK GATE	Misc RF Flags
Receiver												
39	AGCTARGET	RW	––01010	–	–	–	AGCTARGET(4:0)					AGC Target **Must be set to 0x0E**
3A	AGCATTACK	RW	00010110	reserved			AGCATTACK(4:0)					AGC Attack
3B	AGCDECAY	RW	0–010011	reserved	–	reserved	AGCDECAY(4:0)					AGC Decay
3C	AGCCOUNTER	R	––––––––	AGCCOUNTER(7:0)								AGC Current Value
3D	CICSHIFT	R	––000100	–	–	reserved	CICSHIFT(4:0)					CIC Shift Factor
3E	CICDECHI	RW	––––––00	–	–	–	–	–	–	CICDEC(9:8)		CIC Decimation Factor
3F	CICDECLO	RW	00000100	CICDEC(7:0)								CIC Decimation Factor
40	DATARATEHI	RW	00011010	DATARATE(15:8)								Data rate
41	DATARATELO	RW	10101011	DATARATE(7:0)								Data rate
42	TMGGAINHI	RW	00000000	TIMINGGAIN(15:8)								Timing Gain
43	TMGGAINLO	RW	11010101	TIMINGGAIN(7:0)								Timing Gain
44	PHASEGAIN	RW	00––0011	reserved		–	–	PHASEGAIN(3:0)				Phase Gain
45	FREQGAIN	RW	––––1010	–	–	–	–	FREQGAIN(3:0)				Frequency Gain
46	FREQGAIN2	RW	––––1010	–	–	–	–	FREQGAIN2(3:0)				Frequency Gain 2
47	AMPLGAIN	RW	––00110	–	–	–	reserved	AMPLGAIN(3:0)				Amplitude Gain
48	TRKAMPLHI	R	––––––––	TRKAMPL(15:8)								Amplitude Tracking
49	TRKAMPLLO	R	––––––––	TRKAMPL(7:0)								Amplitude Tracking
4A	TRKPHASEHI	R	––––––––	–	–	–	–	TRKPHASE(11:8)				Phase Tracking
4B	TRKPHASELO	R	––––––––	TRKPHASE(7:0)								Phase Tracking

Table 17. CONTROL REGISTER MAP

Addr	Name	Dir	Reset	Bit 7	6	5	4	3	2	1	0	Description
4C	TRKFREQHI	R	————	TRKFREQ(15:8)								Frequency Tracking
4D	TRKFREQLO	R	————	TRKFREQ(7:0)								Frequency Tracking
Misc												
70	APEOVER	R	00000000	APEOVER	OSCAPE	REFAPE	reserved					APE Overrride
72	PLLVCOI	RW	−−000100	–	–	reserved			VCO_I(2:0)			Synthesizer VCO current Leave at default
74	PLLRNG	RW	00−−−000	reserved		–	–	–	reserved		PLLARNG	Auto−ranging internal settings **PLLARNG must be set to 1**
7C	REF	RW	−−100011	–	–	reserved			REF_I(2:0)			Reference adjust Leave at default
7D	RXMISC	RW	−−110110	–	–	reserved				RXIMIX(1:0)		Misc RF settings **RXIMIX(1:0) must be set to 01**

Register Descriptions

REVISION

The register holds the revision index of the chip.

Table 18. REVISION

Name	Bits	R/W	Reset	Description
REVISION	7:0	R	00000010	Silicon Revision

SCRATCH

The SCRATCH register does not affect the function of the chip in any way. It is intended for the micro−controller to test communication to the AX5042.

Table 19. SCRATCH

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

PWRMODE

This register controls the powering and reset of the various blocks of the chip.

Table 20. PWRMODE

Name	Bits	R/W	Reset	Description
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.
REFEN	6	RW	1	Reference Enable, for usage see Table 3: PWRMODE and APEOVER Register States and Table 4: PWRUP and IRQ_TXEN Pin States
XOEN	5	RW	1	Crystal Oscillator Enable, for usage see Table 3: PWRMODE and APEOVER Register States and Table 4: PWRUP and IRQ_TXEN Pin States
PWRMODE	3:0	RW	0000	Powermode; see Table 3: PWRMODE and APEOVER Register States and Table 4: PWRUP and IRQ_TXEN Pin States

NOTES: The REFEN/XOEN bits have no effect unless PWRMODE is set to 0001.

Use the register APEOVER to power the chip down completely.

Before RST can be written to 1, the SPI interface of the chip needs to be reset. This is done by setting the SEL line to high.

XTALOSC

This register controls the transconductance of the crystal oscillator. Optimal settings will depend on the characteristics of the specific crystal that is used. For a table containing the values as a function of the register settings see the AX5042 Datasheet.

Table 21. XTALOSC

Name	Bits	R/W	Reset	Description
XTALOSCGM	3:0	RW	0010	Transconductance of the Crystal Oscillator

FIFOCTRL

This register is used to send control commands that depend on the selected frame mode and holds the FIFO status information. For further information on FIFO settings see section: “Receive and Transmit” and the register FIFODATA.

Table 22. FIFOCTRL

Name	Bits	R/W	Reset	Description
FIFOCMD	1:0	RW	11	FIFO command bits (written to FIFO during next write to FIFO-DATA); see section “Receive and Transmit” for information on the exact operation of these bis
FIFO EMPTY	2	R	–	FIFO is empty if 1
FIFO FULL	3	R	–	FIFO is full if 1; if 1 the FIFO contains 3 words.
FIFO UNDER	4	R	–	FIFO under run occurred since last read of FIFOCTRL when 1. This bit is set when a read operation by the transmitter (transmit mode) or the micro−controller (receive mode) was attempted while the FIFO was empty.
FIFO OVER	5	R	–	FIFO over run occurred since last read of FIFOCTRL when 1. This bit is set when a write operation by the receiver (receive mode) or the micro−controller (transmit mode) was attempted while the FIFO was full.
FIFOSTAT	7:6	R	–	FIFO Status bits associated with current FIFO top word; see section “Receive and Transmit” for exact operation of these bis

NOTE: The FIFO OVER bit may also accidentally be set when the FIFO is completely empty.

FIFODATA

This register is used to read from and write to the 3 level x 10 bit FIFO. For further information on FIFO settings see section: “Receive and Transmit” and the register FIFOCTRL.

Table 23. FIFODATA

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

IRQMASK

This register allows to mask or de−mask interrupts. For further information on interrupt related settings see section: “Interrupts” and the registers IRQREQUEST and IRQINVERSION as well as PINCFG1 and PINCFG2.

Table 24. IRQMASK

Name	Bits	R/W	Reset	Description
IRQMFIFONOTEMPTY	0	RW	0	FIFO not empty interrupt enable
IRQMFIFONOTFULL	1	RW	0	FIFO not full interrupt enable
IRQMPLLUNLOCK	2	RW	0	Synthesizer lock lost interrupt enable
IRQMPLLRNGDONE	3	RW	0	Synthesizer auto−ranging done interrupt enable

IRQREQUEST

This register indicates pending interrupts. For further information on interrupt related settings see section: “Interrupts” and the registers IRQMASK and IRQINVERSION as well as PINCFG1 and PINCFG2.

Table 25. IRQREQUEST

Name	Bits	R/W	Reset	Description
IRQRQFIFONOTEMPTY	0	R	–	FIFO not empty interrupt pending
IRQRQFIFONOTFULL	1	R	–	FIFO not full interrupt pending
IRQRQPLLUNLOCK	2	R	–	Synthesizer lock lost interrupt pending
IRQRQPLLRNGDONE	3	R	–	Synthesizer auto−ranging done interrupt pending

IFMODE

This register is used to configure the interface mode of the AX5042.

Table 26. IFMODE

Name	Bits	R/W	Reset	Description
IFMODE	3:0	RW	0011	See Table 27.

Table 27. INTERFACE MODE BIT VALUES

IFMODE Bits	Meaning
0000	Frame Mode
0010	Synchronous Wire Mode
0011	Asynchronous Wire Mode (RS232)

PINCFG1

This register allows to configure some of the AX5042 pins if they have been set−up to function as General Purpose I/O (GPIO) pins in register PINCFG2.

Table 28. PINCFG1

Name	Bits	R/W	Reset	Description
SYSCLK	3:0	RW	1000	See Table 29.
PWRUPZ	4	RW	1	1: configure PWRUP pin as input (tri−state) 0: configure PWRUP pin as output This bit is only active if PWRUPE = 1
IRQ_TXENZ	5	RW	1	1: configure IRQ_TXEN pin as input (tri−state) 0: configure IRQ_TXEN pin as output This bit is only active if IRQ_TXENE = 1
DCLKZ	6	RW	1	1: configure DCLK pin as input (tri−state) 0: configure DCLK pin as output This bit is only active if DCLKE = 1
DATAZ	7	RW	1	1: configure DATA pin as input (tri−state) 0: configure DATA pin as output This bit is only active if DATAE = 1

Table 29. SYSCLK BIT VALUES

SYSCLK Bits	Meaning
0000	SYSCLK pin as Output ‘0’
0001	SYSCLK pin as Output ‘1’
0010	SYSCLK pin as input (tri−state)
0011	SYSCLK Output inverted f_{XTAL}
0100	SYSCLK Output f_{XTAL}
0101	SYSCLK Output $f_{XTAL}/2$
0110	SYSCLK Output $f_{XTAL}/4$
0111	SYSCLK Output $f_{XTAL}/8$
1000	SYSCLK Output $f_{XTAL}/16$
1001	SYSCLK Output $f_{XTAL}/32$
1010	SYSCLK Output $f_{XTAL}/64$
1011	SYSCLK Output $f_{XTAL}/128$
1100	SYSCLK Output $f_{XTAL}/256$
1101	SYSCLK Output $f_{XTAL}/512$
1110	SYSCLK Output $f_{XTAL}/1024$
1111	SYSCLK Output $f_{XTAL}/2048$

PINCFG2

This register allows to configure some of the AX5042 pins to function as General Purpose I/O (GPIO) pins rather than having their special default function.

Bits PWRUP_IRQ_TXENE, DCLKE and DATAE are used to enable the special function of the respective pin or set it to GPIO.

Bits PWRUPI, IRQ_TXENI, DCLKI and DATAI are used to set the state of the pins, if defined as GPIO and configured as output in PINCFG1. If the pins are configured as special function pins, these bits are used to chose if the output signal should be inverted.

Table 30. PINCFG2

Name	Bits	R/W	Reset	Description	
				GPIO pin	Special pin
PWRUPI	0	RW	0	0: set PWRUP pin to ‘1’ 1: set PWRUP pin to ‘0’	0: no output inversion 1: invert output
IRQ_TXENI	1	RW	0	0: set IRQ_TXEN pin to ‘1’ 1: set IRQ_TXEN pin to ‘0’	0: no output inversion 1: invert output
DCLKI	2	RW	0	0: set DCLK pin to ‘1’ 1: set DCLK pin to ‘0’	0: no output inversion 1: invert output
DATAI	3	RW	0	0: set DATA pin to ‘1’ 1: set DATA pin to ‘0’	0: no output inversion 1: invert output
PWRUP_IRQ_TXENE	5:4	RW	00	00: Enable special function: PWRUP, IRQ_TXEN 11: PWRUP and IRQ_TXEN pins are GPIO pins 01, 10: Invalid values, do not use	
DCLKE	6	RW	0	0: Enable special function: DCLK 1: DCLK pin is a GPIO pin	
DATAE	7	RW	0	0: Enable special function: DATA 1: DATA pin is a GPIO pin	

PINCFG3

GPIO state register: This register holds the signals on the GPIO pins. It can be used to read signals, if PINCFG1 configures the respective pin as input.

Table 31. PINCFG3

Name	Bits	R/W	Reset	Description
PWRUPR	0	R	–	Logic State of PWRUP Pin
IRQ_TXENR	1	R	–	Logic State of IRQ_TXEN Pin
DCLKR	2	R	–	Logic State of DCLK Pin
DATAR	3	R	–	Logic State of DATA Pin
SYSCLKR	4	R	–	Logic State of SYSCLK Pin

IRQINVERSION

This register allows to invert the logic levels of the level−triggered interrupts.

Table 32. IRQINVERSION

Name	Bits	R/W	Reset	Description
IRQINVFIFONOTEMPTY	0	RW	0	FIFO not empty interrupt inversion
IRQINVFIFONOTFULL	1	RW	0	FIFO not full interrupt inversion
IRQINVPLLUNLOCK	2	RW	0	Synthesizer lock lost interrupt inversion
IRQINVPLLRNGDONE	3	RW	0	Synthesizer auto−ranging done interrupt inversion

MODULATION

This register is used to configure the modulation type to be used in data transfers. For a programming guide and coding information see section: “Programming the Chip” and Table 11: Modulation Register Programming.

Table 33. MODULATION

Name	Bits	R/W	Reset	Description
MODULATION	3:0	RW	0010	See Table 11: Modulation Register Porgramming

ENCODING

The register configures the encoder.

Table 34. 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

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.

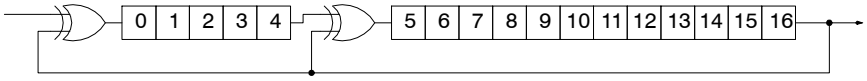

Figure 24. Scrambler Operation

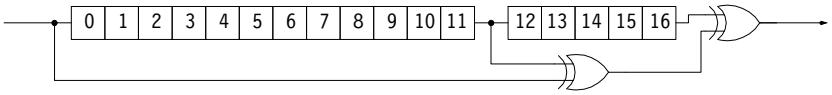

Figure 25. Descrambler Operation

Figure 24 shows a schematic circuit diagram for the scrambler, and Figure 25 for the descrambler. The numbered boxes represent a delay by one bit.

Figure 26 shows a few well known encoding formats used in telecom and Table 35 describes them.

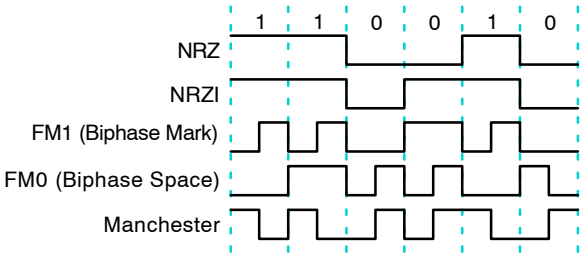

Figure 26. Customary Telecom Encoding Modes

Table 35. CUSTOMARY TELECOM MODES DESCRIPTION

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. The HDLC bit stuffing ensures that there are periodic zeros and thus transitions, and the encoding is inversion invariant, and therefore useable for PSK.
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.

Table 35. CUSTOMARY TELECOM MODES DESCRIPTION

Name	Bits	Description
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 bit rate
- In HDLC mode, use NRZI, NRZI+Scrambler, or NRZ+Scrambler. If HDLC is to be transmitted over PSK, NRZI and NRZI+Scrambler are valid choices.
- In 802.15.4, use NRZ mode.
- In Raw modes, the choice depends on the legacy system to be implemented.

FRAMING

The register sets the framing mode and the CRC type.

Table 36. FRAMING

Name	Bits	R/W	Reset	Description
FABORT	0	S	0	Write 1 to abort current HDLC packet
FRMMODE	3:1	RW	000	Defines framing type. See Table 37.
CRCMODE	5:4	RW	00	Defines the CRC type. See Table 38.
HSUPP	6	RW	0	Suppress unneeded abort / flag / data indications.

Table 37. FRAME MODE BIT VALUES

FRMMODE Bits	Meaning
000	Raw
001	Raw, Soft−Decision
010	HDLC
110	802.15.4
111	Reserved for future use

Table 38. CRC MODE BIT VALUES

CRCMODE Bits	Meaning
00	CCITT (16 bit)
01	CRC−16
10	CRC−32
11	Invalid

CRCINIT3, CRCINIT2, CRCINIT1, CRCINIT0

This register can be used to set the reset value of the CRC calculation. Normally this register is left at all ones.

Table 39. CRCINIT3, CRCINIT2, CRCINIT1, CRCINIT0

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

FREQ3, FREQ2, FREQ1, FREQ0

This registers are used to set the carrier frequency.

Table 40. FREQ3, FREQ2, FREQ1, FREQ0

Name	Bits	R/W	Reset	Description
FREQ	31:0	RW	0x3934CCCD	Frequency; FREQ = [$f_{CARRIER}/f_{XTAL} \cdot 2^{24} + ½$]

Note that to program frequencies in the 433 MHz band registers FREQ3, FREQ2, FREQ1, FREQ0 must be programmed to appropriate values and the bit BANDSEL in the PLLLOOP register must be set to 1.

In 868/915 MHz band mode, swap bits 23 and 24 of the frequency register.

In the 433 MHz band mode some frequencies are not selectable. Specifically, the following divider ratios are not achievable:

Divider ($f_{CARRIER}/f_{XTAL}$)		Carrier Frequency (f_{XTAL}=16MHz)	
From	To	From (MHz)	To (MHz)
…	…	…	…
24.25	24.50	388	392
25.50	25.75	408	412
26.25	26.50	420	424
27.50	27.75	440	444
28.25	28.50	452	456
29.50	29.75	472	476
…	…	…	…

The pattern repeats every 2 divider values or every 32 MHz for a 16 MHz crystal. In this case, the only known work−around is to choose another crystal frequency.

The following C code illustrates the encoding and decoding that needs to be performed on the frequency register values in the 433 MHz band. The function encode433() converts the value computed with the frequency formula above to the value that must be written into the frequency register. abort() marks those frequencies that cannot be programmed. decode433() reverses the computation of encode433().

```
uint32_t decode433(uint32_t f)
{
    uint32_t f1 = f & 0x003fffff;
    f1 |= -((f >> 2) & 0x00400000);
    f = (f & 0xff800000) + ((f << 1) & 0x00800000) + f1;
    return f;
}
uint32_t encode433(uint32_t fr)
{
    uint32_t fr1 = (fr >> 22) & 7;
    if (fr1 == 1 || fr1 == 6) {
        abort();
    }
    fr &= ~0x01c00000;
    if (fr1 & 1) {
        fr1 -= 3;
        fr |= 0x01000000;
    }
    if (fr1 >= 2) {
        fr1 -= 2;
        fr |= 0x00800000;
    }
    if (fr1 >= 2) {
        fr1 -= 2;
        fr |= 0x00400000;
    }
    return fr;
}
```

For the 868/930 MHz band the ratio $f_{CARRIER}/f_{XTAL}$ = = FREQ/2^{24} should not be in the range [62.5 .. 64.5].

For the 433 MHz band the ratio $(2 \cdot f_{CARRIER})/f_{XTAL}$ = = FREQ/2^{23} should not be in the range [62.5 .. 64.5].

If a 16 MHz reference crystal is used, then the above limitations on the division ratio are not limiting for SRD frequencies.

FSKDEV2, FSKDEV1, FSKDEV0

These registers are used to set the FSK frequency deviation.

Table 41. FSKDEV2, FSKDEV1, FSKDEV0

Name	Bits	R/W	Reset	Description
FSKDEV	23:0	RW	0x026666	(G)FSK Frequency Deviation; FSKDEV = [$f_{DEVIATION}/f_{XTAL} \cdot 2^{24}$ + ½]

NOTE: $f_{DEVIATION}$ is actually half the deviation.
The mark frequency is $f_{CARRIER} + f_{DEVIATION}$,
the space frequency is $f_{CARRIER} - f_{DEVIATION}$.
The parameter h is defined by the following equation
$f_{DEVIATION} = h/2 \cdot BITRATE$

IFFREQHI, IFFREQLO

These registers are used to set the IF frequency, for most cases the nominal frequency of 1 MHz is suitable.

Table 42. IFFREQHI, IFFREQLO

Name	Bits	R/W	Reset	Description
IFFREQ	23:0	RW	0x2000	IF Frequency; IFFREQ = [$f_{IF}/f_{XTAL} \cdot 2^{17} + ½$]

PLLLOOP

This register allows to configure the synthesizer loop bandwidth and the frequency band. For recommendations on settings to use see Table 10: Recommended synthesizer loop bandwidth settings.

Table 43. PLLLOOP

Name	Bits	R/W	Reset	Description
FLT	1:0	RW	01	Filter setting. See Table 44.
PLLCPI	4:2	RW	111	Charge pump current multiplier
BANDSEL	6:5	RW	00	Band selection. See Table 45.

Table 44. FILTER BIT VALUES

FLT Bits	Meaning
00	External Loop Filter, do not use
01	Internal Loop Filter, nominal loop filter setting
10	Internal Loop Filter, bandwidth boosted by factor 5, only works if pin LPFILT is left unconnected
11	Internal Loop Filter, bandwidth boosted by factor 2, only works if pin LPFILT is left unconnected

Table 45. BAND SELECTION BIT VALUES

BANDSEL Bit	Meaning
0	868/915 MHz
1	433 MHz

NOTE: Note that to program frequencies in the 433 MHz band registers FREQ3, FREQ2, FREQ1, FREQ0 must be programmed to appropriate values and the bit BANDSEL in the PLLLOOP register must be set to 1.

PLLRANGING

This register is used to initiate and control the auto−ranging of the synthesizer VCO. It also holds the VCO range value that is currently being used. For information on how to use this register consult section: “Synthesizer VCO Auto−Ranging”.

Table 46. PLLRANGING

Name	Bits	R/W	Reset	Description
VCOR	3:0	RW	1000	VCO Range
RNG START	4	RS	0	Synthesizer VCO auto−ranging; Write 1 to start auto−ranging, bit clears when auto−ranging done
RNGERR	5	R	–	Ranging Error; this bit is set when RNG START transitions from 1 to 0 and the programmed frequency cannot be achieved
PLL LOCK	6	R	–	PLL LOCK indicates the state of the synthesizer at the moment of the register access. Synthesizer is locked if 1
STICKY LOCK	7	R	–	STICKY LOCK indicates, the state of synthesizer since last read of the register. if 0, synthesizer lost lock after last read of PLLRANGING register

TXPWR

This register programs the transmit output power level.

Table 47. TXPWR

Name	Bits	R/W	Reset	Description
TXRNG	3:0	RW	1000	Transmit Power, see AX5042 Datasheet for details.

TXRATEHI, TXRATEMID, TXRATELO

These registers set the transmit bit rate.

Table 48. TXRATEHI, TXRATEMID, TXRATELO

Name	Bits	R/W	Reset	Description
TXRATE	23:0	RW	0x09999A	Transmit Bitrate; TXRATE = [BITRATE/$f_{XTAL} \cdot 2^{24}$ + ½]

In asynchronous wire mode, BITRATE < (f_{XTAL}/32)

MODMISC

The behaviour of the transmitter if the synthesizer looses lock is set with this register.

Table 49. MODMISC

Name	Bits	R/W	Reset	Description
PTTLCK GATE	0	RW	1	If set to 1 then the transmitter is automatically disabled if the synthesizer looses lock

AGCTARGET

This register sets the target value which the AGC control loop tries to maintain.

Table 50. AGCTARGET

Name	Bits	R/W	Reset	Description
AGCTARGET	4:0	RW	10110	Must be set to 0x0E

AGCATTACK

This register along with AGCDECAY controls the AGC (automatic gain control) loop slopes, and thus the speed of the gain adjustments. The higher the bit rate, the faster the AGC loop should be set.

Table 51. AGCATTACK

Name	Bits	R/W	Reset	Description
AGCATTACK	4:0	RW	01010	AGC gain reduction speed; 2.5 dB · $f_{XTAL}2^{AGCATTACK-27}$ [dB/s]

1. The recommended AGCATTACK settings can be found in Table 13: AGC Dynamics Register Values.

AGCDECAY

This register along with AGCATTACK controls the AGC (automatic gain control) loop slopes, and thus the speed of the gain adjustments. The higher the bit rate, the faster the AGC loop should be set.

Table 52. AGCDECAY

Name	Bits	R/W	Reset	Description
AGCDECAY	4:0	RW	10011	AGC gain increase speed; 2.5 dB · $f_{XTAL}2^{AGCDECAY-27}$ [dB/s]

1. The recommended AGCDECAY settings can be found in Table 13: AGC Dynamics Register Values.

AGCCOUNTER

This register contains the current setting of the automatic gain control (AGC) and can be used to calculate an RSSI value (RSSI1). See section: “RSSI” for details on RSSI calculation.

Table 53. AGCCOUNTER

Name	Bits	R/W	Reset	Description
AGCCOUNTER	7:0	R	–	Current AGC Gain, in 0.625 dB steps

CICSHIFT

This register must be read to be able to make the calculations for RSSI2, the calculation is described in section: “RSSI”. CICSHIFT is updated when CICDECHI, CICDECLO are written.

Table 54. CICSHIFT

Name	Bits	R/W	Reset	Description
CICSHIFT	4:0	R	00100	CIC Shift factor, used for RSSI2 calculation

CICDECHI, CICDECLO

These registers set the bandwidth of the digital channel filter. For detailed information on programming this register see section: “Parameter Programming”.

Table 55. CICDECHI, CICDECLO

Name	Bits	R/W	Reset	Description
CICDEC	9:0	RW	0x004	CIC Decimation factor. CICDEC = [$(1.5 \cdot f_{XTAL})$ / $(8 \cdot 1.2 \cdot BW)$], if TMGCORRFRAC > 16 or CICDEC = [$(1.5 \cdot f_{XTAL})$ / $(8 \cdot 1.4 \cdot BW)$], if TMGCORRFRAC ≤ 16

DATARATEHI, DATARATELO

These registers specify the receiver data−rate, relative to the channel filter bandwidth. For detailed information on programming this register see section: “Parameter Programming”.

Table 56. DATARATEHI, DATARATELO

Name	Bits	R/W	Reset	Description
DATARATE	15:0	RW	0x1AAB	DATARATE = [$(2^{10} \cdot f_{XTAL})$ / (BITRATE · CICDEC · FSKMUL) + ½]

TMGGAINHI, TMGGAINLO

These registers specify the aggressiveness of the receiver bit timing recovery. More aggressive settings allow the receiver to synchronize with shorter preambles, at the expense of more timing jitter and thus a higher bit error rate at a given signal−to−noise ratio. For detailed information on programming this register see section : “Parameter Programming”.

Table 57. TMGGAINHI, TMGGAINLO

Name	Bits	R/W	Reset	Description
TMGGAIN	15:0	RW	0x00D5	TMGGAIN = [(FSKMUL · DATARATE) / TMGCORRFRAC + ½]

PHASEGAIN

This register controls the bandwidth of the phase tracking loop. For detailed information on programming this register see section: “Parameter Programming”.

Table 58. PHASEGAIN

Name	Bits	R/W	Reset	Description
PHASEGAIN	3:0	RW	0011	Bandwidth of the phase recovery loop

FREQGAIN

This register controls the bandwidth of the frequency tracking loop. For detailed information on programming this register see section: “Parameter Programming”.

Table 59. FREQGAIN

Name	Bits	R/W	Reset	Description
FREQGAIN	3:0	RW	1010	Bandwidth of the frequency recovery loop

FREQGAIN2

This register controls the bandwidth of the frequency tracking loop. For detailed information on programming this register see section: “Parameter Programming”.

Table 60. FREQGAIN2

Name	Bits	R/W	Reset	Description
FREQGAIN2	3:0	RW	1010	Bandwidth of the frequency recovery loop

AMPLGAIN

This register controls the bandwidth of the amplitude tracking loop. For detailed information on programming this register see section: “Parameter Programming”.

Table 61. AMPLGAIN

Name	Bits	R/W	Reset	Description
AMPLGAIN	3:0	RW	0110	Bandwidth of the amplitude recovery loop

TRKAMPLHI, TRKAMPLLO

This register holds the current value of the amplitude of the received signal.

Table 62. TRKAMPLHI, TRKAMPLLO

Name	Bits	R/W	Reset	Description
TRKAMPL	15:0	R	–	Current amplitude tracking value; Used for RSSI2 calculation

This is a signed 16 bit register and should only be read using the 16 bit read access sequence. See section: “RSSI” for details on how to use this register to derive a high resolution RSSI value (RSSI2).

TRKPHASEHI, TRKPHASELO

This register holds the current value of the phase offset to the received signal.

Table 63. TRKPHASEHI, TRKPHASELO

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

This is an unsigned 16 bit register (only 12 bits used) and should only be read using the 16 bit read access sequence. $(TRKPHASE / 2^{11}) \cdot \pi$ converts the register contents to radians.

TRKFREQHI, TRKFREQLO

This register holds the current value of the frequency offset of the received signal.

Table 64. TRKFREQHI, TRKFREQLO

Name	Bits	R/W	Reset	Description
TRKFREQ	15:0	R	–	Current frequency tracking value; Used for AFC

This is a signed 16 bit register and should only be read using the 16 bit read access sequence.

The current frequency offset estimate is

$\Delta f = (TRKFREQ / 2^{16}) \cdot BITRATE$

For a description of the special handling required for this register to guarantee correct PSK reception see the section PSK Frequency Lock.

APEOVER

This register gives you software control over the reference and the oscillator by overriding the automatic power enable of these blocks. The register should only be used in accordance with Table 3: PWRMODE and APEOVER Register States.

Table 65. APEOVER

Name	Bits	R/W	Reset	Description
APEOVER	7:0	RW	0x00	APE override settings

PLLVCOI

This register is used to control the current through the synthesizer VCO.

Table 66. APEOVER

Name	Bits	R/W	Reset	Description
VCO_I	2:0	RW	100	Current through synthesizer VCO; Leave at default

PLLRNG

This register is used to control the synthesizer VCO auto−ranging internal settings.

Table 67. PLLRNG

Name	Bits	R/W	Reset	Description
PLLRNG	0	RW	0	Synthesizer auto−ranging internal setting; **Must be set to 1**

REF

This register is used to program the master reference current of the AX5042.

Table 68. REF

Name	Bits	R/W	Reset	Description
REF_I	2:0	RW	011	Master reference current; Leave at default

RXMISC

This register is used to program internal settings of the receiver.

Table 69. RXMISC

Name	Bits	R/W	Reset	Description
RXIMIX	1:0	RW	10	Mixer current, **Must be set to 01**

REFERENCES

[1] ON Semiconductor. AX5042 Datasheet, see http://www.onsemi.com
[2] ON Semiconductor. AX5042 Evaluation Software, see http://www.onsemi.com
[3] LAN MAN Standards Committee. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low−Rate Wireless Personal Area Networks (LR−WPANs). IEEE Computer Society, 2003.
[4] Wikipedia. High−Level Data Link Control. http://en.wikipedia.org/wiki/HDLC
[5] ON Semiconductor. AX5042/AX5051 Preamble Calculator, see http://www.onsemi.com

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910
Japan Customer Focus Center
Phone: 81−3−5817−1050

ON Semiconductor Website: **www.onsemi.com**

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative