Freescale Semiconductor
Application Note

Document Number: AN4983
Rev. 0.0, 09/2014

Operation of the AFC in the Freescale MC12311 and MKW01 Sub-GHz Transceivers

By Hiroshi Hiraga, Kenny Huang, Cesar Manzanarez, Koichi Matsuo, Bill Poole
Freescale Wireless Connectivity Application Engineering

1 Introduction

The MC12311 and MKW01 are highly-integrated, cost-effective, system-in-package (SIP), sub-1 GHz wireless node solutions. The MKW01 features a low-power ARM® Cortex M0+ 32-bit MCU together with an integrated RF transceiver that operates over a wide frequency range including the license-free Industrial, Scientific, and Medical (ISM) frequency bands at 315 MHz, 433 MHz, 470 MHz, 868 MHz, 915 MHz, 928 MHz, and 955 MHz. The MC12311/MKW01 configurations allow designers to minimize the use of external components. (This application note generally also applies to the earlier MC12311 that features an S08QE 8-bit microcontroller integrated with an RF transceiver nearly identical to that of the MKW01.)

Contents

2 Frequency Error

The most common implementation of the MKW01 or MC12311 is to attach a 32 MHz (30 MHz in some applications) crystal (Y1) to the on-chip reference oscillator at XTA and XTB, with caps (C22, C25) to ground.

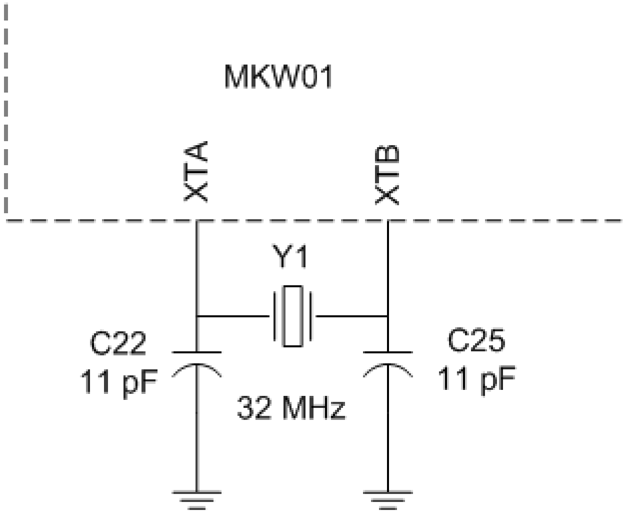

Figure 1. Reference Crystal Circuit

The reference crystal and capacitor combinations on most cost-competitive products can result in several parts per million reference error, which leads to several kHz error at the operation frequency in the 900 MHz band. In narrow bandwidth applications these errors can affect the performance of a radio communications link. If this error is greater than about 5% of the signal bandwidth it can severely degrade performance and sensitivity. For wide modulation bandwidth this frequency error can usually be ignored. For narrow bandwidth applications some method of eliminating the error can improve reliability:

- The use of a low tolerance TCXO in the product instead of a simple crystal can reduce frequency error to an insignificant level. This will increase BOM cost and circuit layout area.
- A technique for tuning the product in factory test can be implemented to set all products to the correct frequency in manufacturing. In production test, the automated test equipment should measure the frequency and save the measured error in non-volatile memory. In actual operation, each time a frequency is to be programmed, the saved frequency error should be recalled by firmware and the programming word corrected prior to writing to the transceiver register. This technique will not correct for aging or temperature-induced frequency variations in later operation.

Neither of the above will ensure reliable communications for a frequency-corrected transceiver communicating with an uncorrected transceiver.

In situations where one transceiver must communicate with another and one or both have a frequency error a mechanism should be used to actively tune each receiver onto the frequency of the incoming signal. The RF transceiver integrated into the MKW01 includes an AFC (Automatic Frequency Correction) block that measures the incoming signal's difference in frequency, then controls the local oscillator frequency in the receiver to tune that signal into the center of the receiver's own passband.

NOTE

> For some applications such as Busy Channel Detect or Carrier Sense, AFC is not helpful and one of the above techniques to set a more accurate frequency will improve reliability.

The function, programming, and operation of the AFC will be addressed below.

3 Automatic Frequency Correction (AFC)

As described in Section 5.7.13 through 5.7.15 of the MKW01 reference manual, the AFC circuit uses an input from the Frequency Error Indicator (FEI) block. The FEI detects the frequency error of the local oscillator (LO) compared with the carrier frequency of the modulated signal at the input of the receiver. For proper performance, the frequency error measurement operation must be performed during reception of the preamble and the sum of the frequency offset and the signal bandwidth must be lower than the receiver's base band filter bandwidth (signal bandwidth is 2*Fdev +BR), detailed below.

4 Frequency Error Indicator (FEI)

The Frequency Error Indicator block stores the result and that FEI value is used by the AFC block, which then subtracts the value from the RF Carrier Frequency registers, driving the LO to the desired frequency. (The FeiValue and AfcValue registers can be read by firmware, if desired.)

5 Launching AFC

AFC can be launched as needed by the MCU under firmware control, or it can be configured to launch each time the receiver is enabled:

- Launches each time the receiver is enabled, if AfcAutoOn = 1
- Launches upon firmware request, by setting bit AfcStart in RegAfcFei, if AfcAutoOn = 0

The AFC value is retained and used the next time receive is started. Alternately, the AFC value can be cleared by firmware with the bit AfcClear or, if AfcAutoOn = 1, it can be automatically cleared at each relaunch of AFC by setting AfcAutoClearOn = 1. This allows the designer to choose to start AFC from the programmed center frequency each time or from the last used corrected frequency. The latter is likely a better choice for a receiver that expects signals from the same transmitter each time and frequency error tends to be about the same every transmission, the former could be a better choice to receive signals from multiple transmitters or where frequency error can be more random.

Under circumstances where a string of packets is transmitted from the same transmitter over a short period of time, it may be useful to perform AFC on the first packet and then turn AfcAutoClearOn off or even have firmware read the AFC value and correct the frequency and turn AFC off for the subsequent packets to prevent false triggers in between later packets.

6 Receive Bandwidth during AFC operation

The receiver's baseband filter bandwidth is set using the register RXBW, (signal bandwidth = 2*Fdev +BR). RXBW just wide enough to pass the desired signal is best for sensitivity but does not allow for any frequency error. Therefore, the receiver has an alternate receiver bandwidth setting to accommodate large LO drifts, which is used during the AFC phase. If the expected received signal may be out of the receiver bandwidth, a wider channel filter bandwidth can be programmed in RegAfcBw. This will raise the receiver noise floor, which will impact sensitivity. Also care should be taken that AFCBW is not set so wide that adjacent channel signals are inadvertently picked up.

7 DC Cancellation Circuit

The receiver DC Cancellation circuit (described at the end of this document) has both a normal setting and an alternate setting used during AFC operation. The normal and AFC DC Cancellation frequencies are recommended to be set at 4% of the RxBw and RxBwAfc, respectively.

8 False Triggering of the AFC

When using AFC, there is a risk that the receiver will falsely trigger on an interference signal, an adjacent channel signal or even simple noise at the receiver input. This can cause the receiver, the FEI, and the AFC to start and immediately tune to the wrong frequency. This means that there will be no valid signal there to demodulate and the receiver will not be available when a desired signal does appear.

9 Handling False triggers

To prevent false triggering, a number of steps can be taken:

- Sufficient Preamble Length
- Selection of RssiThreshold
- Use of SyncAddressMatch signal after the RxReady signal.
- Selection of AFCBW (described above.)

9.1 Sufficient Preamble Length

The transmitted packet or signal begins with a preamble. The receiver starts when energy exceeds the RSSI threshold, just a few bits into the preamble. During the preamble time, the following time-consuming processes will occur before RxReady can be asserted:

1. Group delays of the various filters in the receive path
2. The AGC must settle
3. The AFC must detect the frequency error and correct the PLL
4. Further filter group delay

All of the above must take place during the preamble. The AFC functions better with a longer preamble. Expect a preamble of at least 5 bytes to be required.

9.2 RSSI threshold

The receiver stays in WAIT mode, until RssiValue exceeds RssiThreshold for two consecutive samples.

The RssiThreshold should be set well above the noise level and the level of likely interference so that the receiver only triggers when a desired signal arrives. The threshold should not be so high as to prevent detection of low-level desired signals. Keeping in mind that the noise level will be higher during AFC operation if the AFC Bandwidth is greater than the operational RX bandwidth, the threshold should be set accordingly.

9.3 Using RxReady and SyncAddressMatch

With the concern that the receiver may falsely trigger on noise, which would cause the AFC to tune the receiver off frequency rendering it incapable of receiving a desired signal until a RX restart or some other event causes it to tune closer to the channel frequency, a mechanism is needed to identify if the received signal is real and if not to reset the receiver. The receiver has a number of flags and timers that can be used. These flags are asserted by the RF transceiver, mapped to various DIO pins in accordance with Tables 7-2 and 7-3 of the MKW01 reference manual and Table 1 below, and routed off-chip to appropriate MCU GPIOs as shown in the hardware connections diagram Figure 7 below. DIOs must be mapped and GPIOs with interrupts assigned in firmware (see Figure 5, the “Start RX Mode” flowchart, below).

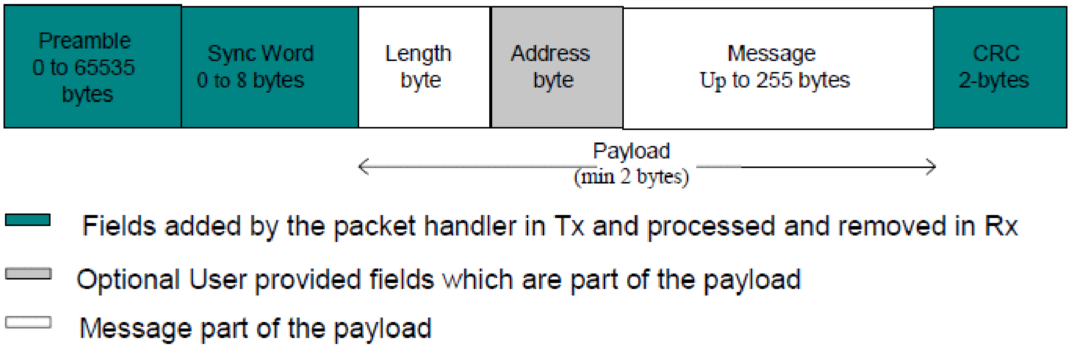
Preamble
0 to 65535
bytes
Sync Word
0 to 8 bytes
Length
byte
Address
byte
Message
Up to 255 bytes
CRC
2-bytes
Payload
(min 2 bytes)
Fields added by the packet handler in Tx and processed and removed in Rx
Optional User provided fields which are part of the payload
Message part of the payload

Figure 2. Typical packet format

As can be seen in the above packet format timing diagram (extracted from Figure 7-10 of the MKW01 Reference Manual) the packet starts with a Preamble and sync word then a payload followed by CRC.

Per the simplified timing chart in Figure 4, below, after the receiver has detected a signal exceeding the RX threshold, an RSSI_Threshold flag is set, Then AFC will occur during the preamble segment of the packet. After AGC, AFC and other receive start up functions are complete, the RxReady flag is set, which can be detected by the firmware running on the MCU. Then the sync word is transmitted. The incoming sync word is compared in the receiver to the programmed sync word, to detect a SyncAddressMatch. See Figure 3, below.

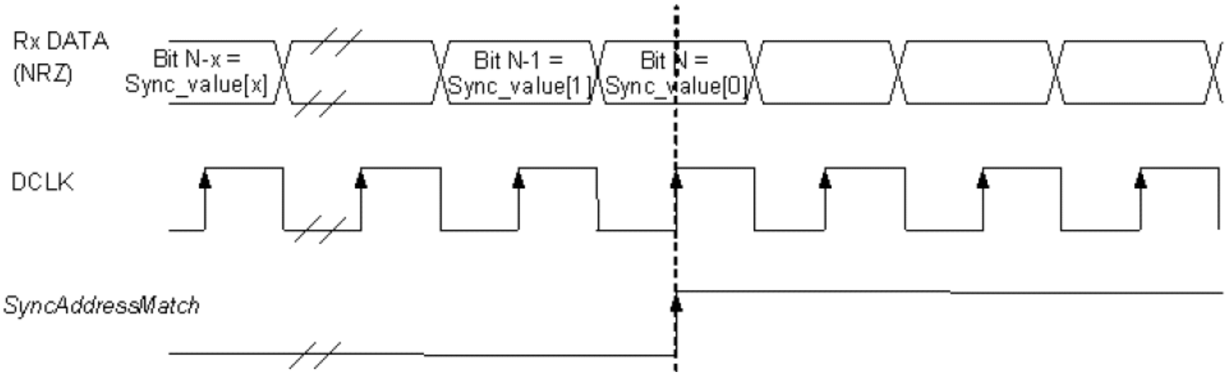

Figure 3. Sync Word Recognition

A timer can be set in firmware at RxReady (see Figure 4 below). If the SyncAddressMatch does not occur, then the timer will expire. This could indicate that the AFC falsely triggered or tuned to some off-frequency noise burst. To recover from this, the receiver can be restarted by firmware. See Figure 5.

If the SyncAddressMatch does occur, then likely a valid packet is being received. Some additional steps can be taken (whether using AFC or not) to read the early bytes in a packet and ensure the receiver does not hang up waiting for a partial, unintended or error laden packet to complete. Upon SyncAddressMatch, a timer can be set that is longer than the maximum expected payload. The FifoThreshold can be set to 1 byte, if the timer expires before that 1 byte is fed into the FIFO there is some problem and RX can be restarted by firmware. Also at this point, that one byte can be read before continuing to extract some protocol-specific information contained in the first byte. This process may be repeated for more information during the packet, depending on the protocol.

If the FifoLevel interrupt occurs, then data is being received and filling the FIFO. The max payload timer can be reset and another timer (PayloadReady) can be set allowing firmware to reset the receiver if PayloadReady never occurs.

An optional Cyclic Redundancy Check (CRC) check can performed on the received packet giving a PayloadReady interrupt if successful.

It is highly recommended that the MCU perform a CRC upon reception of payload data. This can be performed automatically by setting the bit CrcAutoClearOff in RegPacketConfig1. If the Sync Address is correct it is likely that any failure of CRC is due to a bit error rather than a false AFC trigger.

There are other operations performed in firmware as part of normal RX and TX operation that are beyond the scope of this AFC applications note.

See Figure 5 and Figure 6 below for flow charts of the above functions.

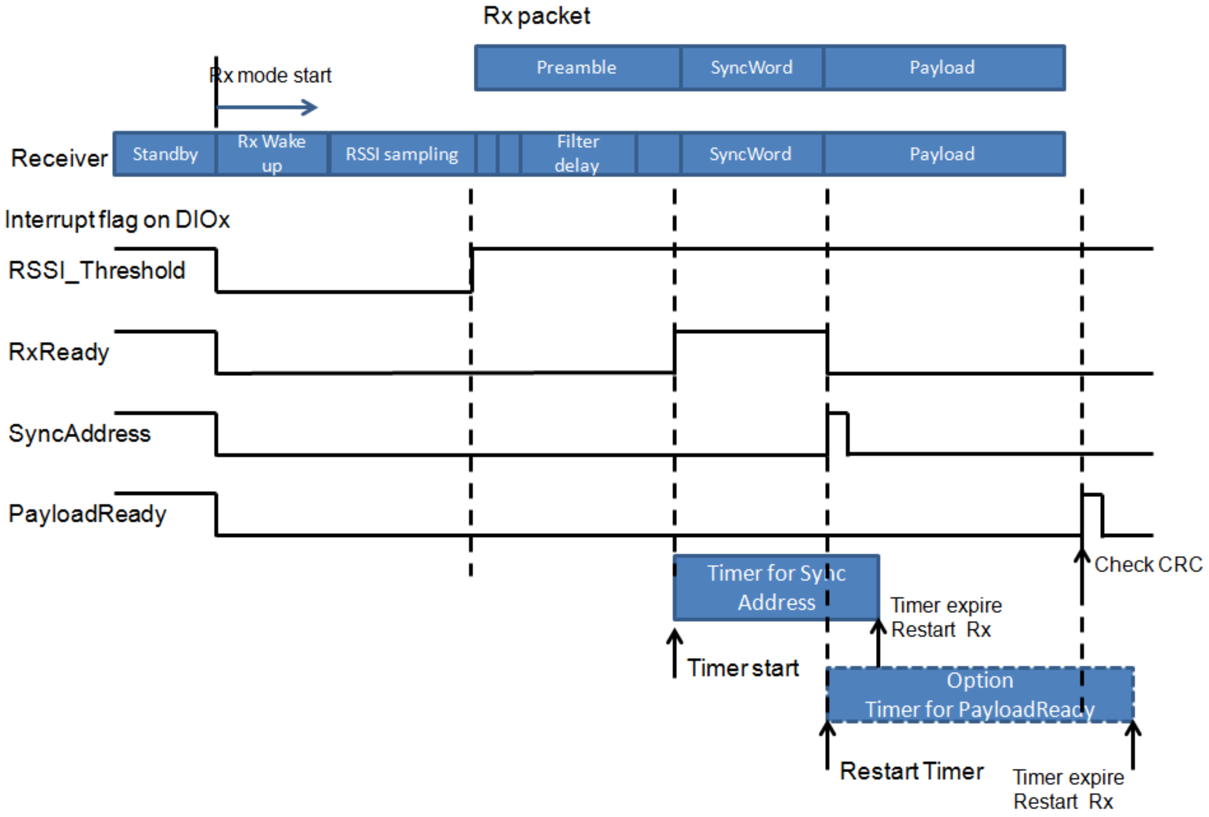

Figure 4. Timing chart of interrupt flags and timeout function

Table 1. DIO mapping for signals used in various modes

Mode	DIO0	DIO1	DIO4
Packet Receive	SyncAddress or PayloadReady	FifoLevel	RXReady
Continuous Receive	SyncAddress		RXReady

The full DIO mapping is shown in tables 7-2 and 7-3 of the MKW01 Reference Manual (tables 6-2 and 6-3 of MC12311 RM). FifoLevel (routed to DIO1 as shown) is used in detecting the payload in packet modes. The flow charts in Figure 5 below show the different interrupt handling depending on how that pin is programmed.

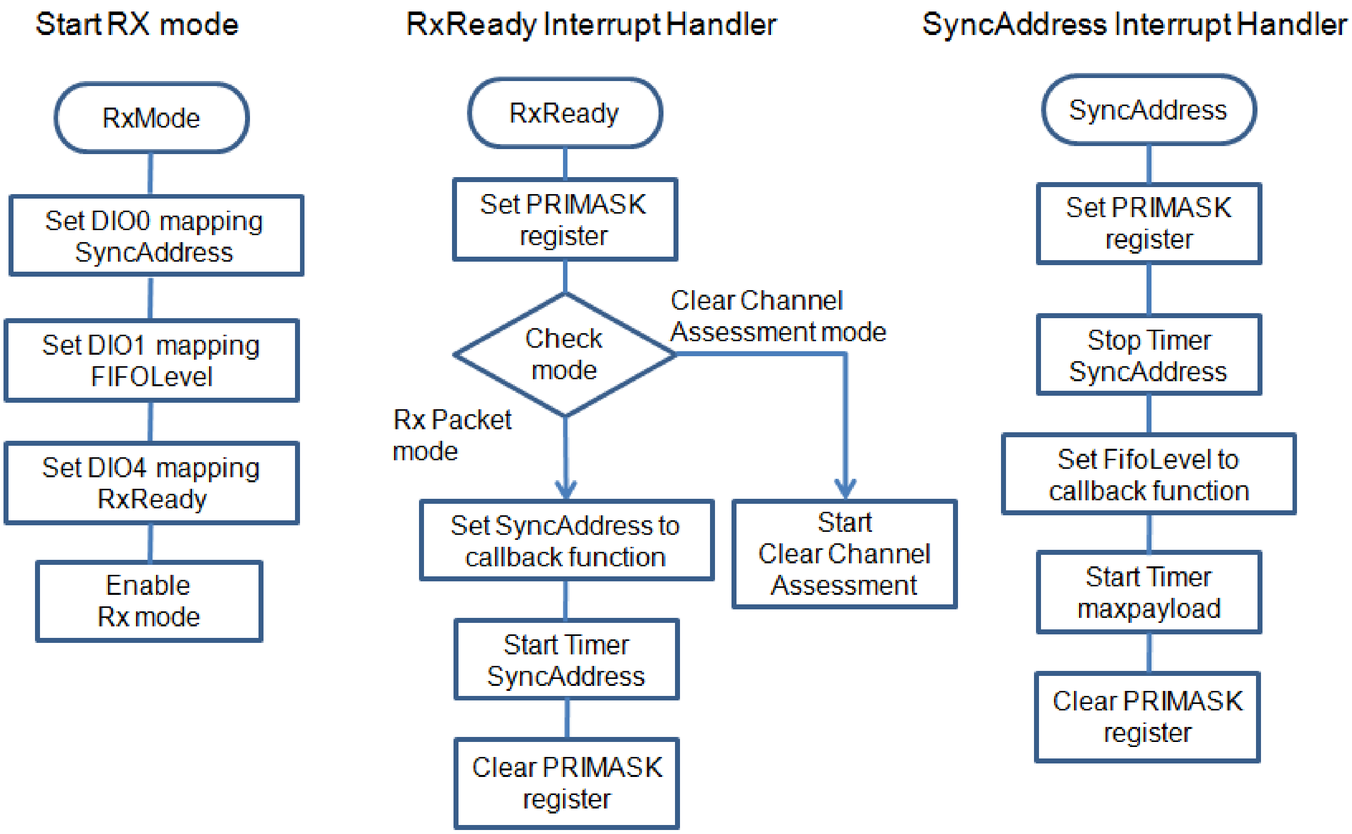

Figure 5. Interrupt handlers flow 1

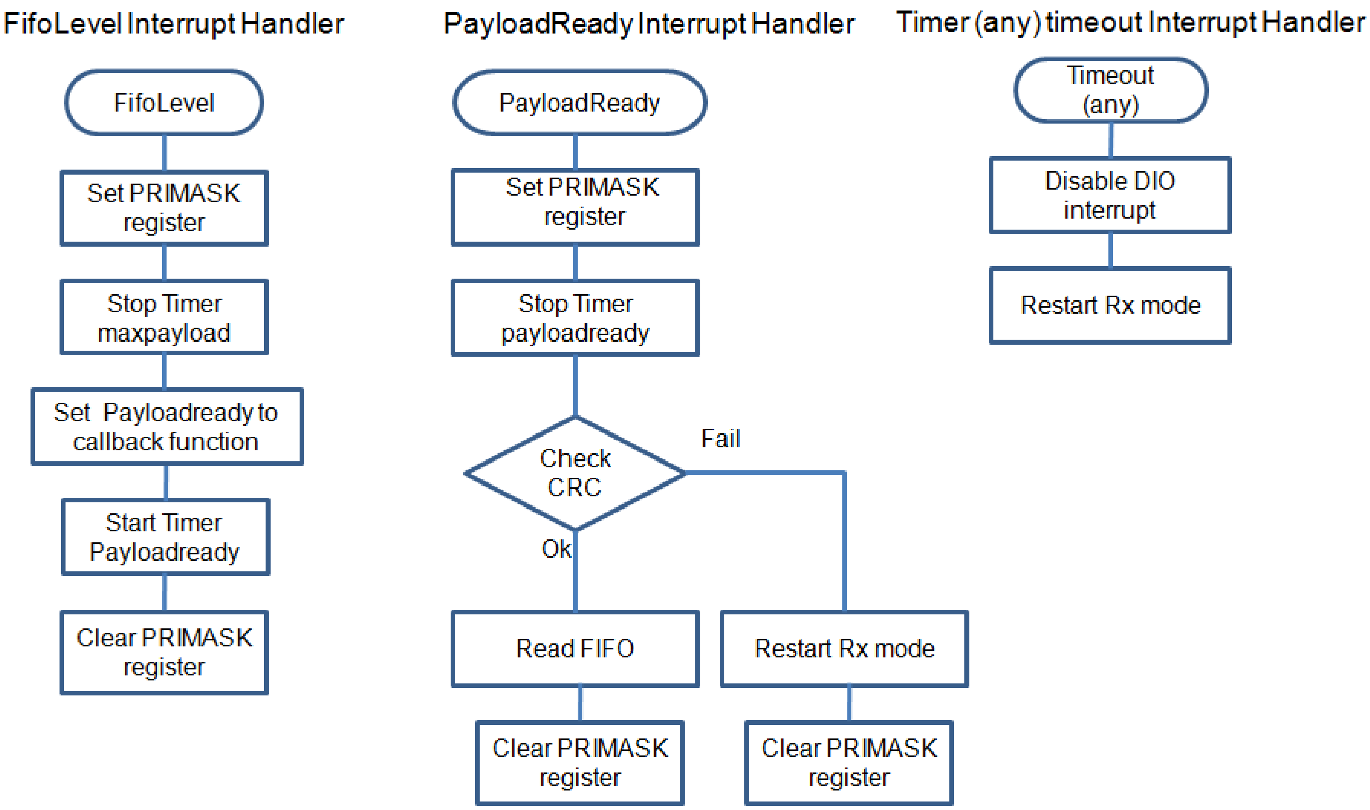

Figure 6. Interrupt handlers flow 2

10 Required Hardware Connections

To implement the timers, the SyncAddress, PayloadReady, RxReady as well as FIFO and timeout signals from DIOs should be routed to GPIO inputs with the Pin interrupt function enabled. CLKOUT should also be routed to the MCU EXTAL0. Recommended connections are shown in the diagram in Figure 7 and are listed below.

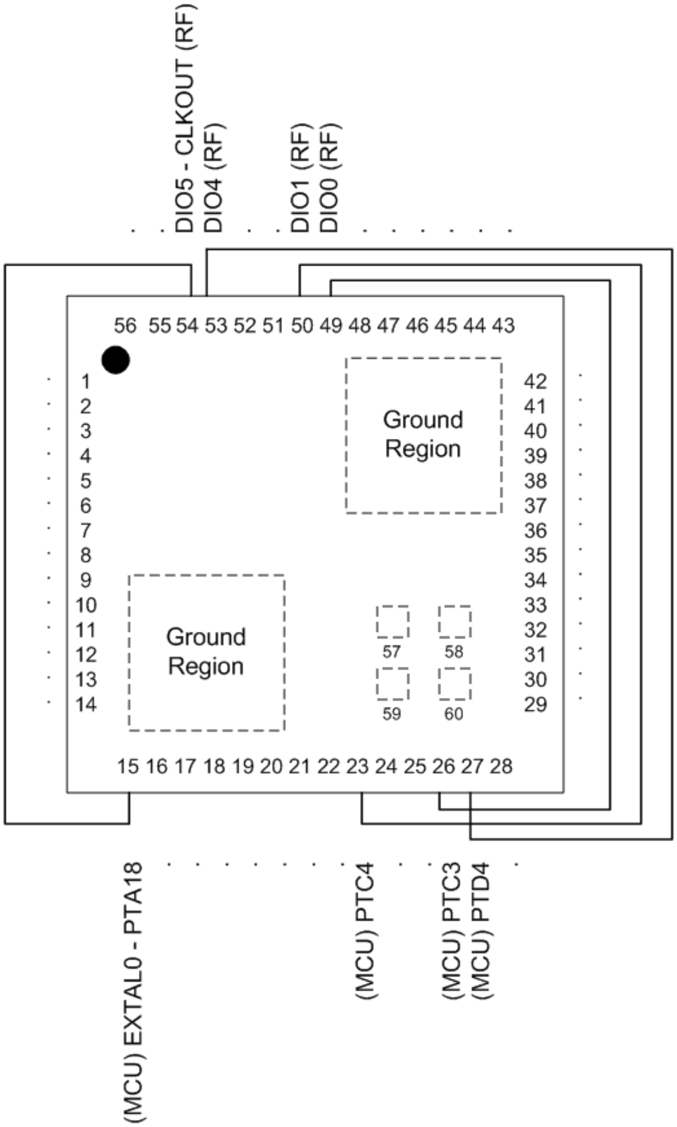

Figure 7. Hardware connections relevant to software control of AFC

- DIO0: DIO interrupt flag signal, see table 1
- DIO1: DIO interrupt flag signal, see table 1
- DIO4: DIO interrupt flag signal, see table 1
- CLKOUT: MCU clock derived from the RF Transceiver crystal oscillator.

11 Measurement of AFC performance

Performance of the MKW01 was measured with AFC on and AFC off and is detailed below.

Table 2. Hardware, RF, and Modulation Configuration used in Testing

Transceiver setting	Value
Frequency channel	924 MHz
Crystal frequency	30 MHz
Bit rate	50 kbps
Frequency deviation	25 kHz
Modulation type	FSK
Modulation shaping gaussian filter BT	1.0
Receiver channel bandwidth (RxBW)	156.24 kHz
Dc cancel frequency	0.5% x RxBW

Table 3. Register settings used in testing

Register	Bits	Variable name	Mode	Measurement setting	Description
RegAfcCtrl (0x0B)	5	AfcLowBetaOn	RW	1	LowBetaOffset is enabled
RegAfcBW (0x1A)	7–5	DccFreqAfc	RW	101	Dc offset frequency is set to 0x5%
	4–3	RxBwMantAfc	RW	10	Channel filter bandwidth during AFC 156.25 kHz
	2–0	RxBwExpAfc	RW	001	Channel filter bandwidth during AFC 156.25 kHz
RegAfcFei (0x1E)	3	AfcAutoClearOn	RW	1	AFC register is cleared before new AFC phase
	2	AfcAutoOn	RW	1	AFC is performed each time AfcStart is set
RegTestDagc (0x6F)	7–0	ContinuousDagc	RW	0x20	Fading margin improvement 0x20 if AfcLowBetaOn =1 0x30 if AfcLowBetaOn = 0
RegTestAfc (0x71)	7–0	LowBetaAfcOffset	RW	0x07	LowBetaOffset is set to 7 x 488 = 3,416 kHz

12 Measurement Results

12.1 RX sensitivity versus frequency offset

With AFC off and on, RX sensitivity was measured versus frequency offset. The results are shown in the Figure 8, below.

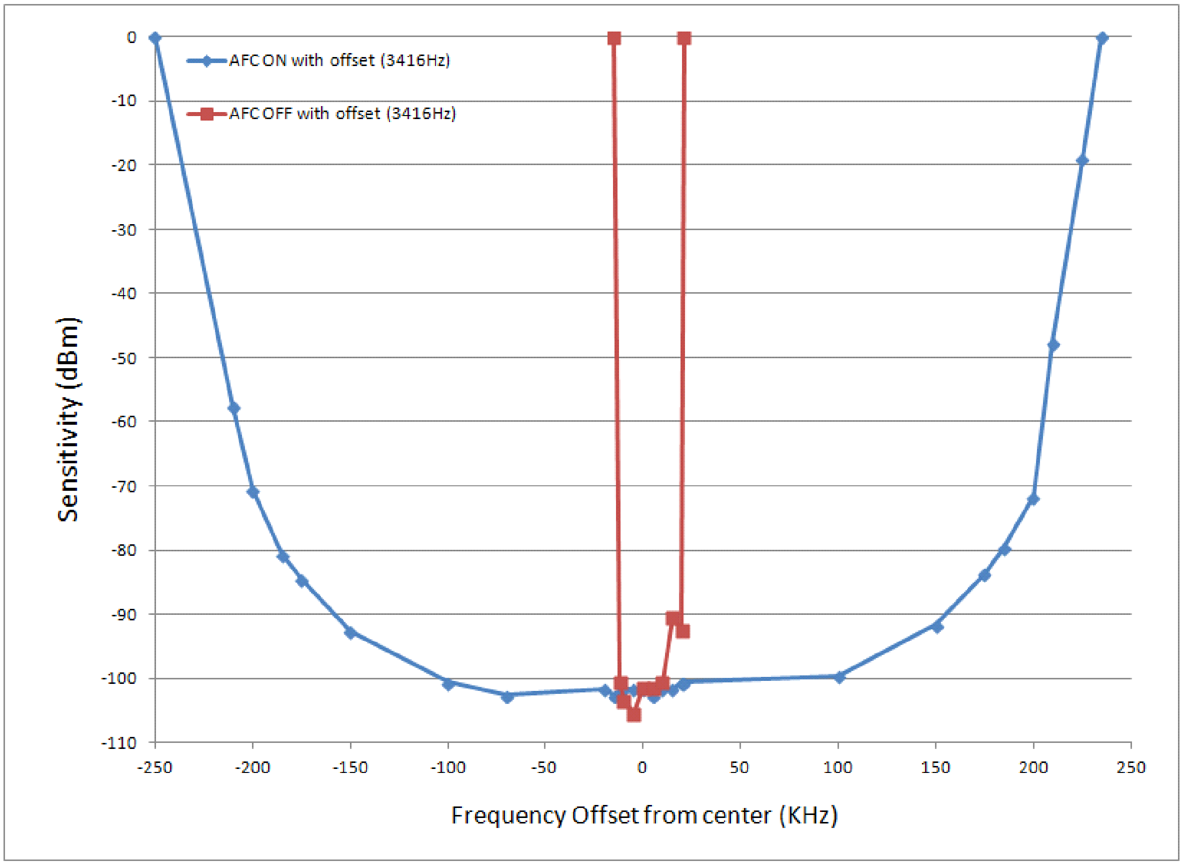

Figure 8. Sensitivity vs. Frequency offset (Both AFC ON and OFF, with 3416 Hz offset)

Figure 8 shows that with AFC operation disabled a 15 to 20 kHz error in frequency results in drastically worse sensitivity. This is about 10% of the RXBW. Whereas with AFC enabled about +/- 150 kHz error can be tolerated.

With AFC enabled the frequency error as much as +/- 100 to 150 kHz can be corrected for. Nearly 10 times as much frequency error can be tolerated without AFC. Care should be taken that AFCBW is not set so wide as to allow adjacent channel signals to be received.

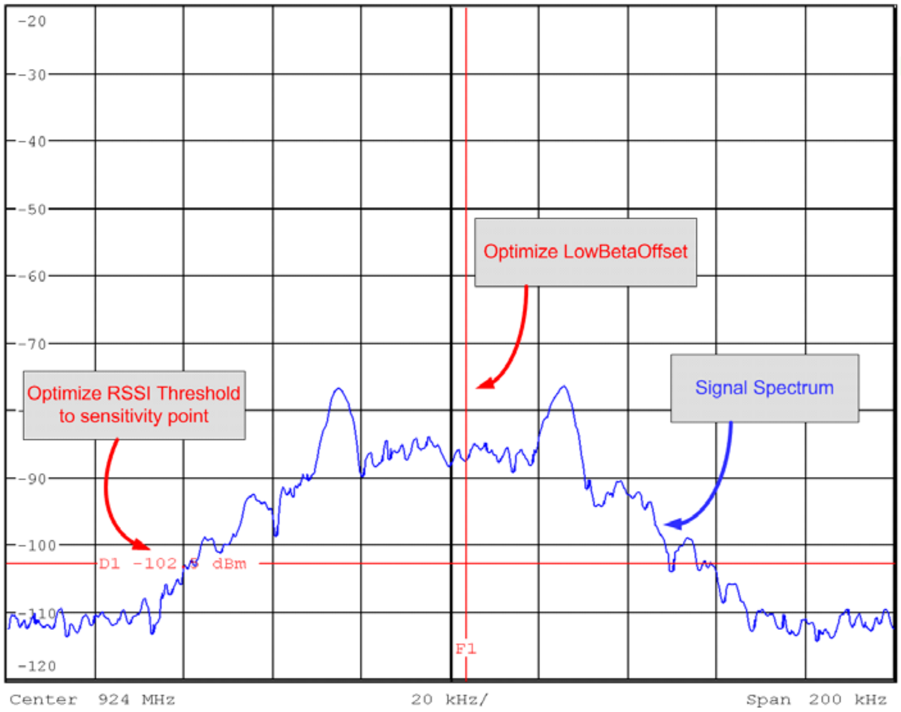

Figure 9. Signal Spectrum with LowBetaOffset and sensitivity illustrated

Figure 9, above, shows that the typical LowBetaOffset is much smaller than the signal bandwidth. The purpose is to move the center of the signal energy away from the DC Correction region where some attenuation occurs. By carefully selecting an optimum LowBetaOffset value, the receiver is offers better performance for signals with a low “Beta”. The recommended value is approximately 10% of expected transmitter frequency deviation.

Also shown is a display line of the previously measured sensitivity, well above the noise floor, where the RSSI threshold is set. Note the signal captured in the display is approximately 20 dB above threshold for clarity.

12.2 Probability of a False Trigger

The likelihood of a false trigger versus the amplitude of the receive signal was plotted.

Table 4. Probability of a false trigger event with different threshold levels (left side threshold is –103 dBm; right side threshold is –94 dBm)

Input power (dBm)	Packet success rate (%)	Avg reported RSSI (dBm)	SyncError
–1	100	–4	0
–2	100	–4	0
–3	100	–5	0
–4	100	–6	0
–5	100	–7	0
–6	100	–8	0
–7	100	–9	0
–8	100	–10	0
–18	100	–19	0
–28	100	–30	0
–38	100	–40	0
–48	100	–51	0
–58	100	–59	0
–68	100	–70	0
–78	100	–79	0
–88	100	–89	0
–89	100	–90	0
–90	100	–91	0
–91	100	–92	0
–92	100	–93	0
–93	100	–95	0
–94	100	–95	0
–95	100	–96	0
–96	100	–97	0
–97	100	–99	0
–98	100	–100	0
–99	100	–101	0
–100	98	–102	2
–101	98	–103	16
–102	94	–104	8
Threshold = –103 dBm	71	–105	6
–104	11	–107	3
–105	0	n/a	n/a

Input power (dBm)	Packet success rate (%)	Avg reported RSSI (dBm)	SyncError
–1	100	–4	0
–2	100	–4	0
–3	100	–5	0
–4	100	–6	0
–5	100	–7	0
–6	100	–8	0
–7	100	–9	0
–8	100	–10	0
–18	100	–19	0
–28	100	–30	0
–38	100	–40	0
–48	100	–51	0
–58	100	–59	0
–68	100	–70	0
–78	100	–79	0
–88	100	–89	0
–89	100	–90	0
–90	100	–91	0
–91	100	–92	0
–92	100	–93	0
–93	100	–95	0
Threshold = –94	100	–95	0
–95	20	–96	40
–96	0	n/a	n/a

As can be seen in the test data shown in Table 4 above, SyncErrors are observed to occur primarily due to false trigger events. False trigger events tend to occur with low signal levels if the RSSI threshold level is set to a level below sensitivity or ambient noise or in the presence of on-channel interference (left side of Table 4). It is recommended to optimize the RSSI threshold level to well above the RX sensitivity point and the expected level of any interference, false trigger errors only occur with signal levels just below the threshold level (right side of Table 4).

13 Conclusion

Enabling AFC functionality yields good receiver sensitivity performance across a wide range of frequency offset. This corrects for large frequency errors due to part tolerance, temperature, aging, etc., of the crystal and related components.

In an asynchronous wireless network communication system, the KW01 offers an RSSI threshold triggering feature to start the receive function. To avoid a false trigger event from noise, the RSSI threshold level should be optimized according to the receiver sensitivity and ambient noise. The KW01 also offers the ability to implement a timer function triggered on RxReady, SyncAddress, and PayloadReady to identify a false trigger and reject it and restart the receiver. A LowBetaOffset feature built into the AFC allows optimizing receiver performance on low modulation index signals that can be degraded by the DC cancellation circuitry.

14 Supporting material

14.1 References

- KW01 Reference Manual, MKW01xxRM.pdf, available from Freescale.com
- KW01 Datasheet, MKW01Z128.pdf, available from Freescale.com

14.2 PER versus BER

Two measures of signal quality are Bit Error Rate BER and Packet Error Rate PER (or its converse Packet Success Rate, PSR). BER is easier to measure in continuous mode with BER test equipment connected to the DATA line. PER/PSR are easiest to measure in packet mode, especially board to board testing on the bench or range testing. By transmitting 100 packets, the count of received packets is the PSR in percent, and 100%-%PSR = %PER. The formula to estimate PER versus BER for a packet of length N bits is shown for reference:

$$\mathrm{PER} = 1 - (1 - \mathrm{BER})^{N} \qquad \textit{Eqn. 1}$$

A BER of 0.1% is the standard for most testing. The PER/PSR equivalent of 0.1% BER is highly dependent upon packet length.

14.3 An alternate technique to artificially increase preamble length

Another technique (that is not included in the testing above) that can be used to increase Packet Success Rate is to set the DCLK to twice the operating bitrate while waiting for the preamble, then changing it the operating bitrate on the fly after receiving SyncAddress or RxReady. The expected preamble should be modified to reflect this. For instance, if the operating bitrate is 25 kHz, and the transmit preamble is 010101. The temporary doubled bitrate at 50 kHz means the expected preamble will read 001100110011. This will give twice as many bits for lock to occur and is useful when the preamble is insufficient for correct operation.

14.4 DC correction circuit

14.4.1 Baseband passband

The baseband filter in the receiver is shaped as shown. With a programmable cutoff frequency driving high end roll-off and a DCC correction circuit that causes a cut off at the low end also.

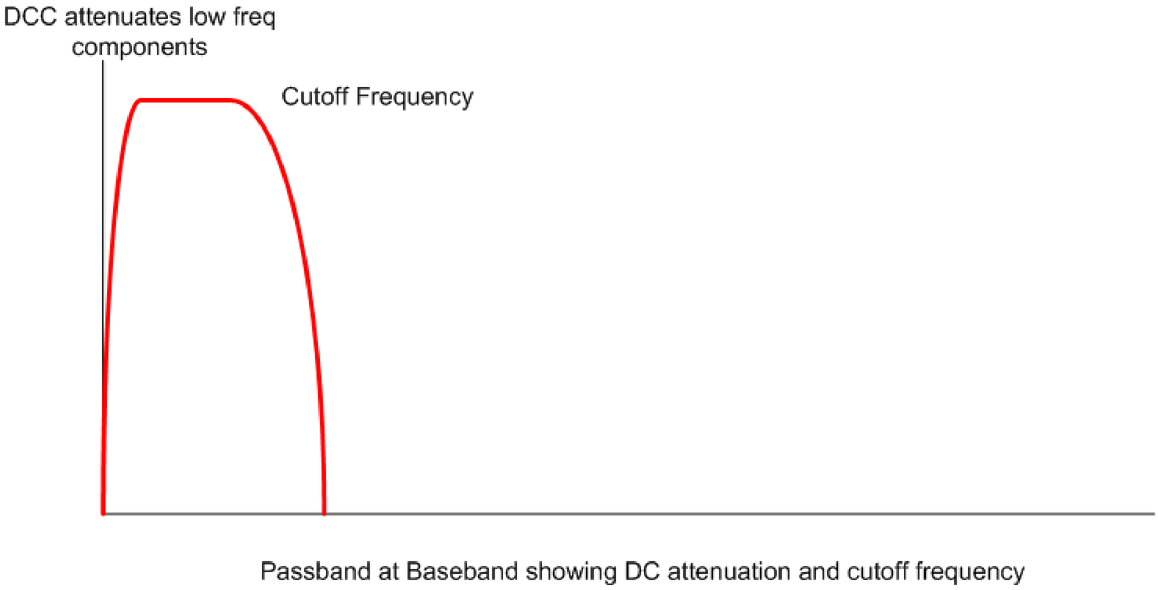
DCC attenuates low freq
components
Cutoff Frequency
Passband at Baseband showing DC attenuation and cutoff frequency

Figure 10. Passband at baseband

14.4.2 RF passband

Due to conversion from the LO, in which both the lower and upper sidebands are downconverted to baseband, the effective RF passband is as shown:

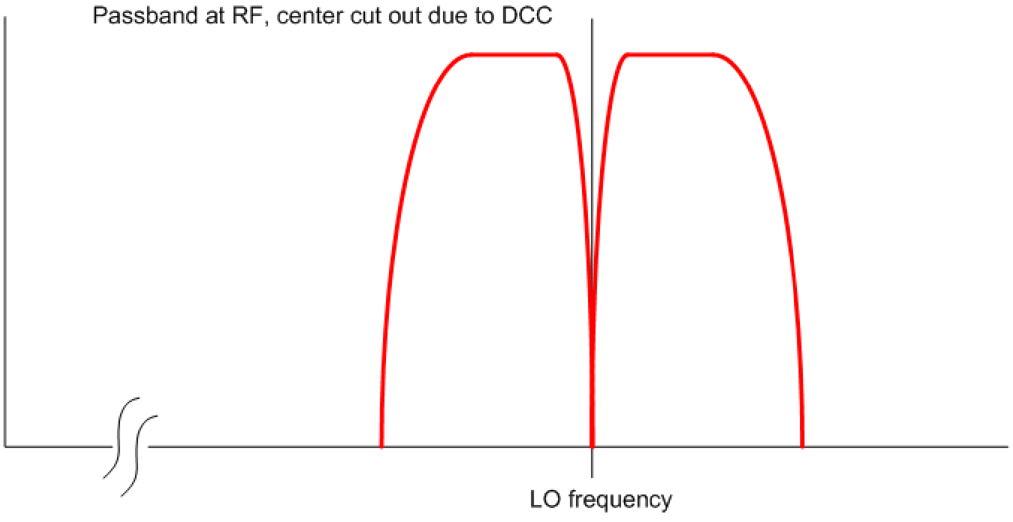

Figure 11. Passband at RF

The DCC notch appears as a cutout in the center of the RF passband.

14.4.3 Low beta signals

The MC12311/MKW01 sub-1 GHz transceivers demonstrate satisfactory BER or PER performance in FSK when the modulation index (Beta, β) is greater than about 2. (β = (2*Fdev)/BR). Such signals have the energy in their spectrum clearly separated into two frequency ranges making demodulation very easy. The receiver should be tuned to the same frequency as the transmitter. The DC correction circuitry (DCC), which actually creates a slight null in the center of the passband, has little effect.

Low beta signals can be considered those modulated with a Bitrate greater than Deviation. Low beta signals have a significant portion of their energy at the center of the signal spectrum, this can make demodulation more difficult in part due to the DCC. By tuning the receiver slightly off-center has been found to improve performance.

14.4.4 Frequency offset for low beta

When AFC is not used the RF frequency should simply be tuned slightly off from center. A good generalization is to offset the receiver's LO by 10% of the expected transmitter frequency deviation.

When AFC is used however, in normal mode it will tune to the center frequency. A feature is included in both MC12311 and MKW01 called LowBetaAfcOffset, which forces the receive center frequency to a programmable offset from the transmitters center frequency.

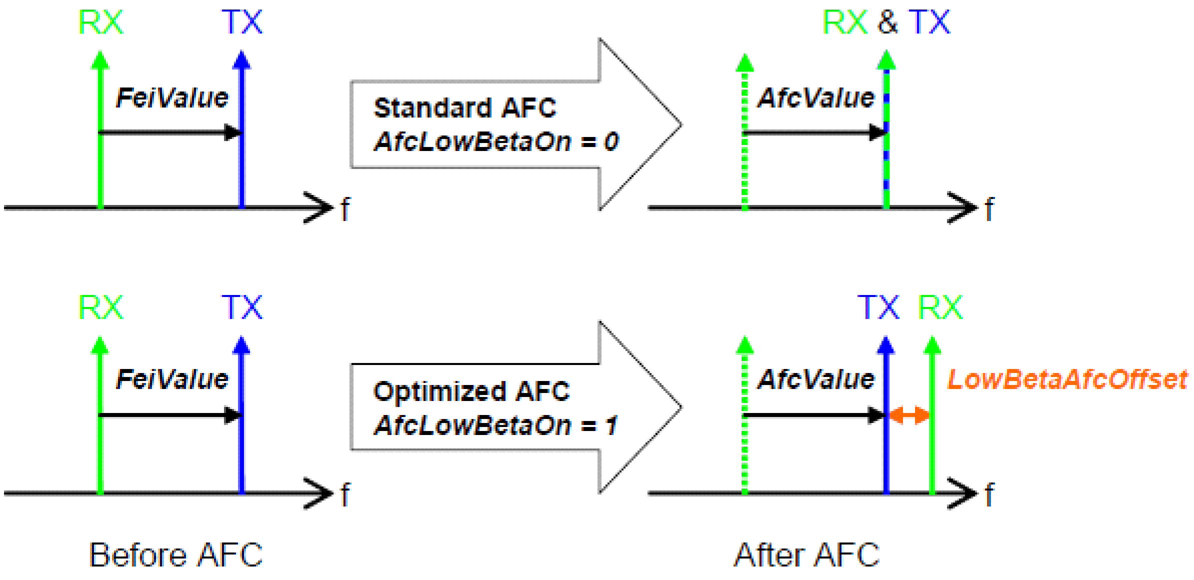

Figure 12. Frequency offset for low beta

Settings to be used:

- RX LO offset should be set to between 10% and 15% of Fdev.
- RX BW should be greater than the single sideband signal bandwidth + RX LO offset.
- DCC should be about 10% of the RX LO Offset.

14.4.5 AFC related registers

The registers related to the AFC (extracted from table 7.5 of the MKW01 Reference Manual and table 6-5 of MC12311 Reference Manual) are listed below.

Table 5. AFC related registers

Register	Bits	Variable Name	Mode	Default Value	Description
RegAfcCtrl (0x0B)	5	AfcLowBetaOn	RW	0	Programmable frequency offset for low index system
RegAfcBW (0x1A)	7–5	DccFreqAfc	RW	100	Dc Offset canceller frequency during AFC
	4–3	RxBwMantAfc	RW	01	Channel Filter bandwidth during AFC
	2–0	RxBwExpAfc	RW	011	Channel Filter bandwidth during AFC

Table 5. AFC related registers (continued)

Register	Bits	Variable Name	Mode	Default Value	Description
RegAfcFei (0x1E)	4	AfcDone	R	1	AFC finished.
	3	AfcAutoClearOn	RW	0	AFC register is cleared before new AFC phase
	2	AfcAutoOn	RW	0	AFC is performed each time AfcStart is set
	1	AfcClear	W	0	Clears the AfcValue if set in Rx mode
	0	AfcStart	W	0	Trigger an AFC when set
RegAfcMsb (0x1F)	7–0	AfcValue(15:8)	R	0x00	MSB of the AfcValue, 2's complement format
RegAfcLsb (0x20)	7–0	AfcValue(7:0)	R	0x00	LSB of the AfcValue, 2's complement format
RegTestDagc (0x6F)	7–0	ContinuousDagc	RW	0x30	Fading Margin improvement. 0x20 if AfcLowBetaOn =1 0x30 if AfcLowBetaOn=0
RegTestAfc (x71)	7–0	LowBetaAfcOffset	RW	0x00	AFC offset for low modulation index system

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.