

# **USER MANUAL**

## **AX-ConfigurationUtility for AXM0F343(with AX5043)**

Version 1.4

20211020



## Table of Contents

<b>1. Introduction.....</b>	<b>3</b>
<b>2. Installation .....</b>	<b>4</b>
<b>3. Create a new Project.....</b>	<b>5</b>
3.1.Import Eclipse Projects.....	5
3.2.Upgrading AXM0F343 CMSIS Pack .....	7
<b>4. AX-ConfigurationUtility Main Panel .....</b>	<b>9</b>
4.1.Wake on Radio Mode (WOR) .....	13
4.2.RX Synchronized with TX Mode.....	15
<b>5. Kit Configuration Panel .....</b>	<b>18</b>
<b>6. Pin Configuration Panel.....</b>	<b>20</b>
<b>7. PHY Panel .....</b>	<b>21</b>
<b>8. Framing Panel.....</b>	<b>24</b>
<b>9. Basic and Regulatory Tests Panel .....</b>	<b>28</b>
<b>10. Programming the Target boards .....</b>	<b>30</b>
<b>11. Comments on C-code Firmware .....</b>	<b>31</b>
11.1. File Structure .....	31
11.2. Define Statements for Tweaking the Firmware .....	31
11.3. SYNC Packet RX Timing .....	32
<b>12. XTAL/TCXO Precision, Frequency Tracking and Bandwidth.....</b>	<b>33</b>
12.1. How to set up AFC range, FSK deviation and RX bandwidth .....	33
Case 1: XTAL/TCXO precision, bitrate and carrier frequency are given .....	33
Case 2: Bitrate and carrier frequency and FSK deviation are given .....	38
12.2. Background information .....	40
<b>13. Revision History .....</b>	<b>42</b>

### 1. Introduction

The AX-ConfigurationUtility is a hands-on configuration and firmware source code generation tool for the AXM0F343.

It creates C-code firmware projects based on the AX-ConfigurationUtilityAPIManual (described in DOCU/ AX-ConfigurationUtilityAPIManual.pdf), which demonstrate the use of the AX5043 transceiver together with the AXM0F343 SoC for packet transmission and reception in several modes.

Using the simple GUI, the transmitter (MASTER) and receiver (SLAVE) can be easily configured into the desired mode. The generated Eclipse-compatible firmware can then be imported within the ON Semiconductor IDE and flashed via the standard AXM0F343 toolchain.

Further modification of the generated firmware source code can be done directly using the ON Semiconductor IDE. Additionally, the project can be at any time be reloaded into AX-ConfigurationUtility, to automatically modify the desired wireless parameters.

This document first illustrates installation requirements and the steps needed to create a new project within a GUI, and how to import it within the ON Semiconductor IDE. Afterwards, each panel of the GUI is described in detail with explanations and hints on how to build the desired radio firmware. Finally, a generic overview on the code and utilities is given.

## 2. Installation

### Hardware and requirements:

- **64-bit Windows 7 or later**
- **at least 4GByte RAM**
- **1GB free hard disk space**
- **free USB port**

### Software requirements:

- **Install the ON Semiconductor IDE and all sub-components. The IDE is used to edit and flash the firmware.**
- **Install the AX-ConfigurationUtility. The GUI is used to generate custom projects**
- **For AX-ConfigurationUtility to work correctly you need to import the ON Semiconductor AXM0F343 CMSIS Pack within the ON Semiconductor IDE first.**  
**Additionally, you need to install the ARM.CMSIS.5.8.0 Pack downloadable from [https://github.com/ARM-software/CMSIS\\_5/releases/tag/5.8.0](https://github.com/ARM-software/CMSIS_5/releases/tag/5.8.0)**
- **To verify and validate SoC AXM0F343, compiler such as ARM GCC (Version 5.0 and above) is required to be installed.**

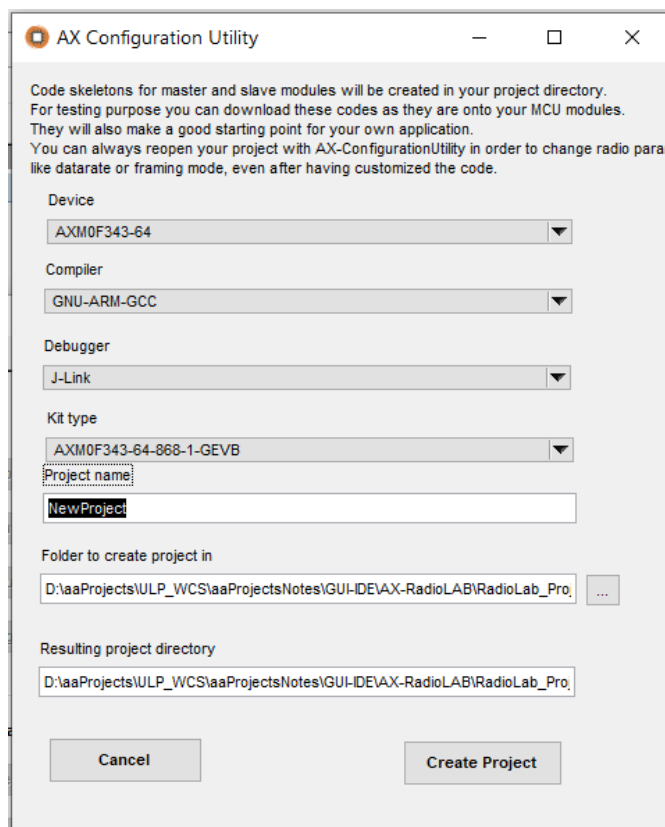
AX-ConfigurationUtility is included as a standalone installer. Please uninstall any previously installed versions of this software before starting a new installation.

At the end of installation, a reboot may be requested; however, this is not mandatory.

## 3. Create a new Project

The AX-ConfigurationUtility GUI can be used to create projects for AXM0F343.

Open the AX-ConfigurationUtility.exe to create a new project as below:



- Device: This populates the devices supported in the current AX-ConfigurationUtility GUI namely AXM0F343-64 and AXM0F343-256.
- Compiler: For either of the device the compiler supported is ARM-GNU-GCC.
- Debugger: JLINK (version 670e and above) is the only debug interface available with the current AX-ConfigurationUtility GUI.
- Kit Types: This option is also populated based on the selected device
  - For AXM0F343-64
    - AXM0F343-64-868-1-GEVB
    - AXM0F343-64-915-1-GEVB
  - For AXM0F343-256
    - AXM0F343-256-868-1-GEVB
    - AXM0F343-256-915-1-GEVB

### 3.1. Import Eclipse Projects

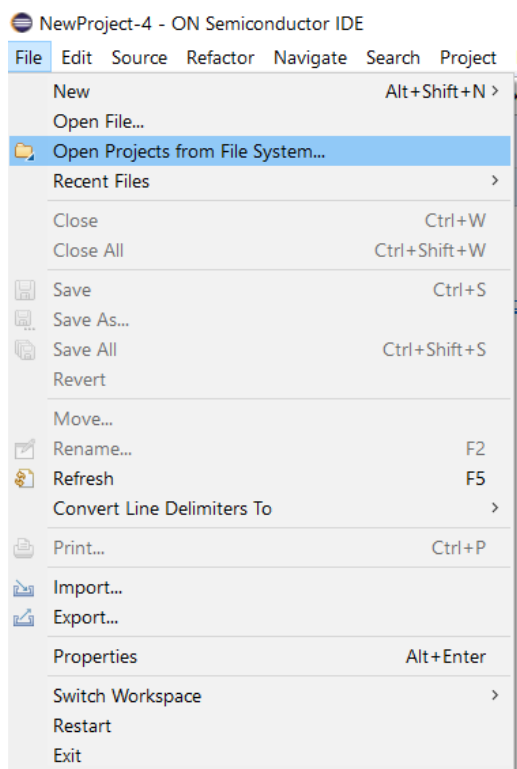
The project created using AX-ConfigurationUtility consists of folder templatefirmware. This directory consists of eclipse projects and template source code specific to AXM0F343-64 and AXM0F343-256 SoC.

## AX-ConfigurationUtility User Manual

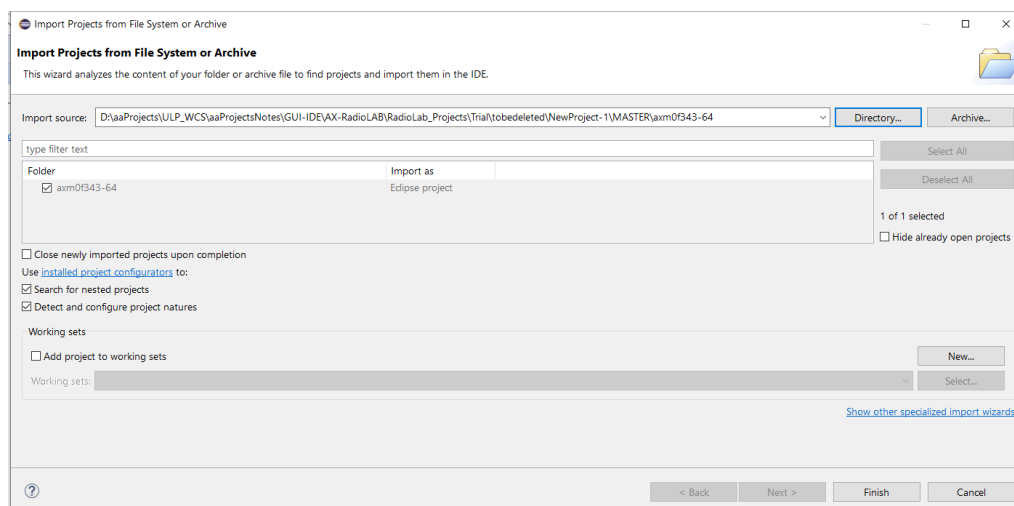
Based on the Target board, the eclipse project can be imported relatively using Projectdir\MASTER\axm0f343-64 or Projectdir\MASTER\axm0f343-256. Similar logic applies to Slave/Tests.

Eclipse projects for various configurations can be opened as below: -

- Open Projects from File System using Eclipse IDE



- Open Eclipse Project



- Enable Message Box Popup in Eclipse if File got updated outside of Eclipse Editor:  
Goto Window -> Preferences -> General -> Workspace and uncheck the option "Refresh on access" in Eclipse.
- To debug the driver firmware in the library included in the pack, path to the library source requires remapping as below
  - a. Goto Eclipse→Window→Preferences→C/C++→Debug→Source Lookup Path→Add
  - b. Add Path mapping and enter the below item in the path mapping
  - c. C:\Users\fg8y4b\ON\_Semiconductor\PACK on the LHS
  - d. C:\Users\ID\ON\_Semiconductor\PACK → ID to be replaced with the User's ID where pack is installed in respective PC's.

By doing the above steps, the firmware drivers source files within the F343 pack is available for debug.

### 3.2. Upgrading AXM0F343 CMSIS Pack

The Template firmware accesses CMSIS pack specific to the AXM0F343 SoC.

The Eclipse projects in Template firmware accesses the CMSIS pack drivers relatively by defining environment variable in the project settings namely AXM0F343\_CMSIS\_PACK\_ROOT. Currently, this variable is set to a value

"\${cmsis\_pack\_root}\ONSemiconductor\AXM0F343\1.4.0"

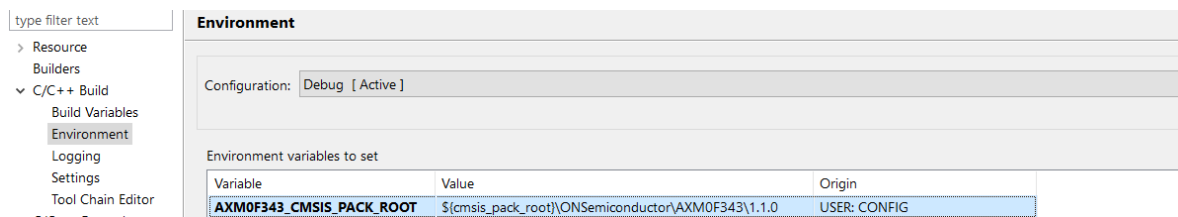
"1.4.0" denotes the PACK version that is configured. This needs to be updated to the upgraded version as below. For example: if the pack version is upgraded to 1.5.0, it should be modified in the eclipse project settings as below in the installation directory of

## AX-ConfigurationUtility User Manual

AX Configuration Utility located at C:\Program Files (x86)\ON Semiconductor\AX-ConfigurationUtility in 6 different locations for debug and release configurations:-

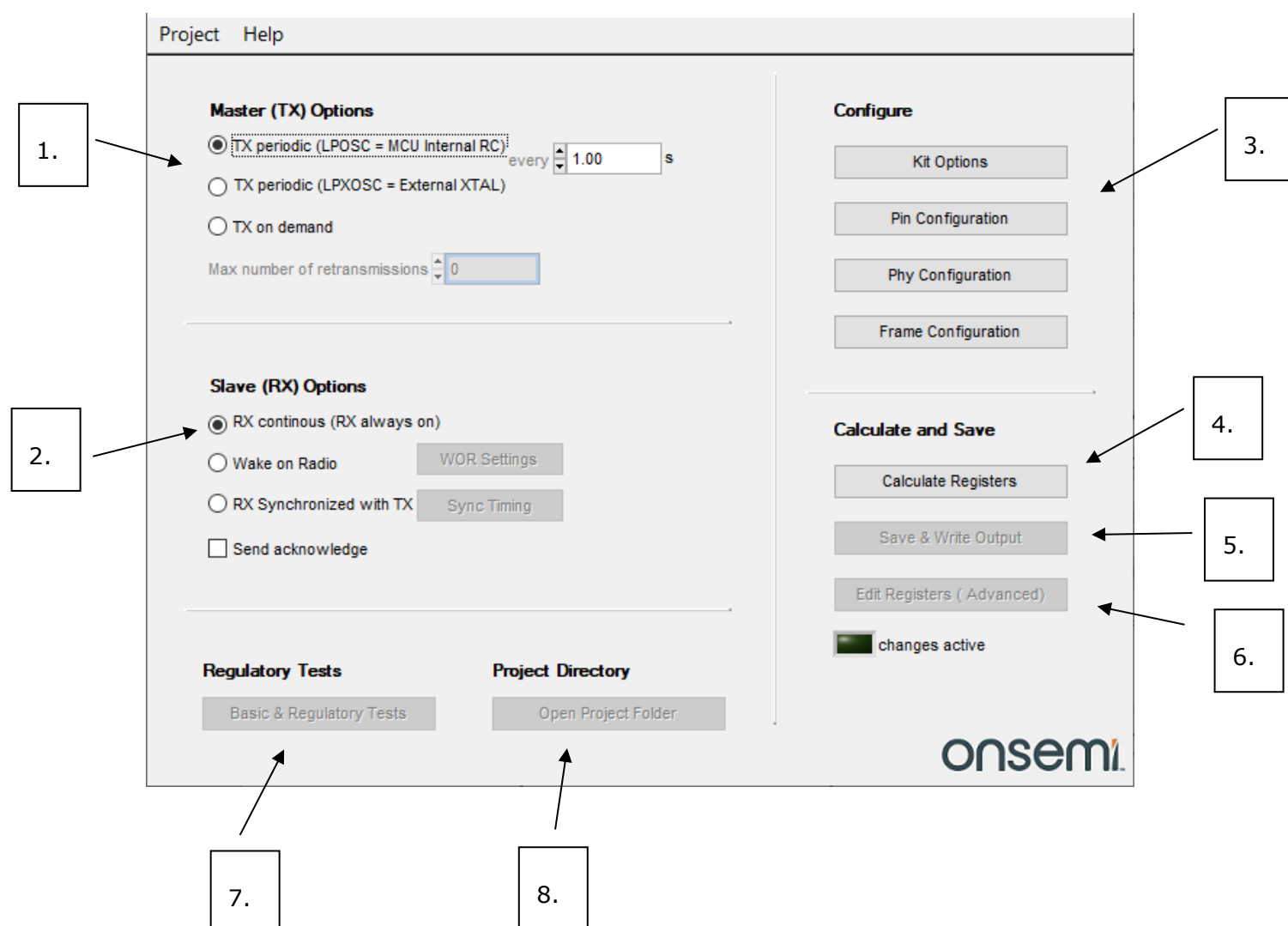
- e. Master
  - i. axm0f343-64
  - ii. axm0f343-256
- f. Slave
  - i. axm0f343-64
  - ii. axm0f343-256
- g. Tests
  - i. axm0f343-64
  - ii. axm0f343-256

The variable is defined in the environment section of the eclipse project in each of the above 6 different sets in debug and release configurations. The Variable can be accessed by navigating to Project→Properties→C/C++ Build →Environment





## 4. AX-ConfigurationUtility Main Panel



Start a new project using the **Project** menu. You must specify a project directory where AX-ConfigurationUtility will create firmware projects called MASTER, SLAVE and TESTS. To open an existing project, use the directory chooser dialog to descend into the project directory (where the axradiolabstate.xml file resides) and hit done. Note: do not open the template\_firmware\_504x project inside the AX-ConfigurationUtility installation directory, as this is the template from which the firmware is copied upon creating a new project.

Follow through the main panel step-by-step to configure your project:

1. Select the TX operation mode:

- **TX periodic (LPOSC):** Periodic packet transmission driven by the MCUs low power RC oscillator.
- **TX periodic (LPXOSC):** Periodic packet transmission driven by the MCU's low power 32kHz tuning fork crystal oscillator

- **TX on demand:** Packet transmission upon pressing a button.
2. Select the RX operation mode: **RX Continuous (RX always on):** Packet reception keeping the receiver continuously on. This mode is used for AC-powered systems and allows asynchronous packet reception with low latency.
    - **Wake on Radio:** Packet reception with the receiver waking up periodically and going immediately back to sleep if no signal is detected. The wake-up is driven by the Radio's low power RC oscillator and does not require any MCU activity. The MCU is only woken up once a valid packet has been received. This mode is well suited e.g. for infrequently operated remote-control applications and achieves very low (idle) current consumption. **Mind that preambles longer than the time between RX wakeups are necessary.** The latter also determines the RX latency.
    - **RX Synchronized with TX:** Periodic packet reception enabling the receiver only for the time slots in which packets are expected. This is appropriate whenever periodic data transmission is required. Timing is driven by the MCU's low power 32kHz tuning fork crystal oscillator and thus requires MCU activity. It is mandatory to use **TX periodic (LPXOSC)** on the transmitter side.
    - The **Send acknowledge** checkbox causes the slave module ("RX") to answer successful packet reception with an acknowledge packet. The master module ("TX") tries to receive the acknowledge packet. The smiley on the LCD/UART print indicates whether the acknowledge packet has been received. Failure is also signaled via LED1. Optionally the MASTER can retransmit a packet if acknowledge fails. This feature is enabled by setting **Max number of retransmissions** to a nonzero value. The acknowledge mode is the reason why the terminology MASTER and SLAVE (rather than TX and RX) has been chosen. The acknowledge feature can be selected in all TX and RX modes.
  3. Buttons to open configuration panels for Kit Configuration, Pin Configuration, PHY layer and Framing. The PHY panel covers physical parameters of the wireless link such as carrier frequency, data rate and the modulation scheme. The Framing panel deals with packet delimiting, CRC etc.
  4. **Calculate Registers** calculates register values to configure the selected device/Radio according to your Pin Configuration, PHY and Framing selections.
  5. **Save & Write Output:**

The MASTER and SLAVE firmware projects are created in your project directory (if not present yet) upon hitting the **Save & Write Output** button. Manual changes in the firmware projects are never overwritten by the GUI except for the configuration files in your\_projectdir /AX\_Radio\_Lab\_output/.

- Saves configuration and register values into your\_projectdir\axradiolabstate.xml, allowing you to reopen your project with AX-ConfigurationUtility.

- Outputs files config.c, configmaster.[ch] and configslave.[ch] to your\_projectdir\AX\_Radio\_Lab\_output\  
These files are included by / linked to the MASTER, SLAVE and TESTS firmware projects.
  - The firmware projects MASTER and SLAVE are written into your project directory if they are not present yet.
6. After calculating register values, you can optionally tweak values using **Expert Settings**. Click values in the rightmost column of the **Expert Settings** panel to edit. Tweaked values are sticky. They keep overriding the defaults even if the defaults have changed (e.g. upon making changes in the PHY panel and hitting **Calculate Registers**). Tweaked values are highlighted in yellow (or green if they accidentally correspond to the default values). The **changes active** indicator in the main panel reminds you that tweaked values are being used. Double click a tweaked register (column 1, 2 or 3) to return to the default value. Hitting **Reset Changes** at the bottom of the **Expert Settings** panel or clicking the **changes active** indicator in the main panel drops all tweaks.
  7. Opens the **Basic & Regulatory Tests** panel, allowing you to transmit CW, simple patterns, or random data and to measure bit error rates.
  8. Open Project Folder – This enables the user to open the project directory in the explorer.

## 9. TX-RX Modes and timer usage

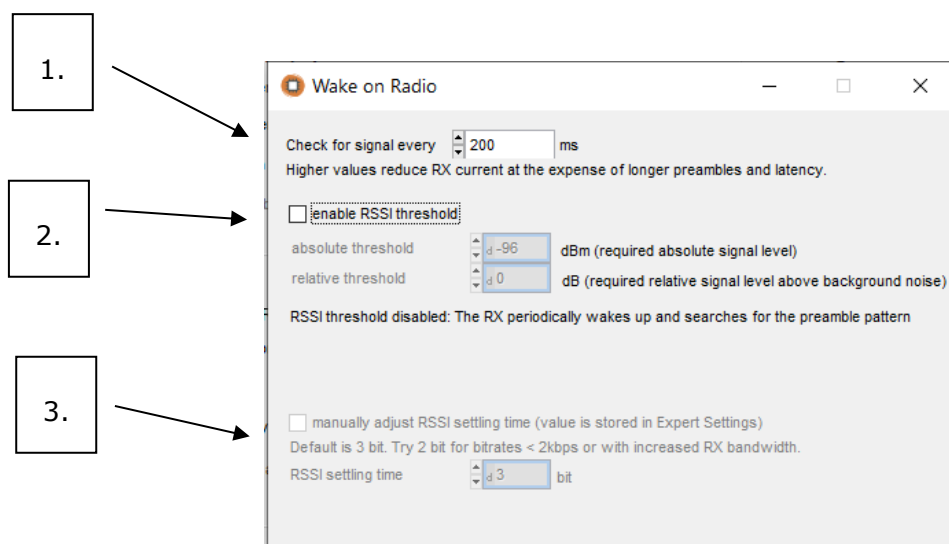
Mode, Purpose	Used timer	Radio / MCU activity <sup>1</sup>
<b>TX periodic (LPOSC),</b> Periodic packet transmission	MCU CMSIS Pack based timer infrastructure with timer0 clocked by MCU LPOSC (640hz RC oscillator)	MCU periodically wakes up, powers up the TX and initiates packet transmission
<b>TX periodic (LPXOSC),</b> Periodic packet transmission (used with RX Synchronized with TX)	MCU CMSIS Pack based timer infrastructure with timer0 clocked by MCU LPXOSC (32kHz tuning fork crystal oscillator)	MCU periodically wakes up, powers up TX and initiates packet transmission.
<b>TX on demand,</b> Packet transmission upon pressing a button or similar event.	MCU CMSIS Pack based timer infrastructure with timer0 clocked by MCU LPOSC (640hz RC oscillator) Only used for generating a timeout in ACK mode	MCU wakes up upon pressing a button, powers up the TX and initiates packet transmission.
<b>RX Continuous,</b> AC-powered systems (allows asynchronous packet reception)	MCU CMSIS Pack based timer infrastructure with timer0 clocked by MCU LPOSC (640hz RC oscillator) Only used for ACK timing and channel state polling.	RX is continuously on. Upon receiving a valid packet, it generates an interrupt waking up the MCU.
<b>Wake on Radio,</b> Remote control (asynchronous packet reception at low idle currents at the expense of long preambles and increased latency)	AX5043 LPOSC (640Hz RC oscillator) for autonomous wake-up of the AX5043 MCU CMSIS Pack based timer infrastructure with timer0 clocked by MCU LPOSC (640hz RC oscillator) Only used for generating a timeout in ACK mode	RX periodically wakes up and checks for signal. Upon receiving a valid packet, it generates an interrupt waking up the MCU.
<b>RX Synchronized with TX,</b> Low power periodic data transmission	MCU CMSIS Pack based timer infrastructure with timer0 clocked by MCU LPXOSC (32kHz tuning fork crystal oscillator)	MCU periodically wakes up, powers up the RX and waits for packet reception or timeout.

**Table 1: Summary of TX and RX modes and corresponding firmware projects for AXM0F343 SoC**

<sup>1</sup> MCU activity apart from timekeeping wake-ups due to the running CMSIS pack-based timer infrastructure.

## 4.1. Wake on Radio Mode (WOR)

In **Wake on Radio** mode the receiver periodically wakes up and searches the channel for a signal. In absence of signal the receiver is switched off again immediately to save power. If a signal is detected the receiver stays on and looks for a valid packet delimiter. The Radio performs this task autonomously. Only after a valid packet has been received and is waiting in the FIFO, the Radio wakes up the micro controller via an interrupt. A typical application of this mode would be an infrequently operated remote control. The "WOR Settings" button opens the Wake on Radio panel:



**Figure 1: Wake on Radio. WOR settings panel.**

1. This field configures how often the receiver wakes up and checks for signal. Waking up less often reduces the average RX (idle) current. However, it increases the necessary preamble length and average latency since the preamble needs to be at least as long as the gap between two successive wake ups.
2. The "enable RSSI threshold" option determines how the RX searches for signal upon wake up:
  - a. If this option is disabled, the RX directly scans for preamble pattern. Sensitivity in this mode is comparable to continuous RX operation.
  - b. If this option is enabled the RX first acquires an RSSI value. The scan for preamble pattern is performed only if RF energy is detected. This reduces current consumption (most notably at low data rates) since the RSSI measurement is faster than the scan for preamble pattern. The drawback is a reduction of sensitivity, since the RSSI threshold has to be set several dB above the sensitivity limit to prevent false positives. The RX proceeds to scanning for preamble if  $RSSI > \text{absolute threshold}$  AND  $RSSI \geq (\text{background RSSI} + \text{relative threshold})$ . Background RSSI is the RSSI average over past wake ups.

3. This field allows customization of the RSSI settling time. Longer settling times give more stable RSSI measurements at the expense of higher current consumption. Changes are written to the expert panel.

Make sure you understand the following points:

- An important part of the process of powering-up the receiver and deciding whether a preamble signal is present takes a time inversely proportional to the data rate. To achieve minimum current consumption, you should therefore use the maximum possible data rate (100kbps). The time the receiver searches for a preamble signal can be customized via the TMGRXPREAMBLE1 register in the expert panel.
- Frequency tracking (AFC) requires settling time, which adds to the receiver start-up time and therefore increases the average current consumption. Large data rates and precise frequency references render frequency tracking unnecessary.
- The average receiver (idle) current is proportional to the wake-up frequency. On the other hand, in the remote control mentioned above, the wake-up frequency determines how quickly the receiver will react.
- Mind that the current estimate displayed in the main panel is the idle current in absence of a signal. Thus, while the RX wakes up periodically, actual packet reception is the exception in this mode, occurring only e.g. if a remote-control button was pressed. This is in contrast to the **Synchronized with TX** mode; in which a packet is expected each time the RX wakes up.

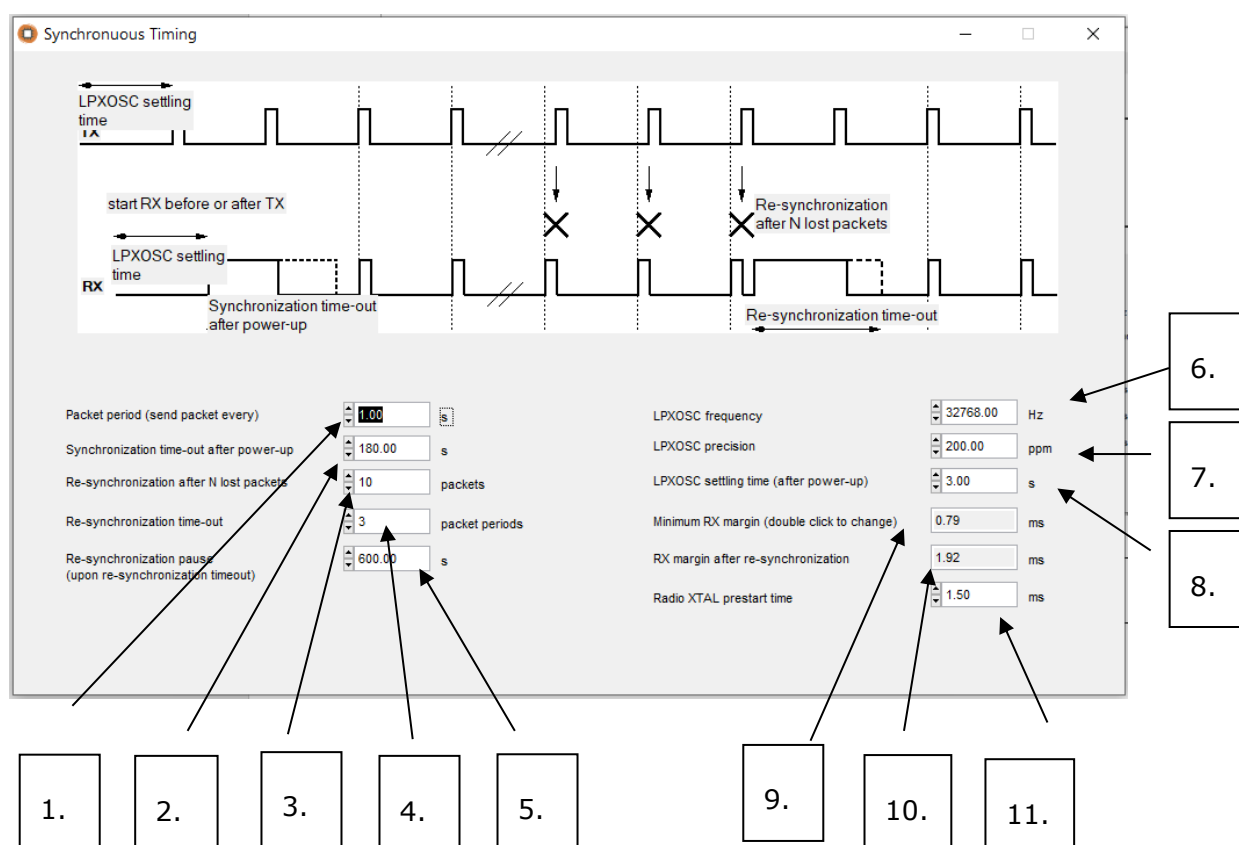
## 4.2. RX Synchronized with TX Mode

The purpose of **RX Synchronized with TX** mode is periodic reception of packets with minimal current consumption by enabling the RX only for the time slots in which packets are expected. Short preambles can be used on the TX side<sup>2</sup>. The following points explain how the **RX Synchronized with TX** mode works:

- **RX Synchronized with TX** mode is used together with **TX periodic (LPXOSC)** on the TX side. Timing is driven by the MCU's low power 32kHz tuning fork crystal oscillator which ensures an accurate packet frequency and thus allows for tight margins in the time slots in which the RX is enabled.
- After power up LPXOSC is settled and the RX is switched continuously on until a packet is received, or a timeout occurs. LED1 indicates the RX is running. When a packet is received the RX is put to sleep and only wakes up again for the time slot in which the next packet is expected.
- With each further packet reception, the RX measures the effective time elapsed since the last packet and (via low pass filtering) corrects its own wake-up frequency accordingly. This corrects for crystal variations as well as temperature differences and slow gradients and minimizes required margins in the RX-on time slots. (An increased margin is used for the packet reception right after synchronization, where the effective period has not been measured yet.)
- If no packet is received during an RX-on time slot, the RX margins are increased for the next time slot. If no packet is received during N subsequent time slots, the RX module switches back to synchronization mode, i.e. the RX is switched on continuously until a packet is received or a timeout occurs. (This synchronization mode is the same as for initial synchronization after power up. The only difference lies in the timeout interval: A timeout (in seconds) can be configured for initial synchronization after power-up (e.g. time to put batteries into the TX module). A second timeout (in packet periods) applies for re-synchronization after losing multiple packets in a row.
- If (re-)synchronization fails, (no packet is received until the timeout), the RX module is put to sleep for a configurable, typically long, period before trying to synchronize again. The rationale here is to limit RX power consumption in the case where the TX is off or out of reach.
- The **Sync Timing** button opens the Sync Timing panel, where various timing parameters can be configured.

---

<sup>2</sup> This contrasts with the asynchronous nature of **Wake on Radio** mode, where a preamble is at least as long as the RX sleep interval is required.



**Figure 2: RX Synchronized with TX, Sync Timing panel for AXM0F343**

- 1. Packet period (send packet every):** This is simply a duplicate of the **TX periodic (every)** field in the main panel for convenience.
- 2. Synchronization time-out after power-up:** Time the RX tries to receive the first packet, before temporarily giving up and switching to re-sync pause mode. E.g. this is the time you have to put batteries into the TX module after powering up the RX.
- 3. Re-Synchronization after N lost packets:** Determines in how many subsequent RX-on time slots packet reception can fail before the RX switches to back to synchronization mode.
- 4. Re-synchronization time-out:** Time the RX tries to receive a packet for (re-) synchronization before temporarily giving up and switching to re-sync pause mode.
- 5. Re-synchronization pause (upon re-synchronization time-out):** Time for which the RX sleeps after failing to re-synchronize before trying again.
- 6. LPXOSC frequency:** Frequency of the low power tuning fork crystal oscillator.
- 7. LPXOSC precision:** Precision of the low power tuning fork crystal oscillator.
- 8. LPXOSC settling time (after power-up):** TX as well as RX modules wait for this time before transmitting / trying to receive packets. This is to prevent synchronization and period measurement with an insufficiently settled clock source.



9. **Minimum RX margin:** RX margin determines how long the RX is running before the expected packet start. The same margin is also used as a timeout: The RX is powered down if no frame delimiter is detected by the time it was expected plus RX margin. (RX running means synthesizer running and analog baseband settled, but no AGC, AFC and bit synchronization, since there is no signal yet.) The minimum value specified here must accommodate timing fluctuations in normal operation. A default value is computed from **packet period** and **LPXOSC precision**. Double click the field to change it manually. If packet reception fails in a given RX-on time slot, the firmware automatically increases the margin for the next time slot.
10. **RX margin after re-synchronization:** An increased RX margin which is used for the RX-on time slot right after (re-) synchronization. This is necessary, since (re-) synchronization only aligns the phases of periodic TX and RX. The RX needs a further packet to measure and adjust the wake-up period. Therefore, the margin of the first RX-on time slot has to be large enough for the full, uncorrected timing deviation.
11. **Radio XTAL prestart time:** Determines how long the XTAL reference oscillator of the radio chip is settled before the RX synthesizer is started.

## 5. Kit Configuration Panel

In the Kit Configuration panel displays the devices/compiler/debugger and the kit type as chosen by the user while creating the radio configurator project.

Further, it allows the user to enable/disable debug prints, sleep options etc

The displayed device info is non-configurable and shows based on the user selection as below.

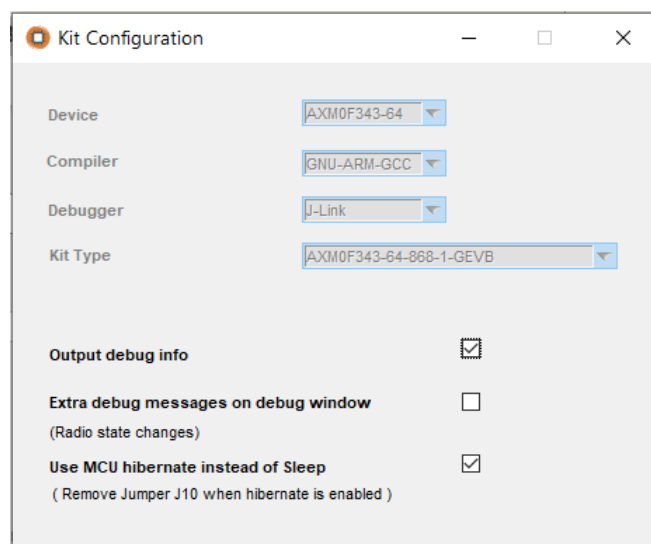
### Devices/MCUs:

- AXM0F343-64
- AXM0F343-256

### Kit types based on the selected device:

- For AXM0F343-64
  - AXM0F343-64-868-1-GEVB
  - AXM0F343-64-915-1-GEVB
- For AXM0F343-256
  - AXM0F343-256-868-1-GEVB
  - AXM0F343-256-915-1-GEVB

The kit configuration panel consists of device, compiler, debugger, and kit type. This panel displays the devices, kit types, compilers and debuggers as selected by the User for reference. Enable/Disable debug print, Sleep options etc could be selected/deselected by the user.



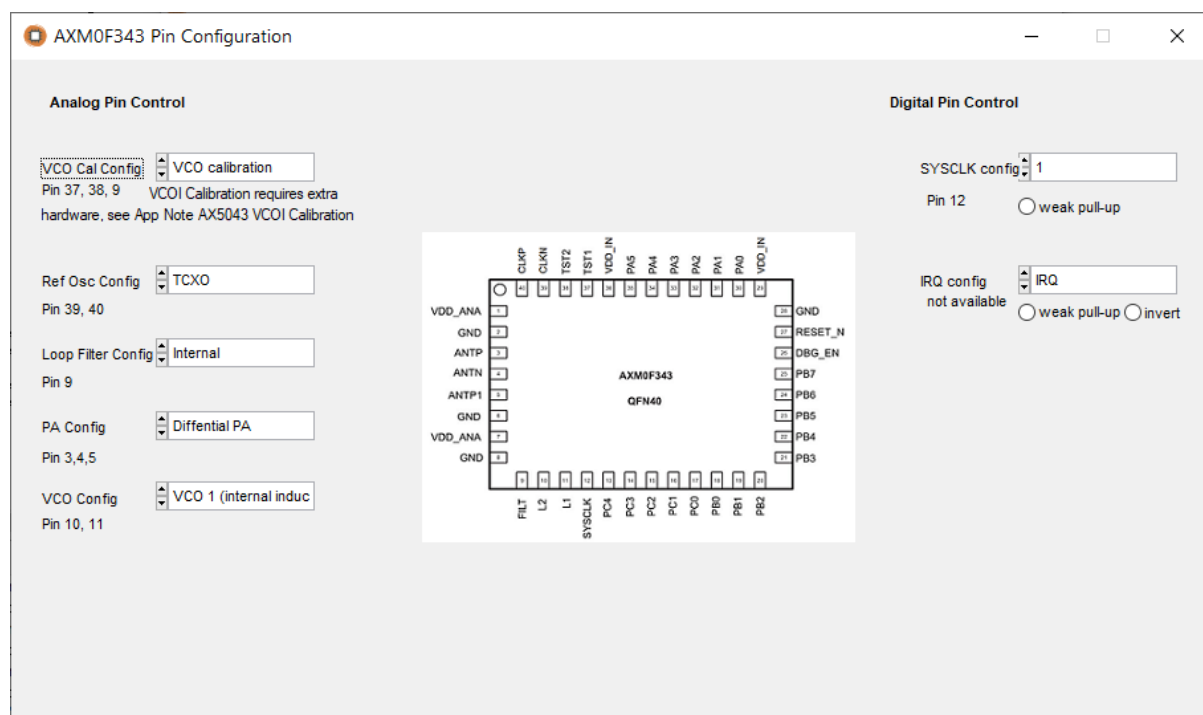
**Figure 3: Kit Configuration panel for AXM0F343**

**Output Debug Info:** For AXM0F343, the debug messages are transmitted to a serial terminal via UART0. This option enables to print debug information. The terminal settings should be configured to set baud rate = 115200, stop = 1, parity none, data = 8 bits.

**Extra debug messages on debug window:** This option produces extra information about the status change of the radio state.

**Use MCU hibernate instead of sleep** option decides if the MCU uses hibernate mode instead sleep mode. When hibernate mode is enabled in the kit configuration panel, after programming the device with the generated firmware, jumper J10 should be removed and hard reset should be done for the hibernate mode to work.

## 6. Pin Configuration Panel



**Figure 4: Pin Configuration panel for AXM0F343**

- VCO Cal Config: VCO calibration is enabled by default
- Ref Osc Config: select XTAL or TCXO
- PA Config: select differential, single ended or both power amplifier(s)
- Loop Filter Config: determines whether the internal or an external PLL loop filter is used.
- VCO Config: Select VCO 1 (fully integrated VCO with on-chip inductor), VCO 2 (using an external inductor), or an entirely external VCO
- SYS CLK config: default selected as 1
- IRQ Config: Output IRQ. No other setting is sensible if the AX5043 module is connected to the main board.

## 7. PHY Panel

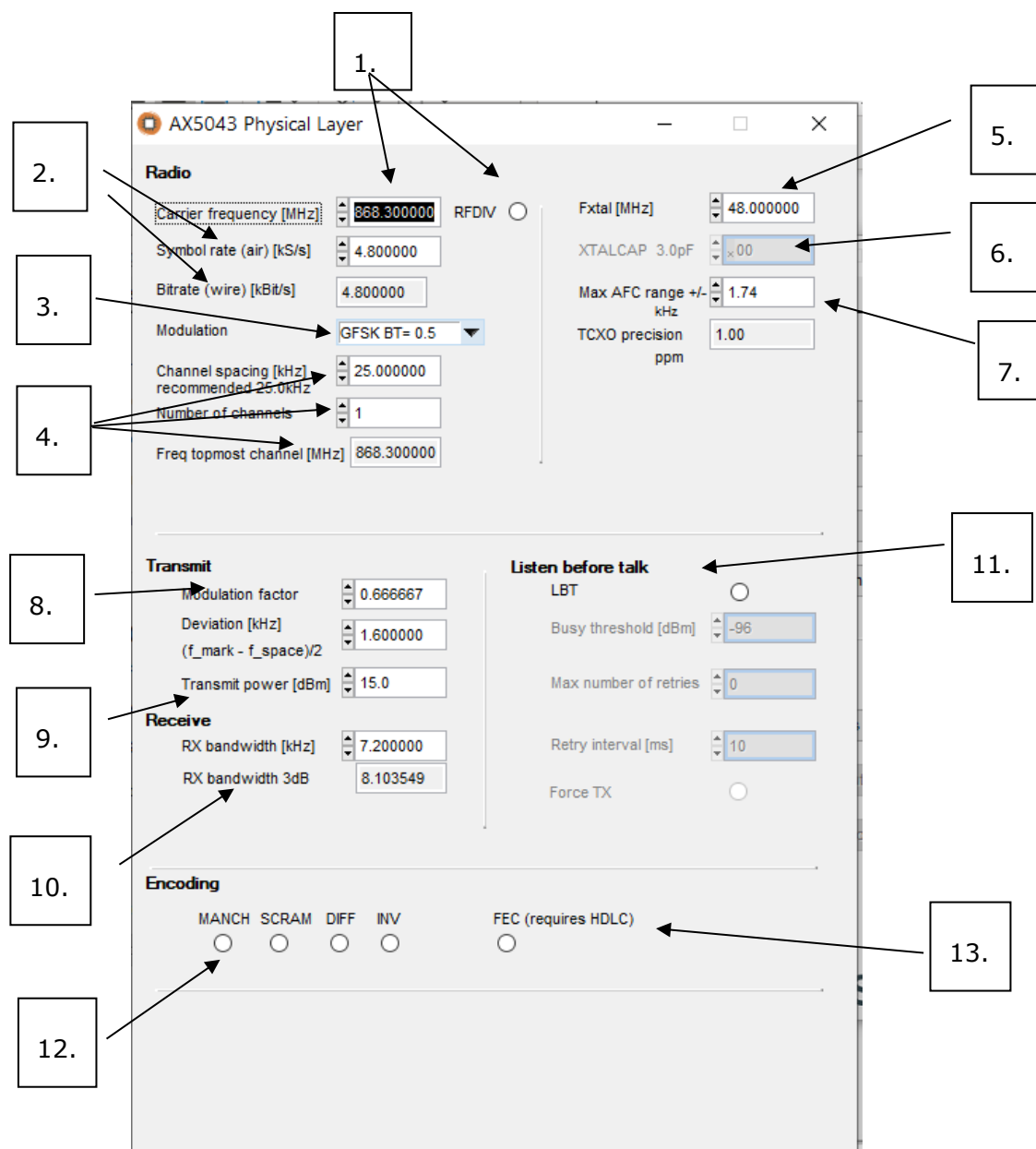


Figure 5: Physical layer for AXM0F343

1. Select Carrier Frequency. If more than one channel is configured (see below) the frequency selected here is that of channel number 0. RFDIV indicates that the VCO is running at 4 times the LO frequency, rather than 2 times (depending on the frequency band). This is an indicator only unless you are using an external inductor and the selected frequency can be reached in both ways (using a different inductor value, of course).
2. Symbol rate & Bitrate specify the data rate. The values are identical except when using Manchester encoding (bitrate = 0.5\*symbol rate) or 4-FSK (bitrate = 2\*symbol rate).

3. Modulation Scheme.
4. Width of your channel. Red values signal a conflict with the TX and / or RX bandwidth. Several channels can be specified. The firmware determines and stores the VCO range for each channel at startup. The carrier frequency of the topmost channel is displayed for convenience.
5. Frequency of the reference oscillator (XTAL / TCXO).
6. XTALCAP adjusts the on-chip XTAL load capacitance. Set to 0 when using a TCXO.
7. Max AFC range +/-: This value determines how far the frequency tracker (AFC) can walk in either direction. Setting it to zero disables frequency tracking. The necessary TCXO (or XTAL) precision is displayed. The computation is  $\text{TCXO precision} = (\text{max AFC range}) / (2 * f_{\text{carrier}})$ , ensuring that the tracking range is sufficient to handle the case, where the TCXO of the TX and RX are maximally off in opposite directions.

For more details and recommendations on setting up the AFC range and FSK deviation see chapter 12: XTAL/TCXO Precision, Frequency Tracking and Bandwidth.

Note: Increasing the range of the AFC allows for the use of a less precise TCXO or XTAL. However, the drawbacks are

- a. Longer preambles may be necessary to settle the AFC. In WOR and RX Sync mode this enlarges the necessary receive windows and thus the duty cycled receiver current consumption.
  - b. Larger RXBW may be necessary, since for successful tracking a sufficient amount of TX spectrum needs to fall into the RXBW. A warning is displayed if this is not guaranteed. See dedicated paragraph and figure below for details.
8. Modulation factor (often denoted as h).  
FSK deviation =  $0.5 * \text{modulation factor} * \text{Symbol rate} = (f_{\text{mark}} - f_{\text{space}}) / 2$ , sometimes called half deviation.  
4-FSK: Symbols are  $f_{\text{carrier}} + \{-3, -1, +1, +3\} * \text{deviation}$
  9. Transmit power in dBm.
  10. Bandwidth of the RX channel filter, the upper field is used to input the desired value, the lower field shows the actual value.
  11. Listen Before Talk (LBT) option: The TX measures the channel RSSI before sending a packet in order to assess whether the channel is free. If the RSSI value is found to be above busy threshold the TX can retry a number of times. If the channel continues to be busy the TX can either give up or transmit anyway (force TX).
  12. Encoding: manchester, scrambler, differential and inversion.

13. Forward Error Correction (FEC). This encoding transparently adds redundancy to the transmit data to make it more immune against noise. FEC can only be used together with HDLC framing and it is mandatory to use HDLC flags (0x7E) as preamble characters. Other encoding modes such as SCRAM are not allowed and unnecessary in FEC mode, since FEC leads to data whitening by itself.

**Note:**

When PSK shaped modulation is selected, the DIFF encoding mode is enabled automatically. There is a pop-up message that indicates "DIFF" encoding mode is enabled. If the user modifies the modulation in PHY panel, DIFF encoding mode must be manually deselected if not required.

## 8. Framing Panel

AX5043 Framing

Framing Mode: Raw, Pattern Match

Total packet length [ms] 32

PREAMBLE	SYNC WORD	LEN & MAC		DATA	CRC
Length [Bits] <input type="text" value="32"/> Character <input type="text" value="AA"/> <input checked="" type="radio"/> unencoded WOR PA [ms] <input type="text" value="240"/> Append additional encoded bits Number of bits <input type="text" value="0"/> Pattern <input type="text" value="00"/>	Syncword length [Bits] <input type="text" value="32"/> Sync word <input type="text" value="33"/> <input type="text" value="55"/> <input type="text" value="33"/> <input type="text" value="55"/> The leftmost byte is sent first. <input type="radio"/> unencoded Quality: Transitions 23 DC 0 Max running DC -2	MAC header length <input type="text" value="3"/> <input checked="" type="radio"/> enable len field Position [Bits] <input type="text" value="0"/> Significant bits <input type="text" value="8"/> Len offset <input type="text" value="0"/> Max packet length <input type="text" value="200"/> (RX drops longer packets) <input type="radio"/> ACK uses sequence number position <input type="text" value="0"/>	<input checked="" type="radio"/> enable address matching Address position <input type="text" value="1"/> Address length [bytes] <input type="text" value="2"/> Master <input type="text" value="32"/> : <input type="text" value="34"/> : <input type="text" value="00"/> : <input type="text" value="00"/> Slave <input type="text" value="33"/> : <input type="text" value="34"/> : <input type="text" value="00"/> : <input type="text" value="00"/> Mask <input type="text" value="FF"/> : <input type="text" value="FF"/> : <input type="text" value="00"/> : <input type="text" value="00"/> <input type="radio"/> TX sender's address at position <input type="text" value="0"/>	00 00 55 66 77 88 For PER test: <input checked="" type="radio"/> insert 16 bit counter counter position <input type="text" value="0"/>	<input type="text" value="CRC-16"/> CRC INIT <input type="text" value="FFFFFFF"/>

☐ enable SFD callback
 
☐ send MSB first  
☐ CRC SKIP FIRST

Figure 6: Framing panel for AXM0F343

### Framing mode:

Options are HDLC, Raw Pattern Match and PN9 Compatibility. In Raw Pattern Match mode an arbitrary SYNC WORD of up to 32 bit marks the beginning of a frame. For legacy systems the PN9 Compatibility mode implements PN9 data whitening in software, based on the AX5043's Raw Pattern Match mode.

### Preamble:

- Recommended default settings are chosen according to the selected modulation, encoding and precision of the reference clock.
- Specify the length of the preamble in bits, as well as one byte character which will be repeated length/8 times.
- The preamble character is always sent MSB first, irrespective of the "send MSB first" option in the DATA chunk. See comment on that option below.
- If length is not an integer multiple of 8, the fractional byte is sent last, e.g. for length = 33 bits the preamble character is sent 4 times followed by once the MSB of the preamble character.
- The "unencoded" option causes the preamble character to bypass the Manchester, scrambler, differential and inversion encoder. (See footnote to SYNC WORD for technical details about inversion.)
- In Wake on Radio mode an additional WOR preamble length (in ms) can be specified. This chunk is transmitted in addition to the number of bits specified above. The length of this chunk should not be shorter than the WOR period.



- Occasionally it may be necessary to conclude the preamble with a few special bits. This can be achieved using the fields below "Append additional encoded bits".

### SYNC WORD

- In Raw Preamble Mode you can choose an arbitrary sync word of up to 32 bit as a start of frame delimiter. However, it is recommended to use a sync word with well peaking auto correlation function.
- The SYNC WORD is always sent MSB first, irrespective of the "send MSB first" option in the DATA chunk. See comment on that option below.
- The "unencoded" option causes the SYNC WORD to bypass the manchester, scrambler, differential and inversion encoder<sup>3</sup>. Correspondingly the RX matching unit detecting the SYNC WORD will operate on the bit stream before decoding.
- DC content, running DC content and the number of transitions is computed in order to indicate the quality of the selected SYNC WORD. Large DC or running DC components can impede correct frequency tracking, whereas a low number of transitions slows down time tracking.
- Selecting the "enable SFD callback" option causes the AXRadioAPI to notify the application level firmware each time a start of frame delimiter (a SYNC WORD or a HDLC flag) has been received.

### LEN & MAC

- "MAC header length" specifies the length of the packet header, which can contain a length field, a sequence number for acknowledge and addresses.
- In Raw Pattern Match mode, a length field can be used to communicate the length of the current packet to the RX. Alternatively (uncheck "enable len field") the RX can be configured to expect packets of a fixed length.
- "Position [Bits]" determines the position of the length field inside the packet header. Zero means the length field follows immediately after the SYNC word.
- "Significant bits" tells the RX how many bits of the length field are used to encode the packet length. This is useful if your protocol uses a shared byte to encode the packet length plus something else.
- "len offset": The actual packet length (excluding SYNC WORD & CRC) assumed by the RX is len + len offset. E.g. set len offset to zero if in your protocol the length byte counts the number of bytes including the length byte. Set len offset = 1 if

---

<sup>3</sup> Bypassing the encoder is achieved by setting the UNENC flag corresponding FIFO DATA chunk. This causes the chunk data to bypass manchester, scrambler and differential encoder, however inversion still applies. For convenience AX-ConfigurationUtility therefore inverts preamble and SYNC WORD bytes before writing them to the firmware code if "unencoded" and INV encoding are selected.

your length field counts the number of bytes excluding the length field itself. Negative values for "len offset" are allowed.

- If you check "enable len field" in HDLC mode, the TX will send a length field, but it will be ignored by the RX.
- Specifying a maximum packet length can limit the impact of bit errors in the length field.
- If the acknowledge feature is used an ACK sequence number field can be configured. This is not mandatory for acknowledge to work. However, without this field the SLAVE cannot detect duplicate packets occurring if packet RX has worked but the ACK packet got lost, causing the MASTER to retransmit the packet.
- Specify a device addresses of up to 4 bytes and a corresponding address mask to restrict reception to packets containing a matching address.
- "address position" specifies the starting position of the address inside the packet. Zero corresponds to the byte immediately after the SYNC WORD / HDLC flag.
- Optionally the sender's address can be added to the header. Note that address works on the destination address field (containing the slave's address in normal MASTER to SLAVE communication and the master's address for packets sent from SLAVE to MASTER, e.g. ACK packets in the demo firmware.)

### DATA

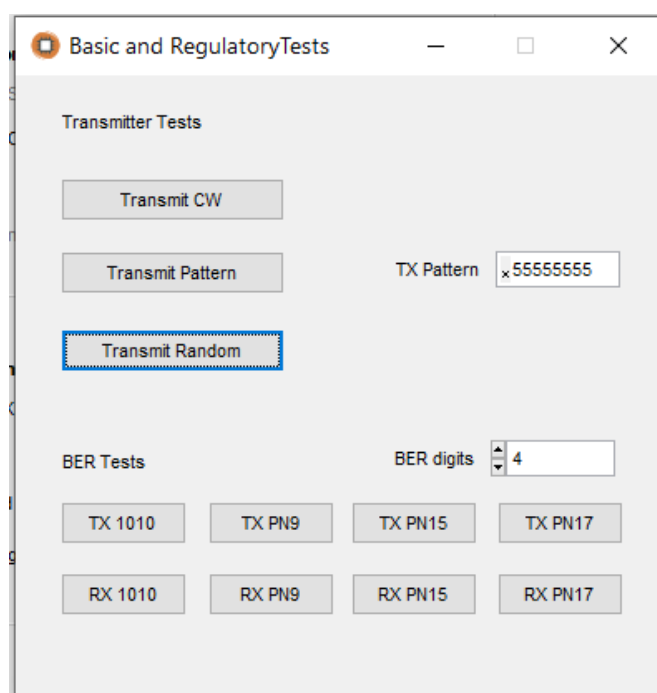
- The DATA field allows you to specify a static sample payload which will be used by the demo firmware.  
Enter each byte as a two-digit hex number (without leading 0x). Bytes may be separated by spaces.
- Optionally the firmware can insert a 16-bit counter into the data field at a specified position. This allows the RX to count the number of lost packets. The two bytes reserved for this counter are displayed as "00 00" in the data field.
- Checking the "send MSB first" button causes each byte in the DATA field to be sent MSB first. Otherwise each byte is sent LSB first.  
Note, however, that preamble character and SYNC WORD are always sent MSB first. (Technically, the RF chip will be configured to operate in MSB first or LSB first mode according to the "MSB first button", which applies to every byte. However, in LSB first mode the AX-ConfigurationUtility GUI reverses the entered preamble and SYNC WORD bytes to keep the representation in the GUI MSB first. The reason for this is, that the byte and nibble order of a SYNC WORD written left to right but read LSB first can be confusing.)
- "CRC SKIP FIRST" causes the first byte of the packet to be excluded from CRC calculation.

### CRC

- Select CRC algorithm and initial value.

- The CRC is automatically calculated by the RF chip and appended to the packet. In HDLC mode the RX will store the received packet including CRC bytes. In Raw Packet Match mode, the RX strips the CRC bytes unless you manually set bit 5 in the radio specific PKTSTOREFLAGS register. For example: AX5043\_PKTSTOREFLAGS.
- Packets with incorrect CRC are silently dropped unless you manually set bit 2 in the Radio specific PKTACCEPTFLAGS register. For example: AX5043\_PKTACCEPTFLAGS

## 9. Basic and Regulatory Tests Panel



**Figure 7: Basic and Regulatory Tests panel for AXM0F343**

- "Set CW" puts the RF chip to TX CW mode.
- "Set Pattern" transmits the specified 8-byte pattern forever. The byte specified by the two leftmost digits is sent first. Each byte is sent LSB first unless the "send MSB first" button in the framing panel is checked. Manchester, scrambler, and differential encoder are bypassed (inversion encoding applies if selected!).
- "Set Random" transmits pseudo-random data forever. (This is realized using the scrambler.)
- "RX 1010" performs measurement of bit error rate (BER). Typically connect the RX to a signal generator sending an infinite 1010 stream, i.e. in FSK mode the generator should toggle between  $\{+1, -1\}$  \* half deviation. In 4-FSK mode, the pattern  $\{-3, -1, +1, +3\}$  \* half deviation should be applied. The BER value can be measured with 3, 4 or 5 digit precision.
- "TX 1010" transmits an infinite 1010 stream suitable for "RX 1010" BER measurement.
- "RX PN9", "TX PN9", "RX PN15", "TX PN15", "RX PN17", "TX PN17" are similar to "RX 1010" and "TX 1010", except that the BER measurement is performed using a pseudo-random data stream. In PN9 mode, the data stream is generated using the polynomial  $x^9 + x^5 + 1$ , in PN15 mode, the polynomial  $x^{15} + x^{14} + 1$  is used, and in PN17 mode, the polynomial  $x^{17} + x^{12} + 1$  is used.

## **AX-ConfigurationUtility User Manual**

Note: When the any of the option in the panel is clicked, the configuration is saved in the firmware. Further, firmware can be downloaded using eclipse workspaces available in Tests folder for 64 and 256 version of the chip.

## 10. Programming the Target boards

The AXM0F343 target boards can be programmed using eclipse workspace along with Jlink debugger interface.

Eclipse workspaces is available for 64/256, Master/Slave and Tests. The workspaces are added for Application files, linked CMSIS based driver as library.

The user could configure various options using AX-ConfigurationUtility, save the configured options using « Save and Write Output » , further use the eclipse workspace to program the SoC.

## 11. Comments on C-code Firmware

### 11.1. File Structure

The MASTER and SLAVE and TESTS firmware projects implement the corresponding modes using the AXRADIO API as a “driver” for the AX5043 RF chip.

The AXRADIO API consists of the following files:

File	Description
COMMON/axradio.h	AXRadio API declaration
COMMON/easyax5043.h	AXRadio AX5043 private header
COMMON/easyax5043.c	AXRadio AX5043 main code
AX_Radio_Lab_output/config.c	Parameters generated by AX-ConfigurationUtility based on the selected device

It is documented in DOCU/AX-ConfigurationUtilityAPIManual.pdf

The application level code consists of the following files:

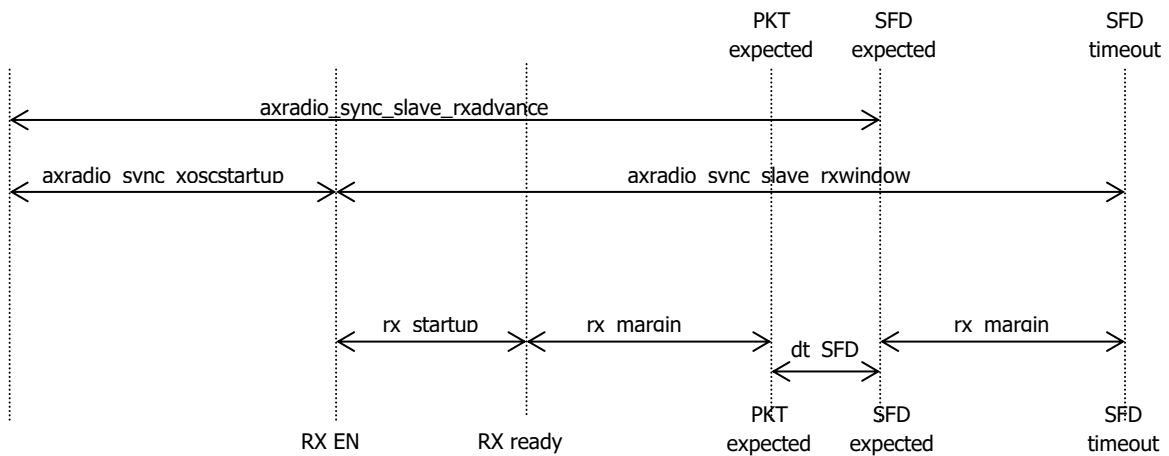
File	Description
MASTER/main.c	MASTER application level main code
SLAVE/main.c	SLAVE application level main code
TESTS/main.c	TESTS application level main code
COMMON/misc.[ch]	Application level helper files
AX_Radio_Lab_output/configslave.[ch]	Application level parameters generated by AX-ConfigurationUtility
AX_Radio_Lab_output/configmaster.[ch]	Application level parameters generated by AX-ConfigurationUtility

### 11.2. Define Statements for Tweaking the Firmware

**#define ADJUST\_FREQREG** (SLAVE/main.c) if defined, after successful packet reception the receiver frequency is adjusted according to the tracked frequency offset. The frequency is not adjusted by the entire tracked frequency offset at once. Rather low pass filtering is applied. Low pass filtering is controlled by **FREQOFFS\_K**, higher values give slower adjustments.

### 11.3. SYNC Packet RX Timing

In the AX-ConfigurationUtility SYNC RX mode timing is controlled by the parameters `axradio_sync_xoscstartup`, `axradio_sync_slave_rxadvance`, `axradio_sync_slave_rxwindow` and `axradio_sync_slave_rxtimeout`. All times are measured in periods of `wtimer0`, which is clocked by the low power crystal oscillator. As shown at the top of Figure 8 the radio XTAL oscillator of the SLAVE is enabled `axradio_sync_slave_rxadvance` before the frame delimiter (SFD) of the next packet is expected. After settling the XTAL for `axradio_sync_xoscstartup` the RX is enabled. The RX then settles and looks for an SFD, timing out after `axradio_sync_slave_rxwindow` if no SFD is received. If an SFD is received the RX stays on until successfully receiving a packet or until a timeout occurs after `axradio_sync_slave_rxtimeout`. (Not shown in the figure.)



**Figure 8: SYNC firmware packet reception timing.**

AX-ConfigurationUtility computes `axradio_sync_slave_rxadvance` and `axradio_sync_slave_rxwindow` based on the `rx_margin` which can be configured in the SYNC Timing panel as shown at the bottom of Figure 8. The goal is to have the RX ready `rx_margin` before the expected start of the packet. (Ready means that the synthesizer and the analog baseband are settled. AGC settling and bit synchronization can be done only if preamble signal is present.) The SFD timeout is set `rx_margin` after the expected SFD. Three instances of `axradio_sync_slave_rxadvance` and `axradio_sync_slave_rxwindow` are computed. The first instance contains the increased `rx_margin` to be used for the first packet after (re-)synchronization when the effective packet period measurement is available yet. The second instance contains the `rx_margin` used in normal operation ("Minimum RX margin" in the SYNC Timing panel). The third instance has `rx_margin` scaled by a factor of 4 and is used after a failed packet reception.

Note: A processing delay occurs in the RX, i.e. the detection of the SFD is not signalled by the RX the very moment it is on air. On the other the RX should be enabled `rx_margin` before the packet is on air. Therefore `dt_SFD` is not simply the length of preamble plus SFD but also contains the processing delay of roughly 19 bit.



## 12. XTAL/TCXO Precision, Frequency Tracking and Bandwidth

This section gives some recommendations and background information on how to setup the PHY panel.

### 12.1. How to set up AFC range, FSK deviation and RX bandwidth

#### Case 1: XTAL/TCXO precision, bitrate and carrier frequency are given

If XTAL/TCXO precision, bitrate and carrier frequency are given, how should the FSK deviation and the RX Bandwidth be set to achieve maximum sensitivity?

##### **Step 1**

Set carrier frequency and bitrate

Set Max AFC range so that XTAL/TCXO precision matches the precision of the crystal or TCXO being used. Note that changes over temperature of the XTAL/TCXO should be included.

**AX5043 Physical Layer**

**Radio**

Carrier frequency [MHz] 868.300000 RFDIV ☐ Fxtal [MHz] 48.000000

Symbol rate (air) [kS/s] 4.800000 XTAL C&P 48.000000

Bitrate (wire) [kBit/s] 4.800000 Max AFC range +/- 17.37 kHz

Modulation GFSK BT= 0.5 TCXO precision 10.00 ppm

Channel spacing [kHz] recommended 25.0kHz 25.000000

Number of channels 1

Freq topmost channel [MHz] 868.300000

0.5\*(RXBW + TXBW) < 2\* max AFC range  
No initial overlap of RXBW and TXBW is guaranteed.  
Increase BW or use a more precise XTAL / TCXO.

**Transmit**

Modulation factor 0.666667

Deviation [kHz] 1.600000

(f<sub>mark</sub> - f<sub>space</sub>)/2

Transmit power [dBm] 15.0

**Receive**

RX bandwidth [kHz] 7.200000

RX bandwidth 3dB 8.103549

**Listen before talk**

LBT ☐

Busy threshold [dBm] -96

Max number of retries 0

Retry interval [ms] 10

Force TX ☐

**Encoding**

MANCH ☐ SCRAM ☐ DIFF ☐ INV ☐ FEC (requires HDLC) ☐

Figure 9: 868.3 MHz, 10 kbps FSK, 10 ppm XTAL precision example for step 1 of PHY setup

## Step 2

Set RX Bandwidth to 2 • Max AFC range

**AX5043 Physical Layer**

**Radio**

Carrier frequency [MHz] 868.300000 RFDIV ☐

Symbol rate (air) [kS/s] 4.800000

Bitrate (wire) [kBit/s] 4.800000

Modulation GFSK BT= 0.5

Channel spacing [kHz] recommended 25.0kHz 25.000000

Number of channels 1

Freq topmost channel [MHz] 868.300000

Fxtal [MHz] 48.000000

XTALCAP 3.0pF x00

Max AFC range +/- kHz 17.37

TCXO precision ppm 10.00

0.5\*(RXBW + TXBW) < 2\* max AFC range  
No initial overlap of RXBW and TXBW is guaranteed.  
Increase BW or use a more precise XTAL / TCXO.

**Transmit**

Modulation factor 0.666667

Deviation [kHz] 1.600000  
(f<sub>mark</sub> - f<sub>space</sub>)/2

Transmit power [dBm] 15.0

**Listen before talk**

LBT ☐

Busy threshold [dBm] -96

Max number of retries 0

**Receive**

RX bandwidth [kHz] 7.200000

RX bandwidth 3dB 8.103549

Retry interval [ms] 10

Force TX ☐

**Encoding**

MANCH ☐ SCRAM ☐ DIFF ☐ INV ☐ FEC (requires HDLC) ☐

Figure 10: 868.3 MHz, 10 kbps FSK, 10 ppm XTAL precision example for step 2 of PHY setup

### Step 3

Increase the TX modulation factor or the deviation until the field XTAL precision is no longer red. This is typically at a value  $Modulation\_Factor = \frac{2 \bullet Max\_afc\_range}{Bitrate} - 1$ . If a modulation factor value larger than 5 emerges, then it is strongly recommended to increase the bitrate and to redo the setup procedure.

**AX5043 Physical Layer**

**Radio**

Carrier frequency [MHz]  RFDIV ☐

Symbol rate (air) [kS/s]

Bitrate (wire) [kBit/s]

Modulation

Channel spacing [kHz] recommended 25.0kHz

Number of channels

Freq topmost channel [MHz]

Fxtal [MHz]

XTALCAP 3.0pF

Max AFC range +/- kHz

TCXO precision ppm

$0.5 \cdot (RXBW + TXBW) < 2 \cdot \text{max AFC range}$   
 No initial overlap of RXBW and TXBW is guaranteed.  
 Increase BW or use a more precise XTAL / TCXO.

**Transmit**

Modulation factor

Deviation [kHz]

$(f_{\text{mark}} - f_{\text{space}}) / 2$

Transmit power [dBm]

**Listen before talk**

LBT ☐

Busy threshold [dBm]

Max number of retries

Retry interval [ms]

Force TX ☐

**Receive**

RX bandwidth [kHz]

RX bandwidth 3dB

**Encoding**

MANCH ☐ SCRAM ☐ DIFF ☐ INV ☐ FEC (requires HDLC) ☐

Figure 11: 868.3 MHz, 10 kbps FSK, 10 ppm XTAL precision example for step 3 of PHY setup

## Case 2: Bitrate and carrier frequency and FSK deviation are given

If the bitrate, carrier frequency and FSK deviation are given, how should the RX Bandwidth be set to achieve maximum sensitivity and what is the XTAL/TCXO precision that is required for this to work?

### Step 1

Set carrier frequency, bitrate and FSK deviation or modulation factor.

**AX5043 Physical Layer**

**Radio**

Carrier frequency [MHz] 868.300000 RFDIV ☐ Fxtal [MHz] 48.000000

Symbol rate (air) [kS/s] 4.800000 XTALCAP 3.0pF

Bitrate (wire) [kBit/s] 4.800000 Max AFC range +/- kHz 17.37

Modulation GFSK BT= 0.5 TCXO precision 10.00 ppm

Channel spacing [kHz] recommended 25.0kHz 25.000000

Number of channels 1

Freq topmost channel [MHz] 868.300000

0.5\*(RXBW + TXBW) < 2\* max AFC range  
No initial overlap of RXBW and TXBW is guaranteed.  
Increase BW or use a more precise XTAL / TCXO.

**Transmit**

Modulation factor 0.666667

Deviation [kHz] 1.600000  
(f<sub>mark</sub> - f<sub>space</sub>)/2

Transmit power [dBm] 15.0

**Listen before talk**

LBT ☐

Busy threshold [dBm] -96

Max number of retries 0

Retry interval [ms] 10

Force TX ☐

**Receive**

RX bandwidth [kHz] 7.200000

RX bandwidth 3dB 8.103549

**Encoding**

MANCH ☐ SCRAM ☐ DIFF ☐ INV ☐ FEC (requires HDLC) ☐

Figure 12: 868.3 MHz, 10 kbps FSK, modulation factor 5 example for step 1 of PHY setup

## Step 2

Set the RX bandwidth to  $(1 + \text{Modulation\_Factor}) \bullet \text{Bitrate}$ .

If necessary, increase the XTAL/TCXO precision by decreasing the Max AFC range until the red warning (No initial overlap of RX BW and TX BW is guaranteed.) disappears. This will occur at a Max AFC range value of RX BW/2.

If your XTAL/TCXO is not precise enough then increase the RX BW but note that this will come at the cost of lower sensitivity.

**AX5043 Physical Layer**

**Radio**

Carrier frequency [MHz] 868.300000 RFDIV ☐ Fxtal [MHz] 48.000000

Symbol rate (air) [kS/s] 10.000000 XTALCAP 3.0pF x.00

Bitrate (wire) [kBit/s] 10.000000 Max AFC range +/- kHz 17.37

Modulation GFSK BT= 0.5 TCXO precision ppm 10.00

Channel spacing [kHz] recommended 50.0kHz 50.000000

Number of channels 1

Freq topmost channel [MHz] 868.300000

0.5\*(RXBW + TXBW) < 2\* max AFC range  
No initial overlap of RXBW and TXBW is guaranteed.  
Increase BW or use a more precise XTAL / TCXO.

**Transmit**

Modulation factor 0.666667

Deviation [kHz] 3.333333

(f<sub>mark</sub> - f<sub>space</sub>)/2

Transmit power [dBm] 15.0

**Receive**

RX bandwidth [kHz] 15.000000

RX bandwidth 3dB 16.612275

**Listen before talk**

LBT ☐

Busy threshold [dBm] -93

Max number of retries 0

Retry interval [ms] 10

Force TX ☐

**Encoding**

MANCH ☐ SCRAM ☐ DIFF ☐ INV ☐ FEC (requires HDLC) ☐

(c) ON Semiconductor  
distribution prohibited

Figure 13: 868.3 MHz, 10 kbps FSK, modulation factor 5 example for step 2 of PHY setup

## 12.2. Background information

Without frequency tracking the worst case frequency deviation between RX and TX is

$$\Delta f = f_{carrier} \bullet (ppm_{TX} + ppm_{RX}),$$

Where  $ppm_{TX}$  and  $ppm_{RX}$  are the precisions of the crystals used in the transmitter and the receiver. The frequency tracker must be allowed to track this deviation, thus the "MAX AFC range" field in the AX-ConfigurationUtility PHY panel (7) should be set to this value,

$$\text{Max AFC Range:} = \Delta f = f_{carrier} \bullet (ppm_{TX} + ppm_{RX}).$$

Note: AX-ConfigurationUtility performs the inverse calculation. Given Max AFC Range, the necessary crystal precision is displayed. AX-ConfigurationUtility assumes identical crystals are used in RX and TX, thus the displayed value is XTAL precision =  $(ppm_{TX} + ppm_{RX})/2$ .

The above setting allows the tracker to walk away  $\Delta f$  in the wrong direction while there is no signal (see Figure 14). In the worst case the receiver frequency can therefore be  $f_{RX}$  when packet transmission starts, that is  $2 \bullet \Delta f$  away from  $f_{TX}$ . For successful tracking a fraction of the TX spectrum should fall into the RX BW. Thus the yellow shaded region should be non-empty:  $(TX\ BW + RX\ BW)/2 \geq 2 \bullet \Delta f$ . AX-ConfigurationUtility issues a warning if this constraint is not fulfilled. Inserting

$$RX\ BW = TX\ BW = (1 + h) \bullet BR = BR + 2 \bullet deviation$$

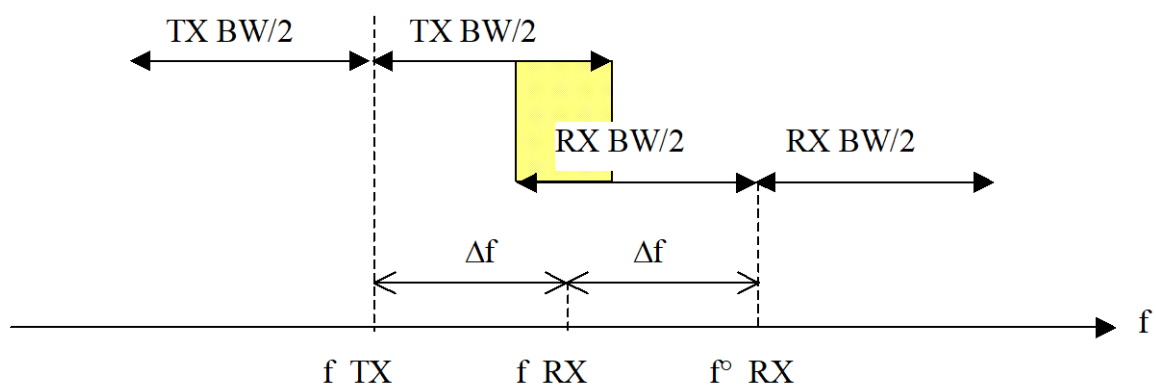
into the overlap condition leads to

$$h_{min} = \frac{2 \bullet \Delta f}{BR} - 1.$$

Irrespective of the overlap condition  $h \geq 0.5$  is required.

For optimal sensitivity  $RX\ BW_{opt} = (1 + h) \bullet BR$  should be chosen. If values  $h > 5$  emerges, increasing the bitrate should be considered as the RX BW for large h values has to be increased beyond  $RX\ BW_{opt}$  to achieve stable operating conditions.





**Figure 14: Frequency precision, tracking and bandwidth**

Note: By default, AX-ConfigurationUtility generated firmware adjusts the programmed receiver frequency upon successful packet reception. This feature can be disabled by commenting out the `#define ADJUST_FREQREG` statement in `SLAVE/main.c`.

### 13. Revision History

Version 1.0	First document release for AXM0F343.
Version 1.1	Main panel changes updated, added AXM0F343_CMSIS_PACK_ROOT details. Updated image for BER panel.
Version 1.2	Typos corrections.
Version 1.3	Matching the F343 pack
Version 1.4	onsemi logo is updated and the images in the document is upgraded to latest

ON Semiconductor and the ON Semiconductor logo are trademarks of Semiconductor Components Industries, LLC dba ON Semiconductor or its subsidiaries in the United States and/or other countries. ON Semiconductor owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of ON Semiconductor's product/patent coverage may be accessed at [www.onsemi.com/site/pdf/Patent-Marking.pdf](http://www.onsemi.com/site/pdf/Patent-Marking.pdf). ON Semiconductor is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

The evaluation board/kit (research and development board/kit) (hereinafter the "board") is not a finished product and is as such not available for sale to consumers. The board is only intended for research, development, demonstration and evaluation purposes and should as such only be used in laboratory/development areas by persons with an engineering/technical training and familiar with the risks associated with handling electrical/mechanical components, systems and subsystems. This person assumes full responsibility/liability for proper and safe handling. Any other use, resale or redistribution for any other purpose is strictly prohibited.

The board is delivered "AS IS" and without warranty of any kind including, but not limited to, that the board is production-worthy, that the functions contained in the board will meet your requirements, or that the operation of the board will be uninterrupted or error free. ON Semiconductor expressly disclaims all warranties, express, implied or otherwise, including without limitation, warranties of fitness for a particular purpose and non-infringement of intellectual property rights.

ON Semiconductor reserves the right to make changes without further notice to any board.

You are responsible for determining whether the board will be suitable for your intended use or application or will achieve your intended results. Prior to using or distributing any systems that have been evaluated, designed or tested using the board, you agree to test and validate your design to confirm the functionality for your application. Any technical, applications or design information or advice, quality characterization, reliability data or other services provided by ON Semiconductor shall not constitute any representation or warranty by ON Semiconductor, and no additional obligations or liabilities shall arise from ON Semiconductor having provided such information or services.

The boards are not designed, intended, or authorized for use in life support systems, or any FDA Class 3 medical devices or medical devices with a similar or equivalent classification in a foreign jurisdiction, or any devices intended for implantation in the human body. Should you purchase or use the board for any such unintended or unauthorized application, you shall indemnify and hold ON Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that ON Semiconductor was negligent regarding the design or manufacture of the board.

This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and may not meet the technical requirements of these or other related directives.

**FCC WARNING** – This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by ON Semiconductor to be a finished end product fit for general consumer use. It may generate, use, or radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment may cause interference with radio communications, in which case the user shall be responsible, at its expense, to take whatever measures may be required to correct this interference.

ON Semiconductor does not convey any license under its patent rights nor the rights of others.

**LIMITATIONS OF LIABILITY:** ON Semiconductor shall not be liable for any special, consequential, incidental, indirect or punitive damages, including, but not limited to the costs of requalification, delay, loss of profits or goodwill, arising out of or in connection with the board, even if ON Semiconductor is advised of the possibility of such damages. In no event shall ON Semiconductor's aggregate liability from any obligation arising out of or in connection with the board, under any theory of liability, exceed the purchase price paid for the board, if any.

#### PUBLICATION ORDERING INFORMATION

## AX-ConfigurationUtility User Manual

### 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](mailto: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

### ON Semiconductor Website:

<https://www.onsemi.com/>

### Order Literature:

<https://www.onsemi.com/orderlit>

For additional information, please contact your local Sales Representative