July 2008

TDA5230 / TDA5231

Universal Low Power ASK/FSK
Single Conversion Multi-Channel
Image-Reject Receiver with
Digital Baseband Processing

Application Note

Quick Start Guide
Version 1.1

Wireless Control

Never stop thinking

Edition 2008-07-15

Published by
Infineon Technologies AG
81726 Munich, Germany

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

SmartLEWIS

Revision History: 2008-07-02, Version 1.1

Previous Version: 1.0

Page	Subjects (major changes since last revision)
different	Typos corrected

Trademarks of Infineon Technologies AG

SmartLEWIS™

Other Trademarks

Microsoft®, Windows® of Microsoft Corporation.

The information in this document is subject to change without notice.

We Listen to Your Comments
Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:
wirelesscontrol@infineon.com

Table of Contents

1 Introduction

This Quick Start Guide shall help you to use the TDA523x Evaluation Boards, the support SW, TDA523x IAF Configuration Tool and the TDA523x Explorer, and guide you through your first application.

Following topics will be handled:

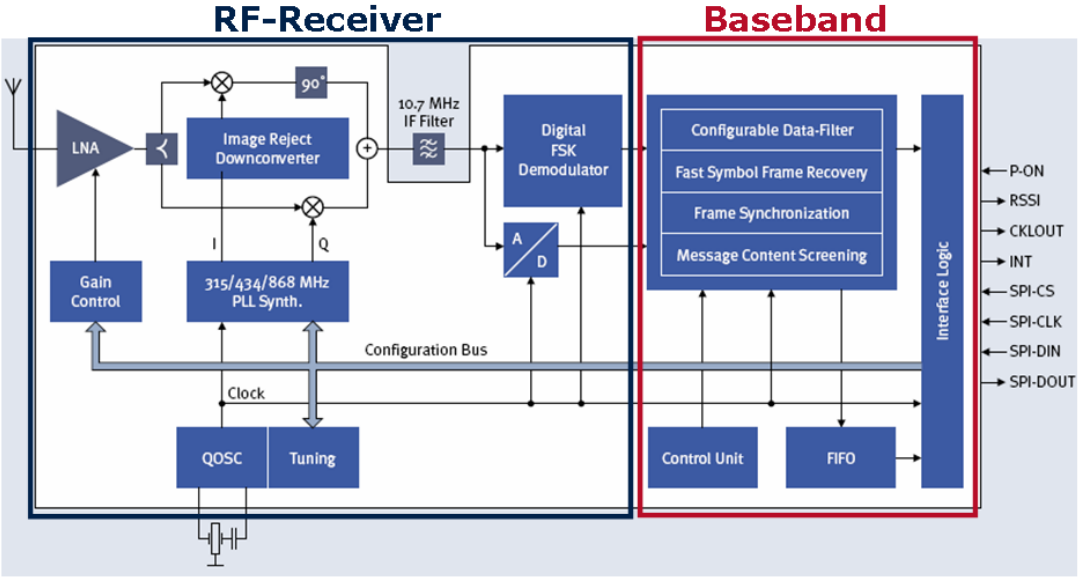
- A short overview about TDA523x
- Typical TDA523x applications
- Structure of data frames
- How sensitivity is defined

- Installation of support software
- Create your own configuration using the IAF Configuration Tool
- Use TDA523x Explorer for threshold settings in configuration
- Debugging hints
- Software implementation hints

2 A Short Overview about TDA523x

TDA523x is a family of universal, highly sensitive, low power, ASK/FSK RF multi-channel receivers for Manchester-coded data signals.

- TDA5230 is used for the ranges of 433 to 450MHz and 865 to 870MHz.
- TDA5231 is used for the range of 302 to 320MHz.



The left part in the block diagram of TDA523x shows a typical RF receiver, containing an LNA, PLL Synthesizer, Image Reject Mixer and Demodulators. The right part shows the difference to legacy products, the Baseband.

While typical traditional receivers require polling and bit and data frame synchronization done in a host µController, the TDA523x offers integrated Baseband processing, a comfortable and autonomous Self Polling Mode, digital data filter, bit clock recovery, and frame synchronization. While sleeping, the host µController is alerted by an interrupt, and afterwards able to download the received data from the integrated data FIFO via SPI interface.

This feature saves valuable SW implementation effort, offloads the µController, and reduces system power consumption significantly.

The Dual Configuration Register Sets allows receiving up to two different types of data frames, where these may differ in channel, modulation, data rate, wake up criterion, synchronization pattern, packet length, and so on.

The Multi-Channel PLL allows reception from 17 different sub-channels. Per register configuration set, definition of up to three different channels is possible.

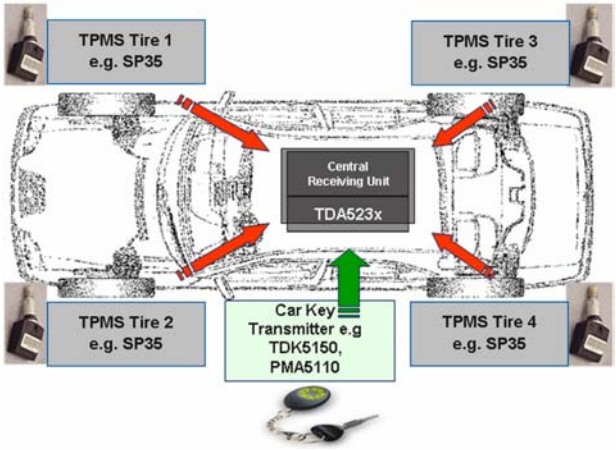
In autonomous receiving mode, the automatic reception according to the two configuration settings and the three channels per configuration, without interaction of the host µController, is supported.

3 Typical TDA523x Applications

TDA523x is ideal for a wide range of applications. Especially in the following cases it should be considered:

- Quick time to market
- Adaptable platform design
- Low power consumption
- Too little processing power of host µController
- Increase of real time capabilities of host µController
- Multi-Channel
- Reception from different transmit sources
- Dual bandwidth applications

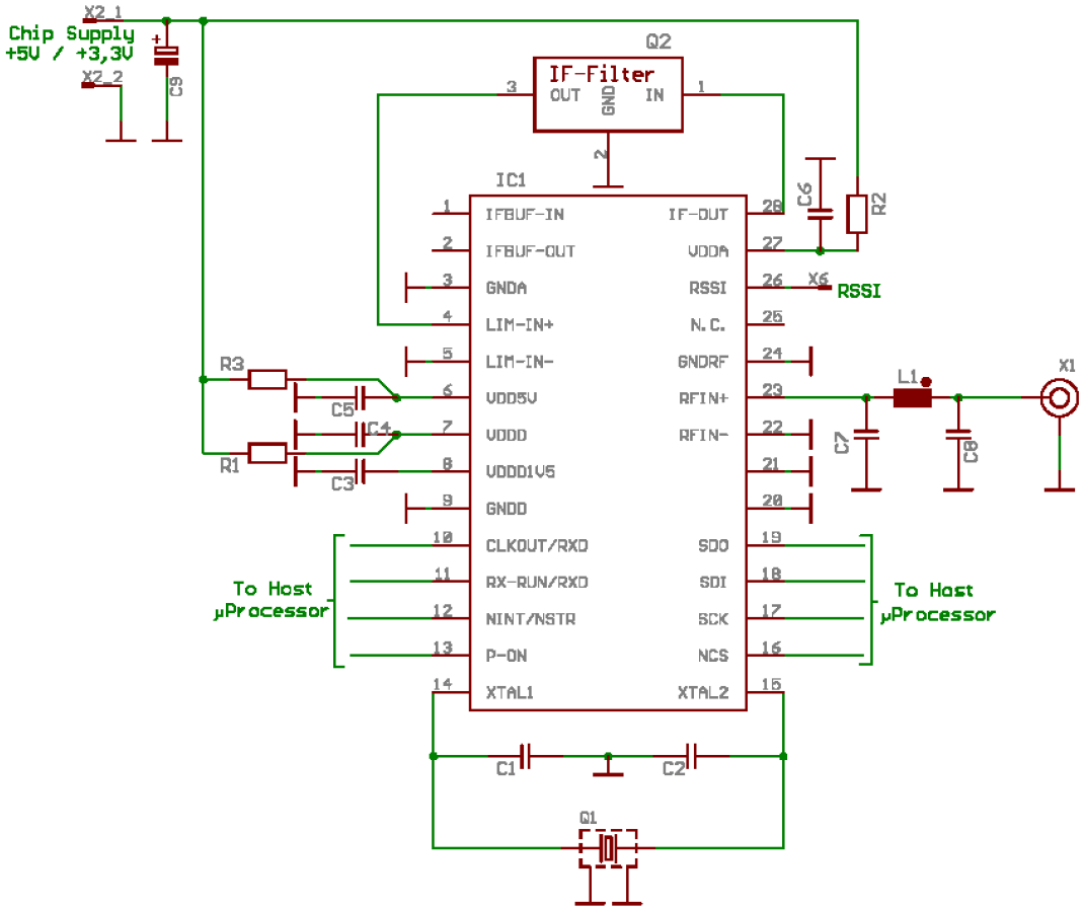
A typical application is the automotive receiver for RKE (Remote Keyless Entry, RF car key) and TPMS (Tire Pressure Monitor System). RKE and TPMS normally use completely different data frames. The power consumption, when the car is parked, has to be very low.



Another typical application for TDA523x is Automatic Meter Reading. Similarly in this case, there is data reception from different sources with different protocols.

TDA523x requires only few external components.

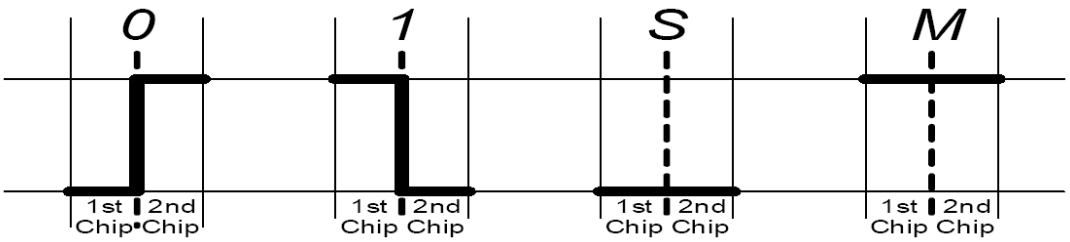
.



Crystal, IF Filter (two IF filters possible to allow bandwidth switching), RF input matching, some capacitors and resistors. In sensitive applications usage of a SAW filter between antenna and TDA523x RF inputs is recommended.

4 Structure of Data Frames

The Baseband of TDA523x is able to work with and encode Manchester-coded bits.



Manchester-coded bits have two so-called chips. Valid bits have an edge in the middle, first and second chip are different. If both chips are equal, this is called a Code Violation (CV), a Space (S), or a Mark (M). CVs are often used in TSIs to mark the end of messages.

Usually data frames have following parts:

- Wake Up is used to wake up the receiver from (self) polling mode. If the receiver is continuously active, the Wake Up is not required. The longer the Wake Up, the longer the receiver can be inactive in self polling mode, and the lower is the power consumption. After the wake up there may be an optional space before Preamble and Payload starts. This space is typical for US applications.
- The RUNIN is used to synchronize the internal generated bit clock of the receiver to the incoming bit-stream. RUNIN requires 3.5 bits in best case (all equal Manchester coded bits, therefore maximum number of edges), or 5.5 bits in worst case (101010 pattern therefore only half number of edges). The RUNIN is not required if there is a Wake Up and no space before the Preamble/TSI.
- The TSI (Telegram Start Identifier) marks the start of the Payload. The first bit of Payload follows directly the last bit of TSI. TSIs are possible up to 16bit. A short TSI may lead to false synchronizations at low input signal level or noise.
- The Payload contains the data which has to be received. The payload is terminated by the End of Message (EOM) criterion. This can be a CV, number of bits, or when bit synchronization is lost.

5 Definition of Sensitivity

Sensitivity is defined as the minimal input signal power [dBm] to achieve a certain bit error rate (BER) limit. Often this limit is 1%.

The TDA523x receives complete data frames. We have defined a reference data frame called “Pattern1”, and we specify a message error rate (MER) limit of 10%. This means sensitivity is the lowest input signal power to receive more or equal to 90% of correct data frames.

BER can be easily converted to MER and the other way round by following formulas:

$$MER = 1 - (1 - BER)^n, \quad BER = 1 - \sqrt[n]{(1 - MER)}$$

n…number of bits in data frame

The Infineon reference data frame “Pattern1” contains 4 bits RUNIN, 8 bits TSI, and 31 bits (PRBS5) payload. Therefore 39 bits have to match, n=39.

$$BER = 1 - \sqrt[39]{(1 - 0{,}1)} = 0{,}0027 >> 0{,}27\%$$

The TDA523x sensitivity limit or MER=10% equals to a BER=0,27%.

6 Installation of Support Software

Infineon Technologies supports TDA523x development by two PC programs, which are described in this Quick Start Guide.

Programs can be downloaded from the Infineon Technologies web site under *Select a Category*>> *Sensors* >> *Wireless Control* >> *Receiver* >> *TDA523x* >> *Documents* >> *Downloads*. Please note that the location for downloads may change.

The TDA523x Explorer is used to control the TDA523x Evaluation and Testboards, and may be also used to control and debug application hardware before software is available.

The IAF TDA523x Configuration Tool is used to quickly generate TDA523x configurations, for download using the TDA523x Explorer or for implementation into the application software.

Execute Setup.Exe in the TDA523x Explorer Setup folder, and follow the instructions. After successful installation connect the TDA523x Evaluation Board via USB to your PC. The driver required for the Windows New Hardware Found Wizard is in the folder USB-Driver.

Execute also Setup.Exe in the TDA523x_IAF_…_Install folder to install the IAF Configuration Tool.

Both programs require Microsoft's .NET Framework Version 1.1, which is available for free from www.microsoft.com/downloads.

.NET Framework Version 1.1 is often already installed on PCs used for technical development. If this is not the case, please download and install this program first.

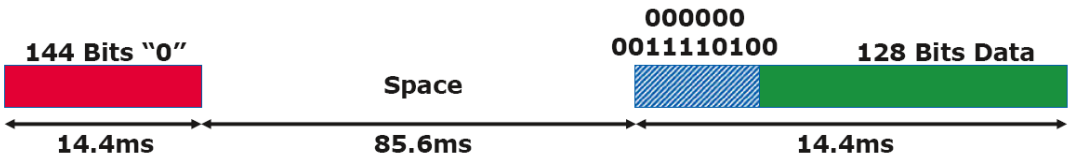
7 Create Your Own First Configuration

The following description relates to an example, which can also be downloaded from our web site. However there are also hints, how your own application can be implemented.

The configuration for TDA523x depends on RF frequency, data rate, data frame structure and some other details.

I use following data frame, which has to be received by TDA523x, for this example:

- RF frequency is 433.92MHz
- Modulation is FSK, minimum FSK deviation is 20kHz
- Data rate is 10kbps, +/-10% tolerance, Manchester-coded
- Data frame:



- Wake up pattern 144 bits of "0" (Manchester-coded, equals to "01" in chips)
- A space of 85.6ms
- A preamble of 16 bits, 4 bits reserved for RUNIN, 000011110100 used as TSI
- 128 bits of data (payload)

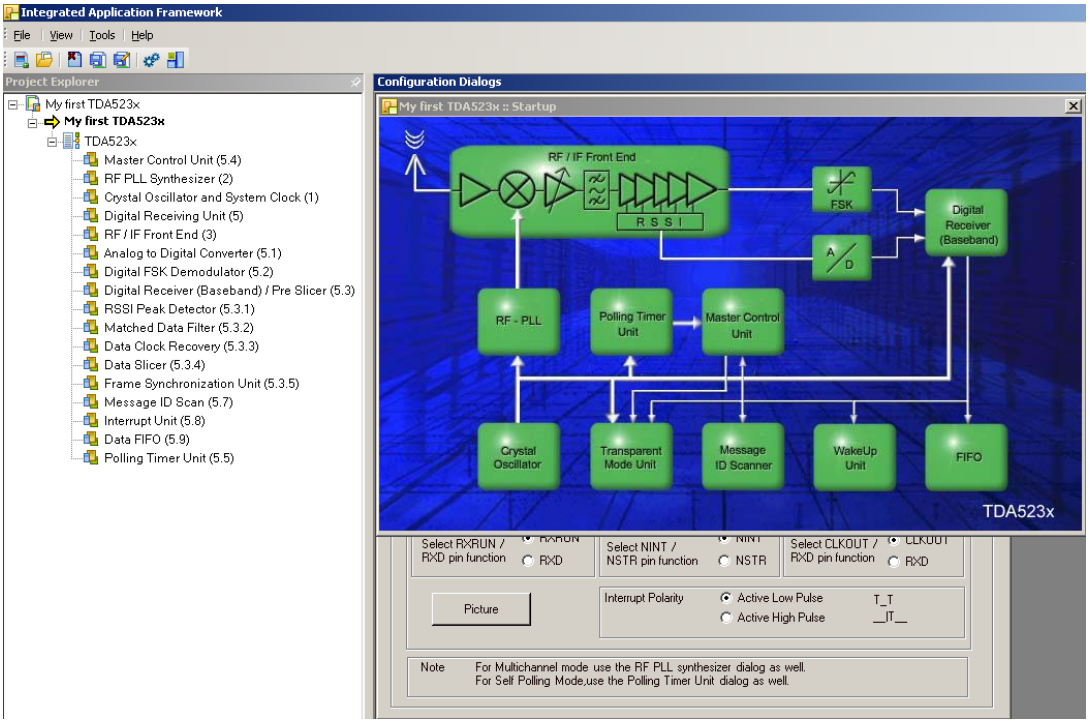
Execute the File IAF.exe. After a successful and typical installation there should be an icon on the Desktop, or it should be found clicking on the *Start Button* >> *Programs* >> *Infineon Technologies* >> *Integrated Application Framework* >> *IAF*.

When IAF is started, select *File* >> *New* >> *Workspace.* Define a name (I have used for this example *My first TDA523x*) for the Workspace and the location.

Select File >> New >> Project. Define a name for the Project (IAF inserts already the same name as for the Workspace, but this project name can be changed), the location is predefined.

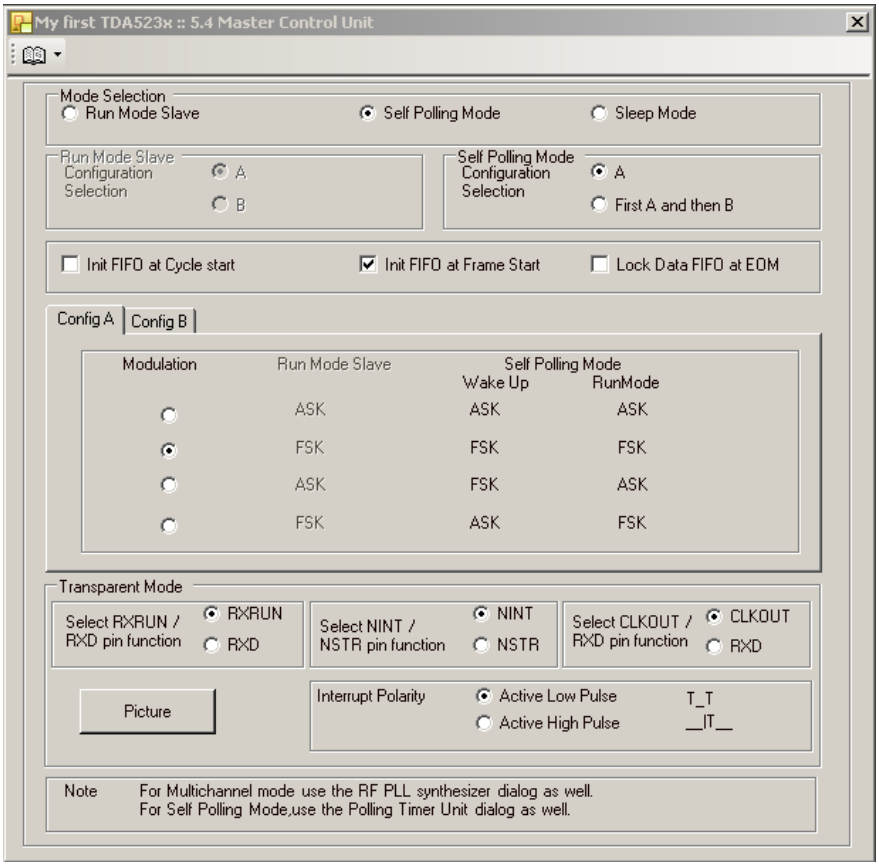
Note: Every configuration requires a separate project, when it should be kept. A Workspace may have several projects (= configurations). But when a Project is moved or sent by mail, the complete Workspace and Project has to be moved or sent. Therefore it is strongly recommended to have **different Projects** also **in different Workspaces** to avoid that sensitive project data is sent together with another project to the wrong recipient.

When the project is going to be created, IAF shows a message. When finished, select the project and double-click the TDA523x icon in the selection window. Now the real creation starts and takes a few seconds. Finally the project is opened.



On the left side the different function blocks can be selected, on the right side are configuration windows, and a simplified block diagram as first page.

7.1 Select Master Control Unit



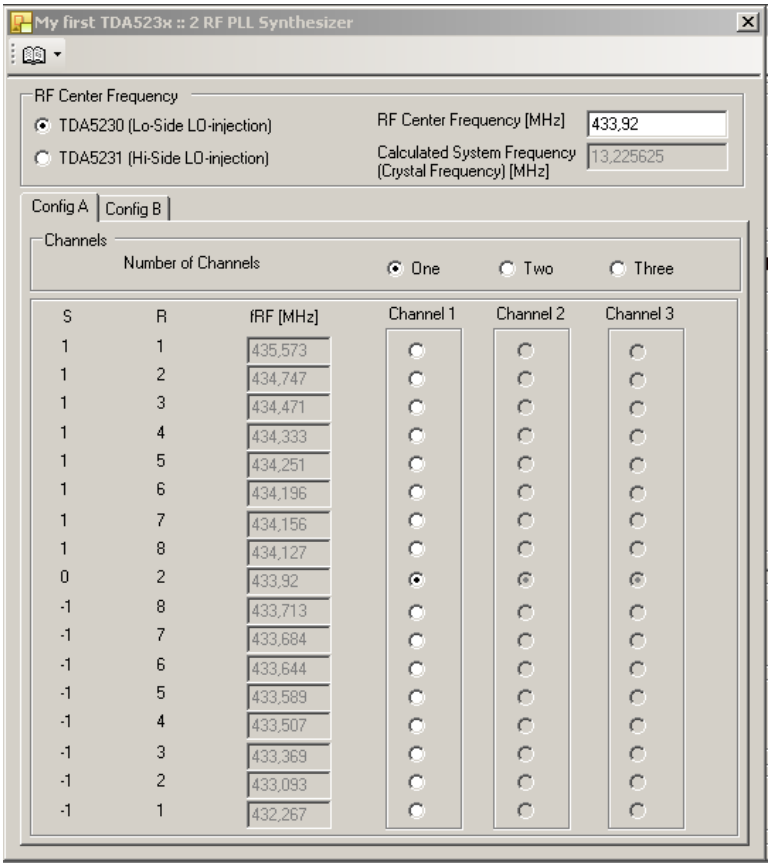
We select *Self Polling Mode*, and *A* as *Configuration Selection*.

The selected *Init FIFO at Frame Start* means, that the data FIFO is cleared, when a new synchronization to a data frame (TSI) has been received.

We select *FSK* for *Wake Up* and *Run Mode*.

The settings for *Transparent Mode* are left as they are, this feature is not required.

7.2 Select RF PLL Synthesizer



TDA5230 is selected for the 434 and 868MHz range, *TDA5231* for the 315MHz range.

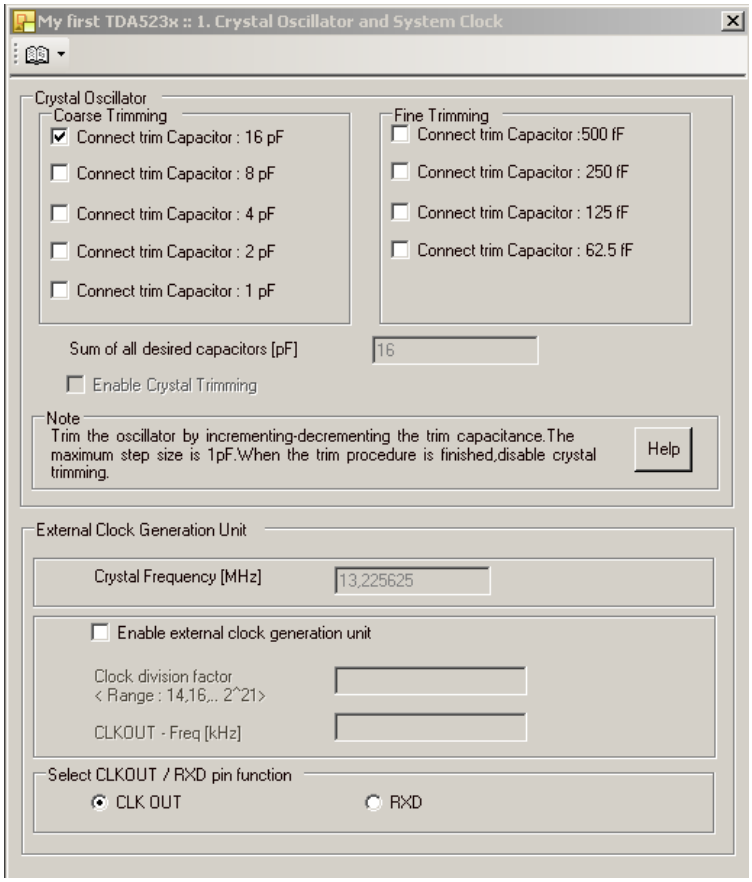
In *RF Center Frequency* the desired RF frequency is entered.

In *Calculated System Frequency* the required crystal frequency is shown.

TDA523x allows scanning up to 3 RF channels (per configuration) one after another. Number of channels and the frequencies are selected in the *Channels* part of this window.

Note: The center frequency is the frequency in the middle of all available RF frequencies for a certain crystal frequency. This does not mean that this frequency has to be used. Any of the frequencies in f_{RF} list can be used also for single channel applications.

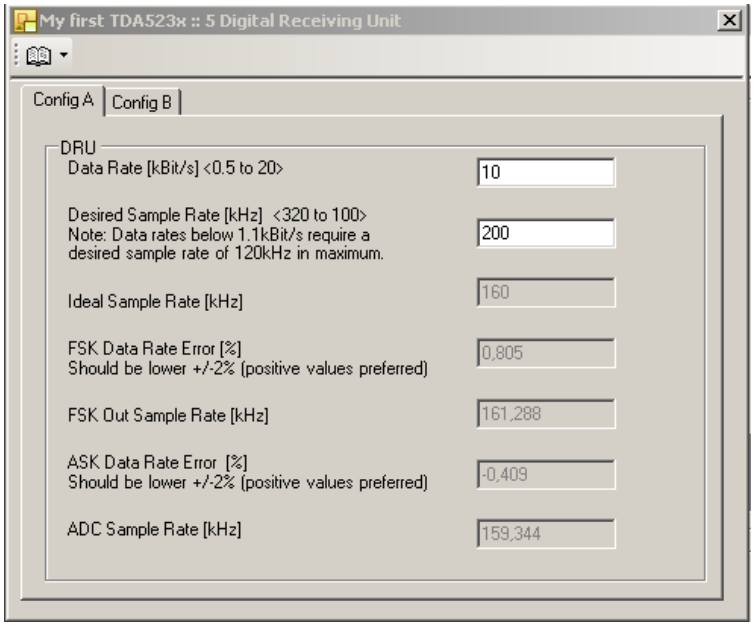
7.3 Select Crystal Oscillator and System Clock



The *Crystal Oscillator Trimming* stays unchanged. Crystal trimming is usually done (if this accuracy is required) individually for each circuit board in the production line. The trimming settings have to be added to the configurations.

The *External Clock Generation Unit* should not be enabled, if not used, to save power consumption. If it is used, the *Clock division factor* has to be defined and entered.

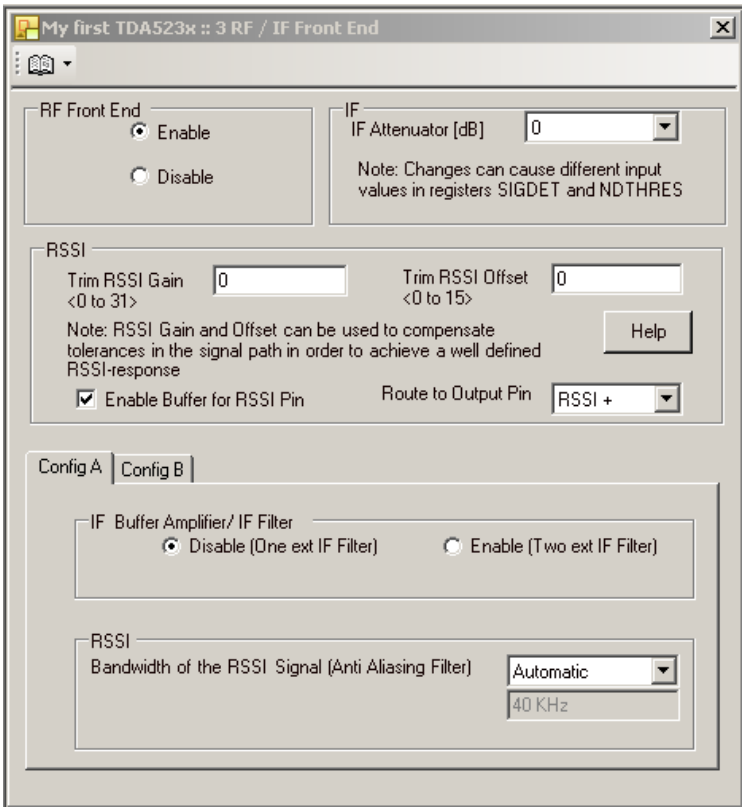
7.4 Select Digital Receiving Unit



Enter the given *Data Rate*.

Select an ADC *Sample Rate*. The sample rate has only a small influence on the performance. For ASK the sample rate should be the highest possible, for FSK it can be lower. When the desired sample rate is entered, the program calculates the closest possible sample rates and data rate errors. It may happen that a calculated sample rate gives a data rate tolerance bigger than 2% and an error message is displayed. In this case change the *Desired Sample Rate* till a sample rate with lower data rate error is calculated.

7.5 Select RF / IF Front End



RF Front End has to be enabled (of course).

The *IF Attenuator* should be only increased if an external LNA is used. In this case attenuation should be set to the gain of the external LNA.

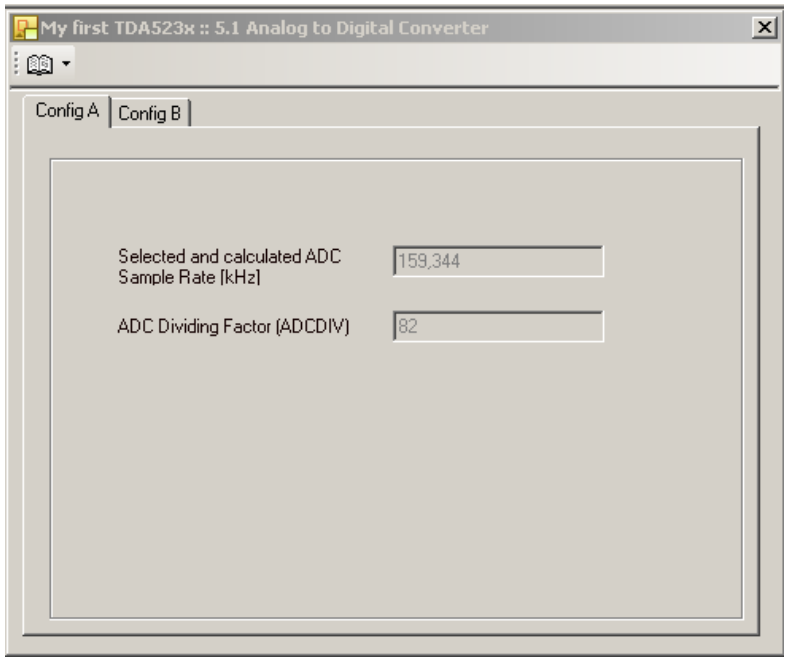
RSSI trimming of *Gain* and *Offset* is typically done individually for each circuit board in the production line, and the trimming values added to the configuration.

Buffer for RSSI Pin should be enabled, because for debugging purposes a data reception of strong signal can be nicely observed with an oscilloscope.

IF Buffer is disabled when only one external IF filter is used, which is typically the case. Two IF filters are used if the IF bandwidth should be different for the two possible configurations.

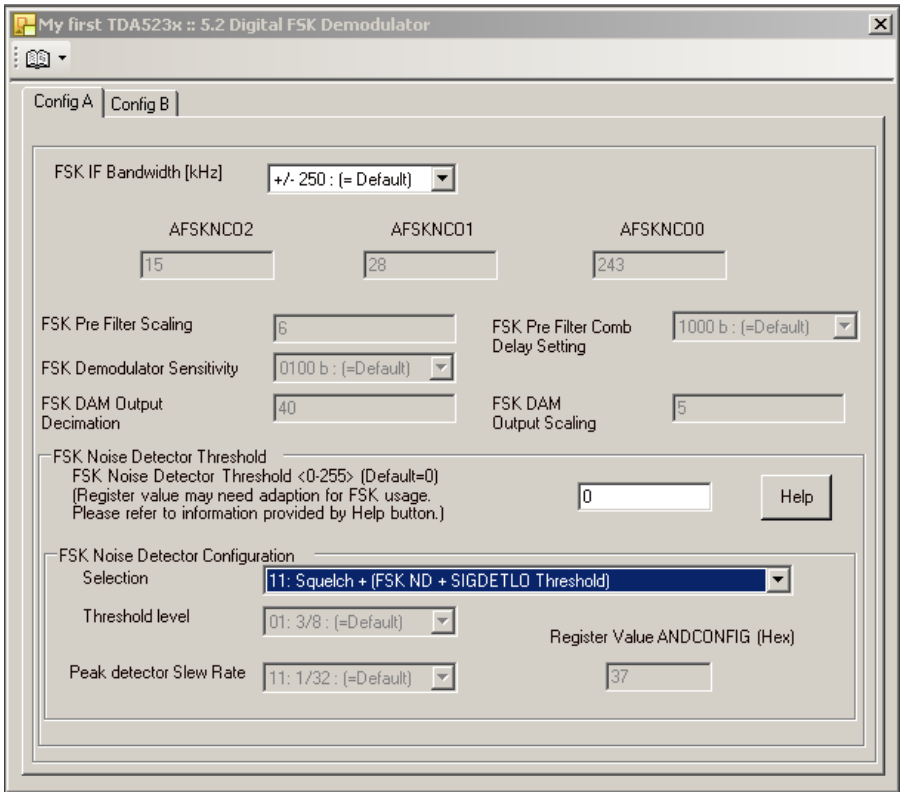
RSSI Bandwidth should be set to *Automatic*. In ASK systems with high duty cycle tolerances it should be generally set to *40kHz*.

7.6 Select Analog to Digital Converter



There is nothing to do on this page.

7.7 Select Digital FSK Demodulator

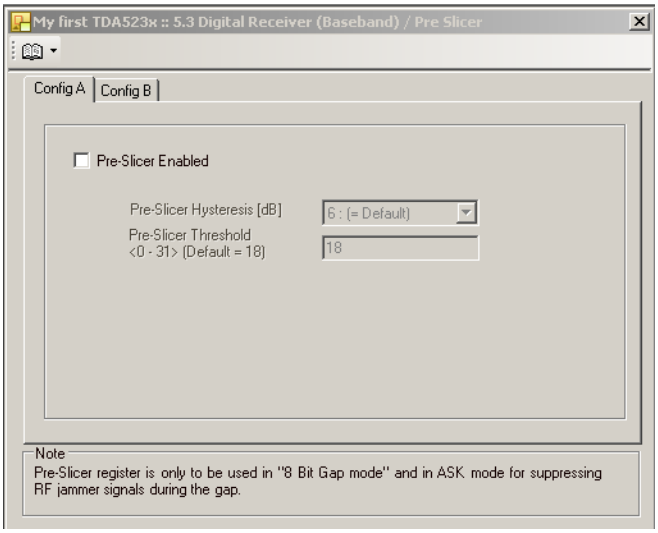


FSK IF Bandwidth should always be set to +/-250 kHz. Selecting a smaller bandwidth may create problems in finding an ADC sample rate.

The *FSK Noise Detector Threshold* will be defined later.

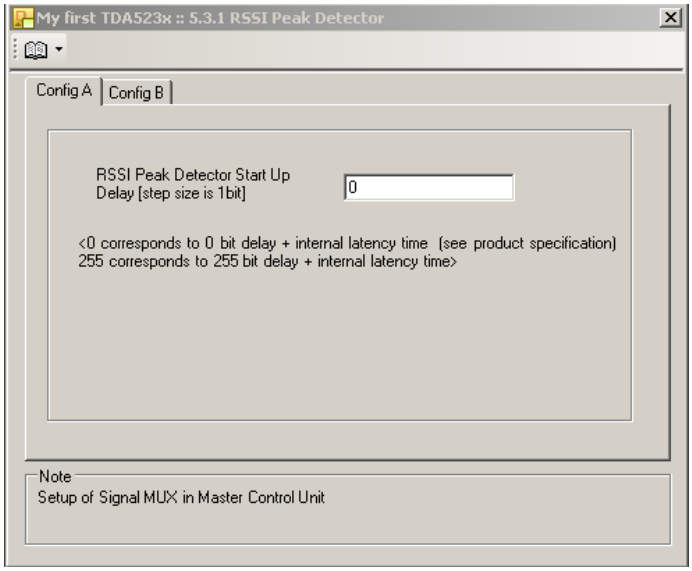
For the FSK *Noise Detector Configuration* use *Squelsh + (FSK ND + SIGDETLO)*.

7.8 Select Digital Receiver (Baseband) / Pre-Slicer



The *Pre-Slicer* remains disabled.

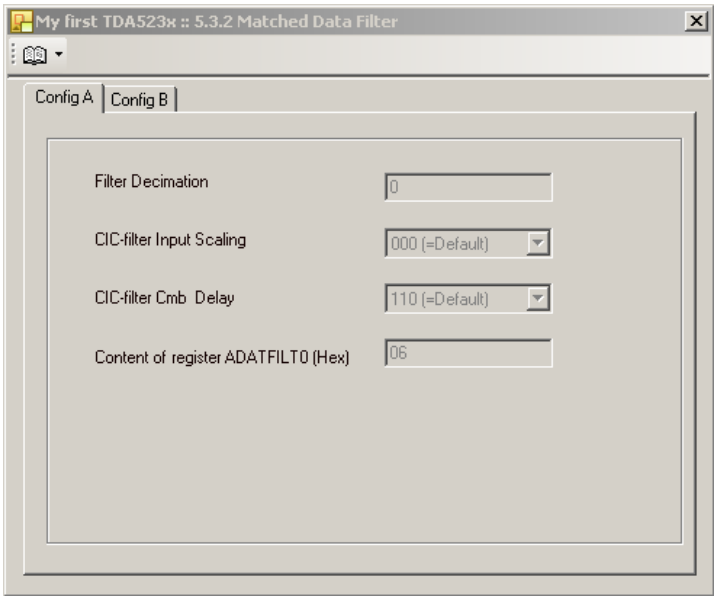
7.9 Select RSSI Peak Detector



The *RSSI Peak Detector* (Register RSSI1) is used to automatically measure the input signal strength during a received data frame.

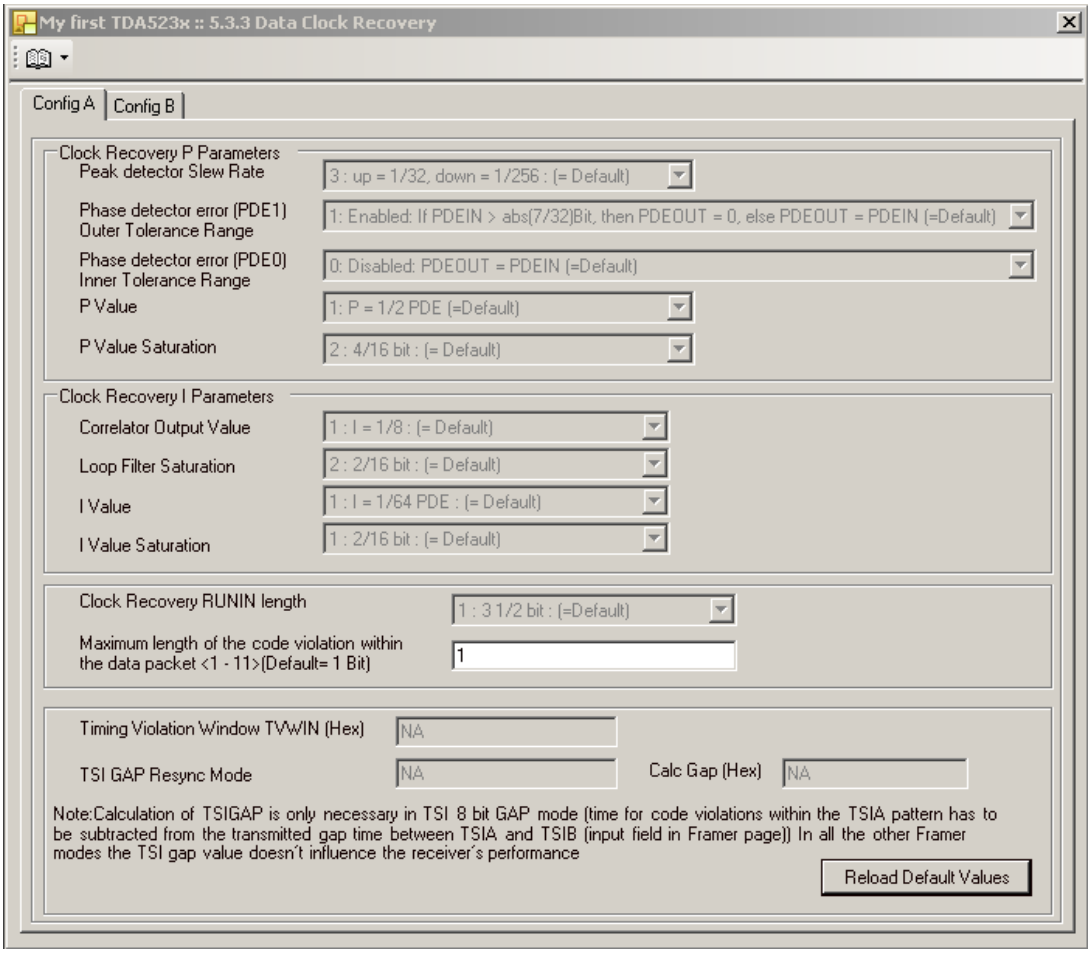
With *RSSI Peak Detector Start Up Delay* the start of this measurement can be delayed. Using this default value “0” means, the measurement starts directly after synchronization to the data frame.

7.10 Select Matched Data Filter



There is nothing to do.

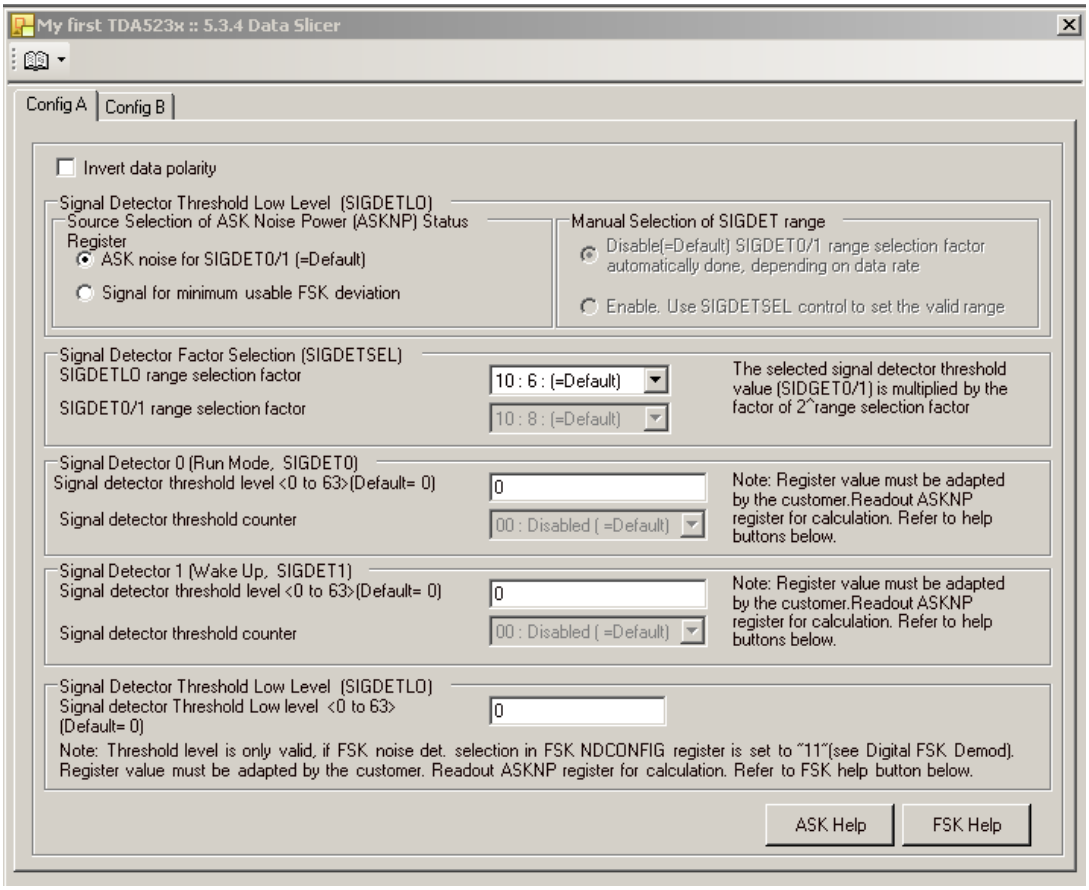
7.11 Select Data Clock Recovery



Only the *Maximum Length of Code Violations* has to be entered.

Manchester code violations in the TSI or data frame have to be considered. The minimum is 1. In this example there is no code violation therefore the minimum is used.

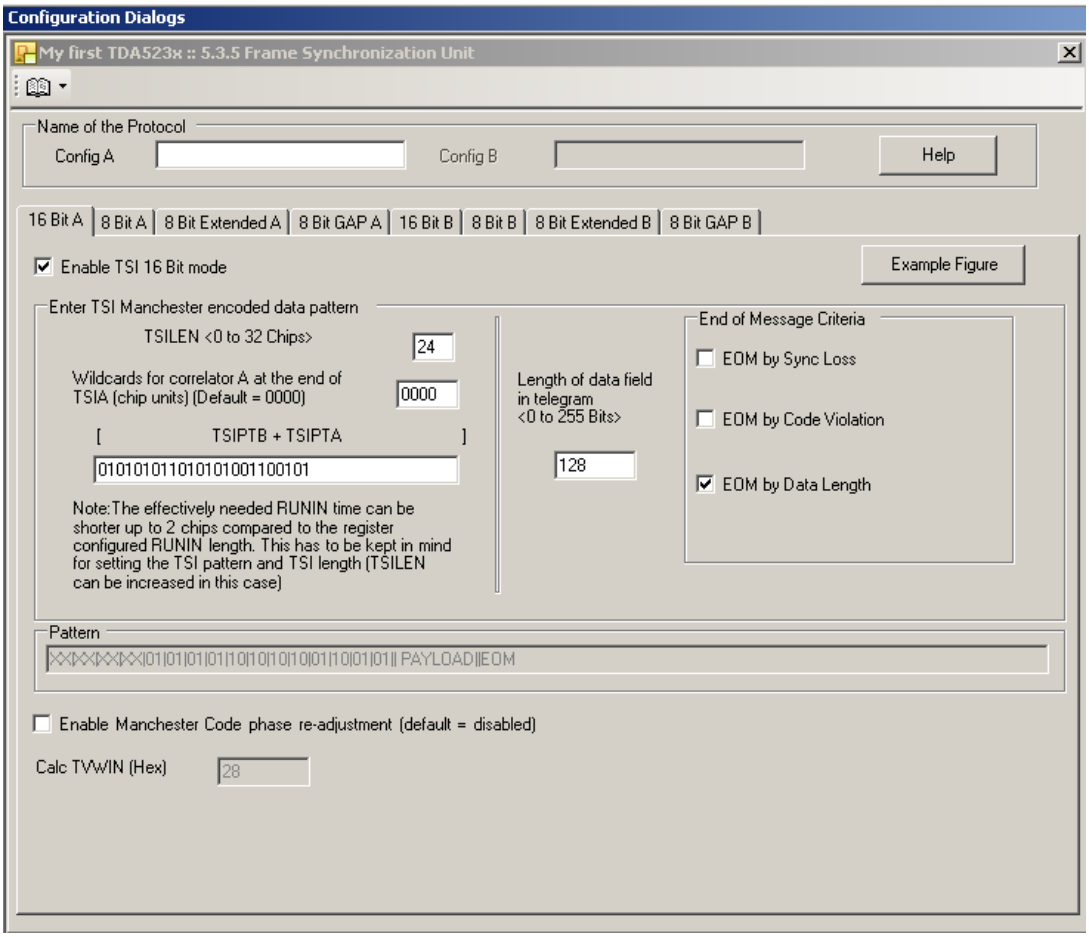
7.12 Select Data Slicer



Invert data polarity is used for Inverted Manchester only

- The values for Signal Detector Thresholds are defined later.

7.13 Select Frame Synchronization Unit



Even if the TSI is shorter than 8 bit, *TSI 16 Bit Mode* should be selected. The *8 Bit Modes* are used, if TDA523x should synchronize on two different TSI patterns. Only one TSI mode can be enabled per configuration.

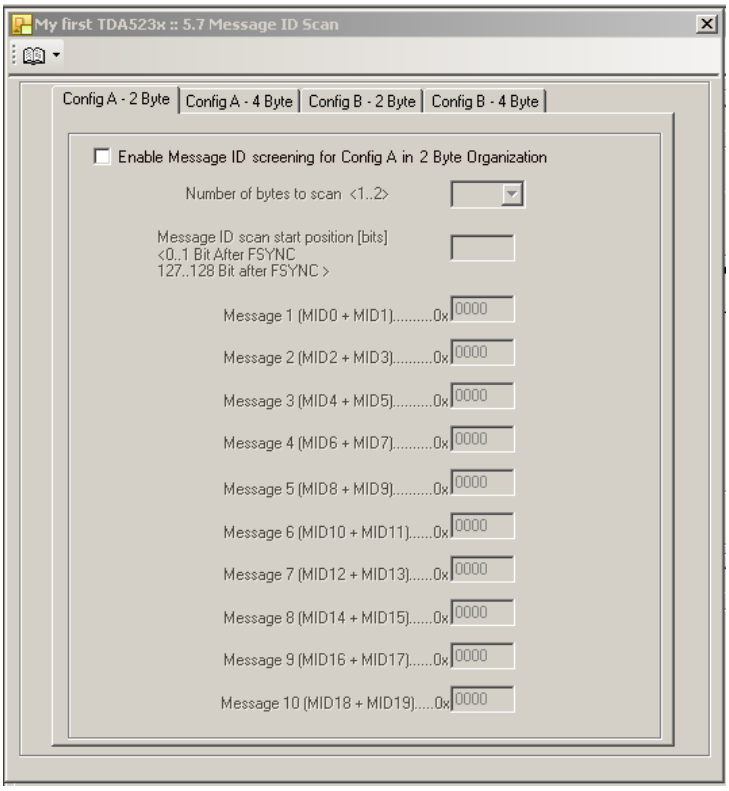
In our example we use 12 bit, which equals to 24 chips. The “24” is entered under *TSILENA*. The first 4 bits of the Preamble are required for RUNIN (minimum 3.5 bits).

The *Wildcards* are used to allow more TSI patterns which differ in the last 4 chips. E.g. if *Wildcards* are set to 0001 the last chip can be either 0 or 1. In this example this feature is not used and the setting remains *0000*.

The synchronization pattern of 000011110100 is translated into chips (010101011010101001100101) and entered under *TSIPTB + TSIPTA*.

If possible (fixed bit count in payload) *EOM by Data Length* should be selected and the bit count entered in the dedicated field.

7.14 Select Message ID Scan

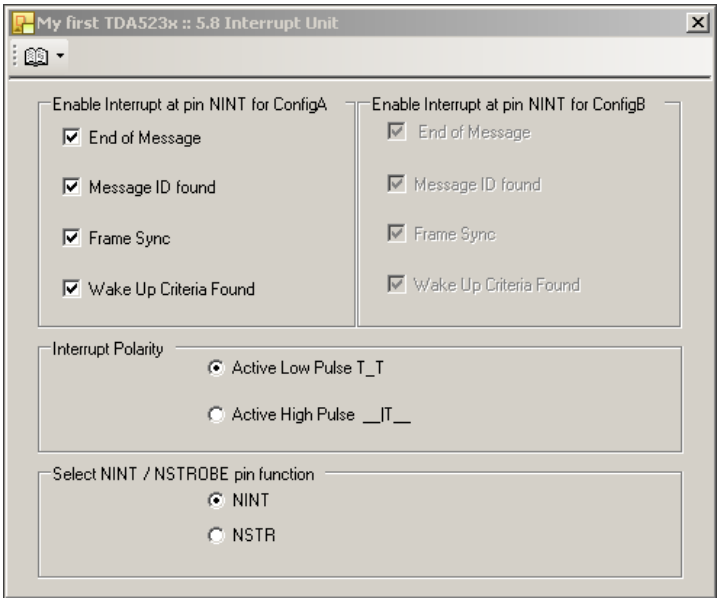


Message ID Screening is not used in our example, and is therefore not enabled.

Two important hints:

- Take care that Message ID Screening is not accidentally enabled on one of the pages in the background (4 Byte or Config B).
- If Message ID Scan is used, then enter the MID values in all lines. Otherwise the default MID of 0000 is also accepted as valid MID.

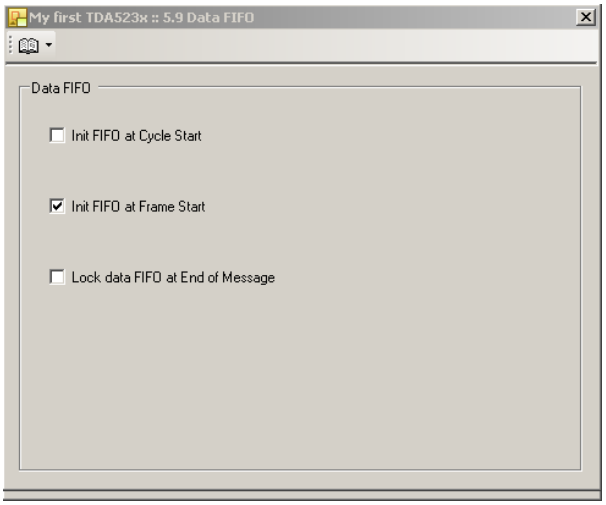
7.15 Select Interrupt Unit



We keep all interrupts enabled. The occurrence of an interrupt gives important information for debugging. After a successful debugging, only the End of Message Interrupt is required in this example.

Interrupt Polarity and *NINT/NSTROBE* pin function remain unchanged.

7.16 Select Data FIFO



Init FIFO at Frame Start means that the data FIFO is initialized and cleared when a new synchronization is detected.

7.17 Select Polling Timer Unit

Before filling this page with content, we need a short discussion about wake up criteria and the calculation of self polling timing.

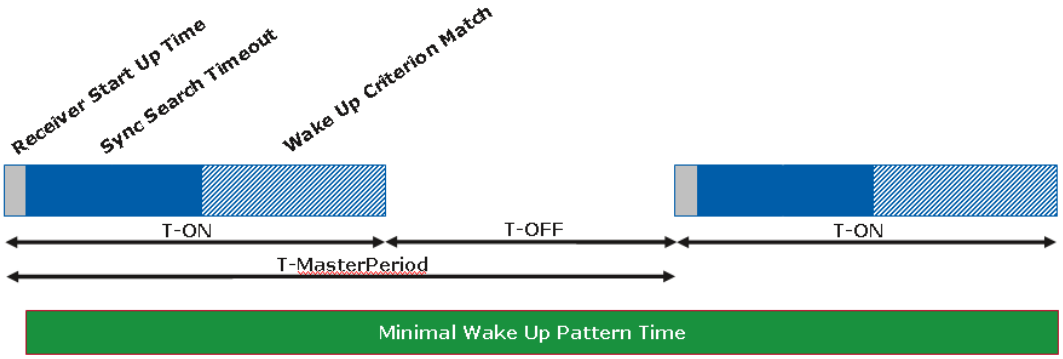
The typical wake up mode, also used for this example, is Fast Fall Back to Sleep. In this mode, TDA523x tries to synchronize for the maximum time of Sync Search Timeout. If no synchronization is possible, TDA523x is switched to the sleep position to save power. If synchronization is possible (which sometimes also happens without a valid input signal), TDA523x waits for the selected wake up criterion to be fulfilled, limited by T_{ON}.

The wake up pattern in this example is a chain of “0”s. The used Wake Up Criterion is therefore Equal Bits Detection.

The selected number of wake up bits is a trade off between power consumption, more bits require a longer T_{ON} and a shorter T_{OFF}, and a higher False Alarm Rate (FAR). We use 6 bits for wake up, which is a good compromise.

Calculation of Self Polling Timing:

Calculations for T_{ON} and T_{OFF} have to be solved. The sum of T_{ON} and T_{OFF} is $T_{MasterPeriod}$.



T_{ON} > Receiver Start Up + Maximum Sync Search + Maximum Wake Up

- Receiver Start Up is the time needed for the internal power up sequence, calculated by the IAF tool and shown in the right button corner of the Polling Timer Unit page (*t_startup*).
- Maximum Sync Search is the time required to synchronize the bit clock at minimum data rate (longest bit duration, in this example the data rate tolerance is 10%). The synchronization requires 7.625 bits. (Sync Search Timeout limits the Sync Search and stops the sync trials to switch back to sleep mode).
- Maximum Wake Up is the time required to fulfill the wake up criterion at minimum data rate (longest bit duration). The wake up criterion is a match to 6 equal bits.

T_{ON} > 0.5759ms + 7.625bit/(0.9*10000bps) + 6bit/(0.9*10000bps)

T_{ON} > 0.5759ms + 1.5139ms = 2.0898ms

$T_{OFF} < T_{MasterPeriod} - T_{ON}$

$T_{MasterPeriod}$ is dependent on the available Minimal Wake Up Pattern Time. The Master Period has to be short enough that if the Sync Search started is missed the first time for a very short time, there is a second chance for a wake up inside the minimal Wake Up Pattern Time.

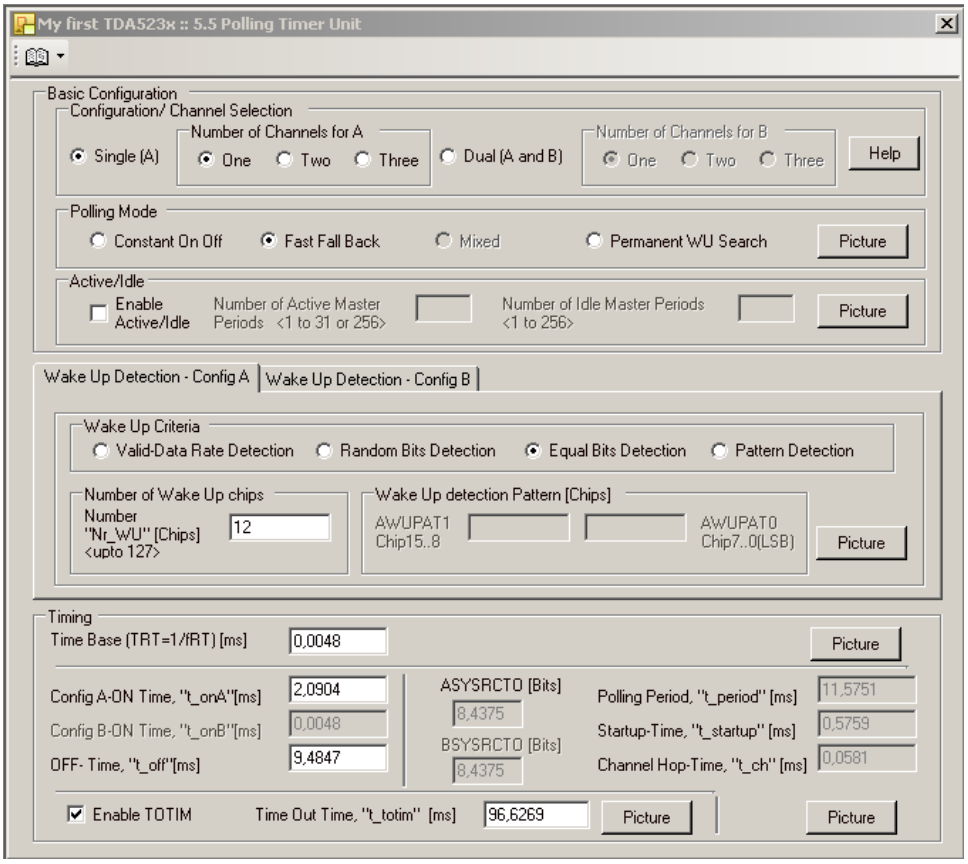
$T_{minimalWakeUpPattern}$ = WakeUp bit count / maximum data rate

$T_{minimalWakeUpPattern}$ = 144bit / (10000bps*1.1) = 13.0909ms

$T_{MasterPeriod} = T_{minimalWakeUpPattern}$ - Maximum Sync Search - Maximum Wake Up

$T_{MasterPeriod}$ = 13.0909ms – 1.5139ms = 11.5770ms

$T_{OFF} < 11.5770ms – 2.0898ms = 9.4872ms$



We select *Polling Mode*, *Fast Fall Back* and *Equal Bits Detection* for *Wake Up Criteria*.

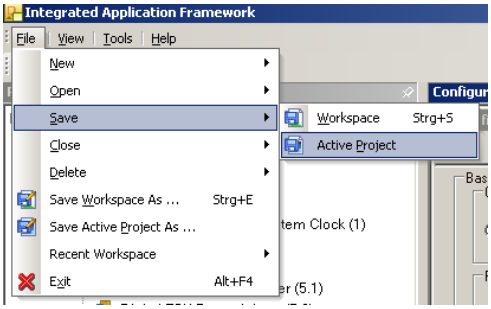
The number of Wake Up chips is 12 (=6 bits).

For the *Time Base*, we use the smallest possible (64/crystal frequency), and then we enter the calculated T_{ON} and T_{OFF}. IAF automatically converts the input to closest possible values.

Check that the final Polling Period calculated by IAF is shorter than the calculated t_period ($T_{MasterPeriod}$). Otherwise T_{OFF} should be reduced.

TOTIM is used to return to Self Polling if a (false) synchronization has occurred, and has always to be enabled! The timeout depends, if there is a gap between wake up pattern and preamble. In this example a gap of 85.6ms is specified with a tolerance of 10%. The selected timeout value is 95ms. IAF will again change to the next possible value.

Save the project.

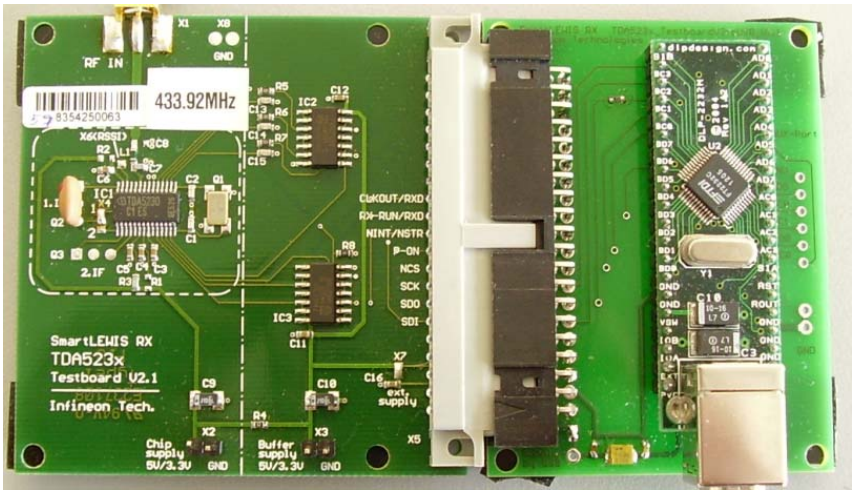


Create the configuration file by clicking  .

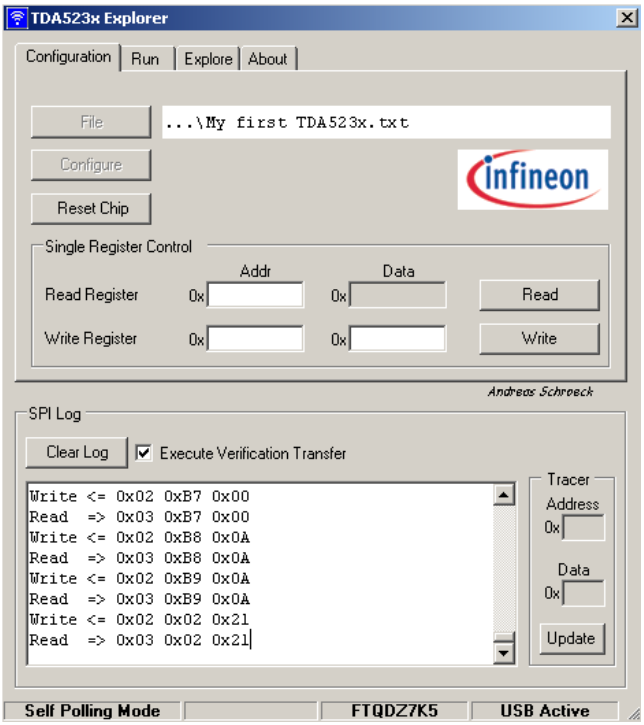
The configuration file is used in the next step to generate the signal- and the noise detector threshold levels.

7.18 Signal and Noise Detector Thresholds

Connect a TDA523x Evaluation Board to the PC.



Start the TDA523x Explorer. After typical installations, it is executed by clicking on the *Start Button* >> *Programs* >> *Infineon Technologies* >> *TDA523x Explorer.exe.* (See chapter 6.0 regarding installation of TDA523x Explorer)

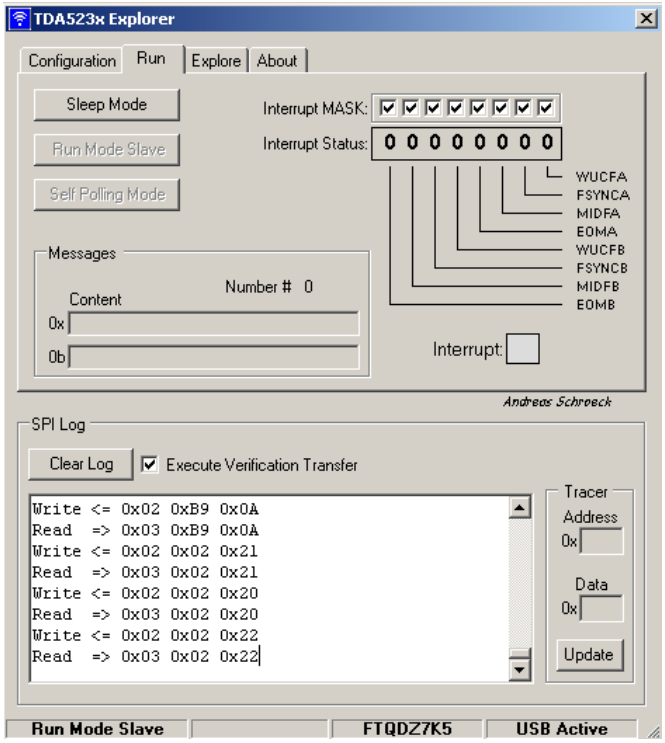


Click on the File button and select the configuration file. It is usually found in the subfolder Source inside of the workspace (in our case /My first TDA523x/Source).

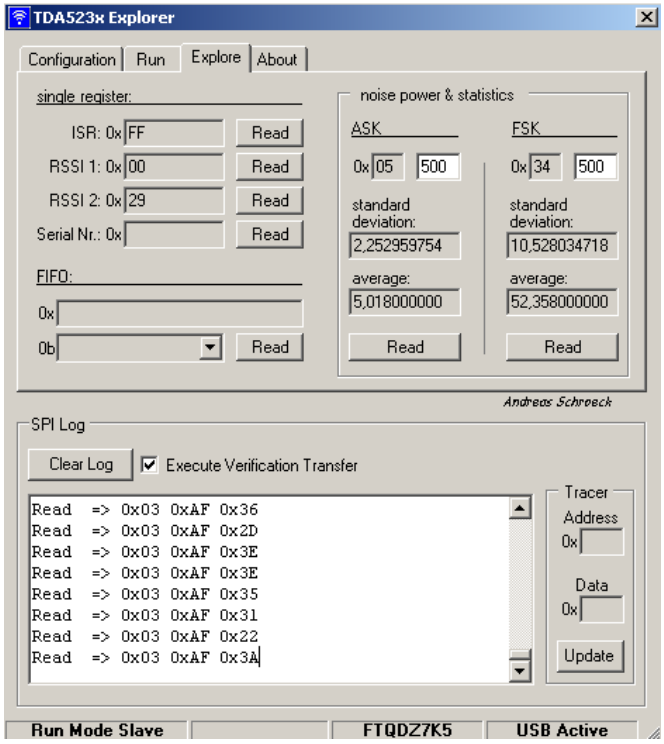
Enable the Execute Verification Transfer tick box (to automatically verify the SPI data transfer), and click on Configure.

The configuration is sent to the evaluation board, and the TDA523x switches to Self Polling Mode.

The measurements required for signal- and noise detector threshold have to be done in Run Mode. Therefore in the TDA523x Explorer, we change to the Run page, click first on Sleep Mode, then on Run Mode Slave.



Change to the Explore page.



In the field noise power & statistics add 500 (this is the number of register readings) for ASK and FSK. Click first on the ASK read button, when readings are finished, repeat on the FSK read button.

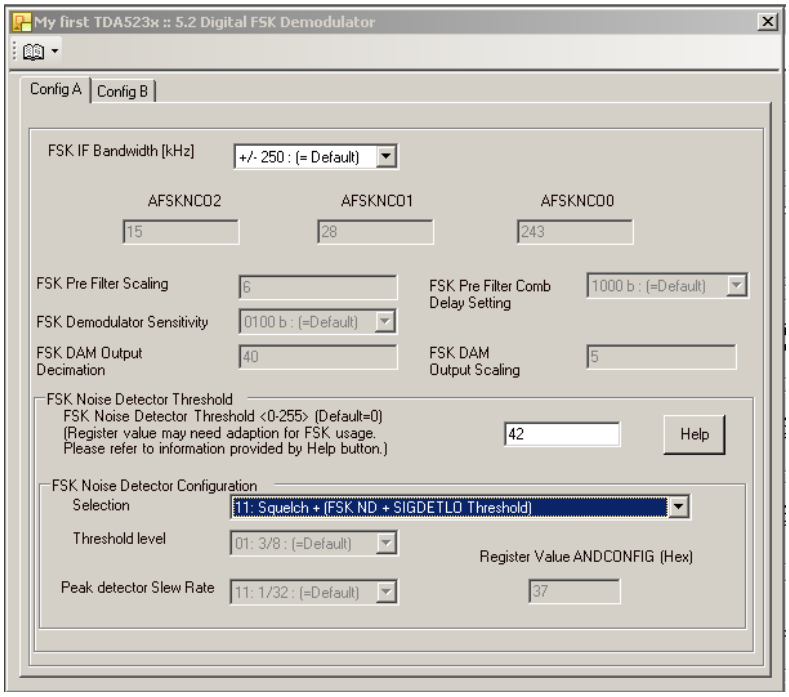
Following calculations have to be done:

- Signal Detector Threshold = ASK-average + 2*ASK-standard deviation
- Signal Detector Threshold = 5 + 4.5 = ~9
- Noise Detector Threshold = FSK-average – FSK-standard deviation
- Noise Detector Threshold = 52.4 – 10.5 = ~42

Note:

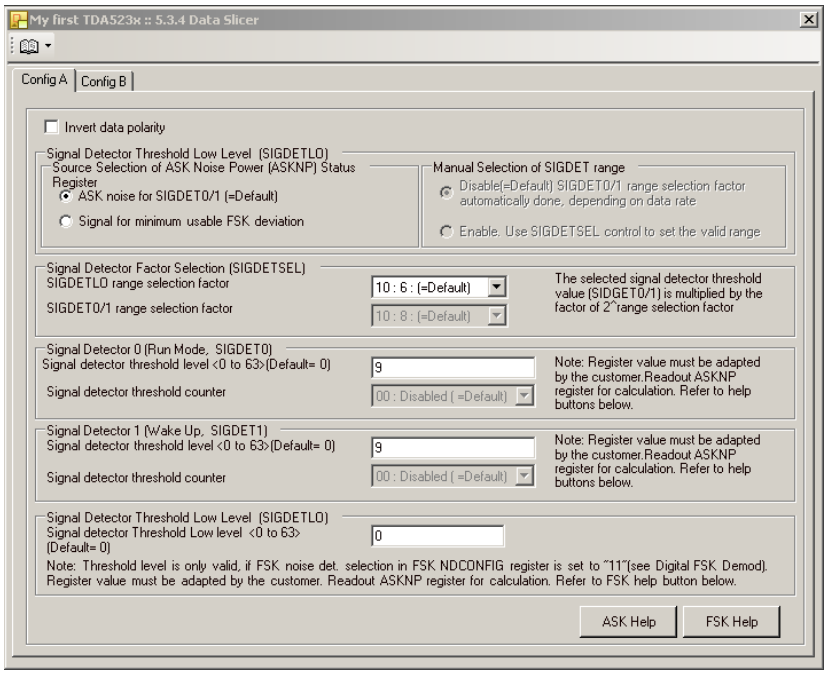
- There is no RF signal applied.
- Use the RF input for this measurement as it is used for evaluation or application. For this example an antenna has been connected.
- When RF input conditions change (different antenna, different matching, different LNA, different filters (e.g. IF filter, different IF attenuation) the measurement has to be repeated and calculated threshold values adapted.
- Also different data rates require different thresholds. Therefore there are two sets of threshold registers available which have to be used in dual configuration applications.
- In ASK applications only ASK measurement is required (FSK will show 0).
- The calculated values are start values and may require optimization regarding sensitivity and false alarm rate. For later optimization iteratively reduce and increase threshold values till the optimal sensitivity / false alarm behavior is achieved.

Open the IAF window and select Digital FSK Demodulator.



Fill in the calculated value for the *FSK Noise Detector Threshold*.

Select Data Slicer.



- Fill in the calculated value for the Signal Detector Threshold as well in the field for *Run Mode* as for *Wake Up*.

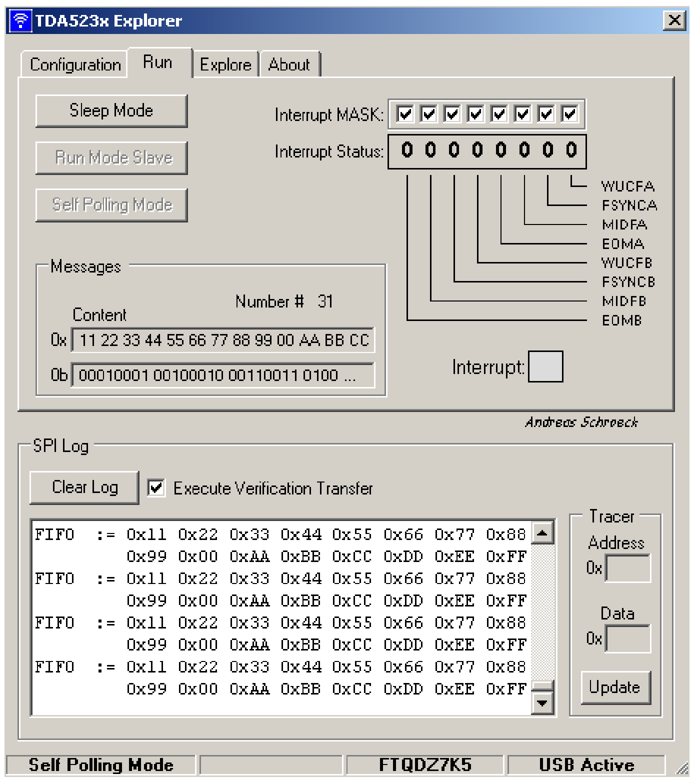
Save the IAF project and create a new configuration as before.

The first TDA523x configuration is now finished and can be used for evaluation.

8 System Evaluation

If you have a RF transmitter available which is able to transmit the pattern fitting to your first TDA523x configuration, you can start to evaluate TDA523x.

Open the TDA523x Explorer as before, download the configuration, and open the Run page.



Received data frames are shown in the Log window. In this example the payload of the data frame is 0x11 to 0xFF.

9 Debugging

If a system does not work, there may be many reasons either on receiver, but also on transmitter side.
Here is a short guide how to debug a non working TDA523x Evalboard RF link.

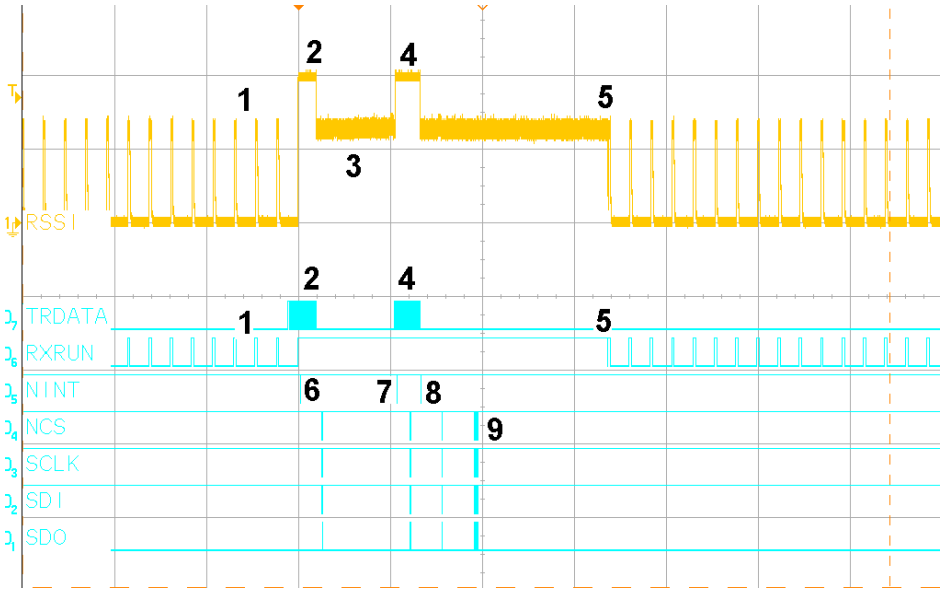
Following test-points are practical for debugging:

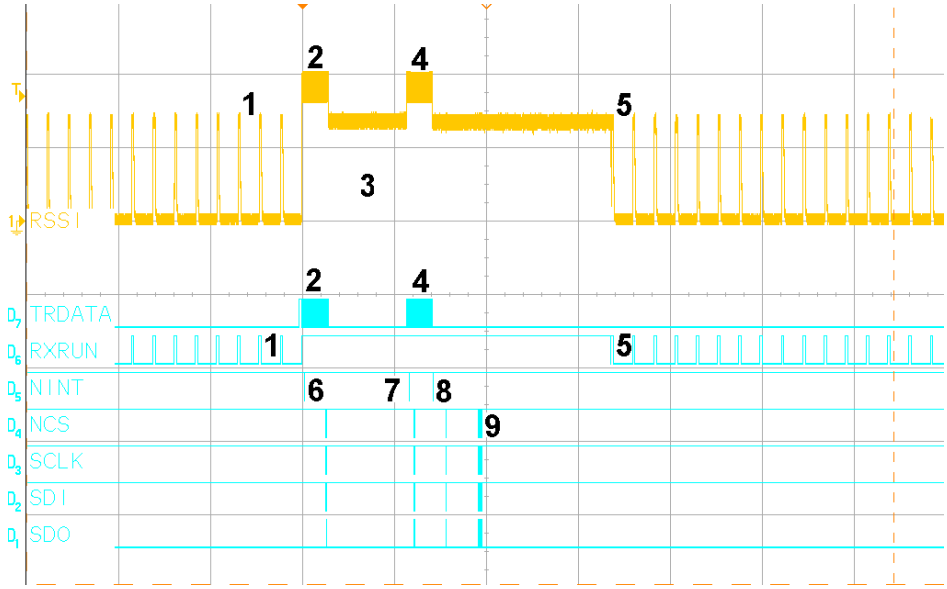
- X2 Chip supply voltage
- R3 Serial resistor for 5V supply, used to measure supply current
- Q2 IF signals before and after the IF filter. A data transfer is only visible for strong input signals. Also the on and off of the RF part is visible.
- X6/RSSI RSSI signal shows the changing input signal strength during data transmission. Data is only recognizable for ASK and strong input signals.
- X5/RX-RUN Shows the on/off of the RF part, the self polling timing is visible
- X5/NINT Inverted interrupt line shows if any kind of receive interrupt occurs
- X5/P-ON A low level puts the TDA523x in power down mode. Has to be always high during operation! This pin can also be used as a low active RESET pin
- X5/NCS SPI chip select, has to be low during SPI traffic
- X5/SCK SPI data clock has to show activity during SPI traffic
- X5/SDO SPI data out carries the data to the host processor
- X5/SDI SPI data in for TDA523x

9.1 Start to Debug

- Move transmitter and receiver (if the connection is not wired) as close as possible together.
- If possible use ASK modulation for debugging.
- It may be helpful to switch TDA523x to Run Mode Slave. You can do this in the TDA523x Explorer, when the TDA523x has been configured. Select the *Run* page, click *Sleep*, then click *Run Mode Slave*. In the bottom left line, *Run Mode Slave* should be visible. Please note that in Run Mode Slave only one configuration, typically A (depends on setting of Register CMC0 bit 3) and channel 1 is used.
- Have all interrupts and RSSI output enabled (as recommended in the IAF chapter).

The following pictures show a successful reception recorded by a mixed signal scope, the first picture for FSK, the second for ASK.

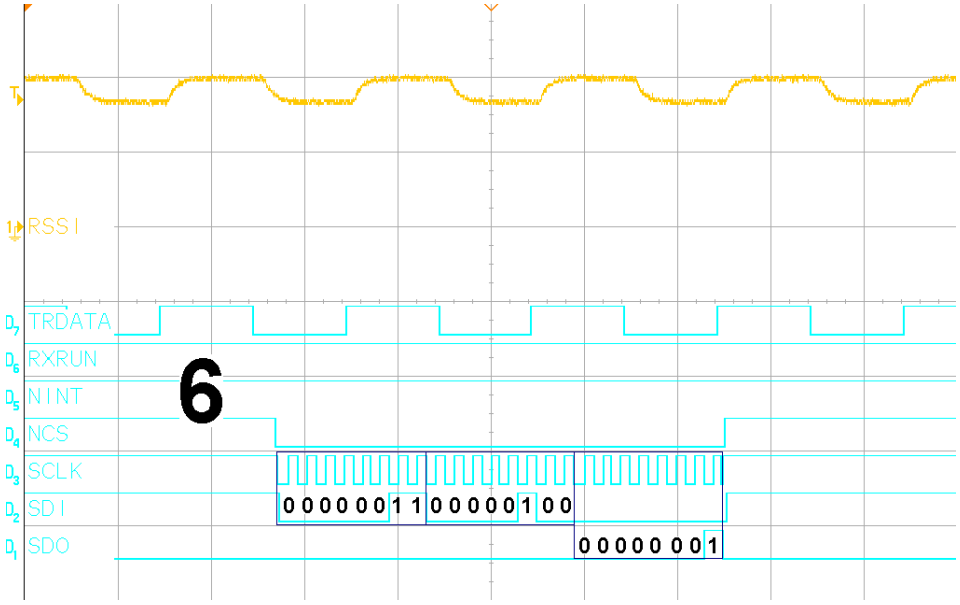




TDA523x is in Self Polling Mode, visible at the toggling of RX-RUN. Also RSSI shows the switching on/off of the TDA523x RF path (1).

The transmitter sends the wake up pattern (2). Now the difference between ASK and FSK is visible on RSSI. In FSK, RSSI shows a stable high, because the signal power is not varied, while in ASK the signal power is used for modulation, and therefore the transmitted data is visible in the RSSI amplitude (shown in detail in a following picture). During data transmission the RSSI signal does not go down to its offset, like during the gap, because the transmitter is not able to switch RF completely off. The TRDATA signal is the data line controlling the modulation input of the RF signal generator.

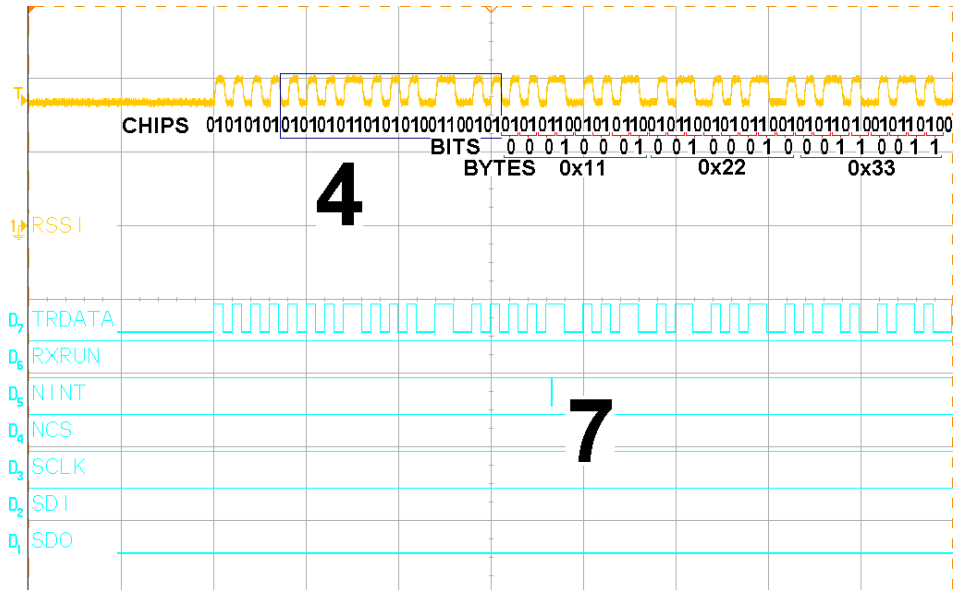
When the wake up pattern is recognized during a polling phase, the wake up interrupt is generated (6) and the hosting SW, in this case TDA523x Explorer, reads the interrupt status register.



Observing the SPI traffic, we see that the command Read from Chip (0000 0011) and the address of the interrupt status register (0000 0100) are sent to the TDA523x. TDA523x answers with a Wake Up Criteria Found (0000 0001) on the SDO line.

After the wake up pattern, there is the gap as used for the IAF example (3).

After the gap the preamble is transmitted. In ASK and at a strong input signal, the received data is visible on RSSI. This is the big advantage having an ASK signal for debugging.

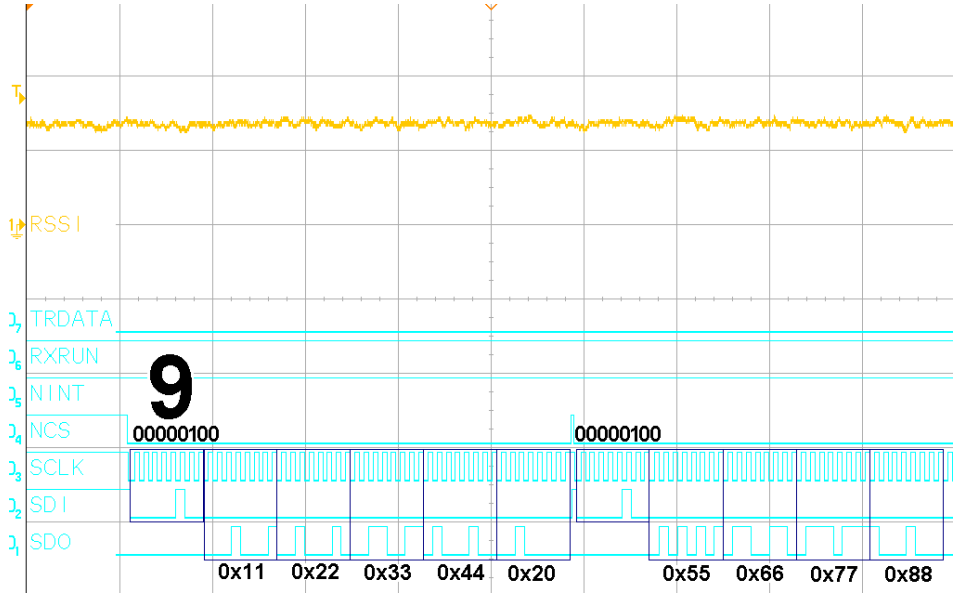


The RSSI allows observation of the incoming bit stream, the first four bits (8 chips, 01010101) for RUNIN, the 24 chips for TSI (010101011010101001100101), and the first bytes of the payload (0x11, 0x22, 0x33).

After the TSI, there is, with a short delay, the FSYNC interrupt (7), and the interrupt status register is read.

After the payload is received, the EOM (End of Message) interrupt is generated, and the interrupt status register read by the hosting SW (8).

After a short time the FIFO is read.



The read from FIFO (9) starts with Read FIFO from Chip command (00000100). Then the chip sends four bytes of FIFO data (0x11, 0x22, 0x33, 0x44) and the status byte. The 0x20 (decimal 32) means, 32 bits of this block are usable. Then the read of the next FIFO block starts with the 00000100, and the next four bytes are sent via SDO (0x55, 0x66, 0x77, 0x88). All together four four-byte blocks are sent to the host controller this way.

After TOTIM has elapsed, TDA523x returns to self polling (5).

9.2 Debugging Checklist

For debugging it is ideal to have RSSI output enabled and all interrupts enabled. An ASK pattern is easier to debug than FSK.

- Check supply voltage
- Check current consumption across R3
- Too low current
  - wrong operating mode?
  - P_ON line low (should be high)?
- Download configuration using Execute Verification Transfer. Any failure message?
- Switch to Run Mode Slave
  - Is data frame for configuration A (or B depending on register CMC0 bit 3) and channel 1 received?
    - If yes, check Self Polling timing, too short T_{ON}, too long T_{OFF}
- Use a known working configuration
  - Does this configuration work?

Connect an oscilloscope, if possible have one probe at digital TX data and trigger with start of telegram.

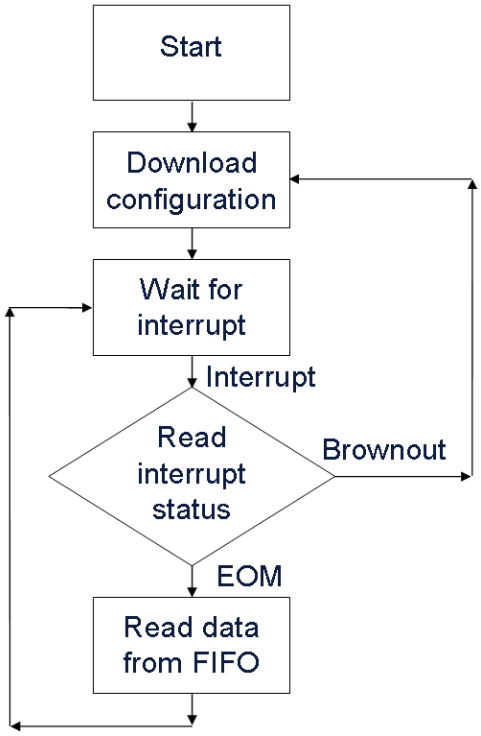
- SPI data transfer correct? NCS active? Is there a SCLK? SPI voltage levels correct? Show SDI and SDO activities? Are SPI voltage levels correct?
- Check crystal clock.
- Check RX-RUN is it toggling in self polling mode?
  - Stuck high
    - Is TOTIM enabled?
    - Correct operating mode selected?
    - Have T_{ON}, T_{OFF} and $T_{MasterPeriod}$ the expected timing?
    - Is the transmitted wake up pattern long enough?
  - RX-RUN does not stop toggling during the wake up pattern.
    - Is definition of wake up pattern correct?
    - Data rate correct?
    - Correct Manchester coding?
- Check interrupts, Read IS register using the TDA523x Explorer to identify interrupts.
  - There is no wake up interrupt
    - Is definition of wake up pattern correct?
    - Data rate correct?
    - Correct Manchester coding?

        - Check RSSI as below
    - There is no FSYNC (synchronization) interrupt
        - TSI in IAF not correct
    - There is no EOM (end of message) interrupt
        - MID scanning accidentally enabled
    - EOM interrupt immediately after FSYNC interrupt
        - EOM criterion Data length set to 0 bits
- Check RSSI. Is the data pattern visible on RSSI for ASK? Is there a stable increased voltage during transmission in FSK?
    - Is the modulation correct?
        - Check modulation setting in IAF and at transmitter
    - Is there any received signal?
        - No RF transmission
        - Channel settings incorrect or transmission is at wrong frequency
    - Is it possible to identify the transmitted data in ASK?
        - Data rate, Manchester coding correct?
        - Wake up pattern, TSI, payload correct?

10 Software Implementation Hints

TDA523x is really easy to use.

See in the following flowchart, how a typical SW implementation looks like:



11 Configuration File

The configuration file generated by the IAF Tool looks like:

- The first column has historical reasons and the “0x02” is a register write.
- The second column is the register address.
- And the third column is the data to be written into the register.

This is the file which is used by the TDA523x Explorer for configuration.

When implementing configuration into the host microcontroller software, one after another register has to be written according the configuration file. To verify writing, the address- and data trace registers can be read after each register access. For details see also the TDA523x Data Sheet.

As you can see, the register 0x02 (CMC0) is the first but also the last register. When setting registers, the TDA523x should be in Sleep mode. Therefore the first access to CMC0 is to switch to Sleep, the last CMC0 access switches to the desired operating mode as set in the IAF tool.

```
//Fsys=13.225625 MHz, 433.92 MHz, A: 10 kBit/s, 200 KHz
0x02    0x02    0x00    //CMC0: Chip Mode Control Register 0
0x02    0x03    0x08    //CMC1: Chip Mode Control Register 1
0x02    0x05    0x00    //IM: Interrupt Mask Register
0x02    0x07    0x01    //SPMC:  Self Polling Mode Control Register
0x02    0x08    0x01    //SPMRT:  Self Polling Mode Reference Timer
0x02    0x09    0xA8    //SPMOFFT0: Self Polling Mode Off Time Register0
0x02    0x0A    0x07    //SPMOFFT1: Self Polling Mode Off Time Register1
0x02    0x0B    0x01    //SPMAP: Self Polling Mode Active Periods Register
0x02    0x0C    0x01    //SPMIP: Self Polling Mode Idle Periods Register
0x02    0x12    0x00    //RFC: RF Control Register
0x02    0x13    0x00    //CLKOUT0: Clock Divider Register0
0x02    0x14    0x00    //CLKOUT1: Clock Divider Register1
0x02    0x15    0x00    //CLKOUT2: Clock Divider Register2
0x02    0x16    0x00    //LOC: Local Oscillator Control Register
0x02    0x1B    0x00    //LIMC0: Trim RSSI Gain
0x02    0x1C    0x10    //LIMC1: Trim RSSI Offset, enable RSSI pin
0x02    0x1F    0xB0    //ASPMONT0:  Conf.A Self Polling Mode On Time Register0
0x02    0x20    0x01    //ASPMONT1:  Conf.A Self Polling Mode On Time Register1
0x02    0x21    0x05    //AMT: Conf.A Modulation Type Register
0x02    0x22    0x29    //ARFPLL1: Conf.A RF PLL setting, channel 1 (SlaveMode & Self Polling Mode)
0x02    0x23    0x09    //ARFPLL2: Conf. ARF PLL setting, channel 2 (SelfPolling Mode)
0x02    0x24    0x09    //ARFPLL3: Conf.A RF PLL setting, channel 3 (SelfPolling Mode)
0x02    0x25    0x02    //AWUC: Conf.A Wake up Control Register
0x02    0x26    0x00    //AWUPAT0: Conf.A Wake Up Detection Pattern0
0x02    0x27    0x00    //AWUPAT1: Conf.A Wake Up Detection Pattern1
0x02    0x28    0x0C    //AWUBCNT: Conf.A Wake Up Bit Count Register
0x02    0x29    0x00    //AMID0: Conf.A Message ID Register0
0x02    0x2A    0x00    //AMID1: Conf.A Message ID Register1
0x02    0x2B    0x00    //AMID2: Conf.A Message ID Register2
0x02    0x2C    0x00    //AMID3: Conf.A Message ID Register3
0x02    0x2D    0x00    //AMID4: Conf.A Message ID Register4
0x02    0x2E    0x00    //AMID5: Conf.A Message ID Register5
0x02    0x2F    0x00    //AMID6: Conf.A Message ID Register6
0x02    0x30    0x00    //AMID7: Conf.A Message ID Register7
0x02    0x31    0x00    //AMID8: Conf.A Message ID Register8
0x02    0x32    0x00    //AMID9: Conf.A Message ID Register9
0x02    0x33    0x00    //AMID10: Conf.A Message ID Register10
0x02    0x34    0x00    //AMID11: Conf.A Message ID Register11
0x02    0x35    0x00    //AMID12: Conf.A Message ID Register12
0x02    0x36    0x00    //AMID13: Conf.A Message ID Register13
0x02    0x37    0x00    //AMID14: Conf.A Message ID Register14
0x02    0x38    0x00    //AMID15: Conf.A Message ID Register15
```

```
0x02    0x39    0x00    //AMID16: Conf.A Message ID Register16
0x02    0x3A    0x00    //AMID17: Conf.A Message ID Register17
0x02    0x3B    0x00    //AMID18: Conf.A Message ID Register18
0x02    0x3C    0x00    //AMID19: Conf.A Message ID Register19
0x02    0x3D    0x00    //AMIDC0: Conf.A Message ID Control Register0
0x02    0x3E    0x00    //AMIDC1: Conf.A Message ID Control Register1
0x02    0x3F    0x00    //AIF0: Conf.A IF Buffer Amplifier Enable
0x02    0x40    0x01    //BSPMONT0:  Conf.B Self Polling Mode On Time Register0
0x02    0x41    0x00    //BSPMONT1:  Conf.B Self Polling Mode On Time Register1
0x02    0x42    0x04    //BMT: Conf.B Modulation Type Register
0x02    0x43    0x29    //BRFPLL1: Conf.B RF PLL setting, channel 1 (SlaveMode & Self Polling Mode)
0x02    0x44    0x09    //BRFPLL2: Conf.B RF PLL setting, channel 2 (Self Polling Mode)
0x02    0x45    0x09    //BRFPLL3: Conf.B RF PLL setting, channel 3 (SelfPolling Mode)
0x02    0x46    0x03    //BWUC: Conf.B Wake up Control Register
0x02    0x47    0x00    //BWUPAT0: Conf.B Wake Up Detection Pattern0
0x02    0x48    0x00    //BWUPAT1: Conf.B Wake Up Detection Pattern1
0x02    0x49    0x00    //BWUBCNT: Conf.B Wake Up Bit Count Register
0x02    0x4A    0x00    //BMID0: Conf.B Message ID Register0
0x02    0x4B    0x00    //BMID1: Conf.B Message ID Register1
0x02    0x4C    0x00    //BMID2: Conf.B Message ID Register2
0x02    0x4D    0x00    //BMID3: Conf.B Message ID Register3
0x02    0x4E    0x00    //BMID4: Conf.B Message ID Register4
0x02    0x4F    0x00    //BMID5: Conf.B Message ID Register5
0x02    0x50    0x00    //BMID6: Conf.B Message ID Register6
0x02    0x51    0x00    //BMID7: Conf.B Message ID Register7
0x02    0x52    0x00    //BMID8: Conf.B Message ID Register8
0x02    0x53    0x00    //BMID9: Conf.B Message ID Register9
0x02    0x54    0x00    //BMID10: Conf.B Message ID Register10
0x02    0x55    0x00    //BMID11: Conf.B Message ID Register11
0x02    0x56    0x00    //BMID12: Conf.B Message ID Register12
0x02    0x57    0x00    //BMID13: Conf.B Message ID Register13
0x02    0x58    0x00    //BMID14: Conf.B Message ID Register14
0x02    0x59    0x00    //BMID15: Conf.B Message ID Register15
0x02    0x5A    0x00    //BMID16: Conf.B Message ID Register16
0x02    0x5B    0x00    //BMID17: Conf.B Message ID Register17
0x02    0x5C    0x00    //BMID18: Conf.B Message ID Register18
0x02    0x5D    0x00    //BMID19: Conf.B Message ID Register19
0x02    0x5E    0x00    //BMIDC0: Conf.B Message ID Control Register0
0x02    0x5F    0x00    //BMIDC1: Conf.B Message ID Control Register1
0x02    0x60    0x00    //BIF0: Conf.B IF Buffer Amplifier Enable, B
0x02    0x61    0x10    //XTALCAL0:Trim XTAL frequency, coarse
0x02    0x62    0x00    //XTALCAL1: Trim XTAL frequency, fine
0x02    0x6B    0x27    //TOTIM: Time Out Timer Register
0x02    0x6C    0x00    //ADIGRXC: Conf.A Global Settings
0x02    0x6D    0x52    //ADCSPLRDIV: Conf.A ADC dividing factor
0x02    0x6E    0x00    //APKBITPOS: Conf.A RSSI Detector Start-up Delay
0x02    0x6F    0x06    //ADATFILT0: Conf.A Matched Filter Scaling and Delay
0x02    0x70    0x00    //ADATFILT1: Conf.A Matched Filter Decimation
0x02    0x71    0x00    //ASIGDET0: Conf.A Signal detector (Run Mode)
0x02    0x72    0x00    //ASIGDET1: Conf.A Signal detector (wake up)
0x02    0x73    0xE6    //ACDR0: Conf.A Clock recovery P parameters
0x02    0x74    0x65    //ACDR1: Conf.A Clock recovery I parameters
0x02    0x75    0x01    //ACDR2: Conf.A Clock recovery RUNIN length
0x02    0x76    0x87    //ASYSRCT0: Conf.A Synchronization search time out
0x02    0x77    0x28    //ATVWIN: Conf.A CV Window Length
0x02    0x78    0xF3    //AFSKNCO0: Conf.A FSK DDS NCO Frequency Offset
0x02    0x79    0x1C    //AFSKNCO1: Conf.A FSK DDS NCO Frequency Offset
0x02    0x7A    0x0F    //AFSKNCO2: Conf.A FSK DDS NCO Frequency Offset
0x02    0x7B    0x01    //AFSKFILBW0: Conf.A FSK Pre Filter Decimation
0x02    0x7C    0x68    //AFSKFILBW1: Conf.A FSK Pre Filter Scaling
0x02    0x7D    0x04    //AFSKDEMBW0: Conf.A FSK Demodulator Sensitivity
0x02    0x7E    0x28    //AFSKDEMBW1: Conf.A FSK DAM Output Decimation
0x02    0x7F    0x05    //AFSKDEMBW2: Conf.A FSK DAM Output Scaling
0x02    0x80    0x00    //ANDTHRES: Conf.A FSK Noise Detector Threshold
0x02    0x81    0x37    //ANDCONFIG: Conf.A FSK Noise Detector configuration
0x02    0x82    0x00    //ATSIMODE: Conf.A TSI Detection Mode
0x02    0x83    0x10    //ATSILENA: Conf.A TSI A Length
0x02    0x84    0x08    //ATSILENB: Conf.A TSI B Length
0x02    0x85    0x00    //ATSIGAP: Conf.A TSI GAP
0x02    0x86    0x65    //ATSIPTA0: Conf.A TSI Data Reference Low Byte A
0x02    0x87    0xAA    //ATSIPTA1: Conf.A TSI Data Reference High Byte A
0x02    0x88    0x55    //ATSIPTB0: Conf.A TSI Data Reference Low Byte B
0x02    0x89    0x00    //ATSIPTB1: Conf.A TSI Data Reference High Byte B
```

```
0x02    0x8A    0x01    //AEOMC: Conf.A EOM Control
0x02    0x8B    0x80    //AEOMDTLEN: Conf.A EOM Data Length Limit
0x02    0x8C    0x06    //BDIGRXC: Conf.B Global Settings
0x02    0x8D    0x00    //BDCSPLRDIV: Conf.B ADC dividing factor
0x02    0x8E    0x00    //BPKBITPOS: Conf.B RSSI Detector Start-up Delay
0x02    0x8F    0x06    //BDATFILT0: Conf.B Matched Filter Scaling and Delay
0x02    0x90    0x00    //BDATFILT1: Conf.B Matched Filter Decimation
0x02    0x91    0x00    //BSIGDET0: Conf.B Signal detector (Run Mode)
0x02    0x92    0x00    //BSIGDET1: Conf.B Signal detector (wake up)
0x02    0x93    0xE6    //BCDR0: Conf.B Clock recovery P parameters
0x02    0x94    0x65    //BCDR1: Conf.B Clock recovery I parameters
0x02    0x95    0x01    //BCDR2: Conf.B Clock recovery RUNIN length
0x02    0x96    0x87    //BSYSRCT0: Conf.B Synchronization search time out
0x02    0x97    0x00    //BTVWIN: Conf.B CV Window Length
0x02    0x98    0xF3    //BFSKNCO0: Conf.B FSK DDS NCO Frequency Offset
0x02    0x99    0x1C    //BFSKNCO1: Conf.B FSK DDS NCO Frequency Offset
0x02    0x9A    0x0F    //BFSKNCO2: Conf.B FSK DDS NCO Frequency Offset
0x02    0x9B    0x01    //BFSKFILBW0: Conf.B FSK Pre Filter Decimation
0x02    0x9C    0x68    //BFSKFILBW1: Conf.B FSK Pre Filter Scaling
0x02    0x9D    0x04    //BFSKDEMBW0: Conf.B FSK Demodulator Sensitivity
0x02    0x9E    0x00    //BFSKDEMBW1: Conf.B FSK DAM Output Decimation
0x02    0x9F    0x00    //BFSKDEMBW2: Conf.B FSK DAM Output Scaling
0x02    0xA0    0x00    //BNDTHRES: Conf.B FSK Noise Detector Threshold
0x02    0xA1    0x07    //BNDCONFIG: Conf.B FSK Noise Detector configuration
0x02    0xA2    0x00    //BTSIMODE: Conf.B TSI Detection Mode
0x02    0xA3    0x00    //BTSILENA: Conf.B TSI A Length
0x02    0xA4    0x00    //BTSILENB: Conf.B TSI B Length
0x02    0xA5    0x00    //BTSIGAP: Conf.B TSI GAP
0x02    0xA6    0x00    //BTSIPTA0: Conf.B TSI Data Reference Low Byte A
0x02    0xA7    0x00    //BTSIPTA1: Conf.B TSI Data Reference High Byte A
0x02    0xA8    0x00    //BTSIPTB0: Conf.B TSI Data Reference Low Byte B
0x02    0xA9    0x00    //BTSIPTB1: Conf.B TSI Data Reference High Byte B
0x02    0xAA    0x00    //BEOMC: Conf.B EOM Control
0x02    0xAB    0x00    //BEOMDTLEN: Conf.B EOM Data Length Limit
0x02    0xB4    0xB2    //APSLC: Conf.A Pre Slicer Control
0x02    0xB5    0xB2    //BPSLC: Conf.B Pre Slicer Control
0x02    0xB6    0x00    //ASIGDETLO: Conf.A Signal Detector Threshold Low Level
0x02    0xB7    0x00    //BSIGDETLO: Conf.B Signal Detector Threshold Low Level
0x02    0xB8    0x0A    //ASIGDETSEL: Conf.A Signal Detector Factor selection
0x02    0xB9    0x0A    //BSIGDETSEL: Conf.B Signal Detector Factor selection
0x02    0x02    0x21    //CMC0: Chip Mode Control Register 0
```

http://www.infineon.com

Published by Infineon Technologies AG