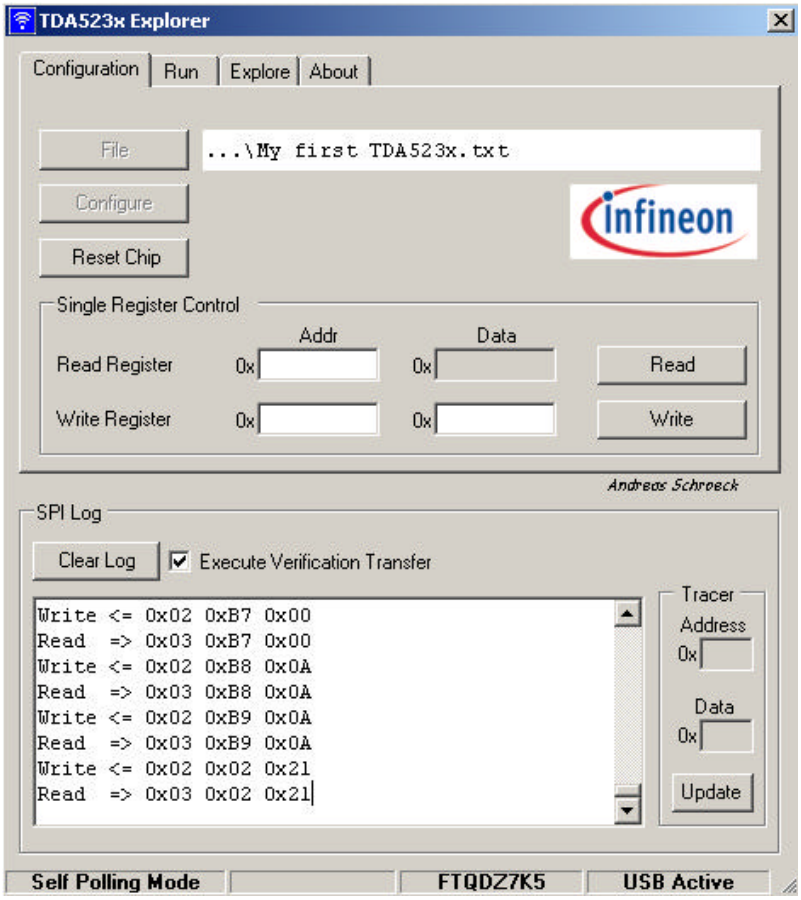
TDA523x Explorer

Documentation V2.2



Index

INTRODUCTION **1**

INSTALLATION **1**

HARDWARE REQUIREMENTS 1
SOFTWARE REQUIREMENTS 1
UPDATE FROM PREVIOUS VERSIONS 1
INSTALLATION PROCESS 1

FEATURES **2**

TAB BASED SECTION **2**

CONFIGURATION – TAB 2
Write Single Register: *3*
Read Single Register: *3*
Valid Configuration File: *4*
Correct Configuration (Power Supply): *5*
Stay Connected: *5*
RUN – TAB 5
Change Operating Mode *6*
Change Interrupt Mask Register *6*
Visualize Interrupts and Received Messages. *6*
EXPLORE – TAB 7
Noise Power Measurement: *7*
Statistic Measurements *8*
FIFO Content *8*
Auto Update *8*
ABOUT – TAB 9

SPI LOG SECTION **9**

Execute Verification Transfer *10*
Tracer Data *10*

STATUS BAR SECTION **10**

HARDWARE DESCRIPTION / FTDI2232C 10

Introduction

The TDA523x Explorer is an interface application for developers and customers of Infineon Technologies. It offers the functionality to download configuration files to the TDA523x, to manipulate single configuration registers of the TDA523x and to visualize basic chip activity. The communication is established via an USB controller which converts the commands from the application into a TDA523x compatible protocol.

Installation

Hardware Requirements

To guarantee a working installation the following hardware is recommended:

- USB 1.1/2.0
- 128MB RAM
- Infineon TDA523x Evaluation Board
- USB cable A/B male/male

Software Requirements

To run the TDA523x Explorer Software following software must be installed to your PC:

- Microsoft Windows 2000/XP/Vista
- .Net Framework 1.1
- FTDI USB Controller Driver (Direct Driver, installed during setup process)

Remark: Make sure that you have administrative privileges on your computer to install the software. If you do not have these rights, please contact your system administrator to install the software.

Update from Previous Versions

If you have already installed an older version of the TDA523x Explorer, it is recommended to update also the USB drivers. The latest release of USB drivers is included in the TDA523x Explorer setup. Install the TDA523x Explorer as usual and delete all installed FTDI drivers. A complete system clean from FTDI drivers can be done manually or by the included removal tool **FTClean.** After completing the installation process a shortcut to the tool is placed in the Programs menu under **Infineon Technologies.** Follow the instructions **on the screen** and close the application after cleanup is completed. Now plug-in the TDA523x Evaluation Board and use the driver included in the installation package.

Installation Process

To install the TDA523x Explorer, run the *Setup.exe* file and follow the instructions on the screen. Do not plug hardware before the setup is finished. After finishing the setup, connect the TDA523x Evaluation Board to the USB interface. The operating system will

automatically start the hardware installation wizard. Follow the step by step instructions on the screen until the wizard asks for driver files. The driver can be found in the TDA523x Explorer installation package. After finishing the driver's installation the TDA523x Explorer is ready to use.

Features

The application offers various complex features to configure, manipulate and visualize the behavior of the TDA523x Receiver IC. To offer a simple and user friendly interface the application is divided into several different sections. In these sections functions which belong together logically, are combined to a visual unit. The three main sections consist of a TAB based section, the SPI Log section and the status bar section. All these section provide an Auto Update function to keep key values actual.

TAB Based Section

Nearly all features for interaction and communication can be found in the TAB based section. The TAB's are organized in a logical sequence following a typical communication process between the user and the TDA523x. This sequence consists of 3 basic steps:

1. Select a configuration file for the TDA523x within the *Open-File Dialog* and download this configuration to the IC on the *"Configuration-TAB"*.
2. Interact and visualize the behavior of the IC with its actual configuration mode on the *"Run-TAB"*.
3. Read status information and several important register values on the *"Explore TAB"*. It is also possible to do some measurements (e.g. ASK noise power) for fine adjustments of the TDA523x Receiver.

Configuration – TAB

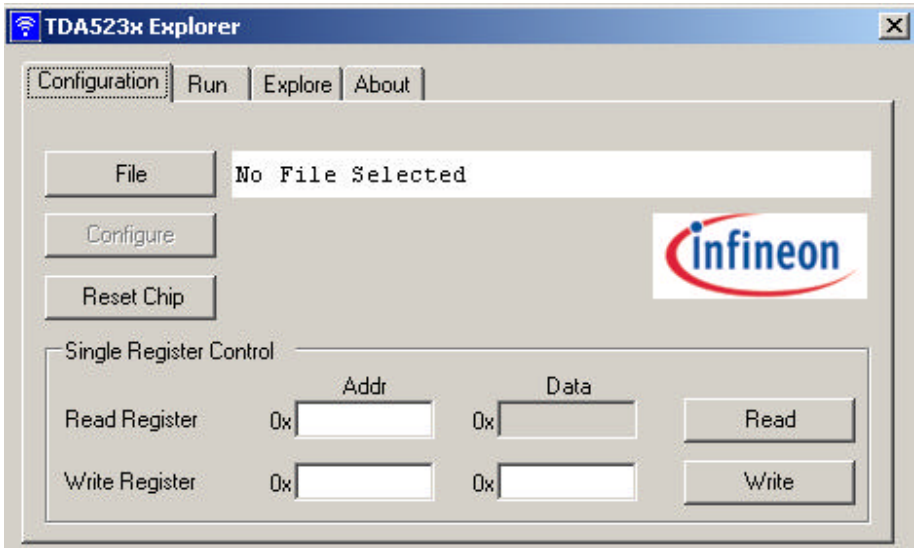


Figure 1: initial view of the configuration TAB – no configuration selected

The Configuration TAB is active on program start-up. This TAB provides a very simple but powerful interface to select configuration files and to configure the TDA523x Receiver IC.
To configure the TDA523x the user has to ensure that following requirements are met.

- The TDA5230 must be in Reset Mode
- A valid configuration file must be selected
- The USB cable must stay connected during the whole configuration process.

If these requirement are fulfilled the configuration process can be started. The "*Open-File Dialog"*, for configuration file selection, can be accessed with the *"File"* button. The filename and the system path of the selected file are displayed within the application window.
If the user aborts the selection process the caption "*No file selected*" can be found on the File label. This text is also displayed, if any error occurs during configuration file I/O. After the file is validated by the application the configuration button gets enabled. During and after the configuration the configure button is disabled to prevent a re-configuration while the chip is running. Every command gets logged to give the user the ability to control the progress of the configuration process in the SPI Log section.
To reactivate the configuration button the user has to *reset* the TDA523x. After that a re-configuration is possible without reselecting of the configuration file. The configuration data stays valid until a new configuration file gets selected over the Open-File Dialog or the application gets restarted.
The user has also the ability to read/write single registers. With this option it is possible to check the actual configuration and to adjust it without reconfiguring the whole chip.

Write Single Register:

To write a register the address and the data value must be placed into the labeled fields. The values of the address and the data value must be given in a hexadecimal format. Any mismatch will be detected and the user gets an error message with an example for a valid input. The value for the address and the data byte must be between *0x00* and *0xFF*. This interaction is also logged within the SPI log section.

Remark: The "Chip Mode Control (CMC0)" Register (address: *0x02)* cannot be written with this function, this has to be done via the configuration file.

Warning: Make sure that the address and the data are valid for the TDA523x and that the register can be written without damaging the chip. There is **no** check within the application.

Read Single Register:

To read a register only the address is needed. The value of the address must be given in a hexadecimal format. The register address value must be between *0x00* and *0xFF*. Make sure that the register is readable. Not all registers of the TDA523x can be read out. If the register cannot be read out, the application will display *0x00*.

Reset Mode:

Is one of the three possible chip modes of the TDA523x Receiver. The actual mode can be controlled by the Run – TAB or by the *"Reset"* Button on the configuration TAB[1]. The current operation chip mode of the chip can be found in the status bar of the TDA Explorer.

Valid Configuration File:

The configuration file can be selected via a standard Windows® *Open-File Dialog*. There are some verification mechanisms which are used to check the syntax of the configuration file. Figure 2 is an excerpt of a valid configuration file for a TDA5230 Receiver. A valid configuration file has to fulfill following criteria:

- The whole command must be in one line.
- Every additional command with in one line is skipped.
- Comments, if needed, can only be added at the end of the command.
- Each command element (command, address and data) must be separated with one single tabulator symbol The address and the data field are not mandatory for a valid command.

```
// C1-Chip,  Pattern=TestProtocol1,  2kBit/s,  Slave Mode,  fSYS=13.225625MHz,  OPM(TME=TSE=0),  FSK-IF-BW=+/-250kHz,
0x02    0x02    0x00    // CMC0:  ConfigA, Sleep Mode (necessary for XTAL trimming)
0x02    0x12    0x00    // RFC:  RF gain adjust - IF attenuation
0x02    0x1B    0x00    // LIMC0:  Limiter RSSI Gain (necessary for ASK only)
0x02    0x1C    0x10    //    LIMC1:  Limiter RSSI Offset (necessary for ASK only)
0x02    0x21    0x04    // AMT:  1Channel:   100:ASK, 101:FSK, 110:FSK-ASK, 111: ASK-FSK
0x02    0x22    0x29    // ARFPLL1:  Band selection, R, S (315MHz: 0x09,  434MHz: 0x29,  868MHz: 0x49,  915MHz: 0x69)
0x02    0x61    0x10    // XTALCAL0:  XTAL trimm coarse - chip internal (@ 3.9pF ext  and  CL=12pF)
0x02    0x6C    0x06    // ADIGRX:  invert polarity + ADC-AntiAliasingFilter
0x02    0x6D    0x32    // ADCSPLRDIV:  ADC dividing factor
0x02    0x6F    0x06    // ADATFILT0:  scaling  and  (delay=RegVal+1)
0x02    0x70    0x07    // ADATFILT1:  decimation factor=RegVal+1
0x02    0x71    0x0C    //    ASIGDET0:  signal threshold  -  TSI-Pattern  --  Evaluate by Software !!!
0x02    0x72    0x0C    //    ASIGDET1:  signal threshold  -  WU-Pattern  --  Evaluate by Software !!!
0x02    0x73    0xE6    // ACDR0:  CDR - DPLL P parameter
0x02    0x74    0x65    // ACDR1:  CDR - DPLL I parameter
0x02    0x75    0x01    // ACDR2:  CDR RUNIN pattern length
0x02    0x76    0x6C    // ASYSRCT0:  sync. Search time out
0x02    0x77    0x28    // ATVIWIN:  CDR CV window / ssync mode
0x02    0x78    0xF3    // AFSKNCO0:  NCO offset freq. [MSB]
0x02    0x79    0x1C    // AFSKNCO1:  NCO offset freq.
0x02    0x7A    0x0F    // AFSKNCO2:  NCO offset freq. [LSB]
0x02    0x7B    0x01    // AFSKFILBW0:  Decimation=RegVal+1
0x02    0x7C    0x68    // AFSKFILBW1:  CIC-Shift=RegVal[6:4];CIC-Delay=RegVal[3:0]+1;
0x02    0x7D    0x04    // AFSKDEMBW0:  FSK-Demodulator-Delay=RegVal[3:0]+1
0x02    0x7E    0x19    // AFSKDEMBW1:  I&D-Decimation=RegVal+1
0x02    0x7F    0x04    // AFSKDEMBW2:  I&D-Shift=RegVal
0x02    0x80    0x00    //    ANDTHRES:  FSK Noise detector Threshold  --  Evaluate by Software !!!
0x02    0x81    0x37    // ANDCONFIG:  Configuration
0x02    0x82    0x00    // ATSISETMOD:  TSI Detection-mode Register
0x02    0x83    0x10    // ATSILENA:  TSI length
0x02    0x84    0x10    // ATSILENB:  TSI length
0x02    0x85    0x00    // ATSIGAP:  TSI gap
0x02    0x86    0x56    // ATSIPTA0:  TSI Data Reference Low Byte Register
0x02    0x87    0x55    // ATSIPTA1:  TSI Data Reference High Byte Register
```

Figure 2: excerpt of a configuration file for the TDA523x

Remark: The configuration file must contain an initialization value for the "Interrupt Mask Register (IMR)" (address: *0x05*). If there is no entry for the IMR the TDA523x Explorer V2.1 and above will initialize the register with the value *0x55*.

[1] The Reset Button generates a min. 10ms long reset pulse which deletes all data on the TDA523x and restarts the device.

Correct Configuration (Power Supply):

To ensure the correct function of the TDA523x Evaluation Board has to choose either if the board is USB powered or self powered. The default configuration is self powered, but can be changed via some soldered jumpers.

Warning: Invalid power settings can lead to a malfunction or destruction of the hardware.

Stay Connected:

The USB connection between Systemboard and PC must be guaranteed over the whole up-time of the application. If the connection gets disabled it could be possible that the USB Controller looses its configuration[2]. Also the configuration of the TDA523x is lost, if the system is not self powered. The connection status of the USB Controller is shown in the status bar.

Run – TAB

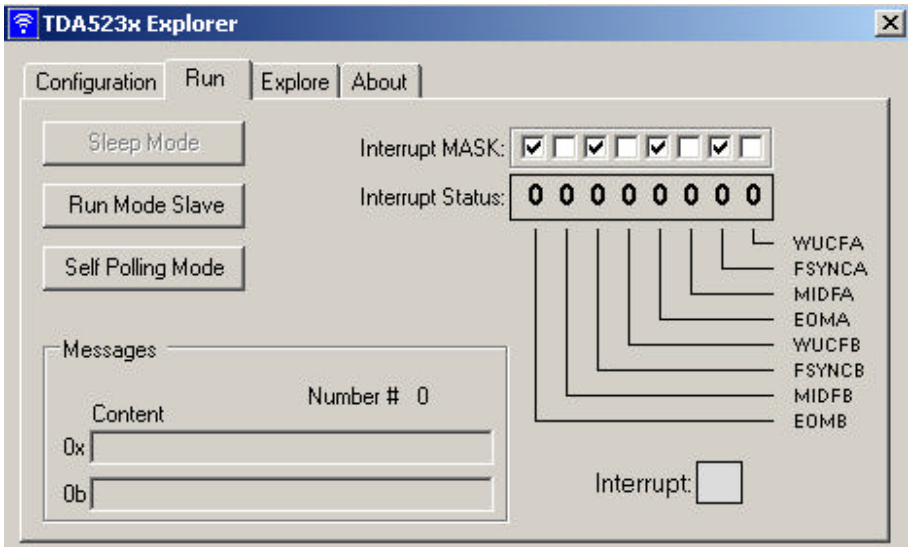


Figure 3: initial view of TAB Run – device is not programmed

The Run TAB consists out of three sub sections. These three sections are logical groups which belong to the following functions:

- Change TDA523x Operating Mode
- Change Interrupt Mask Register
- Visualize Interrupts and received messages

[2] The loss of the configuration may lead to an incorrect function of the controller and to a reinstallation of the driver.

All operations which can be executed on this TAB require a correct and fully functional configured TDA523x to take effect. The application will not check the configuration of the TDA523x.

Change Operating Mode

The TDA523x Explorer offers a three button interface to change the operating mode of the TDA523x. The available modes are depending on the operating mode in which the chip is actually situated. If the chip is in one of the working modes, either in

- *Run Mode Slave* or in
- *Self Polling Mode*

The only possible mode to switch to is the *Sleep Mode*. Once the TDA523x is in Sleep Mode the buttons for the working modes will get enabled and the Sleep Mode button gets disabled. This mechanism prevents the chip of getting damaged.
When chip mode is changed, then the lowest two bits of register *CMC0* (address *0x02*) are updated as shown in the following table:

Mode activating	***Bits changed in register***	***Effected register***
Sleep Mode	XXXX XX00	CMC0
Run Mode Slave	XXXX XX10	CMC0
Self Polling Mode	XXXX XX01	CMC0

Change Interrupt Mask Register

Every change of the "*checkboxes*" state leads to a change of the Interrupt Mask Register of the TDA523x. If an interrupt appears, the application signals the interrupt with the Interrupt LED which will flash up for a short time. If there is another interrupt within the flash up time the LED will shine longer. All changes in the IMR are logged as a write command in the SPI Log section.

Visualize Interrupts and Received Messages.

Every time a complete message is received from the TDA523x the message number and the message content will be displayed in the messages area of the TAB. If the content of the Interrupt Status Register does not show an EOM interrupt, the FIFO content will not be read out and the message will not be displayed. Only the Interrupt LED will signalize that something is going on.
The message content is displayed in two different ways. On one hand as a hexadecimal formatted byte oriented string and on the other hand as a binary formatted byte oriented string. The two labels for the visualization are shown in Figure 3.

Complete Message:

A complete message will be detected every time an EOM interrupt occurs on the interrupt pin. These interrupts can occur on both communication channels of the TDA523x.

Interrupt LED:

The interrupt LED is a notification element for every occurring interrupt on the interrupt pin. The interrupt LED lights up on each recognized interrupt for a time of 40 ms. The light up time will be extended for another 40 ms if the application detects another interrupt before the 40 ms time span is over.

Explore – TAB

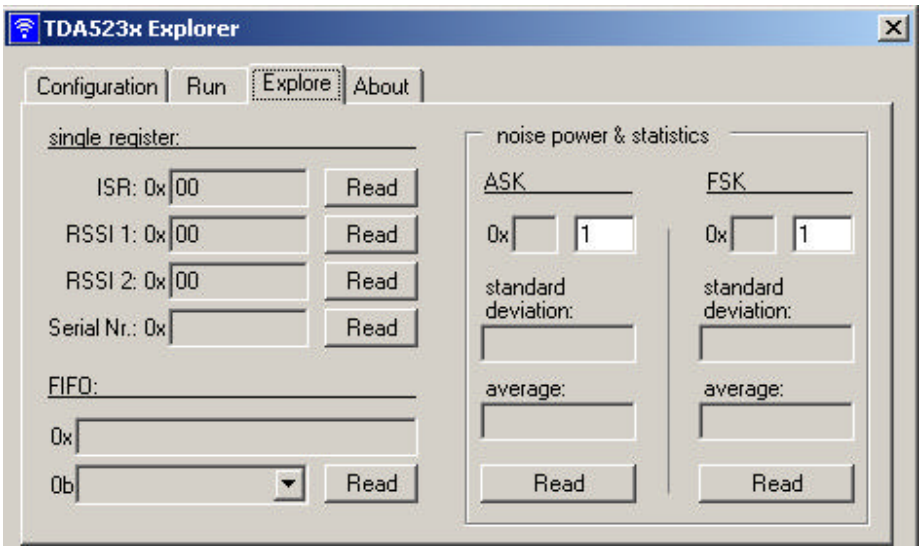


Figure 4: Explore TAB initial view

The purpose of the Explore – TAB is to read a predefined set of registers. This set consists out of the following registers.

- ISR: Interrupt Status Register
- RSSI 1: signal power received from the TDA523x
- RSSI 2: signal power received from the TDA523x
- ASKNP: actual signal power detected from the TDA523x (ASK or FSK)
- FSKNP: actual noise power detected from the TDA523x (FSK)
- Serial Number: unique TDA523x chip serial number
- FIFO Content: actual content of the TDA5230 internal FIFO memory.

A more precisely description of the different registers can be found in the TDA523x data sheet. All registers can be manually updated via the *"Read"* button. The content of the registers will be displayed as a hexadecimal formatted string.

Noise Power Measurement:

The noise power measurement is an additional feature to execute some fine adjustments of the TDA523x. Both measurements, the ASK and the FSK noise power, are an average over a number of measurements, which can be modified by the user. The number of measurements must be between 1 and 1000. The displayed number of measurements is decreasing during the process to observe the progress of the measurement. As long as the measurement is running the application is blocked. No interaction with the application is possible.

Statistic Measurements

In addition to the single ASKNP and FSKNP measurements, it is also possible to calculate average and standard deviation over a specified number of measurements. This feature is used to set Signal- and Noise Detector Thresholds of TDA523x. Please refer to the TDA523x Data Sheet for further information.

FIFO Content

The FIFO content is displayed in two different ways. In the textbox the hexadecimal content of the FIFO is displayed. The binary content is displayed below in the combo-box. It is converted into a byte oriented format. Every byte is placed into a new line to make it easier for the user to compare the received values with the expected values.

Auto Update

All TAB's have an Auto Update function which is needed to make sure that the values of the TDA523x Explorer match the internal values of the TDA5230. The update happens every time the user switches to a TAB and effects also the other sections. The Auto Update function does not update every value which is displayed in the application. The updated values can be found in the list below:

- Configuration – TAB
  - No Auto Update necessary
- Run – TAB
  - Actual operating mode of the TDA5230
  - Interrupt Mask Register value
  - Message area
- Explore – TAB
  - Interrupt Status Register value
  - RSSI 1 level
  - RSSI 2 level
- SPI Log section
  - Auto Update only if the verification transfer is activated
- Status bar
  - Actual operating mode of the TDA5230
  - Actual status of the USB controller

About – TAB

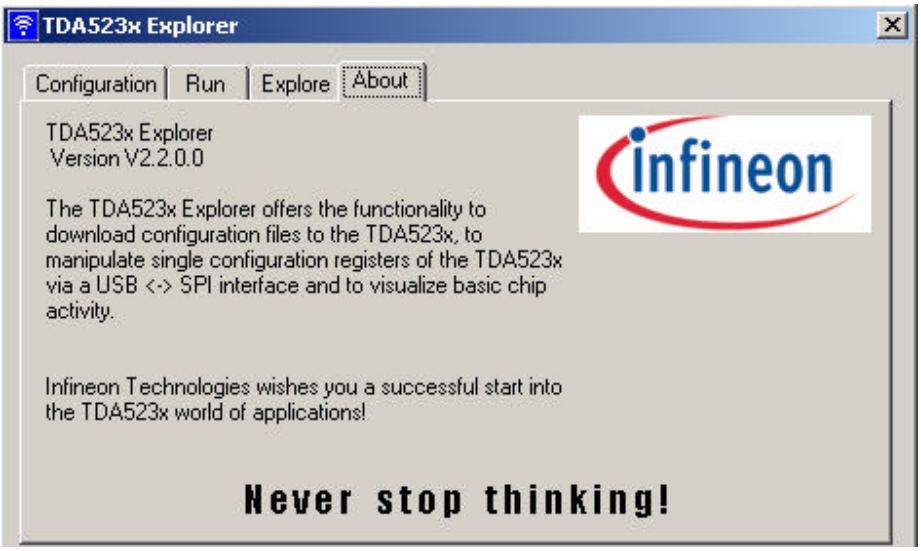


Figure 5: About TAB of the TDA523x Explorer

SPI Log Section

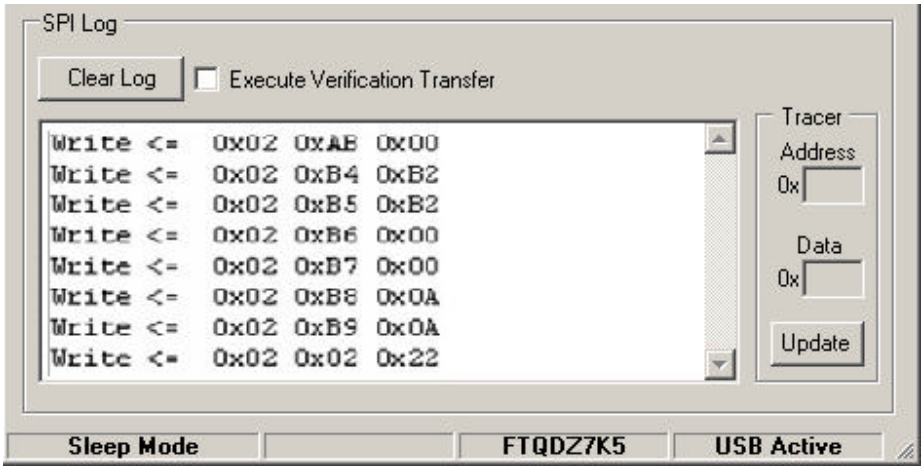


Figure 6: standard view after finished configuration

In the SPI log section every command sent via the SPI interface can be displayed. This log makes it easy to validate the consistence of the data within the application and the TDA523x. To view all transfers it is necessary to activate the Execute Verification Transfer function over the checkbox. The tracer data can be controlled over the update function in the Tracer Data area.
Also the FIFO data is displayed on every FIFO read command. The type of command which is logged is specified by the caption as shown below.

Command to log	*SPI log window*
Read register	Read => 0x03 0x01 0x01
Write register	Write <= 0x02 0x03 0x08
Read FIFO	FIFO := 0x02 0x02 …

The command caption is followed by the corresponding command value. After that the Address and the data value which was read or written are displayed. Only the FIFO data is formatted in another way. The FIFO data is displayed as a byte oriented hexadecimal data stream. If the FIFO data stream is too long for a single line, a new line is created to display the rest of the FIFO data stream.

Execute Verification Transfer

If enabled, this feature will execute an additional read command to get the chip internal tracer data values and compares the values of the address and the data values with the last command which was send by the application. If the data does not match no further commands will be executed e.g. during the configuration process. The user will be alerted with an error message in a messagebox.

Tracer Data

The TDA523x saves the last received command in two registers. These registers contain the address and the data of this command. The chip internal tracer will be displayed in a hexadecimal formatted string after the user executes the tracer update function.

Status Bar Section



The status bar shows basic status information of the USB controller and the TDA5230 Receiver IC. On the left side the operating mode in which the TDA5230 is actually is displayed. On the right side the status of the USB controller gets displayed. If the USB controller could get initialized successfully the caption “USB Active” appears in the status bar.

Hardware Description / FTDI2232C

The FTDI2232C is a dual-channel multi-purpose USB controller. The FT2232C offers the possibility to connect the TDA523x directly to the PC. To establish this communication the USB-controller must offer a possibility to convert the USB protocol into a TDA5230 conformable SPI protocol. This conversion is done by the internal M.P.S.S.E. (Multi-Protocol Synchronous Serial Engine). This unit can be configured to communicate with nearly any device using a synchronous protocol (such as JTAG, SPI, I2C …). This configuration is only available on *channel A*. *Channel B* can not be controlled over the M.P.S.S.E. . Also the modified SPI Protocol of the TDA5230 can be simulated. There are several commands which can be used to communicate with the TDA5230.

For example the “read” (0x20) command of the MPSSE has a variable byte length which is between 1 and 65536 bytes. This command can be used to simulate the normal read and the read FIFO command of the TDA5230. This variation of the read command can also be done with the write command. Furthermore it is possible to use the rising or the falling clock edge to clock in/out the bits/bytes on the data lines.