SmartLEWIS™ RX+

TDA5240/35
Using TDA5240 on TDA5230 UWLink Extension Board

Application Note 1.0,
2014-09-26

Wireless Control

Edition 2014-09-26

Published by Infineon Technologies AG
Am Campeon 1 - 12
85579 Neubiberg, Germany

© 2014 Infineon Technologies AG
All Rights Reserved.

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. 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) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

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.

Using TDA5240 on TDA5230 UWLink Extension Board
Revision History: 2014-09-26, V1.0

Previous Revision:

Page	Subjects (major changes since last revision)

Trademarks of Infineon Technologies AG

APOXI™, BlueMoon™, COMNEON™, CONVERGATE™, COSIC™, FALC™, GEMINAX™, GOLDMOS™, ISAC™, OmniTune™, OmniVia™, OPTIVERSE™, SCEPTRE™, SEROCCO™, SICOFI™, SMARTi™, SMINT™, SOCRATES™, TrueNTRY™, VINAX™, VINETIC™, X-GOLD™, XMM™, X-PMU™, XPOSYS™, XWAY™.

Other Trademarks

Microsoft®, Visio®, Windows®, Windows Vista®, Visual Studio®, Win32® of Microsoft Corporation. Linux® of Linus Torvalds. FrameMaker®, Adobe® Reader™, Adobe Audition® of Adobe Systems Incorporated. NEON™ of Comneon GmbH & Co. OHG. PrimeCell®, RealView®, ARM®, ARM® Developer Suite™ (ADS), Multi-ICE™, ARM1176JZ-S™, CoreSight™, Embedded Trace Macrocell™ (ETM), Thumb®, ETM9™, AMBA™, ARM7™, ARM9™, ARM7TDMI-S™, ARM926EJ-S™ of ARM Limited. AUDO™ of Audi. AUTOSTAR™ of Continental Teves AG & Co. oHG. OakDSPCore®, TeakLite® DSP Core, OCEM® of ParthusCeva Inc. IndoorGPS™, GL-20000™, GL-LN-22™ of Global Locate. mipi™ of MIPI Alliance. CAT-iq™ of DECT Forum. MIPS™, MIPS II™, 24KEc™, MIPS32®, 24KEc™ of MIPS Technologies, Inc. Texas Instruments®, PowerPAD™, C62x™, C55x™, VLYNQ™, Telogy Software™, TMS320C62x™, Code Composer Studio™, SSI™ of Texas Instruments Incorporated. Bluetooth® of Bluetooth SIG, Inc. IrDA® of the Infrared Data Association. Java™, SunOS™, Solaris™ of Sun Microsystems, Inc. Philips®, I2C-Bus® of Koninklijke Philips Electronics N.V. Epson® of Seiko Epson Corporation. Seiko® of Kabushiki Kaisha Hattori Seiko Corporation. Panasonic® of Matsushita Electric Industrial Co., Ltd. Murata® of Murata Manufacturing Company. Taiyo Yuden™ of Taiyo Yuden Co., Ltd. TDK® of TDK Electronics Company, Ltd. Motorola® of Motorola, Inc. National Semiconductor®, MICROWIRE™ of National Semiconductor Corporation. IEEE® of The Institute of Electrical and Electronics Engineers, Inc. Samsung®, OneNAND®, UtRAM® of Samsung Corporation. Toshiba® of Toshiba Corporation. Dallas Semiconductor®, 1-Wire® of Dallas Semiconductor Corp. ISO® of the International Organization for Standardization. IEC™ of the International Engineering Consortium. EMV™ of EMVCo, LLC. Zetex® of Zetex Semiconductors. Microtec® of Microtec Research, Inc. Verilog® of Cadence Design Systems, Inc. ANSI® of the American National Standards Institute, Inc. WindRiver® and VxWorks® of Wind River Systems, Inc. Nucleus™ of Mentor Graphics Corporation. OmniVision® of OmniVision Technologies, Inc. Sharp® of Sharp Corporation. Symbian OS® of Symbian Software Ltd. Openwave® of Openwave Systems, Inc. Maxim® of Maxim Integrated Products, Inc. Spansion® of Spansion LLC. Micron®, CellularRAM® of Micron Technology, Inc. RFMD® of RF Micro Devices, Inc. EPCOS® of EPCOS AG. UNIX® of The Open Group. Tektronix® of Tektronix, Inc. Inte® of Intel Corporation. Qimonda® of Qimonda AG. 1GOneNAND® of Samsung Corporation. HyperTerminal® of Hilgraeve, Inc. MATLAB® of The MathWorks, Inc. Red Hat® of Red Hat, Inc. Palladium® of Cadence Design Systems, Inc. SIRIUS Satellite Radio® of SIRIUS Satellite Radio Inc. TOKO® of TOKO Inc.

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

Last Trademarks Update 2009-02-27

Table of Contents

List of Figures

1 Introduction

This Application Note shows the reqired steps to modify a TDA5230 UWlink for usage with a TDA5240 (or TDA5235). These steps are illustrated in Figure 1 below.

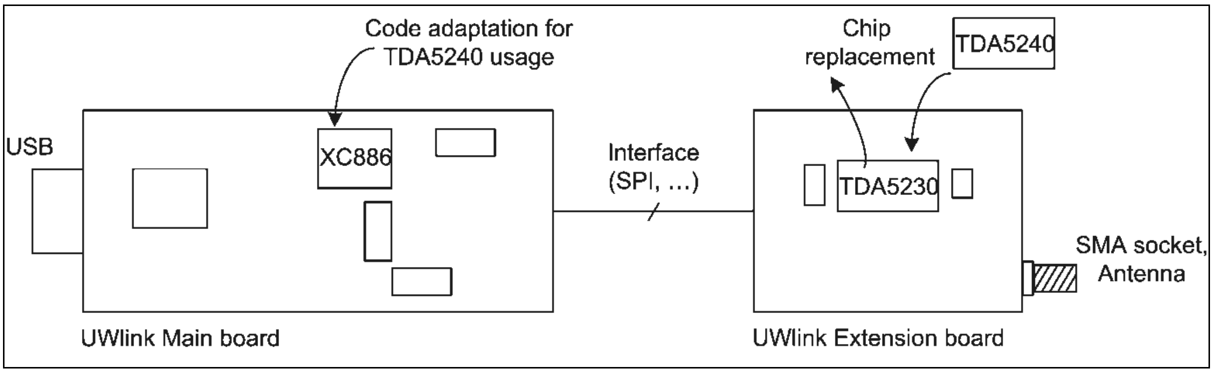

Figure 1 Conversion from TDA5230 to TDA5240/35

TDA5240 UWLink is used to show the operation of TDA5240 in 8 bit microcontroller environment. TDA5240 receiver IC is programmed with Infineon's XC886 microcontroller through SPI interface. The receiver IC is then ready for receiving TPMS or RKE signal. From TDA5240 IC, the received signal is read out by microcontroller and is displayed on Hyper-terminal on PC. Data payload and RSSI information are shown on Hyper-terminal.

This application note provides detailed information on hardware and Software designs of TDA5240 UWLink tool. Users can modify software or configuration registers for receiving their specific signal. Thus TDA5240 tool can be used for receiving signal in a moving car. The reference code can be used by unexperienced engineer in order to shorten development time.

2 Hardware Description

The hardware design is reused from TDA5230 USB UWLink board (Schematics and Layout can be seen in Figure 4 and Figure 5). The BOM (bill of material) for this adaptation to TDA5240/35 is given in Table 1 below. As RF performance is concerned, matching design is taken from TDA5230, which has similar LNA input impedance. So the matching network can be further optimized for usage with TDA5240 or TDA5235. RF SAW filter working at 434 MHz frequency band, is used for reducing RF interference outside frequency band.

Use of two IF filters is implemented on hardware design. Due to cost, a single IF filter is selectable in register setting. Both digital and analog RSSI signals are available for read out from the hardware. TDA5240 UWLink uses 3.3V supply voltage. Table 2 shows hardware connection between TDA5240 and microcontroller. Therefore signals can be measured on oscilloscope.

Figure 2 UWLink Demo Kit (Mainboard and Extension board) with adaptation for TDA5240

Figure 3 UWLink Extension board with adaptation for TDA5240

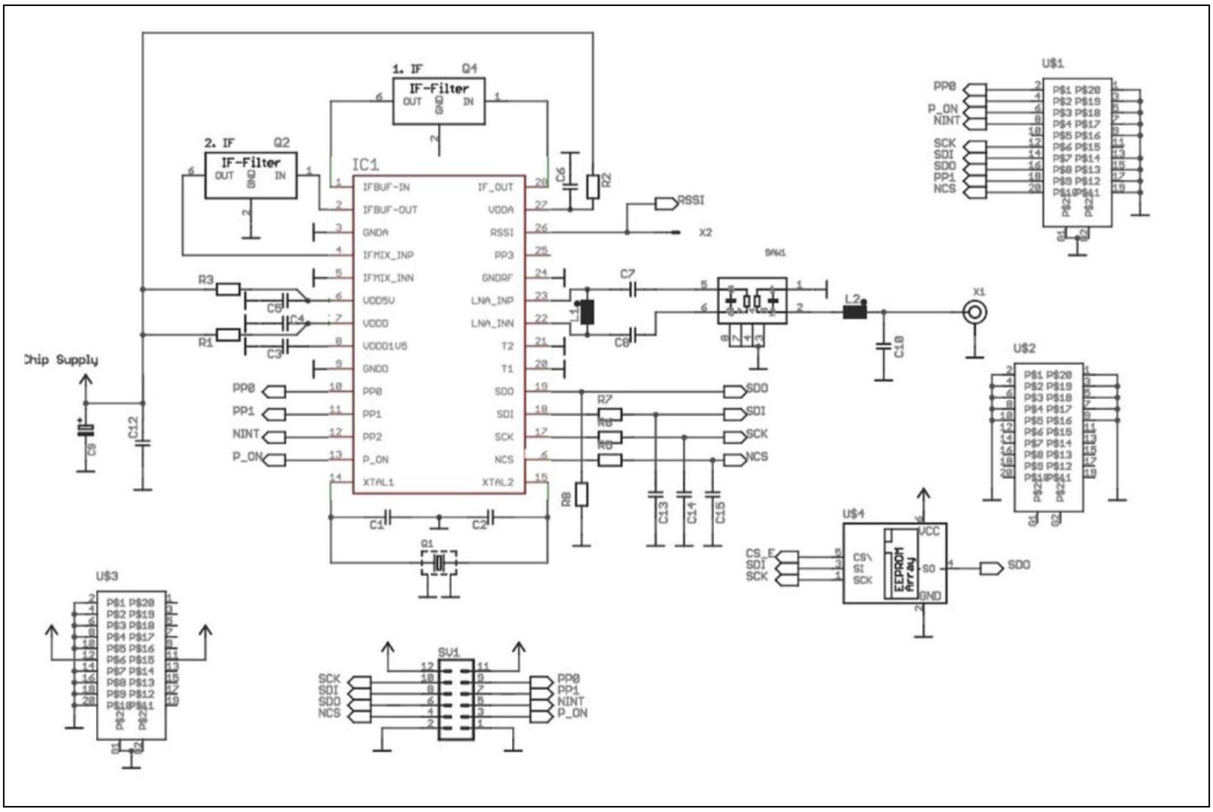

Figure 4 Schematics of UWlink Extension board with TDA5240

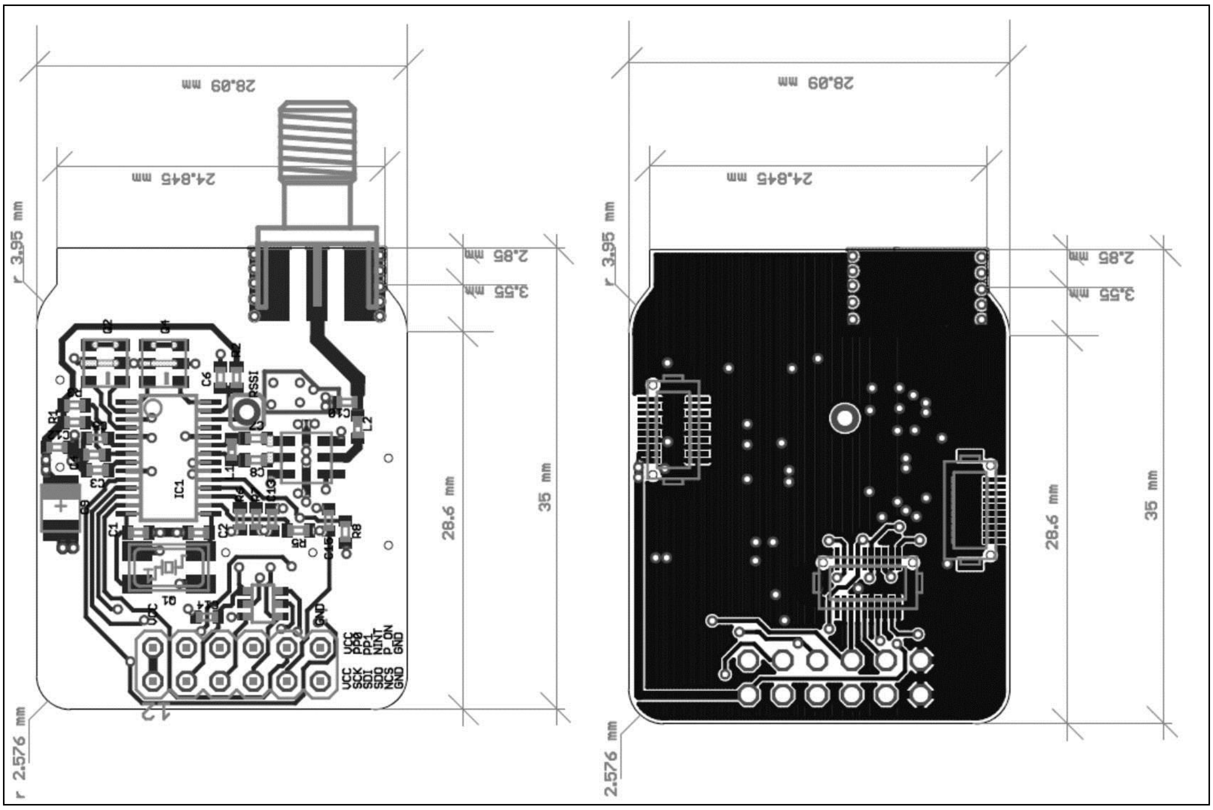

Figure 5 Layout of UWlink Extension board with TDA5240

Table 1 Bill of Materials

Part	Value	Package
IC1	TDA5240	PG-TSSOP-28
R1	10 Ohm	0603
R2	4.7 Ohm	0603
R3	4.7 Ohm	0603
C1	3.9pF	0603
C2	3.9pF	0603
C3	100nF	0603
C4	100nF	0603
C5	100nF	0603
C6	100nF	0603
C7	47nH	0603
C8	47nH	0603
C9	10µF	293B
C10	4.7 pF	0603
C12	100nF	0603
C13	open	0603
C14	open	0603
C15	open	0603
L1	2.2 pF	0603
L2	47 nH	0603
Q1	21.948717 MHz	NX5032SD
Q4	10.7MHz	BW=280kHz
SAW1	433.92MHz	QCC8B
R5	100	0603
R6	100	0603
R7	100	0603
R8	10k	0603

Table 2 Main to Extension board connection summary

µC Port	TDA5240	Comment
P0.5	TDA5240 PP2 / NINT	External Int 0
P0.4	TDA5240 P_ON	µC Output
P0.3	EEPROM NCS	µC Output
P1.6	TDA5240 NCS	µC Output
P1.4 : SPI Master Rx	SDO	Open Drain + Pull-up
P1.3 : SPI Master Tx	SDI	Open Drain + Pull-up
P1.2 : SPI Clock	SCK	Open Drain + Pull-up
P1.1	UART Txd	
P1.0	UART Rxd	
P1.7	TDA5240 PP0 / RX_RUN	I/O
P1.5	TDA5240 PP1 / Data	I/O

3 Software Design

The demonstration program supporting this document was written for the Infineon xc886 Microcontroller using the Keil uVision Integrated Development Environment. A free evaluation version of this IDE comes with the xc800 Family Starter Kit (or can be downloaded from www.Keil.com) which is code size limited to 2K. Unfortunately this demonstration program does not fit within this size restriction mainly due to the use of “printf()” statements to display the received data and status messages. However if the “printf” library files (stdio.h) and UART code (which takes >1k bytes alone) are removed then the remaining code does fit inside this code limit.
The source code example can be found in the ZIP file “TDA5240_UWlink_ApplicationSW”.

Figure 6 shows Tera Term displays received data through UART communication. Tera-Term serial port setup should have settings as follows Port: COM (port selected by PC), Baud rate: 57600, Data: 8 bit, Parity: none, Stop: 1 bit, Flow control: none.

The program was written in a way that hopefully makes the process flow as clear as possible rather than optimizing code size, speed or reliability. It includes modules to handle:
• The “Main” body of the program (“main.c” initially generated by DaVE tool)
• Peripheral initialization (generated by the DaVE tool)
• TDA5240/35 driver functions for Power On Reset, Configuration. Register Read / Write, FIFO read, etc.

The Infineon DaVE Tool was used to create the initialization and peripheral setup functions but the minimum of code was placed inside these functions (apart from the Main() function).
The DaVE Generated modules are:
• MAIN.C
This file performs the project initialization and then runs the project specific user code.
• START_XC.A51
Contains the Startup Code for the Infineon XC8xx devices.
• IO.C
I/O initialization function
• T2.C
Initialization function for Timer 2, which is used as a delay timer (millisecond timer).
• SSC.C
Initialization function for the SPI interface.
• INT.C

Initialization function and interrupt handler for the external interrupt from the TDA5240/35.
• UART.C
Initialisation, interrupt handling and driver code for the UART
further application file is required to complete the programme.
• TDA523x.C
This file contains all the functions to drive the TDA5240/35

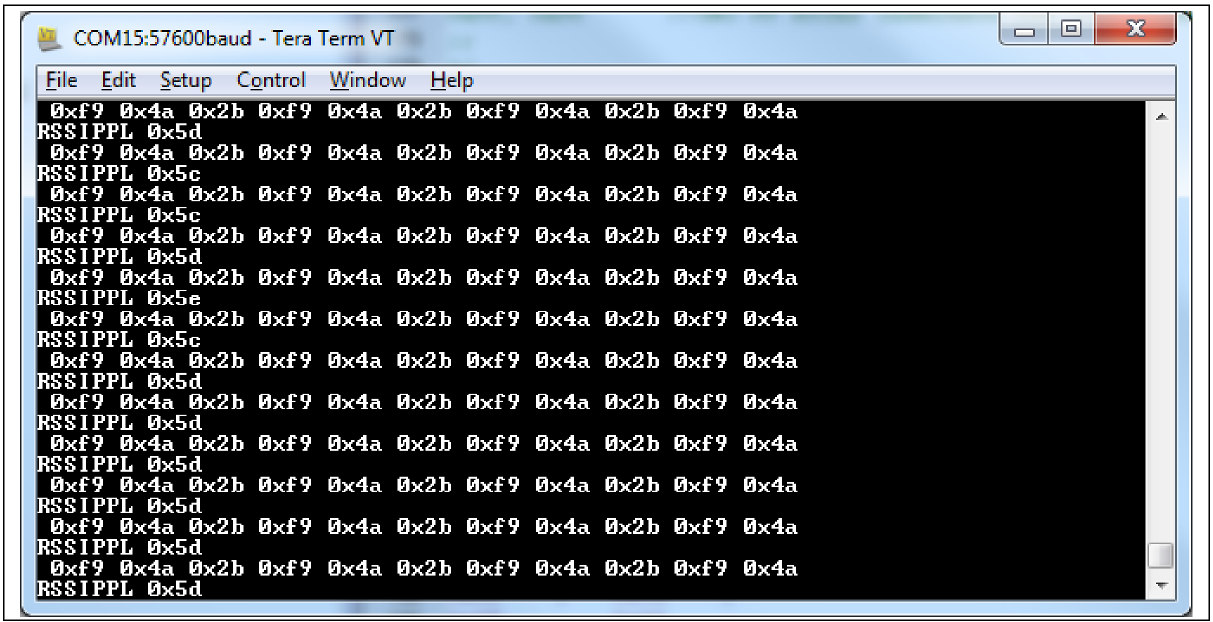

Figure 6 Displaying Received Data through UART

3.1 Power On Reset Routine

Power down and power on of TDA5240 receiver IC is controlled by the pin P_ON. A low at this pin keeps the IC in power down mode and generates an internal reset. All voltage regulators and the internal biasing are switched off. A high at pin P_ON activates the appropriate voltage regulators and the internal biasing of TDA5240.

As shown in Figure 8, reset event is generated by pulling pin P_ON low for at least 100 usec. As a result, its registers are set to default values. There is no need to wait for supply to go low (blocking caps to discharge). On initial power up the Vreg blocking caps (typ 100nF) will charge very quickly (~250us). However if there is an excessive load applied to the vreg, for example a digital I/O line pulled low, then this would take most of the current and the Vreg blocking caps would charge much more slowly, around 4 msec.

When TDA5240 first resets, NINT line is pulled high. And NINT line goes low once reset event has completed. From a power-up condition you should wait around 3msec (T_Reset_max), which is the maximum time for the TDA5240/35 power-up reset.

The Interrupt Status register (IS0) must be read after reset event has completed. As a result, NINT line is pulled from low to high. In SW example, configuration registers are loaded onto TDA5240 after the interrupt status register is read. TDA5240 receiver is operated in Self Polling Mode in order to reduce the average current consumption.

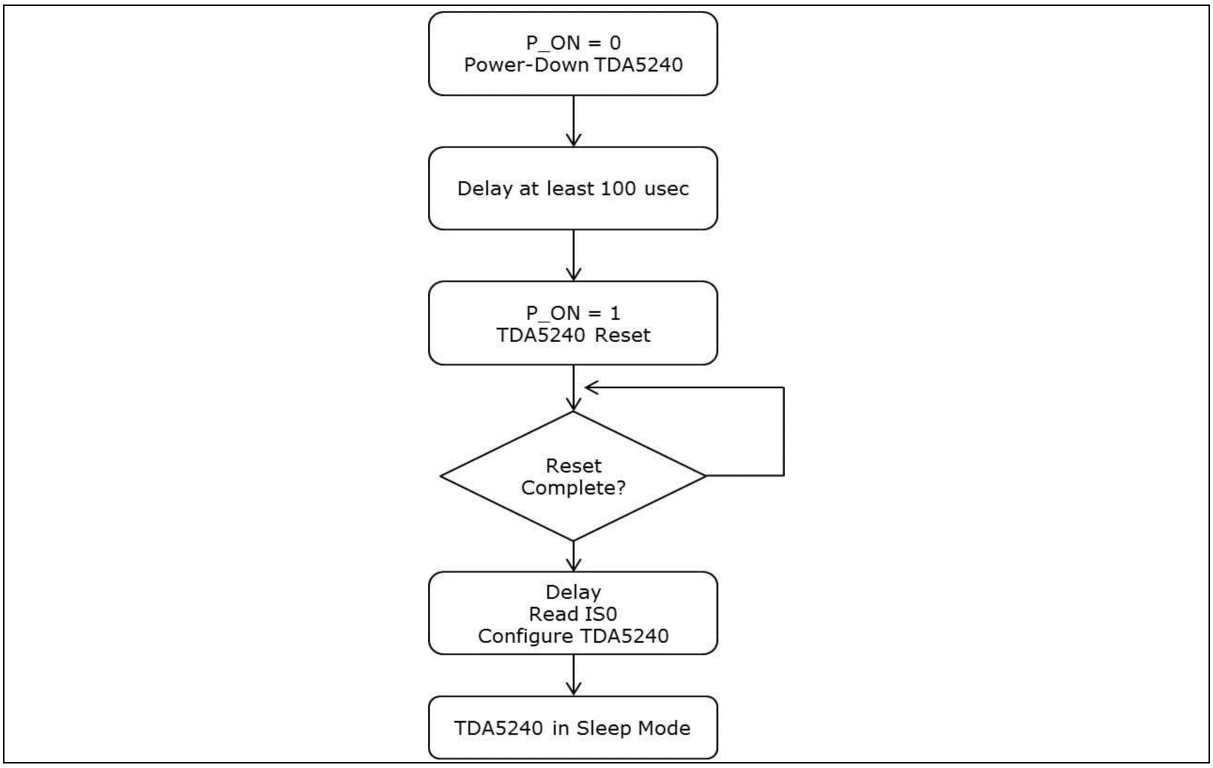

Figure 7 Power On Reset Routine

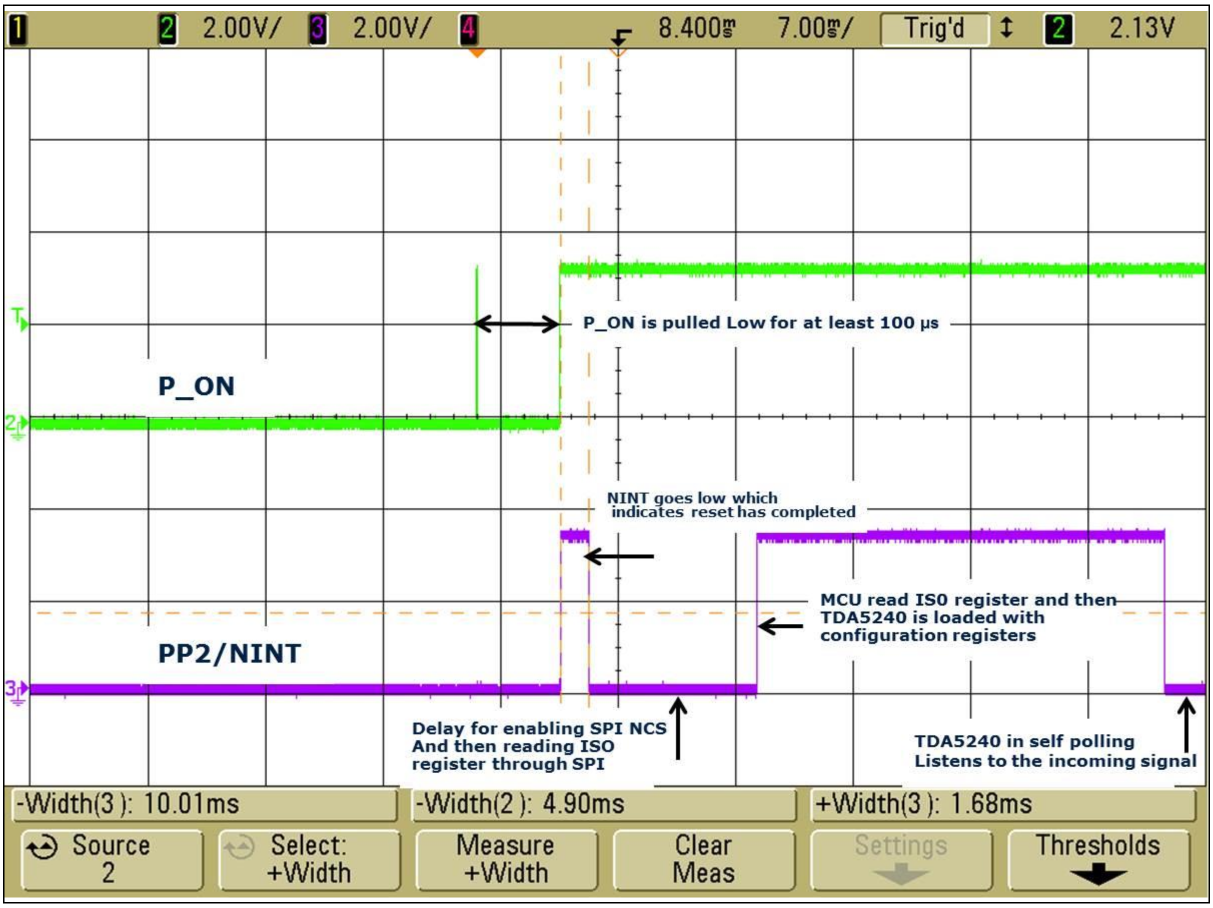

Figure 8 Power On Reset

3.2 Main Routine

When the TDA5240 UWLink is first connected to power supply, MCU performs POR for TDA5240 as described in Power On Reset Routine and then TDA5240 enters sleep mode. In order to operate TDA5240, the device is loaded with configuration registers which can be obtained in TDA5240 Explorer. The configuration registers in the TDA5240 are read only but trace registers (eg. SPIAT, SPIDT) are provided to verify correct reception of address and data after each SPI write command.

During configuration of TDA5240, the device is not able to receive data so it is advisable to perform this function as quickly as possible. This requires a rather high SPI rate (TDA5240/35 supports up to 2.2MHz). Normally the microcontroller provides a hardware SPI module, but the SPI communication can also be programmed in case no hardware SPI module is available. Typical functions to write and read data to the TDA5240 Registers are described in Synchronous Serial Communication Routine.

After TDA5240 configuration, TDA5240 operates either in the autonomous self-polling mode or in run mode slave. In the main routine, MCU wakes up upon receiving an interrupt from TDA5240. Figure 9 describes the flow chart in the main routine. After receiving the interrupt signal from TDA5240, MCU checks whether a reset event occured in TDA5240 before it reads payload data from FIFO of TDA5240. One possible source that triggers a TDA5240 reset is a brownout event. Whenever the integrated brownout detector of TDA5240 measures a voltage drop below the brownout threshold (typical 2.45V) on the digital supply, the integrity of the stored data and configuration can no longer be guaranteed. Thus a reset is generated.

When TDA5240 interrupt occurs and no subsequent reset on TDA5240, MCU reads RSSIPPL register for signal strength level on payload. MCU reads payload on TDA5240 FIFO buffer through SPI. Both RSSIPPL result and the corresponding payload are displayed on the PC's Tera Term window. In this application note, two different configurations (eg.TPMS and RKE configurations) are prepared for working with TDA5240 UWLink.

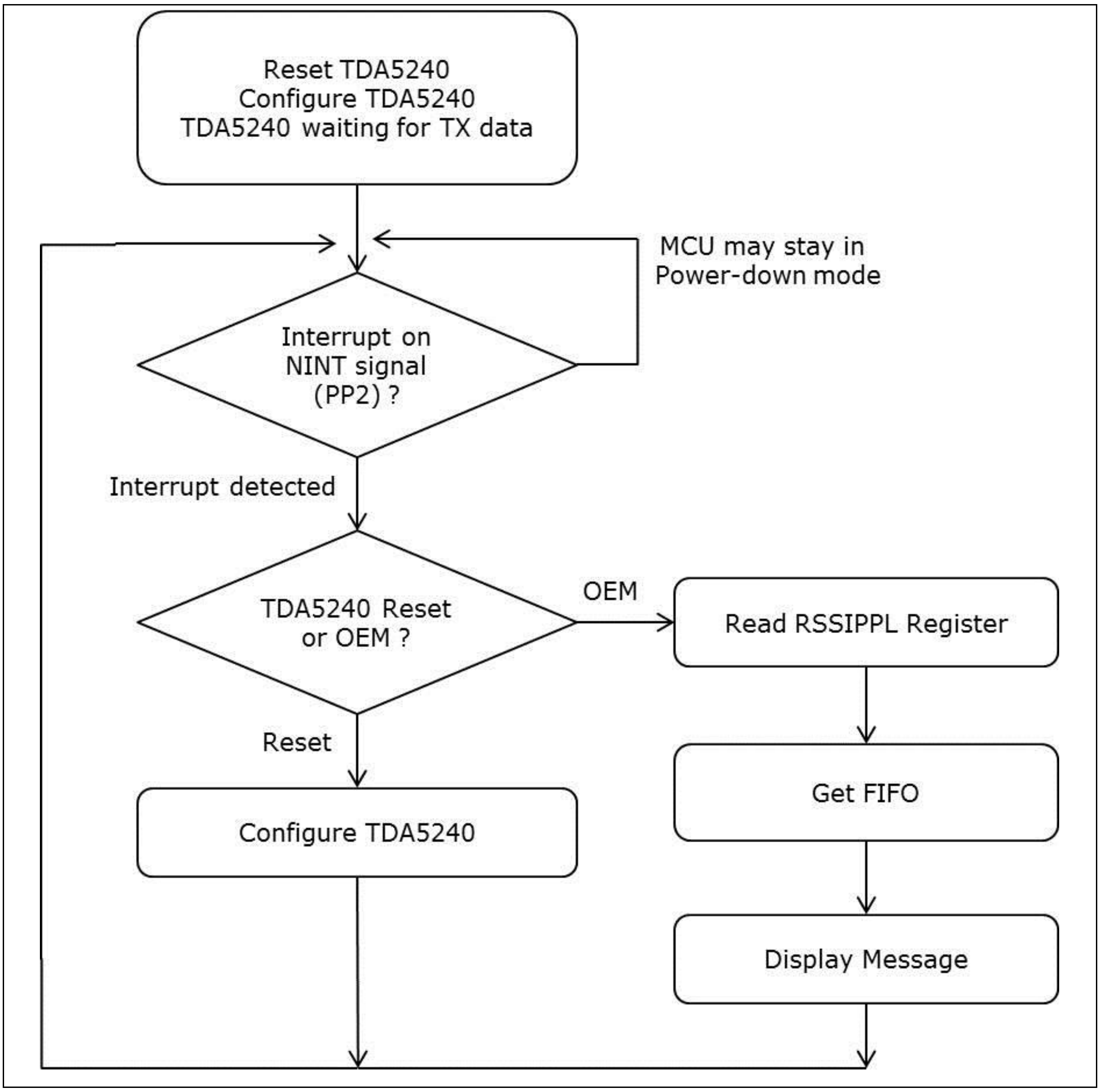

Figure 9 Main Routine

3.3 Synchronous Serial Communication Routines

Serial interface is used for communication between XC886 microcontroller and TDA5240. The microcontroller's communication data-rate is set to 200 kbaud. The following routines are used for reading from SPI address or writing to SPI address based on SPI communication. In the software, TDA523x_getReg perform register read on TDA5240 while TDA523x_putReg performs register write to TDA5240. For retrieving payload from FIFO buffer, TDA5240 uses 'read FIFO' command as described on TDA5240 datasheet. The routine, get_fifo performs reading payload from FIFO buffer on TDA5240.

4 Receiver Configurations

Two different configurations are created to work in TDA5240 UWLink and its detailed information is shown in the sections below. In TPMS application, TDA5240 is configured to operate in Run mode slave and in RKE application, TDA5240 operates in self-polling mode.

4.1 TPMS Application

The following information is used for configuring TDA5240 in TPMS application.

Frequency	433.92 MHz
Modulation	FSK
Deviation	+/35 kHz
Data Rate	9.6 kbps
Coding	Manchester
TSI Bits	0xFE
Payload	11 Bytes

The following shows TPMS frame

Table 3 TPMS Frame

Byte No	Description	Comment
1	Synchronization	Always 0xFF
2	Synchronization	Always 0xFE
3	ID 3	Each Sensor will have a unique 32 bit ID
4	ID 2	
5	ID 1	
6	ID 0	
8	Pressure	Pressure in 8 bit
9	Temperature	Temperature in 8 bit
10	Acceleration	Acceleration in 8 bit
11	Supply Voltage (high byte)	Supply voltage high byte
12	Supply Voltage (low byte)	Supply voltage low byte
13	CRC	CRC in 8 bit

4.2 RKE Application

The following information is used for configuring TDA5240 in RKE application.

WUP bits	TSI bits	Payload

Modulation	ASK
Data Rate	2 kbps
Coding	Manchester
WUP bits	100 bits of Manchester “0”
TSI Bits	0x15
Payload	11 Bytes

4.3 Receiver Measurement

This section describes receiver measurement on oscilloscope after TDA5240 is loaded with configuration registers. The RX_Run signal is active whenever TDA5240 is active or operates. TDA5240 is able to send interrupt signal (NINT signal) to microcontroller on one of the PPx port pins. The Interrupt Generation Unit receives all possible interrupts and sets the NINT signal based on the configuration of the Interrupt Mask registers (IM0 and IM1).

When condition of end of message is found, TDA5240 sends interrupt signal (NINT) on PP2 to the external microcontroller so that the microcontroller starts retrieving data on FIFO buffer. More conditions such as Wake Up bits, Frame Sync, Message ID found, can be set for generating an interrupt signal. Active high or active low of digital output pin (PPx) is configurable via PPCFG2 register (Figure 10 shows inverted polarity for PP2/NINT)

As shown in Figure 10, TDA5240 is configured in self polling mode. In self polling mode the receiver turns periodically on and searches for the wanted signal on the air. If it finds the wanted signal, the receiver continues to receive RF signal until end of message is completely received. If no RF signal is on the air, the receiver goes to sleep mode for saving power consumption.

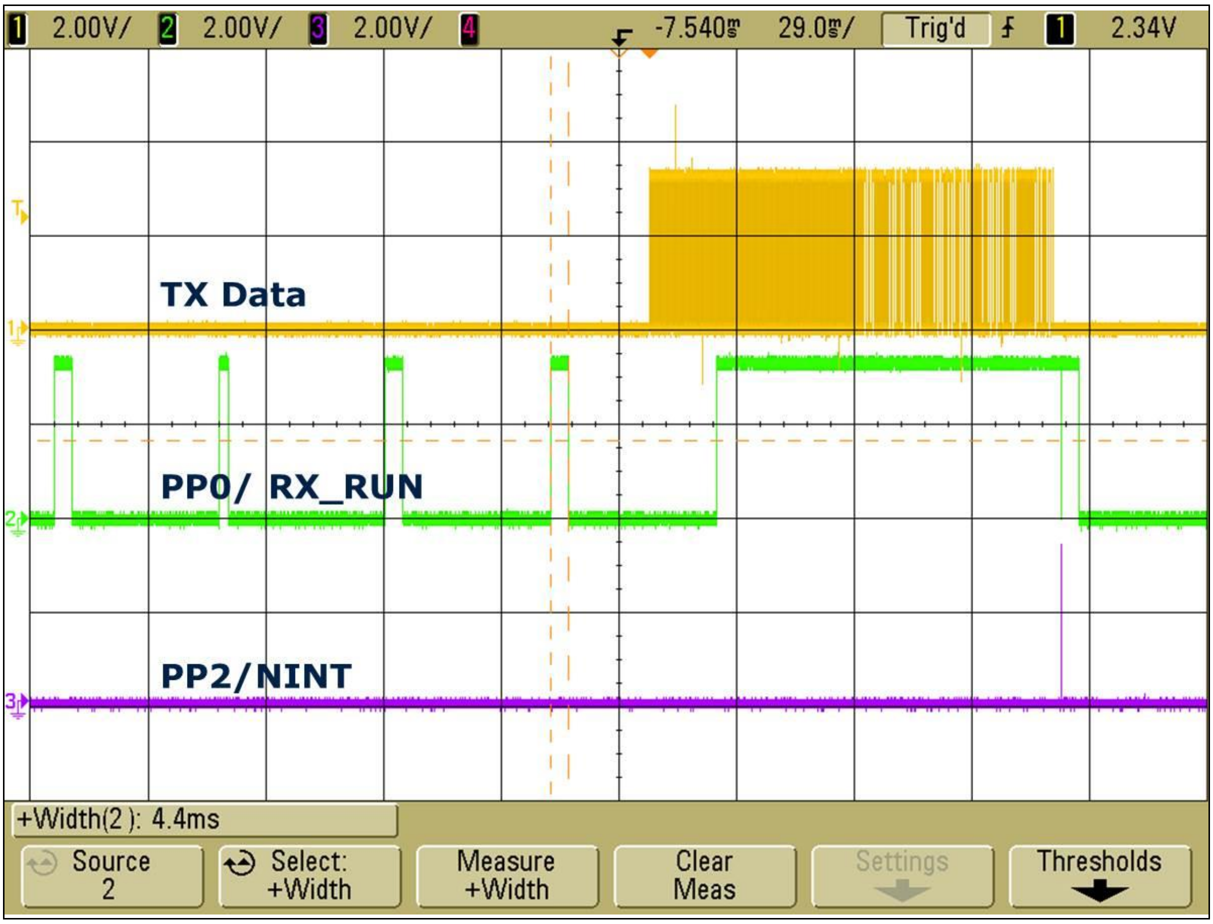

Figure 10 Self-Polling Mode

On TDA5240 UWLink, SPI signals are observable on oscilloscope. The control interface used for device control and data transmission is a 4 wire SPI interface, eg. NCS, SDI, SDO and SCK.

To read from TDA5240, microcontroller (SPI master) has to select TDA5240 (SPI slave) first. Therefore, the master sets the NCS line to low. After this, SCK gets the SPI clock signal. The instruction byte and the address byte are shifted in on SDI and stored in the internal instruction and address register. The data byte at this address is then shifted out on SDO. After completing the read operation, the master sets the NCS line to high. Figure 11 shows measurement where microcontroller reads register IS0 (address 0xA8) and register RSSIPPL (address 0xAC).

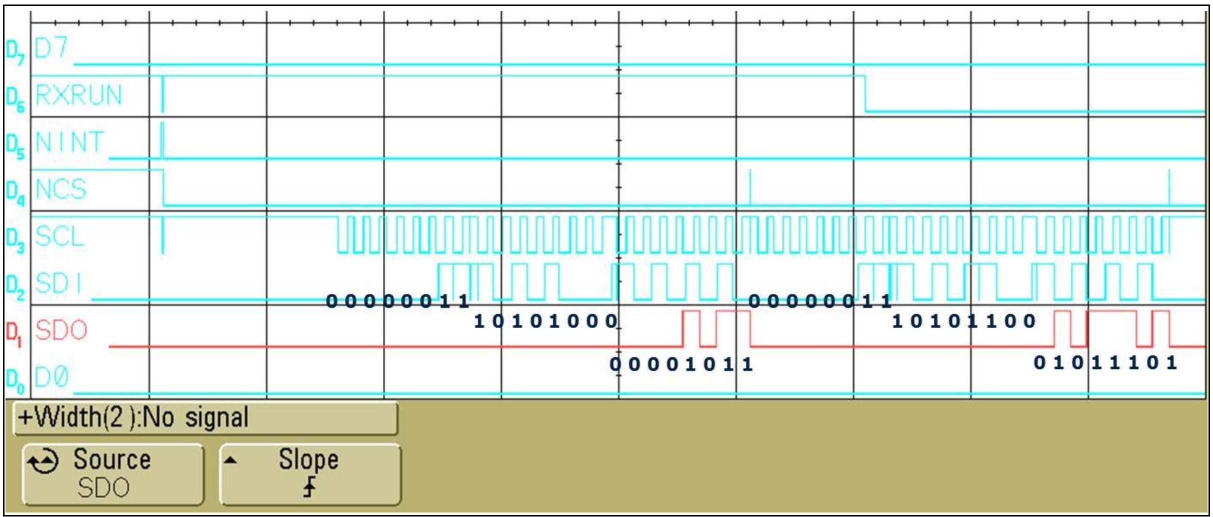

Figure 11 Read SPI Register

To read the FIFO, the SPI master has to select the SPI slave unit first. Therefore, the master must set the NCS line to low. After this, SCK gets the SPI clock signal and the instruction byte is shifted in on SDI and stored in the internal instruction register. The data bits of the FIFO are then shifted out on SDO. The following byte is a status word that contains the number of valid bits in the data packet. After completing the FIFO read operation, the master sets the NCS line to high. Figure 12 shows measurement where microcontroller reads out 0xF9, 0x4A, 0x2B and 0xF9 from FIFO. And its status register 0x20 indicates total bits of 32 are successfully read out.

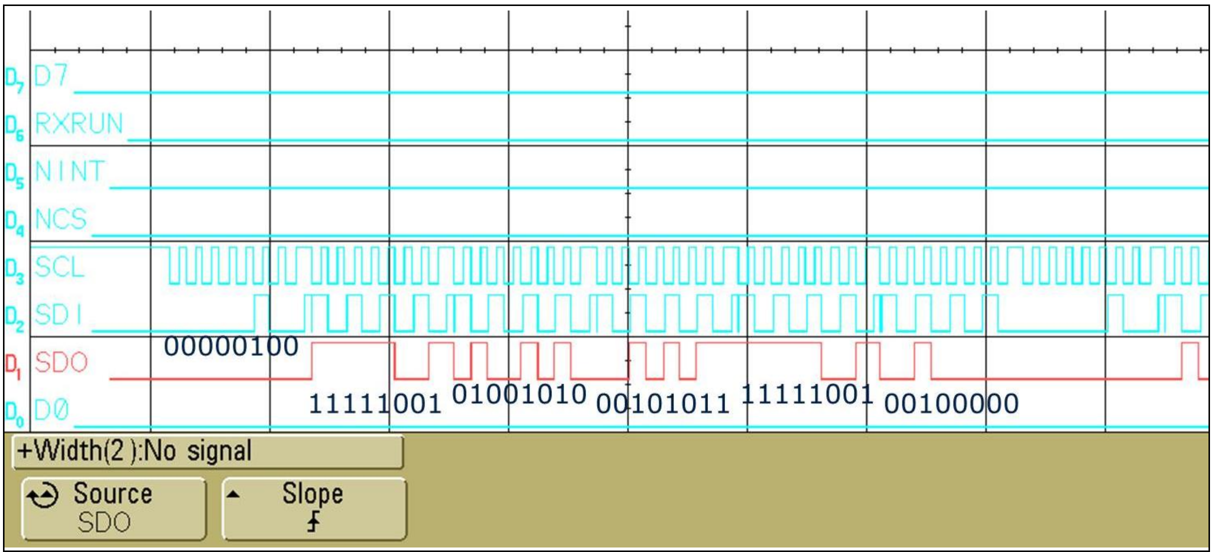

Figure 12 Read FIFO

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG