
DRM103 Designer Reference Manual

Devices Supported:

MMA7361L

MMA745xL

Document Number: DRM103

Rev. 1

08/2008

Safety of Radio Frequency Energy

The manufacturer has evaluated the transmitter for safe operation for uncontrolled use in the general population. The measured power density at 1 cm is under the threshold established by the FCC and is not required to be tested for specific absorption rate. The manufacturer instructs the user that the transmitter should not be handled or placed near the body continuously for more than 30 minutes while operating.

USA:

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

The antenna(s) used for this transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Canada:

This digital apparatus complies with Canadian ICES-003.

Cet appareil numérique est conforme à la norme NMB-003 du Canada.

Europe:

Compliant (CE)

Contents

Chapter 1 Introduction

1.1	Introduction	9
-----	--------------------	---

Chapter 2 ZSTAR3 Wireless Sensing Triple Axis Reference Design Introduction

2.1	Introduction	11
2.2	Features of ZSTAR3	13
2.3	Featured Products	13
2.3.1	Triple Axis Analog Accelerometer MMA7361L	13
2.3.2	Triple Axis Digital Accelerometer MMA745xL	14
2.3.3	The SiP (System in Package) MC13213	15

Chapter 3 ZSTAR3 Sensor Board Description

3.1	Board Overview	17
3.2	Accelerometric Sensor SW Controller	19
3.2.1	Double Sensor Support Software Model	19
3.2.2	Auto Calibration Process	20
3.2.3	Analog Sensor Software Support	20
3.2.4	Digital sensor software support	20
3.2.5	Power Management	21
3.2.6	MC13213 Modem Power Management Features	23
3.3	ZSTAR Sensor Board Hardware Overview	23
3.3.1	Sensors Power Supply	23
3.3.2	Analog Sensor Connection	23
3.3.3	Digital Sensor Connection	24
3.3.4	BDM (Background Debug Mode) Connections	24
3.3.5	Button Connections	25
3.3.6	MC13213 RF Interface	25
3.3.7	Clocking Options for the MC13213	26
3.3.8	LED Indicator	26
3.3.9	Power Supply	26
3.4	Bill of Materials	27
3.5	ZSTAR3 Schematic	29

Chapter 4 USB Stick Board Description

4.1	Board Overview	31
-----	----------------------	----

Chapter 5 Software Design

5.1	Introduction	33
5.2	Simple Media Access Controller (SMAC)	33
5.2.1	SMAC Features	33
5.2.2	Modifications of SMAC for ZSTAR3 RF Protocol	33
5.2.2.1	New Targets Added to SMAC	34
5.2.2.2	New Functions Added to SMAC	34
5.3	The ZSTAR3 RF Protocol	34
5.3.1	ZSTAR3 RF Protocol Features	34
5.3.2	Zpacket Format	35
5.3.2.1	Network Number (NetNum)	35
5.3.2.2	RX Strength	36
5.3.2.3	Zcommand	36
5.3.2.4	Zdata	37
5.3.3	Original ZSTAR Zcommand Description	37
5.3.4	ZSTAR3 Protocol Zcommand Description	37
5.3.4.1	ZSTAR3_DATA	37
5.3.4.2	ZSTAR3_ACK	37
5.3.4.3	ZSTAR3_CONNECT	37
5.3.4.4	ZSTAR3_ION	38
5.3.4.5	ZSTAR3_PYLON	38
5.3.5	ZSTAR3 Protocol SubCommands Description	38
5.3.5.1	ZSTAR3_GUI	38
5.3.5.2	ZSTAR3_MODE	38
5.3.5.3	ZSTAR3_OFF	38
5.3.5.4	ZSTAR3_GSEL	38
5.3.5.5	ZSTAR3_DATARATE	39
5.3.5.6	ZSTAR3_NETNUM	39
5.3.5.7	ZSTAR3_FLAG	39
5.3.6	ZSTAR3 RF Protocol Description	39
5.3.6.1	Typical One Period of ZSTAR3 RF Protocol	40
5.4	ZSTAR3 USB Protocol - Extended STAR Protocol	41
5.4.1	Subset of Original STAR Protocol Commands	41
5.4.2	Subset of Original ZSTAR Protocol Commands	42
5.4.3	Subset of Newly Added ZSTAR3 Protocol Commands	43
5.4.4	Burst Mode	45
5.4.5	Automatic Self-Calibration	46
5.5	Compatibility with Original ZSTAR	46
5.6	Bootloader	46
5.6.1	Switch to Bootloader Procedure	46
5.6.2	Bootloading Procedure	47
5.7	Tri-application Software on the USB Stick	49
5.7.1	CDC - Virtual Serial Port Application	50
5.7.2	HID - Mouse Application	51
5.7.3	HID - Keyboard Application	52
5.7.4	Application Switching	53
5.8	ZSTAR PC library	54

Chapter 6 Application Setup

6.1	ZSTAR3 Installation Procedure	57
6.1.1	USB Stick Installation	57
6.1.2	AN2295 Bootloader Drivers Installation	61

Chapter 7 ZSTAR3 GUI

7.1	Installation	63
7.2	ZSTAR3 GUI	66
7.2.1	Features of ZSTAR GUI	66
7.2.2	The ZSTAR3 GUI Main Controls	67
7.2.2.1	USB Data Flow Monitor	67
7.2.3	Sensor Board Overview Screen	68
7.2.4	RF Overview Screen	69
7.2.5	General Sensor Tasks	70
7.2.5.1	Scope Demo Application	71
7.2.5.2	Acquire Data Demo Application	72
7.2.6	Tilt Tasks	73
7.2.6.1	Filtered Tilt Demo Application	74
7.2.7	Motion Tasks	75
7.2.8	Freefall Task	76
7.2.9	Shock Tasks	77
7.2.10	Digital Tasks	78
7.2.10.1	Sensor Registers Demo Application	79
7.2.10.2	Digital Level Threshold Detection Application	80
7.2.10.3	Digital Pulse Detection Application	81
7.2.11	Freescale Web Links	82
7.3	The ZSTAR3 GUI Update USB Stick Software Utility	83
7.3.1	Update Process (Manual Start)	83

Appendix A References

Figures

The ZSTAR Demo Migration Photo (CR2032 batteries for comparison)	9
ZSTAR3 demo suitcase	10
ZSTAR3 sensor board	11
ZSTAR3 Block Diagram.....	12
Block diagram of MMA745xL	15
ZSTAR3 Sensor Board Overview	17
Sensor Board Block Diagram.....	18
ZSTAR3 Sensor Board Software Overview	19
Transmission/Sleep Cycle Details with a basic 30Hz data rate	22
BDM Connector Layout.....	24
ZSTAR3 Antenna Layout.....	25
Connection Between Antenna and RF Modem.....	26
USB Stick Board Overview	31
The USB Stick Software Overview	32
Zpacket Format.....	35
ZSTAR3_PYLON Zdata format.....	38
ZSTAR3 RF Protocol Period.....	40
CDC application signs (device manager, indication LED)	50
Mouse application signs (device manager, indication LED)	51
Keyboard application signs (device manager, indication LED).....	53
PC Software Composition Overview.....	54
Example Demo Application Based on zstarlib.dll.....	55
Installation Files	63
Start of the Installation	64
Second Step of the Installation Process - Select Destination Directory.....	65
Last step of the Installation Process	66
The ZSTAR3 GUI Main Controls	67
USB Data Flow Monitor	67
Main Sensor Board Overview Screen of the GUI	68
RF Overview Screen.....	69
The General Sensor Tasks Screen.....	70
Scope Window Application	71
Acquire Data Window	72
The Tilt Tasks Screen.....	73
Filtered Tilt Demo Application	74
The Motion Tasks Screen	75
The Freefall Task Screen.....	76
The Shock Tasks Screen.....	77
The Digital Tasks Screen.....	78
Sensors Registers Application Screen.....	79
Level Threshold Detection Application.....	80
Pulse Detection Application	81
The Links Screen	82
Update Software Procedure - Step 1	83
Update Software Procedure - Step 2	83
Update Software Procedure - Step 3	84
Update Software Procedure - Step 4	84

Update Software Procedure - Step 5	85
Original Bootloader Application.....	85

Tables

Sensor Board Bill of Materials - Analog Sensor Version	27
Sensor Board Bill of Materials - Digital Sensor version	28
Original ZSTAR Commands Zcommand List	36
ZSTAR3 Main Commands Zcommand List	36
ZSTAR3 Subcommands Zcommand List	37
Legend of Colors	41
Original STAR Commands	42
Legend of Colors	42
Legend of Colors	43
Original ZSTAR Commands	43
New ZSTAR3 Commands (general commands)	44
New ZSTAR Commands (Digital Sensor part)	45

Chapter 1

Introduction

1.1 Introduction

This paper describes the next generation of a Wireless Sensing Triple Axis Reference design (ZSTAR). This demo represents a wireless demonstration of the 3-axes accelerometer sensors, the MMA7456L (RD3172MMA7456L) and the MMA7361L (RD3172MMA7361L) from Freescale. ZSTAR3 is the successor to the previous Freescale demo ZSTAR and is fully compatible with it. The Demo is built on a new generation of Freescale parts and brings some new extended functionalities.

The reference design will enable you to see how Freescale's accelerometers can add additional functionality to applications in various industries. The accelerometer measurements can be grouped into 6 sensing functions - Fall, Tilt, Motion, Positioning, Shock and Vibration - for multifunctional applications.

The RD3172MMA7456L / RD3172MMA7361L development tool offers robust wireless communication using the powerful, easy-to-use 2.4GHz frequency transceiver and microcontroller in the one package MC13213. Without any changes on board, it can be made with pin to pin compatibility allowing implementation of the MC13214 for ZigBee™ wireless applications.

Only the sensor board was updated completely, as the receiver used is still the USB stick from the previous generation of ZSTAR, but new software was developed that supports all the new functionalities.

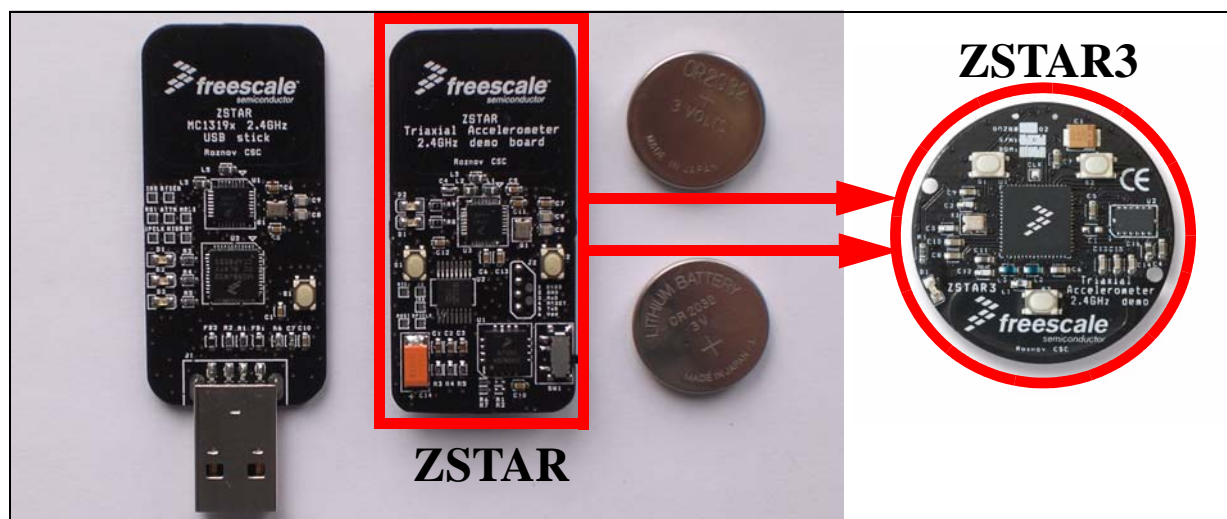


Figure 1-1. The ZSTAR Demo Migration Photo (CR2032 batteries for comparison)



Figure 1-2. ZSTAR3 demo suitcase

Chapter 2

ZSTAR3 Wireless Sensing Triple Axis Reference Design Introduction

2.1 Introduction

The Wireless Sensing Triple Axis Reference Design (ZSTAR3) was designed as a new generation of the previous ZSTAR (RD3152MMA7261Q) demo. A 2.4GHz Radio Frequency (RF) link is also used in this new demo and its based on the new solution modem and microcontroller in one package MC13213 family. It's used for connection between the sensor and PC, allowing visualization of key accelerometer applications as in the previous demo.

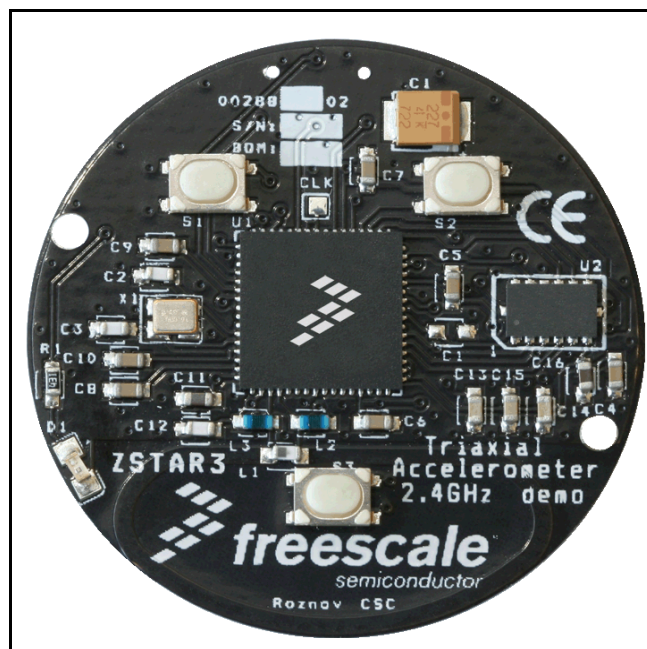


Figure 2-1. ZSTAR3 sensor board

The demo consists of two boards (a new one and an older one with new software):

- Sensor Board (or remote board) is a new board designed for demonstrating Freescale's new 3-axes accelerometer solution for digital (MMA745xL) and analog (MM7361LT) accelerometric data sensing, and the 2.4GHz RF modem with an HCS08 microcontroller in a one package solution as an easy design for remote sensors.
- USB stick, with the MC13191 RF front-end, and the HC08 family MCHC908JW32 for the USB communication. This board is used from the older ZSTAR demo, but with new software.

Both sides communicate over the RF medium utilizing the freely available lightly modified software stack SMAC from Freescale.

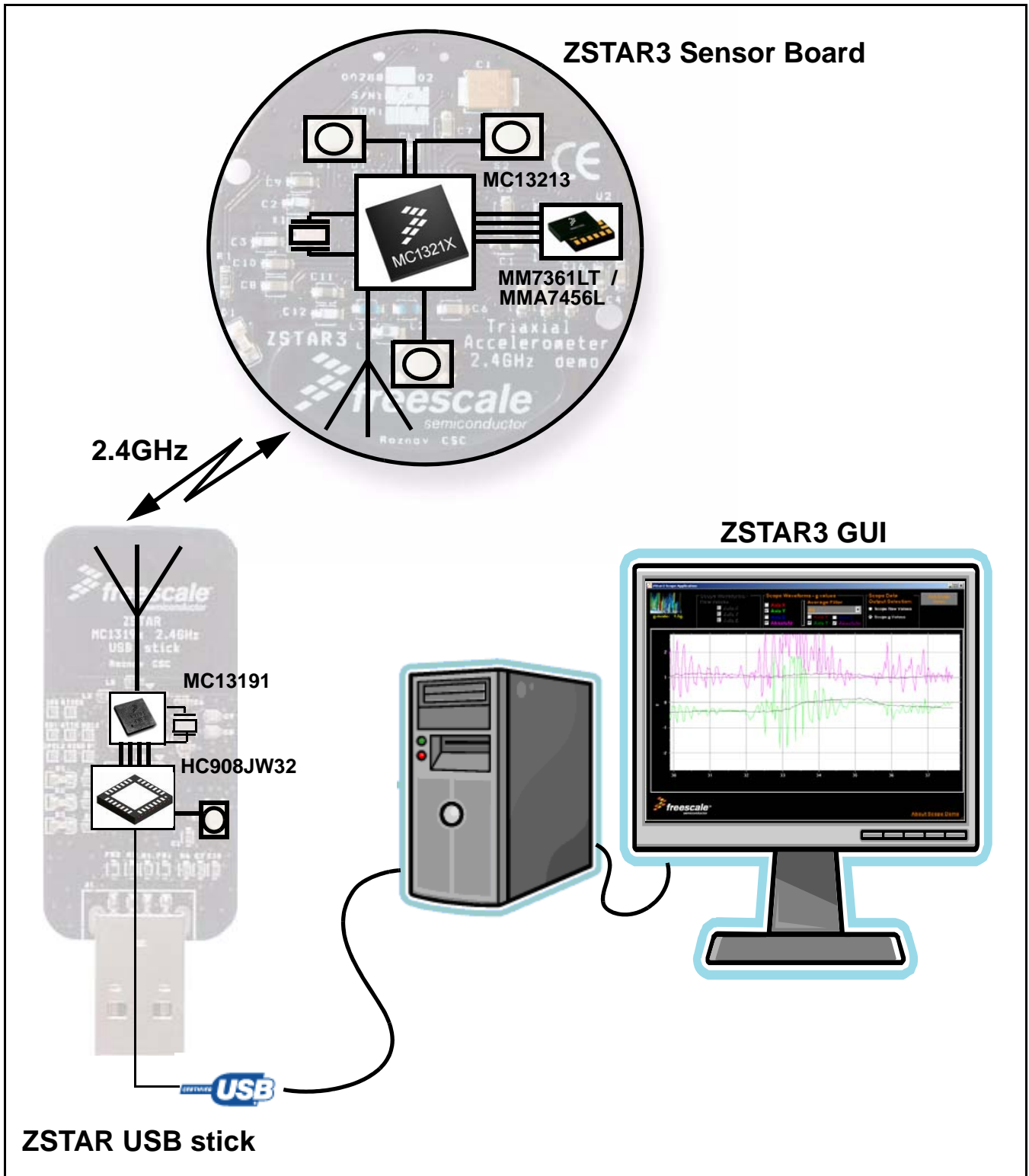


Figure 2-2. ZSTAR3 Block Diagram

2.2 Features of ZSTAR3

- Sensing acceleration in 3 axes
- Handles digital and analog sensors
- Wireless communication with sensors through the **2.4 GHz** band
- **RF protocol supports 16 sensors per one USB stick (receiver)**
- Star topology of the RF network
- Data rate of a sensor is 30, 60 or 120 Hz
- Typical wireless range is 20 m, two walls or one floor
- Auto calibration function of the sensor
- USB communication on the receiver part
 - Virtual serial port - interface for the GUI and serial port terminal
 - HID class - mouse for windows
 - HID class – keyboard (game controller)
- 8-bit/16-bit working modes
- 3 push buttons on the sensor board
- Current consumption:
 - in normal run mode: 1.8 - 3.9 mA, depending on the actual data rate
 - in sleep mode: less than 900 nA
- Power consumption depends on the current output values of the sensor. At a standstill, the board transmits only every 10th packet
- Sensor Board is powered by a coin-sized CR2032 3V battery
- Small size board fits a circular plastic box.

2.3 Featured Products

This demo consists of several Freescale products whose main features are listed below. It is made up of two accelerometers, because the sensor board can be assembled with digital or analog Freescale accelerometers.

2.3.1 Triple Axis Analog Accelerometer MMA7361L

The MMA7361LT is a low power, low profile capacitive micromachined accelerometer featuring signal conditioning, a 1-pole low pass filter, temperature compensation, self test, 0g-detect which detects linear free-fall, and g-Select which allows for the selection between 2 sensitivities. Zero-g offset and sensitivity are factory set and require no external devices. The MMA7361LT includes a sleep mode that makes it ideal for handheld battery powered electronics.

Features:

- 3mm x 5mm x 1.0mm LGA-14-pin package
- Low current consumption: 400 μ A
- Sleep mode: 3 μ A
- Low voltage operation: 2.2 V - 3.6 V

- High sensitivity (800 mV/g at 1.5g)
- Fast turn on time (0.5 ms enable response time)
- Self test for free-fall detect diagnosis
- 0g-Detect for free-fall protection
- Signal conditioning with low pass filter
- Robust design, high shocks survivability
- RoHS compliant
- Environmentally preferred product
- Low cost

2.3.2 Triple Axis Digital Accelerometer MMA745xL

The MMA745xL is a digital output (I2C/SPI), low power, 3 x 5 x 1 mm low profile package capacitive micromachined accelerometer featuring signal conditioning, a low pass filter, temperature compensation, self test, configured to detect 0g through interrupt pins (INT1 or INT2), and pulse (click) detect for quick motion detection. The 0g offset can be customer calibrated using assigned 0g registers and g-Select which allows for command selection of 3 sensitivities (2g/4g/8g). Zero-g offset and sensitivity are factory set and require no external devices. The MMA745xL includes a standby mode that makes it ideal for handheld battery powered electronics.

Features:

- Digital output (I2C/SPI) for processor system performance
- Low-profile 14-pin 3 mm x 5 mm x 1 mm LGA package
- LGA volume is 71 percent smaller than a Quad Flat No-Lead (QFN) package
- XYZ: three axes of sensitivity in one device (2g, 4g, 8g)
- Low current consumption
- Standby mode: 5 μ A
- Low-voltage operation: 2.4 V - 3.6 V
- Customer assigned registers for offset calibration
- Programmable threshold interrupt output
- Level detection for motion recognition (shock, vibration, free-fall)
- Single or double click (pulse) recognition
- High sensitivity
 - 64 LSB/g at 2g
 - 64 LSB/g at 8g in 10-bit mode



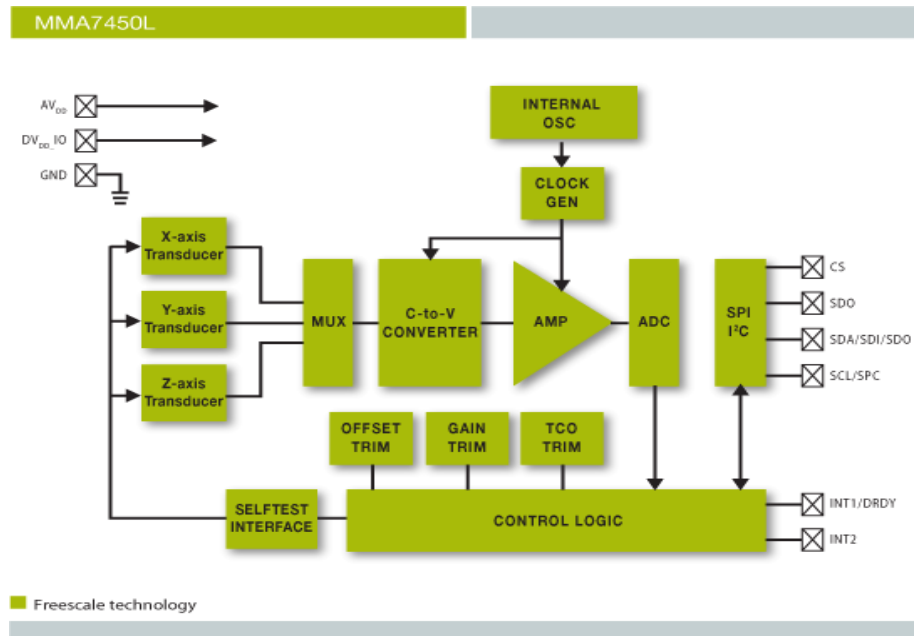


Figure 2-3. Block diagram of MMA745xL

2.3.3 The SiP (System in Package) MC13213

The MC13213 System in Package (SiP) integrates the MC9S08GT MCU with the MC1320x transceiver into a single 9x9mm LGA package. The MC13213 provides 60 K Flash memory and 4 K of RAM. By using the IEEE 802.15.4 Compliant MAC, or BeeStack ZigBee Protocol Stack, the MC13213 is an ideal solution for sensing and control applications that require mesh networking.

Features:

- 40 MHz HCS08 low-voltage, low-power core
- 60 KB Flash and 4KB RAM memory
- Seven addressing modes for the CPU
- Multiple 16-bit timers
- 2V to 3.4V operating voltage with on chip voltage regulator
- -40 to +85 degrees C operating temperature
- Low external component count
- Requires a single 16 MHz crystal
- Programmable frequency clock output for MCU
- Auto-trim feature for crystal accuracy
- Eliminates the need for external variable capacitors
- Allows for automated production frequency calibration
- 9x9x1 mm 71-pin LGA package



- RoHS compliant
- Up to 38 GPIO's
- 8-bit port keyboard interrupt (KBI)
- 8-channel 10-bit analog-to-digital converter (ADC)
- Two independent serial communication interfaces (SCI) supporting up to 115.2 kBaud
- Inter-integrated circuit (I2C) with 100 kbps maximum bus loading
- Internal clock generator (ICG) at 100 kHz or 16 MHz (including internal reference generator)
- Low-voltage detection
- In-circuit debug and Flash programming available via on-chip background debug module (BDM)
- Programmable low voltage interrupt (LVI)
- Common on-chip processor (COP) watchdog timer
- Operates in the 2.4GHz band
- 250 kbps O-PQSK modulation
- 16 selectable channels
- 0 dBm nominal output power
- Programmable from -27 dBm to +3 dBm
- Receive sensitivity of -92 dBm (typical) at 1% PER
- Integrated transmit/receive switch
- Supports single-ended or full differential operation
- Supports external low-noise amplifier (LNA) and/or Power Amplifier (PA)
- Three lower power modes for increased power life
- Supports streaming and data processing modes

Software Features:

- Simple MAC
 - Small memory footprint (< 4 KB)
 - Supports point-to-point and star network configurations
 - ANSI C source code
- IEEE 802.15.4 compliant MAC
 - Supports star, mesh and cluster tree topologies
 - Supports beaconed and non-beaconed networks
 - Supports guaranteed time slots (GTS) for predicable latency
 - 128-bit Asymmetric Encryption Standard (AES)
 - Object Code
- BeeStack ZigBee Protocol Stack
 - ZigBee 2006 Compliant Platform
 - Object Code

Chapter 3

ZSTAR3 Sensor Board Description

3.1 Board Overview

The Sensor Board utilizes a small footprint size dual-layer Printed Circuit Board (PCB) containing all the necessary circuitry for both accelerometer sensors and transferring data over a Radio Frequency (RF).

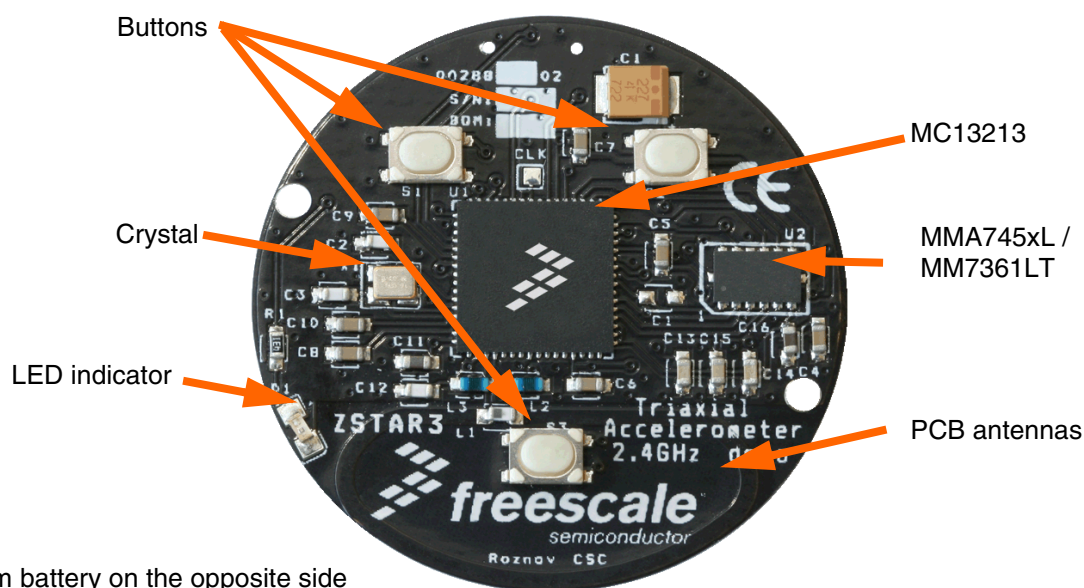


Figure 3-1. ZSTAR3 Sensor Board Overview

The board is powered by a Lithium coin-sized CR2032 battery. The block diagram of the board is as follows:

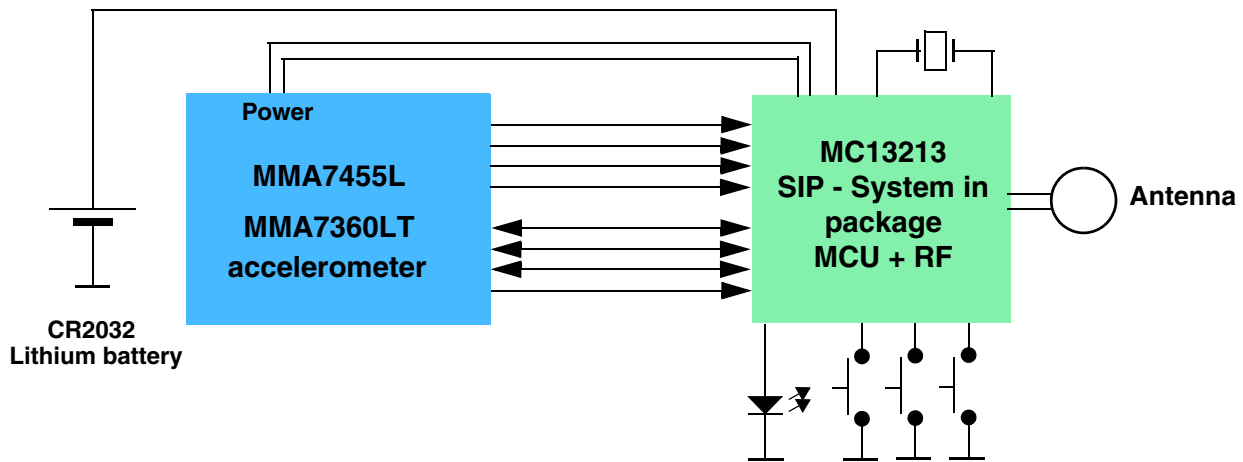


Figure 3-2. Sensor Board Block Diagram

Figure 3-3. ZSTAR3 Sensor Board Software Overview shows in more detail, how different software and hardware modules co-operate with each other. The main task of the Sensor Board is to:

- periodically wake-up from power saving mode
- measure all three XYZ acceleration values from the Sensor
- compose a data frame using simple [The ZSTAR3 RF Protocol](#)
- use [Simple Media Access Controller \(SMAC\)](#) to send this data frame over the RF link
- go to sleep.

This basic loop repeats 30 times per second (period is 33.333 ms) providing nearly a real-time response from the Sensor.

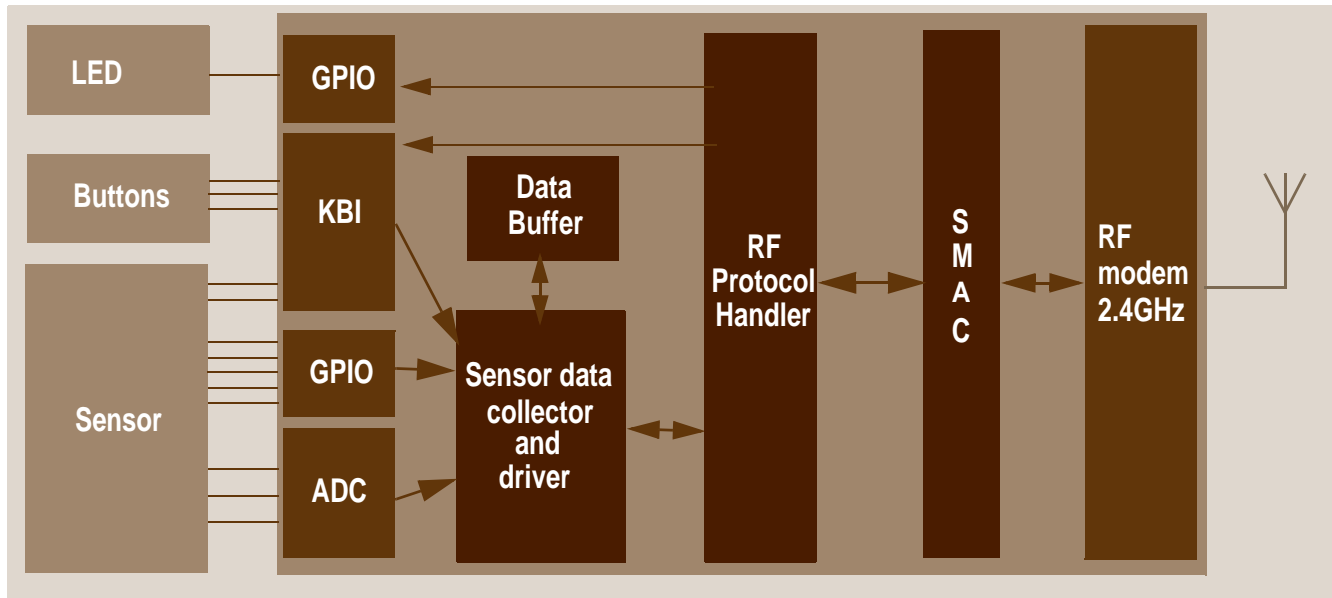


Figure 3-3. ZSTAR3 Sensor Board Software Overview

For the Sensor Board operation, several of the MC13213's hardware modules are used: Analog to Digital Converter (ADC), Synchronous Peripheral Interface (SPI), External Interrupt Request (IRQ), Keyboard Interrupts and General Purpose Input/Output (GPIO).

3.2 Accelerometric Sensor SW Controller

Reading of the XYZ levels and all other operations with sensor is dependent on the currently assembled sensor. The ZSTAR3 sensor board supports two types of Freescale accelerometric sensors, analog (MMA736xL) and digital (MMA745xL). The assembled sensor is powered by the MCU IO pins, because the solution achieves lowest power consumption in sleep mode.

3.2.1 Double Sensor Support Software Model

All common sensor control functions are physically created as two single functions. The first is for the analog, and the second for the digital sensor. The main software only uses volatile pointers to these functions, which are assigned within program initialization, by the `Recognise_Sensor()` function. This function recognizes an assembled sensor and assigns the correct function address of functions to the volatile RAM pointers. For example, in this way the analog sensor uses the ADC to read XYZ sensor values of sensor and the digital sensor uses the digital interface, but in the source code there is only one line:

```
p_Read_Accelerometer((void*) &(accel_data[0].x)).
```

3.2.2 Auto Calibration Process

The software uses for both types of sensor an auto calibration process to get the offset calibration values. It uses a simple 0g X, 0g Y, +1g Z auto calibration method. The sensor board runs the auto calibration process for each g scale of the sensor, and thus uses an individual set of calibration values for each g scale. For more details, see application note AN3447 - Implementing Auto-Zero Calibration Techniques for accelerometers at www.freescale.com.

3.2.3 Analog Sensor Software Support

The 3-axis accelerometer sensor MMA7361L provides three separate analog levels for the X, Y and Z axis. These outputs are ratiometric, which means that the output offset voltage and sensitivity will scale linearly with the applied supply voltage. This is a key feature when interfacing to a microcontroller with A/D converter reference levels tied to a power supply, because it provides a system level cancellation of supply induced errors in the analog to digital conversion process.

During the analog-to-digital conversion in the microcontroller, 10-bit resolution is used. MC13213 A/D channels 0, 1 and 2 are connected to X (channel 1), Y (channel 2) and Z (channel 0) outputs of the MMA7361L. The microcontroller's APCTL1 register enables these ADC channels for I/O pin control by the ADC module.

The ADCCFG register controls the selected mode of operation, clock source, clock divide, and configuration for low power or long sample times.

The MMA7361L sensor implements a digital output for the free-fall detection module. This output is connected directly to the KBI (Keyboard interrupt module) and it's used as the default source for free-fall detection.

3.2.4 Digital sensor software support

The MMA745xL provides a lot of various features that are almost all used of the ZSTAR3 demo. Complete communication with the sensor is through a digital interface. The MMA745xL supports two standards of digital communication: Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I²C). All communication interfaces are realized in software, only without any hardware module. The SPI hw module is used for internal communication between the MCU and the RF modem. For the I²C communication, a software only driver is used, because there aren't any external pull up resistors placed onboard (cost and space saving reasons).

3.2.5 Power Management

A CR2032 Lithium battery provides a fairly limited charge for such a real-time-like demo that demands frequent transmissions. Some sort of power management has to be implemented in order to keep the current consumption at a reasonable level.

Typically, current consumptions of the Sensor Board components are as follows:

- SIP - System in Pack MC13213
 - 2.4GHz transceiver of the MC13213
 - in Off mode, 200nA
 - in Hibernate mode, 2.3 μ A
 - in Doze mode, 35 μ A
 - in Idle mode, 500 μ A
 - in Transmit mode, 30mA
 - in Receive mode, 37mA
 - 8-bit microcontroller of the MC13213
 - in Stop3 mode, 700nA
 - in Wait mode, 560 μ A
 - in Run mode, 6.5mA
- Tri-axial accelerometers
 - low-g tri-axial analog sensor MMA7361L
 - in Sleep mode, 3 μ A
 - in Normal mode, 400 μ A
 - low-g tri-axial digital sensor MMA745xL
 - in Sleep mode, 5 μ A
 - in Normal mode, 400 μ A

It is obvious that in a battery operated application care must be taken to ensure the lowest possible current consumption, especially when the maximum current (provided by the battery) is somehow limited. A CR2032 Lithium battery cannot provide current in the range of 40mA for long periods of time. To alleviate high current surges, an additional large capacitor has been designed.

For transmission and reception using the MC13213, a specific scheme has been used to ensure the battery is not depleted or overloaded, targeting a 30 samples per second (33ms period) transmission rate. For better power management, the software uses a system of skipping transmission if sensor output data is the same or very similar to the last transmitted sample. The maximum count of skip transmissions is 10, then sensor data is always transmitted. Every received transmission from the sensor board clears the time-out counter in the USB stick. The ZSTAR3 sensor board typically opens a receive window every 10th period (~333ms) to maintain synchronization with the USB stick (Communication Master) and to receive possible control data from the master.

The following scheme for one transmission/sleep cycle is used for a typical data transfer:

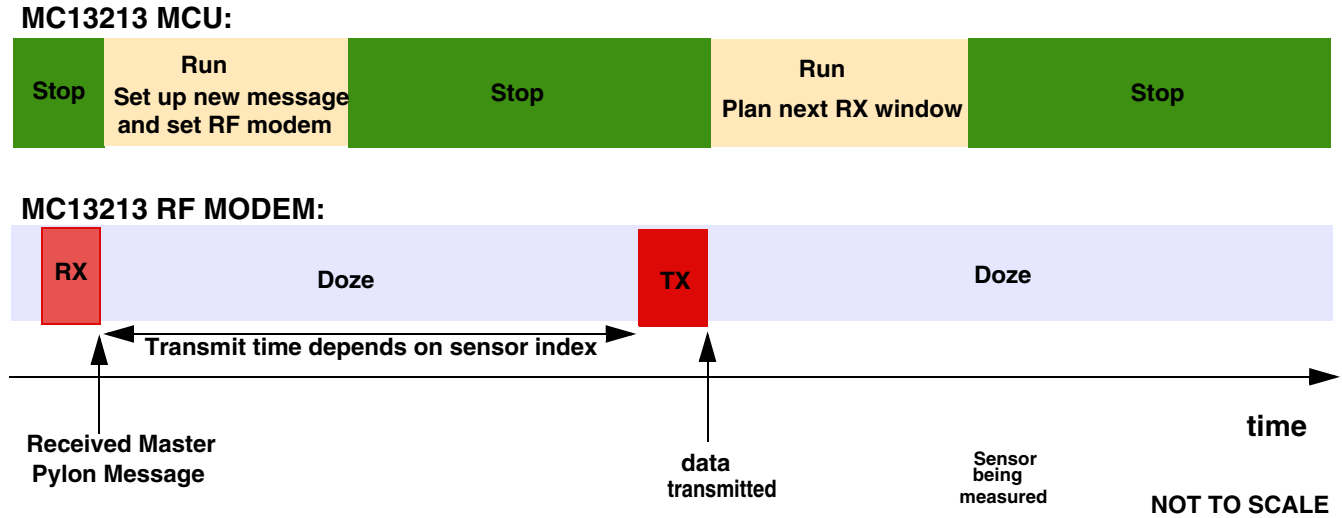


Figure 3-4. Transmission/Sleep Cycle Details with a basic 30Hz data rate

As shown in the previous diagram, microcontroller parts of the sensor board remain most of the time in Doze/Stop modes, in which the total current consumption is below 450 μ A. Most of this power consumption is from a sensor that is always in run mode. The sensor runs because its wake up time is longer than the ZSTAR3 measure period.

The current consumption of the transmitter is ~30 mA at that time, but only for a short period of time (typically ~600 μ s by 30 Hz).

In order to keep the sensor board informed of the connection status (for example, if the data-receiving side - USB stick - is out of range, disconnected, etc.) and still synchronize with the master, the reception has to be turned on after the data has been transmitted. This is not really required within each loop cycle, and in the actual implementation only on every 10th loop is the receive window opened (receiver is enabled to receive the synchronization [ZSTAR3_PYLON](#) message). More in [Section 5.3, The ZSTAR3 RF Protocol](#) description.

The reception window is larger to fit any incoming receive data and the current consumption is also higher during reception, so this portion of current consumption would be one of the largest if the [ZSTAR3_PYLON](#) message was received in every loop cycle.

The “optional receive” feature allows huge power savings, whilst still keeping synchronization of ZSTAR3 network data from the data-receiving side.

3.2.6 MC13213 Modem Power Management Features

The MC13213 modem provides several power saving modes. One of them is called **Doze mode** in which the MC13213 modem crystal oscillator remains active. An internal timer comparator is functional too, providing a power efficient and accurately timed way of waking-up the application after a specified time.

This feature is fully utilized within the sensor board. The microcontroller calculates the time period for which the application should be in power saving mode, then fills in the timer comparator registers in the modem, and the microcontroller goes into Stop mode (modem into Doze mode).

Once the timer reaches the pre-programmed time (a timer compare occurs), the modem's IRQ signal is asserted which brings the microcontroller out of the Stop mode. There are various scaling possibilities that allow periods from a few μs up to 1073 seconds (~17 minutes) to be programmed, without intervention by the microcontroller.

3.3 ZSTAR Sensor Board Hardware Overview

This section describes the Sensor Board in terms of hardware design. The MC13213 SIP drives only the analog or digital tri-axial sensor. Because both the analog and the digital sensors use only one footprint, all necessary pins for both the sensors need to be connected, and these connections have to be a logical fit.

3.3.1 Sensors Power Supply

The sensors are powered by microcontroller I/O pins. This can be done because the sensors consume less than 0.5mA. Three output pins connected together are used to supply the analog VDD power pin of the sensor. This three pin option holds down any dynamics changes in voltage with different current consumptions of the sensor.

The main feature of this solution is a secure current, when the complete board is in deep sleep mode (the entire board consumes less than $1\mu\text{A}$). Thus the ZSTAR3 doesn't need to use sleep modes for the sensors. For more details see [Section 3.3.1, Sensors Power Supply](#).

3.3.2 Analog Sensor Connection

The MMA7361L sensor outputs are connected to the AD0, AD1, and AD2 inputs of analog-to-digital converter via RC filters, formed by internal resistors in the sensor and the C13, C14, C15 external capacitors. These filters are recommended, to minimize clock noise from the switched capacitor filter circuit inside the sensor. Once the software filtering is employed, these RC filters may be completely omitted.

The MMA7361L provides four further digital signals **g-select**, **Self-test**, **0g-detect** and **Sleep**. All these pins are also routed into the microcontroller, but only the g-select and the 0g-detect are controlled by the current version of the ZSTAR3 software.

3.3.3 Digital Sensor Connection

The MMA745xL sensor is controlled by a digital interface and two interrupts pins. The digital interface takes 4 wire that provide both interfaces - I²C and SPI. These pins are routed to general I/O pins and an I²C microcontroller module. The board doesn't provide external pull up resistors and so an I²C module can't be used. This option saves battery energy in deep sleep mode, but the hw is prepared for testing the sensor with an I²C hw module once the user adds external resistors.

- **pin 13 - SDA/SDO/SDI** - I²C data, SPI data input/output
- **pin 12 - SDO** - SPI data output
- **pin 14 - SCL/SPC** - I²C clock, SPI clock
- **pin 7 - CS** - Chip select

The two interrupts pins INT1/DRDY and INT2 are routed directly to the KBI module of the microcontroller. This option allows building software on interrupt events. The signals use internal pull up resistors inside the microcontroller.

- **pin 8 - INT1/DRDY** - interrupt 1 output, Data Ready output
- **pin 2 - INT2** - interrupt 2 output

For the digital sensor, a 0R resistor is placed on footprints C13 and C15 from BOM and schematic. These two resistors ground pin2 and pin4 of the digital sensor.

3.3.4 BDM (Background Debug Mode) Connections

A J2 connector has a non-standard footprint primarily intended for in-factory programming and testing via “spring-needle” type of connections. The J2 connector carries all standard signals for Background Debug Mode communication so if required, one may solder wires and a standard 2 x 3 pins 2.54 mm (100 mil) pitch header for regular BDM re-programming. The pin numbering is shown in Figure 3-5. BDM Connector Layout

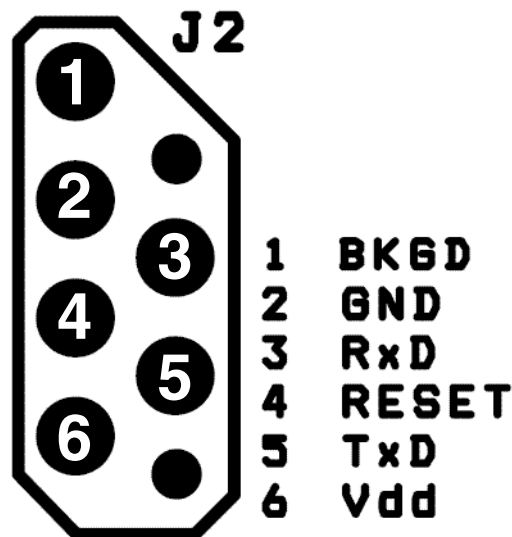


Figure 3-5. BDM Connector Layout

3.3.5 Button Connections

The three buttons (S1, S2 and S3) are connected directly to pins PTA0, PTA6 and PTA7. All these pins have internal pull-up resistors, and are part of the Keyboard Interrupt module, therefore allow a direct microcontroller wake-up from the Stop modes.

3.3.6 MC13213 RF Interface

The RF interface (antenna) was designed with the cost and board size in mind. Among several designs, the PCB layout antenna was in the main consideration (cost). Of several PCB antenna designs available for the 2.4 GHz band (F-antenna, dipole, loop), the loop antenna has been selected mainly because of the size required on the PCB.

The MC13213 transceiver provides an internal antenna switch for RF IN (receive) and PA OUT (transmit) RF paths, so can use a single simple loop antenna.

The antenna was designed as a “smile” layout, 10.7 x 24.3 mm (420 x 960 mil), made of 1.25 mm (50 mil) wide trace of copper.

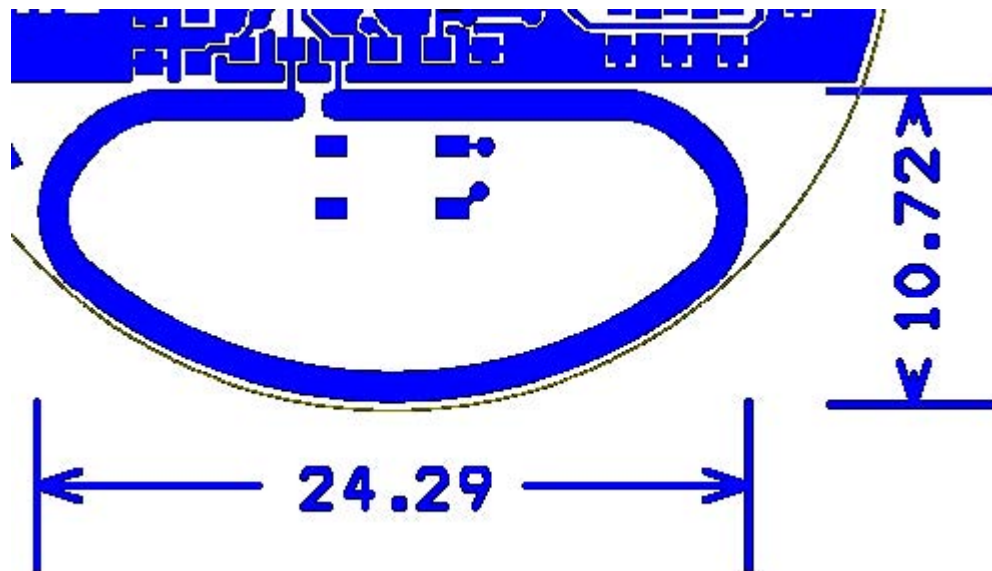


Figure 3-6. ZSTAR3 Antenna Layout

The matching is provided by L1 coil. L2 and L3 coils bias the transmitter output transistors to the CT_Bias level.

The inductors used in this design are from TDK:

- L1 (3.3nH) MLG1608B3N3DT
- L2, L3 (22nH) MLG1608B22NJT

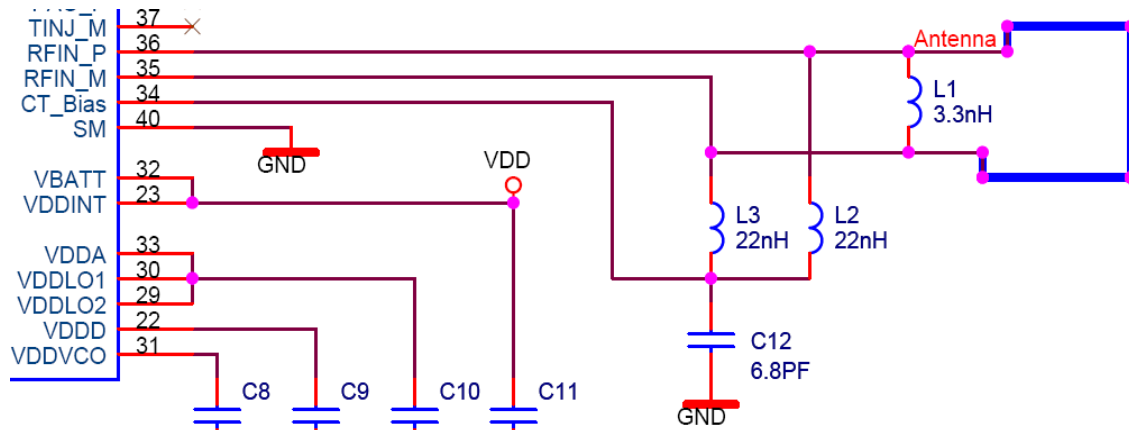


Figure 3-7. Connection Between Antenna and RF Modem

3.3.7 Clocking Options for the MC13213

Although accurate timing provided by the MC13213 internal transceiver is available, an internal oscillator (ICG) in the MC13213 microcontroller is used as the main clock source for the microcontroller. This option was chosen, because the main software uses the Real Time Interrupt (RTI) module, and this module can not run in MCU Stop mode from an external clock source. The protocol related timing is derived from RF modem timers, and the microcontroller itself is clocked from an internal oscillator. The microcontroller clock runs with 20 MHz on the bus.

3.3.8 LED Indicator

On the design only one LED is used, because the ZSTAR3 board is designed to fit a plastic box that has only one position for a light indicator. The LED is connected with the current limiting serial resistor R1 directly to the microcontroller PTC5 pin.

3.3.9 Power Supply

The sensor board is powered by a Lithium coin-sized battery. The primary choice was the popular CR2032. A surface mounted SMTU series battery holder from Renata™ is placed on the underside of the PCB. The SMTU series holders provide (by mechanical construction) battery reverse protection, so no additional circuitry is required.

A large tantalum capacitor (C1, 220 μ F/4 V) improves the response of the power supply to current peaks caused by reception or transmission. The coin-sized Lithium CR2032 batteries are targeted at a maximum continuous discharge current in the range of 3mA. Such a large capacitor helps to supply enough current to the MC13213 during a receive/transmit without significant V_{dd} voltage drops.

The design doesn't have any main power switch, thus switch on is done by pressing any button on the board and switch off is done by software (Time-out, Out of range, RF protocol command).

3.4 Bill of Materials

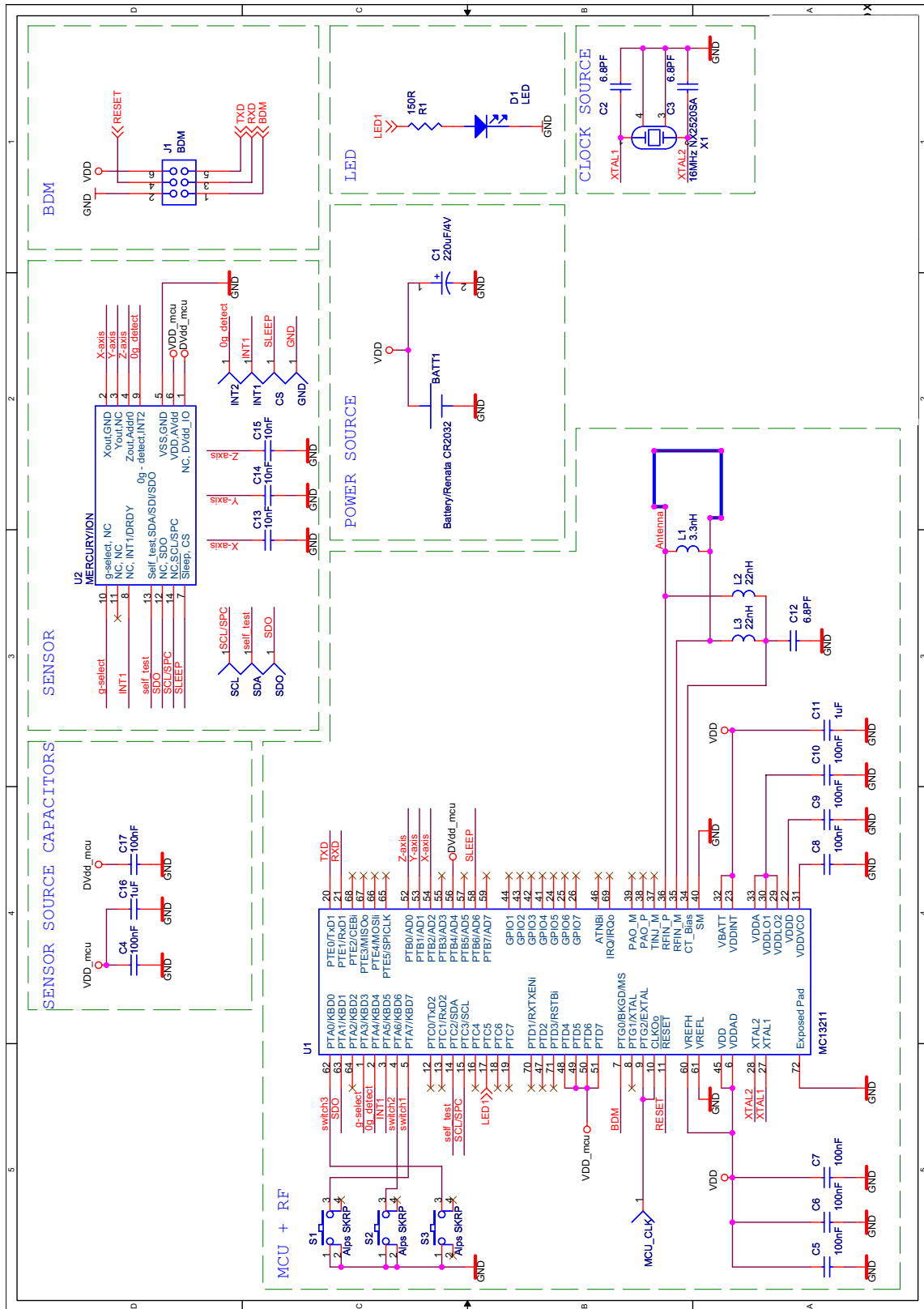
Table 3-1. Sensor Board Bill of Materials - Analog Sensor Version

Item	Quantity	Reference	Part	Manufacturer	Manufacturer Order Code
1	1	BATT1	battery holder CR2032	Renata	SMTU 2032-1
2	1	C1	220uF/4V	AVX	TAJB227M004R
3	3	C2, C3, C12	6.8pF	TDK	C1608CH1H070D
4	7	C4, C5, C6, C7, C8, C9, C10	100nF	TDK	C1608JB1H104K
5	2	C11, C16	1uF	TDK	C1608JB1A105KB
6	3	C13, C14, C15	10nF	TDK	C1608CH1E103J
7	1	L1	3.3nH	TDK	MLG1608B3N3DT
8	2	L2, L3	22nH	TDK	MLG1608B22NJT
9	1	D1	Kingbright KP-1608SEC	Kingbright	KP-1608SEC
10	1	J1	BDM + serial	N/A	
11	1	X1	16MHz NX2520SA	NDK	NX2520SA 16MHz EXS00A-02940 Specification n° EXS10B-07228
12	1	R1	150R	TYCO	RN73F1J150RBTG
13	3	S1, S2, S3	switch SKRP	Alps	SKRPADE010 (or SKRSPACE010 or SKRPABE010)
14	1	U1	MC13213	Freescale	MC13213
15	1	U2	MMA7361LT	Freescale	MMA7361LT
16	1	C17	NA	NA	NA

Table 3-2. Sensor Board Bill of Materials - Digital Sensor version

Item	Quantity	Reference	Part	Manufacturer	Manufacturer order code
1	1	BATT1	battery holder CR2032	Renata	SMTU 2032-1
2	1	C1	220uF/4V	AVX	TAJB227M004R
3	3	C2, C3, C12	6.8pF	TDK	C1608CH1H070D
4	8	C4, C5, C6, C7, C8, C9, C10, C17	100nF	TDK	C1608JB1H104K
5	2	C11, C16	1uF	TDK	C1608JB1A105KB
6	3	C13, C15	0R	PHYCOMP	232270296001
7	1	L1	3.3nH	TDK	MLG1608B3N3DT
8	2	L2, L3	22nH	TDK	MLG1608B22NJT
9	1	D1	Kingbright KP-1608SEC	Kingbright	KP-1608SEC
10	1	J1	BDM + serial	N/A	
11	1	X1	16MHz NX2520SA	NDK	NX2520SA 16MHz EXS00A-02940 Specification n° EXS10B-07228
12	1	R1	150R	TYCO	RN73F1J150RBTG
13	3	S1, S2, S3	switch SKRP	Alps	SKRPADE010 (or SKRSPACE010 or SKRPABE010)
14	1	U1	MC13213	Freescale	MC13213
15	1	U2	MMA745xL	Freescale	MMA745xL
16	1	C14	NA	NA	NA

3.5 ZSTAR3 Schematic



DRM103, Rev. 1

Chapter 4

USB Stick Board Description

4.1 Board Overview

The USB stick board used is the same as in the original ZSTAR demo, but with new software. For more hardware details, please check the RD3152MMA7261 Reference Design manual on www.freescale.com

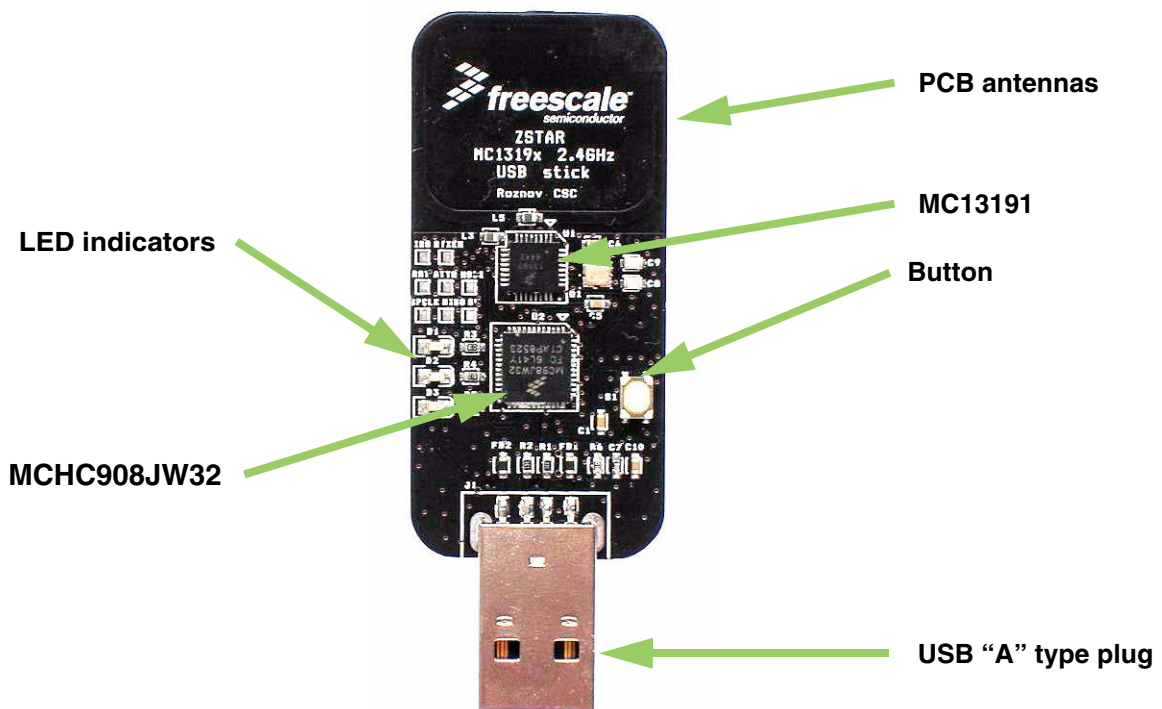


Figure 4-1. USB Stick Board Overview

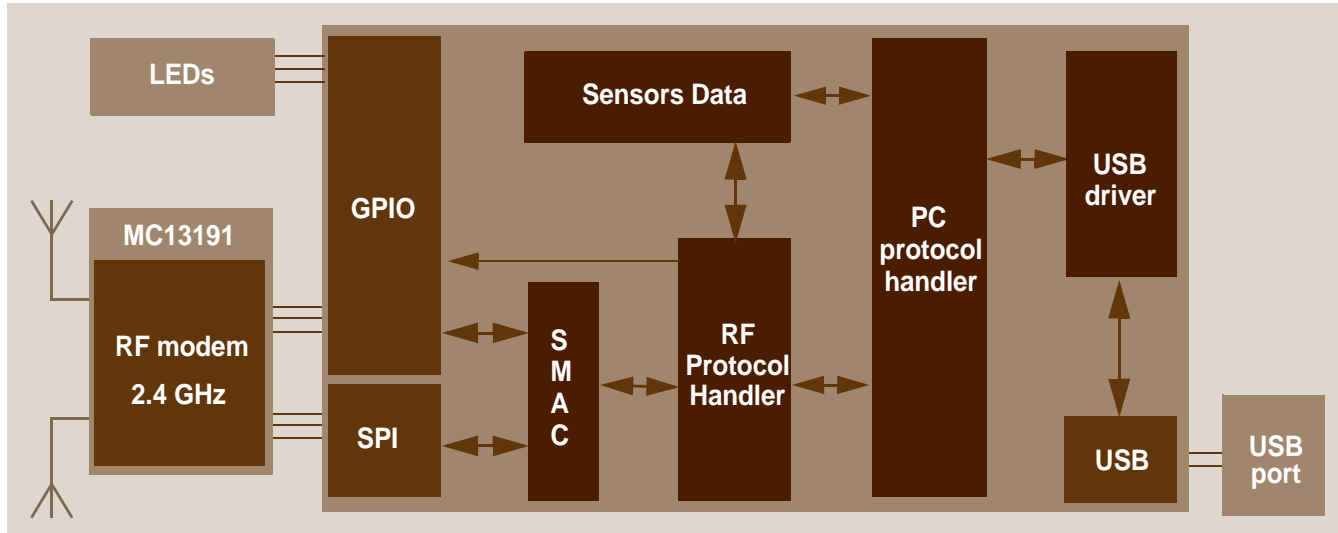


Figure 4-2. The USB Stick Software Overview

Figure 4-2. The USB Stick Software Overview shows, in more detail, how different software and hardware modules co-operate with each other. There are two main tasks of the USB Stick board:

- receive the data from the MC13191 transceiver and store it in RAM buffer
- handle the USB module communication, decode and provide the data from the RAM buffer.

These two are somewhat independent and the only common point between them is the accelerometer and button data buffer in RAM. The RF software communicates with the Sensor Board and retrieves the latest accelerometer data. This is stored in RAM and can be independently read by the PC application via the USB link. The protocol employed on the PC side is just a subset of the Original ZSTAR that is built on simple STAR protocol used in the original RD3112MMA7261Q demo. The protocol is described in section [Section 5.4, ZSTAR3 USB Protocol - Extended STAR Protocol](#).

For the USB Stick board operation, several MCHC908JW32 hardware modules are used: USB 2.0 Full-speed (USB), Synchronous Peripheral Interface (SPI), Keyboard Interrupt (KBI) and a General Purpose Input/Output (GPIO).

Chapter 5

Software Design

5.1 Introduction

This section describes the design of the ZSTAR3 software blocks. The software description comprises these topics:

- [Section 5.2, Simple Media Access Controller \(SMAC\)](#) modifications description
- ‘Air’ [Section 5.3, The ZSTAR3 RF Protocol](#) description
- Serial [Section 5.4, ZSTAR3 USB Protocol - Extended STAR Protocol](#) description
- AN2295 [Section 5.6, Bootloader](#) (over USB) implementation notes

5.2 Simple Media Access Controller (SMAC)

The SMAC is a simple ANSI C based code stack available as sample source code which can be used to develop proprietary RF transceiver applications using the MC1319x and MC1321x.

5.2.1 SMAC Features

- Compact footprint:
 - 4K FLASH
 - 10 bytes (+ maximum packet length) RAM
 - As low as 16 kHz bus clock
- Can be used to demonstrate coin cell operation for a remote control
- MC1319x/MC1321x compatible
- Very-low power, proprietary, bi-directional RF communication link
- ANSI C source code targeted at the HCS08 core and portable to almost any CPU core (including 4-bit)
- Low priority IRQ
- Sample application included, extremely easy to use
- Liberally commented

5.2.2 Modifications of SMAC for ZSTAR3 RF Protocol

The development of the ZSTAR software is based on the free SMAC stack available from Freescale. The SMAC version used was 4.2. To the SMAC has been added three new target files for the Original USB stick / sensor and for the new ZSTAR3 sensor board. Furthermore, a couple of new functions were added to this SMAC version.

A fully detailed description of the SMAC is in the SMAC Reference Manual (SMACRM.pdf), available together with SMAC source code.

5.2.2.1 New Targets Added to SMAC

A modification of the SMAC for individual application were done by target files that exact defines all pins and peripherals. For the ZSTAR3 project purposes was added to the SMAC three new targets:

- MC1319XZSTAR_USB.c/h - these files describes the Original ZSTAR USB stick board
- ZSTAR_SENSOR.c/h - these files describes the Original ZSTAR and the ZSTAR2 Sensor board
- MC1321XZSTAR3.c/h - these files describes the ZSTAR3 Sensor board

5.2.2.2 New Functions Added to SMAC

The original SMAC doesn't have any advanced transceiver time operations implemented, such as delayed transmit / receive and others. The ZSTAR3 RF protocol is based on these functionalities, and so they had to be implemented. A list of the new functions in the SMAC MAC layer:

- **UINT32 MLMEGetActualTime(void)** - function returns the current value of the freely running main counter in the modem
- **UINT32 MLMEGetTimeStamp(void)** - function returns the time value of last received message
- **UINT32 MLMEComputeDelay(UINT32 u32StartTime, UINT32 u32Delay)** - function computes a new time with delay
- **UINT8 MLMEDelayTransceiver(UINT32 u32Time, UINT32 u32TimeOut, tTxPacket *psPacket, tRxPacket *psRxPacket, UINT8 u8mode)** - the most important new function. This function provides delayed transmit and receive with time-out, doze modes, and general time interrupt generation

5.3 The ZSTAR3 RF Protocol

The ZSTAR3 uses a simple time based protocol for an RF transfer of information between sensor boards and the USB receiver. The protocol is based on a simple star topology for a communication network, with one master point (USB stick) and slaves (sensor boards). It provides time slots for up to 16 sensors. Main data of a load contains acceleration (X, Y and Z axis) and basic status data. The protocol is built on top of modified [Section 5.2, Simple Media Access Controller \(SMAC\)](#) drivers that are available for the MC1319x and MC1321x transceivers families. It is bidirectional, allowing the set up of independent connections for up to 16 ZSTAR3 demos, together in one RF space (thus, up to 256 sensor boards).

All data is transferred in so-called Zpackets. This protocol is primarily targeted at simple demo purposes, allowing a fast transfer of the accelerometer data in short packets with minimum overheads and with minimum battery loads (most of the receive windows eliminated, short transmit packets, etc.).

5.3.1 ZSTAR3 RF Protocol Features

- Based on modified Freescale SMAC library
- Star network topology
- One coordinator for the network (master) - NOT ZigBee coordinator
- Support for up to 16 sensors (slaves)
- Network main period is 30Hz (33.333ms)
- Data capture designed network
- Identification Network number (NetNum)
- Automatic/Manual selection of communication channels
- Collect all RSSI (Receive Signal Strength Indicator) information on communication

5.3.2 Zpacket Format

The ZSTAR Zpacket is contained inside the MC1319x/MC1321x standard packet structure, which is consistent with the IEEE 802.15.4 Standard. The SMAC library transparently adds a 16 bit Packet control field (see chapter 7.2.1.1 of IEEE 802.15.4 Standard specifications) to differentiate packets from ZigBee and other standards.

The Zpacket becomes a payload data for the SMAC standard packet and contains the following fields:

- [Section 5.3.2.1, Network Number \(NetNum\)](#)
- [Section 5.3.2.2, RX Strength](#)
- [Section 5.3.2.3, Zcommand](#)
- [Section 5.3.2.4, Zdata](#)

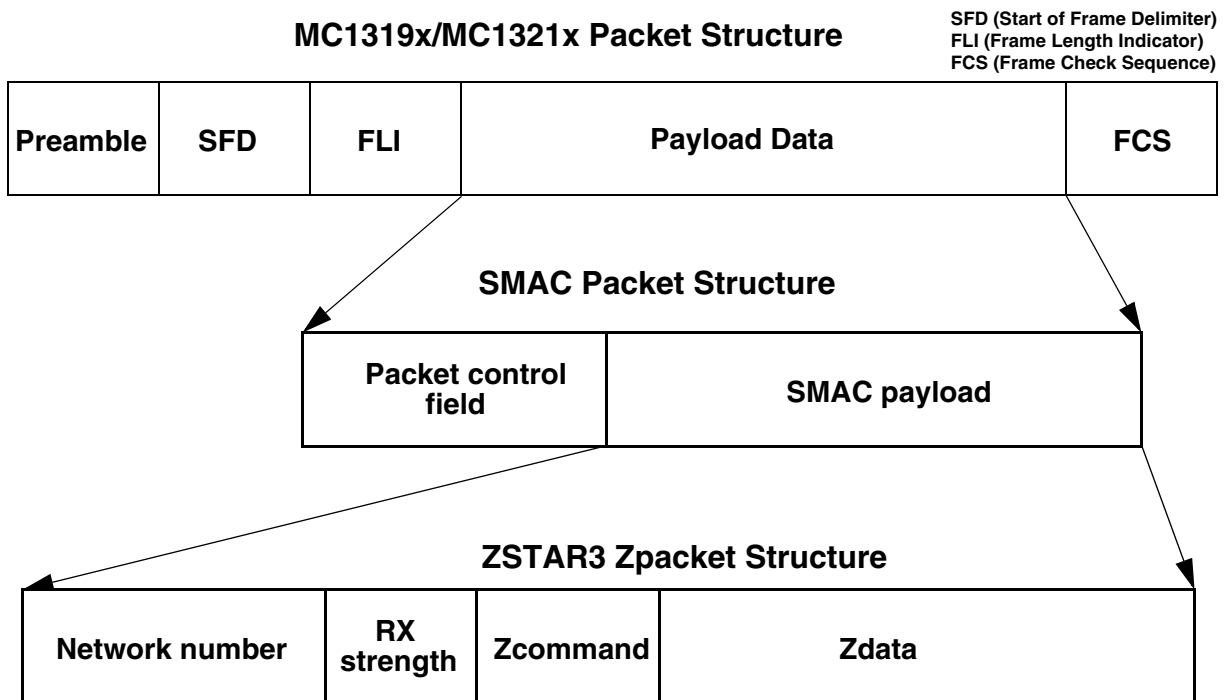


Figure 5-1. Zpacket Format

5.3.2.1 Network Number (NetNum)

The ZSTAR3 uses for RF network identification an ID number called the NetNum. It is placed in FLASH memory of each participant of the ZSTAR3 RF protocol. The USB stick randomly generates it at the first time of its life cycle and it can be changed by a command from the PC. The sensor board has this number erased from manufacture and it is set up with the first valid packet received in the ZSTAR3 RF protocol format. The network number in the sensor board can be cleared by pressing all the buttons together during a board wake up sequence or by a command from the USB stick. Packets with different Network numbers are simply ignored.

This field is 16 bits long.

5.3.2.2 RX Strength

This field reports the strength of the last received packet on the other end of the connection. This value simply tells us how well the other side receives ‘our packets’. This can be used by transmission power management functions to change the transmission power so the other party receives packets with enough strength.

The values reported are retrieved using the `MLMELinkQuality()` SMAC primitive.

This field is 8 bits long.

5.3.2.3 Zcommand

The ZSTAR3 demo protocol uses a couple of simple commands to establish and maintain the data flow between the Sensor Board and the USB stick.

The command is carried in [Section 5.3.2.3, Zcommand](#) field and is 8 bits long. The commands are defined as listed in [Table 5-1., Original ZSTAR Commands Zcommand List](#), [Table 5-2., ZSTAR3 Main Commands Zcommand List](#), [Table 5-3., ZSTAR3 Subcommands Zcommand List](#).

Table 5-1. Original ZSTAR Commands Zcommand List

ZCommand	ZCommand code	Direction	Zdata
ZSTAR_BROADCAST	'b' (0x62)	USB stick to Sensor Board	none
ZSTAR_ACK	'a' (0x61)	USB stick to Sensor Board	none
ZSTAR_CALIB	'k' (0x6B)	USB stick to Sensor Board	calibration data to Sensor Board
ZSTAR_STATUS	's' (0x73)	USB stick to Sensor Board	g-range selection data to Sensor Board
ZSTAR_CONNECT	'c' (0x63)	Sensor Board to USB stick	calibration data from Sensor Board
ZSTAR_DATA	'd' (0x64)	Sensor Board to USB stick	accelerometer values, temperature and bandgap voltage, button levels, g-range selection

Table 5-2. ZSTAR3 Main Commands Zcommand List

ZCommand	ZCommand code	Direction	Zdata
ZSTAR3_PYLON	0xA0	USB stick to Sensor Board	Status and optional subcommand
ZSTAR3_DATA	0x50	Sensor Board to USB stick	Accelerometers values, status info
ZSTAR3_ACK	0x60	Sensor Board to USB stick	none
ZSTAR3_CONNECT	0x40	Both	Calibration data, board info
ZSTAR3_ION	0x70	Sensor Board to USB stick	Optional return read data from ION

Table 5-3. ZSTAR3 Subcommands **Zcommand** List

ZCommand	ZCommand code	Direction	Zdata
ZSTAR3_GUI	'g' (0x67)	USB stick to Sensor Board	none
ZSTAR3_MODE	'm' (0x6D)	USB stick to Sensor Board	Set up data of new mode
ZSTAR3_OFF	'o' (0x6F)	USB stick to Sensor Board	none
ZSTAR3_GSEL	's' (0x73)	USB stick to Sensor Board	g-range selection data to Sensor Board
ZSTAR3_DATARATE	'd' (0x64)	USB stick to Sensor Board	new data rate
ZSTAR3_FLAG	'f' (0x66)	USB stick to Sensor Board	none

5.3.2.4 Zdata

The [Section 5.3.2.4](#), Zdata field follows the [Section 5.3.2.3](#), Zcommand field and may be empty if the actual command doesn't require any additional data. The data format is dependent on the [Section 5.3.2.3](#), Zcommand. A detailed description is in the next chapter.

5.3.3 Original ZSTAR Zcommand Description

For a more detailed description of original ZSTAR Zcommands check RD3152MMA7260Q reference design manual [ZSTAR RM, Reference Design Manual - Original ZSTAR Demo \(RD3152MMA7260Q\) RDM](#).

5.3.4 ZSTAR3 Protocol Zcommand Description

All commands in the direction sensor board to the USB stick are replenished (lower 4 bits) by information on the sensor index.

5.3.4.1 ZSTAR3_DATA

With this command the sensors send measured data on to the USB stick. This is the most frequently used sensor board RF command.

5.3.4.2 ZSTAR3_ACK

This command is used by the sensor to acknowledge successfully received USB stick subcommand.

5.3.4.3 ZSTAR3_CONNECT

With this command the sensor board sends calibration data on to the USB stick. The command is also used to start communication between the sensor board and the USB stick.

5.3.4.4 ZSTAR3_ION

This command is specially designed to meet all the special requirements of the digital sensor device¹.

5.3.4.5 ZSTAR3_PYLON

This is the main, and practically, the only command of the USB stick. This command is used as a net synchronization message and it's the only way to get data and other sub commands to the individual sensor boards. This command is replenished in protocol with mode of operation information (8/16 bits communication). The Subcommand and data in the subcommand fields are not mandatory.

Zdata bytes: 0 1 2 3 4...

List of active sensors	Selected sensor	Subcommand	Data depends on sub commands
------------------------	-----------------	------------	------------------------------

Figure 5-2. ZSTAR3_PYLON Zdata format

5.3.5 ZSTAR3 Protocol SubCommands Description

This group of subcommands relates to the main USB stick RF [ZSTAR3_PYLON](#) command. All these subcommands can only be sent as an extension of [ZSTAR3_PYLON](#). Subcommands are the only way to get data and the majority of commands from the USB stick to individual sensors.

5.3.5.1 ZSTAR3_GUI

This command is designed only to clear the timeout counter² in the sensor board and thus keep the sensor board awake. The sensor board answers with the [ZSTAR3_ACK](#) command.

5.3.5.2 ZSTAR3_MODE

This command is used only with the ION device mounted on sensor board. The command switches individual digital work modes of the digital sensor. The sensor board answers with the [ZSTAR3_ACK](#) command.

5.3.5.3 ZSTAR3_OFF

This command switches off the sensor board. This is only available on those sensor boards without a mechanical power switch.

5.3.5.4 ZSTAR3_GSEL

This command switches the current g range of the sensor to a new one. The sensor board answers with the [ZSTAR3_ACK](#) command.

¹.ZSTAR3 with assembled MMA745xL - ION.

².Each sensor board without a mechanical power switch has an implemented automatic switch off function when a 2 minute timeout overflows.

5.3.5.5 ZSTAR3_DATARATE

This command switches the current data rate for reading the sensor data to a new one. The sensor board answers with a [ZSTAR3_ACK](#) command.

5.3.5.6 ZSTAR3_NETNUM

This command changes the NetNum (ID of the ZSTAR3 RF network) in the sensor board. The sensor board answers with the [ZSTAR3_ACK](#) command.

5.3.5.7 ZSTAR3_FLAG

This is a special additional subcommand that is used only with the initial ZSTAR_BROADCAST command to recognize an RF protocol type - Original ZSTAR RF or ZSTAR3 RF Protocol. It was added only for compatibility purposes between the ZSTAR3 demo and the original ZSTAR.

5.3.6 ZSTAR3 RF Protocol Description

The ZSTAR3 RF protocol is based on a star RF network topology. The USB stick is the master of the network and the sensor boards are slaves. The entire network runs within the time based RF protocol, thus each device in the ZSTAR3 RF network has its own time slot in each basic period of the network to transmit. Reception time is a little different, all the sensor boards open receive window at some time and try to catch synchronization/data transmission sent from the master, but the USB stick (master) opens a receive window for the whole time that a sensor boards transmits its own data.

The ZSTAR3 devices runs within simple state machine:

1. **Init state** - the devices initializes all necessary hardware peripherals and software drivers. After initialization is complete, the state machine jumps to the Broadcast state.
2. **Broadcast state:** this is a little bit outdated state that is kept only for compatibility of the ZSTAR3 protocol with the original ZSTARs running the original protocol.
 - **Master (USB stick)** - the ZSTAR3 protocol needs only to find out the free channel for the prepared RF network, but because the ZSTAR3 demo supports original ZSTAR RF protocol, then this state is important only for compatibility of the ZSTAR3 demo.
 - **Slave (sensor board)** - In this state the sensor board is looking for the ZSTAR3 communication. And when it finds the RF communication with the right NetNum number and its time position is free, the sensor board jumps to the Run State of the RF protocol. In other cases, it jumps to the Sleep state.
3. Compatibility state with original ZSTAR:
 - **Master (USB stick)** - the USB stick of the ZSTAR3 supports the original ZSTAR RF protocol and this state provides it. For more details check the manual for the original ZSTAR [ZSTARRM, Reference Design Manual - Original ZSTAR Demo \(RD3152MMA7260Q\) RDM](#).
 - **Slave (sensor board)** - the sensor board software contains this state and it can support the original ZSTAR too. All sensor boards are ordered with prohibited compatibility mode, because the compatibility mode causes a lot of problems when the demo runs in the same RF space as any original ZSTAR.

4. ZSTAR3 Protocol run state:
 - **Master (USB stick)** - the master is periodically transmits synchronization `ZSTAR3_PYLON` messages and then is receives all incoming messages from the sensor boards within the same NetNum. In this state the master collects all necessary information about connection status of all the sensor boards and other sensor payloads.
 - **Slave (sensor board)** - In this state the slave periodically opens a receive window to catch a master synchronization message, measures sensor data and transmits the measured data within its own time slot window to the air (master).
5. Sleep state:
 - Master (USB stick) - Doesn't have this state.
 - Slave (sensor board) - When the sensor board goes into this state, it switches off all peripherals on the board and inside the microcontroller and prepares the entire board for the least power consumption.

5.3.6.1 Typical One Period of ZSTAR3 RF Protocol

The ZSTAR3 RF protocol (in normal run mode) period starts with a `ZSTAR3_PYLON` message from the master. In an ideal case, all active slaves have an opened receive window and catch the incoming `ZSTAR3_PYLON` message from the master. When the slaves receive this synchronization message, all measure sensor output values at the same time. Then they prepare future transmissions, each in its unique time. This time depends on the index of each slave device. In the meantime, the master prepares a new receive window and opens it right before the first prospective message from Slave 0, and this window stays open until the time for all 16 devices elapses. When the master catches any message from the slaves, it processes it and immediately opens a new receive window if the last message isn't from the 16th Slave device.

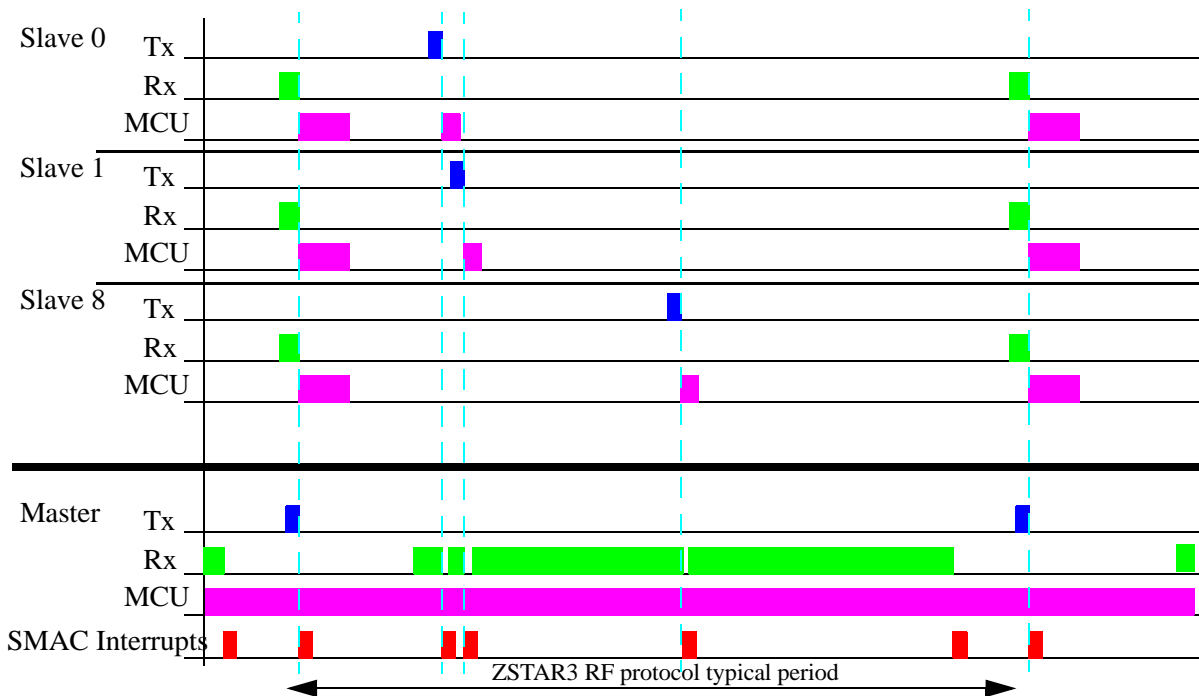


Figure 5-3. ZSTAR3 RF Protocol Period

5.4 ZSTAR3 USB Protocol - Extended STAR Protocol

The ZSTAR3 demo uses a subset of the original ZSTAR demo protocol commands. The Original ZSTAR is built on the STAR demo protocol. This way, most of the software originally developed for the RD3152MMA7260Q (ZSTAR) and RD3112MMA7260Q (STAR) is also usable with the ZSTAR3 demo.

The STAR demo communicates over the RS232 serial line with a simple text-based protocol. The same protocol is used in ZSTAR/ZSTAR3 for communication between the USB stick and a PC (over a virtual serial port). The PC application sees the same interface (serial port) and the same protocol as if a STAR demo was connected. The ZSTAR3 uses extended STAR/ZSTAR protocol to provide all the new features.

Because the ZSTAR3 can serve up to 16 connected sensor boards, all commands that work with the sensor boards are communicated with one selected board by the 'N' command from [Section 5.4.3, Subset of Newly Added ZSTAR3 Protocol Commands](#).

5.4.1 Subset of Original STAR Protocol Commands

The ZSTAR3 software contains all unchanged original STAR commands. Some commands can send extended information in the ZSTAR3 mode - the ZSTAR3 USB protocol extends some original commands when the PC sends any command from a new subset of the command table. This extended information returns to the PC information resulting from individual commands.

Table 5-4. Legend of Colors

Legend	
Red	Direction PC -> USB stick
Blue	Direction USB stick -> PC, regular data
Magenta	Direction USB stick -> PC, extended data

Table 5-5. Original STAR Commands

Normal Command			Extended Asynchronous Answer		Note
Command	Detail of command data	Length tx/rx (B)	Detail	Length rx (B)	
'R'	'N'	1, 1	-	-	Reset to 8 bit mode
'V'	"x' value 'y' value 'z' value	1, 6 ¹	-	-	Read acceleration data
'G'	g-select value	1, 1	-	-	g-select read
'g'	g-select value, 'G', result	2, 0, 2	{'g', result, index}	7	g-select write
'K'	"X' g(0) g(1) 'Y' g(0) g(1) 'Z' g(0) g(1)	1, 9 ²	-	-	read calibration values ³
'k'	xg(0), xg(1), yg(0), yg(1), zg(0), zg(1), 'K', result	7 ⁴ , 0, 2	{'k', result, index}	7	write calibration values ⁵

¹ Length of reply with measured data depends on current work mode, 8/16bits. 6 bytes for 8 bits, 9 bytes for 16 bit mode

² Length of reply with calibration values depends on current work mode, 8/16bits. 9 bytes for 8 bits, 15 bytes for 16 bit mode

³ If a connected sensor has the ZSTAR3 protocol, then this command returns ideal calibration constants

⁴ Length of the calibration data command depends on current work mode, 8/16bits. 7 bytes for 8 bits, 13 bytes for 16 bit mode

⁵ If a connected sensor has ZSTAR3 protocol, then the calibration data command isn't important

5.4.2 Subset of Original ZSTAR Protocol Commands

The ZSTAR3 software contains all slightly changed original ZSTAR commands.

Table 5-6. Legend of Colors

Legend	
Red	Direction PC -> USB stick
Blue	Direction USB stick -> PC, regular data

Table 5-7. Original ZSTAR Commands

Normal Command			Extended Asynchronous Answer		Note
Command	Detail of command data	Length tx/rx (B)	Detail	Length rx (B)	
'r'	'z'	1, 1	-	-	Reset to 16 bit mode
'v'	'x' value 'y' value 'z' value 's' status 't' temperature 'b' bangap	1, 17	-	-	Read extended 16 bits acceleration data ¹
'I'	Text information about project	1, -	-	-	Return text information about ZSTAR project
'U'	Debug information	1, -	-	-	Switch on sending debug information
'u'		1, 0	-	-	Switch off sending debug information
'Q'	'Q'	2, -	-	-	Debug auto calibration command
'S'	Sensor RSSI, USB RSSI	1, 2	-	-	Read Receive Strength Signal Indicators

¹ ZSTAR3 RF protocol does not support bandgap and temperature values, so when the USB stick runs with the ZSTAR3 RF protocol this command always returns zeros in these fields

5.4.3 Subset of Newly Added ZSTAR3 Protocol Commands

This table describes all newly added ZSTAR3 commands. There are two tables, one for general commands and the second specially designed for digital sensors. Some commands returns extended information resulting from individual commands.

Table 5-8. Legend of Colors

Legend	
Red	Direction PC -> USB stick
Blue	Direction USB stick -> PC, regular data

Table 5-9. New ZSTAR3 Commands (general commands)

Normal Command			Note
Command	Detail of command data	Length tx/rx (B)	
'A'	'a', channel	1, 2	return actually selected RF channel
'a'	channel, 'A'	2, 1	set new RF channel, channel > 15 means automatically selection by USB stick
'B'	'b', Working State	1, 2	Get actual working state of ZSTAR
'C'	'c'	1, 1	Simple HandShake
'H'	'h', netnum	1, 3	Read Net ID number of RF communication
'h'	netnum, 'H'	3, 1	Set new Net ID number of RF communication
'M'	datarate, 'm', result	2, 2	Set new data rate of sensor
'm'	datarate	1, 1	Read actual data rate of sensor
'N'	sensor index, 'n', result	2, 2	Change focus of communication to new sensor index
'n'	sensor index, connected sensors mask	1, 3	Read index of focused sensor and mask of connected sensors
'O'	'X' calibration offset 'Y' calibration offset 'Z' calibration offset	1, 6	Get true calibration offset values generated by auto calibration process
'W'	sensor type, board type, sensor subtype, ZSTAR version	1, 5	Read information about type of board, sensor and USB stick
'X'	burst sensor mask, 'x', connected sensors mask	3, 3	Switch on burst mode of ZSTAR
'x'	'X'	1, 1	Switch off burst mode of ZSTAR
'Z'	'z'	1, 1	Go to Bootloader device

Table 5-10. New ZSTAR Commands (Digital Sensor part)

Normal Command			Extended Asynchronous Answer		Note
Command	Detail of command data	Length tx/rx (B)	Detail	Length rx (B)	
'J'	address of reg, length of data, new values of reg 1-6 bytes, 'j', result	3-9, 2	{'J',result,index}	7	Direct access into ION registers, write
'j'	address of reg, length of reading, 'J', result	3, 2	{'j',result,data 2 - 8 x2 Bytes}	9-23	Direct access into ION registers, read
'L'	LDTH, SetByte, 'l', result	3, 2	{'L',result,index}	7	Level threshold detection switch on
'l'	'L', result	1, 2	{'l',result,index}	7	Level threshold detection switch off
'P'	PDTH, PW, LT, TW, Setbyte, 'p', result	6,2	{'P',result,index}	7	Pulse detection switch on
'p'	'P', result	1, 2	{'p',result,index}	7	Pulse detection switch off
'>'	Digital interface, '<', result	2, 2	{'>',result,index}	7	Change type of digital interface on communication with ION
'<'	'>', Digital interface	1, 2	-	-	Read type of digital interface on communication with ION

5.4.4 Burst Mode

A new mode for the ZSTAR USB communication is burst mode. The burst mode is designed to simplify the reading process of new acceleration data. In the burst mode, the USB stick sends all newly received acceleration data from the individual enabled sensor boards without any request command from the PC.

Burst mode content:

- time of receipt - 24 bits time information with a 4 μ s step
- manage byte - contains the current data rate, mode 8/16bits and the index of the sensor board
- acceleration data + data status - this field can be repeated, to repeat up to 4 samples depending on the current data rate. This is the main part of the burst mode packet.
- Status - contains information on the selected g range, buttons status and events of the last sample

The Burst mode frame format examples:

[tttttmmxxyzzdss] - 8 bits 30 Hz frame

[tttttmmxxxxxyyyzzzzdss] - 16 bits 30 Hz frame

[tttttmmxxxxxyyyzzzzdxxxxxyyyzzzzdxxxxxyyyzzzzdxxxxxyyyzzzzdss] - 16 bits 120 Hz frame - longest case of burst communication

where: t - receive time, m - manage byte, x- acceleration in X axis, y- acceleration in Y axis, z- acceleration in Z axis, d - data status byte, s - frame status byte, [- start char, and] is end char of the frame.

Real example of a burst mode frame:

”[BBBCDCA0D0001400400000D6002C00410000D9002D003D0000E1002F003D000020]”.

5.4.5 Automatic Self-Calibration

For the purpose of easier automatic calibration of the ZSTAR demo without a PC GUI, the additional Calibration command 'Q' (0x51) has been added. This command is usually issued over a terminal (e.g. HyperTerminal) software.

The user is required to place the Sensor Board into a horizontal position (for example on a desk), in which the Earth's gravity will induce a maximum acceleration in the Z axis. The first command 'Q' sent to the USB stick only prepares the user for the auto calibration process by text help, and the second command 'Q' in the line starts the auto calibration process of the sensor board.

5.5 Compatibility with Original ZSTAR

The new ZSTAR3 and the new USB stick sw are fully compatible with the original ZSTAR on USB communication, thus can be used in all application designed for the original ZSTAR. For example, the demo application RD3152MMA7261Q_SW. Compatibility was achieved by keeping all the original commands in the USB communication protocol.

The USB stick provides a compatibility mode with the original ZSTAR RF protocol. If the first received ZSTAR packet in broadcast mode is from a sensor board with original software, the USB stick is switched to the Compatibility mode and starts the original ZSTAR RF protocol. When compatibility mode is active, no other sensor can be connected.

5.6 Bootloader

There's bootloader software implemented in MCHC908JW32 microcontroller. The bootloader is based on [AN2295, Application Note - Developer's Serial Bootloader for M68HC08 and HCS08 MCUs](#) and AN2295SW related software. The original AN2295 bootloader targets serial connections between the PC and applications, and since the MCHC908JW32 implements a virtual serial port application, the USB version of the AN2295 bootloader has been created to allow reprogramming of Flash memory in the USB stick.

The USB virtual serial port software is fully described in [AN3153, Application Note - Using the Full-Speed USB Module on the MCHC908JW32](#). The MCHC908JW32 bootloader implements the same virtual serial port but under a different PID (the PC sees that serial port as a different application from ZSTAR).

The bootloader drivers installation guide can be found in [Section 6.1.2, AN2295 Bootloader Drivers Installation](#).

5.6.1 Switch to Bootloader Procedure

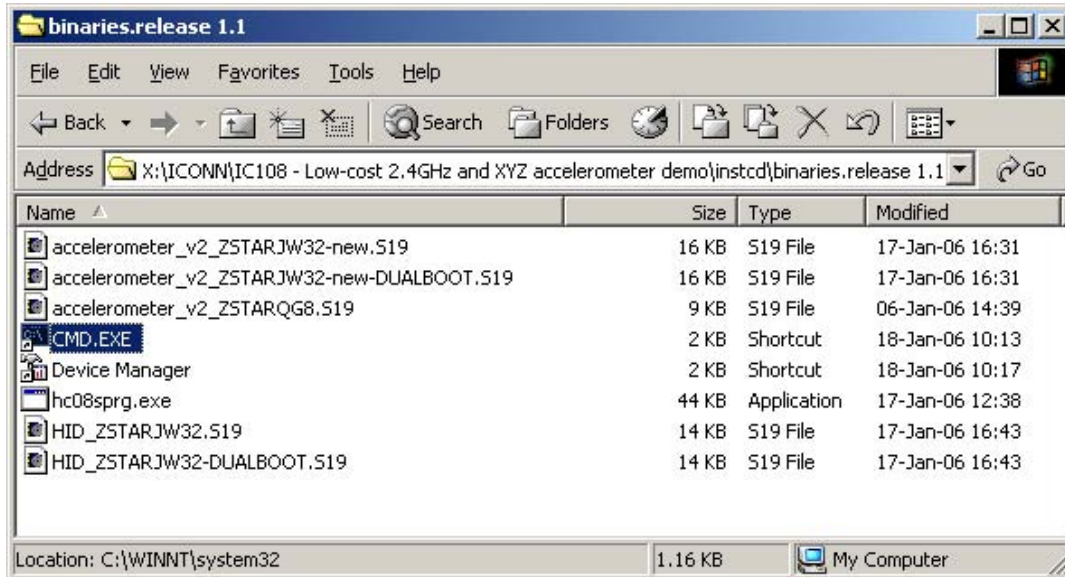
The Bootloader in the ZSTAR can be started by the following procedure:

1. Unplug the USB stick from the USB port
2. Press the button on the USB stick
3. Keep the button pressed and plug the USB stick back into the USB port
4. Release the button.

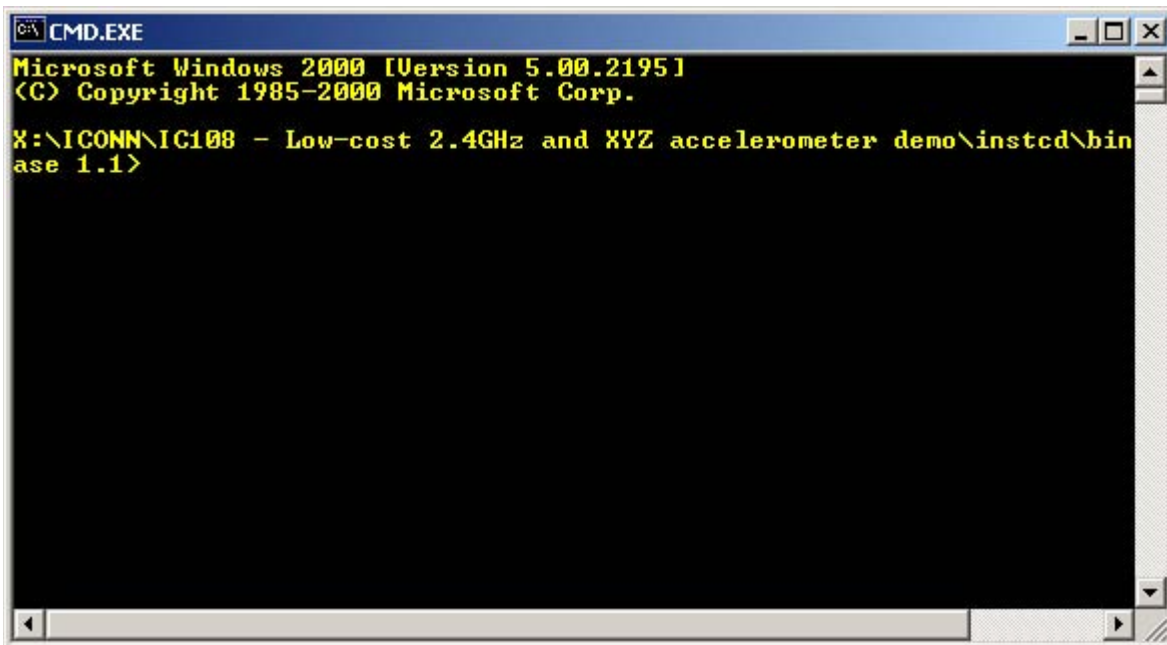
5.6.2 Bootloading Procedure

The easiest way to Flash the newest firmware onto a USB stick is to use the ZSTAR3 GUI application and through this application open a Bootloader COM port¹. But if you want to proceed to the Bootloader procedure manually, then follow the next steps.

1. Find on the installation CD the folder with binaries:



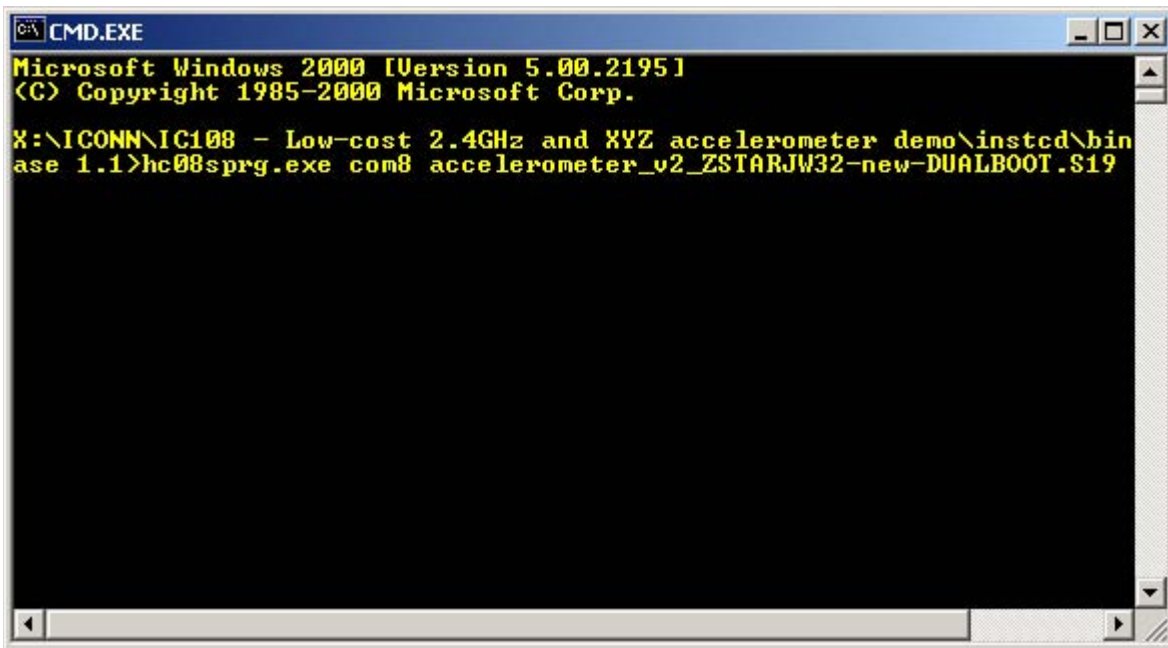
2. Start (double-click) the CMD.EXE shortcut, and a command line window should appear:



¹ For more details check Chapter 7 of this Manual.

- Now type: `hc08sprg [bootloader com port number] [binary (S file) that you want to bootload]`, just like this:

```
hc08sprg.exe com8 accelerometer_v2_ZSTARJW32-new-DUALBOOT.S19
```

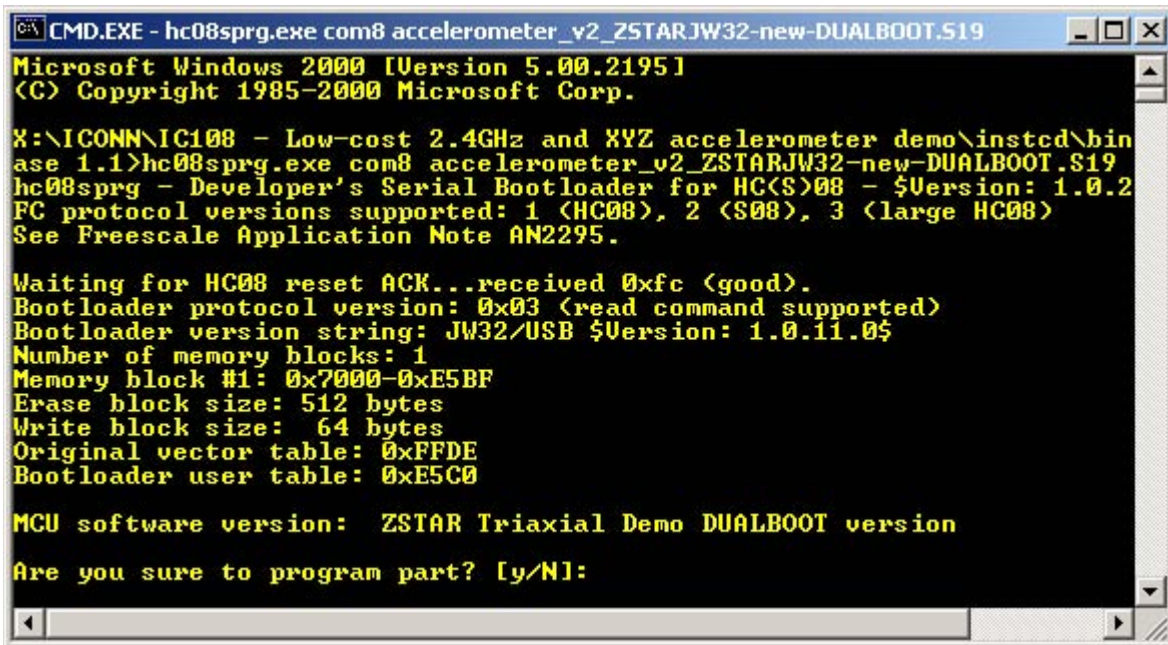


```

C:\> CMD.EXE
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

X:\ICOMM\IC108 - Low-cost 2.4GHz and XYZ accelerometer demo\instcd\bin
ase 1.1>hc08sprg.exe com8 accelerometer_v2_ZSTARJW32-new-DUALBOOT.S19
  
```

- Press ENTER and initial bootloader communication will start:



```

C:\> CMD.EXE - hc08sprg.exe com8 accelerometer_v2_ZSTARJW32-new-DUALBOOT.S19
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

X:\ICOMM\IC108 - Low-cost 2.4GHz and XYZ accelerometer demo\instcd\bin
ase 1.1>hc08sprg.exe com8 accelerometer_v2_ZSTARJW32-new-DUALBOOT.S19
hc08sprg - Developer's Serial Bootloader for HC(S)08 - $Version: 1.0.2
FC protocol versions supported: 1 (HC08), 2 (S08), 3 (large HC08)
See Freescale Application Note AN2295.

Waiting for HC08 reset ACK...received 0xfc (good).
Bootloader protocol version: 0x03 (read command supported)
Bootloader version string: JW32/USB $Version: 1.0.11.0$
Number of memory blocks: 1
Memory block #1: 0x7000-0xE5BF
Erase block size: 512 bytes
Write block size: 64 bytes
Original vector table: 0xFFDE
Bootloader user table: 0xE5C0

MCU software version: ZSTAR Triaxial Demo DUALBOOT version

Are you sure to program part? [y/N]:
  
```


If this screen does not appear, remove the USB stick and start from the beginning.

```

C:\>ase 1.1>hc08sprg.exe com8 accelerometer_v2_ZSTARJW32-new-DUALBOOT.S19
hc08sprg - Developer's Serial Bootloader for HC(S)08 - $Version: 1.0.2
FC protocol versions supported: 1 (HC08), 2 (S08), 3 (large HC08)
See Freescale Application Note AN2295.

Waiting for HC08 reset ACK...received 0xfc (good).
Bootloader protocol version: 0x03 (read command supported)
Bootloader version string: JW32/USB $Version: 1.0.11.0$
Number of memory blocks: 1
Memory block #1: 0x7000-0xE5BF
Erase block size: 512 bytes
Write block size: 64 bytes
Original vector table: 0xFFDE
Bootloader user table: 0xE5C0

MCU software version: ZSTAR Triaxial Demo DUALBOOT version

Are you sure to program part? [y/N]: y
Memory programmed: 100%
Memory verified: OK

X:\ICONN\IC108 - Low-cost 2.4GHz and XYZ accelerometer demo\instcd\bin
ase 1.1>

```

The bootloader disappears (in Device Manager) and the newly loaded software starts to execute.

Using this procedure, the software in the USB stick can be changed anytime.

5.7 Tri-application Software on the USB Stick

The USB stick provides three different USB devices that can be changed in run-time. A list of the ZSTAR3 USB Devices:

- CDC (Communication Device Class) - virtual serial port.
- HID (Human Interface Device) - Mouse device
- HID (Human Interface Device) - Keyboard device.

5.7.1 CDC - Virtual Serial Port Application

This is the main application of the ZSTAR3 project. Only this application can provide the ZSTAR3 USB protocol (extended STAR protocol) commands and run all the functions and features of the ZSTAR3 demo. When CDC application is active LED3 blinks are depending on USB communication activity.

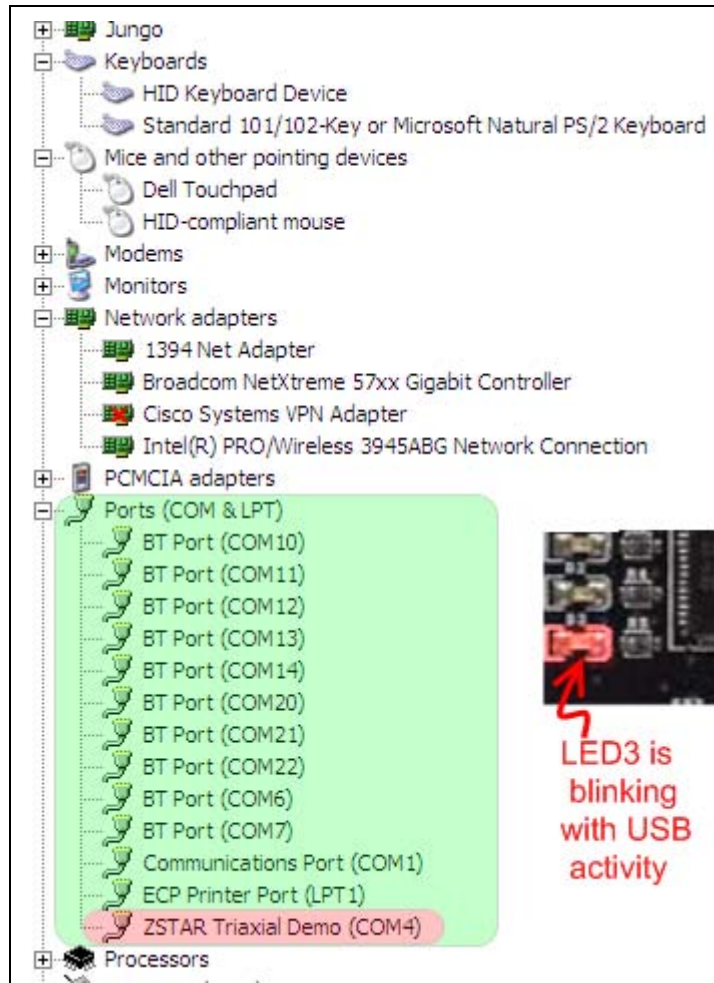


Figure 5-4. CDC application signs (device manager, indication LED)

5.7.2 HID - Mouse Application

A second application of the ZSTAR USB stick is a computer mouse demo. This application looks like a real USB mouse. Movement of the cursor on screen is done very simply by tilting the sensor board in axes XY. The first two buttons on the sensor board have the same functions as buttons on a real mouse. The third button, introduced on the new ZSTAR3 sensor boards, substitutes the wheel of a real mouse. When this button is pushed, the cursor stops moving on the screen, and the USB stick starts to generate wheel movement from the Y axis. Only the sensor board with index 0 can work as an input device. When Mouse application is active, LED3 is blinking fast (15Hz)

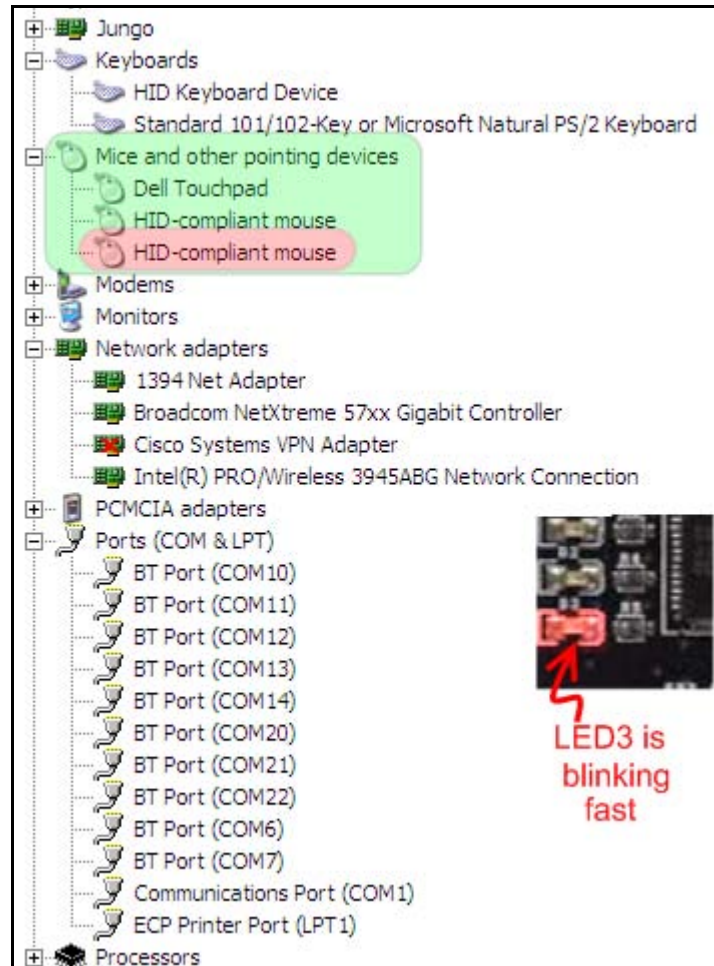


Figure 5-5. Mouse application signs (device manager, indication LED)

5.7.3 HID - Keyboard Application

The third application has a very specific target end application, mainly as a controller for simple PC games that are controlled by the keyboard. This application looks like a general USB keyboard. The USB stick simulates the pressing of keys on the keyboard through the tilt of the sensor boards. The ZSTAR3 recognizes four steps of tilt and can thus generate the frequency and duty cycle of the pressed keys. This feature helps for better game control. The ZSTAR in this mode supports the first 6 sensor boards indexes. Each index has a four key assignment shown in the table below:

Index	Up	Down	Left	Right
0	w	s	a	d
1	t	g	f	h
2	i	k	j	l
3	z	x	c	v
4	b	n	m	p
5	1	2	3	4

A very nice freeware game that we used for testing is GeneRally from the <http://generally.rscsites.org/> site. When Keyboard application is active, LED3 is blinking slow (4Hz).

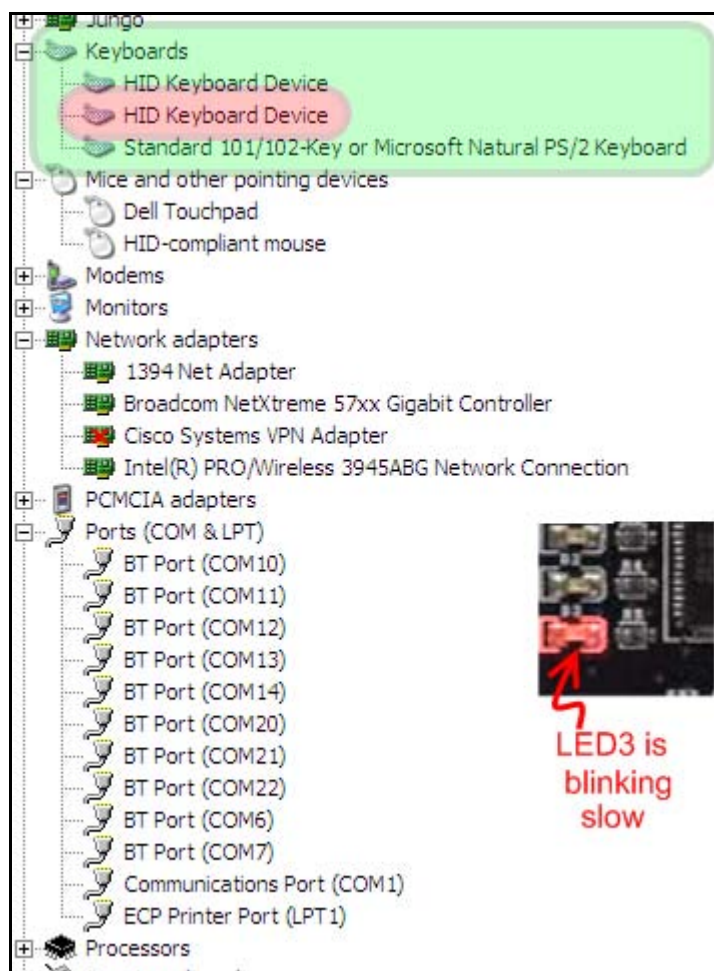


Figure 5-6. Keyboard application signs (device manager, indication LED)

5.7.4 Application Switching

All embedded applications can be switched by just a quick press of the button on the USB stick (with the USB stick inserted into the USB slot). The applications will appear and disappear accordingly.

In order for the 'tilt' mouse application to work, the sensor board must be calibrated correctly (e.g. using RD3152MMA7261Q_SW.exe, The ZSTAR3 GUI application or the [Section 5.4.5, Automatic Self-Calibration](#) procedure).

5.8 ZSTAR PC library

The ZStarlib library (zstarlib.dll) is tool which provides API (Application Interface) to manage all ZSTAR functionalities as reading sensor output values, setting up all the ZSTAR3 parameters, managing sensors and also provides the various events (data ready, calibration end, ZSTAR state changed etc.).

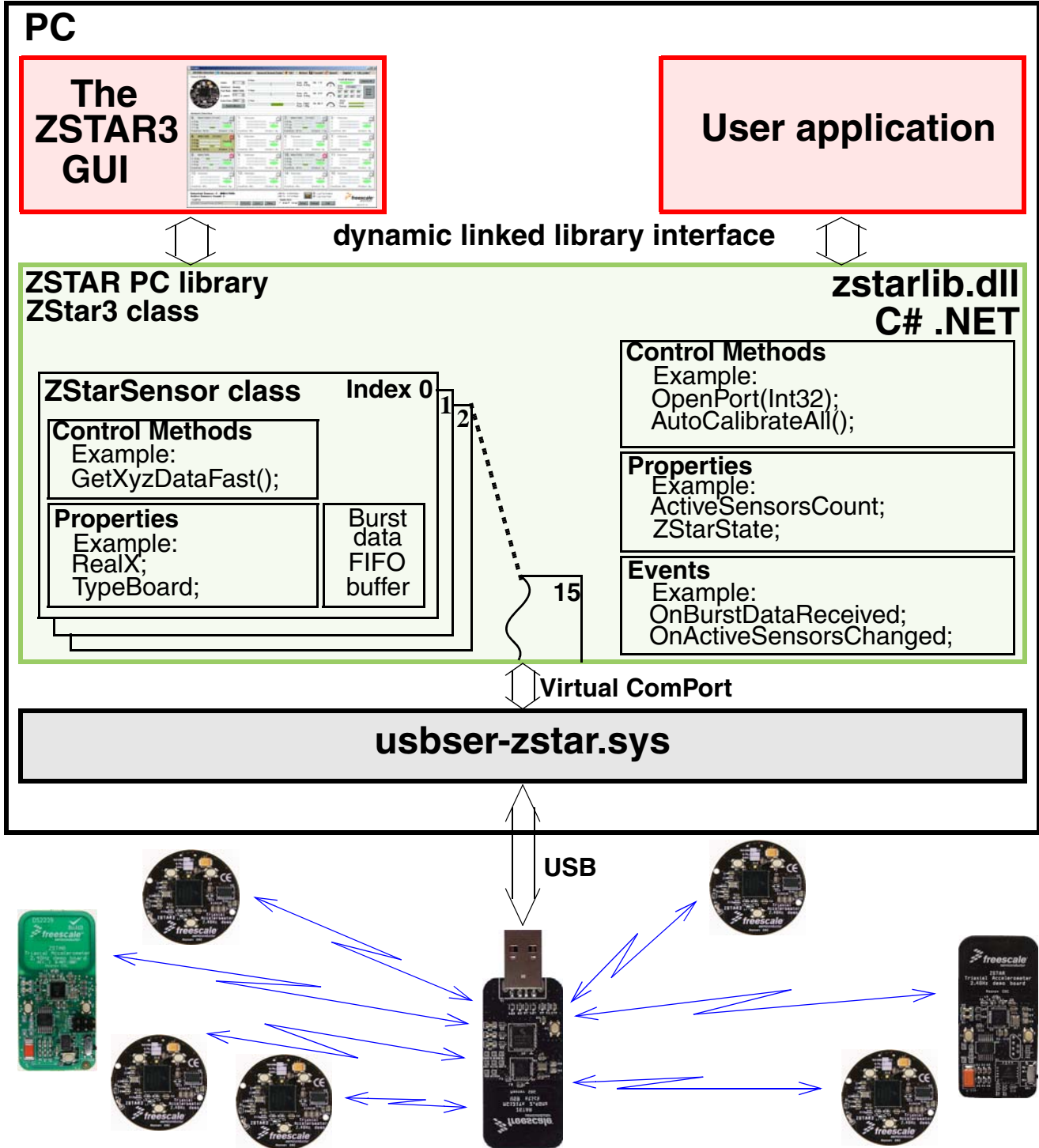


Figure 5-7. PC Software Composition Overview

It was developed in C# language, so it is based on Microsoft .NET Framework. The ZStarlib package also contains help file and source code of the simple demo application. The ZSTAR3 GUI application is based on this library and shows the most of its possibilities.

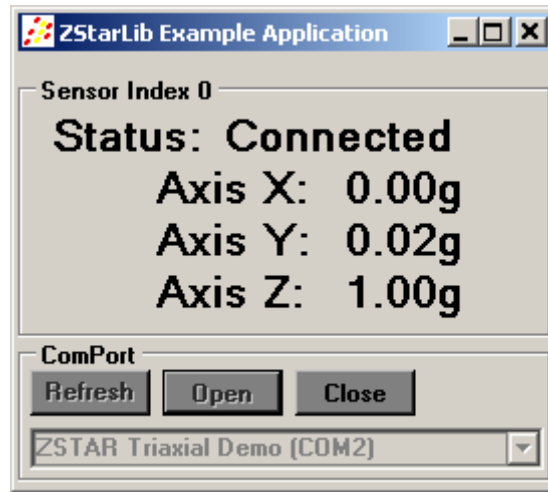


Figure 5-8. Example Demo Application Based on zstarlib.dll

Chapter 6

Application Setup

6.1 ZSTAR3 Installation Procedure

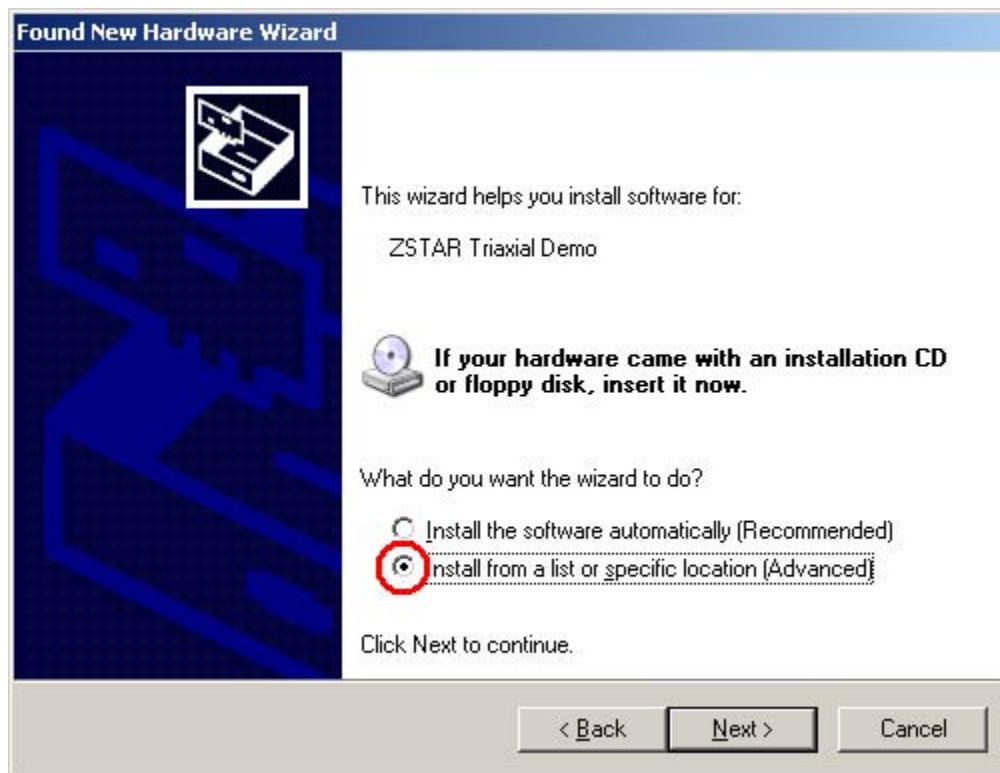
6.1.1 USB Stick Installation

First of all, we have to install the USB Stick on your PC. Please follow the next steps.

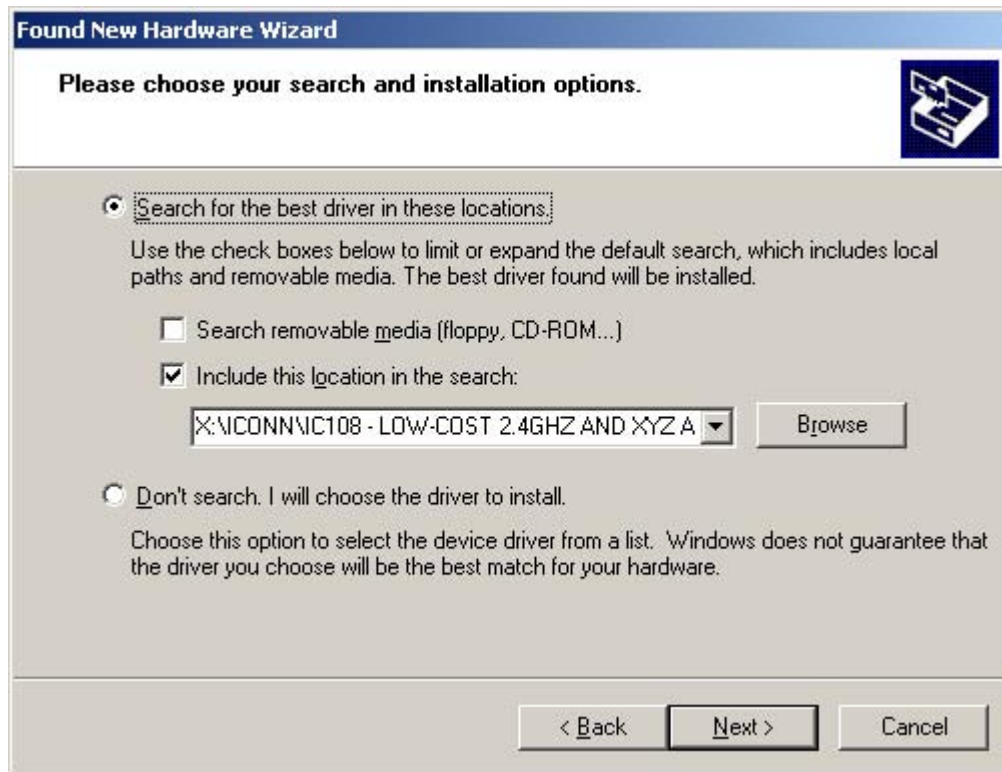
1. Plug the USB Stick into a USB slot. The ‘Found New Hardware’ announcement should appear:



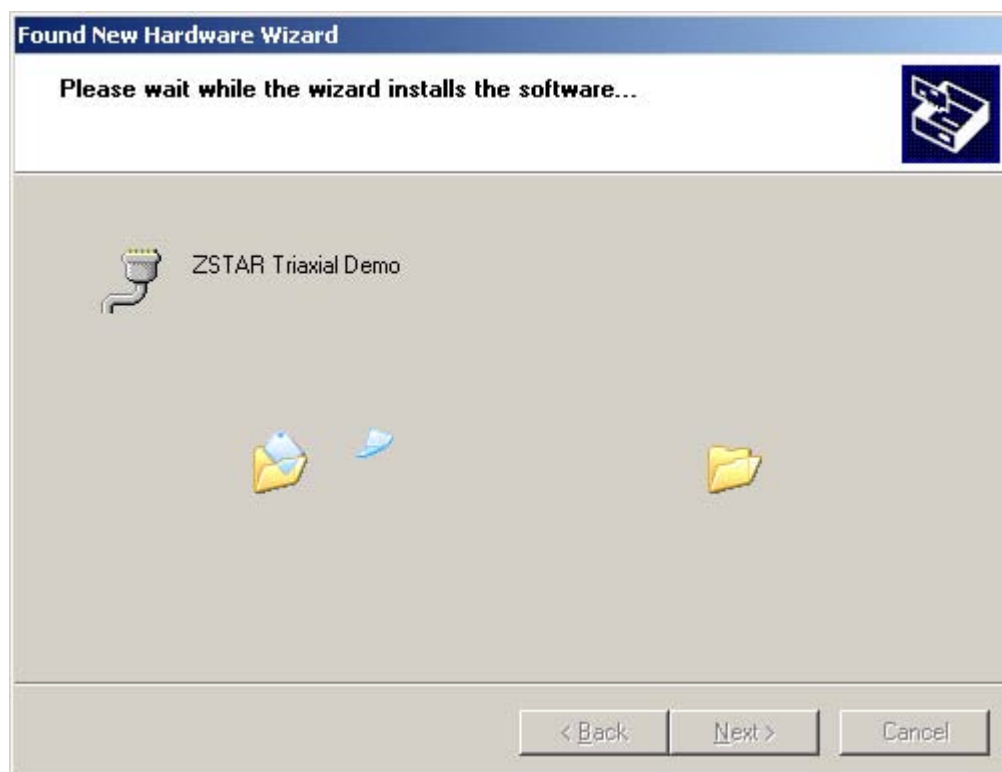
2. Then the installation wizard starts for new hardware. Choose “Install from a list or special location”



3. Point to the Installation CD as the driver path:



4. Installation should continue:



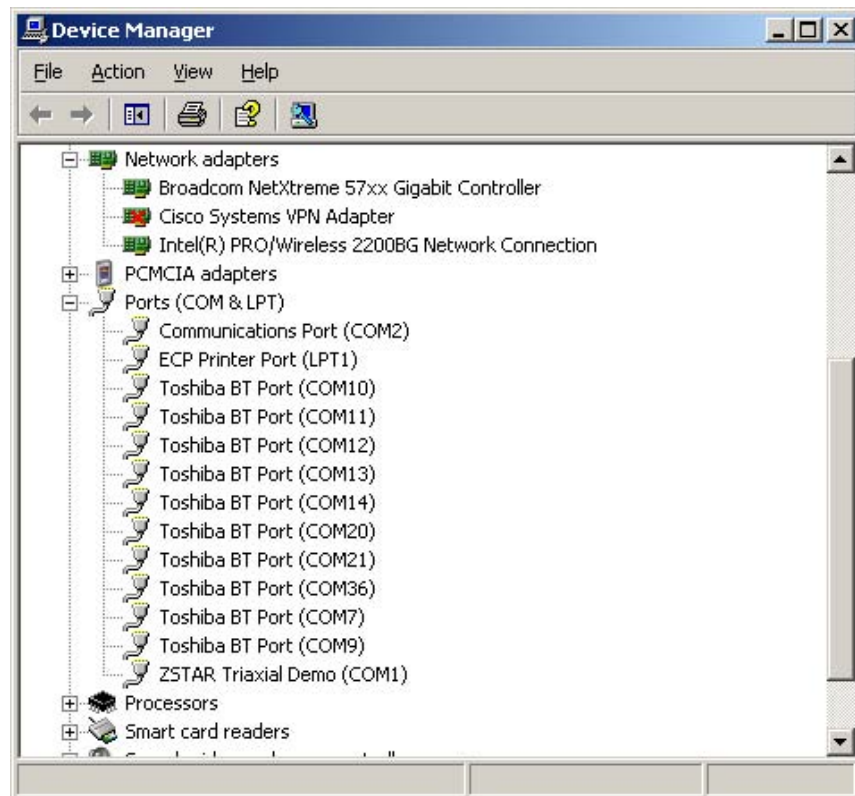
5. If you are asked to stop or continue installation because the drivers are not certified by Microsoft, select the “Continue Anyway” button.



6. Installation should successfully finish.



7. Check whether a new serial port (ZSTAR Tri-axial Demo) has appeared in your Device Manager (Right click **My Computer** on the Desktop > **Properties, Hardware** tab, **Device Manager** button):



6.1.2 AN2295 Bootloader Drivers Installation

This procedure assumes that ZSTAR Demo drivers are already installed. The drivers are false common for the bootloader (= are already present in Windows folders). If not, the procedure will be identical to the ZSTAR drivers installation.

1. Press the Button on the USB Stick and insert it into a USB connector (keeping the button pressed when inserted).

The following window appears:

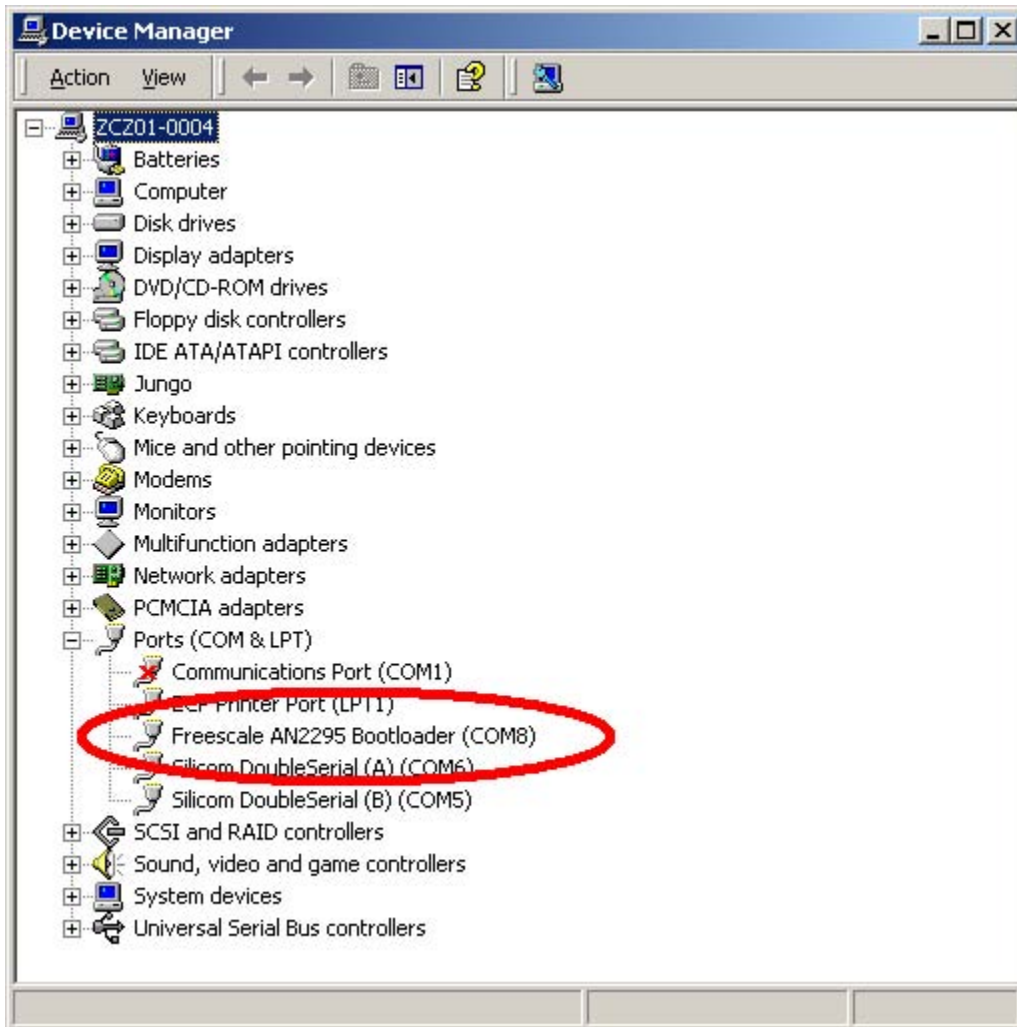


2. The PC searches for an appropriate driver (as in the ZSTAR Demo, in some instances a folder with the drivers (`star.inf` and `usbser-zstar.sys`) needs to be selected), then the following window should appear:



3. Just click **Yes**, and the bootloader port will be installed (as seen in the Device manager):
4. Right click **My Computer** on the Desktop > **Properties**, **Hardware** tab, **Device Manager** button.

5. A similar setup should be observed:



6. Note down the COM port number (here, COM8); this is the port number of the Bootloader

Once the software in the USB stick needs to be updated, the Bootloader can be invoked anytime, just by pressing the button while inserting the USB Stick into the USB slot.

Chapter 7 ZSTAR3 GUI

The ZSTAR3 demo brings a new GUI that supports all the new features and functions of the new ZSTAR3. It's distributed as a complete installation package with all the necessary files, including the Microsoft.NET Framework package.

7.1 Installation

1. Run the setup.exe file in the installation directory to start installing the ZSTAR3 GUI on the computer.

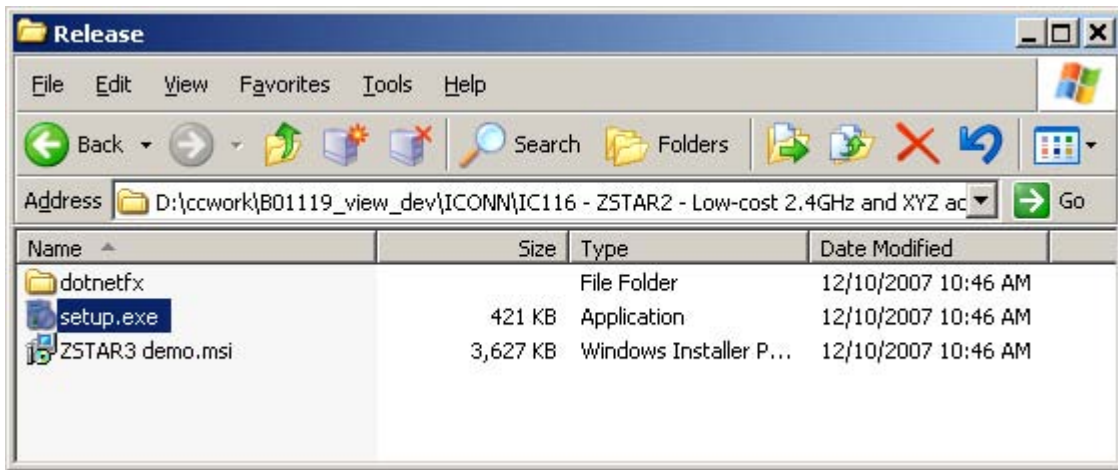


Figure 7-1. Installation Files

2. On first screen of ZSTAR3 installation process only press the “Next“ button.



Figure 7-2. Start of the Installation

3. In this step, select the destination directory and some other possibilities and press the “next” button. Please follow all the next steps to get to the end of the installation process.

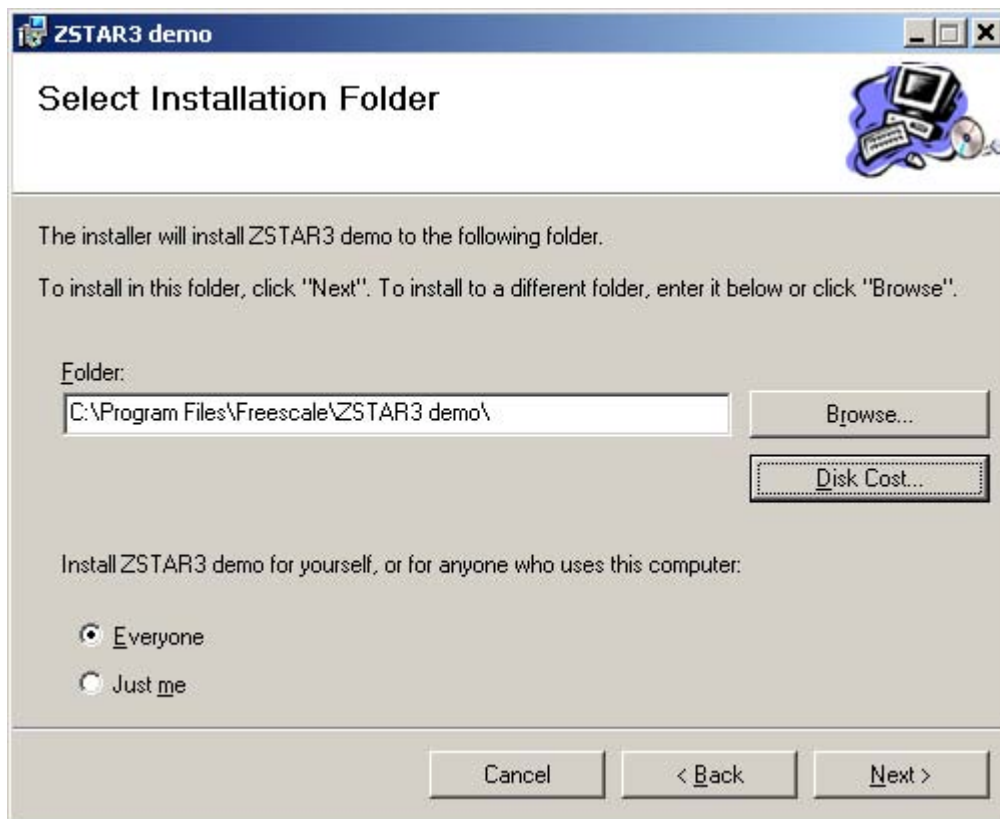


Figure 7-3. Second Step of the Installation Process - Select Destination Directory

4. Final step of the installation process.

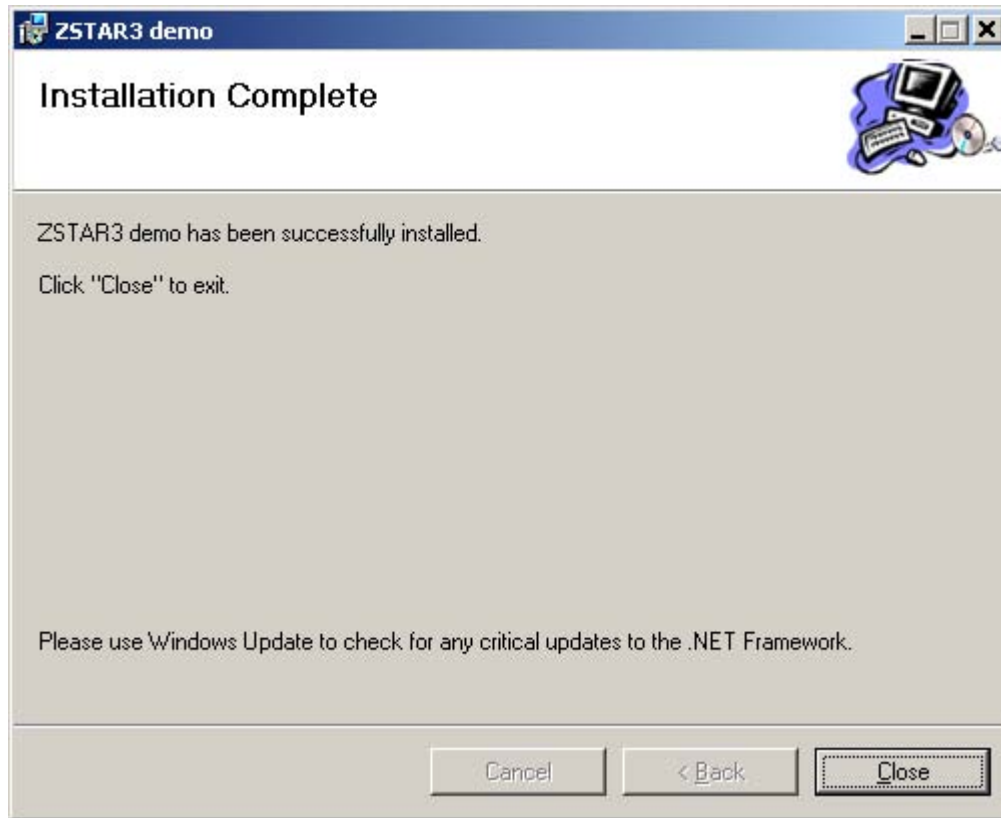


Figure 7-4. Last step of the Installation Process

7.2 ZSTAR3 GUI

The ZSTAR3 GUI is a specially designed PC application for Freescale ZSTAR demos. It fully supports all the new functionalities of the ZSTAR demo and all available ZSTAR family sensor boards.

7.2.1 Features of ZSTAR GUI

- Shows 16 sensors per screen. Displays basic information for each sensor, including the current acceleration values and free-fall events.
- Easy selection of an active sensor board.
- Detailed window of the focused sensor board on the overview screen.
- RF page showing all RF settings and information. RF settings such as the Netnum and the RF channel are modifiable via this screen.
- Lots of various demos of typical accelerometer applications or Freescale sensor presentations.
- User help through about screens in the demo windows.
- Update function for older ZSTAR USB sticks. The GUI can reflash all older USB stick boards with the newest software that is compatible with the older sensor boards.
- USB data flow monitor

7.2.2 The ZSTAR3 GUI Main Controls

This panel is visible on all tabs of the ZSTAR3 GUI application. It shows the most important controls and information of ZSTAR demo.



Figure 7-5. The ZSTAR3 GUI Main Controls

The GUI main controls¹:

1. USB stick connection panel
2. The ZSTAR resolution switch of 8/16 bits
3. Resetting the ZSTAR hardware
4. USB data flow monitor
5. Log file controls
6. Information on the total count of connected sensor boards and the currently selected one.

7.2.2.1 USB Data Flow Monitor

This is a special window designed for monitoring the data rates of communication via the USB interface². The window shows all basic information of the current USB load and it draws a graph of the last 40 seconds.

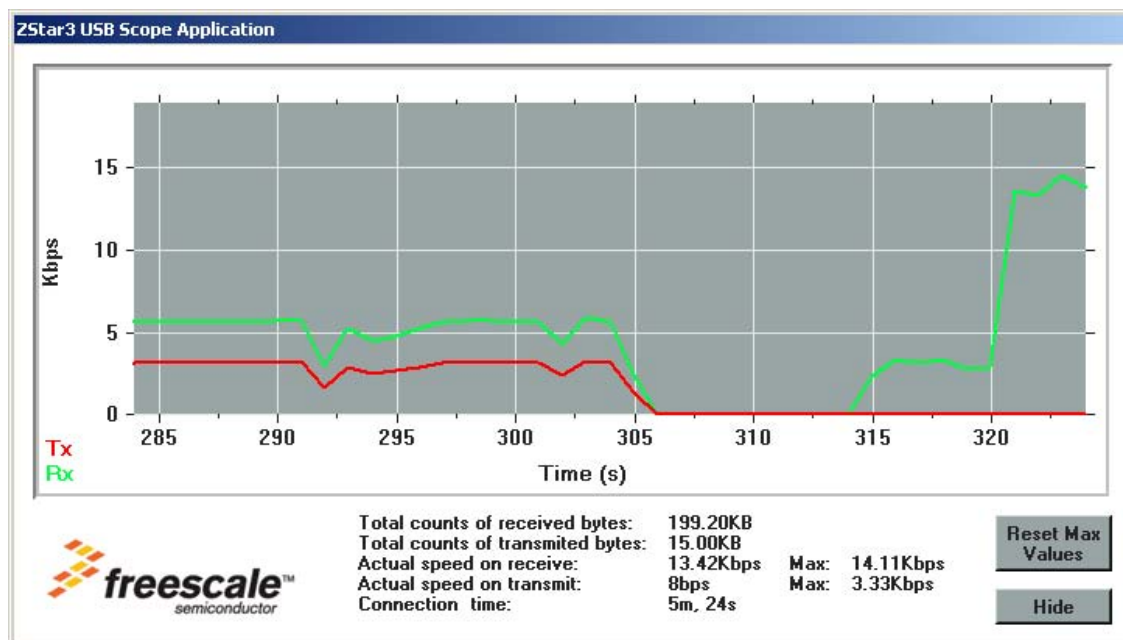


Figure 7-6. USB Data Flow Monitor

1. Red numbers in the picture mark individual controls described in the list below.
2. Only user payload - ZSTAR data.

7.2.3 Sensor Board Overview Screen

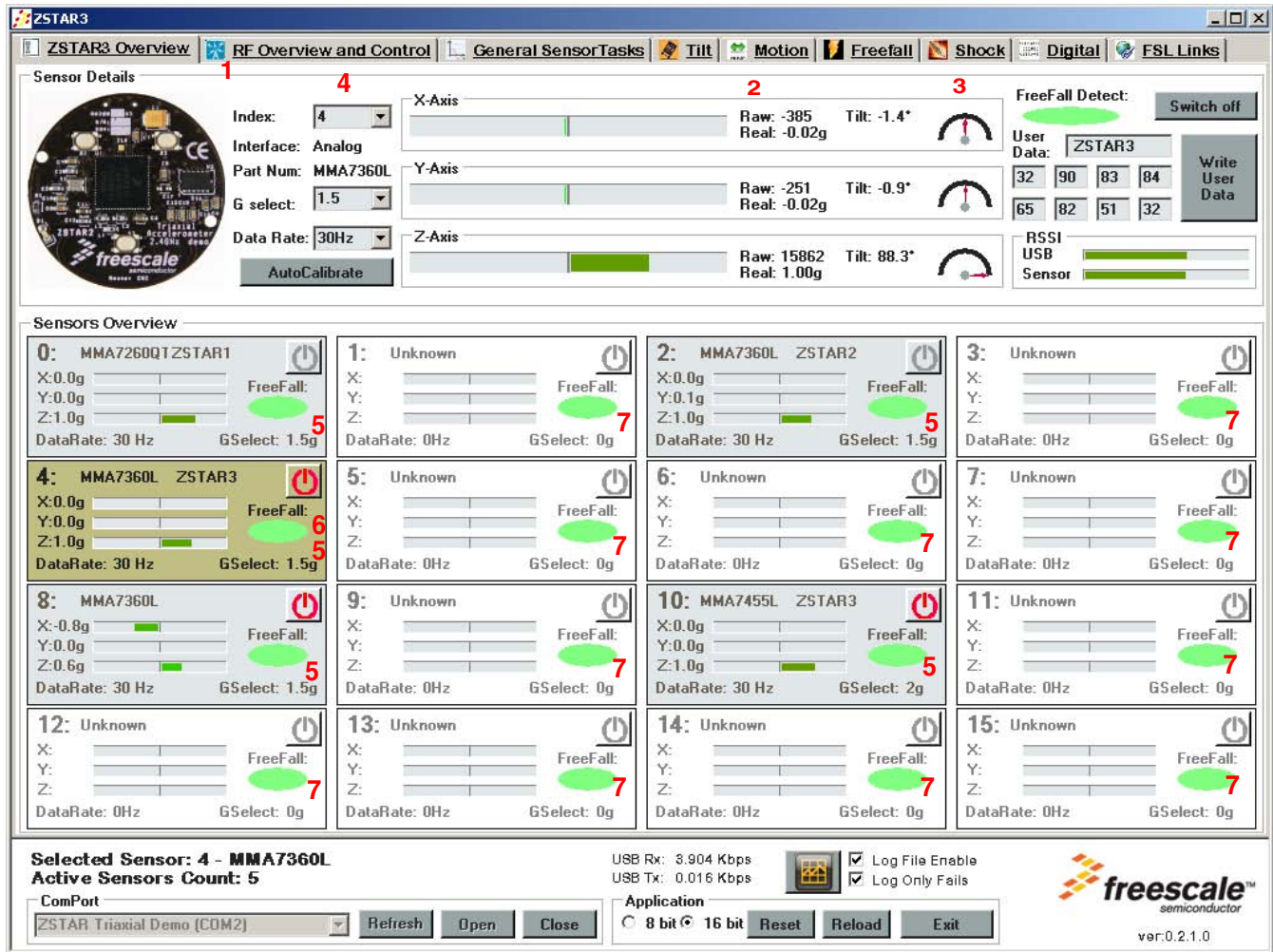


Figure 7-7. Main Sensor Board Overview Screen of the GUI

The GUI main sensor board overview controls¹:

1. Detailed panel of the selected sensor board.
2. Current accelerometric data in graphical and text form.
3. Current tilt of the selected board in graphical and text form.
4. Basic settings of the selected sensor board.
5. Active sensor boards.
6. Currently connected sensor board.
7. Select an active sensor board.
8. Free slots for Sensor boards.

1. Red numbers in the picture mark individual controls described in the list below.

7.2.4 RF Overview Screen



Figure 7-8. RF Overview Screen

The GUI RF overview controls¹:

1. Current working state of the ZSTAR demo.
2. Information on the last measured activity on individual RF channels.
3. Flags for other detected ZSTAR communication on other channels.
4. Currently used RF channel.
5. RF channel change control.
6. Netnum of RF communication change control.
7. RSSI² information of all connected sensors.
8. Scope of RSSI information from selected sensor board.
9. Version of the Demo.

1. Red numbers in the picture mark individual controls described in the list below.
2. Receive signal strength information.

7.2.5 General Sensor Tasks

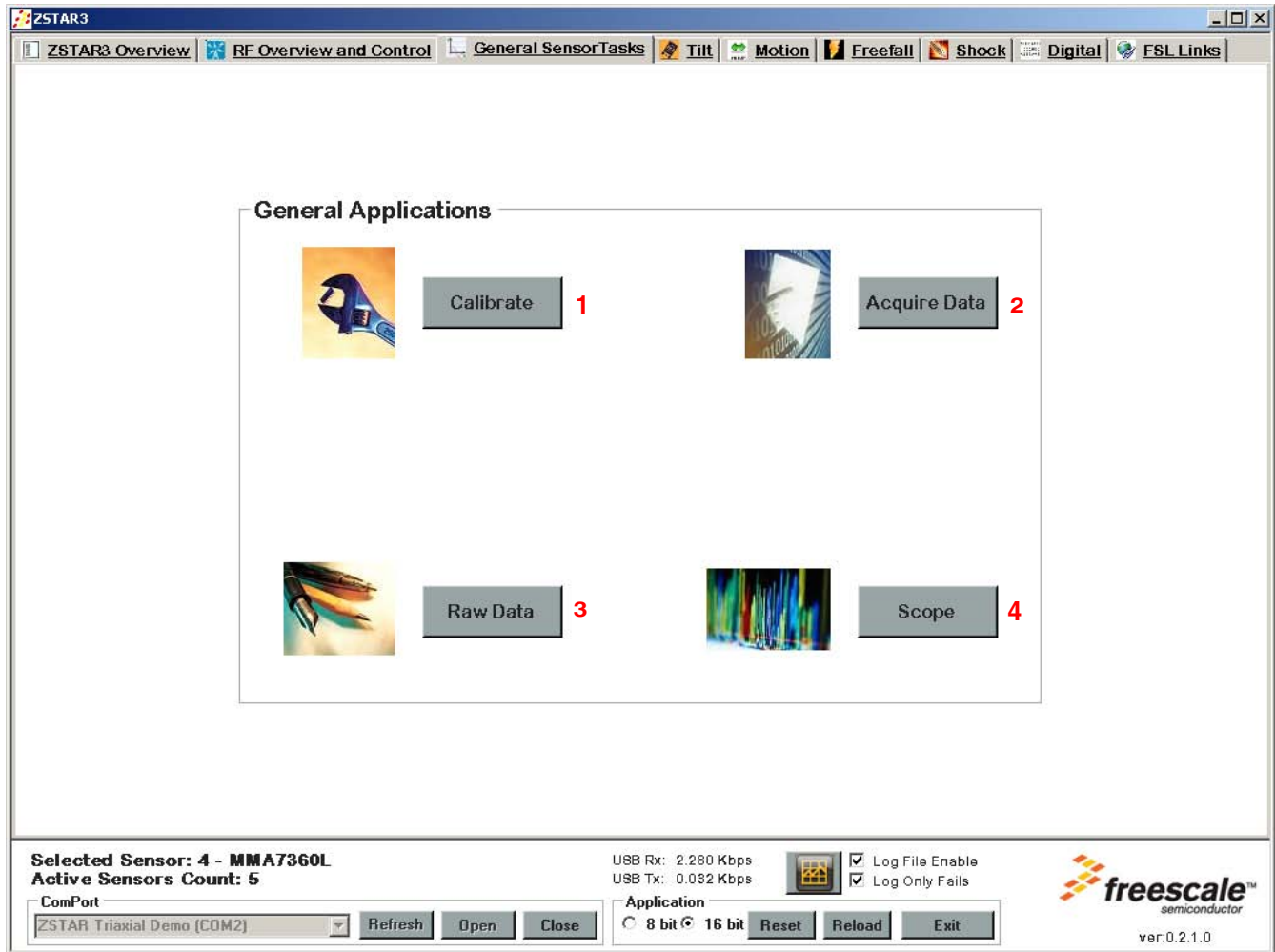


Figure 7-9. The General Sensor Tasks Screen

The GUI General sensor tasks controls¹:

1. Calibration screen - This screen allows a more detailed calibration of the sensor on the selected sensor board.
2. Acquire data screen - This ZSTAR application acquires data on all connected the sensors into an Excel or text file.
3. Raw data screen - This screen shows basic data of the sensor board in general text format.
4. Scope window - This is a useful window that shows acceleration in various formats selectable by the user.

¹ Red numbers in the picture mark individual controls described in the list below.

7.2.5.1 Scope Demo Application

This demo application shows graphical output of measured and computed values from the selected sensor board. It can show unchanged raw data received directly from the sensors or computed values in g. This window also provides computed absolute values from all three axes. All computed values can be displayed as filtered.

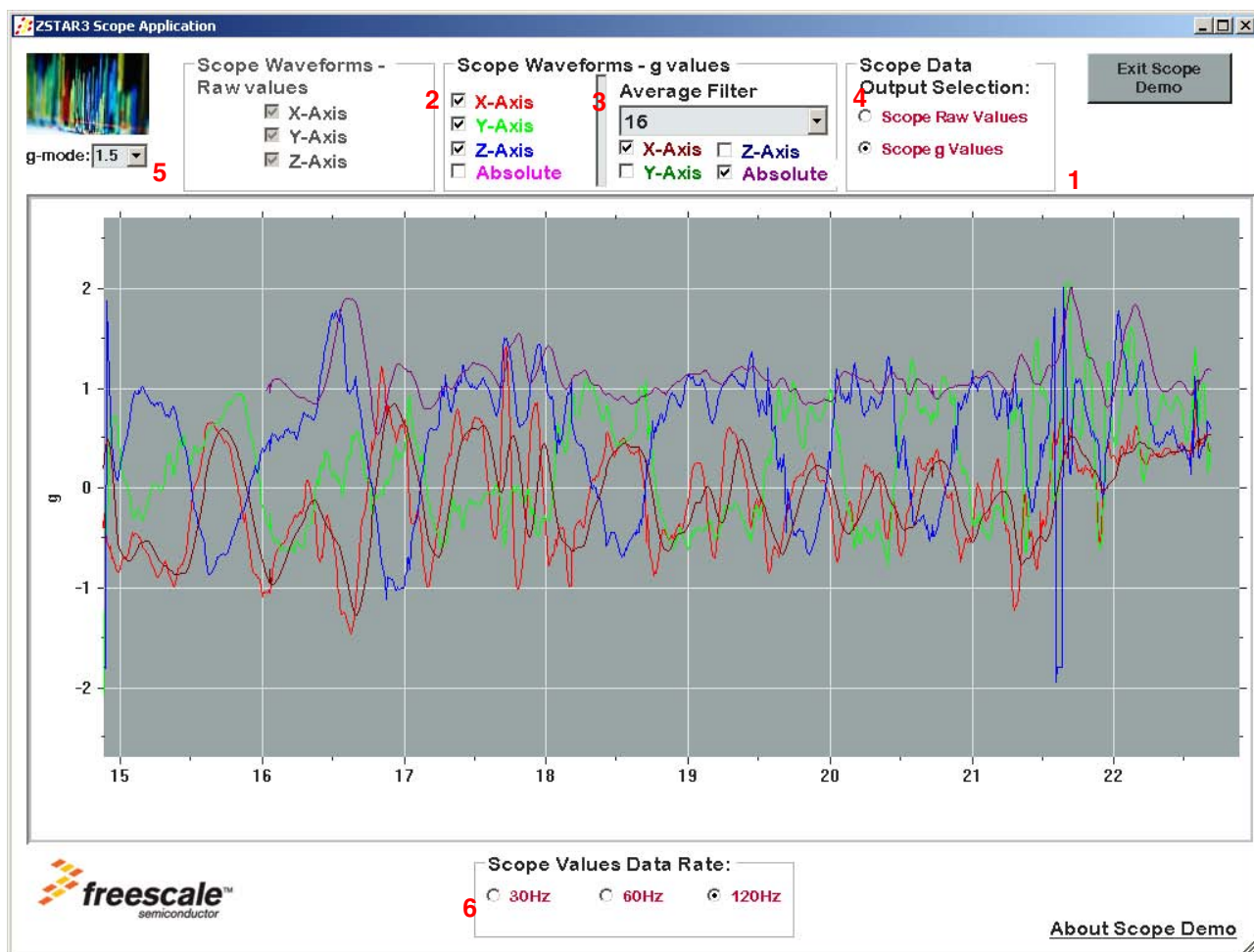


Figure 7-10. Scope Window Application

The Scope window application controls¹:

1. Data source selection - Select which data will be displayed, RAW or REAL.
2. Raw data panel - in this panel you can select which axes will be displayed.
3. Real data panel - in this panel you can select which axes will be displayed.
4. Filtered real data panel - in this panel you can select which filtered axes will be displayed and pick the average filter.
5. The g range select control - this control allow changing g range of sensor in run-time of scope window.
6. Data rate change control - this control allow changing data rate of sensor in run-time of scope window.

1. Red numbers in the picture mark individual controls described in the list below.

7.2.5.2 Acquire Data Demo Application

This application is designed to acquire all measured data from the sensors. It allows selecting the sensors to acquire, selecting the data rate for reading acceleration values, selecting which data will be saved and the output of measured data. There are two choices of output:

- Directly to the Microsoft Excel application (tested on Microsoft Excel 2003 version).
- To a CSV file - the GUI saves all measured data into a text file that can be imported to Excel or another tabular application.

The capture time is not restricted.

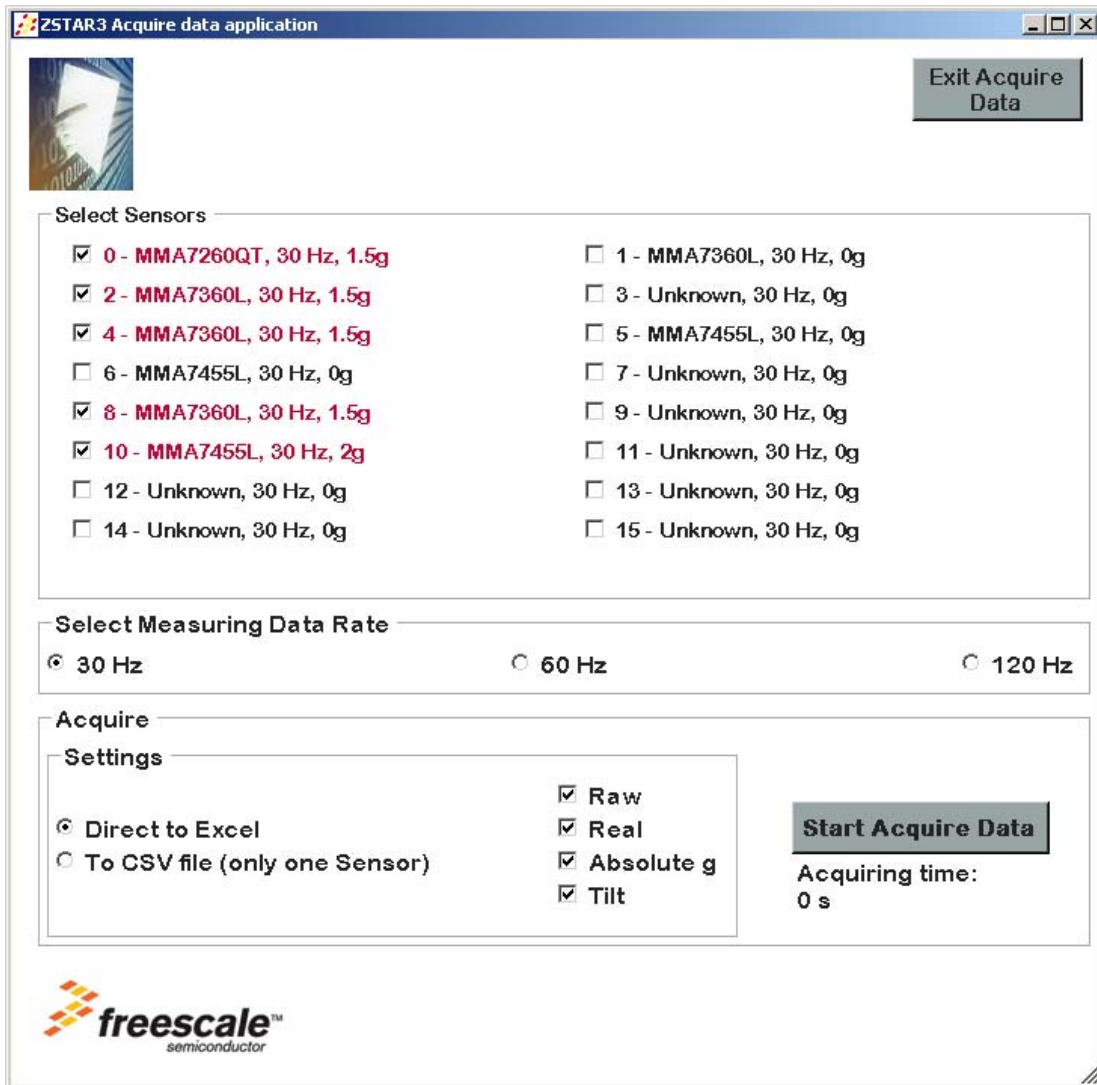


Figure 7-11. Acquire Data Window

7.2.6 Tilt Tasks

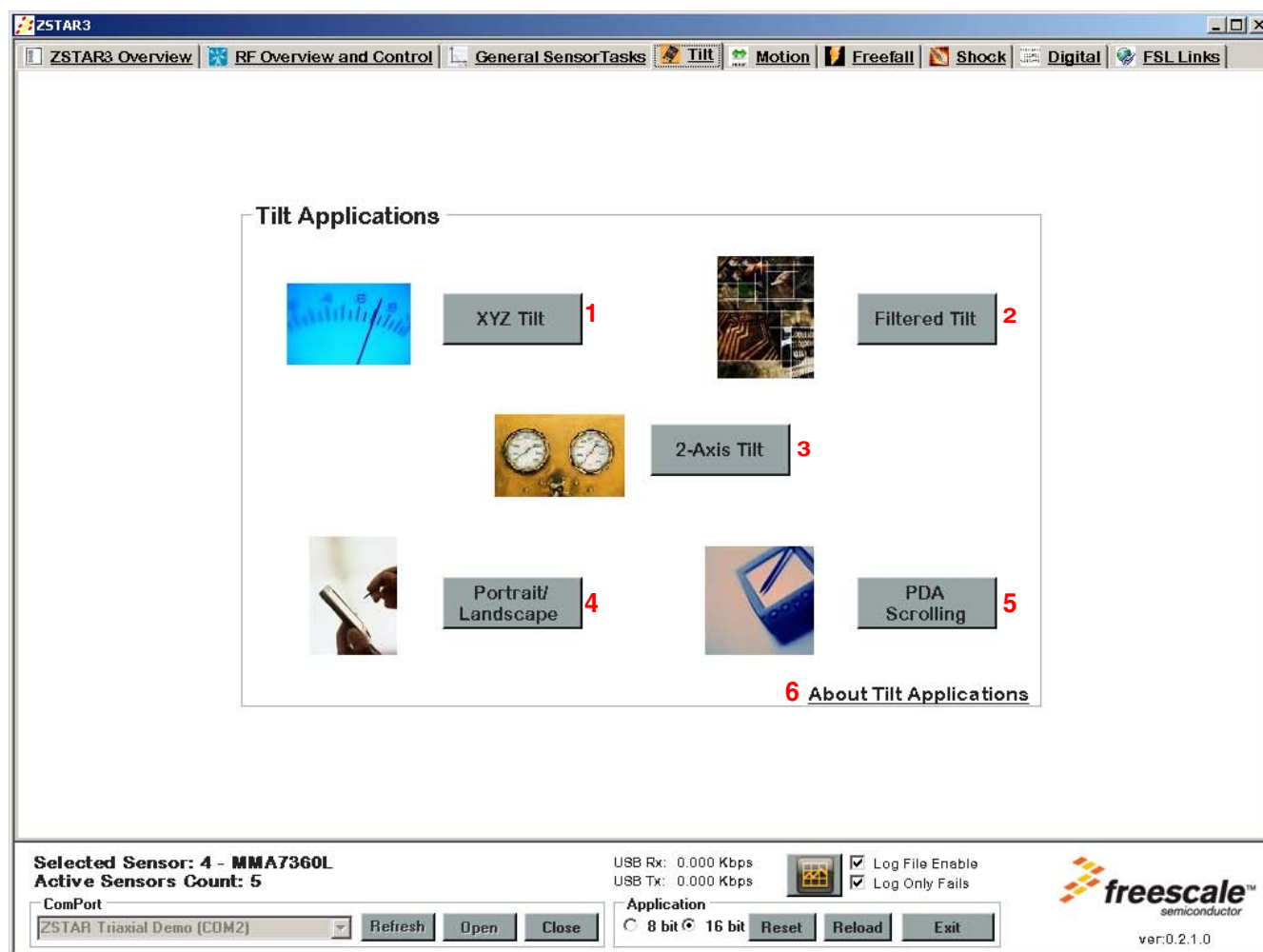


Figure 7-12. The Tilt Tasks Screen.

The GUI Tilt tasks controls¹:

1. Tilt XYZ screen - Window shows the tilt of all axes. Selectable normal / filtered values.
2. Filtered tilt - This ZSTAR application shows the differences between normally measured tilt and filtered.
3. Tilt 2 Axes screen - This is a special demo application that shows how to get from two axes a full range of 0° - 360° tilt.
4. Portrait/Landscape screen - This demo shows one of a couple of tilt detection applications for a PDA.
5. PDA Scrolling screen - This demo shows one of a couple of tilt detection applications for a PDA.
6. About box - This about box contains various information on the Tilt accelerometer applications.

¹ Red numbers in the picture mark individual controls described in the list below.

7.2.6.1 Filtered Tilt Demo Application

This demo window shows differences between measured tilt values and those software filtered. This demo uses a basic average filter algorithm. On the screen is placed four panels:

- Currently measured tilt indicator.
- Filtered value of the currently measured value indicator.
- Setting box - select a measured accelerometer axis and choose an average filter of up to 128 samples.
- Scope showing differences between current and filtered values through time.

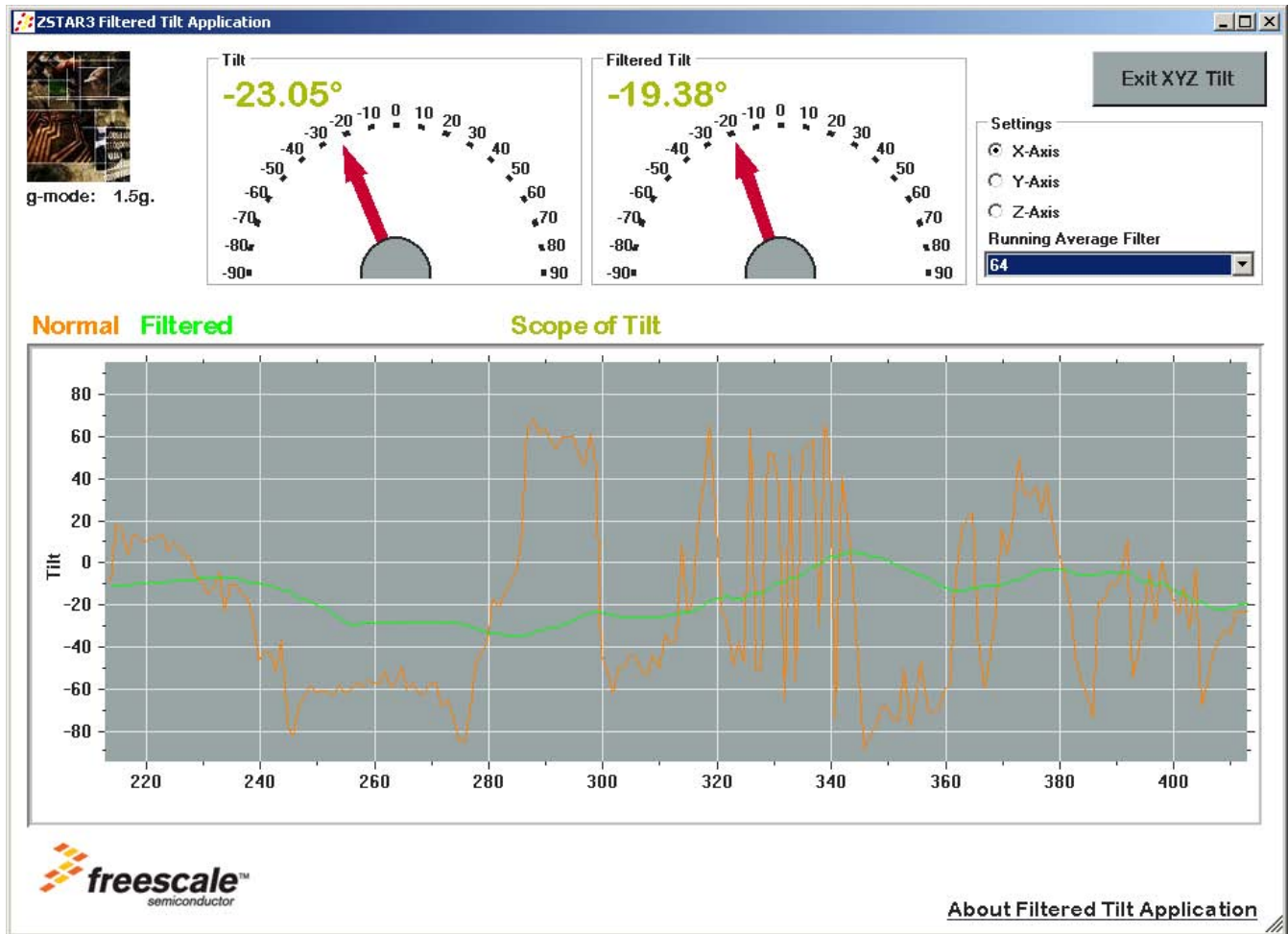


Figure 7-13. Filtered Tilt Demo Application

7.2.7 Motion Tasks



Figure 7-14. The Motion Tasks Screen

The GUI Motion tasks controls¹:

1. Anti-Theft screen - Window shows accelerometer possibilities in Anti - Theft applications.
2. Battery saver screen - This is a demo that shows how accelerometers can be used in some specific applications to save the battery.
3. About box - This about box contains various information on Motion accelerometer applications.

¹ Red numbers in the picture mark individual controls described in the list below.

7.2.8 Freefall Task



Figure 7-15. The Freefall Task Screen

The GUI Freefall tasks controls¹:

1. Freefall screen - Demo show accelerometer possibilities for Freefall detection.
2. About box - This about box contains various information on Freefall accelerometers applications.

¹ Red numbers in the picture mark individual controls described in the list below.

7.2.9 Shock Tasks



Figure 7-16. The Shock Tasks Screen

The GUI Shock tasks controls¹:

1. Shipping and handling screen - This demo shows how accelerometers can be used in Shipping.
2. Threshold Detection screen - This window shows the measurement of g threshold by accelerometers.
3. Shock Detection screen - This window shows the measurement the g peaks by accelerometers.
4. About box - This about box contains various information on Shock accelerometer applications.

¹ Red numbers in the picture mark individual controls described in the list below.

7.2.10 Digital Tasks

This tab contains demo especially designed for the Freescale digital accelerometer sensor (ION) MMA745xL.

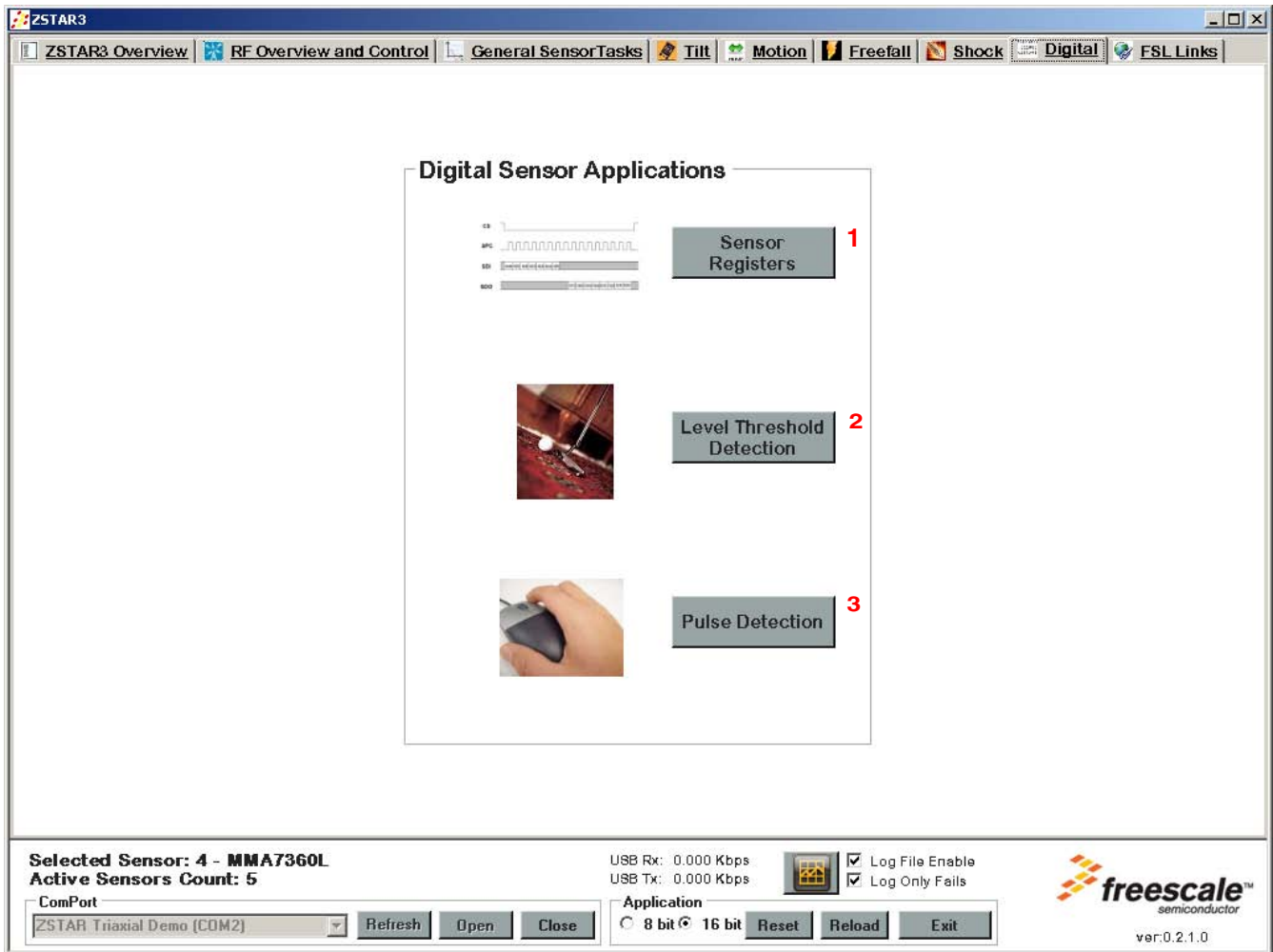


Figure 7-17. The Digital Tasks Screen

The GUI Digital tasks controls¹:

1. Sensor registers screen - This demo shows all registers inside a digital sensor in a user friendly format.
2. Level Threshold Detection screen - This window shows the measurement of g threshold by digital accelerometers.
3. Pulse Detection screen - This window shows the measurement of g pulses by digital accelerometers.

1. Red numbers in the picture mark individual controls described in the list below.

7.2.10.1 Sensor Registers Demo Application

This is a specially designed window that shows all internal registers of the MMA745xL accelerometer. It is split into two main parts:

- The left half contains a list of all registers in the MMA745xL accelerometer and their values in hex format. At the bottom of this panel are placed two check boxes. The first is an Auto Select option, that allows fast mode selection of an individual register, and the second option allows for an automatic refresh value of the selected register.
- Content of the right half is dependent on the left. It shows details of the selected register in a friendly format. Active options are displayed in white color.

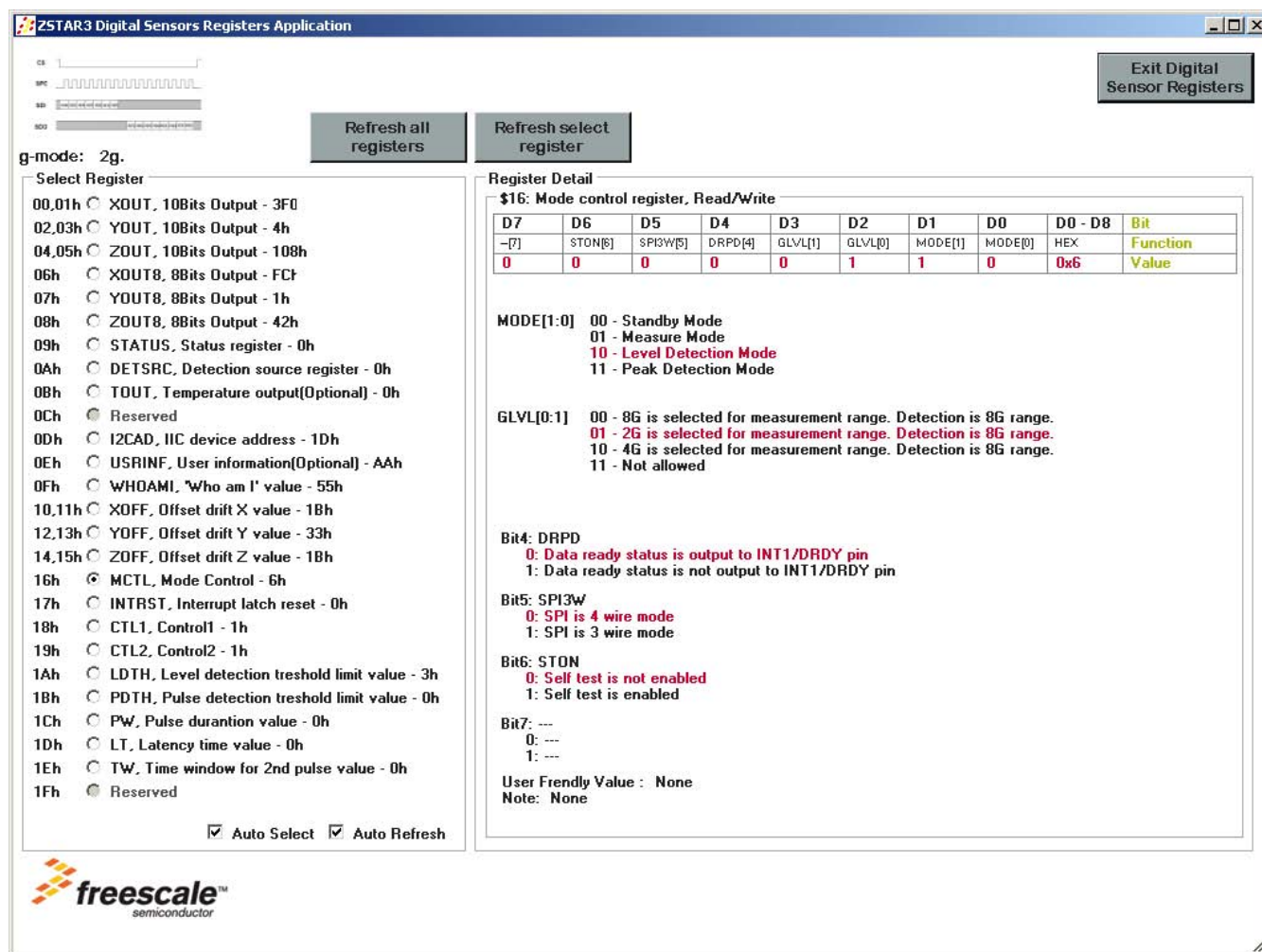


Figure 7-18. Sensors Registers Application Screen

7.2.10.2 Digital Level Threshold Detection Application

This application demonstrates the functionality of the built in Level Detection Interrupts on the MMA45xL digital accelerometer.

By clicking on the “Freefall Detection Settings” button on the right the sensor is configured for the optimal settings for a linear freefall detection event using the Level detection. Then the “Set Up Sensor” button must be pressed to load in the parameters.

The impacted Register Bits in the sensor which can also be seen in the datasheet are displayed in the box to show how the radio buttons and the sliders change these registers.

When clicking the “Motion Detection Settings” button on the right hand side the optimal motion detection settings are programmed. Then by clicking the “Set Up Sensor” button these values are loaded into the sensor. The user can also try their own configuration by clicking the different radio buttons, check boxes and changing the values of the slider. The changed parameters are updated in the “Impacted Register Bits” box area. Remember always to click “Set up Sensor” button to have the new updated values loaded into the sensor to test.

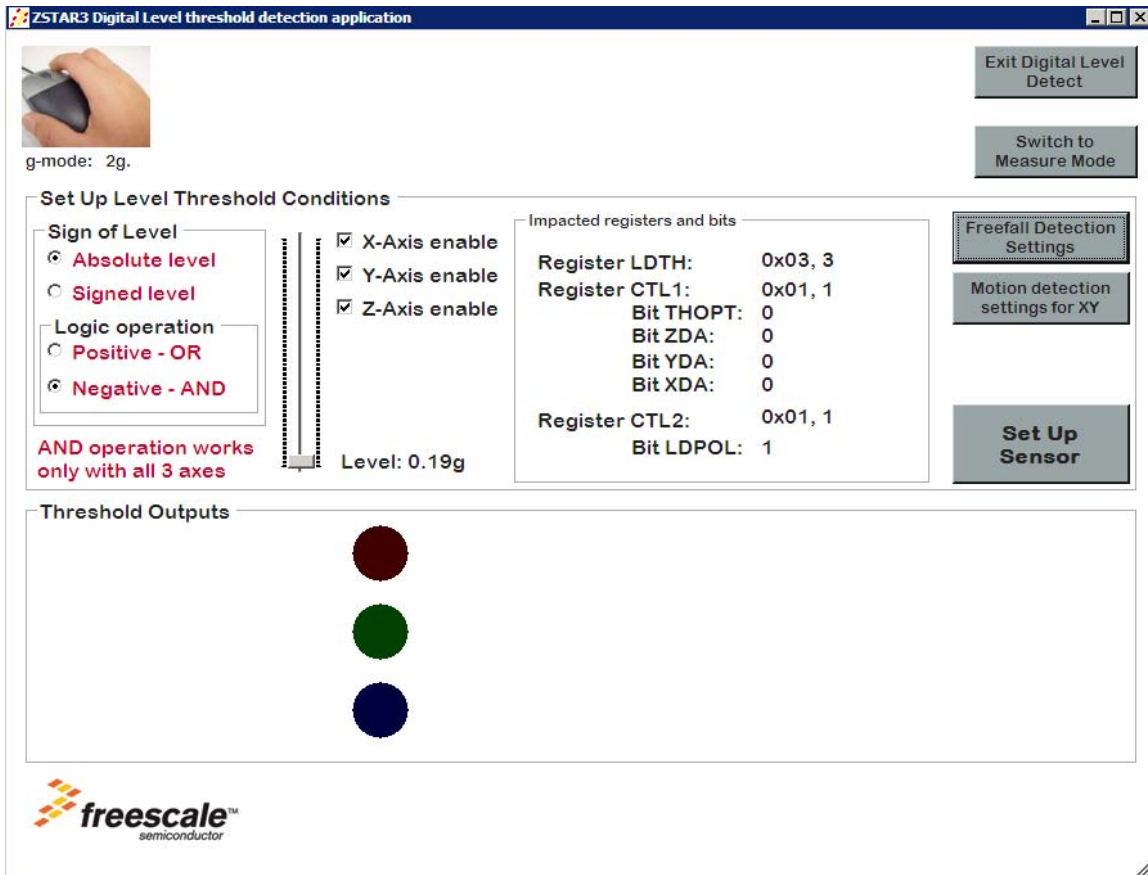


Figure 7-19. Level Threshold Detection Application

7.2.10.3 Digital Pulse Detection Application

In this application the optimal settings for a single pulse or for a double pulse have been preprogrammed and can be selected by clicking either the “Pulse Detection Settings” button for a single pulse or the “Double Pulse Detection Settings for Z-axis” button for a double pulse on the Z axis. Then by clicking the “Set-Up Sensor” button the values are loaded into the sensor. The user can also manually set the parameters and then load them in by clicking the Set-up Sensor. The Logic Operation Negative-AND is used for Freefall detection. For this function X, Y, and Z axes must be enabled, a pulse duration must be set and a latency time value must be set. Then the “Set Up Sensor” button will load these values to set up the sensor for a freefall event.

Figure 7-20. Pulse Detection Application

7.2.11 Freescale Web Links.

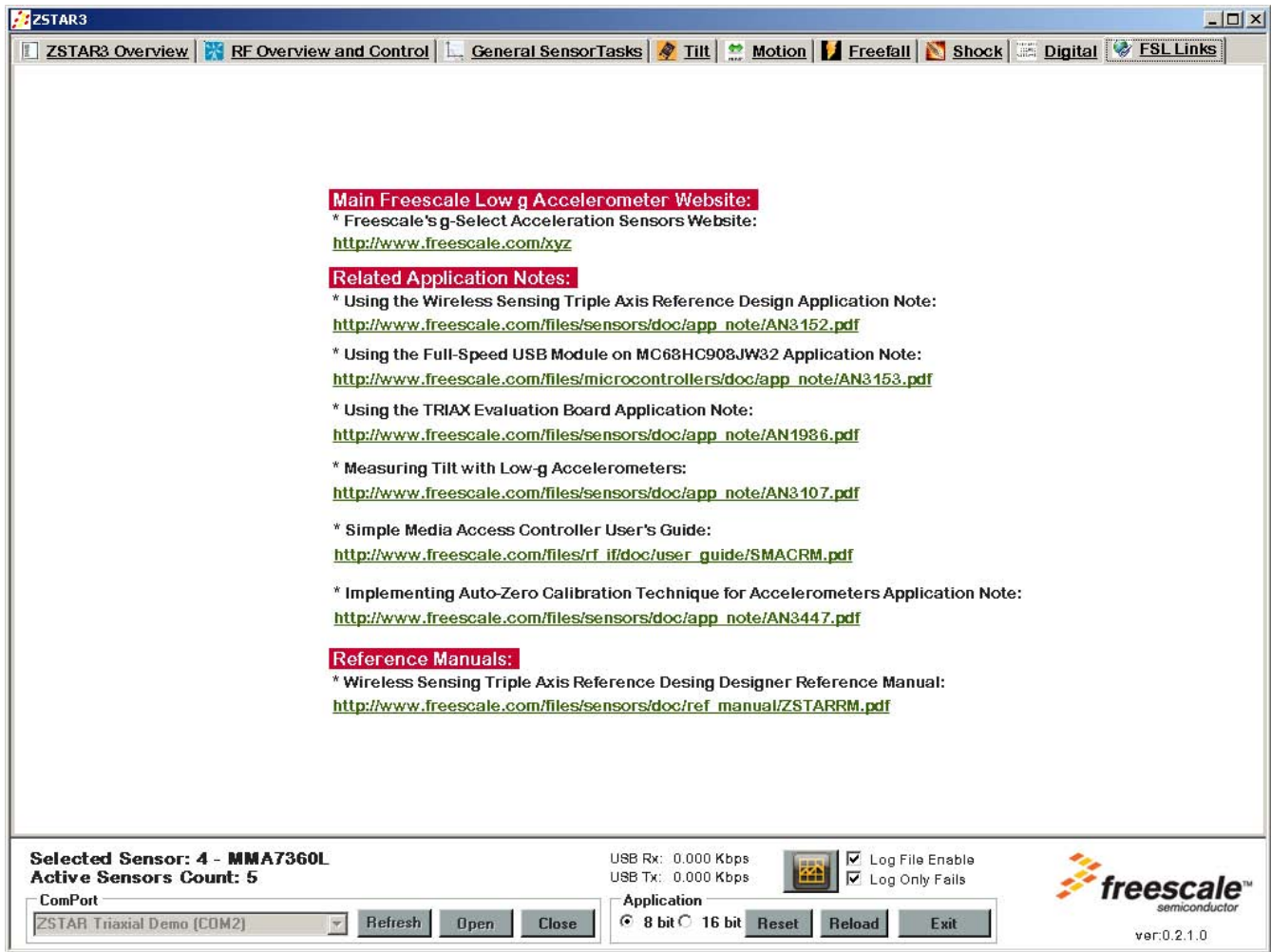


Figure 7-21. The Links Screen

This screen contains a couple of interesting links on Freescale and its accelerometric sensors.

7.3 The ZSTAR3 GUI Update USB Stick Software Utility

The ZSTAR3 GUI can easily upgrade firmware in the USB stick by Bootloader capability. There are two different ways to start upgrade utility:

- Self-start-up - the GUI always checks the software version of the USB stick with an open port action. If it finds an older version than it has, the GUI displays an information message which asks user to start the software update utility.
- Manual start - the GUI can directly open a registered Bootloader Com port.

7.3.1 Update Process (Manual Start)

1. For a manual start, select the Bootloader Com port¹ from the Comport list and press the Open button.



Figure 7-22. Update Software Procedure - Step 1

2. When the GUI shows an Information message box about the Update procedure, click the OK button to continue.

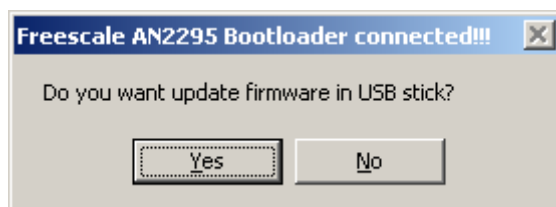


Figure 7-23. Update Software Procedure - Step 2

1. Bootloader port is started by the procedure described in [Section 5.6.1](#), "Switch to Bootloader Procedure."

- 3. Now the GUI shows the Update window. Here you must select the correct type of the currently connected ZSTAR USB stick.

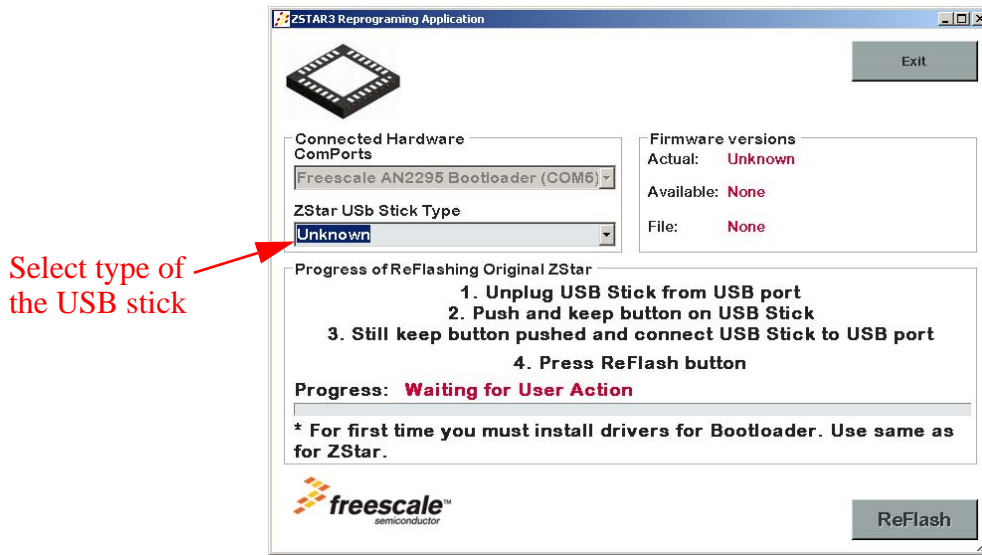


Figure 7-24. Update Software Procedure - Step 3

- 4. When you select the correct type of USB stick, the GUI knows which files it has and offers the latest version of available software.

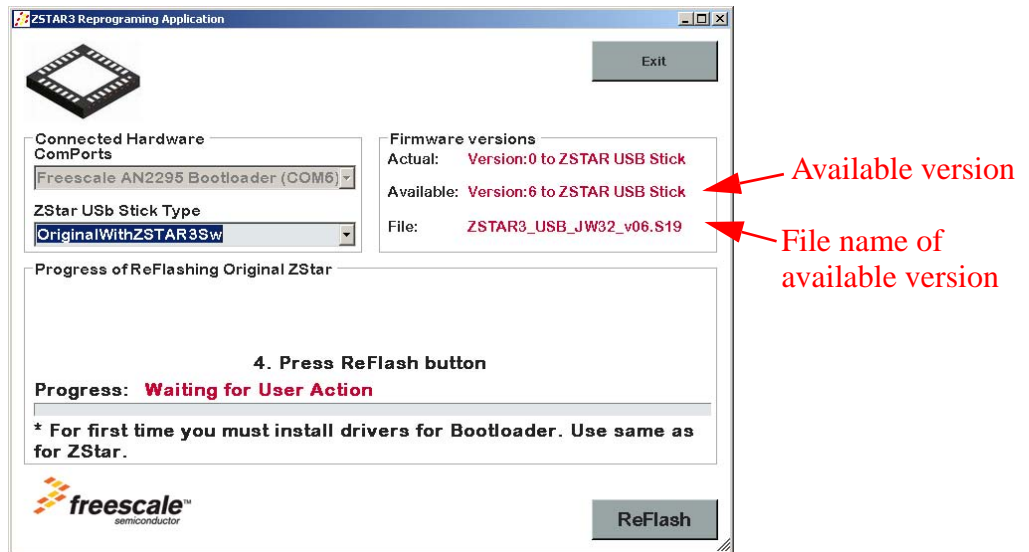


Figure 7-25. Update Software Procedure - Step 4

- Now you can press the ReFlash button. The GUI starts looking for an available Bootloader port.

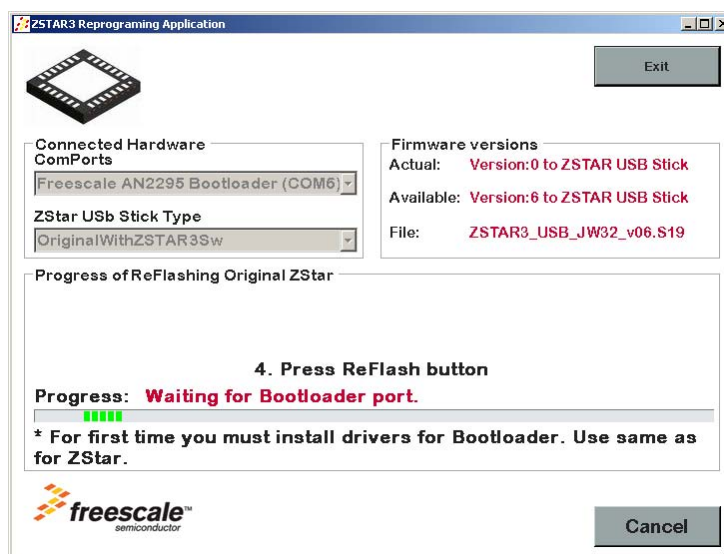


Figure 7-26. Update Software Procedure - Step 5

- When the GUI finds a Bootloader port, the original bootload procedure starts. Please wait till the end of the bootloader process.

```

C:\D:\ccwork\B01119_view_dev\ICONN\IC116 - ZSTAR2 - Low-cost 2.4GHz and XYZ accelerometer demo...
FC protocol versions supported:
  0x01 <HC08>
  0x03 <large HC08>
  0x02 <S08>
  0x06 <non-bin S08>
  0x0A <large S08>

See Freescale Application Note AN2295 and AN2295SW for updates.

-----
Parsed $-record lines: 694 Bytes total: 22012
Source address range: 0x7000-0xFFFF

Waiting for HC(S)08 reset ACK...received 0xfc <good>.

Bootloader protocol version: 0x03 <large HC08, read command supported>
Bootloader version string: JW32/USB $Version: 1.0.15.0$
Number of memory blocks: 1
Memory block #1: 0x7000-0xE5BF
Erase block size: 512 bytes
Write block size: 64 bytes
Original vector table: 0xFFDE
Bootloader user table: 0xE5C0

Memory programming: W 0xA940 66%

```

Figure 7-27. Original Bootloader Application

When the Update process finishes successfully, close the update window and start to enjoy all the new features that brings new version of the ZSTAR firmware.

Appendix A References

The following documents can be found on the Freescale web site: <http://www.freescale.com>.

1. AN2295, Application Note - Developer's Serial Bootloader for M68HC08 and HCS08 MCUs
2. AN3153, Application Note - Using the Full-Speed USB Module on the MCHC908JW32
3. AN3447, Application Note - Implementing Auto-Zero Calibration Technique for Accelerometers
4. ZSTARRM, Reference Design Manual - Original ZSTAR Demo (RD3152MMA7260Q) RDM
5. MC9S08QG8, Data Sheet
6. MCHC908JW32, Data Sheet
7. MMA7260QT, Data Sheet
8. MMA7361LT, Data Sheet
9. MMA745xL, Data Sheet
10. MC13191, Data Sheet
11. MC13213, Data Sheet

References