Wireless M-Bus Suite and Silabs specific documetation

Quick Start Guide

Steinbeis Transfer Center
Embedded Design and Networking

January 17, 2014

Document History

1.00	2013-10-11	First release	mr
1.01	2013-11-11	Setup chapter updated	mr
1.02	2013-11-14	Note in chapter 'Firmware' added	mr
1.03	2014-01-17	Added chapter 4.1.4	ar

Table 1: Document History

1   Introduction

This document is a hardware and software guide for Silabs devices and the Wireless M-Bus Stack from Steinbeis Transfer Center Embedded Design and Networking [STZEDN] and describes the installation of Wireless M-Bus devices to test, to evaluate and to configure with the Wireless M-Bus Suite.

The guide contains the following chapters:

- Chapter 5.1: System Requirements

  This chapter specifies the system requirements for the PC to use the Wireless M-Bus Suite for 32-Bit and 64-Bit operating systems.

- Chapter 5.2: COM Port Settings

  This chapter contains the required settings to connect the target hardware to the PC.

- Chapter 4: Hardware Setup and Firmware Installation

  This chapter describes the hardware setup, defines performance and peripheral information and specifies different ways to upload the Wireless M-Bus Stack from STZEDN for each Silabs microcontroller family.

- Chapter 5: Wireless M-Bus Suite Quick Start Guide

  The Wireless M-Bus Suite quick start guide leads the user step by step through the included demo configuration to setup a Wireless M-Bus network in a fast and easy way.

  The Wireless M-Bus Suite from STZEDN is a computer program to test, to evaluate and to configure Wireless M-Bus devices running the Wireless M-Bus Stack. It contains mechanisms to build up a Wireless M-Bus network. It provides interfaces to control the different device types of the Wireless M-Bus Stack - collector, meter and protocol monitor (Wireless M-Bus sniffer). The Wireless M-Bus Suite is designed to handle different network configurations. Configurations can easily be changed, updated or stored. Furthermore it is capable to run Wireless M-Bus devices with different parameters and settings (e.g. security, device addresses, uni- or bidirectional).

  The Wireless M-Bus Suite uses a serial interface to communicate with the Wireless M-Bus devices and is able to upload a new firmware image to a Wireless M-Bus device.

- Chapter 6: Wireless M-Bus Protocol Monitor

  The Wireless M-Bus Protocol Monitor is an integrated application in the Wireless M-Bus Suite. It is a Wireless M-Bus Sniffer and can be used to monitor Wireless M-Bus packets. Besides meter and collector configurations, the Wireless M-Bus Stack from STZEDN contains configurations is capable of Wireless M-Bus Sniffer configurations.

- Chapter 7: Demonstration Application

  This chapter presents the demonstration application of the Wireless M-Bus Stack with a sample experimenter

board. Furthermore, it defines the nomenclature, specifies the different delivery packages of the Wireless M-Bus Stack and shows how to use the Wireless M-Bus Stack with and without the Wireless M-Bus Suite.

Table 2 lists the supported Transceivers from Silabs, as well as the supported Starter Kits.

	SI4461	SI4463
EFM32LG890	X	X

Table 2: Supported Transceivers and Starter Kits

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus
wireless

Contents

Quick Start Guide
Version 1.02   Steinbeis Transfer Center
Embedded Design and Networking   M-Bus wireless

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus
wireless

Steinbeis Transfer Center
Embedded Design and Networking

List of Tables

Quick Start Guide
Version 1.02

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus
wireless

List of Figures

Quick Start Guide
Version 1.02

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus
wireless

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus

2   Wireless M-Bus Basics

2.1   Introduction

The Metering Bus (or in short "M-Bus ") is a field bus specialized for transmission of metering data from gas, electricity, heat, water or other meters to a data collector. It is described by European Norm (EN 13757), which includes the specification of wired and Wireless M-Bus. The specification is divided into five parts:

- EN 13757-1 ([1]):

  Communication systems for meters and remote reading of meters - Part 1: Data exchange

  The first part describes the basic communication between the meters and a central data collector. It provides an overview of the communication system.

- EN 13757-2 ([2]):

  Communication systems for meters and remote reading of meters - Part 2: Physical and link layer

  The second part includes the specification of the physical data transmission using wired connections. It also includes the description of the protocol to transmit the data.

- EN 13757-3 ([3]):

  Communication systems for meters and remote reading of meters - Part 3: Dedicated application layer

  The third part describes a standardized application protocol to enable multivendor capability. So devices of different manufacturers may be combined in one system.

- EN 13757-4 ([4]):

  Communication systems for meters and remote reading of meters - Part 4: Wireless meter readout (Radio meter reading for operation in the 868 MHz to 870 MHz SRD band)

  This part specifies the wireless communication of M-Bus and is the main source document for this implementation. It includes the Physical and the Data Link Layer for wireless devices. It corresponds to specification EN 13757-2 for wired communication.

- EN 13757-5 ([5]):

  Communication systems for meters and remote reading of meters - Part 5: Relaying

  This last part includes different proposals for relaying data frames to overcome the range problem between remote meters and data collectors.

The sixth part of this norm relates only to wired busses and is disregarded in this document.

All parts of EN 13757 are compliant to the European Norm EN 870-5 [6].

2.2   Protocol Stack

2.2.1   Introduction

M-Bus is compatible to the international ISO/OSI-model, but only the layers 1, 2 and 7 are implemented.

Layer 7	Application Layer (EN 13757-3)
Layer 2	Data Link Layer (EN 13757-2 or EN 13757-4)
Layer 1	Physical Layer (EN 13757-2 or EN 13757-4)

Table 3: Operating Modes of Wireless M-Bus

Up to now, the application layer implements all other protocol layers required for a specific appliance. Especially if routing is required according to [5], it is implemented in the application layer. The reduced modularity leads to compact implementations running on very small devices with minimum computing resources. But the lack of modularity certainly is one of the reasons why standardized routing protocols are currently not available for Wireless M-Bus. The M-Bus favours asymmetric network topologies with low-cost or low-power metering devices on the one side and data collectors or gateways with higher performance on the other side. Currently, only point-to-point or star network topologies can be supported. Mesh or multi-hop topologies are not possible. This chapter describes the settings for implementations of the data collector and the metering devices to setup a network in star topology.

2.2.2   Communication Modes

Dependly on the application there are various combinations of communication modes for data collectors and metering devices. These settings define the communication flow and the configuration of the radio channel. Table 4 lists available communication modes.

Mode	Communication	Description
S1	Unidirectional	In the stationary mode, the metering devices send their data several times a day. In this mode, the data collector may save power as the metering devices send a wakeup signal before transmitting their data.
S2	Bidirectional	Bidirectional version of S1.
T1	Unidirectional	In the Frequent Transmit mode, the metering devices periodically send their data to collectors in range. The interval is configurable in terms of several seconds or minutes.
T2	Bidirectional	Bidirectional version of T1. The data collector may request dedicated data from the metering devices.
C1	Unidirectional	"Compact Mode" mode. This mode is similar to mode T but it allows for transmission of more data within the same energy budget and with the same duty cycle. It is suitable for walk-by and/or drive-by readout. The common reception of mode T and mode C frames with a single receiver is possible.
C2	Bidirectional	Bidirectional version of C1. The data collector may request dedicated data from the metering devices.

Table 4: Operating Modes of Wireless M-Bus

Table 5 lists the combinations of the communication modes of meters and data collectors.

In T2 mode, the communication settings for the two data directions are different. Due to different requirements, both communication devices have to support fast switching of communication speed, coding scheme, and frequency.

Data Collector channels	Meter Device's channels			Collector Device's Tx radio settings (Freq., data rate, modulation)
	S1, S2	T1, T2	C1, C2	
S1, S2	X			868,30 MHz 32.768 kcps Manchester enc.
T1, T2		X		868,30 MHz 32.768 kcps Manchester enc.
C1, C2		X		868,30 MHz 32.768 kcps Manchester enc.
C1, C2			X	869,525 MHz 50 kcps NRZ enc.
Metering Device's Tx radio settings (Freq., data rate, modulation)	868,30 MHz 32.768 kcps Manchester encoding	868,95 MHz 100 kcps 3-out-of-6 encoding	868,95 MHz 100 kcps NRZ encoding	

Table 5: Compatibility matrix for Wireless M-Bus operating modes

Table 5 shows that although the T- and S-modes cannot be combined, a T2 data collector may not well coexist with a parallel S-mode communication int its vicinity. Table 5 also shows that the C-mode supports the common reception of mode T and mode C frames with a single receiver.

2.2.3   Unidirectional vs. Bidirectional Communication

If unidirectional communication is used, data will be sent from the metering device to the data collector only. This enables simple transmitters as metering devices while the data collector only needs to receive. Because listen-before-talk (LBT) and dynamic network configuration are not possible, it is recommended to use the unidirectional modes for small and simple constellations with little network load.

In case of bidirectional communication, the collector device can request data from the meter device. This is for example the case in S2, T2, C2 and N2 mode. In this modes, a bidirectional communication will only be established if further data or commands need to be exchanged (cf. Fig. 1). Figure 1 shows a typical communication flow.
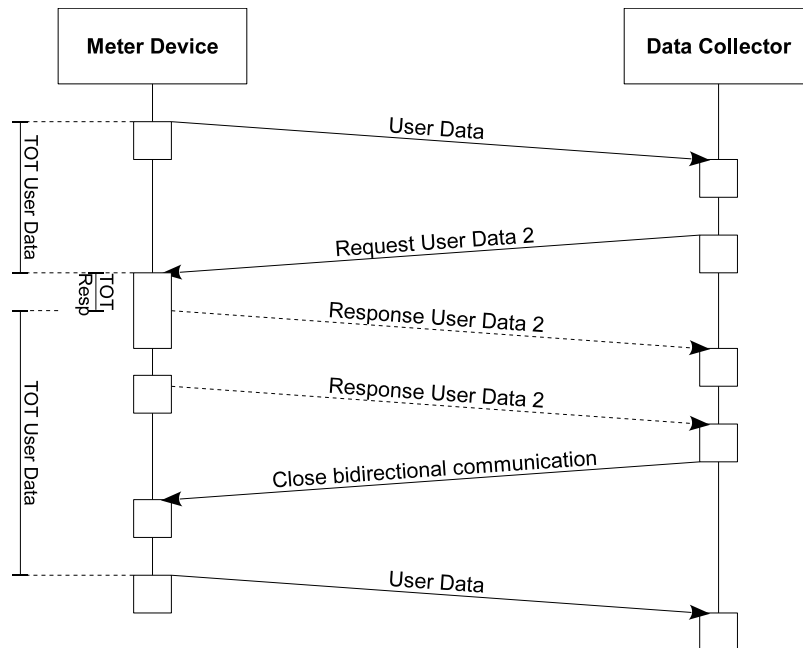
Figure 1: Bidirectional Communication with a request after the reception of a periodical meter packet. Example for mode S2. (Supported by the following modes: T2, S2, C2 and N2)

Steinbeis Transfer Center
Embedded Design and Networking

3   The STZEDN Protocol Stack

3.1   Wireless M-Bus Protocol Stack

The Wireless M-Bus Stack from STZEDN is a fully Wireless M-Bus compliant protocol stack, including OMS v.3.0.1 [7]. It implements the functionality described in EN-13757 ([1], [2], [3], [4], [5]). Refer to figure 2 for a graphical overview.
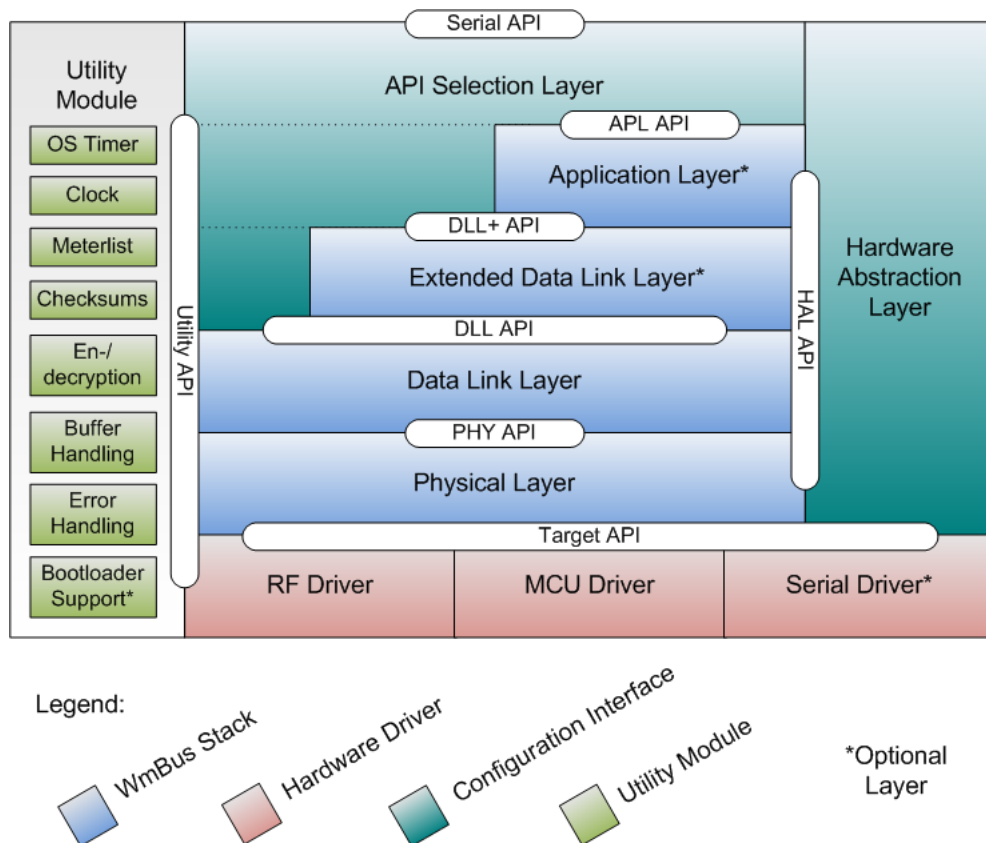
Figure 2: Wireless M-Bus Protocol Stack

The Wireless M-Bus protocol stack from STZEDN basically consists of four different parts:

- Wireless M-Bus Stack:

  The Wireless M-Bus Stack part of the protocol stack contains all Wireless M-Bus functionality.

- Hardware drivers:

  This part of the protocol stack contains all low level drivers to control the microcontroller unit, such as communication interfaces, memory resources and pheripheral units.

- Utility Module:

  This part of the protocol stack contains mechanisms and functions which the stack uses to forward information through the protocol layers, to report runtime information, to handle time intervals and clocks, to manage meterlists and to execute software updates.

- Configuration interfaces:

  The configuration interfaces are used to setup basic parameters of the target, such as location of the communication settings or input interrupt sources.

STZEDN offers different packages of the Wireless M-Bus protocol stack. All packages support full Wireless M-Bus functionality. Refer to table 6 for an overview. To find the most suitable Wireless M-Bus version for the desired application, please contact STZEDN (refer to chapter 8).

Layer	Basic	Extended	Extended serial	Full
Application Layer		x	x	x
Extended Data Link Layer	x	x	x	x
Data Link Layer	x	x	x	x
Physical Layer	x	x	x	x
RF Driver	x	x	x	x
MCU Driver	x	x	x	x
Serial Driver			x	x
Bootloader support				x

Table 6: Available Wireless M-Bus Protocol Stack Packages

Table 7 lists the different memory models of the stack. The memory model specifies where the Wireless M-Bus Stack saves internal data and meterlists.

Package	Volatile memory	Non-volatile memory
Basic	x	x
Extended	x	x
Extended serial	x	x
Full	x	x

Table 7: Available Wireless M-Bus memory models

Please contact STZEDN for information about currently supported microcontrollers and transceivers. Adaptions of the Wireless M-Bus protocol stack are possible upon request as well as driver developments for microcontrollers and transceivers. Please refer to chapter 8 for contact information.

3.2   Wireless M-Bus Protocol Stack Deliveries

3.2.1   Demonstration Applications

Demonstration applications for development boards were delivered as binary files. Please note, that the demonstration application contain the full Wireless M-Bus Stack from STZEDN. The Wireless M-Bus Suite can be used to study, test and evaluate the wmbus demonstration application and Wireless M-Bus, e.g. bidirectional communication between a Wireless M-Bus collector and a Wireless M-Bus meter device. Furthermore, Wireless M-Bus Suite can download the binary files to development boards to switch between Wireless M-Bus device types and Wireless M-Bus modes. Please contact STZEDN for information about supported development boards. The Wireless M-Bus Suite also

supports Wireless M-Bus sniffer applications for all Wireless M-Bus channels, including decryption and encryption of data.

3.2.2   Wireless M-Bus Stack as Library

STZEDN delivers the Wireless M-Bus Stack for the Wireless M-Bus devices types and Wireless M-Bus modes in libraries. Refer to table 8 for an overview. Please contact STZEDN for information about supported integrated development environments. Refer to chapter 8 for contact information. The customer is able to develop application using the Wireless M-Bus Stack libraries. For a fast and easy development, the Wireless M-Bus Stack comes with application code example for a collector and a meter device.

Mode	Meter Device	Collector Device
S1	x	
S2	x	x
T1	x	
T2	x	x
C1	x	
C2	x	x
N1	x	
N2	x	x

Table 8: Available Types and Modes

Please note, that Wireless M-Bus sniffer configrations are only delivered as binaries.

3.3   Configuration of the Wireless M-Bus Stack

STZEDN delivers the Wireless M-Bus protocol stack in fully configured libraries for different device types and modes (refer to table 8). Depending on the target hardware, the Wireless M-Bus Stack supports several communication interfaces, to enable customized hardware designs.

Target configuration file: *stzedn_target_config.h*

The target configuration file contains definitions about pin and port definitions of the communication interfaces. This includes e.g. the SPI and UART communication, as well as GPIO input pin definitions. Please note, that the number of configuration parameters for the hardware drivers depends on the target hardware. Refer to the hardware specific configuration guide.

3.4   Memory Models

3.4.1   Internal data

Table 9 lists the internal data of the Wireless M-Bus Stack, which is effected by the choice of the memory model.

Description	Meter	Collector
Device status	x	x
RF channel	x	x
Frequency offset	x	x
Serial baud rate	x	x
Meter Wireless M-Bus address	x	x
Collector Wireless M-Bus address	x	
Periodical transmission interval	x	
Device specific encryption key	x	x
General encryption key	x	x
Wireless M-Bus standard specific data	x	x
Meterlist		x

Table 9: Internal data of a Wireless M-Bus meter device.

3.4.2   Volatile memory model

The Wireless M-Bus Stack saves all internal vairables as well as the meterlist in volatile memory. More precisely, in the random access memory (RAM). Thus, all data which changes at runtime is lost after a reset of the device. Refer to table 9 for a list of parameters which are effected.

3.4.3   Non-volatile memory model

3.4.3.1   Data storage

The non-volatile memory model stores the data listed in table 9 in the non-volatile memory. The Wireless M-Bus Stack does not specify which kind of non-volatile memory must be used. E.g. FLASH, EEPROM or FRAM.

In general, the Wireless M-Bus Stack buffers most parameters in volatile memory. It synchronizes the volatile memory with the non-volatile memory at specific points in time. A complete synchronization of the data in table 9 takes place at the following actions/events:

- During initialization of the Wireless M-Bus Stack.

- A forced synchronization, initiated by user application.

- When the Wireless M-Bus Stack goes to sleep mode, initiated by user application.

However, some data cannot be buffered in volatile memory. Thus, some API function calls cause a write access to non-volatile memory immediately - API functions which cause changes of meterlist entries. Thus, this effects only Wireless M-Bus collector devices. Furthermore, these functions do not initiate a complete synchronization, but apply only the changes to the non-volatile memory which is described by the API function. Refer to table 10 for a corresponding list of API functions.

API
stzedn_apl_meterAdd
stzedn_apl_meterRemove
stzedn_apl_meterSetRfAdapter
stzedn_apl_meterSetKey

Table 10: API functions, which case a non-volatile write access imediately.

Another direct non-volatile memory write access takes place, when a Wireless M-Bus collector device calls the event stzedn_apl_evt_newMeter, if the return value is set to TRUE. In this case, a new meter which is currently not in the meterlist will be added to the meterlist.

3.4.3.2   Non-volatile memory management

Figure 3 shows the basic workflow of the Wireless M-Bus Stack regarding the memory management. After the start-up of the device and the initialization of the Wireless M-Bus Stack, the function stzedn_apl_start function must be called with appropriate parameters. The Wireless M-Bus Stack then synchronizes with the non-volatile memory.

Synchronization at stzedn_apl_start : In this function, the Wireless M-Bus Stack first checks, if the non-volatile memory contains valid information. If the non-volatile memory contains no valid information, the Wireless M-Bus Stack uses the parameters from the stzedn_apl_start function to initialize the device and stores the parameters afterwards into the non-volatile memory. If the information is valid, the parameters in the start function stzedn_apl_start were omitted and all parameters were loaded from the non-volatile memory. In this case, the Wireless M-Bus Stack also omits the optional static meterlist and loads the one in the non-volatile memory, even if the list is empty (only for collector). After calling this function, the Wireless M-Bus Stack is ready for operation.

Synchronization at stzedn_apl_sleep : When the Wireless M-Bus Stack is set into sleep mode, all internal parameters from table 9 were stored into non-volatile memory. If the function returns success, the user application is allowed to set the microcontroller into a hardware sleep mode.

Synchronization at stzedn_apl_wake : This function must be called to run the Wireless M-Bus Stack after a sleep mode. The stack synchronizes the buffered parameters in the volatile memory with the parameters in the non-volatile memory.

Synchronization at meterlist write/change operations: When the user changes the meter entries (add, change, remove), the Wireless M-Bus Stack writes the changes immediately into the non-volatile memory. The reason for this is that the meterlist entries were not buffered in the volatile memory.

Synchronization at meter installation mode: When a collector detects a new meter, it is possible to add the meter into the meterlist of the collector. The Wireless M-Bus Stack fires the event stzedn_apl_evt_newMeter and the user application can decide, if the collector should add the meter into the own meterlist. In case the collector adds the meter into the meterlist, the stack writes the meter parameters into the non-volatile memory immediately.

In all other cases, there is no non-volatile memory write access.
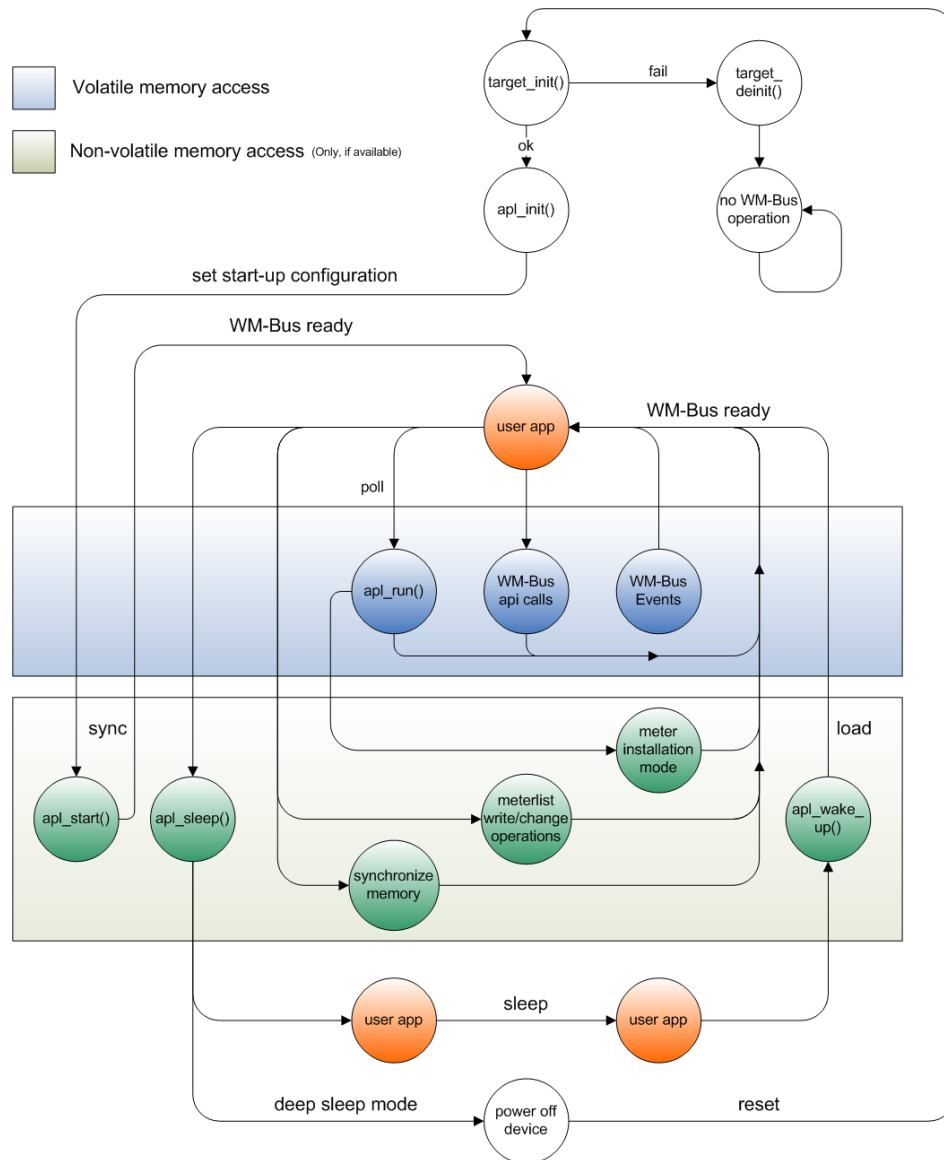
Figure 3: Non-volatile memory work flow.

3.5   Encryption Modes

Refer to table 11 for the supported encryption modes of the Wireless M-Bus protocol stack. The only mandatory encryption mode defined in OMS v.3.0.1 is mode 5.

Mode	Description
0	No encryption.
1	AES Counter Mode (AES-CTR).
5	AES Cipher Block Chaining Mode (AES-CBC) with dynamic initialization vector.

Table 11: Encryption Modes

3.6   Sleep modes

The stack provides several sleep modes, which can be used to save energy, while the stack is not busy.

- **Sleep**

  It is assumed that the mcu will enter a lpm state in which the RAM content will be kept. The radio will be set to power down mode.

- **Deep sleep**

  It is assumed that the mcu will enter a lpm state in which the RAM content will not be kept. The radio will be set to power down mode.

3.7   Wireless M-Bus Channels

The modes of Wireless M-Bus are using different channels. Table 12 lists all available channels. Depending on the device's mode, only a subset of this table can be accessed. Using channels that are not supported result in an error message at compilation of the firmware.

Mode	Frequency
N1a, N2a (Meter Device and Data Collector)	169.406 MHz
N1b, N2b (Meter Device and Data Collector)	169.418 MHz
N1c, N2c (Meter Device and Data Collector)	169.431 MHz
N1d, N2d (Meter Device and Data Collector)	169.443 MHz
N1e, N2e (Meter Device and Data Collector)	169.456 MHz
N1f, N2f (Meter Device and Data Collector)	169.468 MHz
N2g (Meter Device and Data Collector)	169.437 MHz
T2 (Data Collector TX, Meter Device RX)	868.30 MHz
T1, T2 (Meter Device TX, Data Collector RX)	868.95 MHz
S1, S2 (Meter Device and Data Collector)	868.30 MHz
C2 (Data Collector TX, Meter Device RX)	869.525 MHz
C1, C2 (Meter Device TX, Data Collector RX)	868.95 MHz

Table 12: Available channels

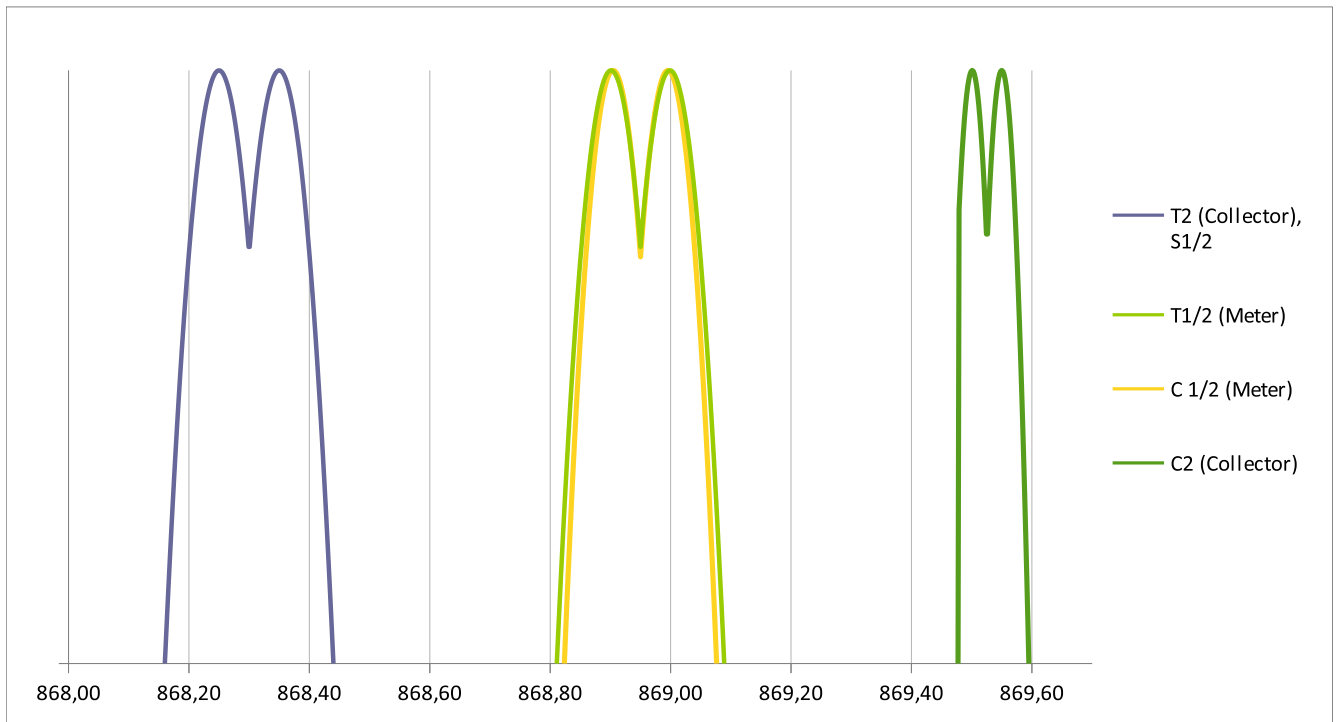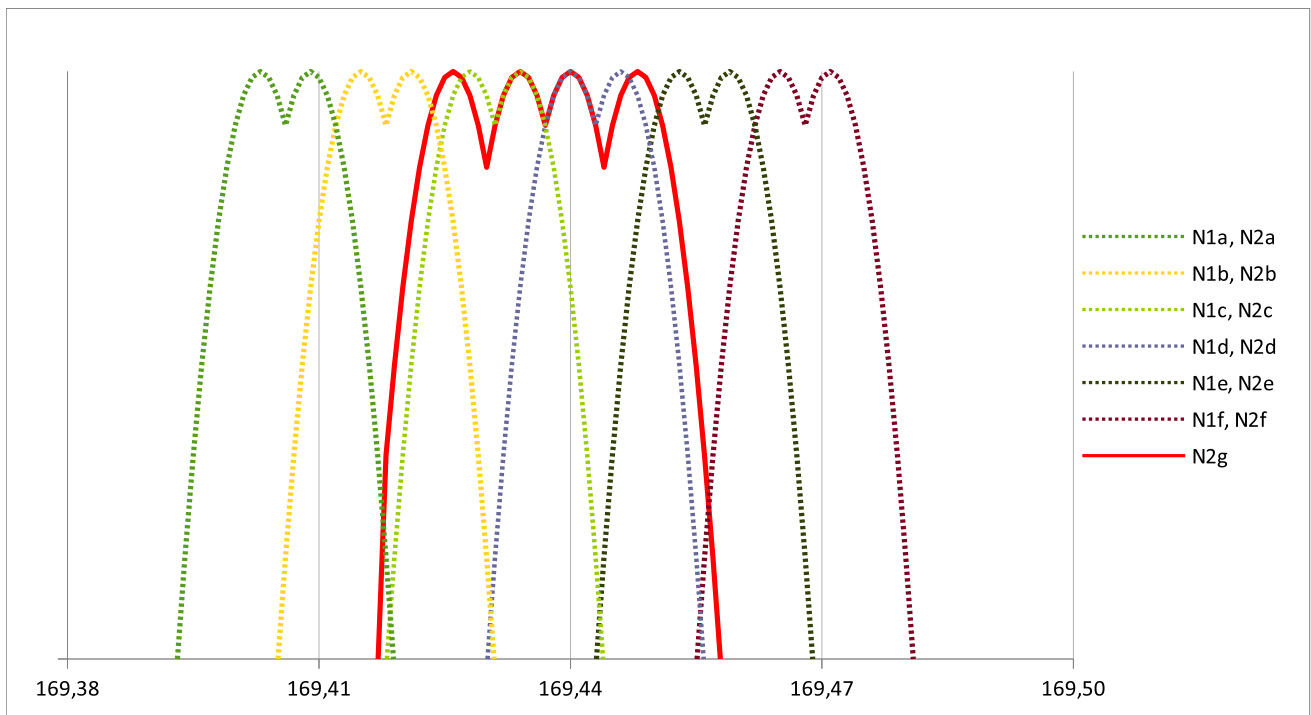Figure 4: List of frequencies used by Wireless M-Bus mode S, T and C

Figure 5: List of frequencies used by Wireless M-Bus mode N

Steinbeis Transfer Center
Embedded Design and Networking

4   Setup

4.1   Hardware

4.1.1   Supported Radio Types

The following EZRadioPRO radio ICs are supported by the STZEDN W-MBUS Protocol Stack:

1. Si4461 Transceiver

2. Si4463 Transceiver

The transceivers are delivered on RF Pico Boards. The radio module contains the radio IC, the corresponding matching network and the SMA connector matched to 50Ω.

Part Number	Board Description
4461-PCE14D868	868 MHz matching, Transceiver, 14dBm output power, Direct tied antenna connection
4463-PSQ20D169	169 MHz matching, Transceiver, 20dBm output power, Direct tied antenna connection

Table 13: Supported Transceivers and RF Pico Boards

4.1.2   Hardware requirements

The following devices are needed to build either a complete W-MBUS capable data collector or meter device.

Description	Part Number
MCU	EFM32LG-STK3600 Leopard Gecko Starter Kit
RF Adapter Board	MSC-RFP2EFM
RF Pico Board	4461-PCE14D868 / 4463-PSQ20D169
Antenna	868MHz / 169MHz

Table 14: Hardware requirements for a meter / collector device

4.1.3   Wireless M-Bus Hardware Platform

The W-MBUS Hardware Platform is a demo, evaluation and development platform for EZRadioPRO radio ICs. It consists of a Leopard Gecko Starter Kit, an RF Pico Adapter Board and an interchangeable RF Pico Board.
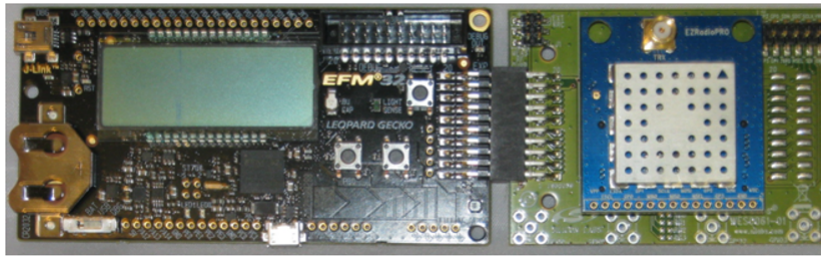
Figure 6: The 32bit W-MBUS Hardware Platform

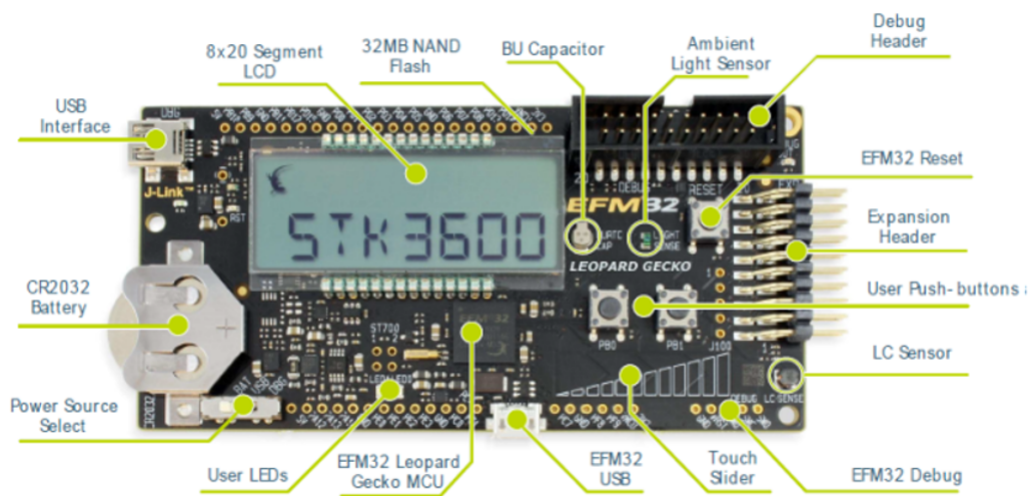### 4.1.3.1   EFM32LG-STK3600 Leopard Gecko Starter Kit

Figure 7: EFM32LGSTK3600 hardware layout

Many of the EFM32's pins are routed out to "breakout pads" at the top and bottom edges of the kit. A 2.54mm pitch pin header can be soldered in for easy access to these pins. On the right hand side of the board an angled 20 pin expansion header is provided to allow connection of peripherals or plugin boards. The connecter contains a number of I/O pins that can be used with most of the EFM32 Leopard Gecko's features. Additionally, the VMCU, 3V3 and 5V power rails are also exported.
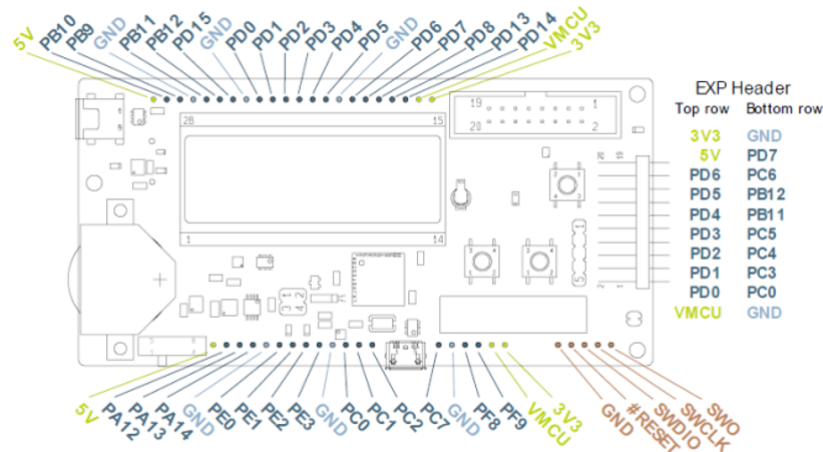
Figure 8: Breakout pads and expansion header

Figure 9: EXP Header Pins on Leopard Gecko Module

4.1.3.2 MSC-RFP2EFM Adapter Board

The adapter module makes it possible that the EFM32 Leopard Gecko MCU can be connected to the EZRadioPRO radio IC easily. On the left hand side of the board an angled 20 pin expansion header is provided to allow connection of EFM32 Leopard Gecko module. The same expansion header can be found on the right hand side. The digital signals of the radio (SCLK, MISO, MOSI, NSEL, SCLK, VRF, GND and RF_GPIOs) can be accessed on test pins at the top right side of the board. Four SMA connectors can be soldered on the bottom side for measurement purposes. The RF Pico Board can be connected in the middle of the adapter board. By default, the radio is powered through the VMCU pin. If the leopard gecko module is powered by a battery, the supply voltage must be provided to the radio through the external power source pin. The current consumption of the radio can be measured separately from the leopard gecko module on current measurement pin.
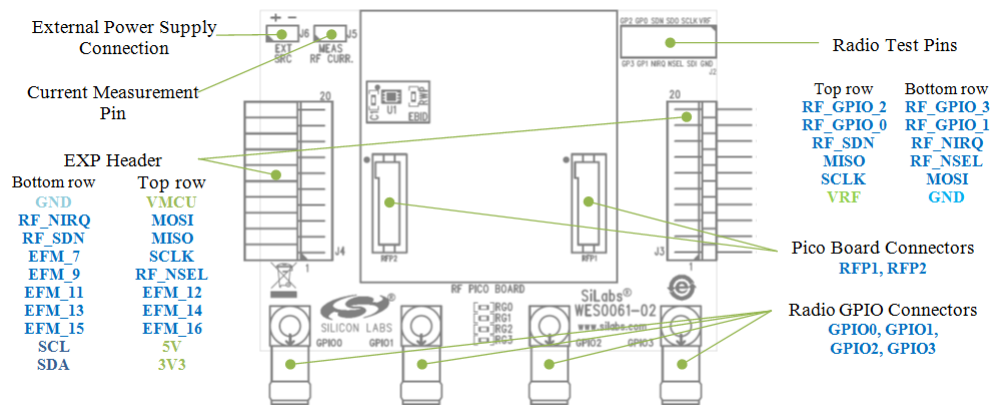
Figure 10: MSC-RFP2EFM Adapter Board connections

Figure 11: EXP header pins on MSC-RFP2EFM Adapter Board

Figure 12: RF Pico Board Pins on MSC-RFP2EFM Adapter Board

4.1.3.3   Connection between Leopard Gecko MCU and EZRadioPRO transceiver

The following table shows which MCU and radio pins are required to be connected for the operation of the stack.

Steinbeis Transfer Center
Embedded Design and Networking

Si446x transceiver	EFM32LG990F256
RF_SDN	PC3
RF_GPIO_0	PC4[1]
RF_GPIO_1	PC5[1]
MOSI	PD0
MISO	PD1
RF_NSEL	PD3
SCLK	PD2

Table 15: Pin compatibility

4.1.3.4   EZRadioPRO RF Pico Board
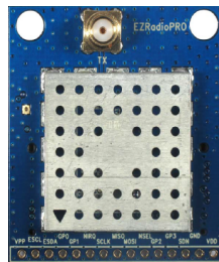
Figure 13: RF Pico Board Front Side

The RF Pico Board is a radio module that contains an EZRadioPRO radio IC, a matching network and an SMA connector on the top side. These components apart from the antenna connector are covered by a metal shield for noise reduction. The digital signals of the radio (SCLK, SDI, SDO, NSEL, SCL, SDA, VDD and GND) can be accessed on test points at the edge of the board. The boards also have a factory loaded board identification memory (EBID) on the bottom side that contains data that describes the board properties. Via the unified RF Pico connector pair on the bottom side of the board, any RF Pico board can be connected to the RF Pico Adapter Board.

4.1.4   Required hardware resources

The following table shows the hardware resources that are required from the MCU.

---

[1]Note that RG0=RG1=0Ω resistors must be soldered on the adapter board so that RF_GPIO_0 and RF_GPIO_1 will be available for the MCU. RF_GPIO_2 and RF_GPIO_3 are not used by the stack.

Function	Resources
MCU SPEED	28MHz
SPI Bus Speed	8MHz
Used Timer	TIMER_1
SPI Interface	USART_1_Location_1
USART Interface	USART_0_Location_2
Flash for Collector informations	0x3E800-0x40000 Collector
Flash for Meter informations	0x3E800-0x3F8FF Meter
Used RF GPIOs of radio	RF_GPIO_0, RF_GPIO_1

Table 16: Hardware Requirements

The following table shows the memory consumptions of the different configurations for collectors and meters.

Configurations	Read-only code memory	Read-write code memory	Read-only data memory	Read-write data memory
Collector C2 with serial	47272	288	1000	6135
Collector C2 with apl	32204	288	760	4787
Collector T2 with serial	45328	288	960	6103
Collector T2 with apl	30264	288	720	4755
Collector S2 with serial	45156	288	708	6103
Collector S2 with apl	30088	288	468	4755
Collector N2 with serial	45260	288	972	6139
Collector N2 with apl	30196	288	732	4791
Meter C2 with serial	43472	288	716	6124
Meter C2 with apl	29300	288	472	4766
Meter T2 with serial	41900	288	716	6124
Meter T2 with apl	27660	288	472	4766
Meter S2 with serial	41732	288	676	6124
Meter S2 with apl	27492	288	432	4766
Meter N2 with serial	41988	288	940	6128
Meter N2 with apl	27748	288	696	4770

Table 17: Memory consumption

4.2   Firmware

This section describes how to setup the firmware of the EFM32LG-STK3600 in use with the Wireless M-Bus Suite. Please note, that if you want to set up the enclosed demo network with the quick start for Wireless M-Bus Suite in chapter 5, you have to configure at least two devices: one with the collector and one with the meter firmware. The EFM32LG-STK3600 firmware can either be provided as binary[2] ('*.bin'file) or as source code that is compiled and flashed using IAR compiler. Please check, if a valid serial firmware is available on your boards.

You can do this with Wireless M-Bus Suite with a ping test. If there is no firmware available, refer to chapter 4.2.1 for firmware installation using Simpl. Studio or refer to chapter 4.2.2 for installation using IAR IDE. .

---

[2]Note that these binaries are SERIAL related and the location in our file hierarchy is at the location of ...\WMBUS Tools\Firmware\Serial binaries\

4.2.1   Firmware Installation with Simplicity Studio

To install a firmware using Simplicity Studio the following steps must be performed.

1. Connect the mini USB port of the target board to your PC using a mini USB cable (J-Link interface)

2. Connect the micro USB port of the target board to your PC using a micro USB cable (serial interface).

3. Open the Simplicity Studio software (Studio software [8]).
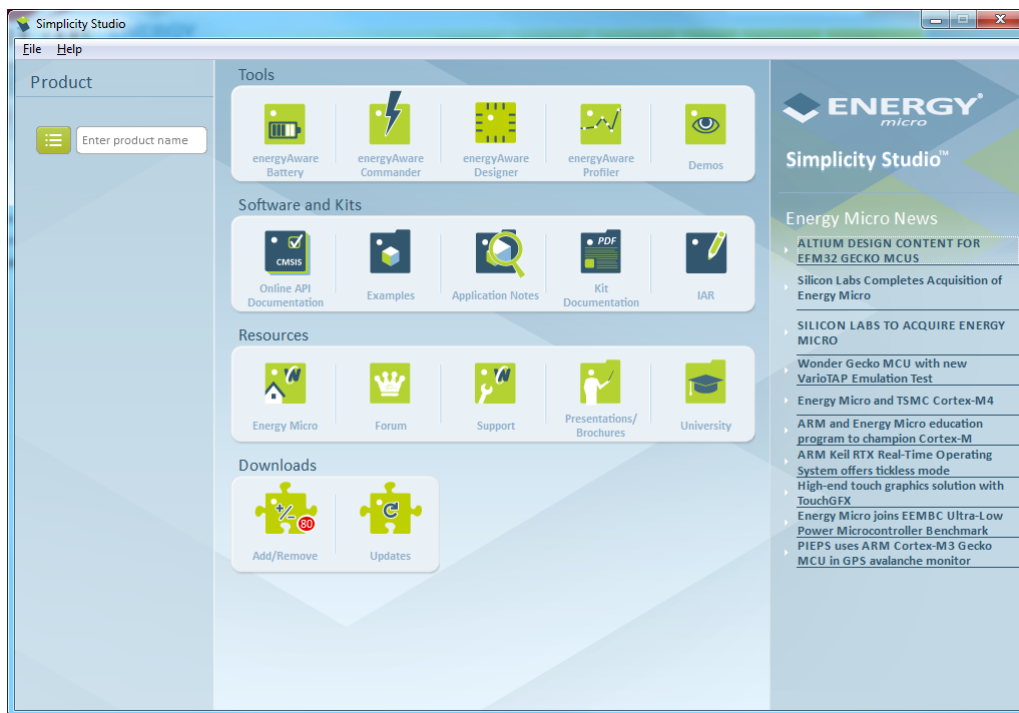
Figure 14: Simplicity Studio

4. Select "energyAwareCommander"

Steinbeis Transfer Center
Embedded Design and Networking
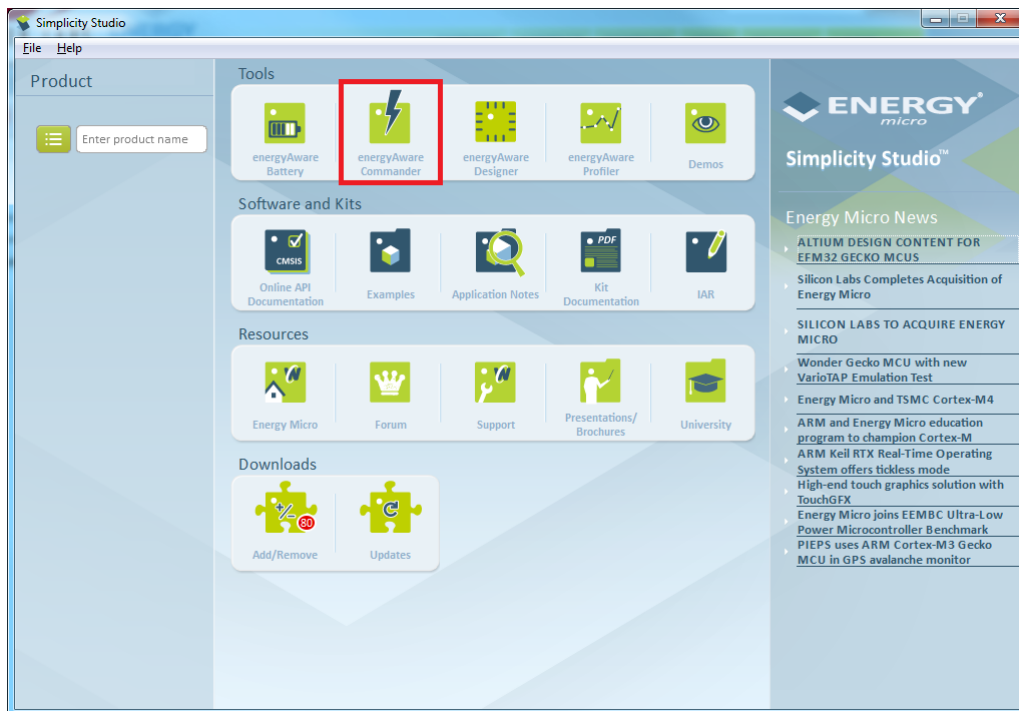
Figure 15: Select "energyAwareCommander"

5. The energyAwareCommander application will start.

6. Select the connected J-Link-Decvice and push the "Connect" button.
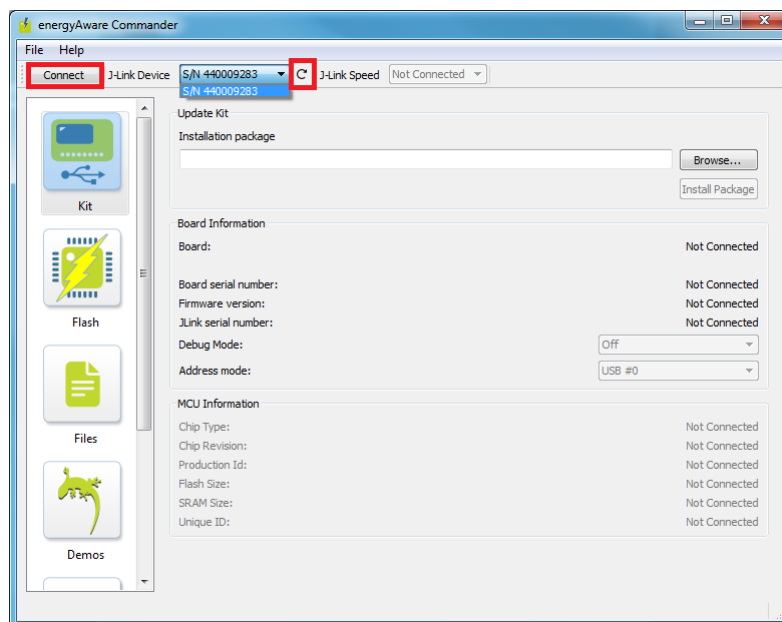
Figure 16: Select the connected board

7. Select option "Flash".
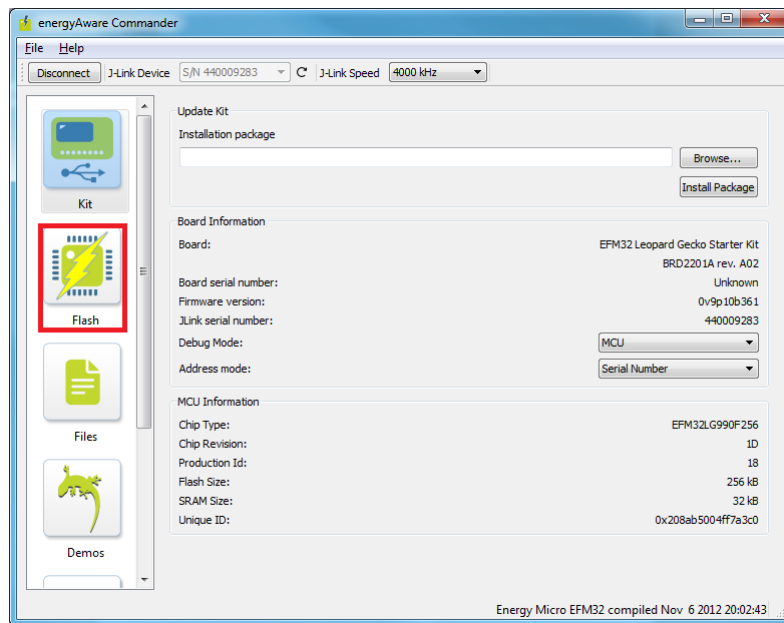
Figure 17: Select "Flash"

8. Choose a "*.bin" file and push the button "Flash EFM32".
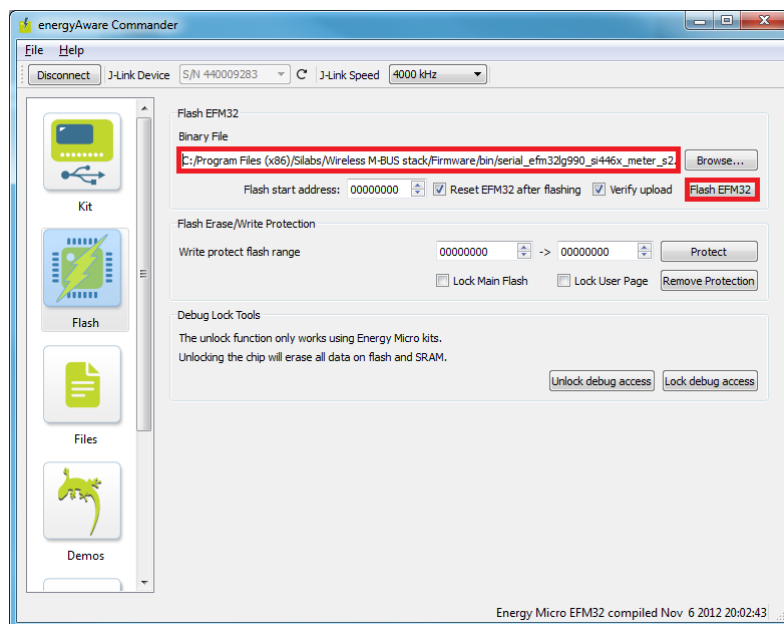
Figure 18: Flash the selected firmware

9. Wait until the J-Link-Device is programmed successfully.

Figure 19: Download

10. To program additional devices, repeat the steps as described above. Make sure to upload the appropriate firmware for each device.

11. Now your boards are ready to run the application.

4.2.2   Install Firmware with IAR compiler

1. Connect the mini USB port of the target board to your PC using a mini USB cable (J-Link interface).

2. Connect the micro USB port of the target board to your PC using a micro USB cable (serial interface).

3. Open the project file (*.eww) with the IAR ARM compiler.

4. There are different configurations available in the workspace window: One for the meter device and one for the collector device with the different supported modes. Refer to figure 20 for the selection.
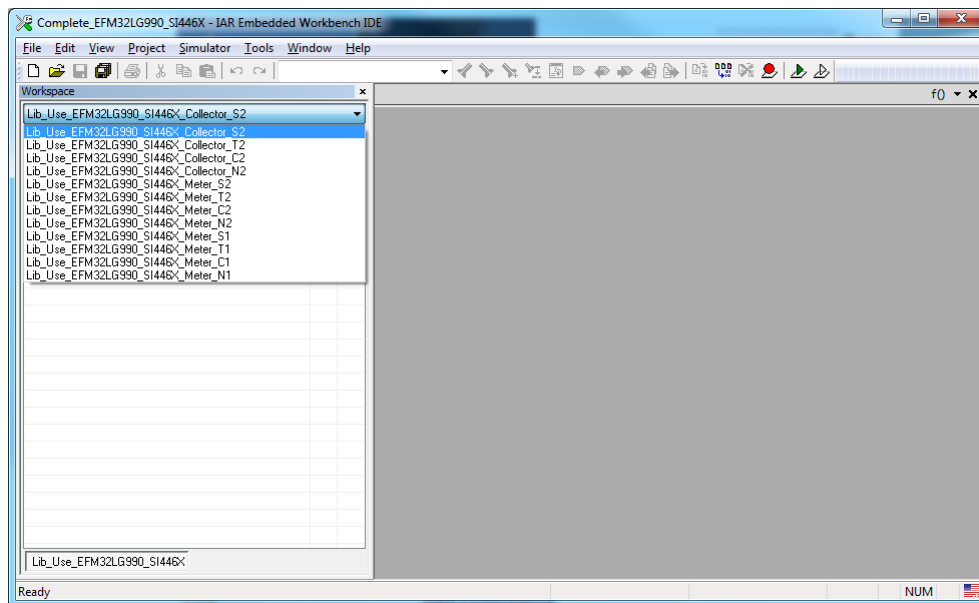
Figure 20: Workbench configuration

5. Select a workspace configuration.

6. Before debugging, please check, if the 'Debugger' is selected in the project options. Right-click on the project and select 'Options'.

Steinbeis Transfer Center
Embedded Design and Networking
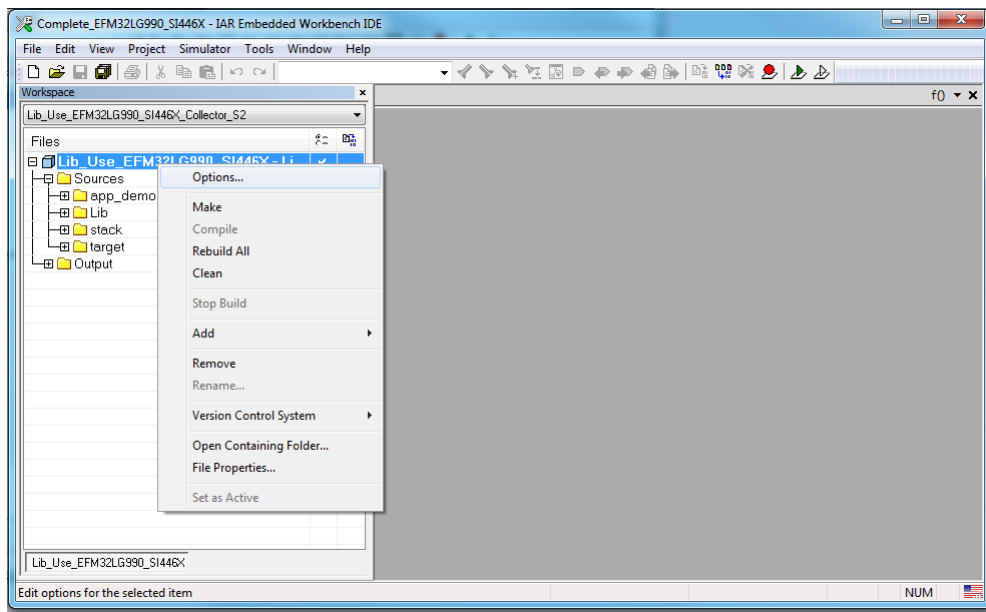
Figure 21: Open project options

7. Switch to the tab "Debugger" and verify that the "J-Link/.." option is selected.
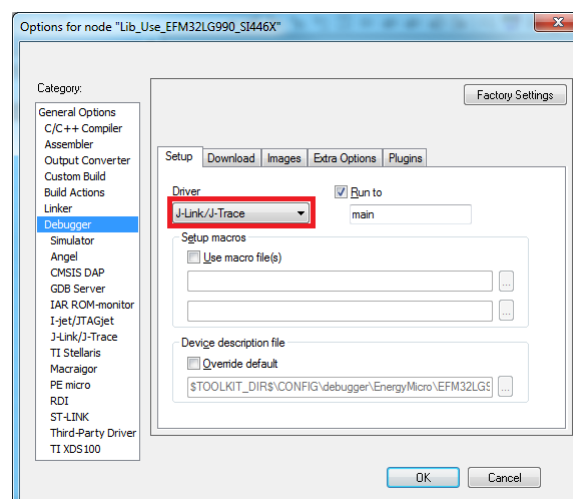
Figure 22: Driver selection

8. Now you can download and debug the source code.
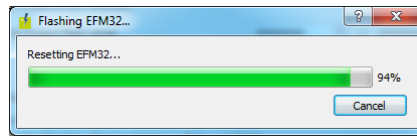
9. To program additional devices, repeat the steps as described above. Make sure to upload the appropriate firmware for each device.

5   Wireless M-Bus Suite

5.1   System Requirements

The Wireless M-Bus Suite is available for 32-Bit and 64-Bit Win XP/WinVista/Win 7 operating systems. The tool detects the version of the installed java runtime environment and notifies the user if a newer version is available. It requires the user to download at the official java homepage (www.java.com). Recommended minimum system requirements are:

- 512 MB RAM for Win XP and 1 GB RAM for Win Vista / Win 7

- 1 GHz CPU

- 100 MB free hard disk space

- Serial interface or USB-to-Serial connector

Before installing Wireless M-Bus Suite, please check if your operating system is 32-Bit or 64-Bit:

- $WindowsXP$: Open system information: please select Start->Run. Input 'winmsd.exe' and press 'enter'. Select 'System Summary'. Check the 'System Type' parameter: if the value is $X86-basedPC$, it is a 32-Bit Windows XP, if the value is $X64-basedPC$, it is a 64-Bit Windows XP.

- $WindowsVista$/7 : Please go to Start->System control->System. The operating system settings can be found under the parameter 'System Type'.

Please make sure to run the correct installer for your system type.

5.2   COM Port Settings

This chapter describes how to get information about the com pot settings on Windows operating systems.

Go to 'Start->System control->System->Hardware->Device manager' and check the 'COM and LPT' connections. Refer to figure 23 and 24 for an example COM port configuration in a device manager of a PC. In the following, the numbers in angle brackets were used to refer to the red numbers in figure 23 and 24. Please expand the node <1> 'Ports(COM & LTP)'. Based on physical connection, there are two types of COM ports to distinguish:

- Physical COM port of the PC <2> (serial interface, usually COM1)

- Virtual COM ports
  In case of a USB connection, the PC will assign a virtual COM port to the USB interface as shown by <3> in figure 24, COM10. Please note, that the COM port number assignement can vary. If you are not sure which COM port was assigned to your USB connection (and your device), open the 'Device Manager', unplug the USB cable. Please wait three seconds and plug in the USB cable again. The 'Device Manager' and the available COM ports of the PC updates automatically and new devices will be added to the COM port list.

Please note, that connections declared with 'LPT' are no COM ports and cannot be used for a serial communication.
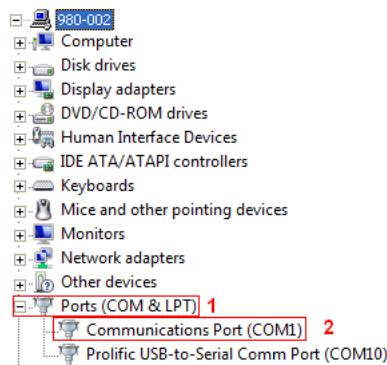

Figure 23: Standard COM port
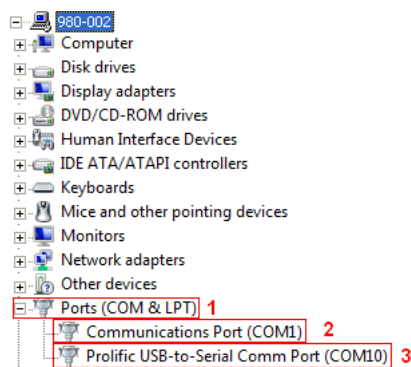
Figure 24: Virtual COM port

This chapter demonstrates how to start a basic Wireless M-Bus network with the Wireless M-Bus Suite. Wireless M-Bus Suite comes with a demonstration network project. The project settings are stored in configuration files (*.xml). This quick start guide uses the demonstration project to show the basic concepts of the Wireless M-Bus Suite. The provided demo configuration file includes one collector and one meter device, which communicate via Wireless M-Bus. Please note that you need to have at least two compatible evaluation boards with a valid application firmware. You will need one board with the collector and one board with the meter firmware to use the demo application. The first step is to connect the boards to your pc and to power them on.

The following chapters of the quick start guide can be read independently. Chapters 5.3 to 5.5 provide information, how to load the demonstration project as well as basic information, which are used for collector and meter devices. Chapters 5.7 and 5.8 specify how to perform transmission tests. However, it is recommend to read all chapters in order of appearance.

5.3   Loading the demo project

The demo project is integrated in the Wireless M-Bus Suite and can be opened after starting the Wireless M-Bus Suite. Please perform the following steps:

1. Start the Wireless M-Bus Suite.

2. A 'Welcome' page will be displayed. Press 'Watch the tutorial' to open a video tutorial for the Wireless M-Bus Suite.

3. Press 'Open the demo Project'.

4. This loads the 'wmbs_demo.xml' file from the Wireless M-Bus Suite installation folder.
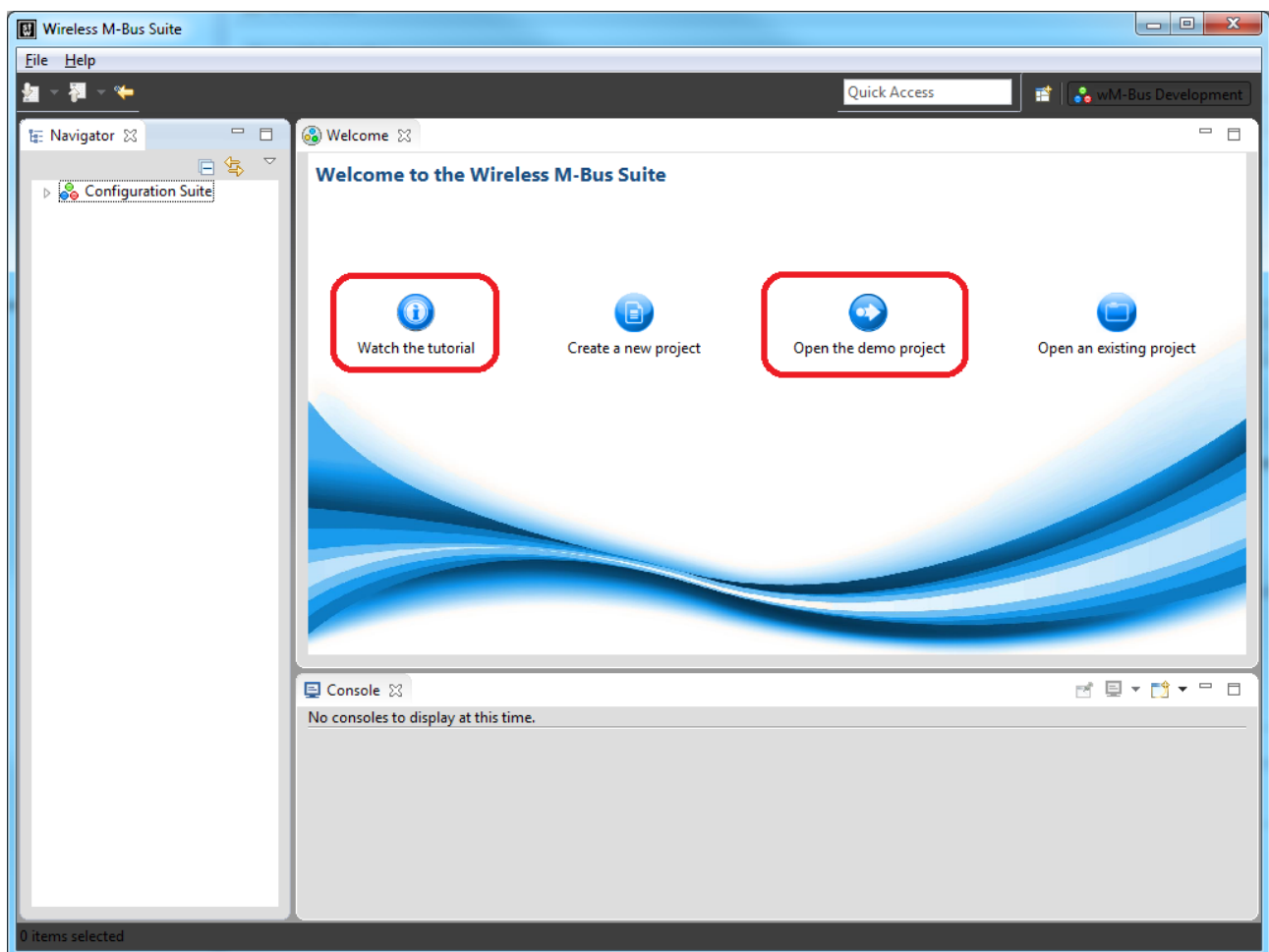

Figure 25: Start -up dialog

5. Now you can see the 'Data Collector' and the 'Smart Meter' in the configuration suite tree. In figure 26, the correspondig items are framed in red.

Steinbeis Transfer Center
Embedded Design and Networking
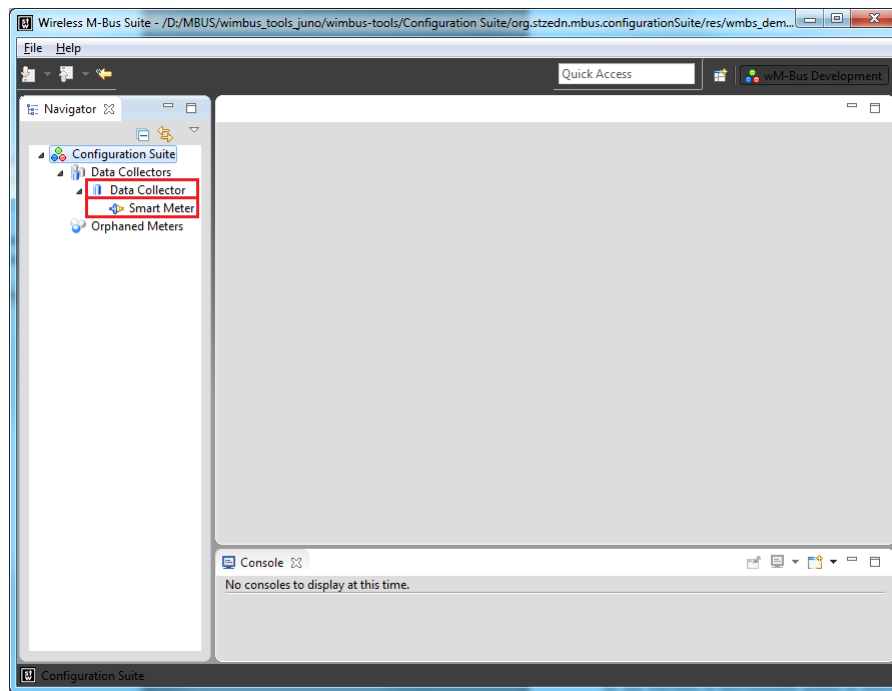
Figure 26: Collector and Meter device

Now the 'wmbs_demo.xml' is loaded. In the configuration suite tree, the 'Smart Meter' is located in the subtree of the 'Data Collector'. This means, that the 'Smart Meter' is assigned to the Wireless M-Bus network of the 'Data Collector'.

5.4   Collector short description

The collector parameters can be displayed by double clicking on the 'Data Collector' in the navigation tree. By default, it has a device address and an encryption key. There are several options in the 'Collector configuration' window, which are grouped in different sections. These sections (e.g. 'Collector operations' or 'Collector details') can be expanded individually (refer to figure 27 - red framed window in the 'Collectors operations' section).

The red framed window in the section 'Collectors meter list' in figure 27 shows the meter devices, which are associated with this collector. In this case it is the 'Smart Meter' device.
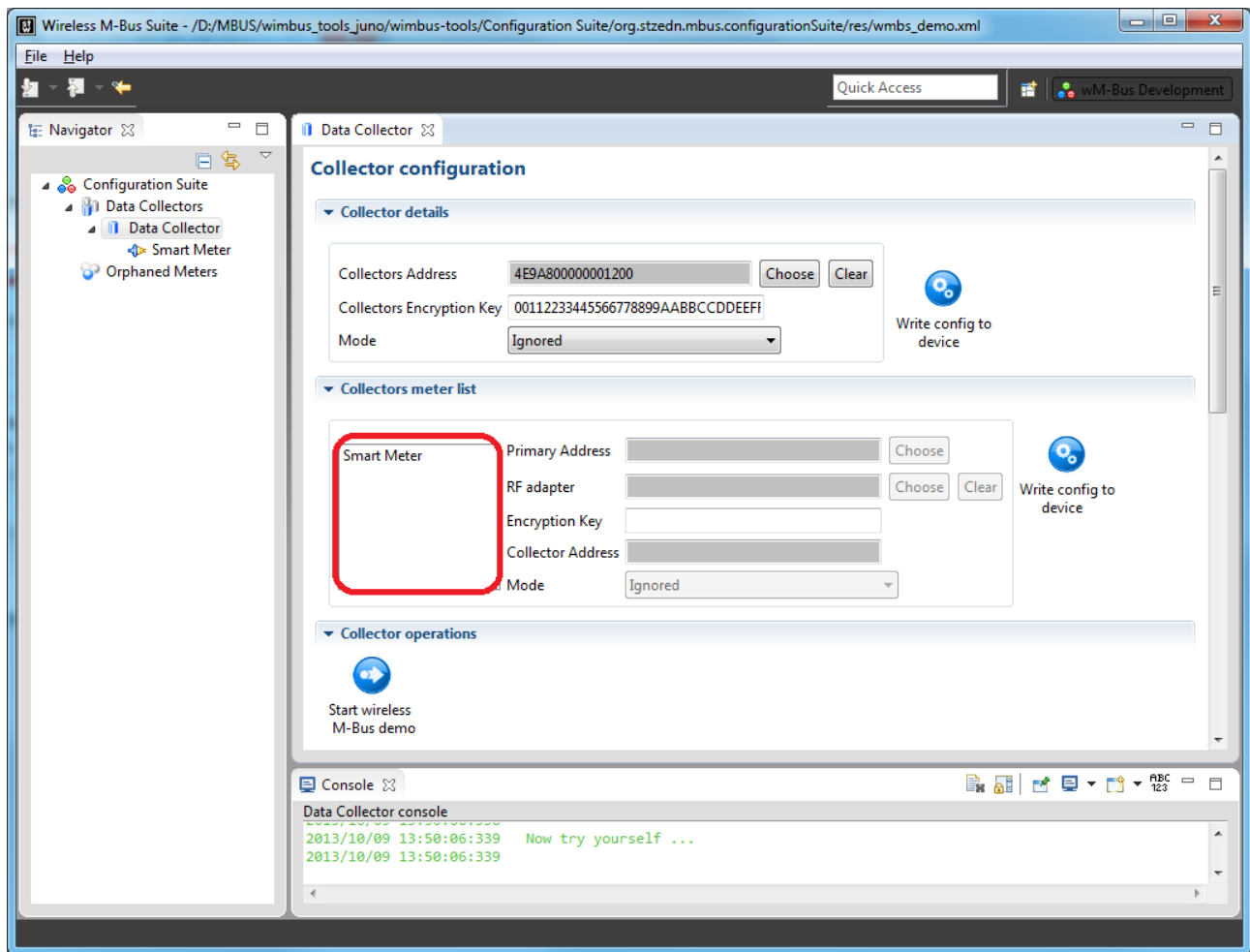
Figure 27: Collector editor

If the 'Smart Meter' device is selected in the red framed window, the corresponding parameters such as device address etc. are displayed. It has a unique device address, an encryption key and the address of the collector device. Note that the collector device address must be consistent with the parameters of the 'Data Collector' device. Otherwise no communication will be possible. The status of the configuration of the 'Smart Meter' meter and the 'Data Collector' will be displayed with the icon of the 'Smart Meter' in the navigation tree. A well defined connection is displayed with the icon in figure 28, otherwise figure 29 notifies the user that the meter is not connected. Figure 30 appears, if a meter has no collector address registered or the collector has no address configured. If two meters within the Wireless M-Bus network have the same device addresses, figure 31 is shown.

Figure 28: Meter connected

Figure 29: Meter disconnected

Smart Meter

Figure 30: Meter without collector address

Smart Meter

Figure 31: Meter addresses duplicate

To start the Wireless M-Bus network with the default settings, please press 'Start wireless M-Bus demo' and select the COM port related to the 'Data Collector' device.

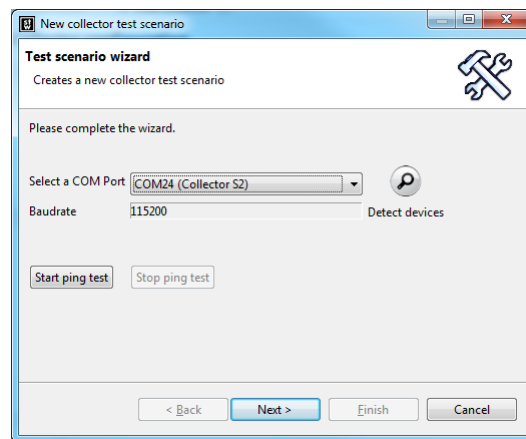Figure 32: COM Port selection

Now press 'Next' and select the meters which shall be used together with the collector board and press 'Finish'.
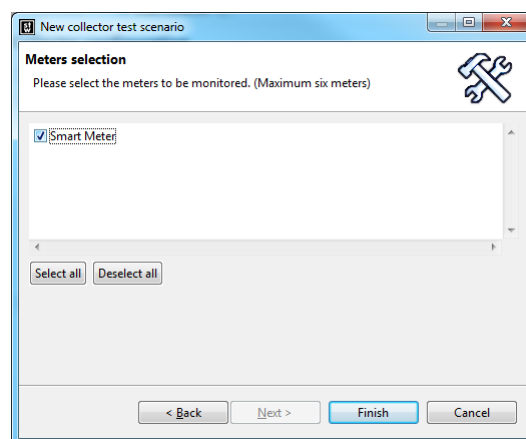
Figure 33: Select 'Smart Meter' meter

After this, settings are stored in the data collector's flash memory. If the device was programmed correctly, the following dialog is shown.

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus
wireless
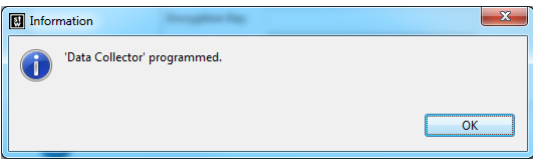
Figure 34: 'Data Collector' programmed

5.5   Meter short description

You can view the meter parameters by double clicking on the 'Smart Meter' in the navigation tree. The 'Smart Meter' editor is shown in the following figure. There are several options in the 'Meter configuration' window, which are grouped in different sections. These sections (e.g. 'Meter operations' or 'Meter details') can be expanded individually.
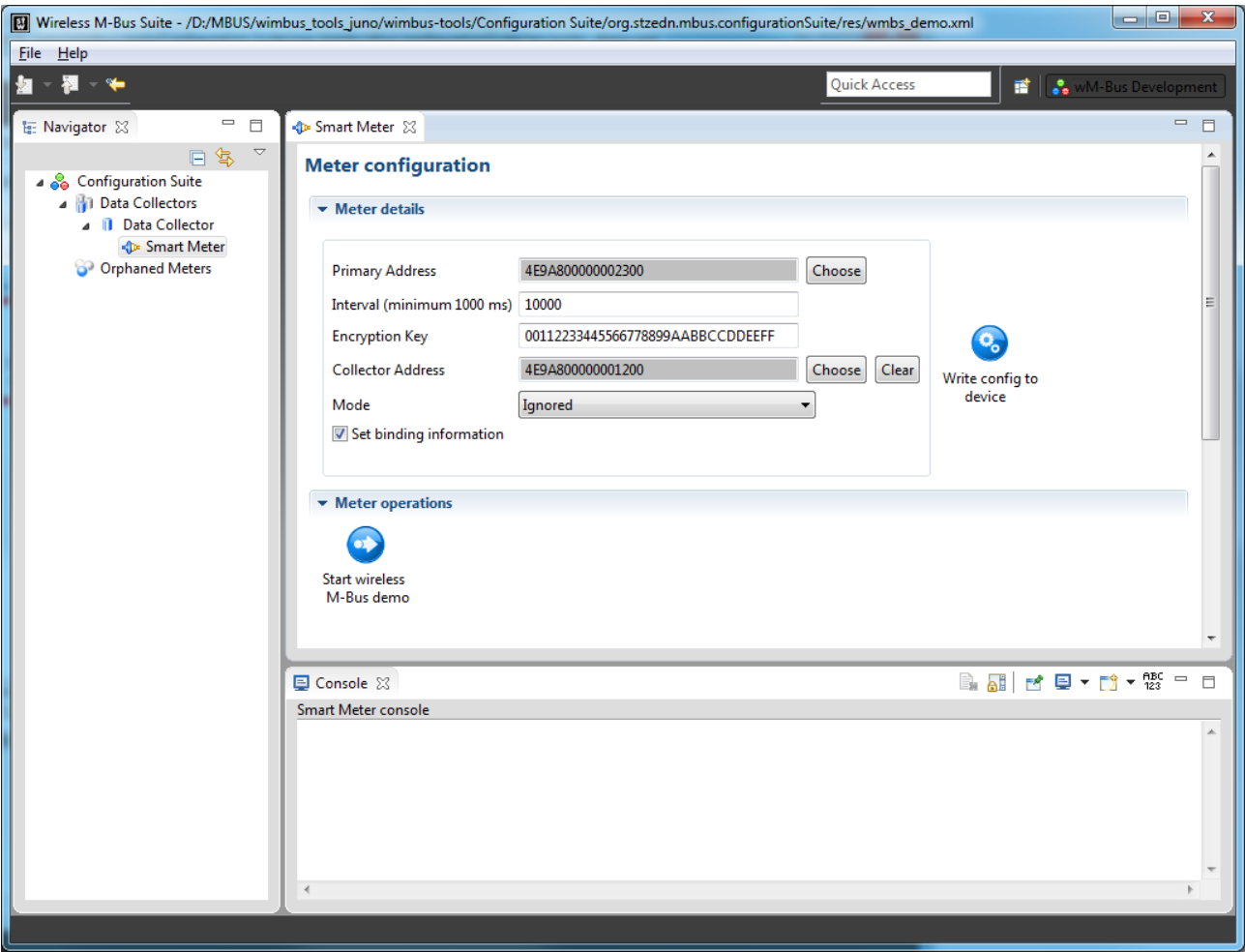
Figure 35: 'Smart Meter' Window

Table 18 lists the parameters of the 'Smart Meter':

Steinbeis Transfer Center
Embedded Design and Networking

Parameter	Purpose
Device address	Address of the 'Smart Meter'.
Collector address	Address of the 'Data Collector', to which the 'Smart Meter' sends data.
Encryption key	The key encrypts the Wireless M-Bus wireless telegrams.
Binding information	This checkbox sets the 'Smart Meter' to the connected state.
Time interval	The time interval [ms] of periodical data.

Table 18: 'Smart Meter' Parameters

The 'Smart Meter' meter will send messages in the configured time interval. The 'Set binding information' checkbox is used to set manual connections. Please refer to the manual of the Wireless M-Bus Suite for detailed information regarding this attribute.

To start the network using the default settings, please press 'Start wireless M-Bus demo', select the COM port related to the 'Smart Meter' device. (refer to figure 32). Then press 'Finish' and the device will be programmed. If the device was programmed correctly, the following dialog is shown.
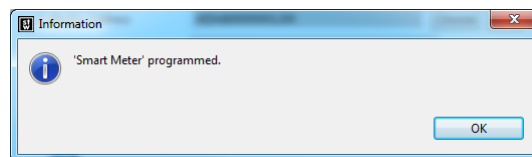
Figure 36: 'Smart Meter' programmed

In the demo project, the 'Smart Meter' sends a default packet to the collector every 10 seconds.

5.6   Using the Wireless M-Bus Suite

This chapter describes specific device configurations and special functions of the Wireless M-Bus Suite.

5.6.1   Mode selection

The 'Data Collector' and the 'Smart Meter' window includes a 'Mode' selection box. The Wireless M-Bus mode can be selected here. The selection of the mode is necessary only in case of mode C or mode N. The following picture shows the mode selection boxes.
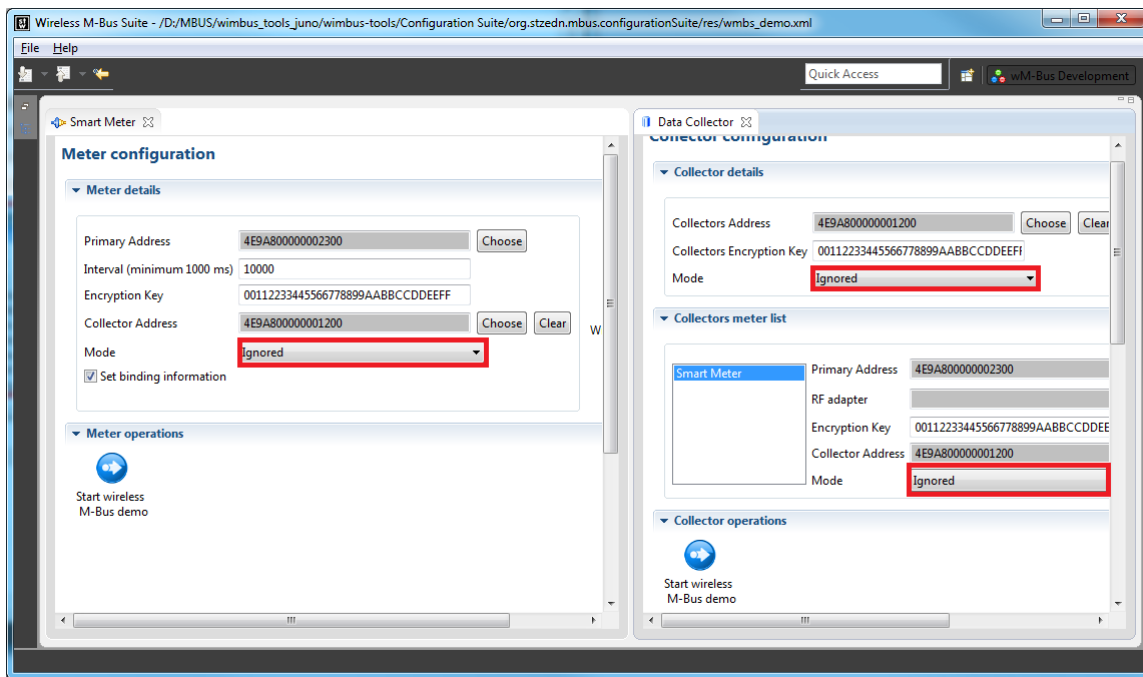
Figure 37: Mode selection boxes

The selection box can be ignored for modes S and T, as well as for a mode C meter device. If a C collector shall be used with a T meter, please set the meter to mode T (as shown in figure 38). The reason is that the collector needs the information of the supported channels of the meter. Otherwise the collector will only be able to receive packets of the meter (RX), but is not able to send packets to the meter (TX).
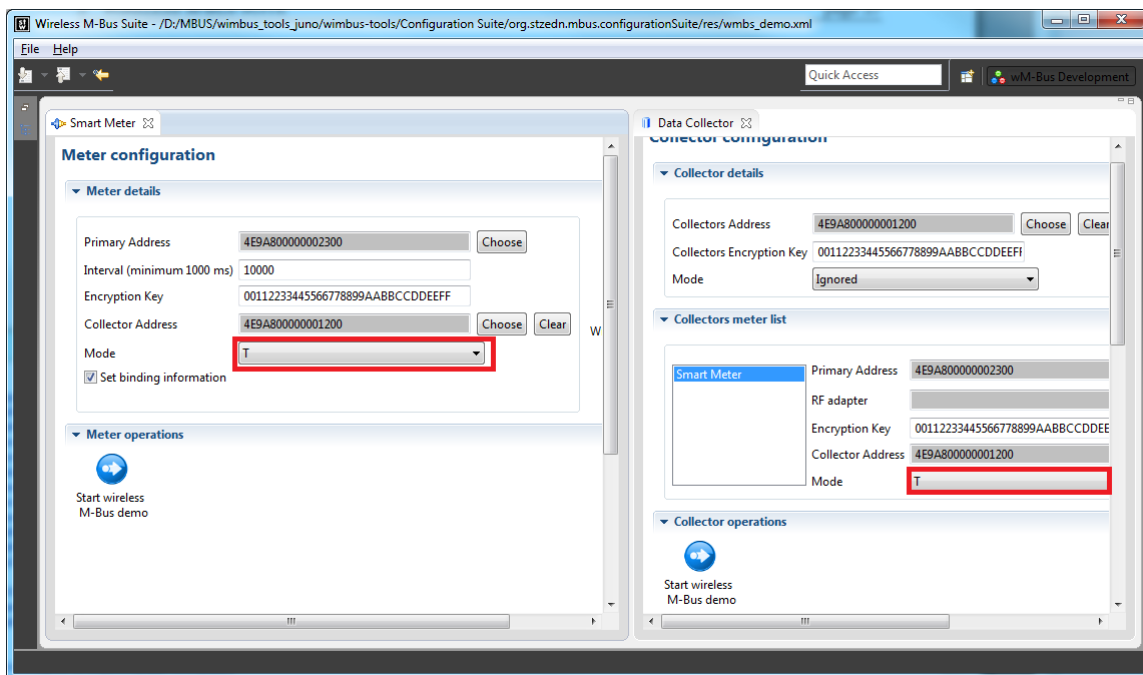
Figure 38: Collector C and Meter T

Steinbeis Transfer Center
Embedded Design and Networking

For using the device in mode N the selection box must be used to select and configure the channel. The figure 39 shows an example for the mode Nb.

Figure 39: Mode Nb example

5.6.2   Console

The Console View displays the serrial communication flow.

Figure 40: Console View

The commands available in the Console View are listed below.

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus

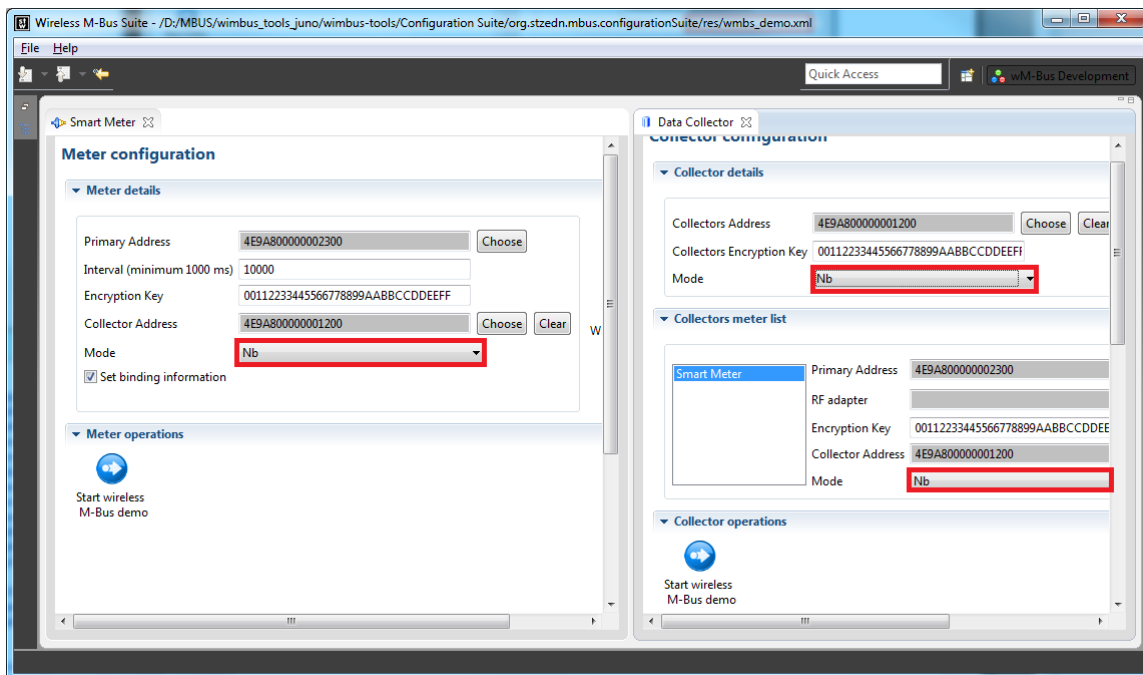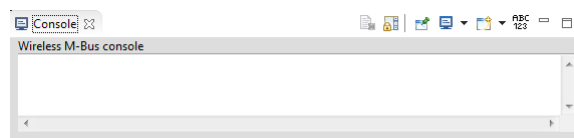| Command	Name	Description
	Clear Console	Clears the currently active console.
	Display Selected Console	Opens a listing of currently opened consoles and allows to select the one to display.
	Open Console	Opens a new console of the selected type.
	Pin Console	Pins the current console to remain on top of all other consoles.
	Scroll Lock	Changes if scroll lock should be enabled or not in the current console.
	Text on/off	Enables/Disables the serial text output.

Table 19: Console View Commands

The 'Text on/off' button enables or disables the text output for the serial commands (see figure 41).
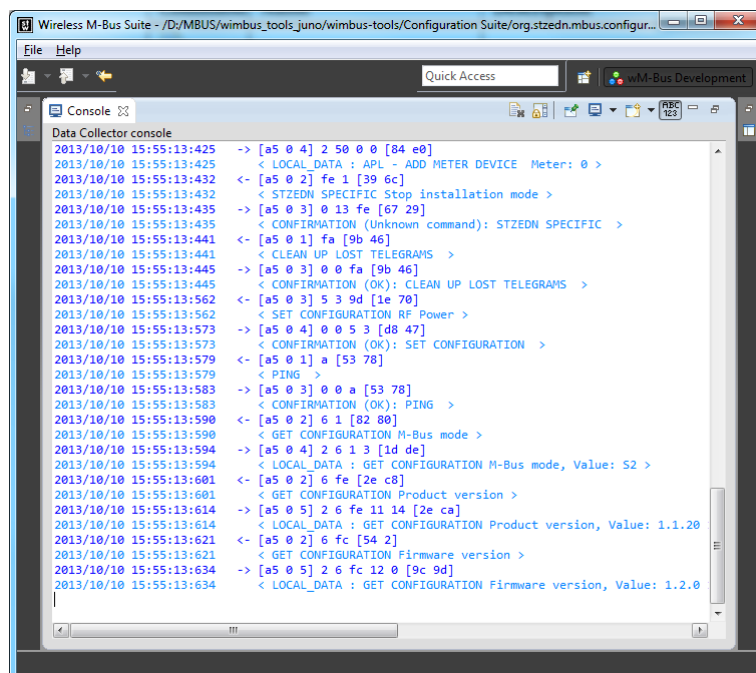
Figure 41: Console with serial text output enabled

5.7   Perform a ping test

This chapter describes how to send a ping message to a device using the Wireless M-Bus Suite. A ping is a simple message over the UART interface which is used to verify if a firmware is available on the connected board. If your board responds to the ping request, a firmware is currently being executed. To send a ping message to your board, perform the following steps:

1. Make sure that your device is connected to your PC properly.

2. Load the demo project as described in chapter 5.3.

3. Double click on the 'Data Collector' device in the device tree. Press on the red framed button 'Write config to device' or 'Start wireless M-Bus demo' (refer figure 42).
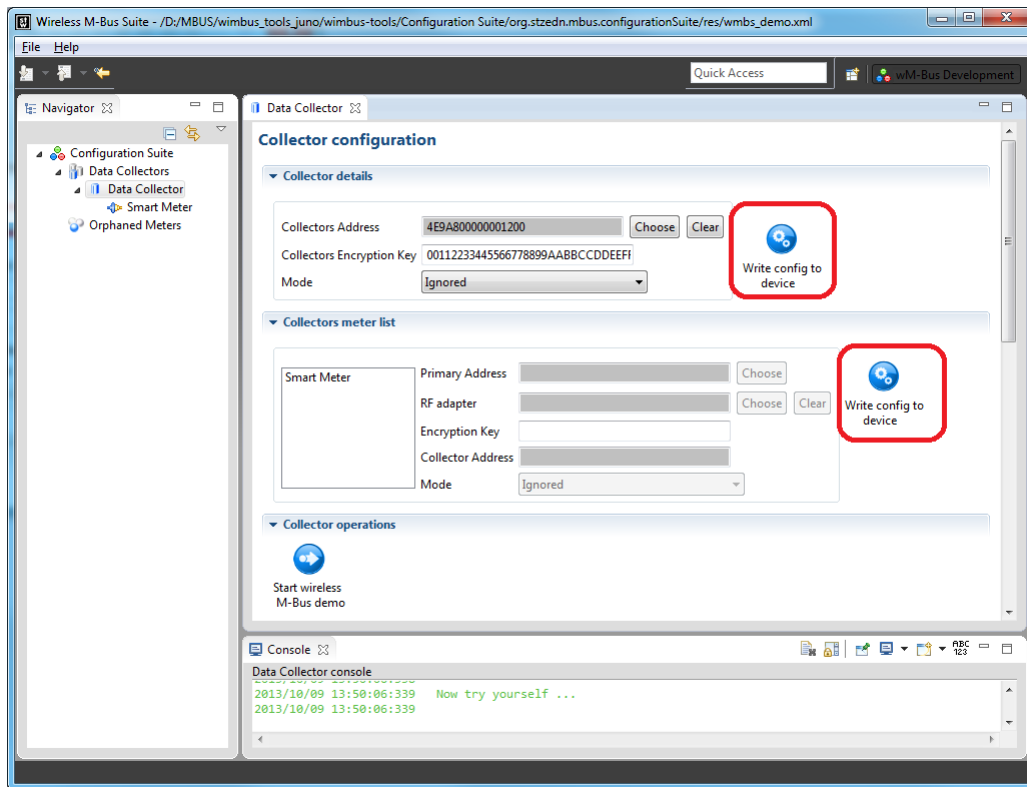
Figure 42: Write config to device

4. The wizard shown in figure 43 pops up. Configure the COM port settings to the correct COM port. Then press 'Start ping test'.
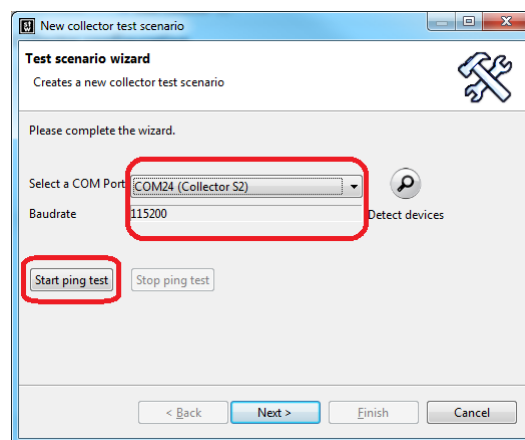
Figure 43: Start ping test

The Wireless M-Bus Suite now tries to send a ping message to the device. If the connection settings are configured correctly and the ping test fails, the device does not execute a firmware and dialog 44 will pop up. In this case you have to load a firmware on your device, before you can perform the next steps in chapters 5.4 to 5.8.
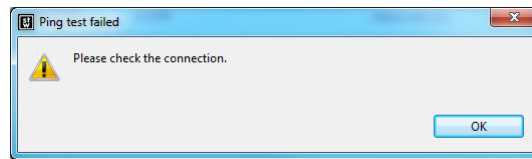
Figure 44: Ping test failed

If the ping test is successful, the console window will show the communication of the ping request and the response of the board. Refer to figure 45. If the ping test is successful, you can stop the ping test with the 'Stop ping test' button.
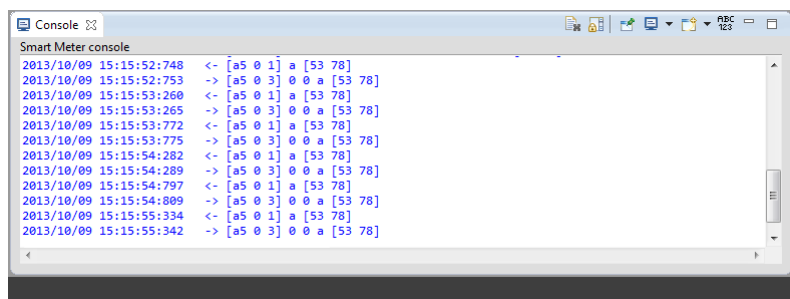
Figure 45: Ping test ok

5.8   Using the transmission tests

Perform the following steps for the 'Data Collector' collector:

1. Open Wireless M-Bus Suite

2. Load the demo project as described in chapter 5.3.

3. Open the 'Data Collector' device by double clicking on the icon in the navigation tree.

4. Press 'Start wireless M-Bus demo' and select the COM port of the collector device.
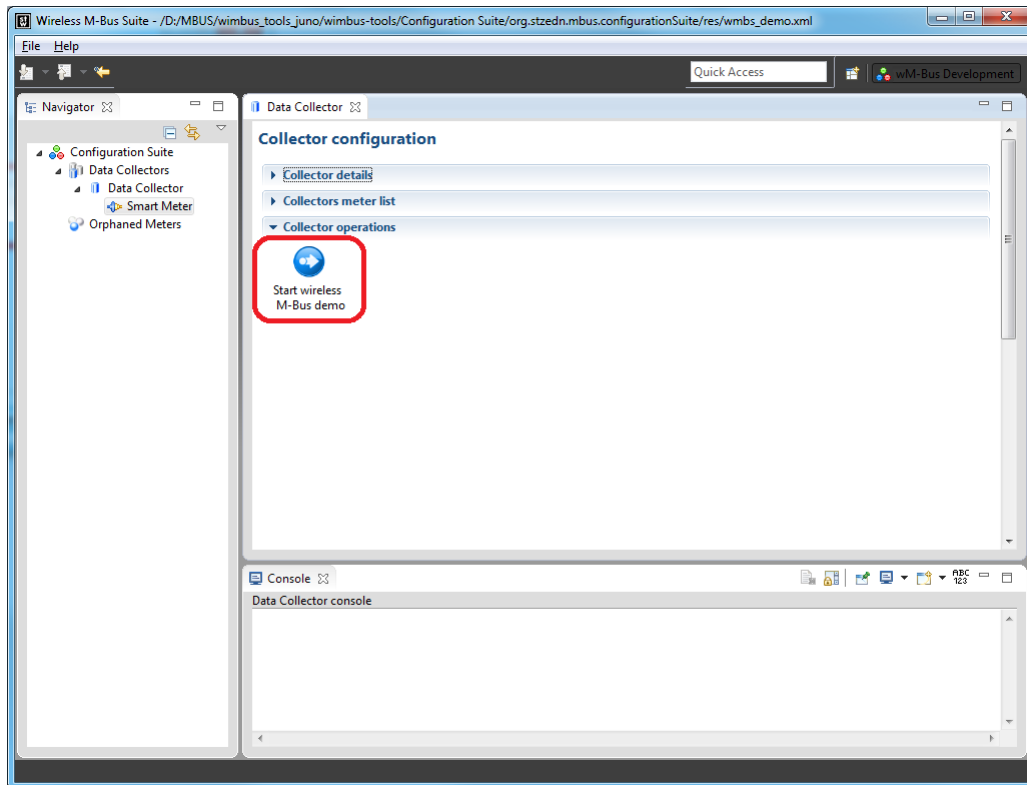
Figure 46: 'Start wireless M-Bus demo' button 'Data Collector'

5. Press 'Next' and set the checkbox for the 'Smart Meter' meter. (refer to figure 33)

6. Open the 'Smart Meter' device by double clicking on the icon in the navigation tree.

7. Press 'Start wireless M-Bus demo' and select the COM port of the meter device.

Steinbeis Transfer Center
Embedded Design and Networking
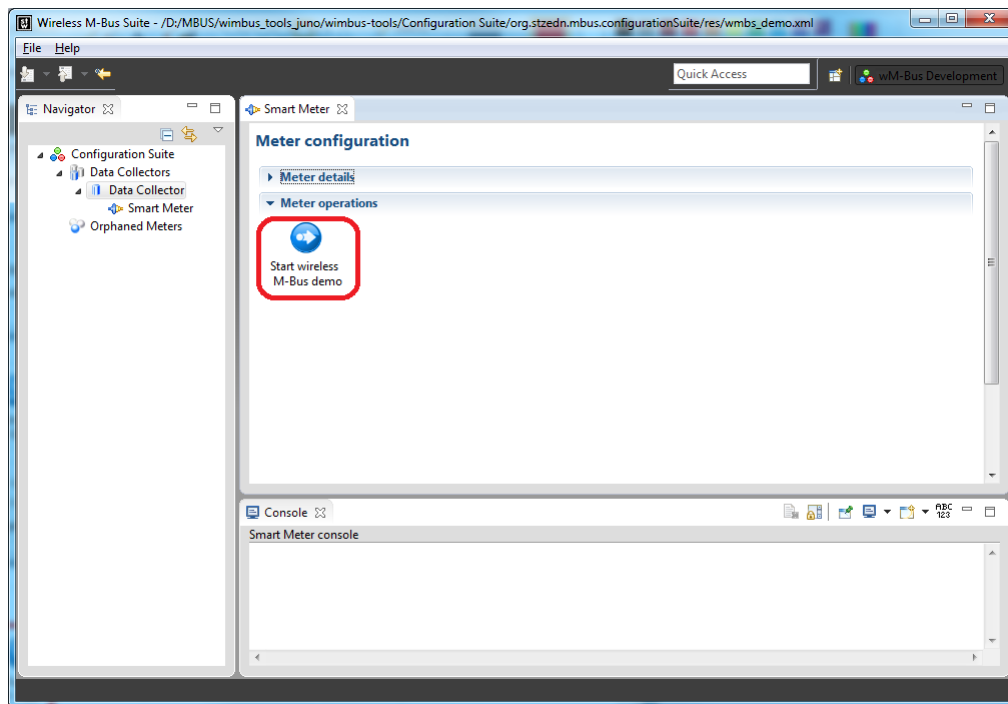
Figure 47: 'Start wireless M-Bus demo' button 'Smart Meter'

8. Press 'Finish'.

The meter device will now send every time interval a message to the collector. Make sure that the 'Connection status' is correct. The 'Data Collector'device will receive the data which is sent by the 'Smart Meter' meter. The *RSSI* value is displayed in a diagram, shown in figure 48.
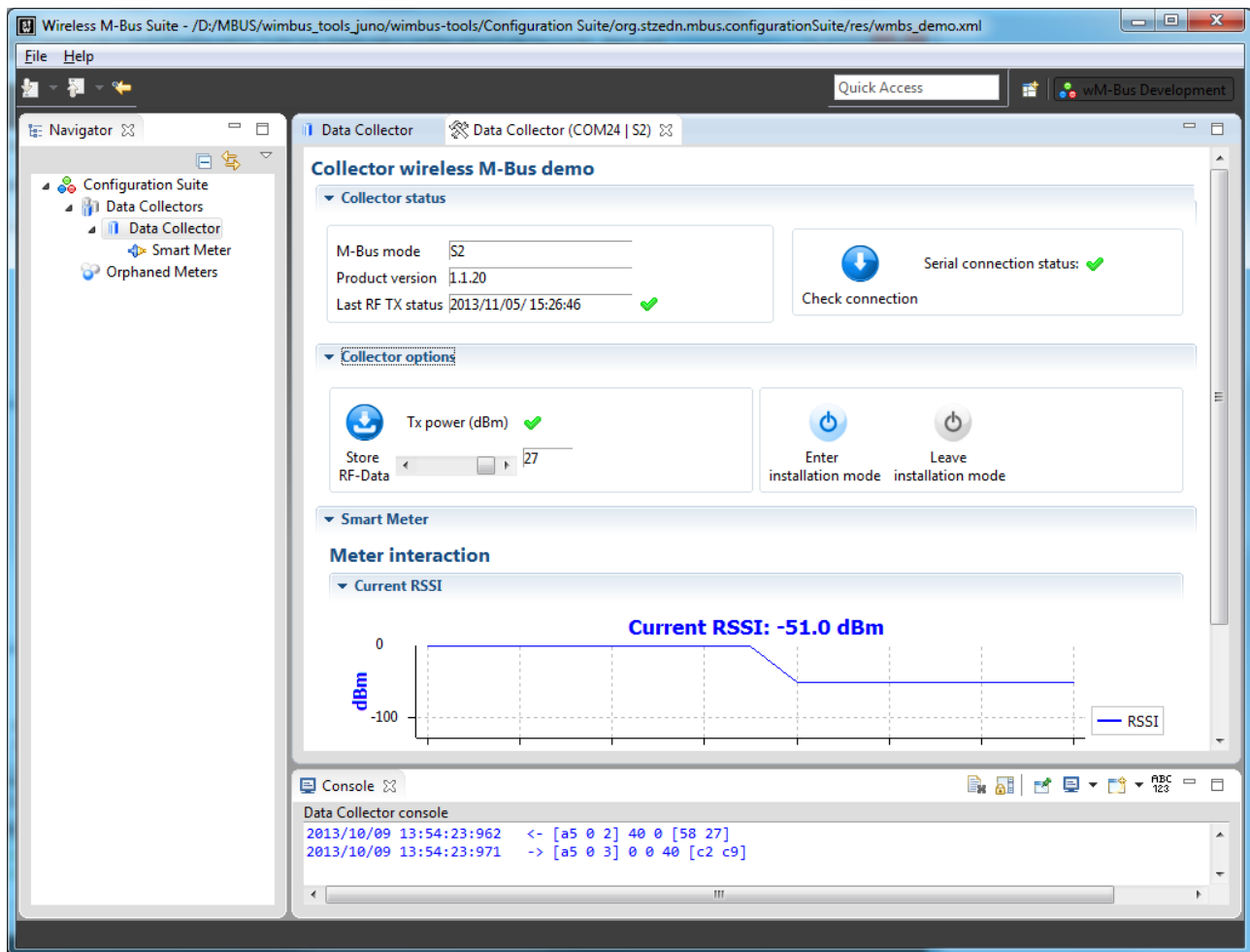
Figure 48: 'Data Collector' Window

You can also send a string from the 'Smart Meter' meter to the 'Data Collector' collector. Input a message in the text field and press 'Send data'. The 'Data Collector'will receive and display this message as shown in figure 49.
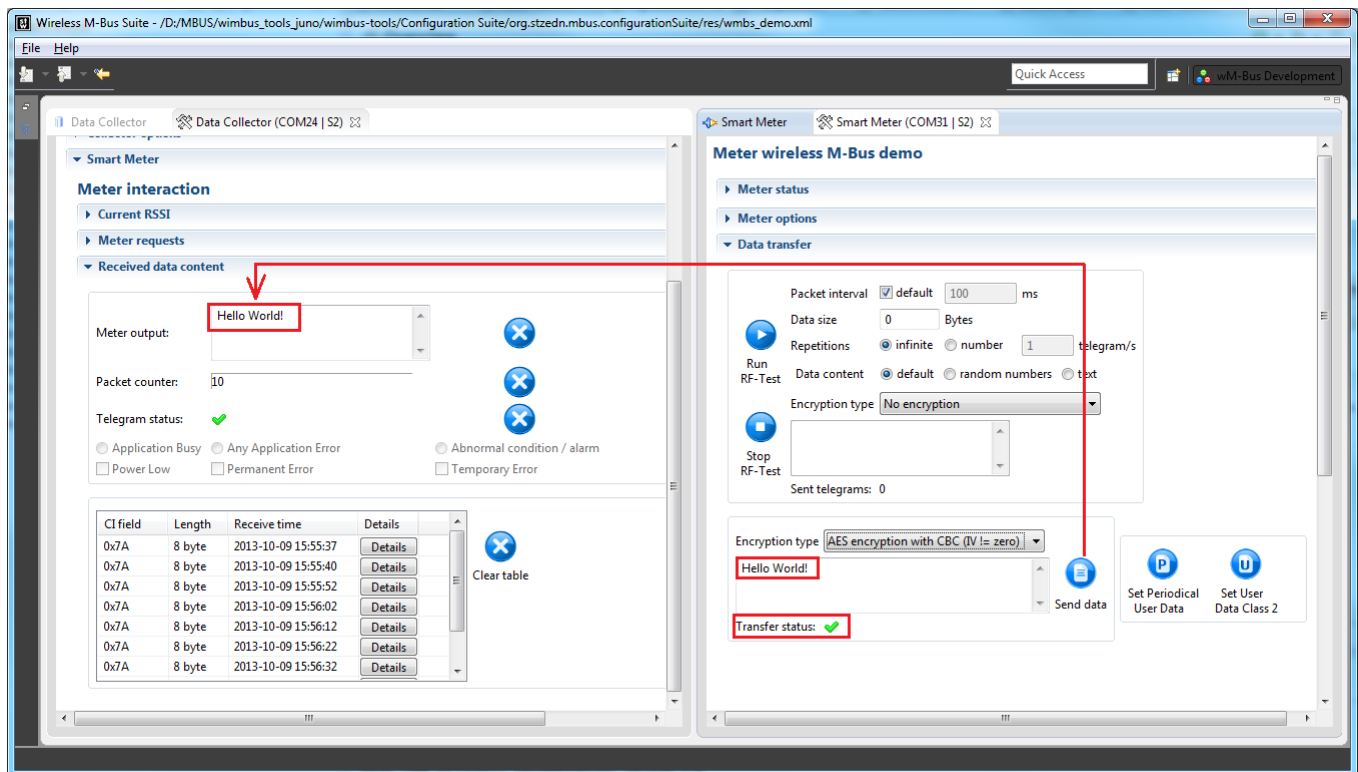
Figure 49: Send data test

Another test which you can perform is the 'RF-Test'. Configure the transmission delay and press 'Run RF-Test' on the 'Smart Meter'. The 'Smart Meter' will send a standard message to the 'Data Collector' periodically. You can now use the Wireless M-Bus Suite to view the 'Smart Meter' which is sending data, and the 'Data Collector', which is receiving data. This demonstration Wireless M-Bus network represents a star-network in an area with different device roles. Generally, the collector device is not intended to send custom data to a meter device.

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus wireless

5.9   Wireless M-Bus Protocol Monitor Quick Start Guide

The wireless M-Bus protocol monitor is a application to monitor wireless networks. It provides a full-featured web based capture monitor system. This document covers instructions for the use of wireless M-Bus protocol monitor. Please make sure to read the instructions carefully. If open questions remain after reading this manual, please contact STZEDN for support.

Requirements:

(a) The Wireless M-Bus device which contains the protocol monitor firmware.

(b) A valid connection of the Wireless M-Bus device to the PC.

(c) Wireless M-Bus Suite.

Perform the following steps to start the wireless M-Bus protocol monitor:

(1) Open Wireless M-Bus Suite.

(2) Select 'Connect to protocol monitor' or 'Protocol Monitor' (refere to figure 50).
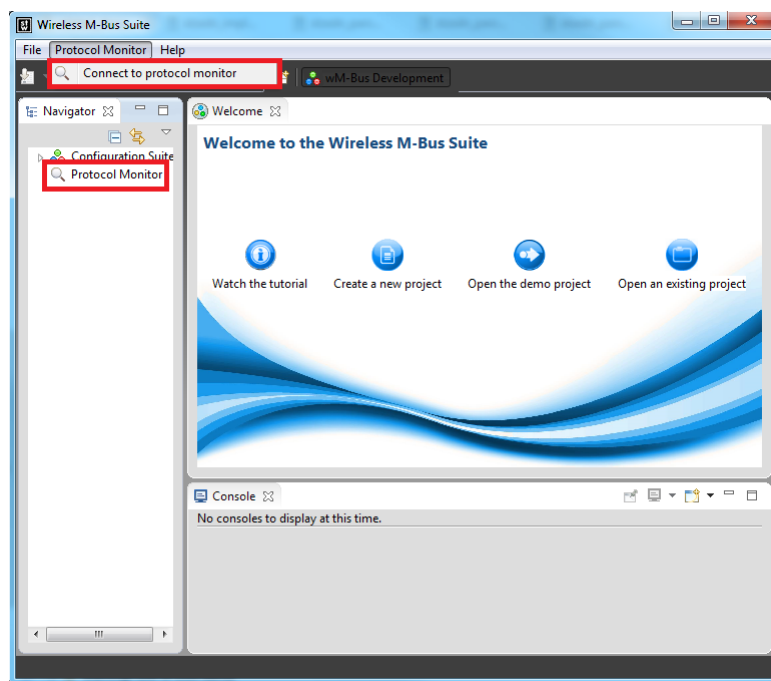
Figure 50: Open the 'Wireless M-Bus Protocol Monitor' wizard

(3) Choose the serial connection and press 'Next'. (See figure 51.)
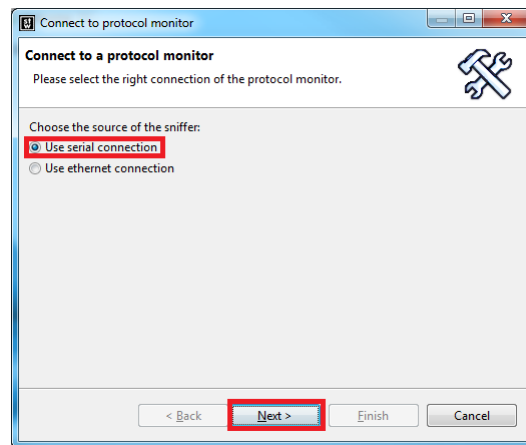
Figure 51: Select the serial connection

(4) Select the protocol monitor device and press 'Next' as shown in figure 52.
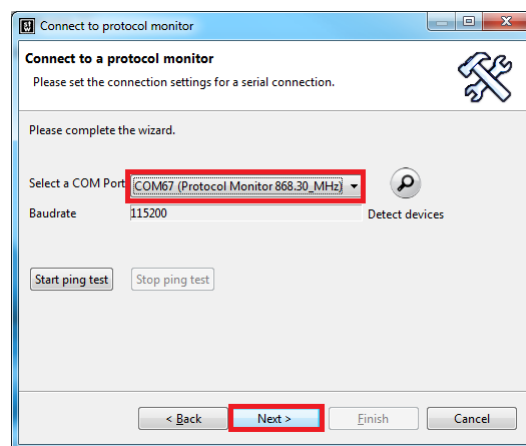
Figure 52: Select the protocol monitor device

(5) Select the mode and press 'Finish' as shown in figure 53.
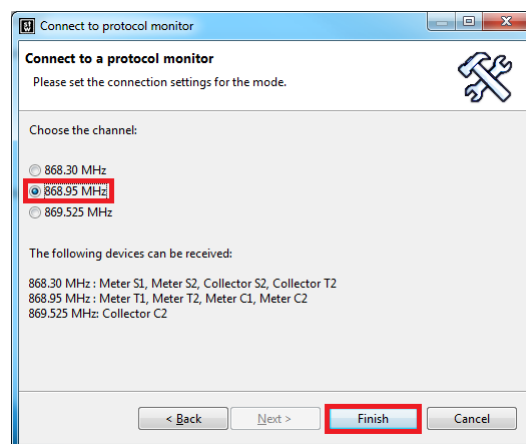
Figure 53: Select the protocol monitor device

(6) Now the wireless M-Bus protocol monitor window will be opend.

(7) To start capturing packets click on the green arrow at the top of the screen as shown in figure 54. Now all packets are captured from the protocol monitor device. If there are already some packets in the device buffer they are read, too.
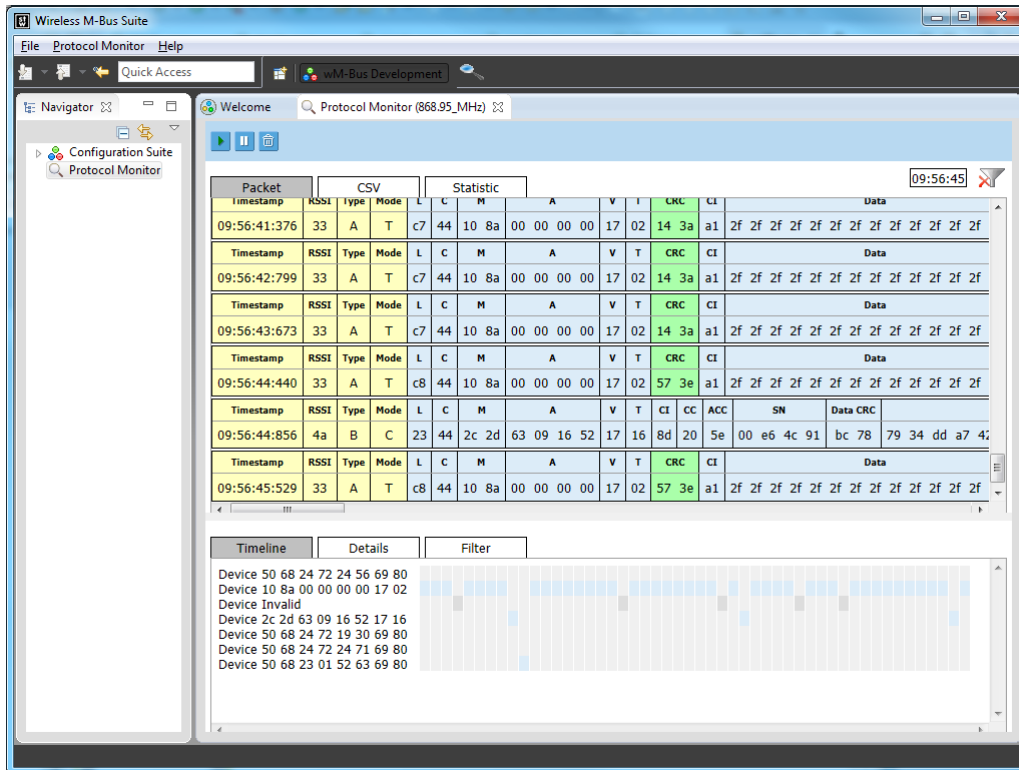
Figure 54: Wireless M-Bus Protocol Monitor

For more informations read chapter 6.

6   Wireless M-Bus Protocol Monitor

6.1   Introduction

This document covers instructions for the use of wireless M-Bus protocol monitor. The first chapter contains the specifications of the wireless M-Bus protocol monitor. A quick start guide follows, that describes in short terms how to get the protocol monitor into operation for the first time. In the third part, the web interface, being the main user interface, is described. Especially focussing on the different views of captured data.

6.2   Wireless M-BUS protocol

A Wireless M-Bus packet consists of several blocks. The first block contains 12 bytes, the following blocks may contain up to 16 bytes. For each block, a 16 bit wide checksum of the current block is transmitted (2 bytes).

Name	Size (Byte)	Description
Length (L)	1	Number of packet bytes excluding the length field and all CRCs.
Type (C)	1	Packet type.
Manufacturer Id (M)	2	ID of the manufacturer, according to DLMS.
Type (A)	6	Address of the meter device.
Checksum (CRC)	2	Checksum of the current block.
Application Type (CI)	1	Type field of the application layer.
(Application Layer)	x	Data of the application layer

Table 20: List of Wireles M-Bus fields.
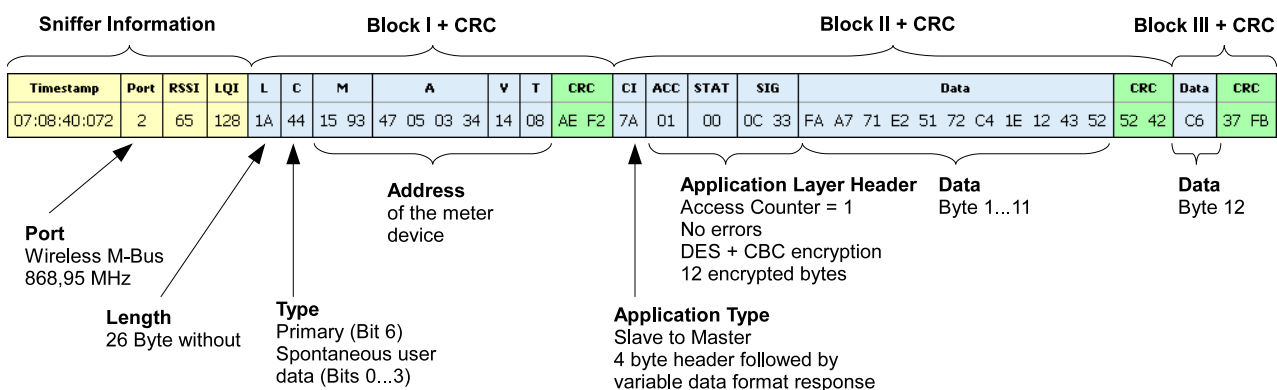
Figure 55: Example of a Wireless M-Bus packet.

This document assumes that the reader is familiar with the Wireless M-Bus specification, according to EN 13757 [9] and its application layer extensions of Open Metering System (OMS) [10] and the Dutch Smart Meter Requirements (DSMR). Otherwise, please contact STZEDN [11] for further documentation or training.

6.3   Running the protocol monitor
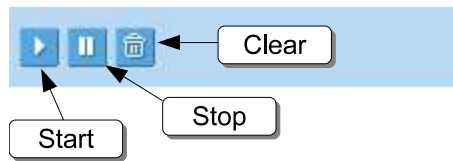
Figure 56: Starting and stopping the wireless M-Bus protocol monitor

To start capturing packets click on the start button of the button bar at the top of the screen. Now all packets from the protocol monitor device are captured. If there are already some packets in the protocol monitor buffer they are read, too.

To stop sniffing click on the stop button. This only stops receiving the packets of the browser application, the protocol monitor device continues to capture all packets. This allows to run several instances of the protocol monitor application on several hosts.

To start sniffing again click on the start button.

To clear the screen click on the clear button. This action removes all cached packets of the browser application.

6.4   Packet View

Figure 57: Packet View.

In the packet view all received packets are shown from top to bottom as they are received over the air. New packets are appended at the bottom of the list, while the list scrolls up automatically over the time.

Steinbeis Transfer Center
Embedded Design and Networking
M-Bus
wireless

One line represents one packet.

Different blocks delimit the various fields of the protocol such as header and data sections.

The first four fields, however, contain data created by the protocol monitor:

- Timestamp: This value contains the time the telegram was received by the protocol monitor device. The timestamp is calculated relatively to the current time of the web browser. A relative time stamp is being received by the protocol monitor device.

- RSSI and LQI: These values describe the link quality of the received telegram. Read chapter 6.8 to get further information.

Protocol sections not active for the selected packet are hidden.

Each field of the packet has two rows:

- The name of the protocol section.

- The data of the protocol section in hexadecimal notation.

6.5   CSV View
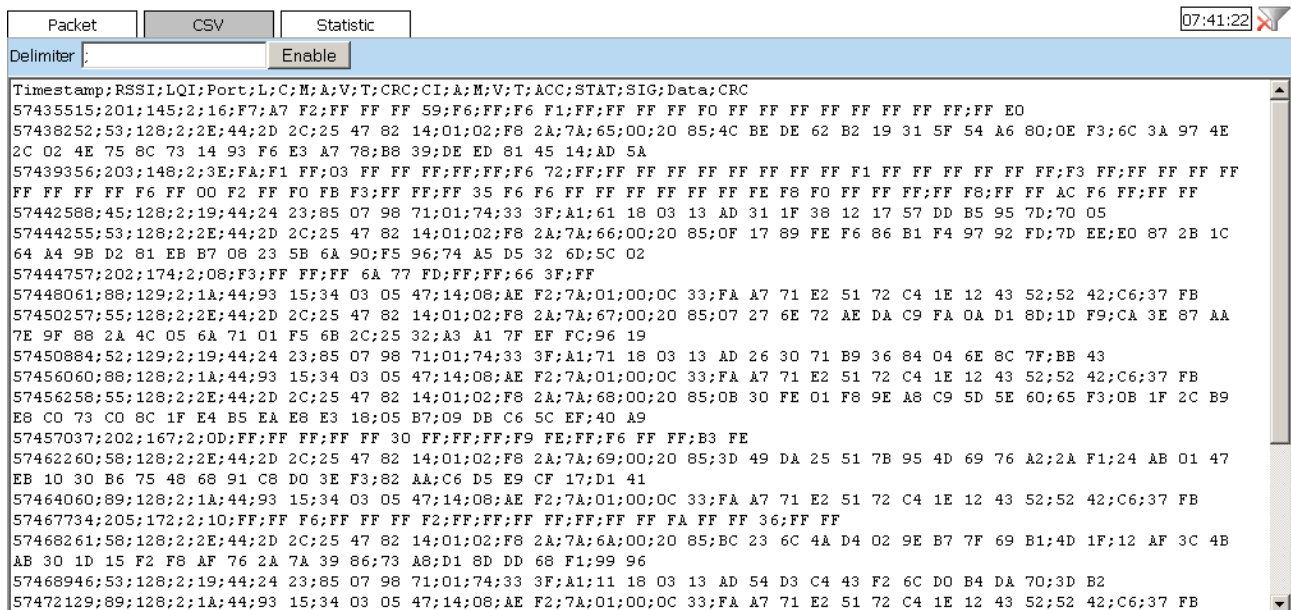
Figure 58: CSV View.

The delimiter character, which separates the fields is set by default to a semicolon ';'. However, you can enter a different character the input text field "Delimiter". To change this field enter your desired delimiter and press set. The delimiters are changed immediately.

6.6   Statistics View
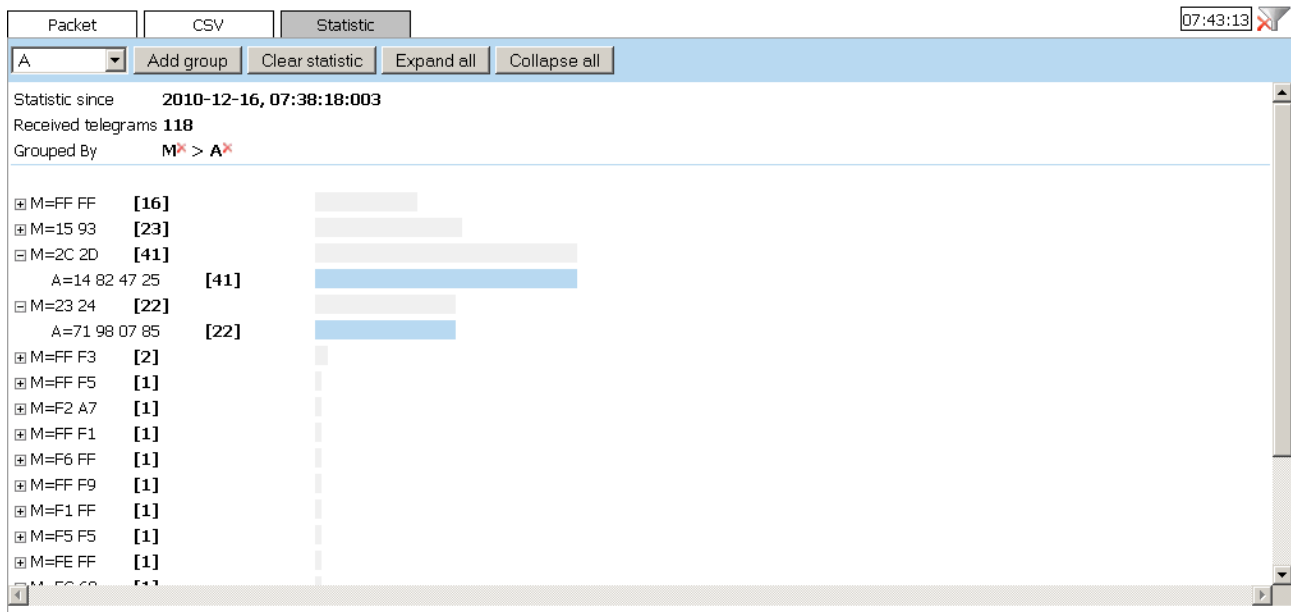
Figure 59: Statistic View.

The statistics view is used to group packets. Packets with the same field values can be set up as statistic to get the overall count of the number of packets.

To create a group select the field of the drop down list in the header row and click on "Add group". Now all packets are grouped by this field. To add further groups repeat this step.

To remove a grouping field, click on the red x behind the field value of "Grouped by" line.

The grouped packets are shown as tree view. If several groups are chosen, click on the [+] in front of the group name to display the subgroups of the current group. For faster opening or closing all groups click on the button "Expand all" or "Collapse all".

The statistics view works independently of the other views. If all packets of the views cleared by pressing the clear button, the packets in the statistic are kept. Click on the button "Clear statistic" to clear the packets from the statistic view.

6.7   Timeline View
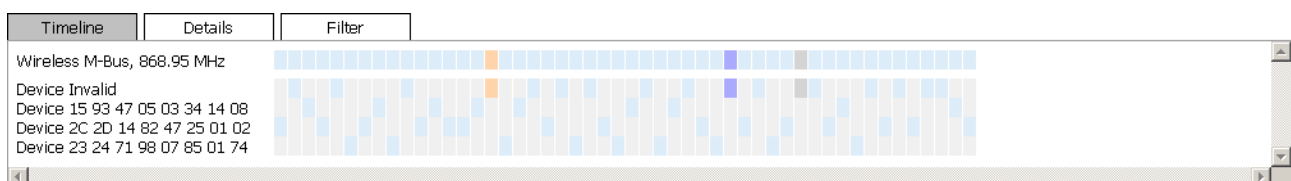
Figure 60: Timeline View.

Additionally to the main view, a second section provides additional information about the radio activity. In the timeline view, all received packets are displayed by ports and addresses.

In the first lines the activities on the ports are shown. The ports may represent different radio settings for the internal modules. Each coloured box marks an activity on the port.

The next lines show the received packets categorised by their address. Selecting a packet in the timeline automatically selects it in the main window as well and vice versa.

6.8   Details View

Figure 61: Detail View.

The details view expands the protocol sections with further details as listed in the table or packet views. Additional protocol specific information are listed here for the currently selected packet.

To display the details of a specific packet in the detail view, select any packet in either the packet view (chapter 6.4) or the time line view (chapter 6.7). The same packet will be selected across all views and its extended information in the detail view can be inspected.

Moving the mouse over a specific packet highlights the packet in the different views without selection. For selection and packet detail investigation, left-clicking is necessary.

6.9 Filter View

The wireless M-Bus protocol monitor allows to define filter rules to concentrate on packets of interest. E.g., it is possible to hide all packets of a special device.

For defining the filter rules open the tab "Filter".

By default, one dummy filter is displayed. For each filter, the following options can be set:

- The field, to which the rule applies.

- The logic operation the value is compared to (==, !=, >, <, >=, <=).

- The value, which has to be used for comparison.
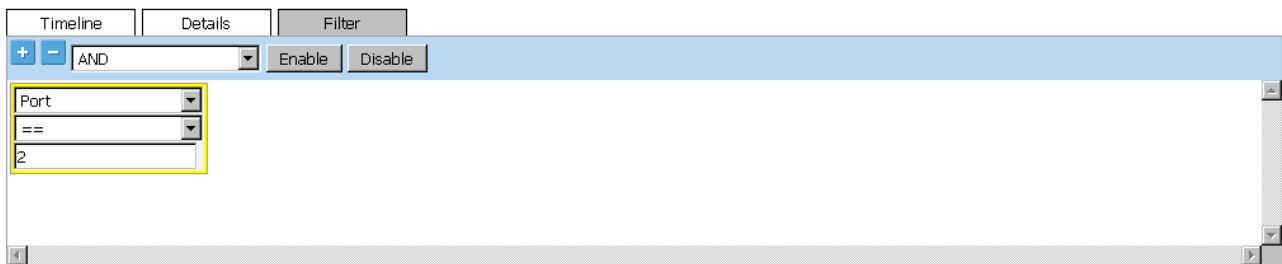
Figure 62: Defining a filter.

To set a filter press the button "Enable". The icon at the top of the screen is then switched to green.

The rules are applied to all views and all received packets. If new packets are detected, they are stored to the local packet memory, but they will be only displayed, if they meet the filter rules. Consequently if the filter is disabled, older packets, which were filtered out, appear then.

If several rules shall be defined, press "+" at the top of the filter tab. A further box to specifiy rules is opened. To combine multiple rules, select the desired operation next to the set-button:

- AND: To display a packet, all rules have to be true.

- OR: At least one rule has to be true to display the packet.
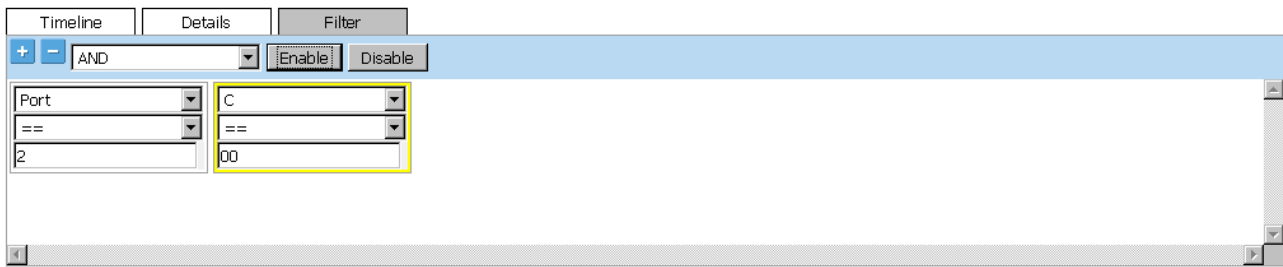
Figure 63: Defining several rules.

To remove a rule, select it by left-clicking and press the remove button "-".

After a rule is changed, the button "Enable" has to be pressed to update the views.

For disabling all filters, click "Disable". The icon at the top of the screen is then switched to red.

6.10   No warranty

The wireless M-Bus protocol monitor is provided "as is" and use of it is at user's risk. STZEDN expressly disclaims all warranties, representations of conditions of any kind, express or implied or statutory, including but not limited to, any warranty of merchantable quality or fitness for a particular purpose and any warranty that the protocol monitor is delivered free of any third party claims by way of infringement or otherwise. STZEDN does not warrant nor make any condition or representations that the functions contained in the protocol monitor will meet the requirements of the customer, that the operation of the protocol monitor will be uninterrupted nor error free that defects in the protocol monitor will be corrected. Further, STZEDN does not warrant or make any conditions or representations regarding the use or the results of the use of the protocol monitor in terms of its correctness, accuracy, reliability or otherwise. No oral or written information or advice given by STZEDN or others shall create a warranty, condition or representation or in any way extend the scope of the "as is" provision of the protocol monitor.

7   Demonstration Application

7.1   Introduction

The following chapters specify the nomenclature and the usage of the different Wireless M-Bus Stack resources. Chapter 7.3 is a description of the demonstration application in general. Furthermore, chapters 7.4 to 7.6 show how the demonstration application can be downloaded to the EFM Starter Kit to run the application. Please note, that in this chaper 'device' and 'board' are used as synonyms for the EFM Starter Kit.

The Wireless M-Bus Stack from STZEDN is delivered in different assamblies:

- Pre-compiled binary files.

- Pre-compiled library sources which can be used with the IAR Workbench.

- Source code IAR Workbench project.

7.2   The Wireless M-Bus Stack Nomenclature

This chapter describes the nomenclature of the Wireless M-Bus Stack resources from STZEDN.

7.2.1   Pre-compiled Firmware Files

7.2.1.1   Wireless M-Bus Stack Firmware Files

In general, the Wireless M-Bus Stack firmwares **without** a serial interface and **without** Wireless M-Bus Suite are made up of

- the prefix 'demo',

- the name of the development board,

- the corresponding transceiver,

- the device type,

- and the Wireless M-Bus mode.

Feel free to change the names of the firmware files. These firmware files are generated by the project in chapter 7.2.3.2.

Name	Transceiver	Type	Mode	File Name
EFM32LG990	SI446X	Collector	c2	*demo_efm32lg990_si446x_collector_c2.bin*
EFM32LG990	SI446X	Collector	n2	*demo_efm32lg990_si446x_collector_n2.bin*
EFM32LG990	SI446X	Collector	s2	*demo_efm32lg990_si446x_collector_s2.bin*
EFM32LG990	SI446X	Collector	t2	*demo_efm32lg990_si446x_collector_t2.bin*
EFM32LG990	SI446X	Meter	c2	*demo_efm32lg990_si446x_meter_c2.bin*
EFM32LG990	SI446X	Meter	n2	*demo_efm32lg990_si446x_meter_n2.bin*
EFM32LG990	SI446X	Meter	s2	*demo_efm32lg990_si446x_meter_s2.bin*
EFM32LG990	SI446X	Meter	t2	*demo_efm32lg990_si446x_meter_t2.bin*

Table 21: Nomenclature of the Wireless M-Bus Stack firmware files **without** a serial interface and **without** Wireless M-Bus Suite.

In general, the Wireless M-Bus Stack firmwares **with** a serial interface and **with** Wireless M-Bus Suite are made up of

- the prefix 'serial',

- the name of the development board,

- the corresponding transceiver,

- the device type,

- and the Wireless M-Bus mode.

Feel free to change the names of the firmware files. These firmware files are generated by the project in chapter 7.2.3.4. Please note, that all serial Wireless M-Bus Stack firmwares use 115200 baud.

Name	Transceiver	Type	Mode	File Name
EFM32LG990	SI446X	Collector	c2	*serial_efm32lg990_si446x_collector_c2.bin*
EFM32LG990	SI446X	Collector	n2	*serial_efm32lg990_si446x_collector_n2.bin*
EFM32LG990	SI446X	Collector	s2	*serial_efm32lg990_si446x_collector_s2.bin*
EFM32LG990	SI446X	Collector	t2	*serial_efm32lg990_si446x_collector_t2.bin*
EFM32LG990	SI446X	Meter	c2	*serial_efm32lg990_si446x_meter_c2.bin*
EFM32LG990	SI446X	Meter	n2	*serial_efm32lg990_si446x_meter_n2.bin*
EFM32LG990	SI446X	Meter	s2	*serial_efm32lg990_si446x_meter_s2.bin*
EFM32LG990	SI446X	Meter	t2	*serial_efm32lg990_si446x_meter_t2.bin*

Table 22: Nomenclature of the Wireless M-Bus Stack firmware files **with** a serial interface and **with** Wireless M-Bus Suite.

7.2.2   Wireless M-Bus Stack Libraries

In general, the library file names are made up of

- the name of the development board,

- the corresponding transceiver,

- the device type,

- and the Wireless M-Bus mode.

Name	Transceiver	Type	Mode	Type
EFM32LG990	SI446X	Collector	c2	*EFM32LG990_SI446X_Collector_C2.a*
EFM32LG990	SI446X	Collector	n2	*EFM32LG990_SI446X_Collector_N2.a*
EFM32LG990	SI446X	Collector	s2	*EFM32LG990_SI446X_Collector_S2.a*
EFM32LG990	SI446X	Collector	t2	*EFM32LG990_SI446X_Collector_T2.a*
EFM32LG990	SI446X	Meter	c2	*EFM32LG990_SI446X_Meter_C2.a*
EFM32LG990	SI446X	Meter	n2	*EFM32LG990_SI446X_Meter_N2.a*
EFM32LG990	SI446X	Meter	s2	*EFM32LG990_SI446X_Meter_S2.a*
EFM32LG990	SI446X	Meter	t2	*EFM32LG990_SI446X_Meter_T2.a*

Table 23: Nomenclature of the Wireless M-Bus Stack library files

7.2.3   IAR Projects and Configurations

It is possible to open all IAR projects with the project file:

Project Name: *Complete_efm32lg990_si446x.eww*

7.2.3.1   Library Build IAR Workbench Project

The library build project produces the Wireless M-Bus Stack libraries for the different Wireless M-Bus network devices and the different Wireless M-Bus modes. The project contains all configurations with the type and mode corresponding settings listed in table 24.

Project Name: *Lib_Build_efm32lg990_si446x.ewp*

Type	Mode	Configuration Name
Collector	c2	Lib_Build_EFM32LG990_SI446X_Collector_C2
Collector	n2	Lib_Build_EFM32LG990_SI446X_Collector_N2
Collector	s2	Lib_Build_EFM32LG990_SI446X_Collector_S2
Collector	t2	Lib_Build_EFM32LG990_SI446X_Collector_T2
Meter	c2	Lib_Build_EFM32LG990_SI446X_Meter_C2
Meter	n2	Lib_Build_EFM32LG990_SI446X_Meter_N2
Meter	s2	Lib_Build_EFM32LG990_SI446X_Meter_S2
Meter	t2	Lib_Build_EFM32LG990_SI446X_Meter_T2

Table 24: Configuration Names of the Library Build porject

7.2.3.2   Library Usage IAR Workbench Project

The library usage project includes the Wireless M-Bus Stack libraries. The customer can use this project to program own applications based on the Wireless M-Bus Stack libraries. A sample main file for the meter and the collector device is included in the project to quickly setup a wireless communication. The project contains all configurations with the type and mode corresponding settings listed in table 25.

Project Name: *Lib_Use_efm32lg990_si446x.ewp*

Collector Main File: *app_apl\main_collector.c*

Meter Main File: *app_apl\main_meter.c*

Configuration File: *app_apl\stzedn_wmbus_global.h*

Type	Mode	Configuration Name
Collector	c2	Lib_Use_EFM32LG990_SI446X_Collector_C2
Collector	n2	Lib_Use_EFM32LG990_SI446X_Collector_N2
Collector	s2	Lib_Use_EFM32LG990_SI446X_Collector_S2
Collector	t2	Lib_Use_EFM32LG990_SI446X_Collector_T2
Meter	c2	Lib_Use_EFM32LG990_SI446X_Meter_C2
Meter	n2	Lib_Use_EFM32LG990_SI446X_Meter_N2
Meter	s2	Lib_Use_EFM32LG990_SI446X_Meter_S2
Meter	t2	Lib_Use_EFM32LG990_SI446X_Meter_T2

Table 25: Configuration Names of the Library Usage porject

7.2.3.3   Source Code IAR Workbench Project with serial interface disabled

The source code project contains all source files from the Wireless M-Bus Stack. A sample main file for the meter and the collector device is included in the project to quickly setup a wireless communication. The project contains all configurations with the type and mode corresponding settings listed in table 26.

Project Name: *Debug_Demo_efm32lg990_si446x.ewp*

Collector Main File: *app_apl\main_collector.c*

Meter Main File: *app_apl\main_meter.c*

Configuration File: *app_apl\stzedn_wmbus_global.h*

Type	Mode	Configuration Name
Collector	c2	Debug_Demo_EFM32LG990_SI446X_Collector_C2
Collector	n2	Debug_Demo_EFM32LG990_SI446X_Collector_N2
Collector	s2	Debug_Demo_EFM32LG990_SI446X_Collector_S2
Collector	t2	Debug_Demo_EFM32LG990_SI446X_Collector_T2
Meter	c2	Debug_Demo_EFM32LG990_SI446X_Meter_C2
Meter	n2	Debug_Demo_EFM32LG990_SI446X_Meter_N2
Meter	s2	Debug_Demo_EFM32LG990_SI446X_Meter_S2
Meter	t2	Debug_Demo_EFM32LG990_SI446X_Meter_T2

Table 26: Configuration Names of the Debug porject with serial interface disabled

7.2.3.4   Source Code IAR Workbench Project with serial interface enabled

The source code project contains all source files from the Wireless M-Bus Stack. A sample main file for the meter and the collector device is included in the project to quickly setup a wireless communication. The project contains all configurations with the type and mode corresponding settings listed in table 27. Please note, that there is one shared *main* file for the collector and meter.

Project Name: *Debug_Serial_efm32lg990_si446x.ewp*

Collector Main File: *app_serial\stzedn_app_serial.c*

Meter Main File: *app_serial\stzedn_app_serial.c*

Configuration File: *app_serial\stzedn_wmbus_global.h*

Type	Mode	Configuration Name
Collector	c2	Debug_Serial_EFM32LG990_SI446X_Collector_C2
Collector	n2	Debug_Serial_EFM32LG990_SI446X_Collector_N2
Collector	s2	Debug_Serial_EFM32LG990_SI446X_Collector_S2
Collector	t2	Debug_Serial_EFM32LG990_SI446X_Collector_T2
Meter	c2	Debug_Serial_EFM32LG990_SI446X_Meter_C2
Meter	n2	Debug_Serial_EFM32LG990_SI446X_Meter_N2
Meter	s2	Debug_Serial_EFM32LG990_SI446X_Meter_S2
Meter	t2	Debug_Serial_EFM32LG990_SI446X_Meter_T2

Table 27: Configuration Names of the Debug porject with serial interface enabled

Please note, that all serial Wireless M-Bus Stack configurations use 115200 baud.

7.3   The Demonstration Application

7.3.1   Introduction

The demonstration application builds up a wireless network with two network devices - a collector and a meter device. Thus, the application needs two EFM Starter Kits. Figure 64 shows the basic setup.

The first step is to download the collector and the meter firmware to the EFM Starter Kits. Refer to chapters 7.4 to 7.6 to get detailed information about how to download a firmware to the boards.
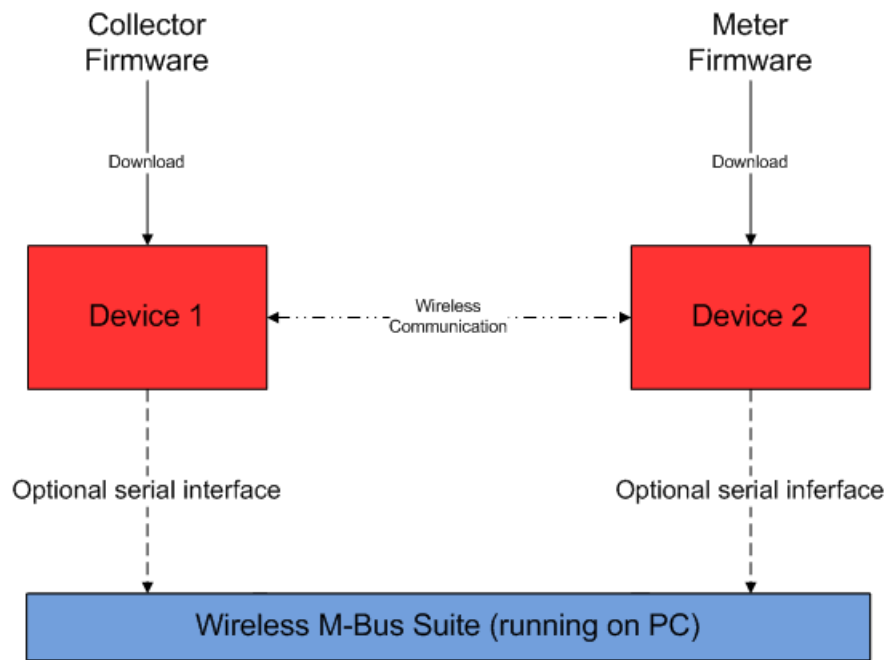
Figure 64: Demonstration Application

There are two different setups to run the demonstration application. One setup runs without the Wireless M-Bus Suite (refer to chapter 7.3.2) and one setup with it (refer to chapter 7.3.3) [1].

> ⚠ *The Wireless M-Bus Stack firmware differs for the two setups. Please make sure to select the correct firmware files. Refer to chapter 7.2 for details about the nomenclature of the firmware files or IAR projects.*

7.3.2 Demonstration Application without Wireless M-Bus Suite

In this setup, the demonstration application works without the optional serial interface and thus, without Wireless M-Bus Suite. The following Wireless M-Bus Stack firmware files or projects are applicable:

- The pre-compiled Wireless M-Bus Stack firmware files in chapter 7.2.1, table 21. Refer to chapter 4.2.1 for details about downloading a binary firmware file to a EFM Starter Kit.

- The IAR Workbench project in chapter 7.2.3.2 or the project in chapter 7.2.3.3. Refer to chapter 4.2.2 for details about downloading an IAR Workbench Project to a EFM Starter Kit.

If the demonstration application is installed on both boards, the meter device starts sending sample data packets to the collector device.

---

[1]It is recommended to use the USB interface as power supply. If the EFM Starter Kits are powered with batteries, the display may not work correctly.

7.3.3   Demonstration Application with Wireless M-Bus Suite

The Wireless M-Bus Suite (refer to chapter 5) can be used to control the demonstration application on both boards via the serial interface. In the Wireless M-Bus demonstration application without Wireless M-Bus Suite, the meter device sends a telegram to the collector device in a defined time interval. Please note, that the serial demonstration application is limited in the number of supported meter devices. There are many advantages and extensions for the demonstration application, which allow fast and easy wireless communications, if Wireless M-Bus Suite is used:

- Perform an RF-Test.

- Set serveral user telegrams and send them from the meter to the collector device.

- Perform a bidirectional communication.

- Communication with and without encryption.

- Access to internal parameters of the devices. E.g. in case of a collector device the 'Received Signal Strength Indication' value is displayed for a specific meter.

- Access to the serial console. Thus, all commands from the serial specification from STZEDN can be send to the devices.

- Create fast and easily own Wireless M-Bus configurations.

- Wireless M-Bus Suite supports Wireless M-Bus Stack sniffer (refer to chapters 5.9 and 6).

- Wireless M-Bus Suite includes an AES encryption and decryption tool.

- Download binary firmware files to the EFM Starter Kits, to switch or update applications, if a bootloader is available.

The following Wireless M-Bus Stack firmware files or projects are applicable:

- The pre-compiled Wireless M-Bus Stack firmware files in chapter 7.2.1, table 22. Refer to chapter 4.2.1 for details about downloading a binary firmware file to a EFM Starter Kit.

- The IAR Workbench project in chapter 4.2.1. Refer to chapter 4.2.2 for details about downloading an IAR Workbench Project to a EFM Starter Kit.

7.4   Download Binary Files

This chapter describes how to prepare the the demonstration application of the Wireless M-Bus Stack from STZEDN with a pre-compiled firmware, delivered as a '*.bin' binary file. To start the demonstration application, the Wireless M-Bus Stack firmware must be downloaded to the EFM Starter Kits. The firmware files in chapter 7.2.1, table 21 and table 22, are applicable.

To upload the firmware, please perform the steps in chapter 4.2.1.

Quick Start Guide
Version 1.02

Steinbeis Transfer Center
Embedded Design and Networking

M-Bus
wireless

Please refer to chapter 7.3 for a description of the demonstration application on the EFM Starter Kits.

7.5   The Wireless M-Bus Stack as Library

This chapter describes the usage of the Wireless M-Bus Stack as library. Regarding the libraries, there are two IAR Workbench projects available. The project in chapter 7.2.3.1 builds the Wireless M-Bus Stack libraries. The project in chapter 7.2.3.2 uses the libraries to setup an application project. If you want to build the libraries, go to chapter 7.5.1. If the libraries should be built, please refer to chapter 7.5.2.

Please refer to chapter 7.3 for a description of the demonstration application on the EFM Starter Kit. Refer to chapter 4.2.2 for more detailed information about uploading an IAR Workbench project to the EFM Starter Kits.

7.5.1   Build Libraries

The corresponding IAR Workbench project is the 'lib_build' project in chapter 7.2.3.1. The purpose of the project is only to build Wireless M-Bus Stack libraries. Thus, there are no main files necessary. The project contains all preconfigured configuration for the different Wireless M-Bus types and modes. Refer to table 23 in chapter 7.2.2 for detailed information about the nomenclature of the library files. Please note, that the serial interface is disabled in all project configurations.

7.5.2   Use Libraries

The corresponding IAR Workbench project is the 'lib_use' project in chapter 7.2.3.2. The purpose of this project is to use the Wireless M-Bus Stack libraries to develop applications. For a fast and easy development, the project contains sample main files for the meter and the collector device which implement the demonstration application. The user can change and extend these files for own applications. The project contains preconfigured configurations for all supported Wireless M-Bus types and modes. Please make sure to include the correct library file for the intended Wireless M-Bus type and mode. Please note that the serial interface is disabled in all project configurations.

7.6   The Wireless M-Bus Stack as Source Code

The corresponding IAR Workbench projects are described in chapter 7.2.3.3 and chapter 7.2.3.4. In the first project, the serial interface is enabled. In the second project, the serial interface is disabled. However, these projects support full access to every source file at each communication layer of the Wireless M-Bus Stack from STZEDN. Thus, the user is able to change e.g. the availibility of the serial interface.

Please note, that changes in the source code can have major impact on the principle of operation of the Wireless M-Bus Stack. Please perform changes on the source code only, if you are completly confident about the impacts. There is no support from STZEDN to projects with changed source code (The only exception to this restriction are the main application files and the configuration header file, **stzedn_wmbus_global.h***).*

Please make sure, that the following requirements are fulfilled:

Number	Description
1	The function *stzedn_tmr_tick()* must be called 2000 times per second.
2	The microcontroller main clock must be fast enough to ensured that the timings in the bidirectional communication can be hold.

Table 28: Wireless M-Bus Stack Requirements

Please refer to chapter 7.3 for a description of the demonstration application on the EFM Starter Kit. Refer to chapter 4.2.2 to get detailed information about uploading an IAR Workbench project to the EFM Starter Kits.

8   Contact information

In case of questions, requests for quotations or ideas for further improvements, please contact:

Steinbeis Transfer Center for Embedded Design and Networking (STZEDN)

Poststrasse 35

D-79423 Heitersheim

Germany

Tel: +49 7634-6949-340

Fax: +49 7634-5049-886

Url: http://www.stzedn.de/smart-metering.html

E-mail: metering@stzedn.de

References

[1] "Communication systems for meters and remote reading of meters." Part 1: Data exchange; English version EN 13757-1, 2002.

[2] "Communication systems for meters and remote reading of meters." Part 2: Physical and link layer; English version EN 137572, 2004.

[3] "Communication systems for meters and remote reading of meters." Part 3: Dedicated application layer; English version EN 137573, 2004.

[4] "Communication systems for meters and remote reading of meters." Part 4: Wireless meter readout (Radio meter reading for operation in the 868 MHz to 870 MHz SRD band); German version EN 137574, 2005.

[5] "Communication systems for meters and remote reading of meters." Part 5: Wireless Relaying; English version prEN 13757-5, 2009.

[6] "Telecontrol equipment and systems." Part 5: Transmission protocols; EN 870-5, 2002.

[7] "Open metering system specification - volume 2: Primary communication." http://www.oms-group.org/download/OMS-Spec_Vol2_Primary_v301.pdf, 2011.

[8] S. Labs, "Simplicity studio homepage, "http://www.energymicro.com/software/simplicity-studio"."

[9] "European committee for standardization." [Online; accessed 15-December-2011] http://www.cen.eu.

[10] "Oms-group." [Online; accessed 15-December-2011] http://www.oms-group.org.

[11] "Steinbeis transfer center for embedded design and networking." [Online; accessed 15-December-2011] http://www.stzedn.de.