Wireless M-Bus Documentation

Quick Start Guide

STACKFORCE

embedded.connectivity.solutions

March 20, 2015

Chapter 1

Scope of this document

This document is a guide for quick start using devices and starter kits by Silicon Labs for evaluating, configuring and starting development of own Wireless M-Bus applications, based on the Wireless M-Bus Stack by STACKFORCE.

This guide contains chapters explaining

- the very basics of Wireless M-Bus (2),
- how to use the provided demo applications (4) and
- how start and use the Wireless M-Bus Suite (3) shipped with this package.

Contents

1 Scope of this document II

Contents III

List of Tables V

List of Figures VI

Acronym 1

2 Wireless M-Bus Basics 2

2.1 Standards . . . 2
2.2 The STACKFORCE Protocol Stack . . . 3
2.3 Topology . . . 4
2.4 Wireless M-Bus (WM-Bus) Modes . . . 4
2.5 Unidirectional and Bidirectional Communication . . . 7

3 Wireless M-Bus Suite 8

3.1 System Requirements . . . 8
3.2 Quickstart . . . 9
3.2.1 Loading the demo project . . . 9
3.2.2 Collector short description . . . 11
3.2.3 Meter short description . . . 13
3.2.4 Using the Wireless M-Bus Suite . . . 15
3.2.4.1 Mode selection . . . 15
3.2.4.2 Console . . . 17
3.2.5 Perform a ping test . . . 18
3.2.6 Using the transmission tests . . . 20
3.3 COM Port Settings . . . 23

4 Demonstration Application 25

4.1 Introduction . . . 25
4.1.1 Top level layers . . . 25
4.1.2 Demo projects . . . 25
4.2 The Demo App . . . 26
4.2.1 Basics . . . 26
4.2.2 Programming . . . 27

4.2.3 Playing with the Wireless M-Bus Suite 28
4.3 The Libraries 28
4.3.1 Wireless M-Bus Stack library 29
4.3.2 RF driver library 29

5 Contact information **30**

Bibliography **30**

List of Tables

2.1 Wireless M-Bus operating modes 5
2.2 Wireless M-Bus mode parameter settings. 5
2.3 Compatibility matrix for Wireless M-Bus modes. 6

3.1 'Smart Meter' Parameters 14
3.2 Console View Commands 18

List of Figures

2.1 Architecture of the Wireless M-Bus Protocol Stack. . . . 3
2.2 Frequencies of Wireless M-Bus modes S, T and C. . . . 6
2.3 Frequencies of Wireless M-Bus mode N. . . . 6
2.4 Bidirectional communication example for WM-Bus modes $S2$, $T2$, $C2$ and $N2(a-f)$. . . . 7

3.1 Welcome screen 10
3.2 Collector and Meter device at the navigation bar 10
3.3 Collector editor 11
3.4 Meter connected 12
3.5 Meter disconnected 12
3.6 Meter without collector address 12
3.7 Meter addresses duplicate 12
3.8 COM Port selection 12
3.9 Select 'Smart Meter' meter 13
3.10 'Data Collector' programmed 13
3.11 'Smart Meter' Window 14
3.12 'Smart Meter' programmed 15
3.13 Mode selection boxes 16
3.14 Collector C and Meter T 16
3.15 Mode Nb example 17
3.16 Console View 17
3.17 Console with serial text output enabled 18
3.18 Write config to device 19
3.19 Start ping test 19
3.20 Ping test failed 20
3.21 Ping test ok 20
3.22 'Start wireless M-Bus demo' button 'Data Collector' 21
3.23 'Start wireless M-Bus demo' button 'Smart Meter' 21
3.24 'Data Collector' Window 22
3.25 Send data test 23
3.26 Standard COM port 24
3.27 Virtual COM port 24

4.1 Demonstration Application 27

Acronym

APL	Application Layer
DLL	Data Link Layer
ELL	Extended Data Link Layer
EN	European Norm
FAC	Frequent Access Cycle
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
HAL	Hardware Abstraction Layer
M-Bus	Meter-Bus
MCU	Microcontroller Unit
NRZ	Non-Return-to-Zero
OMS	Open Metering System
PHY	Physical Layer
RF	Radio frequency, also commonly used as a synonym for the transceiver
RX	Receive
TPL	Transport Layer
TX	Transmit
WM-Bus	Wireless M-Bus

Chapter 2

Wireless M-Bus Basics

2.1 Standards

Meter-Bus (M-Bus) is a field bus specialized for the transmission of metering data from gas, electricity, heat, water or other meters to a data collector. The European Norm (EN) 13757 defines the standard, which includes the specification of wired and wireless M-Bus. The specification contains five parts:

- EN 13757-1 [?]:
  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 [?]:
  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 [?]:
  Part 3: Dedicated application layer
  The third part describes a standardized application protocol to enable multi-vendor capability. So devices of different manufacturers may be combined in one system.
- EN 13757-4 [?]:
  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 the implementation of the WM-Bus stack from STACKFORCE. It includes the Physical Layer (PHY) and the Data Link Layer (DLL) for wireless devices. It corresponds to specification EN 13757-2 for wired communication.
- EN 13757-5 [?]:
  Part 5: Relaying
  This part includes different proposals for relaying data frames to overcome the range problem between remote meters and data collectors.

All parts of EN 13757 are compliant to the EN 870-5 [?].

2.2 The STACKFORCE Protocol Stack

The protocol stack consists of the following major parts:

Wireless M-Bus Stack The stack implements the PHY layer, DLL, Extended Data Link Layer (ELL), Transport Layer (TPL) and the Application Layer (APL).

RF driver The Radio frequency, also commonly used as a synonym for the transceiver (RF) driver configures and controls the connected transceiver. The RF driver somehow is part of the Hardware Abstraction Layer (HAL), as it is abstracting the access to transceiver for the stack, but in parallel the RF driver needs abstracted access to some hardware provided by the Microcontroller Unit (MCU), e.g. the SPI interface and GPIOs.

HAL The HAL is responsible to abstract all the hardware resources required by the stack and the RF driver. E.g. this includes SPI interface, UART interface, access to non-volatile memory, GPIO, …

Figure 2.1 provides an overview over the WM-Bus protocol stack implementation of STACKFORCE.

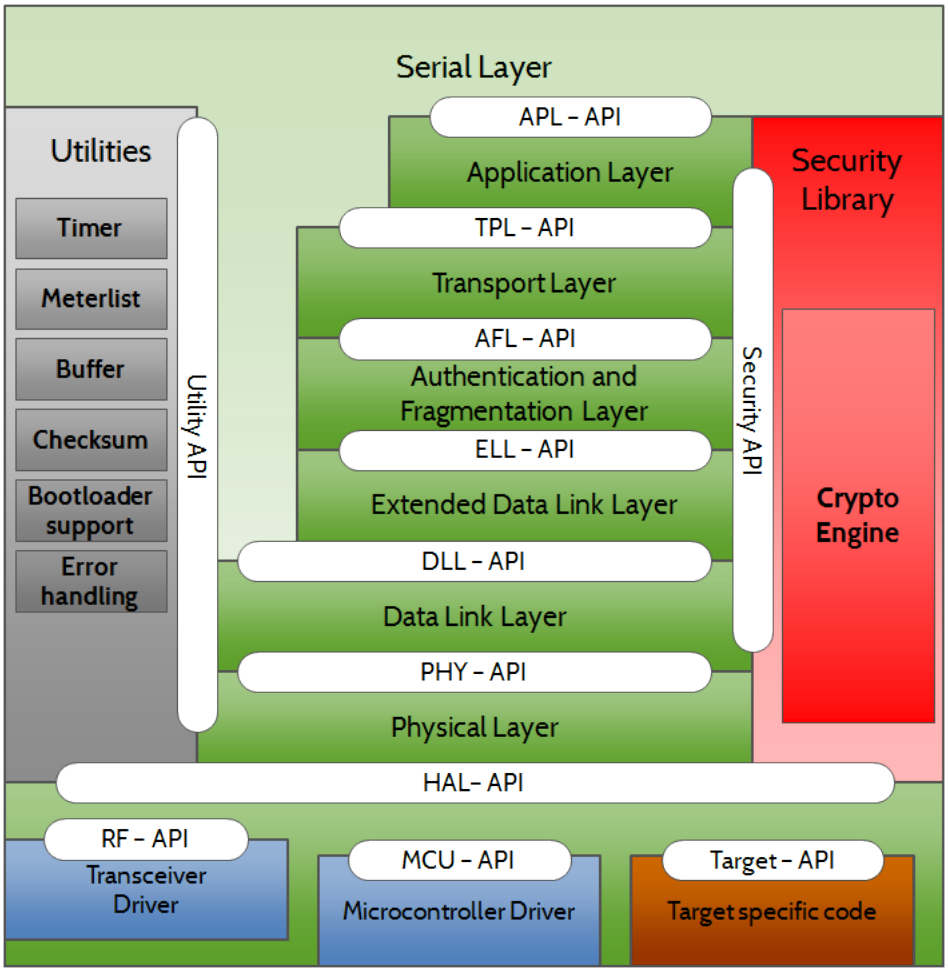

Figure 2.1: Architecture of the Wireless M-Bus Protocol Stack.

Please note, this picture illustrates the complete stack as available and developed by STACKFORCE. The received package may not include all parts that are illustrated. For details on which parts are shipped with this package please contact support at distributor of your package or STACKFORCE.

Please contact the STACKFORCE to obtain information about configurations, extensions and adaptations of the Wireless M-Bus Stack. Chapter 5 contains contact information.

2.3 Topology

In general, WM-Bus supports two types of devices [?]:

- "Meter" device:
  A "Meter" device is a device which collects and forwards information to an "Other" device. E.g., a meter could be an electric meter or a gas meter.
- "Other" device:
  An "Other" device is a system component which receives information from a "Meter" device. This could be "*mobile readout devices, stationary receivers, data collectors, multi-utility concentrators or system network components*" [?]. In the following, this document denotes an "Other" device as data collector.

WM-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 are supported.

2.4 WM-Bus Modes

Wireless M-Bus supports various comminication modes. The Wireless M-Bus modes define the communication flow and the configuration of the radio channel. Table 2.1 lists supported communication modes from the STACKFORCE Wireless M-Bus Stack.

Mode	Description
$S1, S2$	In the **S**tationary 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.
$T1, T2$	In the Frequent **T**ransmit mode, the metering devices periodically send their data to collectors in range. The interval is configurable in terms of several seconds or minutes.
$C1, C2$	**C**ompact 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.
$N1(a-f), N2(a-f)$	**N**arrowband communication mode for long range transmissions.

Table 2.1: Wireless M-Bus operating modes

In general, Wireless M-Bus modes can have different data rates, data encodings (e.g. Non-Return-to-Zero (NRZ) or Manchester), frequency modulations (e.g. Frequency Shift Keying (FSK) or Gaussian Frequency Shift Keying (GFSK)) and carrier frequencies. Table 2.2 lists the related parameters with respect to the Wireless M-Bus mode, the device type and whether a device is in Receive (RX) or Transmit (TX) mode.

Mode	Meter		Collector		Data Rate	Endocing	Modulation	Frequency [MHz]
	RX	TX	RX	TX				
$N1a, N2a$	✓	✓	✓	✓	4.8 kbit/s	NRZ	GFSK	169.406250
$N1b, N2b$	✓	✓	✓	✓	4.8 kbit/s	NRZ	GFSK	169.418750
$N1c, N2c$	✓	✓	✓	✓	2.4 kbit/s	NRZ	GFSK	169.431250
$N1d, N2d$	✓	✓	✓	✓	2.4 kbit/s	NRZ	GFSK	169.443750
$N1e, N2e$	✓	✓	✓	✓	4.8 kbit/s	NRZ	GFSK	169.456250
$N1f, N2f$	✓	✓	✓	✓	4.8 kbit/s	NRZ	GFSK	169.468750
$T2$	✓			✓	32.768 kcps	Manchester	FSK	868.30
$T1, T2$		✓	✓		100 kcps	3-out-of-6	FSK	868.95
$S1, S2$	✓	✓	✓	✓	32.768 kcps	Manchester	FSK	868.30
$C2$	✓			✓	50 kcps	NRZ	GFSK	869.525
$C2$				✓	32.768 kcps	Manchester	FSK	868.30
$C2$			✓		100 kcps	3-out-of-6	FSK	869.95
$C1, C2$		✓	✓		100 kcps	NRZ	FSK	868.95

Table 2.2: Wireless M-Bus mode parameter settings.

Figure 2.2 provides a graphical overview over the TX frequencies in Wireless M-Bus modes S, T, C and N.

Figure 2.3 provides a graphical overview over the TX frequencies in Wireless M-Bus modes $N1(a-f)$ and $N2(a-f)$.

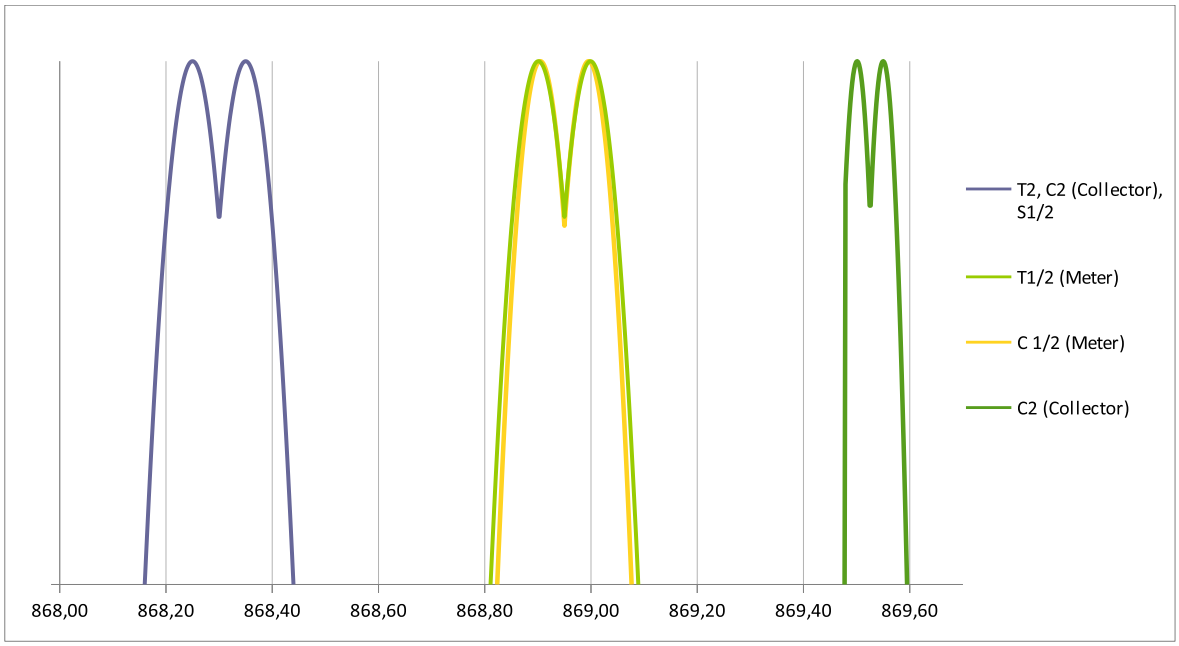

Figure 2.2: Frequencies of Wireless M-Bus modes S, T and C.

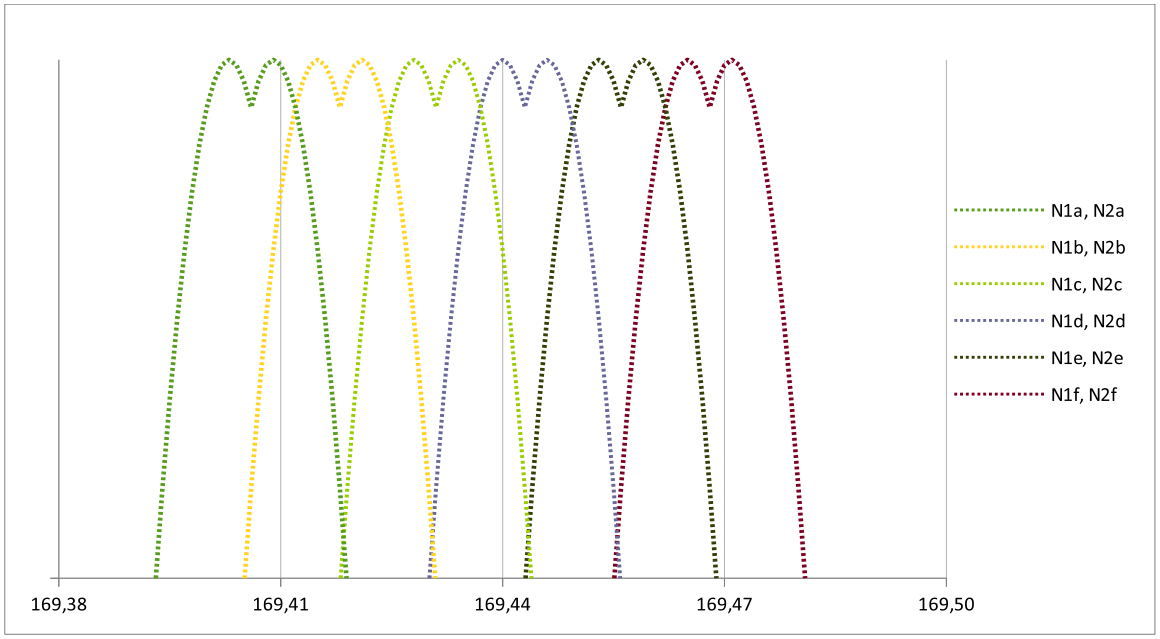

Figure 2.3: Frequencies of Wireless M-Bus mode N.

Table ?? specifies the compatibility of different Wireless M-Bus modes.

	Meter S	Meter T	Meter C	Meter $N(a-f)$
Collector S	✓			
Collector T		✓		
Collector C		✓	✓	
Collector $N(a-f)$				✓

Table 2.3: Compatibility matrix for Wireless M-Bus modes.

In summary, the Wireless M-Bus modes S and N (in mode N, every sub-mode $a-f$ is only compatible with the same sub-mode) are only compatible with the same Wireless M-Bus mode. But T is partially compatible to C - a data collector in mode C is able to communicate with a meter device in mode T.

2.5 Unidirectional and Bidirectional Communication

Based on the WM-Bus mode, two communicatino models are available:

1. Unidirectional
2. Bidirectinoal

Unidirectional WM-Bus modes support only data transmission from a meter device to a data collector. The advantages of those modes is the low overhead implementation of the single devices - a meter device must only transmit data, and a data collector must only receive data.

In contrast, bidirectional WM-Bus modes addiontally support a communication from a data collector to a meter device. Data collectors support bidirectional modes only and are able to request information from a bidirectional meter devices. Figure 2.4 shows a typical communication flow. The data collector sends a telegram "*Request User Data 2*" to the collector. The meter device receives the telegram and answers with a "*Response User Data 2*" telegram containing related information to the collector. The answer of the meter will be repeated until the collector closes the communication or a timeout occurs (according to the Frequent Access Cycle (FAC) defined in [?]).

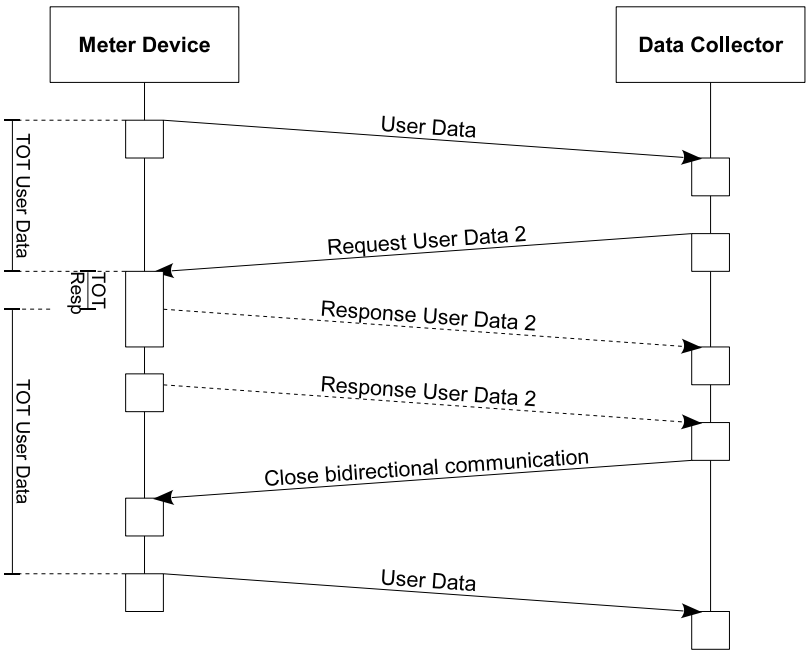

Figure 2.4: Bidirectional communication example for WM-Bus modes $S2$, $T2$, $C2$ and $N2(a-f)$.

Chapter 3

Wireless M-Bus Suite

3.1 System Requirements

The Wireless M-Bus Suite is available for 32-Bit and 64-Bit Win XP/WinVista/Win 7/Win 8/Win 8.1 operating systems. The Wireless M-Bus Suite detects the version of the installed java runtime environment and notifies the user if a newer version should be used. It requires the user to download at the official java homepage (www.java.com).

Please note, for running the Wireless M-Bus Suite on Win 8.1, the suite must be started in compatibility mode for Win 8.

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 the system type you have. This also applies for installing the correct version of the Java runtime.

3.2 Quickstart

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 starter kits with a valid serial demo application firmware programmed. You will need one board with the programmed being a collector and one board with the meter firmware to use the Wireless M-Bus Suite demo project.
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 3.2.1 to 3.2.3 provide information, how to load the demonstration project as well as basic information, which are used for collector and meter devices. Chapters 3.2.5 and 3.2.6 specify how to perform transmission tests. However, it is recommend to read all chapters in order of appearance.

3.2.1 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. A 'Welcome' screen will be displayed (see 3.1)
2. Press 'Open the demo Project'.
3. This loads the 'wmbs_demo.xml' file from the Wireless M-Bus Suite installation folder.
4. Now you can see the 'Data Collector' and the 'Smart Meter' in the configuration suite tree. In figure 3.2, the corresponding items are framed in red.

Now the 'wmbs_demo.xml' demo project configuration is loaded. In the navigation bar tree view the 'Smart Meter' is located underneath the 'Data Collector'. This means, that the 'Smart Meter' is assigned to the Wireless M-Bus network of the 'Data Collector'.

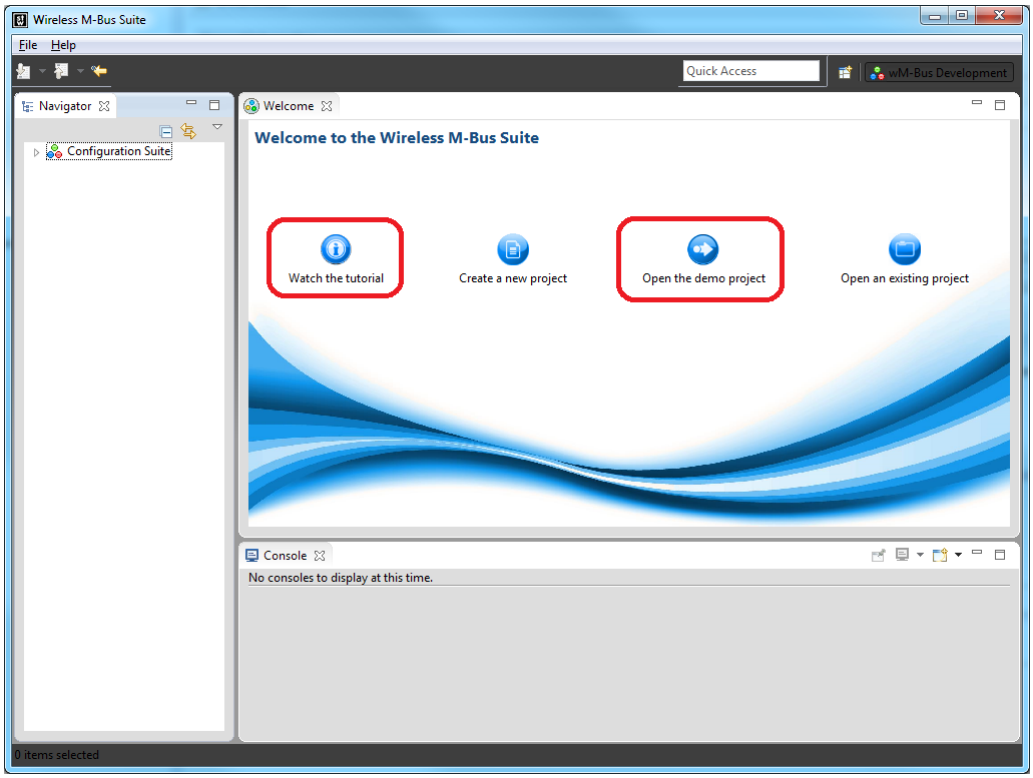

Figure 3.1: Welcome screen

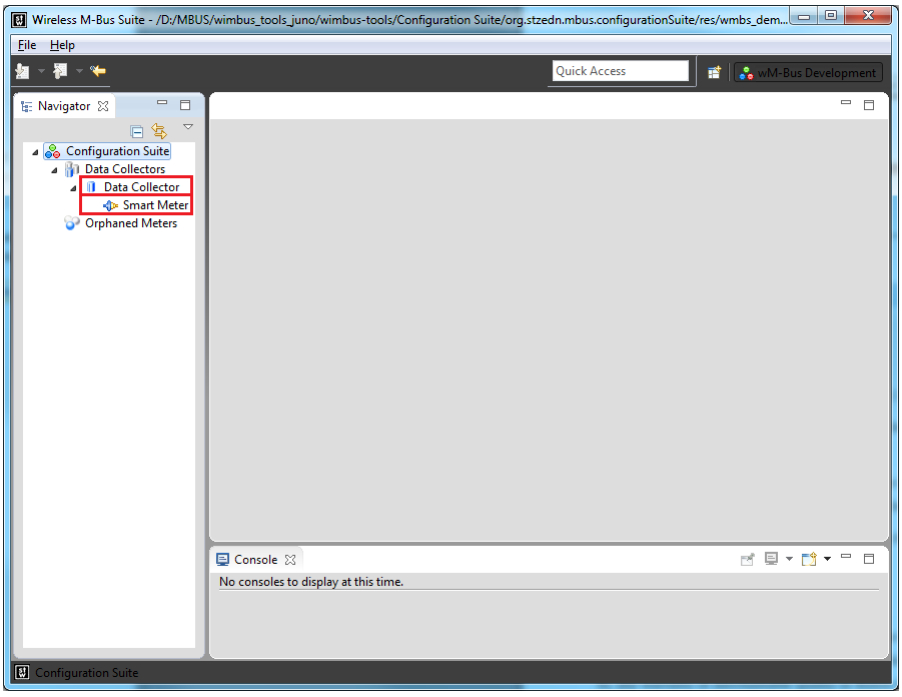

Figure 3.2: Collector and Meter device at the navigation bar

3.2.2 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 3.3 - red framed window in the 'Collectors operations' section).

The red framed window in the section 'Collectors meter list' in figure 3.3 shows the meter devices, which are associated with this collector. In this case it is the 'Smart Meter' device.

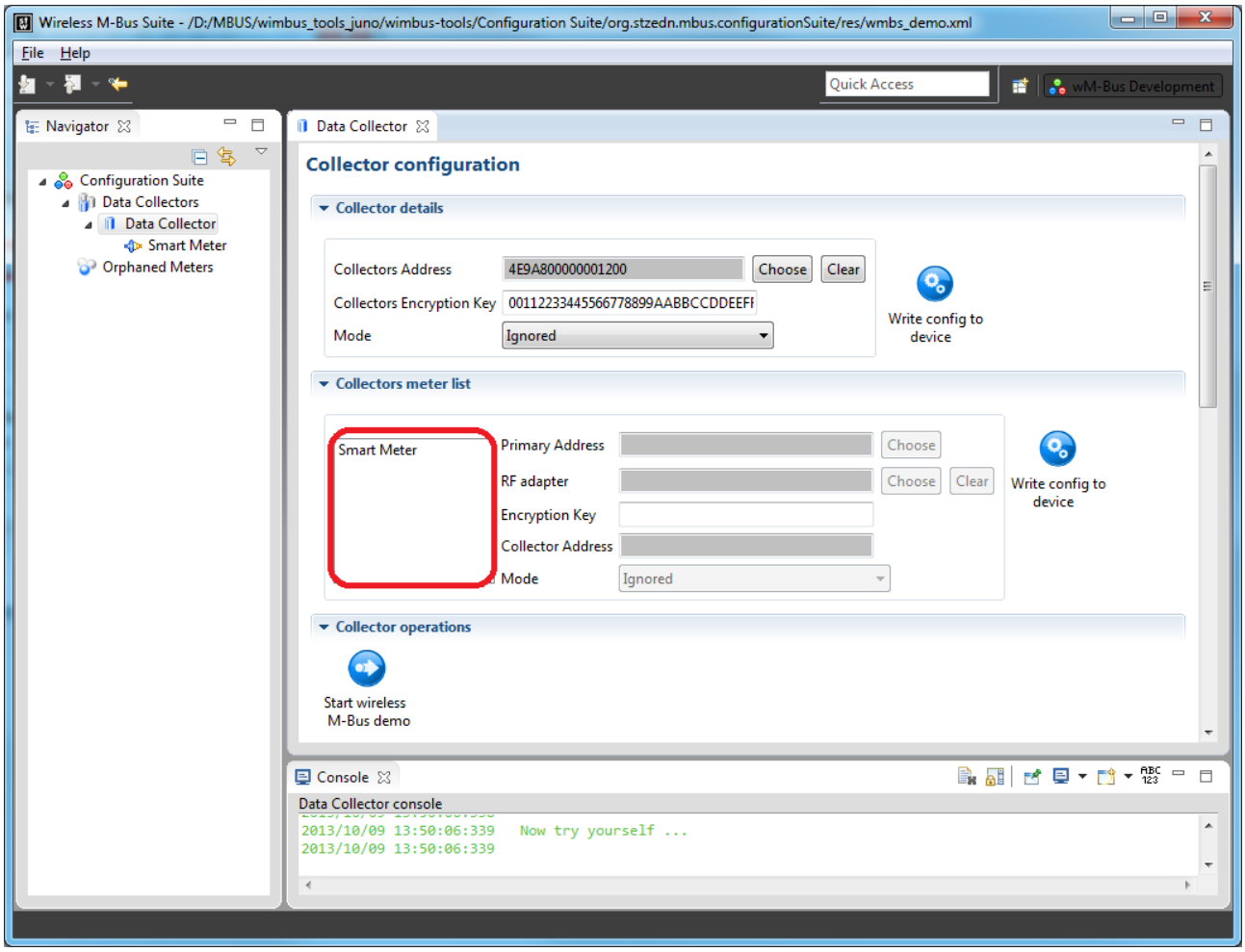

Figure 3.3: Collector editor

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 match the address used for 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 3.4, otherwise figure 3.5 notifies the user that the meter is not connected. Figure 3.6 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 3.7 is shown.

Figure 3.4: Meter connected

Figure 3.5: Meter disconnected

Figure 3.6: Meter without collector address

Figure 3.7: 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.

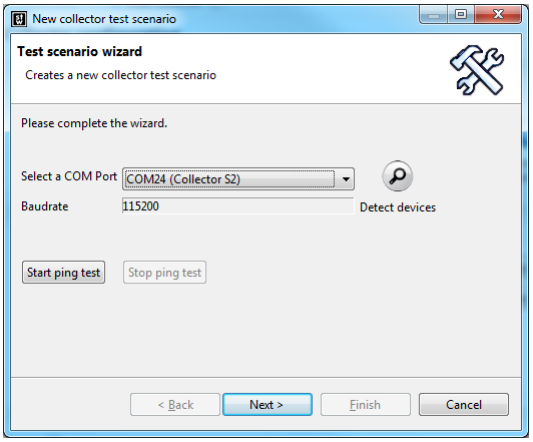

Figure 3.8: COM Port selection

Now press 'Next' and select the meters which shall be used together with the collector board and press 'Finish'.

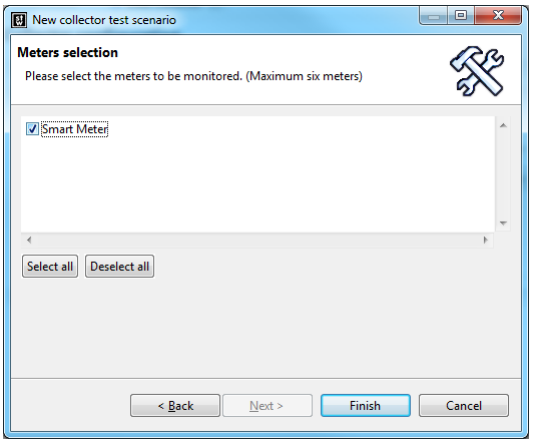

Figure 3.9: Select 'Smart Meter' meter

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

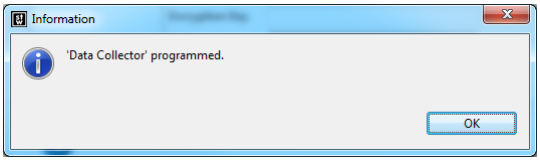

Figure 3.10: 'Data Collector' programmed

3.2.3 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.

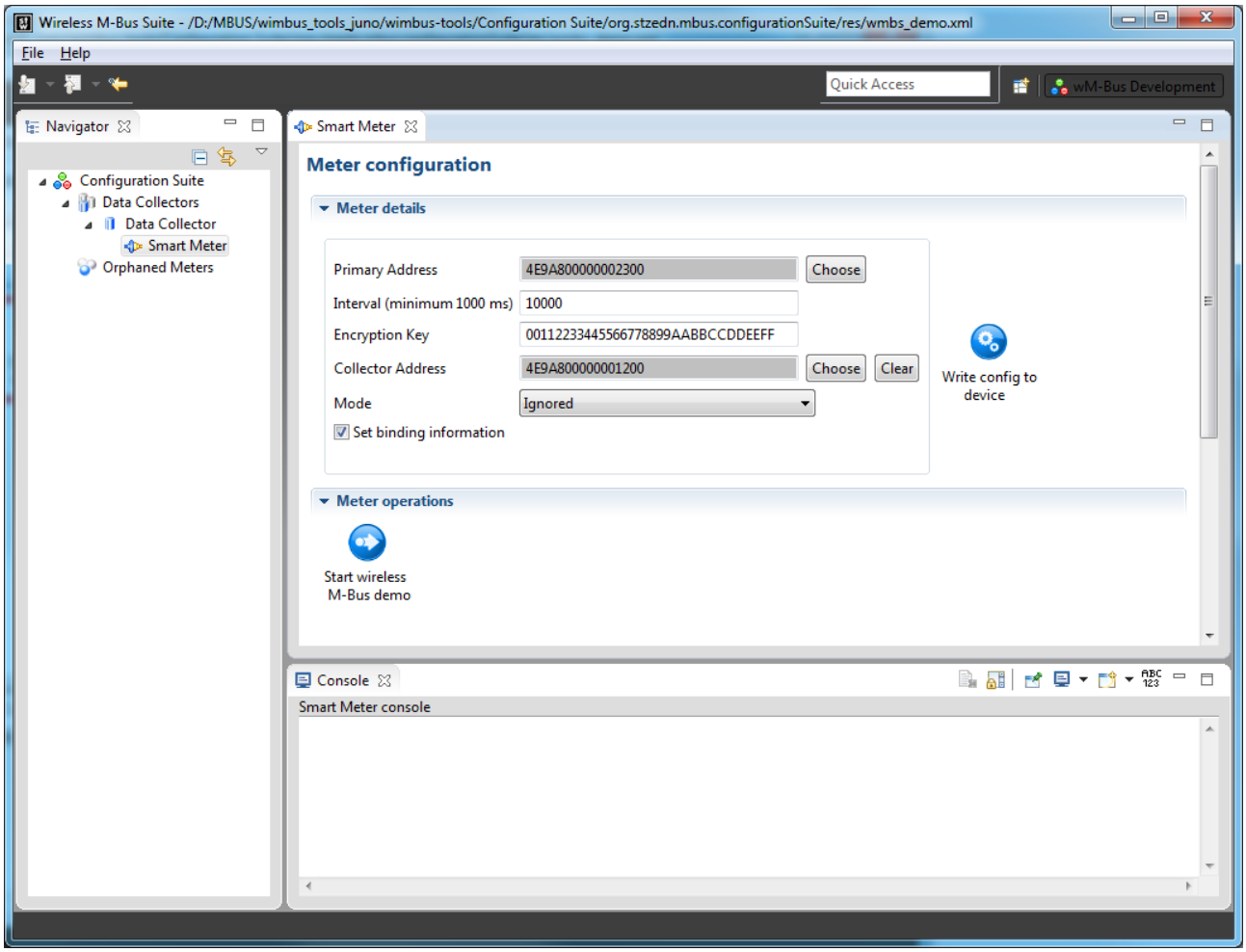

Figure 3.11: 'Smart Meter' Window

Table 3.1 lists the parameters of the 'Smart Meter':

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 3.1: '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 3.8). Then press 'Finish' and the device will be programmed. If the device was programmed correctly, the following dialog is shown.

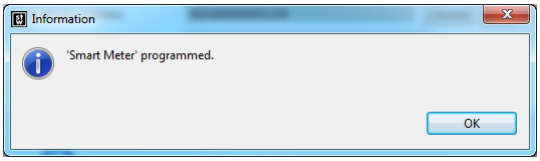

Figure 3.12: 'Smart Meter' programmed

In the demo project, the 'Smart Meter' sends a default packet to the collector every 10 seconds. This can be tracked by two possibilities:

- Navigate to the tab "*Received data content*" at the collectors view and check the table below is recording received telegrams frequently.
- Check the console for frequent activity. You should enable the textual output in order to understand what is happening more easily. Refer to chapter 3.2.4.2 for more details on how to use the console.

3.2.4 Using the Wireless M-Bus Suite

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

3.2.4.1 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.

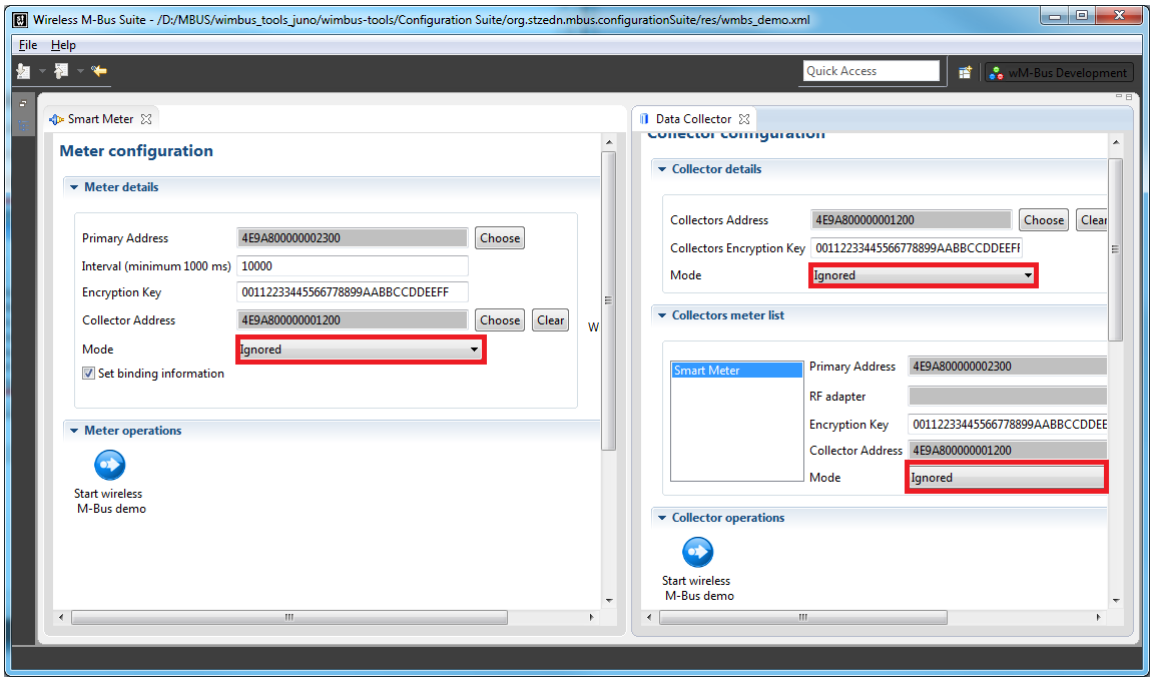

Figure 3.13: Mode selection boxes

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 3.14). 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).

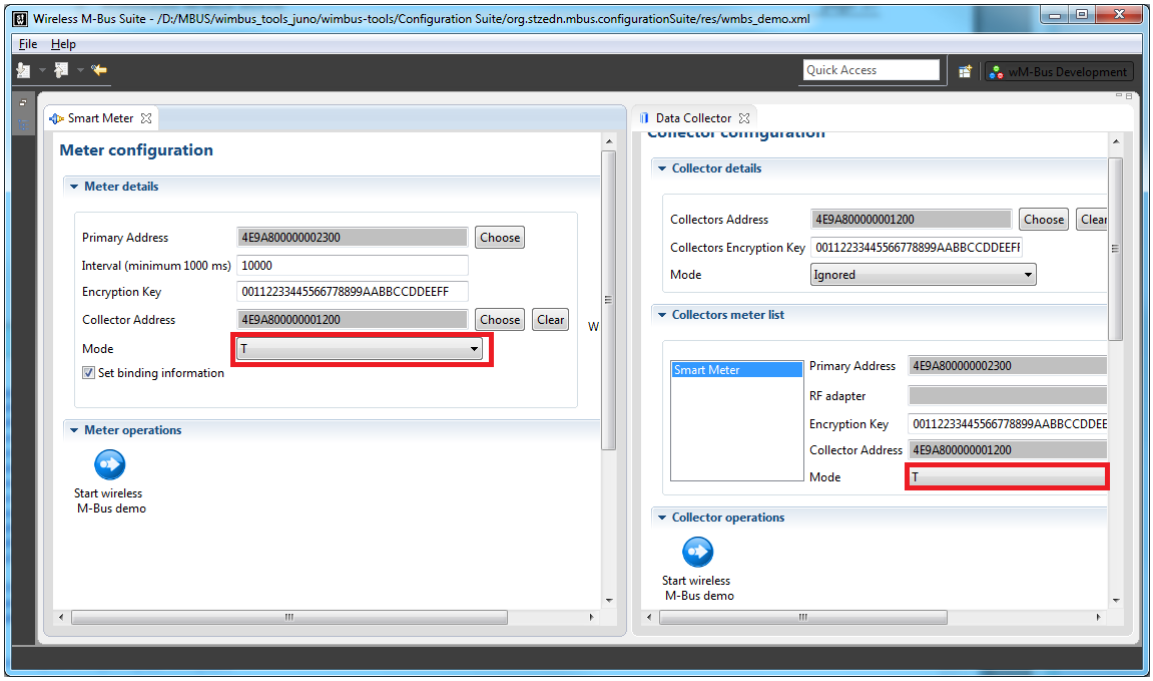

Figure 3.14: Collector C and Meter T

For using the device in mode N the selection box must be used to select and configure the channel. The figure 3.15 shows an example for the mode Nb.

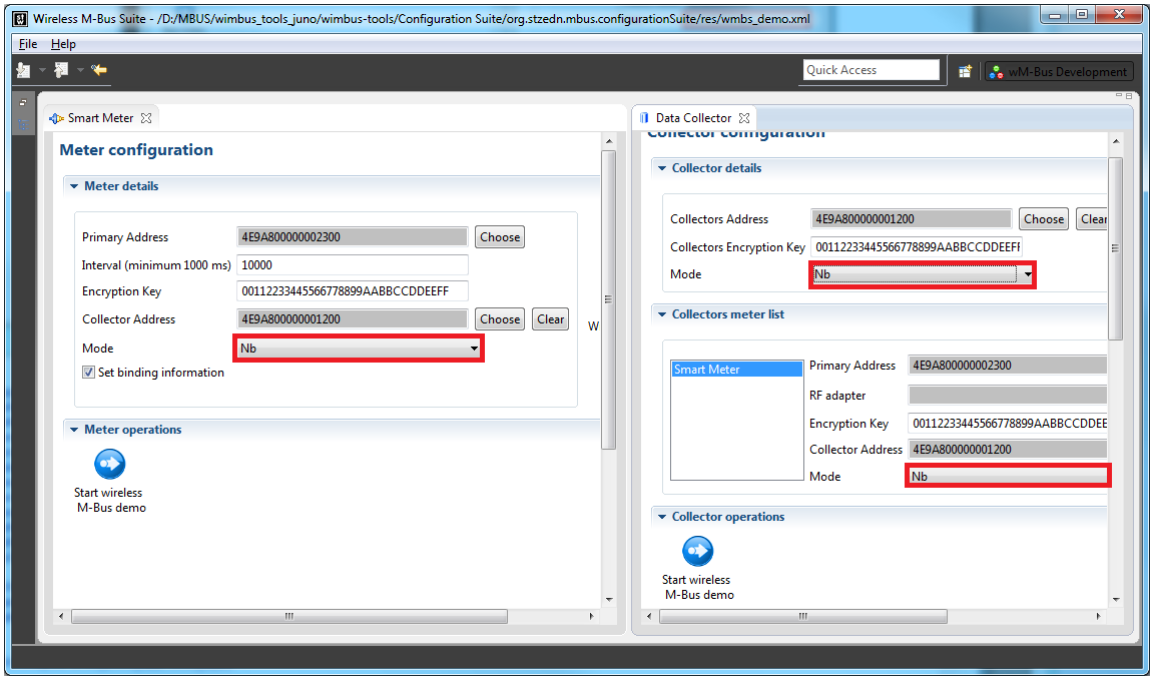

Figure 3.15: Mode Nb example

3.2.4.2 Console

The Console View displays the serrial communication flow.

Figure 3.16: Console View

The commands available in the Console View are listed below.

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 3.2: Console View Commands

The 'Text on/off' button enables or disables the text output for the serial commands (see figure 3.17).

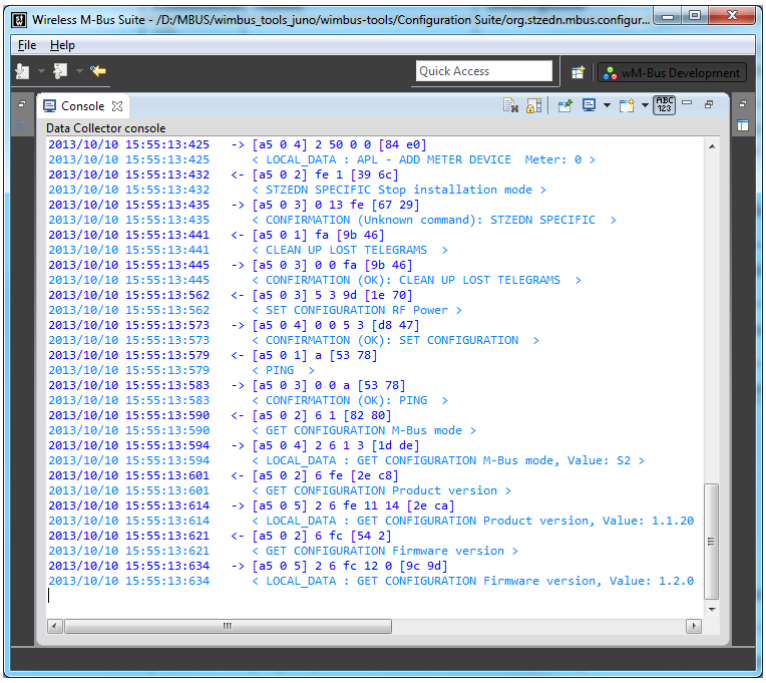

Figure 3.17: Console with serial text output enabled

3.2.5 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 3.2.1.
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 3.18).

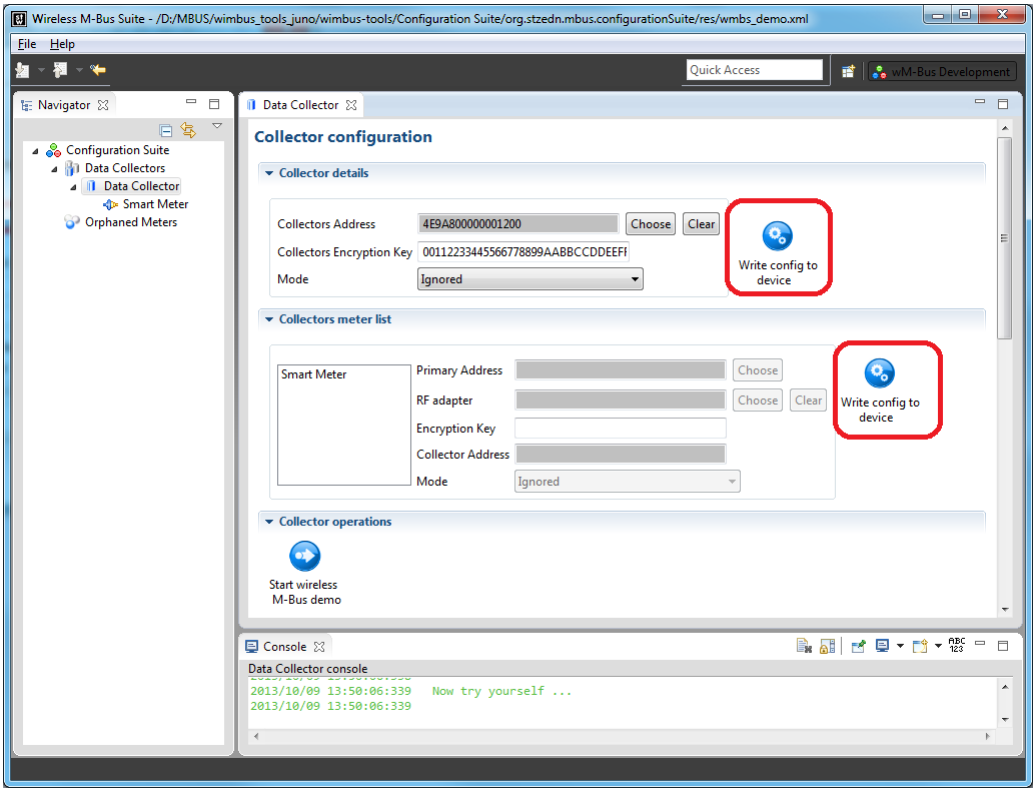

Figure 3.18: Write config to device

4. The wizard shown in figure 3.19 pops up. Configure the COM port settings to the correct COM port. Then press 'Start ping test'.

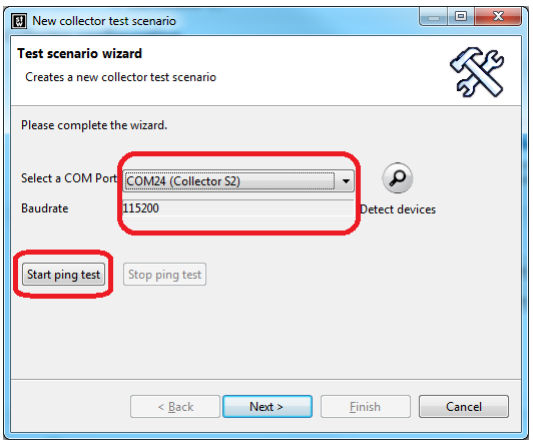

Figure 3.19: Start ping test

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 3.20 will pop up. In this case you have to load a firmware on your device, before you can perform the next steps in chapters 3.2.2 to 3.2.6.

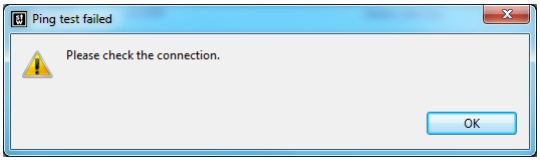

Figure 3.20: Ping test failed

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 3.21. If the ping test is successful, you can stop the ping test with the 'Stop ping test' button.

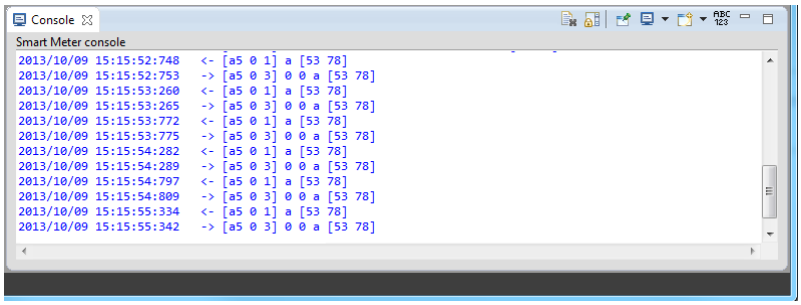

Figure 3.21: Ping test ok

3.2.6 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 3.2.1.
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.
5. Press 'Next' and set the checkbox for the 'Smart Meter' meter. (refer to figure 3.9)
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.
8. Press 'Finish'.

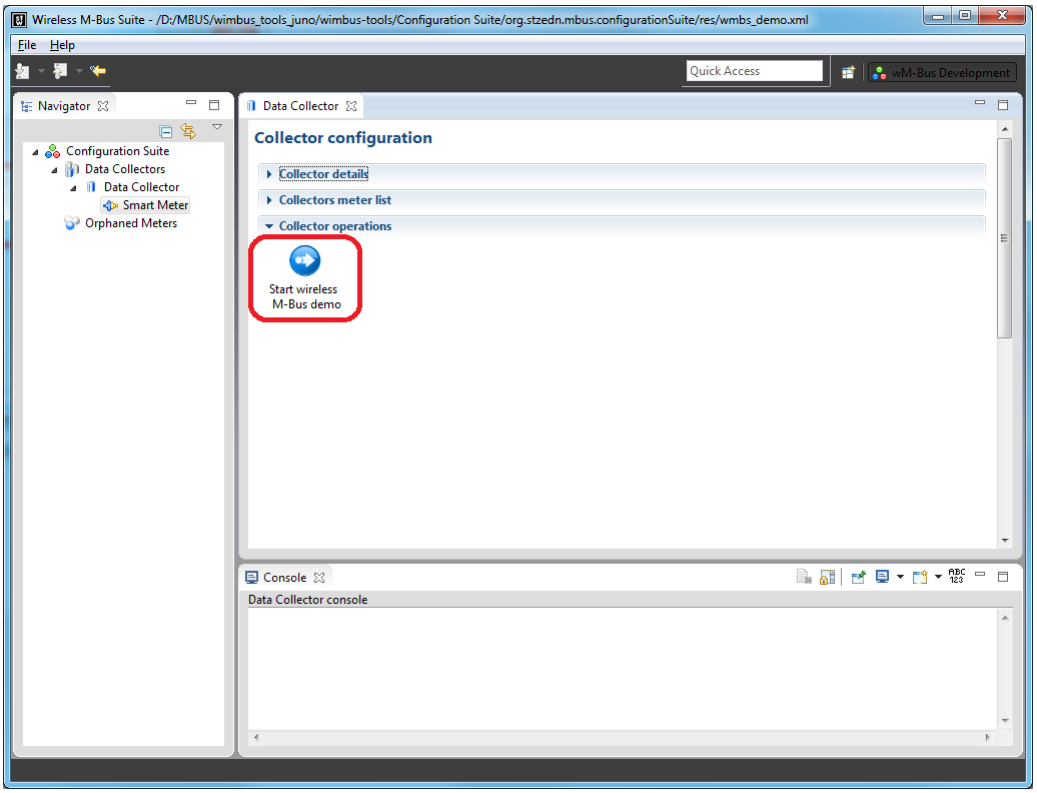

Figure 3.22: 'Start wireless M-Bus demo' button 'Data Collector'

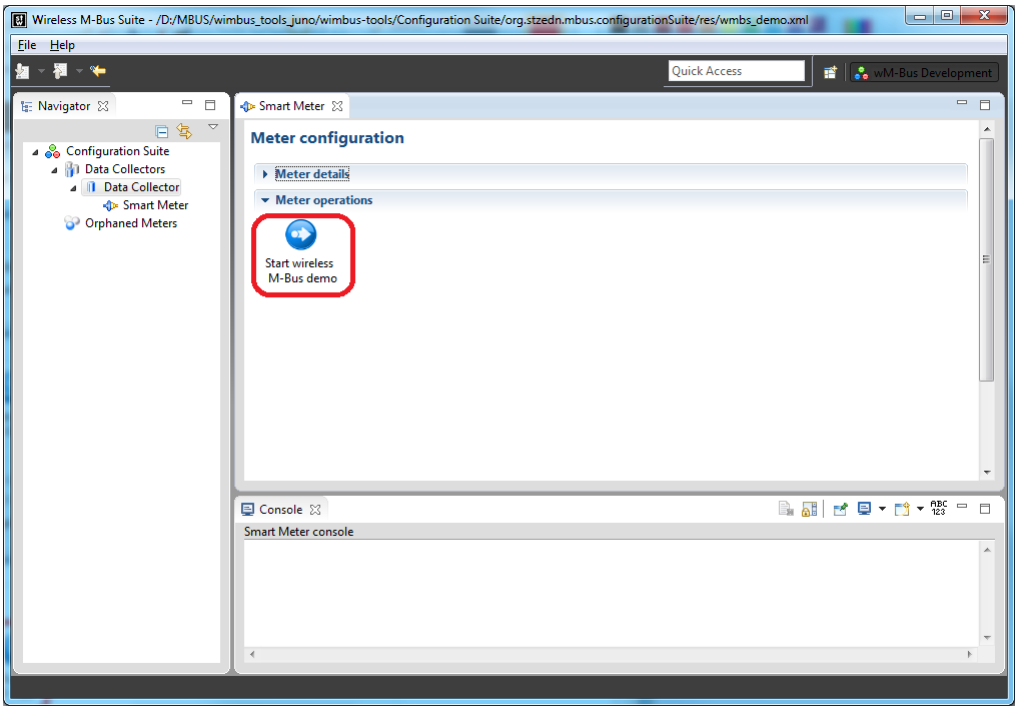

Figure 3.23: 'Start wireless M-Bus demo' button 'Smart Meter'

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 3.24.

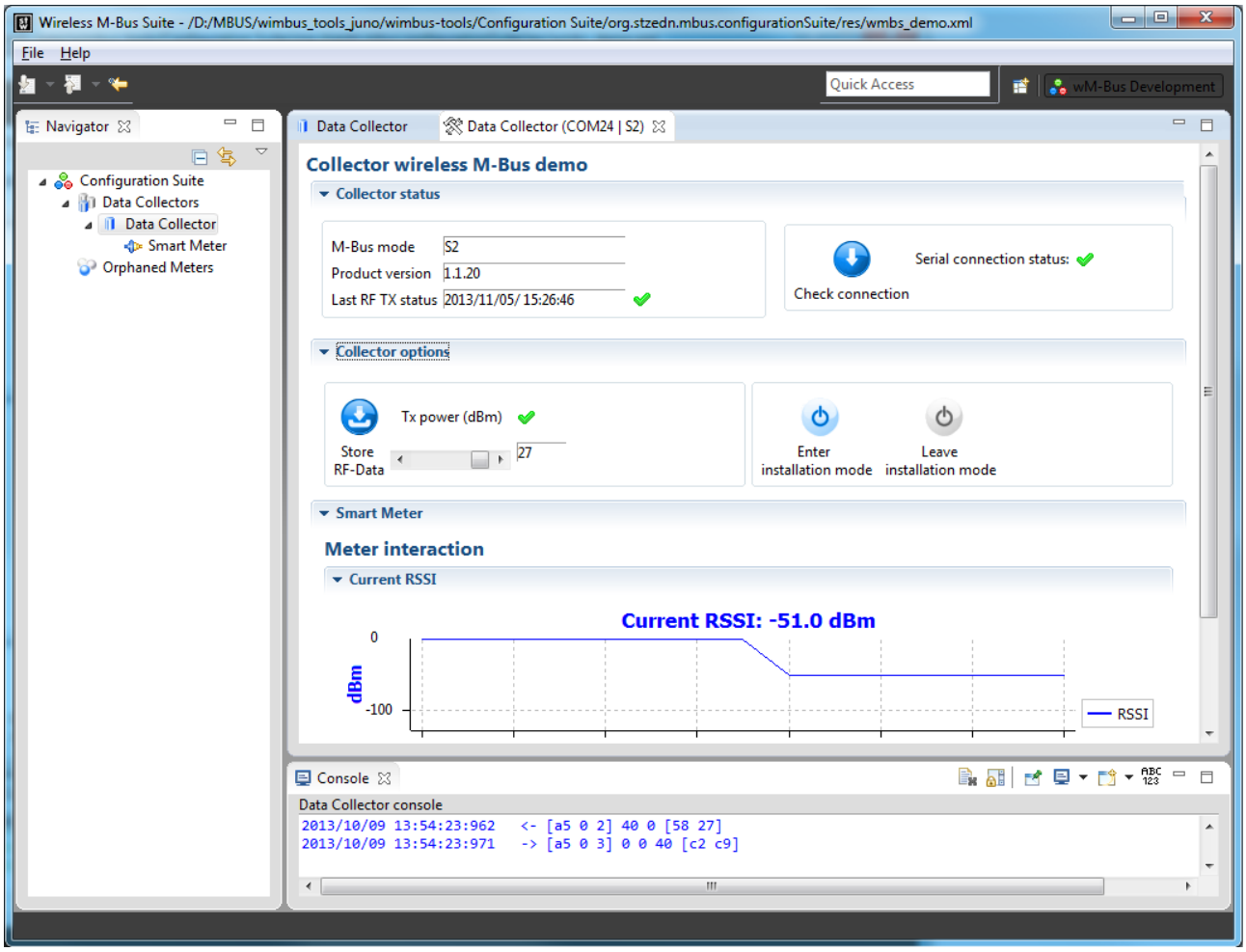

Figure 3.24: 'Data Collector' Window

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 3.25.

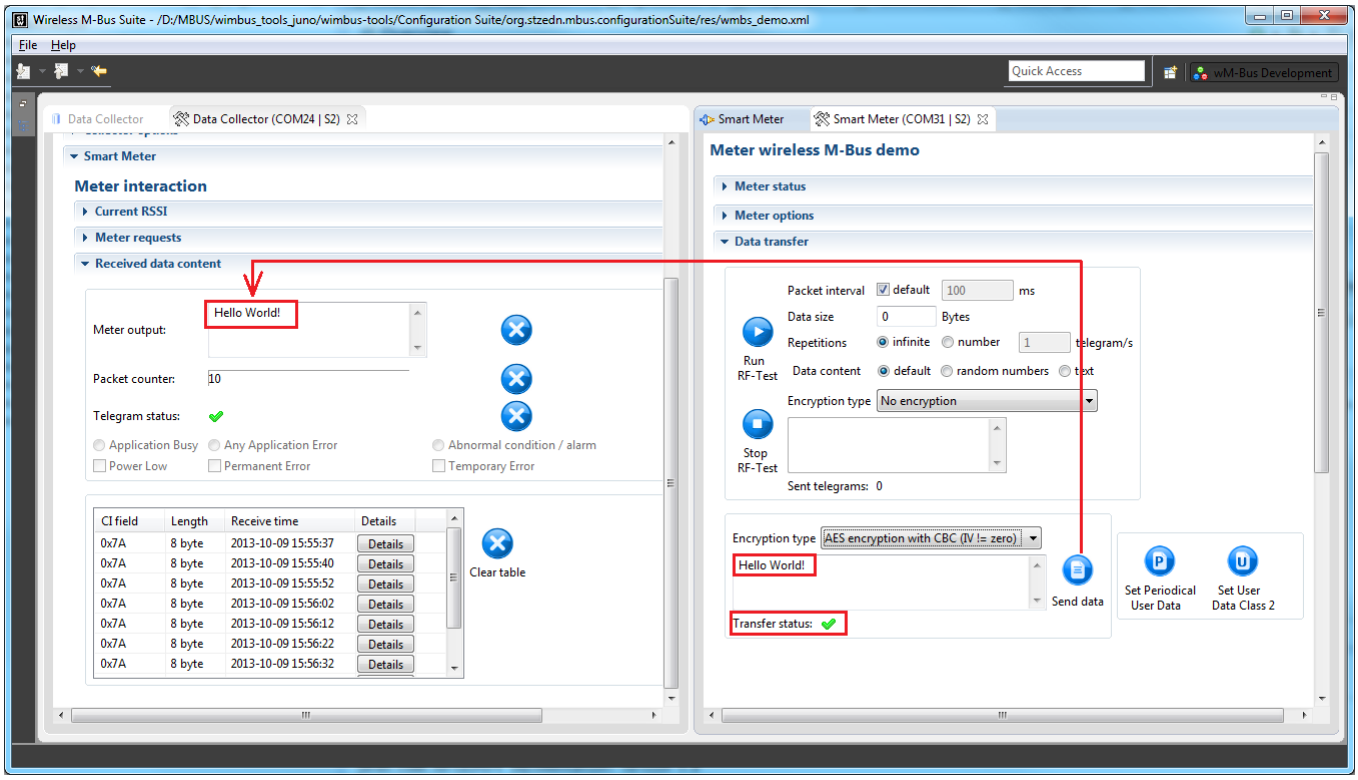

Figure 3.25: Send data test

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.

3.3 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 3.26 and 3.27 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 3.26 and 3.27. 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 3.27, 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.

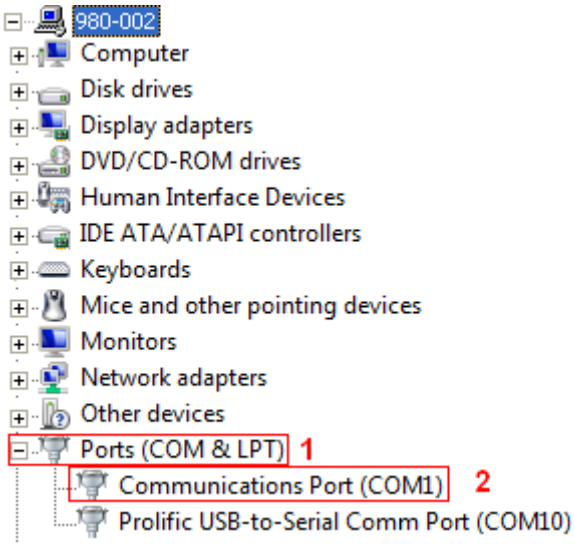

Figure 3.26: Standard COM port

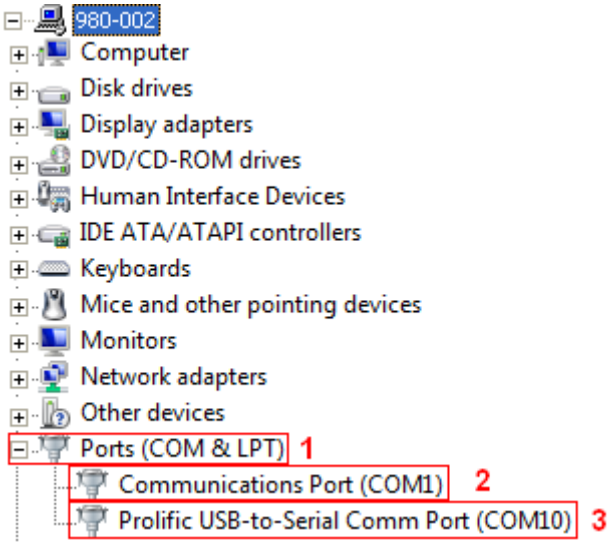

Figure 3.27: Virtual COM port

Chapter 4

Demonstration Application

4.1 Introduction

4.1.1 Top level layers

The Wireless M-Bus Stack package by Silicon Labs provides three variants of Wireless M-Bus Stack :

Serial: The *Serial* version of the stack has a serial layer on top of the APL and is commonly used in case the Wireless M-Bus Stack is operated at a separated radio module. Furthermore it's the easiest way to get familiar with Wireless M-Bus and the Wireless M-Bus Stack by using the Wireless M-Bus Suite to play around.

APL: The use case for using the Wireless M-Bus Stack on the same MCU like the application is covered by the *APL* variant of the stack. It has the APL as top level layer and can be accessed by the application directly.

TPL: The package also covers the a TPL variant of the stack. This variant has the TPL as the highest layer of the Wireless M-Bus Stack . However, the almost all of customers do need an APL on top of the stack and do not need to access the TPL directly. Please contact the support at Silicon Labs if you feel you need to access TPL for your application, to ensure this is the correct approach for your needs.

4.1.2 Demo projects

At directory `/ide/iar/` the following IAR workspaces can be found:

Demo_SLWSTK6220A: This project is meant as a base for the EZR32WG starter kit called SLWSTK6220A.

Demo_SLWSTK6200A: This project is meant as a base for the EZR32LG starter kit called SLWSTK6200A.

Demo_STK3200: This project is meant as a base for the EFM32ZG starter kit called STK3200 in combination with a Si4460 transceiver module.

Demo_STK3600: This project is meant as a base for the EFM32LG and EFM32WG starter kits called STK3600 (EFM32LG) and STK3800 (EFM32WG) in combination with a Si4460 transceiver module.

All workspaces do hold the following projects:

Demo_appserial: The easiest way to get familiar with the Wireless M-Bus Stack is to set up a serial application, that means the stack is equipped with a serial interface and must be controlled by the Wireless M-Bus Suite or any

terminal application at your host machine that is able to send *binary* commands via the serial connection.

Demo_appapl: This project is for programming an application using the APL interface of the Wireless M-Bus Stack directly. This is the best starting point if you target to use the stack on the same microcontroller that shall be used for the application as well.

Demo_apptpl: This project is for programming an application using the TPL interface of the Wireless M-Bus Stack directly. Programming directly on top of the TPL requires a deeper knowledge of EN 13757 and Open Metering System (OMS) standards.

Please note, that any changes in the source code may have impact on the principle operation of the Wireless M-Bus Stack. Please contact support at Silicon Labs in case your modifications won't work.

4.2 The Demo App

4.2.1 Basics

The demonstration applications consist of two network devices - a collector and a meter device. Thus, the application needs two starter kits.

Figure 4.1 shows the basic setup for usage in of the demo project "`Demo_appserial`" (refer to chapter 4.1) combination with the Wireless M-Bus Suite.

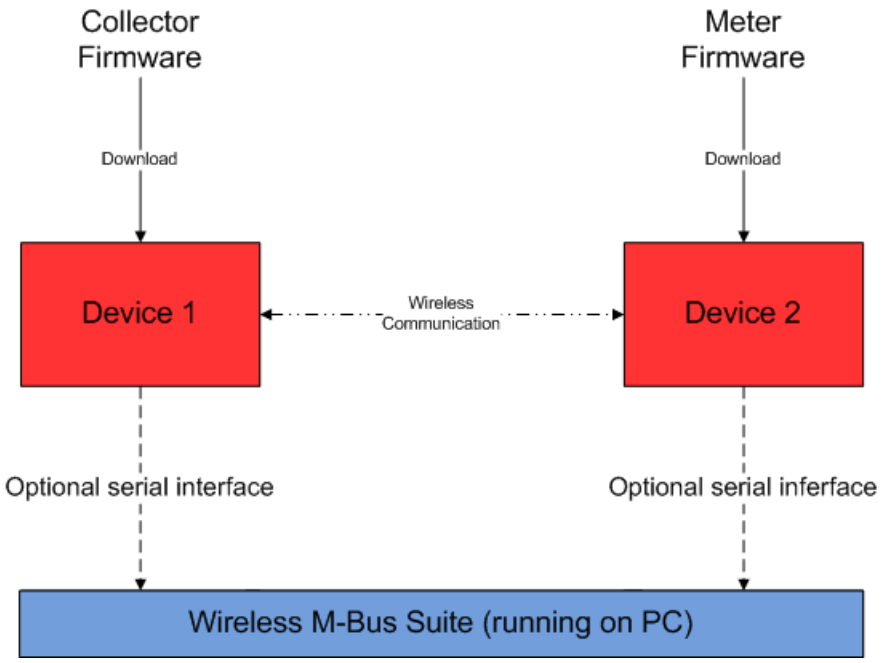

Figure 4.1: Demonstration Application

The other demo applications ("`Demo_appapl`" and "`Demo_apptpl`") do not provide a serial interface and therefore cannot be controlled via the Wireless M-Bus Suite.

4.2.2 Programming

The first step is to program the collector and the meter firmware to the starter kits.

1. Connect a starter kit to your PC. Refer to the user guide for the chosen starter kit on how to connect for programming and debugging.
2. Open up the IAR workspace and select the appropriate demo project, e.g. "`Demo_appserial`". Refer to chapter 4.1.2 for locating the IAR workspaces and to understand the differences between the projects.
3. Select the appropriate configuration, e.g. "*Meter_S1*", navigate to "*Project->Download and Debug*" in order to start building the firmware and programming to the connected starter kit.
4. When the device has been successfully programmed, the IAR debugger should be waiting at the very first line of the main function.
5. Now the IAR debugger needs to be stopped and all steps need to be repeated, but choosing the appropriate other configuration in step 3, e.g. "*Collector_S1*".

Now you can start using the programmed devices.

After programming it is a good practice to remove any debugging connection and cycle the power supply of the starter kit. However, this very much dependents on the starter kit respective board you're using.

If you have been choosing the Wireless M-Bus APL demonstration application, the meter starts transmitting a telegram to the collector device using the interval as specified within `gs_startAttr` at the main file `/src/stack/src/apps/demos/apl/main_meter.c`.

If you have been programming the "`Demo_appserial`", you now can start the Wireless M-Bus Suite , open the demo project and starte configuring the devices. Please note, that the serial demonstration application is limited in the number of supported meter devices.

4.2.3 Playing with the Wireless M-Bus Suite

The Wireless M-Bus Suite (refer to chapter 3.2) can be used to control the demonstration application on both devices (collector and meter) simultaneously via the serial connections. Of course both devices need to be connected to the PC running the Wireless M-Bus Suite.

For exploring the stack functionality, the Wireless M-Bus Suite provides the following features:

- 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 can be send to the devices.
- Create fast and easily own Wireless M-Bus configurations.
- Wireless M-Bus Suite includes an AES encryption and decryption tool.
- Download binary firmware files to the s, to switch or update applications, if a bootloader is available.

4.3 The Libraries

The Wireless M-Bus package provides a set of different libraries that are need to set up a proper Wireless M-Bus project.

- *libstack*: The library holding the core of the Wireless M-Bus Stack.
- *librf*: The RF driver for the Si446x transceiver series.

4.3.1 Wireless M-Bus Stack library

All the Wireless M-Bus Stack libraries can be found in `/libs/stack`. The directory is sorted like this: `/libs/stack/`*‹mcu-family›*`/`*‹wmbus-device-mode›*`/`*‹app-type›*`/libstack.a`

- *‹mcu-family›*: This subdirecty corresponds to the Cortex family the MCU you are using belongs to.
- *‹wmbus-device-mode›*: This subdirectory corresponds to the wanted Wireless M-Bus mode and device type.
- *‹app-type›*: This subdirectory corresponds to the wanted top level layer the stack has. Refer to 4.1 for details.

For example, if you'd like to create a Wireless M-Bus application for a Collector using mode T2 based on an EFM32WG MCU, you need to include the following library into your project: `/libs/stack/CortexM4F/Collector_T2/apl/libstack.a`

4.3.2 RF driver library

All the required RF driver libraries can be found in `/libs/rf`. The directory is sorted like this: `/libs/rf/`*‹app-type›*`/`*‹mcu-family›*`/`*‹wmbus-device-mode›*`/librf.a`

- *‹model›*: This subdirectory corresponds to the Si446x derivative to be used. Please note, the Rev.C models need a different library which is indicated by a following *C*, e.g. *Si4460C*.
- *‹mcu-family›*: This subdirecty corresponds to the Cortex family the MCU you are using belongs to.
- *‹wmbus-device-mode›*: This subdirectory corresponds to the wanted Wireless M-Bus mode and device type.

For example, if you'd like to create Wireless M-Bus firmware for a unidirectional Meter using C mode based on a EFM32LG MCU using a Si4460 transceiver, you need to include the following library into your project: `/libs/rf/Si4460/CortexM3/Meter_C1/librf.a`

Chapter 5

Contact information

Silicon Labsis licensed to use and distribute the Wireless M-Bus Stack developed by STACKFORCE . Usage is limited to usage in combination with products by Silicon Labsonly.

Please contact the Support at Silicon Labsfor support on the Wireless M-Bus package distributed by Silicon Labs.

For requests on modification, extension or porting of the stack as well as embedded development services beyond the scope of Wireless M-Bus or simply for sharing ideas for further improvements, please contact:

STACKFORCE GmbH
Poststrasse 35
79423 Heitersheim
Germany

Freiburg HRB 711613
Managing Director: David Rahusen

tel: +49 7634-69960-20
fax: +49 7634-69960-30
web: www.stackforce.de
mail: metering@stackforce.de