

Design, Simulation, and Verification Techniques for Highly Portable and Flexible Wireless M-Bus Protocol Stacks

Axel Sikora^{a*}, Dirk Lill^b

^a University of Applied Sciences Offenburg, Badstrasse 24, D77652 Offenburg, Germany

^b Steinbeis Transfer Center Embedded Design and Networking, Poststrasse 35, D79423 Heitersheim, Germany

Abstract

Wireless M-Bus according to EN 13757 is a major contender for Local Metrological Network (LMN) of Smart Metering and Smart Grid applications, as it holds the promise of a flexible, albeit optimized solution. It enjoys wide popularity in continental Europe, but increasingly in many other regions of the world. However, Wireless M-Bus is characterized by a wide variety of different operation modes (C-, N-, P-, Q-, R-, S-, and T-modes), which work in different frequencies (i.e. 868 MHz, 433 MHz, and 169 MHz). It can be enhanced by extensions from groups, like Open Metering System (OMS) Group, or national bodies.

This contribution describes requirements, design techniques and experiences from the development of highly efficient Wireless M-Bus protocol stacks, which also support good flexibility with regard to MCU- and RF-hardware. The presented approach is not limited to the use of modern software engineering design processes, as such, but also includes essential additional features like testing or simulation, as well as tools for commissioning and monitoring.

Keywords: Local metrological network, wireless M-bus, EN13757, software engineering

1. Introduction

Efficient, low-cost and stable communication solutions are a major stepping stone for smart metering and smart grid applications. This especially holds true for the so-called primary communication or Local Metrological Network (LMN) between a local sensor or actuator and a data collector or gateway. LMNs have the potential to become the first machine-to-machine-(M2M)-application with really large-scale multi-vendor installations.

Wireless M-Bus according to EN 13757 is a major contender for LMN of Smart Metering and Smart Grid applications, as it holds the promise of a flexible, albeit optimized solution. It enjoys wide popularity in continental Europe, but increasingly in many other regions of the world. However, Wireless M-Bus is characterized by a wide variety of different operation modes (C-, F-, N-, P-, Q-, S-, and T-modes), which work in different frequencies (i.e. 868 MHz, 433 MHz, and 169 MHz). It is enhanced by extensions from groups, like Open Metering System (OMS) Group [6], or national bodies, e.g. [11].

2. Wireless M-Bus and its Derivatives

The Metering Bus (short M-Bus) is a field bus specialized for primary communication in metering networks. Several meter devices are sending its data to a data collector saving and forwarding it e.g. to a in-home display or an energy supplier. The data collector may be installed in the premises, in a fixed location close out the premises or may be used as a mobile readout unit.

The different versions of M-Bus are specified by the European Standard EN 13757, which is worked

* Manuscript received June 14, 2012; revised August 10, 2012.

Corresponding author. Tel.: +49-781-205-416; E-mail address: axel.sikora@hs-offenburg.de.

out by TC294 at CEN/CENELEC. The EN 13757 is divided into the five parts from Table 1. The M-Bus standard describes physical and data link layers as well as the application layer, which promises vendor-independent interoperability.

Table 1. EN-13757 - Communication systems for meters and remote reading of meters [1]

Part	Title	Description
1	Data exchange	The first part describes the basic communication between the meters and a central data collector.
2	Physical and link layer	This part includes the specification of the physical data transmission using wired connections. It also includes the description of the protocol to transmit the data.
3	Dedicated application layer	The third part of M-Bus describes an application protocol, which allows the data transmission of meters multivendor capability. So devices of different manufacturers may be combined in one system.
4	Wireless meter readout	This part specifies the wireless communication of M-Bus and is the main document for this application mode. It includes the Physical and the Data Link Layer and can be operated together with wired devices with specification (EN 13757-2).
5	Relaying	The last part includes different proposals for routing the meter data to provide larger distance between the meters and the data collector.

Table 2. Operating modes of wireless M-bus

mode	frequency	uni-/bi-directional	description	
stationary	S1	868.3 MHz	unidir	The metering devices send their data several times per day. In this mode, the data collector may sleep most of the time. It must be woken up before the metering devices are sending data.
	S1-m	868.3 MHz		identical to S1, but the data collector must be constantly switched on
	S2	868.3 MHz	bidir	bidirectional version of S1.
frequent transmit	T1	868.95 MHz	unidir	The metering devices send their data at intervals of several seconds or minutes. suitable for walk-by and/or drive-by readout.
	T2	868.95 MHz 868.3 MHz	bidir	bidirectional version of T1.
frequent receive	R2	868.33 MHz 868.03 MHz + n*60kHz	bidir	by using frequency multiplexing several metering devices may be read simultaneously
	Q			protocol using precision timing
	P	868.33 MHz 868.03 MHz + n*60kHz	bidir	protocol using routers a network consists of one primary station (data collecting unit or concentrator) and a number of secondary stations. The network topology is hierarchical, with the primary station being the root node; secondary stations can act as router nodes, meter devices, or both. In order for the data collecting unit to establish the transmission path to any given network node in the tree, a search procedure and a network management protocol are defined which allow finding the nodes directly reachable and discovering the path to nodes not directly reachable.
compact	C1	868.95 MHz	unidir	similar to mode T, but allows higher data rate with identical energy budget and duty cycle
	C2	868.95 MHz 869.525 MHz	bidir	The common support of mode T and C frames with a single receiver is possible.
narrow-band VHF	N1a-f	169 MHz	unidir	optimized for narrowband operation in the 169 MHz frequency band, which is a reserved band for meter reading and a few other services.
	N2a-f	(@ 12.5 kHz)	bidir	
	N1g	169 MHz	unidir	sub-mode N2g mainly for long range secondary communication using multi-hop repeaters
frequent receive and transmit	N2g	(@ 50 kHz)	bidir	
	F2			the bidirectional sub-mode F2 transmits a frame and waits for a short period for the reception of a response. This response will open a bidirectional communication channel.
	F2-m	433.820 MHz	bidir	wake up messages from a stationary or mobile transceiver to the meter device to open a communication channel.

In these parts, EN-13757 describes different communication schemes (the so-called modes) for unidirectional and bidirectional data flow. These are described in Table 2, which also shows the dazzling variety of options. The different modes allow an optimized fit to the different requirements of different markets, regions, and topologies.

As the early versions of the Wireless M-Bus protocol had only insufficient support for commissioning and full application interoperability, space was left for extensions on the application layer. Dutch Smart

Meter Requirements (DSMR) and the German centric Open Metering Specification (OMS) are the most prominent examples for these extensions, which fit into the general stack according to Fig. 1. Especially, the OMS group is in the process to develop a quite complete ecosystem, which includes testing and certification support. However, it also seems to leave some space for neighbouring activities.

Manufacturer Specific Application		
Open Metering Specification (OMS)	Dutch Smart Meter Requirements (DSMR)	
Application Layer (EN 13757-3)		
Routing Layer (EN 13757-5) (optional)		
<u>Wireless</u> (EN 13757-4) Data Link Layer Physical Layer		<u>Wired</u> (EN 13757-2) Data Link Layer Physical Layer

Fig. 1. Stack overview of wireless M-bus

3. State of the Art and Available Solutions

A good number of software implementations exist for the Wireless M-Bus protocol stack. Many of them come directly from module manufacturers and are closely linked to the specific hardware business, and therefore do not need this high degree of modularity and portability. In contrast to these solutions, the authors provide a portable and flexible solution, which potentially can also be directly integrated into a single chip system solution together with customer specific functionality.

In comparison to its brother ZigBee, which has found broad reception also in the research community, Wireless M-Bus has not succeeded in receiving this attention from scientific world, yet. Consequently, most of the developments are directly product related, use neither new design approaches nor new paradigms, and are neither experimental nor future-oriented.

The different Wireless M-Bus modes are mainly using FSK modulation mostly within the specifications of the proprietary transceiver ICs from various silicon manufacturers. Whereas, an broad multitude of devices are able to support the 868 MHz or the 433 MHz specifications, only recent product launches meet the specifications for 169 MHz n-mode.

The transceiver ICs pose various challenges: 1) the interfaces to the transceiver ICs from different generations have many variations. 2) Some of the coding schemes of the different Wireless M-Bus modes (i.e. the 3-out-of-6- and the Manchester-codes) are not supported by all transceiver ICs, and then have to be emulated on the MCU. 3) Some of the preamble schemes of the different Wireless M-Bus modes (i.e. 18 bit synchronization word) are not supported by all transceiver ICs, and then have to be emulated on the MCU. 4) Partial AES encryption is not supported by most transceiver ICs in hardware and thus it also has to be executed on the MCU.

A variety of module manufacturers e.g. provide hardware and firmware solutions. The authors' team cooperate with [2], who provides hardware modules for the authors' firmware.

4. Projects Background

The authors' team is active in a broad variety of projects on both the research and development level, i.e. it has developed an own Wireless M-Bus stack being licensed and applied in a variety of projects with meter manufacturers and silicon providers. In addition, research projects help to investigate in future solutions and directions, and show, how modern design techniques can support real developments.

4.1. DEMAX

The target of the DEMAX project („Decentralized Energy and Network Management Using Flexible Energy Prices, http://www.vdivde-it.de/innonet/projekte/in_pp199_DEMAX.pdf) is the distributed measurement and control of energy consumption and production at individual household level. For this,

legacy tertiary communication techniques are used for the data exchange between premises and utilities. For the DEMAX in-house primary communication, the Wireless M-Bus protocol stack was developed.

4.2. ME³GAS

ME³GAS ('mi:gas) is the acronym for MEEEGAS, which again stands for "Smart Gas Meters & Middleware for Energy Efficient Embedded Services" (<http://www.me3gas.eu/>). The objective of the ME³GAS project is to put consumers in control of their energy efficiency and appliances at home, as it is imposed by European Directives on energy consumption. In this context, the ME³GAS project addresses the development of a new generation of smart gas meters, based on embedded electronics, communications and the remote management of a shut-off valve, which shall offer a whole range of added values: management of multiple tariffs and payment modalities, remote gas cut off, security alarms, absolute index, temperature correction, and alike. In order to hide the complexity of the underlying device and communications technologies for application developers and to raise the level of programming abstraction to a web services layer, the ME³GAS platform shall provide the necessary functionality and tools to add energy efficiency features to device networks. ME³GAS will only have commercial and residential relevance if it is successful in saving energy and cost in real-world applications.

To demonstrate this capability without any impact on the user comfort or convenience, the platform includes a critical step of retrofit installation of the developed hardware and GUI platforms into real applications. ME³GAS uses real-time energy information as energy-awareness services for all residents and combine household specific services with a community portal. This enables collective, community activity motivating positive competition in saving energy, complemented by courses on towards the education on energy efficiency, sustainability and clarifying complicated legislation aspects. Details of the communication related aspects of the ME³GAS project can be found at [3].

4.3. WiMBex

The WiMBex project (Remote wireless water meter reading solution based on the EN 13757 standard, providing high autonomy, interoperability and range, <http://www.wimbex.com/>) adds a powerful set of new features to the Wireless M-Bus platforms to answer the needs of the emergent Automatic Water Meter Reading (AWMR) market in Europe. WiMBex shall exploit the powerful new features of the P- and Q-mode of the Wireless M-Bus standard and extend the use and impact of the European standard. The new Q-mode enables precise network time synchronization, so that the high-energy requirements typically associated with multi-hop wireless networks can be significantly reduced with the implementation of a Time Division Multiple Access (TDMA) MAC protocol and with an efficient energy-aware routing protocol. Energy aware routing protocols have been developed by the research community over the past decade, and have proven to be effective in large-scale network deployments. The metrics for the proposed energy-aware routing protocol are based on link quality and mote energy use (energy harvesting backup and regeneration rate) introducing fairness in the way energy is available across the network when determining optimal routing schedules. Details of the WiMBex project are described in [4] and [5].

5. Elements of the Embedded Software Engineering

The subsequent chapter describes some of the key elements of the software engineering process, which is applied here to the design of the embedded software. Many of these technologies and approaches are widely used in standard software engineering, but – up to now – found only limited application in embedded software design, due to cost, performance, and energy efficiency reasons. We will show how these objectives can be reached, while still supporting the high efficiency of a modern design flow.

The design flow starts with a detailed requirement analysis, where all aspects of the standard and customer specific requirements are explicitly listed. Based on the overall text-driven requirements specification, a full model driven design flow is executed. Models exclusively use UML2.4.1 as modeling

language, and start with sequence diagrams. From there, a finite-state machine driven design leads to the design of the corresponding state machines. These are not restricted to the mere functionality, but also include a systematic set of error handling, time-outs, and alike. In here, a large portion of complexity does not come from the pure functionality, but from all these different measures to support stability.

The platform selection includes the elements exemplified in Fig. 2 for the projects mentioned in ch. 4: (1) There are two platforms, one using native C, and the other on using TinyOS [8] with coding NesC as an event-driven OS-platform. (2) The MCU selection shows a broad variety of energy-efficient 16- and 32-Bit MCUs. (3) Various COTS transceivers support the physical layer. (4) One special case is the usage of single-chip solutions, which integrate MCU and transceiver into one system-on-chip-(SOC)-device, like the CC430. This also poses additional challenges to the software architecture, described below. (5) Another very specific case comes from the consistent usage of the developed firmware not only for the real hardware implementation, but also with the simulation environments described below.

The software architecture has to cover the multitude of operation modes from Table 2 and the multitude of the platforms from Fig. 2. It therefore supports a very modular approach.

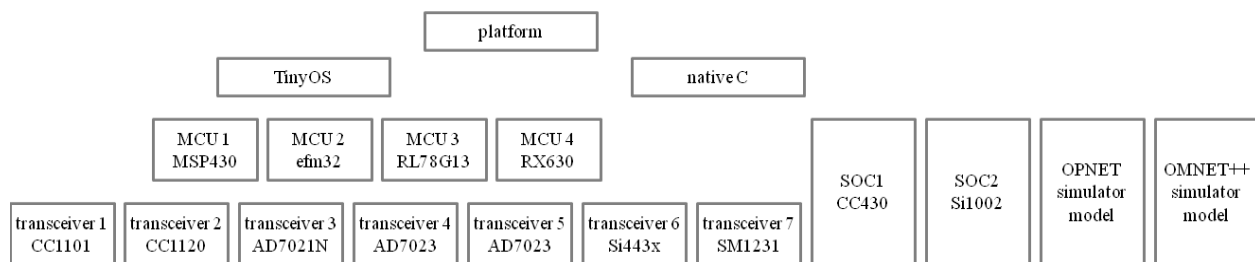


Fig. 2. Platform selection for wireless M-bus projects

Four elements support a well-defined and comfortable automatic test environment, which efficiently also supports regression testing throughout the complete software engineering process. (1) Unit test provide the lowest level and guarantee the functioning of the individual modules. (2) The second level is provided by a PC-based software, which verifies the correct and stable functionality of each communication node. (3) A network emulator provides the third level of the test environment [10]. (4) The final step is the interoperability test against third party devices. The OMS group [6] has very currently launched a standardized interoperability test-suite that is currently evaluated at the authors' team. As long as this suite is not available, each provider currently runs an own test-suite.

The use of network simulations offers several benefits regarding the development as well as the prediction of the suitability of a network. It is a special advantage of simulations that a variety of scenarios can easily be executed. The dedicated simulation environment was created with a set of Matlab functions and scripts for the simulation engine OPNET modeler [9]. Only the Hardware Abstraction Layer (HAL) and the RF drivers need to be adapted to the simulation engines' interfaces. The core stack implementation remains completely unchanged. An API Selection Layer supports automatic scenario and protocol selection from the simulation environment. The C-based Wireless M-Bus stack is directly used in a network simulator [7]. For the TinyOS based implementation, the functionality shall be simulated in the open-source network simulator OMNET++ using NesCT.

6. Tools

A Java-based fat-client has been developed for the parameterization of the nodes, for the commissioning of network and also for the execution of the functional tests, as they are described below.

For the monitoring of the spatially distributed networks, a gateway and sniffer platform provides a direct and bidirectional access to the wireless nodes. This web-server based platform supports long-term monitoring with or without online connectivity. As management traffic is HTTP and XML, the client platform shall be capable to run a JavaScript enabled web client. Thus, it can be executed on a PC, but also on portable devices, such as PDAs and iPhones. A dedicated hardware platform was developed for

this monitoring device. The software architecture of this platform contains the following elements: Its heart is the emBetter embedded web server of the authors' team. A serial handler reads and writes the data to the wireless modules. The web server software gains access to these telegrams via an exposed API, which allows for retrieval of specific telegrams as well as initialization of the buffers and deletion of the content. The data is retrieved from a web-client via HTTP. At first, the complete web page including the Java Script Libs is downloaded. Later, only the XML-Feeds follow. Thus, the communication channel remains lean. Functionality, like display, sorting, filtering, is executed on the client. This platform can also support M2M-architectures, when an automated http-client retrieves data from the distributed web servers to be stored in a central database.

7. Outlook

Further activities of the authors' team will be mainly directed towards the following topics: 1) integration of security solutions, as they are anticipated by BSI (Federal Office of Security in Information Technology [12]) into the Wireless M-Bus stack and the gateway solution, 2) involvement of harmonization between currently heterogeneous solutions on the application and the management level, and 3) support of middleware solutions for efficient operation of heterogeneous and horizontally integrated networks, which include also home and building automation networks.

Acknowledgements

The DEMAX project was supported by the German Federal Ministry of Economics and Technology (contract no. 16IN0594). The ME³GAS project is supported by ARTEMIS-JU and the participating Member States (contract no. 100266). The WiMBex project is supported by the European Commission (contract no. FP7-SME-2011-286753). For this, authors and project partners are grateful.

References

- [1] Communication systems for meters and remote reading of meters – Part 1: Data exchange; EN 13757-1; Part 2: Physical and link layer; EN 13757-2; Part 3: Dedicated application layer; EN 13757-3:2004; Part 4: Wireless meter readout (Radio meter reading for operation in the 868 MHz to 870 MHz SRD band); EN 13757-4; Part 5: Relaying; prEN 13757-5.
- [2] <http://www.pedeu.panasonic.de/pdf/engl/164Productflyer.pdf>
- [3] Sikora A, Landwehr K. Communication solutions for smart gas meters and energy efficient embedded services. Presented at: Embedded World Conference.
- [4] Sikora A, Villalonga P, Landwehr K. Extensions to wireless M-bus protocol for smart metering and smart grid application. Presented at: Int'l Conference on Advances in Computing, Communications, and Informatics (ICACCI-2012).
- [5] Sikora A, Digeser P, Klemm M, et al. Model based development of a TinyOS-based wireless M-bus implementation. Presented at: the 1st IEEE Int'l Symposium on Wireless Systems (IDAACS-SWS).
- [6] http://www.oms-group.org/en_oms.html
- [7] Sikora A, et al. Network simulation of wireless metering networks. Presented at: Embedded World Conference Nuremberg, 2012.
- [8] <http://www.tinyos.net/>
- [9] <http://www.opnet.com/>
- [10] Sikora A. An embedded Web2.0 monitoring and gateway platform for spatially distributed wireless networks. Presented at: the 5th IEEE Int'l Workshop on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2009).
- [11] www.cig.it/
- [12] https://www.bsi.bund.de/DE/Themen/SmartMeter/TechnRichtlinie/TR_node.html