DOI 10.5162/sensor2013/C6.2

# Design, Simulation, and Verification of Highly Portable and Flexible Communication Stacks for Automatic Meter Reading (AMR)

*Axel Sikora*
*University of Applied Sciences Offenburg, Badstrasse 24, D77652 Offenburg, Germany*
*axel.sikora@hs-offenburg.de*

**Abstract:**
Automatic Meter Reading (AMR) is a major enabler for the upcoming smart grid. Potentially, it will be one of the first really large-scale M2M-communication solutions for sensor applications.

To date, the definition of the standardized communication stacks for Local Metrological Network (LMN) in AMR is still ongoing. This holds true both for ZigBee Smart Energy Profile and for Wireless M-Bus according to EN 13757. During this process, there is the necessity for flexible, albeit optimized solutions, which support the different existing and upcoming versions of the communication protocols. In the case of Wireless M-Bus, the major contender for European and possibly Asian installations, this is valid not only for the 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) but also for the application layer, where additional bodies, like EN137575, Open Metering System (OMS) Group, or national bodies follow their approaches.

This contribution describes requirements, design techniques and experiences from the development of highly efficient Wireless M-Bus protocol stacks with support of good flexibility and portability between microcontroller platforms and RF-transceivers. 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.

**Key words:** 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-, D, 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 [**Fehler! Textmarke nicht definiert.**], or national bodies, e.g. [**Fehler! Textmarke nicht definiert.**].

## 2. Wireless M-Bus and its Derivatives

The Metering Bus (short M-Bus) is a field bus, which is specialized for transmitting metering data from gas-, heat-, water- or other meters. Several meter devices are sending its data to a data collector, which saves it and forwards it e.g. to a display or to an energy supplier. The data collector may be installed in the houses or it is used as mobile reading out unit. So a member of the energy supplier may drive through the streets and collect the current energy data from each household for billing.

The different versions of M-Bus are specified by the European Standard EN 13757, which is worked out by TC294 at CEN / CENELEC. The EN 13757 is divided into five parts. The M-Bus standard describes physical and data link layers as well as the application layer, which promises vendor-independent interoperability.

In these parts, EN-13757 describes different communication schemes (the so-called modes) for unidirectional and bidirectional data flow. These are described in Table 1, 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. However, the variety also calls for a well structured and modular software architecture (cf. ch. 3) in order to support efficient reuse.

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. Especially, the OMS group is in the process to develop a quite complete eco-system, which includes testing and certification support. However, it also seems to leave some space for neighboring activities.

## 3. Elements of the Embedded Software Engineering

### 3.1. Introduction

This chapter describes some of the key elements of the software engineering process applied for 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. This is all the more valid in these cases where software of very cost-efficient implementation with regard to memory footprint, processing performance, and energy efficiency shall be implemented. However, the experience from the above projects shows that these objectives can be reached, while still supporting the high efficiency of a modern design flow.

### 3.2 Specification

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, where models description exclusively use UML2.4.1 and start with sequence diagrams. From there, a finite-state machine driven design is pursued, which leads to the design of the corresponding state machines. These state machines are not restricted to the mere functionality, but also include a systematic set of error handling, time-outs, and alike. As nearly always, the largest portion of complexity does not come from the pure functionality, but from all these different measures to support stability.

### 3.2. Platform Selection

The platform selection includes the following elements:

- The basic OS- vs. non-OS-decision was answered with either native C, with a software architecture shown in Fig. 1 or with TinyOS as an event-driven OS-platform. In the latter case, coding is done in NesC.
- The microcontroller selection shows a broad range from energy-efficient 16- and 32-Bit platforms.
- The transceivers can be selected from a variety of standard products.
- 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.
- Another very specific case comes from the consistent usage of the developed firmware not only for the real hardware implementation, but also for the simulation environments from Subsection 3.4.

The software architecture has to cover the multitude of operation modes and the multitude of the platforms from Fig. 1. It therefore supports a very modular approach, shown in Figs. 2 and 3.

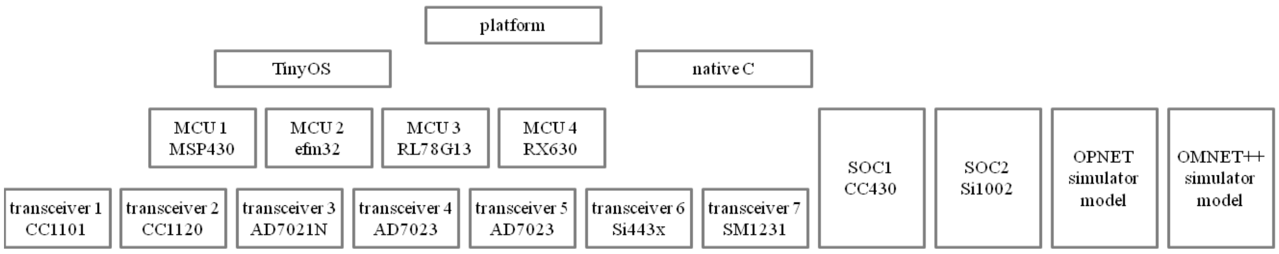


*Fig. 1. Platform Selection for Wireless M-Bus projects.*

### 3.3. Test & Verification

Four elements support a well-defined and comfortable automatic test environment, which efficiently also supports regression testing throughout the complete software engineering process.

- Unit test provide the lowest level and guarantee the functioning of the individual modules.
- The second level is provided by a PC-based software, which verifies the correct and stable functionality of each communication node.

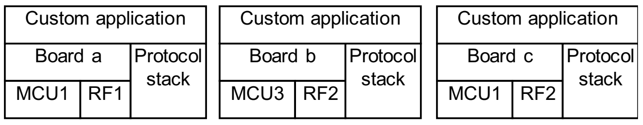


*Fig. 2. Software architecture for flexible platform selection for wireless M-Bus projects.*

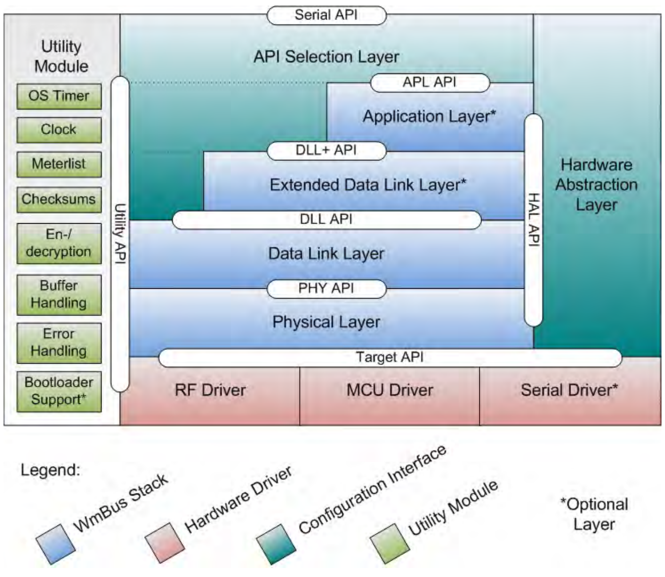


*Fig. 3. Functional Split for Flexible Platform Selection for Wireless M-Bus projects*

- A network emulator provides the third level of the test environment. This emulator was originally inspired by [10], but significantly enhanced with regard to automatic and web-based control. It is shown in Fig. 8 and interconnects the nodes with RF-wave guides, splitter-combiner and attenuation elements. These elements are remotely controlled by microcontrollers, so that arbitrary network topologies and coexistence scenarios can be generated and reproduced, and full network tests can be performed.
- The final step is the interoperability test against third party devices. The OMS group [**Fehler! Textmarke nicht definiert.**] has recently launched the first elements of a standardized interoperability test-suite. Before, each provider ran his own test-suite.

### 3.4. Simulation

The use of network simulations offers several benefits for the development, the prediction of the suitability, and the parameterization of a network. The main objectives of simulations are

- the provisioning of an early and comfortable development environment that behaves close to the later target system. Since the development and production of target hardware may be time consuming and costly, a simulation environment with abstracted hardware opens the possibility to start an early implementation and verification of concepts.
- a better observability of the internal behavior, since processes are reproducible and can be analyzed in detail. This also helps to get to know the system. Especially in cases of errors or incorrect behavior, this offers a practical starting point for debugging and problem solving.
- a better controllability and reproducibility of processes
- evaluation and prediction of the behavior e.g. of large network topologies at low efforts. The results of a simulation might help to indicate eventual bottlenecks or risks of the system and to improve its performance.

In simulations a variety of scenarios can easily be executed. Therefore, an automated simulation environment was developed to support the selection the scenarios as well as the easy parameterization of the participating models via a user interface. A further benefit of the environment is the filtering and preparation of the simulation output for a presentation according to the chosen output parameters.

The simulation environment was created with a set of Matlab functions and scripts. For the simulation engine OPNET modeler is used.

It has to be highlighted that the identical code can can be used for the embedded firmware and the simulation models. Only the Hardware Abstraction Layer (HAL) and the RF drivers need to be adapted for the integration into the OPNET simulation engine. The core stack implementation remains completely unchanged, since it does not contain any platform dependent directives. For convenience, an API Selection Layer was added to automatically select scenarios and protocols from the simulation environment.

Fig. 9 shows the node architecture of a wireless M-Bus stack with the required adaptations for the use in a network simulator. More details on the simulation setup and some results can be found at [9]. For the TinyOS based implementation, the firmware functionality can be simulated in the open-source OMNET++ using NesCT.

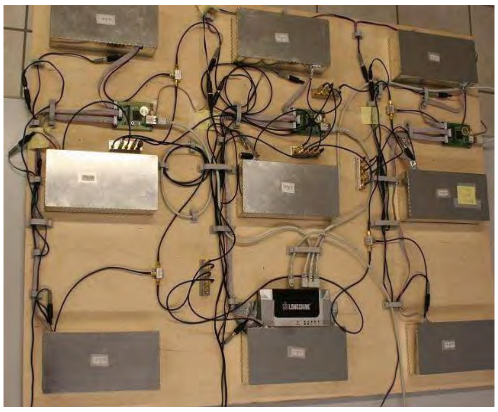


*Fig. 4. Physical test bed for automated verification of routing mechanisms.*

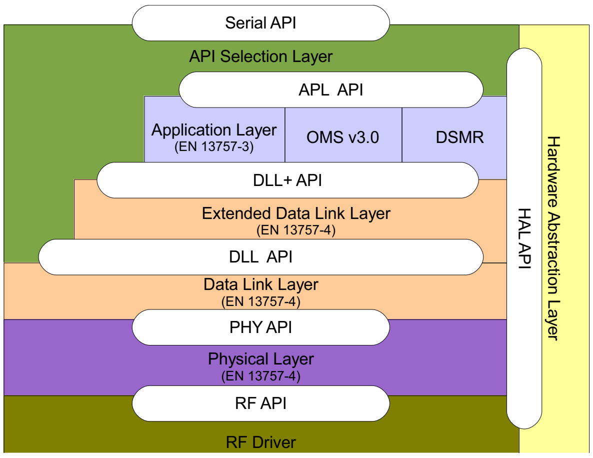


*Fig. 5. Node architecture of a wireless M-Bus stack adapted for the use in a network simulator.*

## 4. Tools

### 4.1. Commissioning

A Java-based fat-client was developed for the parameterization of the nodes, for the commissioning of network and also for the execution of the functional tests. A screenshot of this tool is shown in Fig. 6.

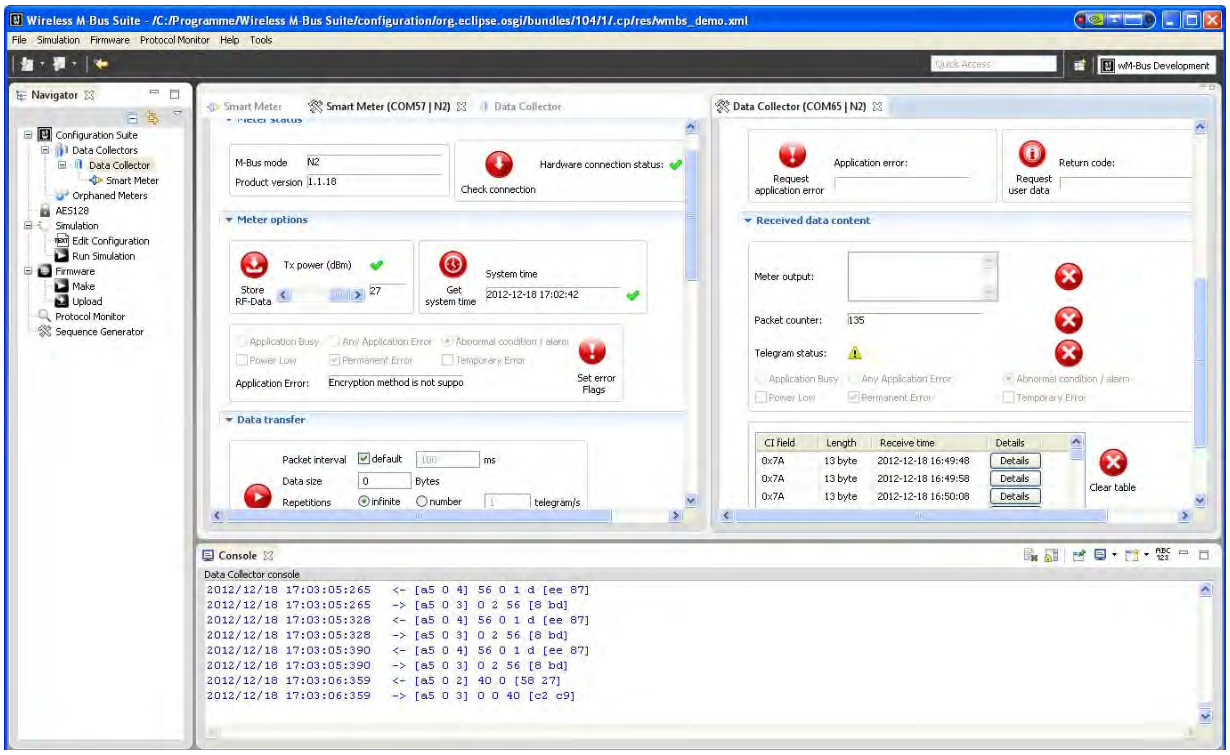
### 4.2 Monitoring

For the monitoring of the spatially distributed networks, a gateway and sniffer platform was developed to provide a direct and bidirectional access to the wireless nodes. This web server based platform supports long-term monitoring with or without online connectivity. Remote access is provided via wide-area networks (WAN), i.e. GPRS, WLAN or Ethernet. A client computer accesses the input from the management nodes. As management traffic is pure http and XML, there is only a single requirement to the client computer: It shall be capable to run a JavaScript enabled web client. Tests were performed with PC platforms, but also with portable communication devices, i.e. smart phones.

A dedicated hardware platform was developed (cf. Fig. 7). Interfaces to wireless modules allow the access to the wireless network.

The software architecture (cf. Fig. 8) contains the following elements:

- The heart is an embedded web server from the author's team.



M-Bus mode N2
Product version 1.1.18
Hardware connection status: ✓

Tx power (dBm) 27 ✓
System time 2012-12-18 17:02:42 ✓

Application Busy, Any Application Error, Abnormal condition / alarm, Power Low, [✓ Permanent Error], Temporary Error
Application Error: Encryption method is not suppo

Packet interval [✓ default] 100 ms
Data size 0 Bytes
Repetitions [✓ infinite] number 1 telegram/s

Application error:
Return code:

Meter output:
Packet counter: 135
Telegram status: ⚠
Application Busy, Any Application Error, Abnormal condition / alarm, Power Low, [✓ Permanent Error], Temporary Error

| CI field | Length | Receive time | Details |
|---|---|---|---|
| 0x7A | 13 byte | 2012-12-18 16:49:48 | Details |
| 0x7A | 13 byte | 2012-12-18 16:49:58 | Details |
| 0x7A | 13 byte | 2012-12-18 16:50:08 | Details |

Data Collector console

```
2012/12/18 17:03:05:265   <- [a5 0 4] 56 0 1 d [ee 87]
2012/12/18 17:03:05:265   -> [a5 0 3] 0 2 56 [8 bd]
2012/12/18 17:03:05:328   <- [a5 0 4] 56 0 1 d [ee 87]
2012/12/18 17:03:05:328   -> [a5 0 3] 0 2 56 [8 bd]
2012/12/18 17:03:05:390   <- [a5 0 4] 56 0 1 d [ee 87]
2012/12/18 17:03:05:390   -> [a5 0 3] 0 2 56 [8 bd]
2012/12/18 17:03:06:359   <- [a5 0 2] 40 0 [58 27]
2012/12/18 17:03:06:359   -> [a5 0 3] 0 0 40 [c2 c9]
```

*Fig. 6: Wireless M-Bus suite for commissioning, parameterization, and testing purposes.*

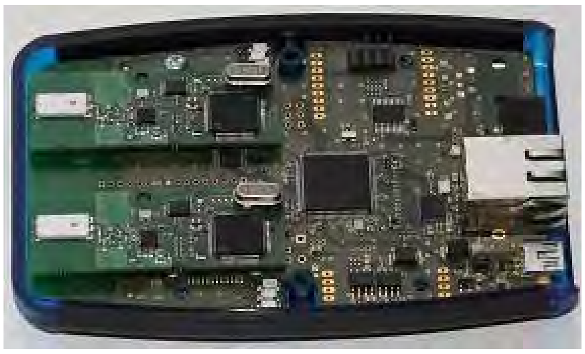


*Fig. 7. PCB for gateway platform*

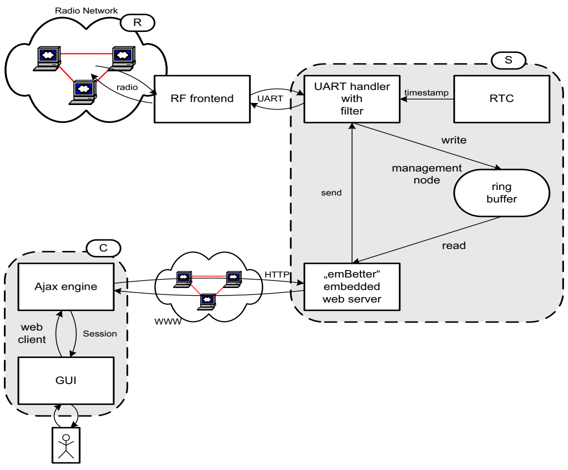


*Fig. 8. Embedded web2.0 web server in radio networks.*

- 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. Whereas the whole web page including the Java Script Libs has to be downloaded at the first run, after that it is only the XML-Feeds that follow. Thus, the communication channel can be kept very lean. It should be highlighted that all functionality is performed on the client, i.e. display, sorting, filtering, storing.

It is also well possible to access the data within an M2M-architecture. Then, an automated http-client retrieves the data from the distributed web servers and stores it in a backend database.

## 5. Outlook

Further activities of the authors' team are mainly directed towards the following topics:

- integration of security solutions, as they are anticipated by BSI (Federal Office of Security in Information Technology), into the Wireless M-Bus stack and the gateway solution, in order to secure a very vulnerable part of the infrastructure against cyberattacks [11]
- involvement of harmonization between currently heterogeneous solutions on the application and the management level, and
- support of middleware solutions for efficient operation of heterogeneous and horizontally integrated networks, which include also home and building automation networks.

## Acknowledgement

The presented material was partially developed in publicly funded projects. The author would like to mention

- the project DEMAX („Decentralized Energy and Network Management Using Flexible Energy Prices“) being partially supported by the German Federal Ministry of Economics and Technology (BMWi) under contract number 16IN0594.
- the ME³GAS project being partially funded by the ARTEMIS-JU and by the participating Member States under contract number 100266.
- the WiMBex project being partially funded by the European Commission in the 7th Framework Program under contract number FP7-SME-2011-286753.

The author and all project partners are grateful for this support.

## References

[1] R. Abe, H. Taoka, D. McQuilkin, “Digital Grid: Communicative Electrical Grids of the Future”, IEEE Transactions on Smart Grid, Vol. 2, No. 2, June 2011, pp. 399-410.

[2] R. Padil, “A Best Fit for “Short Haul Communication Protocol” in Smart Metering”, Embedded World Conference 2012, Nürnberg.

[3] G. Wu, S. Talwar, K. Johnsson, N. Himayat, K.D. Johnson, "M2M: From Mobile to Embedded Internet", IEEE Communications Magazine, April 2011, pp.36-43.

[4] Commission of the European communities, Internet of Things in 2020, EpoSS, Brüssel, 2008.

[5] Communication systems for meters and remote reading of meters EN-13757: Part 1: Data exchange, Part 2: Physical and link layer, Part 3: Dedicated application layer, Part 4: Wireless meter readout (Radio meter reading for operation in the 868 MHz to 870 MHz SRD band), Part 5: Relaying, Part 6: Local Bus.

DOI 10.5162/sensor2013/C6.2

[6] A. Sikora, K. Landwehr, “Communication Solutions for Smart Gas Meters and Energy Efficient Embedded Services”, Embedded World Conference 2012, Nürnberg.

[7] A. Sikora, P. Villalonga, K. Landwehr, "Extensions to Wireless M-Bus Protocol for Smart Metering and Smart Grid Application", International Conference on Advances in Computing, Communications, and Informatics (ICACCI-2012), Chennai, India, 2012.

[8] A. Sikora, P. Digeser, M. Klemm, M. Tubolino, R. Werner, “Model Based Development of a TinyOS-based Wireless M-Bus Implementation”, 1st IEEE Int’l Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Offenburg, 2012.

[9] A. Sikora, M. Schappacher, “Network Simulation of Wireless Metering Networks”, Embedded World Conference 2012, Nürnberg.

[10] M. Ringwald, K. Römer, "Deployment of Sensor Networks: Problems and Passive Inspection", 5th Workshop on Intelligent Solutions in Embedded Systems (WISES'07), Madrid, Spain, 2007.

[11] A. Hahn, M. Govindarasu, “Cyber Attack Exposure Evaluation Framework for the Smart Grid”, IEEE Transactions on Smart Grid, Vol. 2, No. 4, Dec. 2011, pp. 835-843.