Forschungsbericht WS 2015/2016

Kommunikation im Local Metrological Network eines Smart Meter Gateway

Prof. Dr.-Ing. Rainer Bermbach

Einleitung

Im sog. Smart Grid, dem intelligenten Stromnetz, spielt das Smart Metering eine große Rolle zur Erreichung der Energiewende. Hierfür spezifiziert das Forum Netztechnik/Netzbetrieb im VDE (FNN) im Rahmen des Projektes "MessSystem 2020" (MS-2020) [1] ein modular aufgebautes Messsystem, in dem intelligente Zähler, Smart Meter Gateways (SMGW), Zusatzgeräte und Energiedatenmanagementsysteme herstellerübergreifend miteinander kommunizieren können. Die entsprechenden Spezifikationen für das MS-2020 finden sich in einer Anzahl von sog. FNN-Lastenheften [2-5].

Neben dem FNN macht auch das Bundesamt für Sicherheit in der Informationstechnik (BSI) wichtige (System-) Vorgaben, vornehmlich in seiner Technischen Richtlinie TR-03109 [6-12]. Auch die Physikalisch-Technische Bundesanstalt (PTB) stellt beispielswiese in ihren PTB-Anforderungen PTB-A 50.8 bzw. PTB-A 50.7 [13, 14] Forderungen auf, die Komponenten eines solchen Messsystems - insbesondere aus Sicht des Eichrechts - erfüllen müssen. Solche Komponenten müssen wiederum sowohl von der PTB als auch vom BSI für ihren Einsatz im Feld zertifiziert werden.

Wesentlicher Teil der „Intelligenz“ des MS-2020 ist die Kommunikation innerhalb des Messsystems und mit den externen Einrichtungen. Smart Meter Gateways kommt hierbei eine Schlüsselrolle zu. Sie koppeln die verschiedenen Kommunikationsnetze (s. Abb. 1):

- das Wide Area Network (WAN), das die Schnittstelle zum jeweiligen SMGW Administrator sowie zu Energielieferanten, Netzbetreibern etc. bildet,
- das Home Area Network (HAN), in dem u.a. die Schnittstellen für den Verbraucher und den Servicetechniker liegen,
- das Local Metrological Network (LMN), in dem die Sensoren (praktisch meist Zähler) ihre Messwerte aufnehmen und an das SMGW übermitteln.

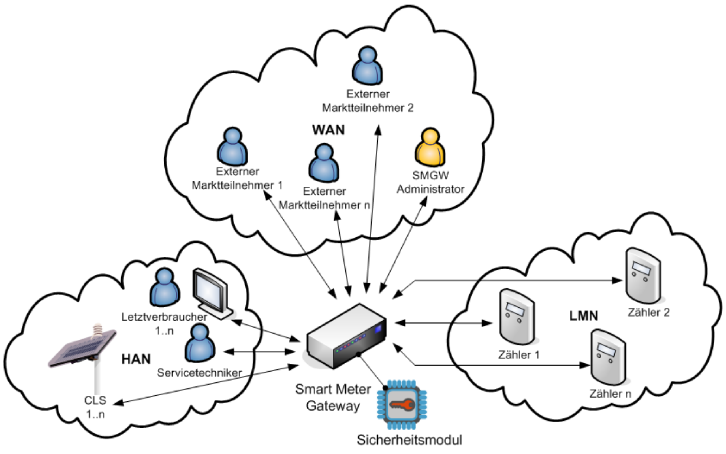

Abb. 1: Das SMGW und seine Schnittstellen [7]

Darüber hinaus verfügt das SMGW noch über eine spezifische Schnittstelle zu seinem Sicherheitsmodule (Security Module – SM), das es bei Verschlüsselung und anderen Sicherheitsaufgaben unterstützt.

Kommunikationsvorgaben im LMN

Für den Anschluss von sog. Basiszählern im Local Metrological Network spezifizieren die TR-03109 sowie die FNN-Lastenhefte weitgehend Protokoll-Stacks und physische Ausführungsformen. Unterschieden werden hierbei leitungsgebundene und per Funk angeschlossene (wireless) Zähler. Darüber hinaus müssen auch ältere elektrische sowie nicht elektrische Zähler mit SMGWs verbunden werden. Um auch hier eine sichere Übertragung gewährleisten zu können, sind sog. Kommunikationsadapter notwendig. Hier macht das FNN-Lastenheft „Kommunikationsadapter zur Anbindung von Messeinrichtungen an die LMN-Schnittstellen des Smart Meter Gateways“ [15] entsprechende Vorgaben. Die Vielfalt an „Altlasten“ macht den Anschluss an SMGWs nicht einfacher.

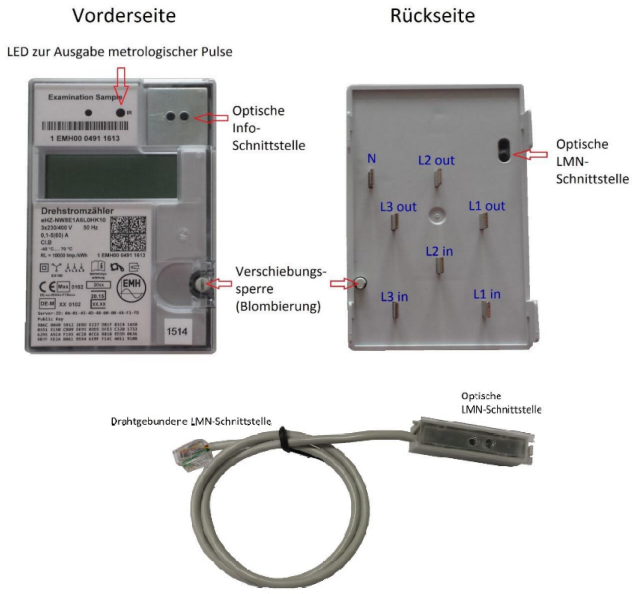

Abb. 2: Basiszähler in eHZ-Bauform und optischer Tastkopf [16]

Für die leitungsgebundenen Basiszähler (HZ: Haushaltszähler) sind zwei Bauformen vorgesehen [5]: der eHZ in Stecktechnik (s. Abb. 2) und der 3.HZ (mit Drei-Punkt-Befestigung, s. Abb.3). Da der eHZ nur einen optischen Ausgang auf der Rückseite hat, wird noch ein Wandler (OKK) auf die geforderte RS-485-Schnittstelle benötigt.

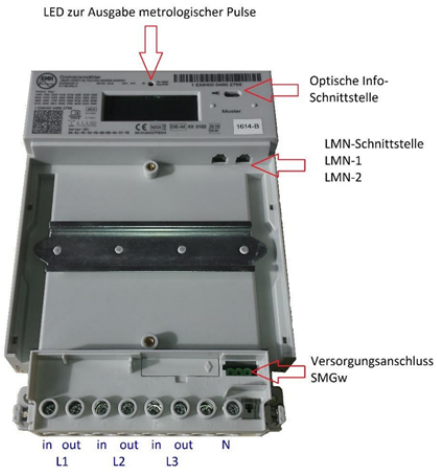

Abb. 3: Basiszähler mit Drei-Punkt-Befestigung (3.HZ) [16]

Für die Kommunikation im LMN sind die in Abb. 4 dargestellten Protokolle von BSI [7-11] bzw. FNN [3] vorgeschrieben. Dabei wird unterschieden zwischen der leitungsgebundenen Kommunikation (außen) und der Funkvariante „wireless M-Bus“

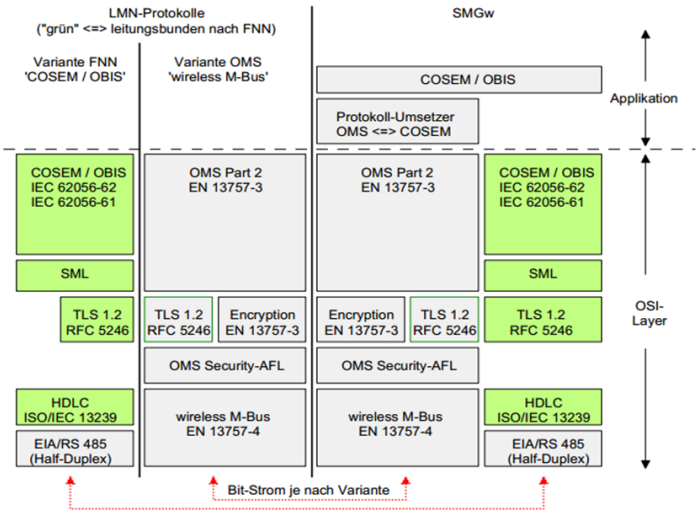

Abb. 4: Protokollstapel für die Kommunikation im LMN [3]

Auf oberster Ebene (vgl. auch Abb. 5) eines SMGW wählen OBIS-Zahlen (Object Identification System) die benötigte Information. Diese wer-

den auf vordefinierte COSEM-Klassen (Companion Specification for Energy Metering) abgebildet. Die darunter liegende Schicht verpackt die COSEM-Klassen in SML Files (Smart Message Language), die eingerahmt in Open- und Close-Nachrichten die entsprechenden Requests weiterreichen [11]. Diese verschlüsselt die TLS-Schicht [12] für den Schutz der Datenübertragung. Die HDLC-Ebene [10] verwaltet letztlich die über RS485 am Bus differenziell angeschlossenen Zähler. HDLC sendet regelmäßig Broadcasts, um neu angeschlossene Zähler zu erkennen und eine Adresse auszuhandeln, und fragt ebenso regelmäßig nach, ob Zähler aus dem Netz herausgenommen wurden („verstummte Zähler"). Für HDLC existieren spezifische Vorgaben für die Verwendung im LMN. So kommt nur der Frame Type 3 ohne Fragmentation zum Einsatz. Die Adressen sind immer zwei Byte groß, wovon das obere zur Adressierung der LMN-Teilnehmer dient, während das untere den sog. Protokoll-Selektor bildet (z.B. Übertragung mit oder ohne TLS etc.).

Auf der Empfangsseite (Zähler) werden die Protokollebenen in umgekehrter Reihenfolge durchlaufen. Die Signale auf dem RS485-Bus werden an die HDLC-Sicherungsschicht und weiter an TLS gereicht. Anschließend gelangen die entschlüsselten Anfragen zur SML-Ebene, die den Request in die entsprechende COSEM-Klasse und letztlich in die OBIS-Zahl umsetzt. Die eigentliche Zählersoftware erkennt, was angefragt ist und sendet den Wert (z.B. die Wirkarbeit) wieder durch den Protokoll-Stack zurück zum SMGW.

Damit die Verschlüsselung funktionieren kann, benötigen die Kommunikationsteilnehmer ihre jeweiligen Zertifikate und Schlüssel. Über eine symmetrische Verschlüsselung (SYM-über-HDLC, s. rechts in Abb. 5) erfolgt der notwendige Austausch. Der sogenannte Master Key des Zählers mit 128 Bit ist fest verbaut und muss bei der Inbetriebnahme des Zählers am SMGW dort sicher eingebracht und gespeichert werden.

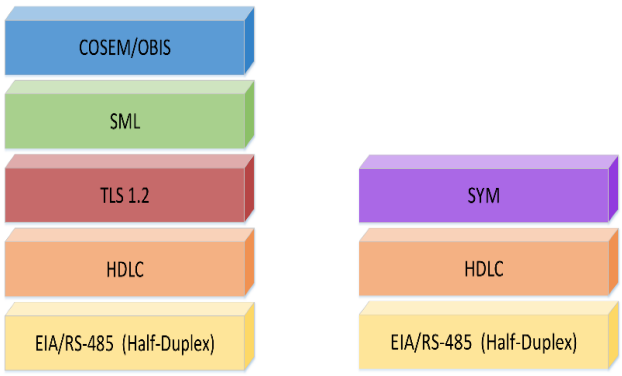

Abb. 5: Verwendete Protokollstapel im LMN [16]

Analyse der verfügbaren Spezifikationen

Im Rahmen des Projekts „Kommunikation im Local Metrological Network eines Smart Meter Gateway" wurden zunächst die entsprechenden Verordnungen, Richtlinien, Lastenhefte, Normen etc. gesichtet und detailliert analysiert. Dabei lag der Focus auf der leitungsgebundenen Kommunikation. Alle Angaben schienen verständlich und eindeutig umsetzbar. Wie sich aber im weiteren Verlauf, insbesondere bei Implementierung und Test herausstellte, kann man bestimmte Stellen der Spezifikationen verschieden interpretieren. So zeigten sich sowohl in unserer Realisierung als auch in Vergleichsuntersuchungen an Zählern verschiedener Hersteller, dass Unterschiede bestanden. Durch Gespräche und Diskussionen mit den Partnern und Herstellern konnten diese interpretationsbedürftigen Teile der Specs geklärt werden, so dass zurzeit alle uns zur Verfügung stehenden Zähler mit entsprechend aktualisierter Firmware mit unseren Implementationen fehlerfrei arbeiten. Ein erstes Vormuster eines kommerziellen Kommunikationsadapters, das von der PTB zur Verfügung gestellt wurde, arbeitet ebenfalls einwandfrei mit der Kommunikationssoftware zusammen.

Im Anschluss an die Analysephase folgte die Erstellung von Entwicklungsunterlagen wie Sequenzdiagrammen etc. Weiterhin wurden verwendbare Bibliotheken z.B. für die Ver-

schlüsselung identifiziert und untersucht, bevor eine erste Implementation begonnen wurde.

Exemplarische Implementierungen

Ein erster Implementationsansatz des LMN-Kommunikationsinterface eines SMGW in Java begann vielversprechend. Allerdings zeigte sich, dass die Realisierung der TLS-Verschlüsselung in Java abseits der Wege normaler Browser-Kommunikation (https) schwierig bis unmöglich ist. Es fanden sich keine geeigneten Bibliotheken, die die spezifischen Anforderungen im LMN erfüllten, bzw. die Anpassungen an Bibliotheken bzw. Programmsystemen hätten den Gesamtaufwand extrem erhöht. Dies machte eine Änderung des Konzeptes notwendig. So entstand eine erste Version, die in den unteren Schichten eine C/C++-Implementierung vorsah, für die oberen Ebenen die Java-Programme nutzte. Selbst hier ergaben sich Schwierigkeiten: Der prinzipiell mögliche Aufruf der C-Realisierung von Java aus (über das JNI) scheiterte daran, dass das C-Interface von Java nicht mit dem ständig laufenden C-Programm zurechtkam (u.a. Timeout-Probleme durch langsamen Verbindungsaufbau für TLS). Letztlich realisierte die erste lauffähige Version die Kommunikation zwischen den obersten Schichten in Java und dem darunterliegenden C-Programmen über Dateien [16]. Die Übertragung der benötigten Zertifikate und Schlüssel vom SMGW in die Zähler konnte vollständig in Java verwirklicht werden.

Die unbefriedigende Realisierungsvariante mit Java und C/C++ machte einen vollständigen Neuansatz nötig, wie bei der Vorgängerversion unter Linux. Da die erforderlichen Bibliotheken, insbesondere für Embedded Implementationen, im Wesentlichen nur für C/C++ verfügbar sind, kam nur eine Realisierung in dieser Sprache infrage. Passende Crypto-Bibliotheken (GnuTLS, OpenSSL, wolfSSL, mbed TLS) wurden eingehend auf ihre Eignung untersucht, wobei sich wolfSSL (ehemals CyaSSL, davor yaSSL) [19] als am besten geeignet und für Embedded Anwendungen optimiert erwies. Sie unterstützt u.a. die geforderten vier Cipher Suites und die dazugehörigen fünf Kurvenparameter. Außerdem war die verfügbare Dokumentation eine der besten der untersuchten Bibliotheken. Eine Schwierigkeit, die bei allen Bibliotheken auftritt, ist ihre Ausrichtung auf übliche Netzwerkverbindungen. Im Umfeld der LMN-Kommunikation muss es aber die Möglichkeit geben, die TLS-Funktionen zusammen mit den anderen Protokollebenen einzusetzen, unabhängig von typischen TCP/IP-Verbindungen. Dies erlauben bei wolfSSL sog. Callback-Funktionen.

In der HDLC-Schicht kam keine Bibliothek zum Einsatz, da seitens des FNN [3, 10] verschiedene Unterschiede zu Standard-HDLC vorgegeben waren. Für SML konnte die libsml der TU Berlin [20] verwendet werden, die allerdings im Wesentlichen Basisfunktionalitäten zur Verfügung stellt und somit das gesamte Paket-Handling mit den Library-Funktionen selbst aufgebaut werden muss.

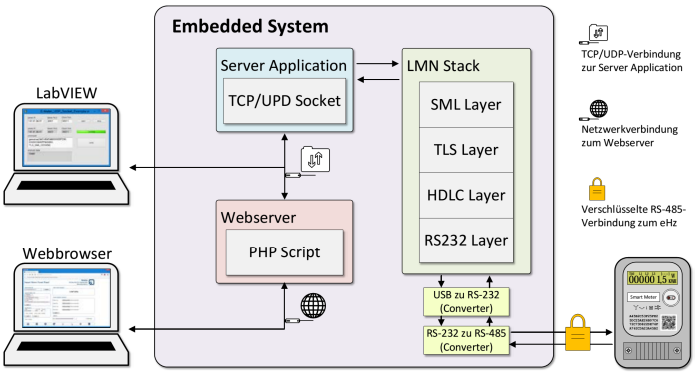

Abb. 6: Schnittstellen der Kommunikationssoftware [17]

Unter Verwendung der genannten Bibliotheken entstand die zweite, neu strukturierte Version des LMN-Interface eines SMGW [17]. Auch das Handling von Zertifikaten und Schlüsseln wurde passend integriert [18]. Um den Protokoll-Stack universell nutzen zu können (s. Abb. 6), stellt eine darauf aufbauende Server-Applikation eine einfach nutzbare Schnittstelle in Form eines Sockets zur Verfügung, über den per TCP oder

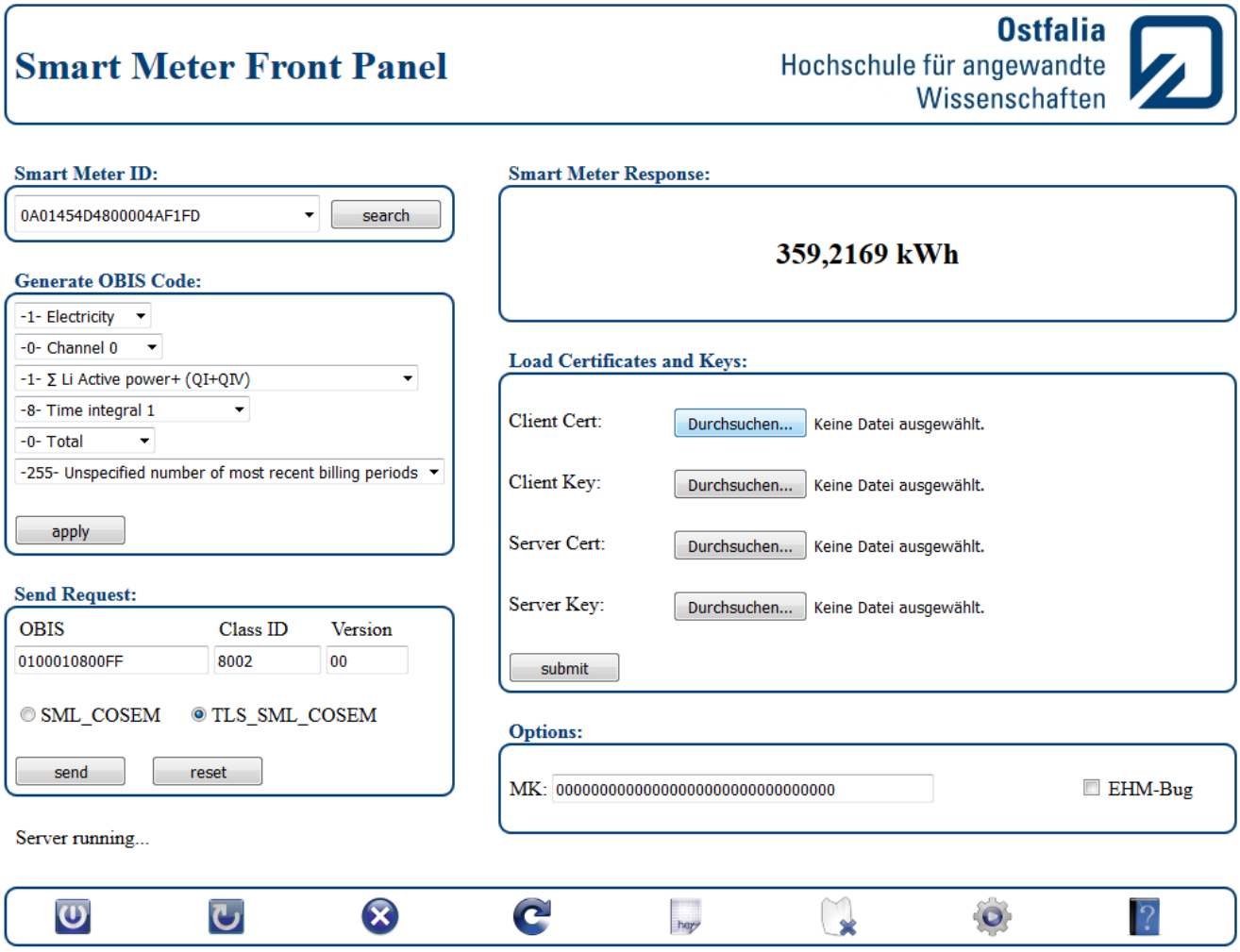

Abb. 7: Oberfläche des Webserver-Interface

UDP über den Stack mit angeschlossenen Zählern kommuniziert werden kann.

Die Funktionalität der Schnittstelle demonstrieren eine Implementierung in LabView sowie ein integrierter PHP-Webserver, der von jedem Browser aus angesprochen werden kann (s. Abb. 7). Um leicht von verschiedenen Arbeitsplätzen auf den Webserver zugreifen zu können, ist er zurzeit im Hochschulnetz (oder von außerhalb per VPN) unter http://... erreichbar. Weiterhin ist er im lokalen WLAN des Labors für Datentechnik verfügbar, so dass auch per WLAN mit verschiedensten Geräten auf ihn zugegriffen werden kann.

Die unterste Schicht des Protokoll-Stack, die mit 921.600 Baud arbeitet, wandelt nicht direkt nach RS485. Vielmehr wurde eine Implementierung nach RS232 über die USB-Schnittstelle gewählt, da das Betriebssystem dies direkt unterstützt. Zur Umsetzung auf RS485 entstand ein Adapter (s. Abb. 8, blau, zwischen Zähler und Prozessor-Board), der intern zuerst von USB auf RS232 und dann von RS232 auf das differenzielle RS485 wandelt. Daran kann nun ein Basiszähler angeschlossen werden. Um auch mehrere Zähler betreiben zu können, wurde noch ein (kaskadierbarer) Hub entwickelt, der bis zu vier Ausgänge bereitstellt.

Abb. 8 Messanordnung mit Raspberry Pi 2 und eHZ

Die gesamte Software sollte flexibel auf unterschiedlicher Hardware implementierbar sein. Aus diesem Grund entstand eine Version, die nur durch Änderung weniger Parameter jeweils eine passende Softwareausführung für x86-, ARMv6- und ARMv7-Architekturen generiert. Natürlich benötigt man dazu die entsprechenden Cross-compiler-Chains. Außer auf PC läuft das Programm also z.B. auf Raspberry Pi, auf Raspberry Pi 2 (s. Abb. 8) sowie dem Zybo Board (Digilent, Xilinx). Eine Anpassung auf ähnliche Boards oder auf andere Architekturen sollte mit wenig Aufwand möglich sein.

Weitere Arbeiten

An der Realisierung des LMN-Kommunikationsinterface hat auch der Kooperationspartner PTB großes Interesse, da er Basiszähler zulassen muss, er sie bislang aber bestenfalls mit herstellerspezifischen Tools prüfen konnte. Eine herstellerunabhängige Referenzimplementierung erlaubt universellere und potenziell aussagekräftigere Prüfungen. Die PTB unterstützte das Projekt mit Baumustern von Zählern sowie benötigten Spezialmessgeräten für Untersuchungen und Implementationstests. Die LMN-Kommunikationssoftware wird mittlerweile von der PTB eingesetzt sowie auch in Teilen für vergleichbare Aufgaben in der Messautomatisierung adaptiert benutzt. Auf Wunsch der PTB geschahen auch Anpassungen, die in der allgemeinen Verwendung so nicht vorgesehen, aber für Messaufgaben sinnvoll sind.

Im Rahmen des Projekts konnte auch mit Vorarbeiten und Entwicklung von sog. Kommunikationsadaptern begonnen werden. Diese sog. BAB (BSI konformer Adapter für Bestandszähler) erlauben es, Zähler nach alten Normen einzubinden und ihre Kommunikation mit SMGWs regelgerecht gemäß der TR-03109 und den FNN-Lastenheften durchzuführen. Bislang entstand eine Implementierung des kompletten zählerseitigen Protokoll-Stack [19]. Sie verfügt über eine universelle Softwareschnittstelle zum Anschluss von Altzählern. Einige Softwareteile, wie z.B. das Zertifikats-Handling stehen noch aus. Realisiert wurde sie ebenfalls auf einem Raspberry PI (s. Abb. 9). Die dedizierten, im Lastenheft [15] spezifizierten Schnittstellen (EDL-BAB, SyM2-BAB, diverse Schnittstellen für anzuschließende Gaszähler) wurden noch nicht angegangen, sollen aber zumindest teilweise (EDL-BAB) in der näheren Zukunft versuchsweise realisiert werden. Die bisherige Implementierung arbeitet problemlos mit der oben beschriebenen LMN-Kommunikationssoftware zusammen. Ein Test mit einem der ersten kommerziellen SMGW steht noch aus.

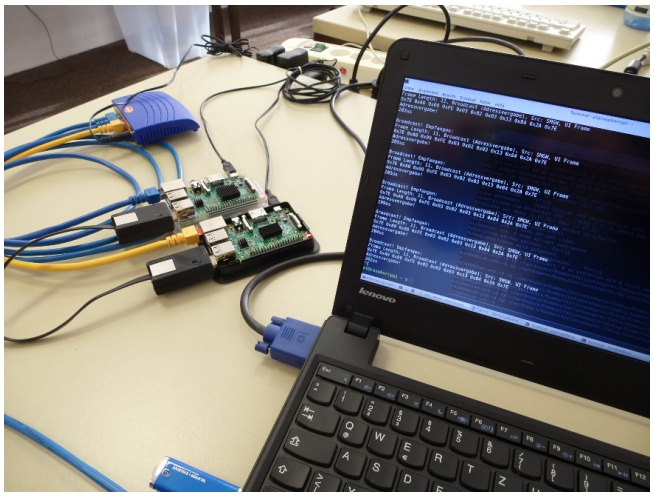

Abb. 9 Kommunikationsadapter am LMN-Kommunikationsinterface

Die PTB hat auch am Kommunikationsadapter großes Interesse, da hiermit ein Zählersimulator mit einstellbaren Parametern und Fehlerzustän-

den realisiert werden könnte, wie er zum Test und zur Zulassung von SMGWs benötigt wird.

Literatur

[1] VDE: *FNN-Projekt MessSystem 2020*. VDE – Verband der Elektrotechnik Elektronik Informationstechnik e.V., Frankfurt am Main. URL: ms2020.de

[2] FNN: *Lastenheft Smart-Meter-Gateway, Funktionale Merkmale – Version 1.0*. Forum Netztechnik / Netzbetrieb im VDE, Berlin, 2016.

[3] FNN: *Lastenheft Leitungsgebundene LMN-Protokolle – Version 1.1*. Forum Netztechnik / Netzbetrieb im VDE, Berlin, 2015.

[4] FNN: *Lastenheft Basiszähler, Funktionale Merkmale – Version 1.2*. Forum Netztechnik / Netzbetrieb im VDE, Berlin, 2015.

[5] FNN: *Lastenheft Konstruktion, Basiszähler und Smart-Meter-Gateway – Version 1.2*. Forum Netztechnik / Netzbetrieb im VDE, Berlin, 2015.

[6] BSI: *Technische Richtlinie BSI TR-03109*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2013.

[7] BSI: *Technische Richtlinie BSI TR-03109-1: Anforderungen an die Interoperabilität der Kommunikationseinheit eines intelligenten Messsystems*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2013.

[8] BSI: *Technische Richtlinie BSI TR-03109-1 Anlage III: Feinspezifikation „Drahtlose LMN-Schnittstelle“ Teil a: „OMS Specification Vol. 2, Primary Communication“*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2013.

[9] BSI: *Technische Richtlinie BSI TR-03109-1 Anlage III: Feinspezifikation „Drahtlose LMN-Schnittstelle“ Teil b: „OMS Technical Report Security“*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2013.

[10] BSI: *Technische Richtlinie BSI TR-03109-1 Anlage IV: Feinspezifikation „Drahtgebundene LMN-Schnittstelle“ Teil a: „HDLC für LMN“*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2013.

[11] BSI: *Technische Richtlinie BSI TR-03109-1 Anlage IV: Feinspezifikation „Drahtgebundene LMN-Schnittstelle“ Teil b: „SML – Smart Message Language“*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2013.

[12] BSI: *Technische Richtlinie BSI TR-03116-3: Kryptographische Vorgaben für Projekte der Bundesregierung Teil 3 - Intelligente Messsysteme*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, 2015.

[13] PTB: *PTB-A 50.8 – Smart Meter-Gateway*. Physikalisch-Technische Bundesanstalt, Braunschweig, 2014.

[14] PTB: *PTB-A 50.7 – Anforderungen an elektronische und softwaregesteuerte Messgeräte und Zusatzeinrichtungen für Elektrizität, Gas, Wasser und Wärme*. Physikalisch-Technische Bundesanstalt, Braunschweig, 2002.

[15] FNN: *Kommunikationsadapter zur Anbindung von Messeinrichtungen an die LMN-Schnittstellen des Smart Meter Gateways*. Forum Netztechnik / Netzbetrieb im VDE (in Zusammenarbeit mit dem DVGW – Deutscher Verein des Gas- und Wasserfachs), Berlin, 2015.

[16] Köhne, M.: *Charakterisierung des Zeitverhaltens der LMN-Schnittstelle bei Zählern nach FNN-Spezifikationen*. Bachelorarbeit. Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschaften, 2015.

[17] Lüthe, D.: *Entwicklung eines Embedded Systems zur Kommunikation mit einem Smart Meter gemäß BSI TR-03109 und FNN-Lastenheften*. Masterarbeit. Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschaften, 2016.

[18] Heine, K.: *Implementierung des initialen Zertifikatsaustausches mit einem FNN-Basiszähler über symmetrische Verschlüsselung*. Studienarbeit. Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschaften, 2015.

[19] Heine, K.: *Entwicklung eines Kommunikationsadapters zur Bedienung der LMN-Schnittstelle nach BSI TR-03109 und FNN-Lastenheften*. Bachelorarbeit. Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschaften, 2016.

[19] wolfSSL: *Embedded SSL Library*. Edmonds, USA, 2016. URL: https://wolfssl.com

[20] Glass, J., Runge, M., Sayed, N.: *libsml: Smart Message Language (SML) library for smart meter communication*. Publication 887, DAI-Labor. Berlin: Technische Universität, 2011.

Kontaktdaten

Ostfalia Hochschule für angewandte Wissenschaften
Fakultät Elektrotechnik
Prof. Dr.-Ing. Rainer Bermbach
Salzdahlumer Straße 46/48
38302 Wolfenbüttel
Telefon: +49 (0)5331 939 42620
E-Mail: r.bermbach@ostfalia.de
Internet: www.ostfalia.de/pws/bermbach