OPEN meter Open Public Extended Network metering		**WP:**	WP4
		Type of document:	Deliverable
		Date:	14/06/2011

Energy Theme; Grant Agreement No 226369

Partners:	Iberdrola Distribución, Itron, EDF, Elster, ENDESA, CTI, KEMA, Landis+Gyr, ENEL, RSE, Netbeheer Netherlands, ZIV Medida		
Responsible:	ENDESA and KEMA		
Circulation:	■ **Public** ☐ **Confidential** ☐ **Restricted**		
Title:	Report on Final Test Results and Recommendations	**Version: 1.0**	Page: 1 / 255

D4.4

REPORT ON FINAL TEST RESULTS AND RECOMMENDATIONS

Project Funded by the European Commission under the 7th Framework Programme

Project coordinated by IBERDROLA

Document History

Vers.	Issue Date	Content and changes
0.1	14.02.2011	First draft, including KEMA Test rapport on PRIME / DLMS Interoperability Tests of February.
0.2	13.04.2011	Second draft, including KEMA Test rapport on PRIME / DLMS Interoperability Tests of March.
0.3	04.05.2011	Third draft, including ENDESA Test report on METERS and MORE, RSE Test report on METERS and MORE, EDF test report on SFSK, IBERDROLA report on PRIME/DLMS and EDF test report on PLC G3 tests results, as well as an updated version of KEMA test results on PRIME DLMS.
0.4	24.05.2011	New draft including many of the comments received on the WP4 F2F meeting at Bilbao on the 12th of May 2011.
0.5	26.05.2011	Final draft with open topics included.
0.6	29.05.2011	Final remarks on 0.5 included. Chapter structure made consistent. Spelling check performed. Comment from telco processed.
0.7	14.06.2011	Recommendations to PRIME Standard as level 3 paragraph. Manufacturer names restored. Section about CEM/OEM and Interoperability added. Comment from TB members processed.
1.0	15.06.2011	First version finalized after TB review

Document Authors

Partners	Contributors
CURRENT	Andreas Brunschweiler
EDF	Aline Pajot, Gaizka Alberdi
Elster	Michael Arzberger
Endesa	Amador Gómez, David Martinez, Carmelo Rodríguez
Iberdrola	Iñigo Larumbe
Itron	Auguste Ankou
Landis + Gyr	Guillaume Timmins
KEMA	Maurice Adriaensen, Bas Roelofsen
RSE	Giuseppe Mauri, Paolo Gramatica
ZIV	Laura Marrón

Document Approvers

Partners	Approvers
CURRENT	Markus Bittner
ENDESA	Robert Denda / Carmelo Rodríguez
Iberdrola Distribución	Iñigo Berganza
KEMA	Jan Maarten van der Valk
Landis + Gyr	Thomas Schaub

TABLE OF CONTENTS

LIST OF FIGURES:

LIST OF TABLES:

REFERENCES

[1] OPEN Meter D4.1 Definition on Test Procedures v2.0

[2] OPEN Meter D4 3 Physical test facilities and report on these facilities v1.0

[3] OPEN Meter D4 1 Annex for SFSK v0.2

[4] IEC 62056 – 61 Electricity metering – Data exchange for meter reading, tariff and load control – Part 61: Object identification system (OBIS)

[5] IEC 62056 –62 Electricity metering – Data exchange for meter reading, tariff and load control – Part 62: Interface classes

[6] IEC 62056 –53 Electricity metering – Data exchange for meter reading, tariff and load control – Part 53: COSEM Application layer

[7] IEC 62056 –47 Electricity metering – Data exchange for meter reading, tariff and load control – Part 53: COSEM Application layer

[8] IEEE 802.15.4 physical layer and media access control for low-rate wireless personal area networks (LR-WPANs)

[9] RFC 4944 Transmission of IPv6 Packets over IEEE 802.15.4 Networks

[10] RFC 2460 Internet Protocol, Version 6 (IPv6) Specification

[11] RFC 768 User Datagram Protocol (UDP)

GLOSSARY, ACRONYMS & ABBREVIATONS

DC: Data Concentrator.

Dx : (OPEN Meter) Deliverable.

DLMS: Device Language Message Specification.

NT: Not Tested

OPEN meter: Open Public Extended Network metering.

PRIME: PoweRline Intelligent Metering Evolution.

S-FSK: Spread Frequency Shift Keying

TC: Test Case.

WPx: (OPEN meter) Work Package x.

INTRODUCTION

This document describes the tests performed and observations, conclusions and recommendations resulting from these tests. The tests have been performed both in laboratory and field environment.

The functional and interoperability tests as described in deliverable D4.1 (Definition of test procedures) for the Open Meter project selected technologies (METERS and MORE and G3, PRIME and SFSK with DLMS as application layer) were performed by engineers and audited by other WP4 members. The prototypes have been developed in WP 4.2 and are described in D4.3 (Physical test facilities and report on these facilities). These prototypes and test facilities have been used in WP 4.3 (lab testing and 4.4 field testing). The results from these tests are described in this document, D4.4.

The tests were audited by OPEN Meter partners on specific dates. The audits focused at verifying if the procedures and test cases as described in D4.1 were followed, if the prototypes and test facilities as developed in D4.2 were used and if observations and recommendations are correctly reflected in this report.

Equipment Manufacturers and Interoperability

This paragraph has the purpose to provide transparency in regards to the meaning of the word interoperability within this report. Therefore the distinction between Contracted Equipment Manufacturer (CEM) in contrast to Original Equipment Manufacturer (OEM) is introduced, as well as additional information and clarification on the ownership of the hardware and software designs of tested devices is provided. References to PLC modems fitted into the meters/prototypes are also given. These three types of information combined will provide the transparency and clarity that is needed in using the term interoperability.

In a CEM business model, the hiring firm approaches the manufacturer with a design or formula. The CEM will quote the parts based on processes, labor, tooling, and material costs. Typically a hiring firm will request quotes from multiple CEMs. After the bidding process is complete, the hiring firm will select a source, and then, for the agreed-upon price, the CEM acts as the hiring firm's factory, producing and shipping units of the design on its behalf.

An OEM is a company which designs and manufactures a product which is specified by another entity for sale. A primary attribute of this business model is that the OEM owns and/or designs the product in-house. This is in contrast to a CEM.

In the following sections, it is indicated for each of the involved prototype whether it is a OEM or CEM device. In case of OEM, the ownership of the firmware and the origin of the PLC modem is also mentioned.

This data is provided for the sake of transparency and will provide the reader with sufficient information to interpret the word 'interoperability' each time is it used in the following chapters of this document.

Equipment used at Meters&More tests:

1. Meter Manufacturer: ENEL. Meters are produced by KAIFA (CEM). Meter Hardware and Firmware design is owned by ENEL. PLC modem based on STMicroelectronics ST7581.
2. Meter Manufacturer: ENEL. Meters are produced by BITRON (CEM). Meter Hardware and Firmware design is owned by ENEL. PLC modem based on STMicroelectronics ST7581.
3. Data Concentrator Manufacturer: ENEL. Data concentrators are produced by KAIFA (CEM). Data concentrator Hardware and Firmware design is owned by ENEL. PLC modem based on STMicroelectronics ST7581.

Equipment used at S-FSK tests:

1. Meter Manufacturer: Landis+Gyr. Meters are produced by Landis+Gyr (OEM). Meter Hardware and Firmware design is owned by Landis+Gyr. PLC modem based on ON-Semiconductor AMIS-49587 C587 1.
2. Meter Manufacturer: Itron. Meters are produced by Itron (OEM). Meter Hardware and Firmware design is owned by Itron. PLC modem based on Texas Instruments TMS320F2802PZA-60.
3. Data Concentrator Manufacturer: Landis+Gyr. Data concentrators are produced by Landis+Gyr (OEM). Data concentrator Hardware and Firmware design is owned by Landis+Gyr. PLC modem based on ON-Semiconductor AMIS-49587 C587 1.
4. Data Concentrator Manufacturer: Itron. Data concentrators are produced by Itron (OEM). Data concentrator Hardware and Firmware design is owned by Itron. PLC modem based on Texas Instruments TMS320F2802PZA-60.

Equipment used at PRIME tests:

1. Meter manufacturer: Circutor. Meters are produced by Circutor (OEM). Meter Hardware and Firmware design is owned by Circutor. PLC modem based on STMicroelectronics ST7590.
2. Meter manufacturer: Elster. Meters are produced by Elster (OEM). Meter Hardware and Firmware design is owned by Elster. PLC modem based on Texas Instruments TMS320728.
3. Meter manufacturer: Itron. Meters are produced by Itron (OEM). Meter Hardware and Firmware design is owned by Itron. PLC modem based on Texas Instruments TMS320728.
4. Meter manufacturer: Landis+Gyr. Meters are produced by Landis+Gyr (OEM). Meter Hardware and Firmware design is owned by Landis+Gyr. PLC modem based on STMicroelectronics ST7590.
5. Meter manufacturer: Sagemcom. Meters are produced by Sagemcom (OEM). Meter Hardware and Firmware design is owned by Sagemcom. PLC modem based on Texas Instruments TMS320728.

6. Meter manufacturer: Sogecam. Meters are produced by Sogecam (OEM). Meter Hardware and Firmware design is owned by Sogecam. PLC modem based on ADDSemi ADD1021.
7. Meter manufacturer: ZIV. Meters are produced by ZIV (OEM). Meter Hardware and Firmware design is owned by ZIV. PLC modem based on general-purpose DSP, Analog Devices BF531.
8. Data Concentrator Manufacturer: Current. Data concentrators are produced by Current (OEM). Data concentrator Hardware and Firmware design is owned by Current. PLC modem based on STMicroelectronics ST7590.
9. Data Concentrator Manufacturer: ZIV. Data concentrators are produced by ZIV (OEM). Data concentrator HW and FW design is owned by ZIV. PLC based on general-purpose DSP, Analog Devices BF527.

Equipment used at G3-PLC tests:

1. Meter manufacturer: Sagemcom. Meters are produced by Sagemcom (OEM). Meter Hardware and Firmware design is owned by Sagemcom. PLC modem based on MAXIM MAX2992.
2. Data Concentrator Manufacturer: Sagemcom. Data concentrators are produced by Sagemcom (OEM). Data concentrator Hardware and Firmware design is owned by Sagemcom. PLC modem based on MAXIM MAX2992.

1 REPORT ON TEST RESULTS FOR METERS AND MORE

1.1 Introduction

1.1.1 METERS and MORE laboratory tests

METERS and MORE functional tests described in this report were performed on 8th to 10th March 2011 at 'RSE Lab 877', Milan, Italy.
All tests described in the report were executed in the presence of auditors.

The following table gives an overview of performed test cases and tested prototypes. Test cases are described in [1]. Prototypes are described in [2].

Test cases:	Described in sections of [1]:	Prototypes or devices:	Described in sections of [2]:
METERS and MORE Functional test	5.4.1 to 5.4.11; 5.4.13	KAIFA Meter	4.3.6.1
METERS and MORE Functional test	5.4.1.5; 5.4.2.1; 5.4.3.1	BITRON Meter	4.3.6.1

Table 1-1: Overview of OPEN Meter testing at RSE

Following persons were involved:

Test Engineers:	Auditors	Date in which auditors participated in the audit
G. Mauri, RSE	L. Marrón, ZIV	8th and 9th of March 2011
P. Gramatica, RSE	B. Roelofsen, KEMA	9th and 10th of March 2011
Support Engineers:	A. Gómez, ENDESA	From 8th to 10th of March 2011
V. Rizzo, ENEL		
M. Guttuso, ENEL		

Table 1-2: Overview of involved persons at Meters and More laboratory tests at RSE

Photos below show location and times during tests. User interfaces from PLC Tool were showed 'live' on the projector screen, so that witnesses got direct insight in test details.

Figure 1-1: Photo's of test setup at RSE

1.1.2 METERS and MORE field tests

All the field tests described in this report were performed at the premises of ENDESA in Seville, Spain, on 5th April 2011.
All tests described in this report were executed in the presence of auditors.

The following table gives an overview of the performed test cases and the tested devices.

Test cases are described in [1].
The prototypes are described in [2].

Test cases:	Described in sections of [1]:	Prototypes or devices:	Described in sections in [2]:
METERS AND MORE Field Tests	5.4.1; 5.4.2; 5.4.3; 5.4.4; 5.4.5; 5.4.6; 5.4.8; 5.4.9; 5.4.10; 5.4.11; 5.4.13	KAIFA METER	4.3.6.1
		ENEL DC	4.3.6.2

Table 1-3: Overview of OPEN Meter testing at ENDESA

Section 1.2.1 gives a summary of the test setup used during the tests. The results of these tests are described in section 1.2.2. In subclause 1.2.3 recommendations are given for the test setup, the test cases and the future standard.

The following people were involved:

Test Engineers:	**Auditors (5th April 2011)**
Amador Gómez, ENDESA	Elena Henríquez, KEMA
David Martínez, ENDESA	Giuseppe Mauri, RSE
Support Engineers:	Andreas Wohlrath, ELSTER
Manuel Tellechea, ENDESA	Vito Rizzo, ENEL
María José González, ENDESA	
Sergio López, ENDESA	

Table 1-4: Overview of involved persons at METERS and MORE field tests in Seville

1.1.2.1 Map of the test area

Figures below shows a map and a simplified scheme of the field test area:

Transformer Station/ Concentrator	
Customer's premise/ Meters	

Figure 1-2: Map of the test area

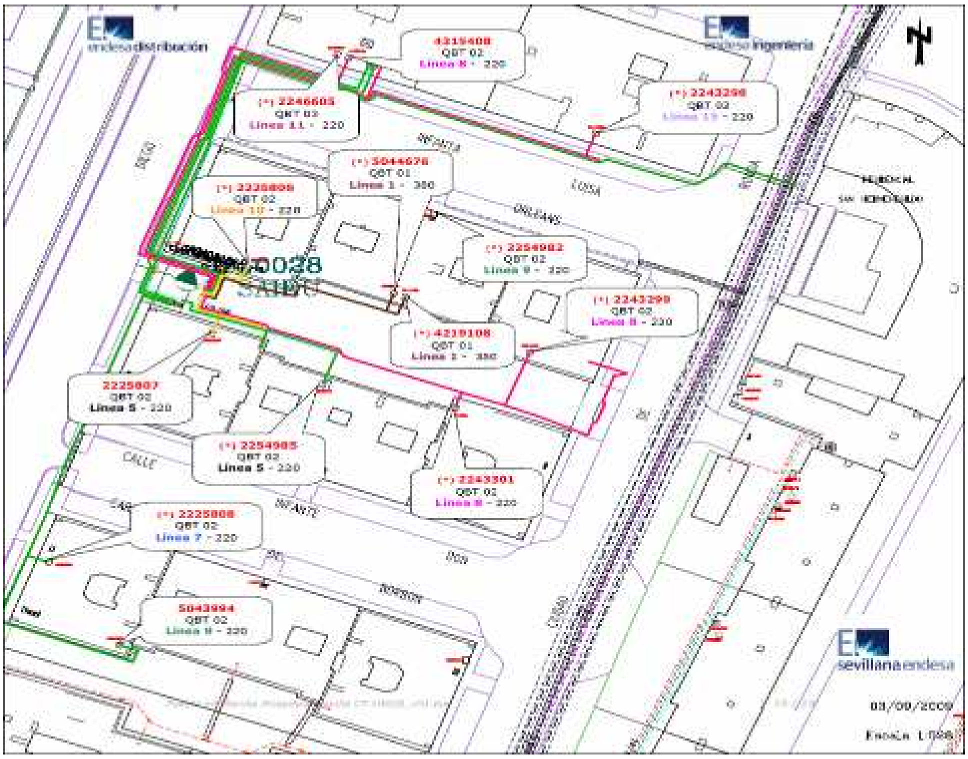


Figure 1-3: Simplified scheme of the test area

Figure 1-4: Some of the meters at consumer premises

1.2 Report on laboratory tests for METERS and MORE

1.2.1 Test setup for METERS and MORE Functional and Interoperability tests

The test facility used for the METERS and MORE Functional and Interoperability tests is described in [2], section 3.1.2. The location used was the Lab 877 with the test setup showed in Figure 1-5 and briefly described below.

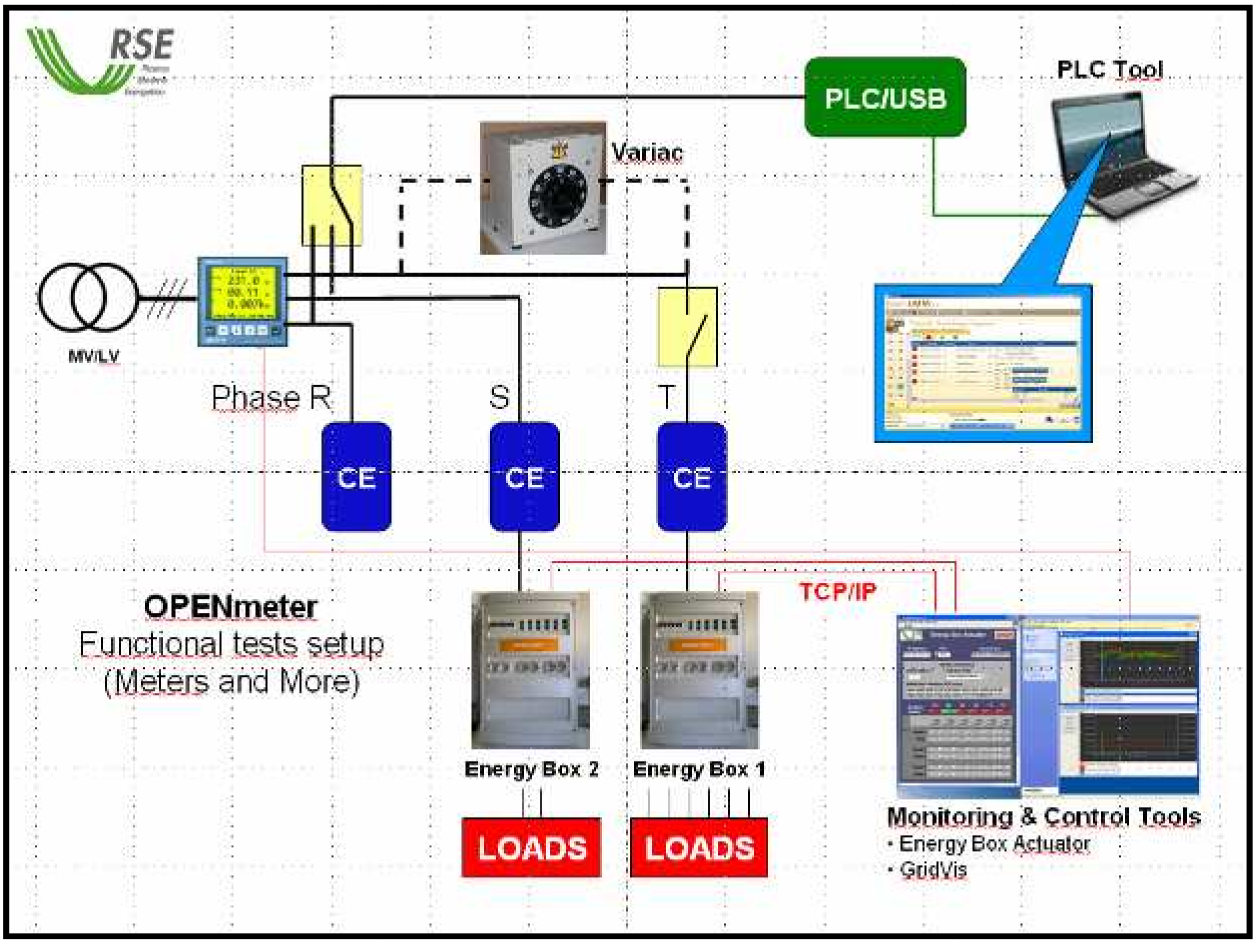


Figure 1-5: Tests Setup of laboratory tests

The Lab is powered directly from the **MV/LV Substation**. All Electrical parameters are measured by the **UMG 510 Janitza Universal Measuring Device** and showed on the **GridVis Monitoring Tool (Figure 1-6**) . Three **DUT (CE)** are connected respectively on the phases for this purpose:

- on phase R the CE for test case 5.4.7.1 Clock Synchronization

- on phase S the CE for test case 5.4.13.1 Load Profile Management
- on phase T the CE (after a breaker) for all other test cases. On this phase a **VARIAC**, for voltage variation, has been put in through for test case 5.4.10.1.

One **PLC/USB** box (simulating the concentrator) is connected on the phase for direct communication, using **Enel-LMM (PLC Tool)**, with the three CE.

Two **Energy Boxes** (**Figure 1-7**) are then connected after CE on phase S and T for the load selection during the test cases.

The Energy Box is a six channels solid state relay remotely controlled, via TCP/IP, by **Energy Box Actuator Tool** (**Figure 1-8**).

Figure 1-6: GridVis Monitoring Tool

Figure 1-7: Energy Box

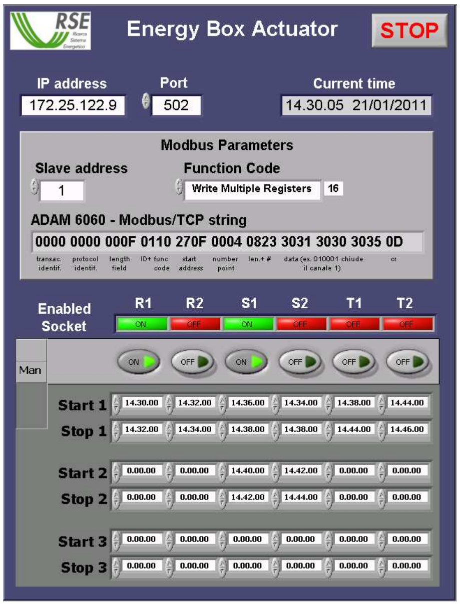


Figure 1-8: Energy Box Actuator Tool

1.2.2 Results of METERS and MORE Functional and Interoperability tests

Table below identify devices (CE) and test cases [1] performed on it:

Product 1	**E meter**
Supplier name	**KAIFA**
Test performed	**5.4.1.1 – 5.4.1.2 – 5.4.1.3 – 5.4.1.4 – 5.4.1.5 – 5.4.1.6**
	5.4.2.1
	5.4.3.1
	5.4.4.1
	5.4.5.1 – 5.4.5.2 – 5.4.5.3
	5.4.6.1 – 5.4.6.2
	5.4.8.1 – 5.4.8.2 – 5.4.8.3
	5.4.9.1
	5.4.10.1
	5.4.11.1
Product 2	**E meter**
Supplier name	**KAIFA**

Test performed	**5.4.7.1**
Product 3	**E meter**
Supplier name	**KAIFA**
Test performed	**5.4.13.1**
Product 4	**E meter**
Supplier name	**BITRON METER**
Test performed	**5.4.1.5**
	5.4.2.1
	5.4.3.1

A summary of executed tests is presented in ANNEX A. Some feedback, measurement from GridVis monitoring tool and screenshot from Enel-LMM tool are listed below. Functional Tests started at 10:30 on 8th March 2011 with the test case 5.4.13.1 on Product 3 followed by the test case 5.4.7.1 on Product 2 at 10.47 and all the other test cases performed on Product 1 at 13:44. On 10th March 2011 three test cases performed on Product 1 have been repeated on Product 4 (from different manufacturer).

TEST 5.4.13 - Load Profile Management

Case 5.4.13.1 Provide load profile - The goal of the test is to provide the load profile. In agreement with the partners attending, the step 4 of the test was modified to be short in time and with the load profile in 24 hours as follow:

- at 11:00 connect a load of 1.1kW for 60 minutes
- at 13:30 connect a load of 0.1kW for 90 minutes
- at 18:00 connect a load of 1.1kW for 60 minutes
- at 04:00 connect a load of 0.1kW for 90 minutes.

After 28 hour at 15:00 on 9th march 2011 the test was stopped. Result from the meter recorded profile (showed in **Figure 1-9**) can be succesfully compared with the data acquired by the GridVis monitoring tool in **Figure 1-10**.

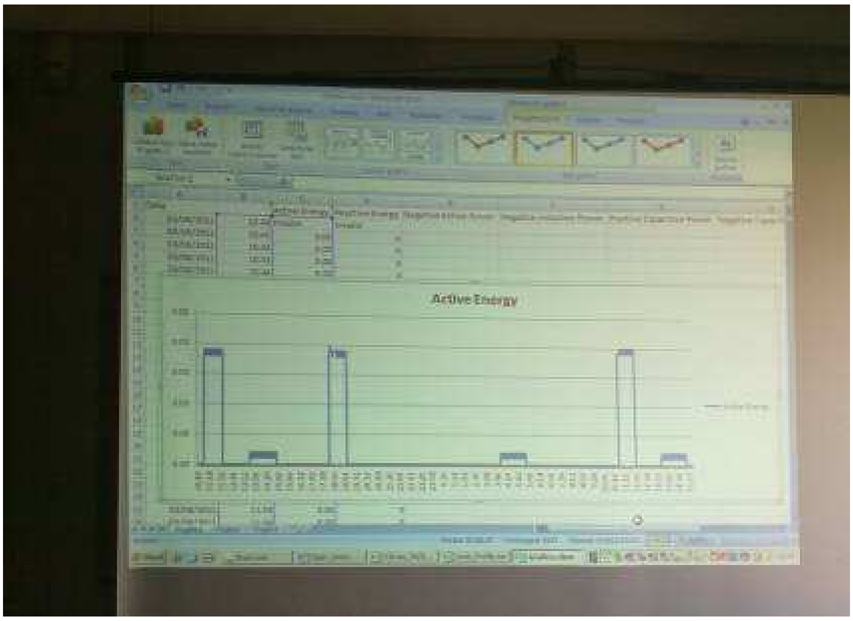


Figure 1-9: Load profile obtained from meter

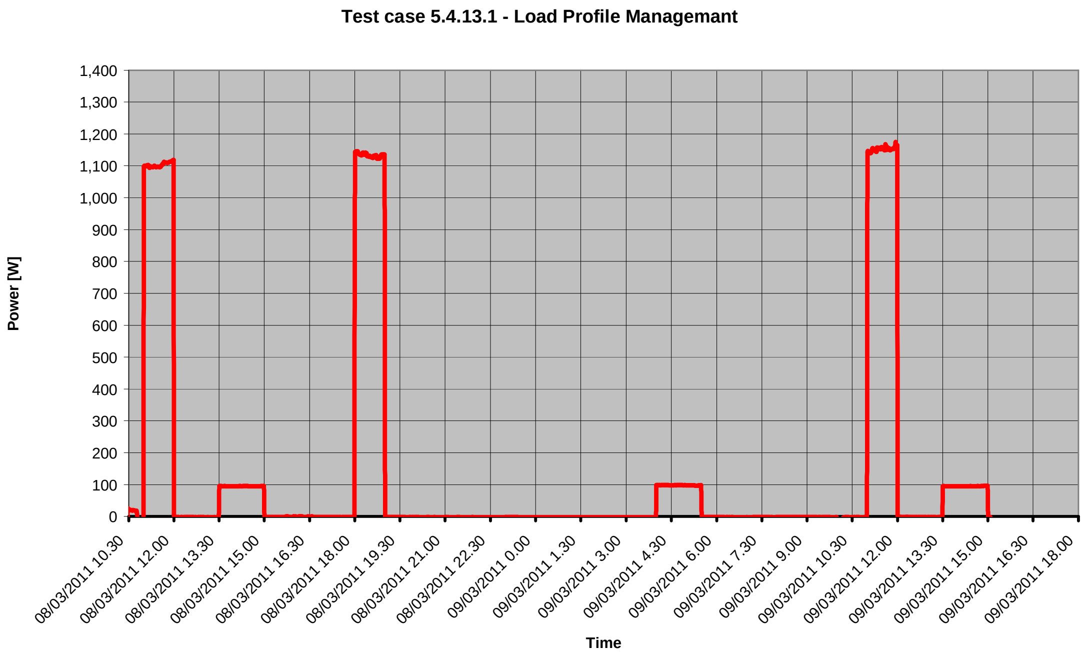


Figure 1-10: Load Profile acquired from GridVis

TEST 5.4.7 - Clock Syncronization

Case 5.4.7.1 Synchronization (Electricity-equipment) - The synchronization of Product 2 started on 10:47:10 on 8th March 2011 and stopped after the 25 hours on 9th March 2011. The synchronization between the E meter and the Computer simulating the concentrator was as expected in the range like Fit criterion conditions.

TEST 5.4.1 - Meter Registration

Case 5.4.1.1 Set up communication - The set up communication has been automatically established after the meter was physically installed. On the mask below we can see that the meter and the simulated Central System are connected.

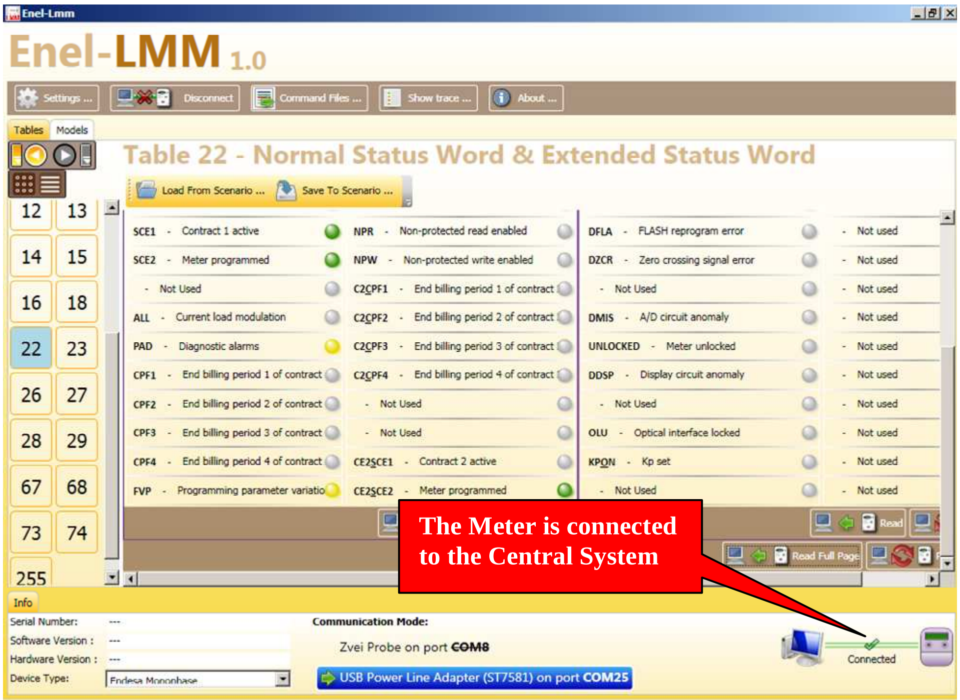


Figure 1-11: Set up communication

Case 5.4.1.2 Communication Check - For the communication check a reset command has been sent to the meter and the “FVP - Programming parameter variation” led turned from yellow (see screenshot before) to grey.

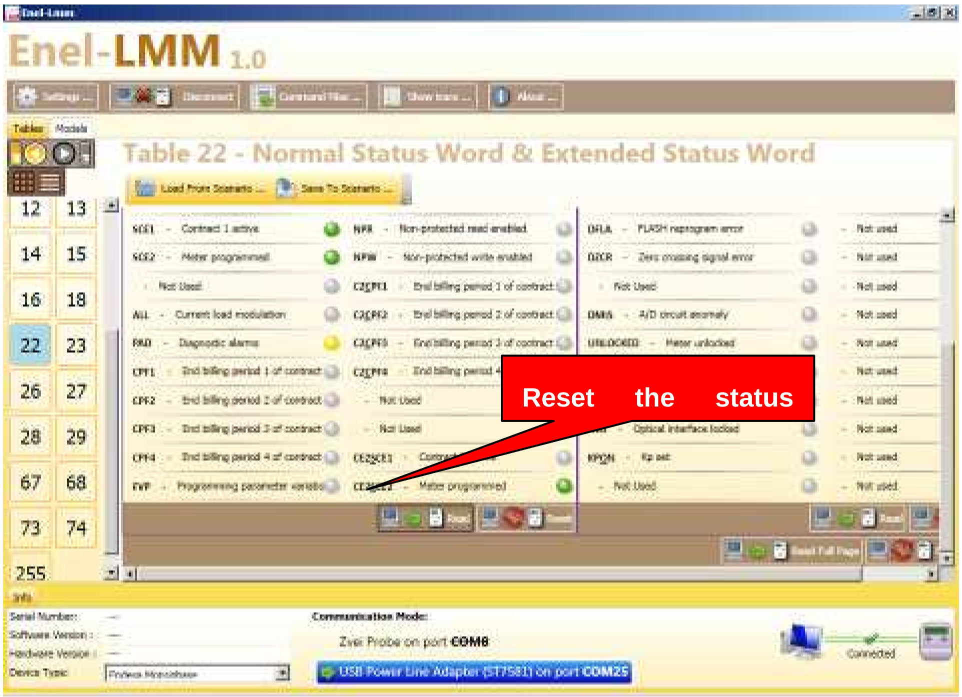


Figure 1-12: Communication check

Case 5.4.1.3 Install/Adjust configuration Electricity meter/Communication hub - The tariff that work all over the day (tariff T5 from 00:00 to 24:00 of Mon. to Fri.) has been configured.

Figure 1-13: Tariff configuration

Set, in the special day table, 7th October as Public Holidays

Figure 1-14: Special day configuration

Case 5.4.1.4 Perform self-check Electricity meter/Communication hub - No screenshot available

Case 5.4.1.5 Retrieve Electricity meter/Communication hub state - The screenshot below shows the threshold value parameter for the corresponding curve of the power control switch (step 2a)

Figure 1-15: Threshold parameters setup

Case 5.4.1.6 Un-install Electricity meter/Communication hub - This test is the replay of the previous one, it exists just because it is required by Open Meter documentation. According to the reviewers, this test was considered successfully done but has not been performed.

TEST 5.4.2 - Remote Tariff Programming

Case 5.4.2.1 Shift Tariff times electricity - The screenshot below showed a weekly two tariff T1 (from 00:00 to 12:00) and T2 (from 12:00 to 24:00) with T4 (fro 00:00 to 24:00) for holidays.

Figure 1-16: Two tariff configuration

TEST 5.4.3 - Meter Reading (on Demand)

Case 5.4.3.1 Provide on demand meter reads - Actual today daily energy values and yesterday values are showed below:

Figure 1-17: Meter readings

TEST 5.4.4 - Meter Reading (for Billing)

Case 5.4.4.1 Provide periodic meter reads - No screenshot available

TEST 5.4.5 - Remote Disconnection and Reconnection

Case 5.4.5.1 (Dis)connect Electricity

Figure 1-18: Logging of Open Cutoff Power Control

Case 5.4.5.2 and Case 5.4.5.3 - No screenshot available

TEST 5.4.6 - Power Control

Case 5.4.6.1 Apply threshold (electricity) - Unlike the one written in Deliverable 4.1 about test case 5.4.6.1 the I_n was reduced from 25 A to 5 A due to the high value not available in the lab. 877. The graphs below shows the thresholds load applied to the E meter with the corresponding steps and times captured by an electronic timer. The step 12 (well showed in the second graph) was repeated due to lower load level resulted in 52” instead of 51”; the second time a greater load was applied.

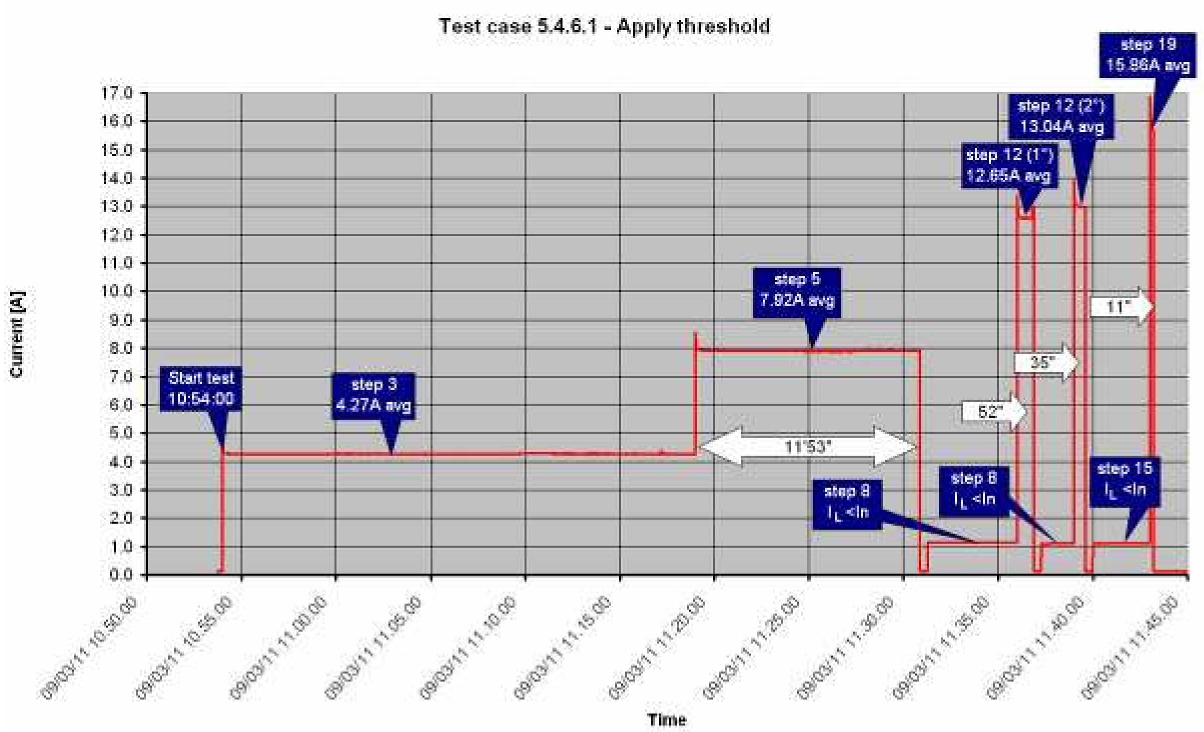


Figure 1-19: Load profile during power control test with step and time

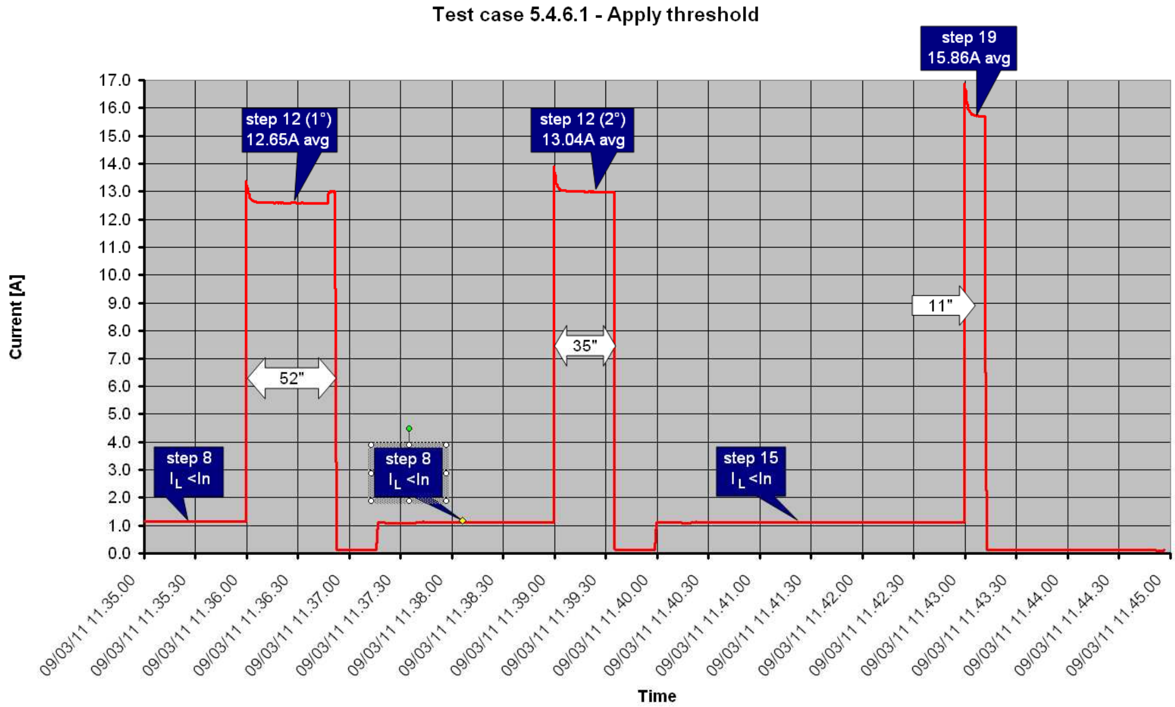


Figure 1-20: Zoom on the last steps of the power control test

Case 5.4.6.2 Changing the shedding factor 'K'

Figure 1-21: Configuration of the K factor

TEST 5.4.8 - Remote Firmware Update

Case 5.4.8.1 Activation at a defined time - The new version of Firmware can be downloaded to the meter with two options:

- Step 3

Figure 1-22: Logging of Firmware download

- Step 9

Figure 1-23: New Firmware activation at defined time

Case 5.4.8.2 Activation immediately

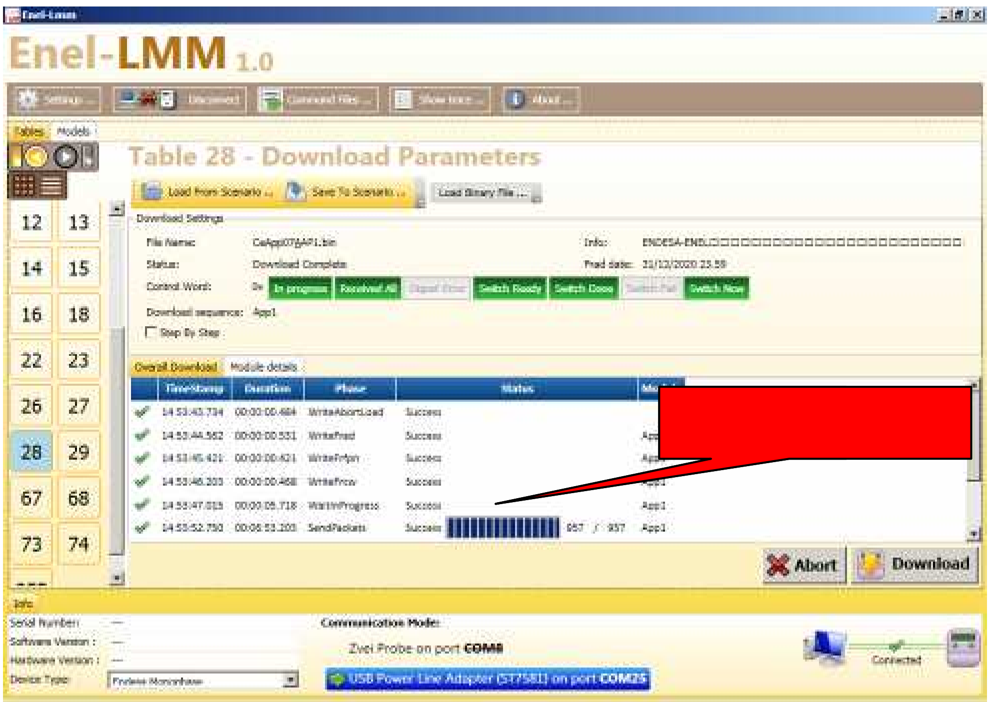


Figure 1-24: Firmware download succesfully

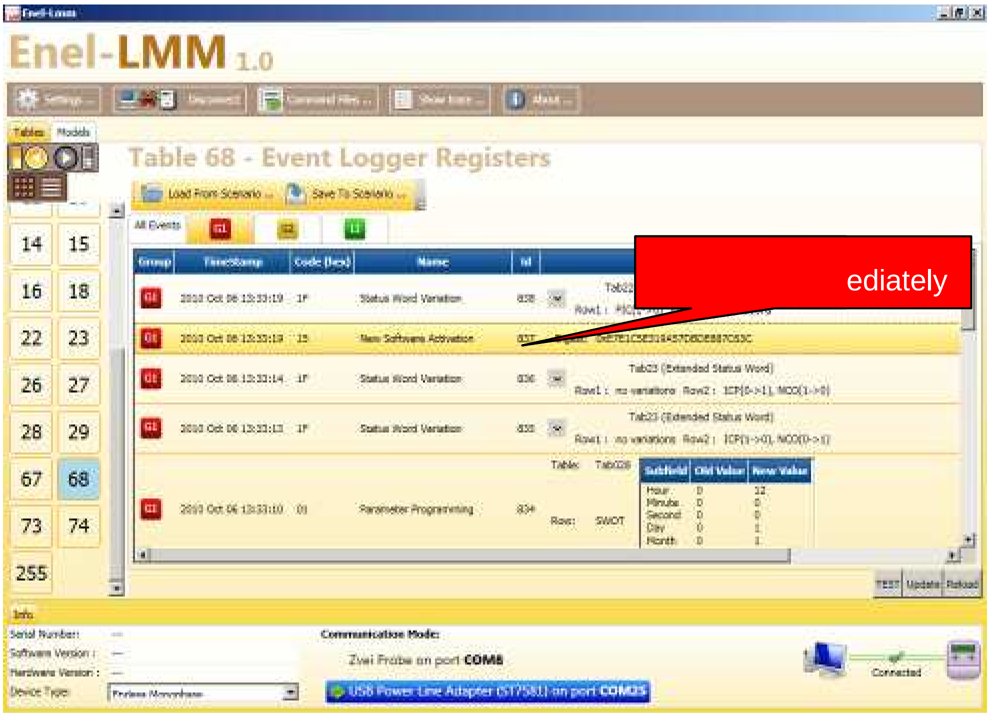


Figure 1-25: Firmware activation immediately

The Download aborted means that the the software has been discarded due to incomplete or inconsistent software *(case 5.4.8.3)*

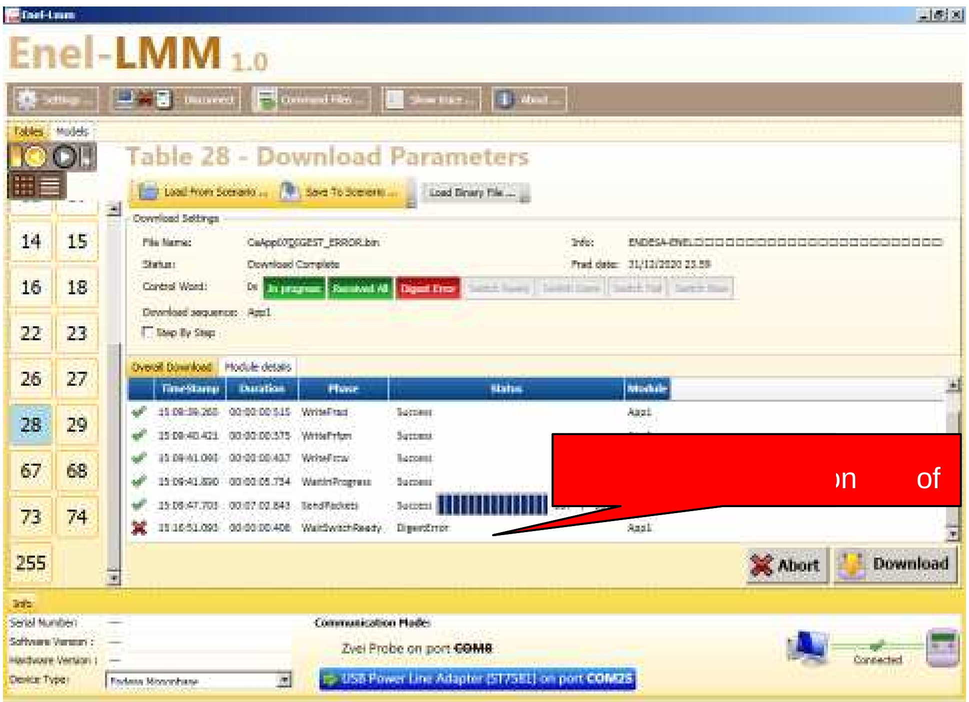


Figure 1-26: Corrupted Firmware download

Figure 1-27: Logging download aborted

TEST 5.4.9 - Alarm and Event Management

Case 5.4.9.1

Figure 1-28: Event logger registers

TEST 5.4.10 - Interruption Information

Case 5.4.10.1 Provide Interruption Information

Figure 1-29: Interruption parameter programming

Figure 1-30: Logging of interruption information

TEST 5.4.11 - Fraud Detection

Case 5.4.11.1 Provide Tamper History (tamper detection)

Figure 1-31: Logging of tamper detection

Repeater Test

For the purpose, the tools LMM can define a device as repeater and another as final meter or final destination. In the first case the BITRON device has been defined as repeater and KAIFA device as final meter. To execute this test, a sequence of a contract interruption and a reconnection command were sent to the KAIFA device with successful result. After that the BITRON device has been powered off. Repeating the command sequence described before the KAIFA device was not reachable. The Repeater error message appeared in the tool.

Figure 1-32: Logging of communication with KAIFA device

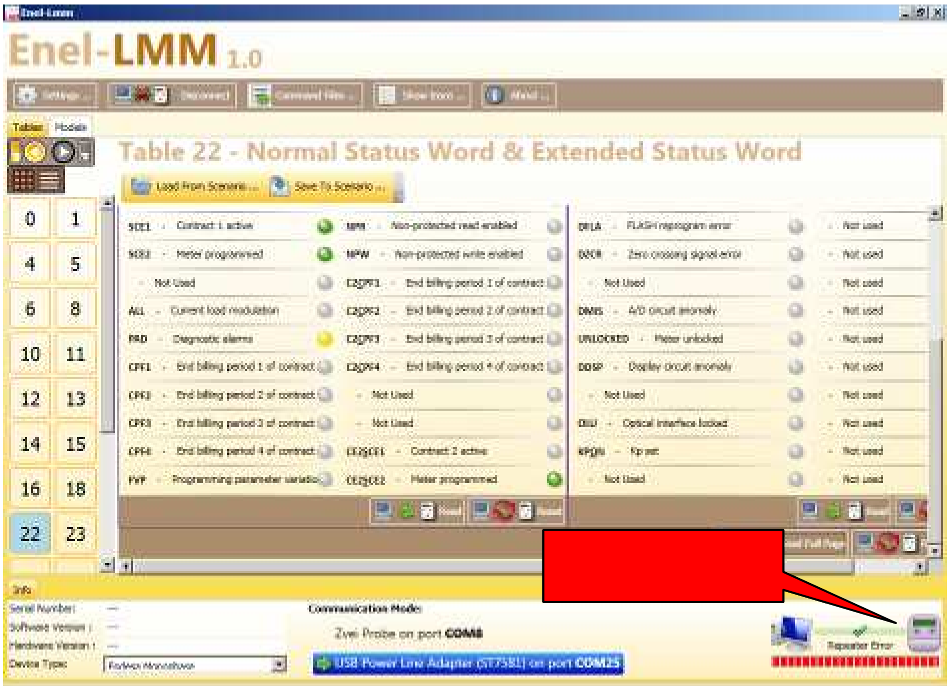


Figure 1-33: Repeater error due to BITRON METER powered off

BITRON METER Test

On the BITRON device was repeated test cases 5.4.1.5, 5.4.2.1 and 5.4.3.1 performed on KAIFA device (product 1) with successful results. Due to the results obtained and according to reviewers tests done on BITRON were enough to consider tests 6.4.4.1 and 6.4.4.2 PASSED.

1.3 Report on field tests for METERS and MORE

1.3.1 Test setup for Meters and More field tests

The test setup used for Meters and More field tests is shown below:

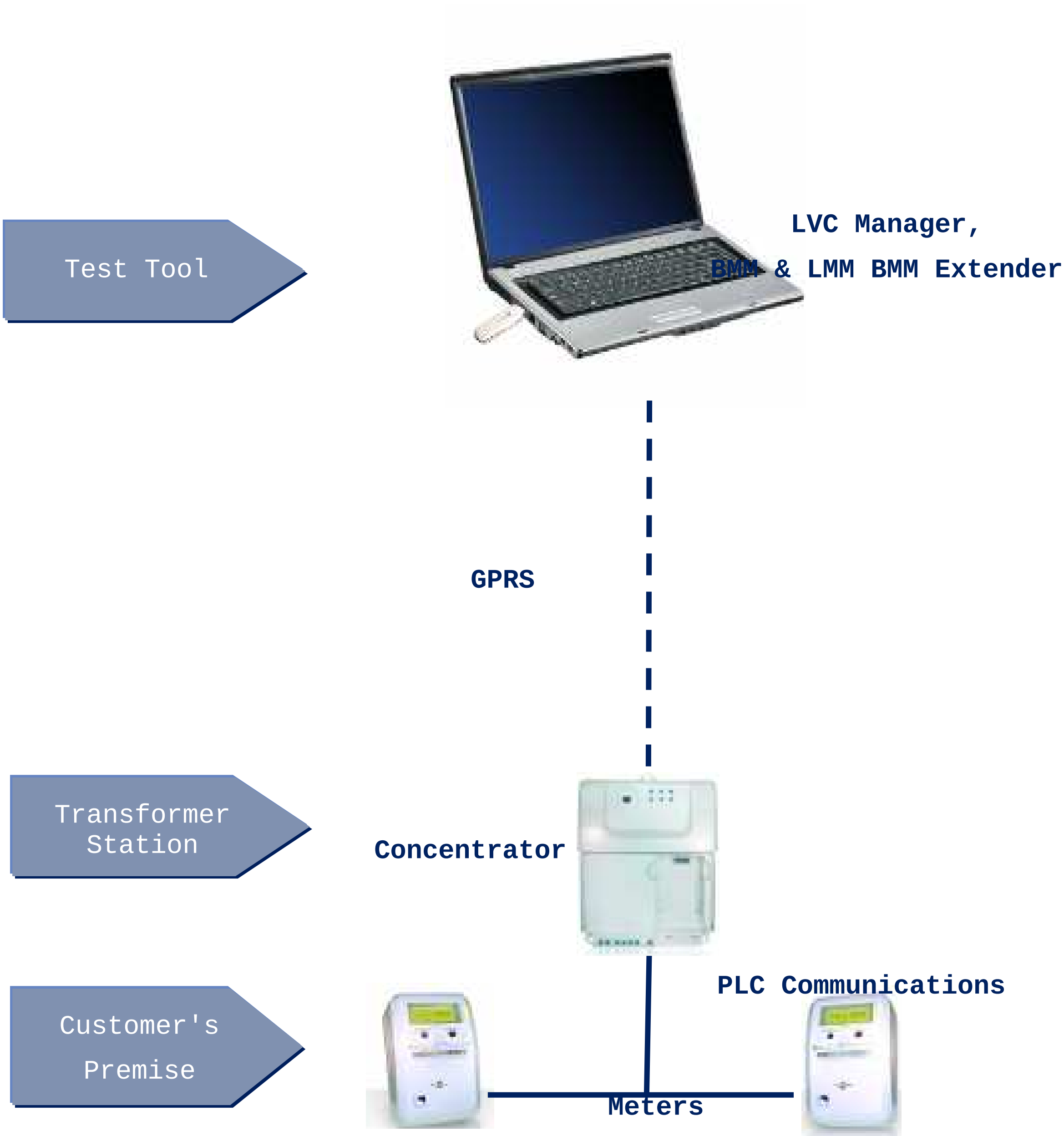


Figure 1-34: Test setup for Meters and More field tests

Meters located at the customer's premise communicate with a concentrator located at the transformer station through PLC communications. Then the concentrator communicates with a computer (that simulates the central system) using GPRS communications.

1.3.1.1 Test tools

The following test tools have been used for the Meters and More field tests:
LVC Manager: test tool that simulates the central system and has been developed by Endesa.

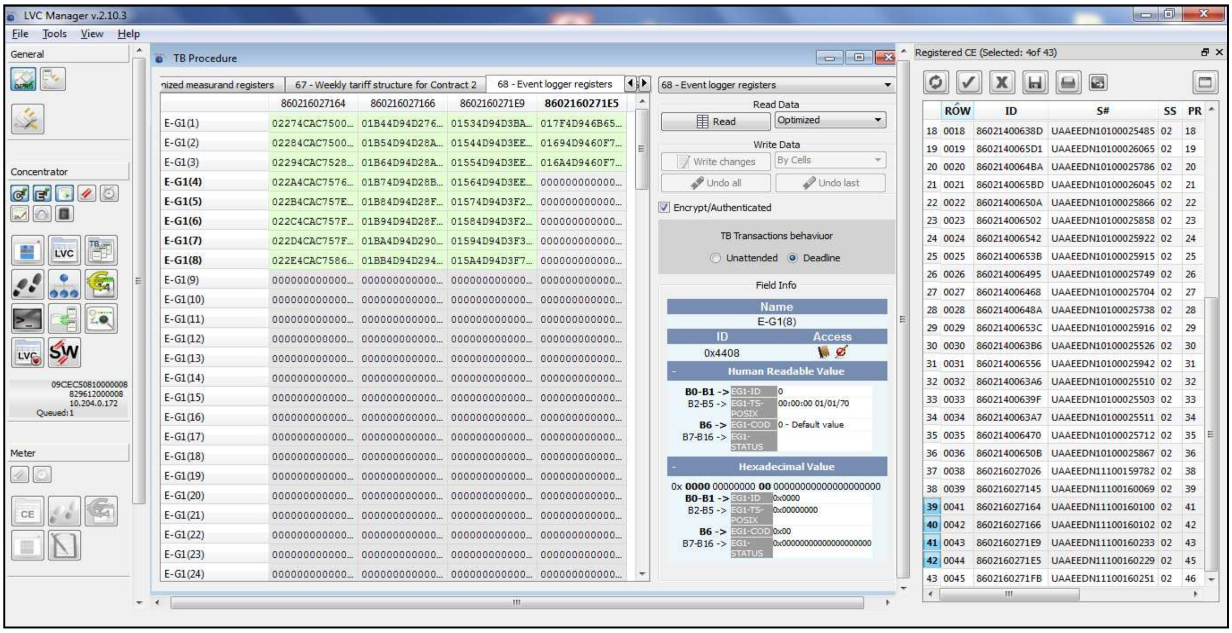


Figure 1-35: Example of registers readings with LVC Manager

BMM and LMM BMM Extender: test tools that simulate the central system and have been developed by Enel.

Figure 1-36: Example of load profile reading with LMM

1.3.1.2 Devices used in Meters and More functional tests

Details on software versions and device capabilities of the tested devices can be found below.

Tables below identify devices (CE) and test cases [1] performed on it:

Product 1	**E meter**
Supplier name	**KAIFA**
Serial number	**UAAEEDN11100160100**
List of test performed	• **5.4.1.2** • **5.4.1.3** • **5.4.1.4** • **5.4.1.5** • **5.4.2.1** • **5.4.3.1** • **5.4.4.1** • **5.4.5.1** • **5.4.5.2** • **5.4.5.3** • **5.4.6.1** • **5.4.6.2**

	• **5.4.8.2** • **5.4.8.3** • **5.4.9.1** • **5.4.11.1**

Table 1-5: Meter UAAEEDN11100160100

Product 2	**E meter**
Supplier name	**KAIFA**
Serial number	**UAAEEDN10100024068**
List of test performed	• **5.4.8.3**

Table 1-6: Meter UAAEEDN10100024068

Product 3	**E meter**
Supplier name	**KAIFA**
Serial number	**UAAEEDN10100003118**
List of test performed	• **5.4.10.1**

Table 1-7: Meter UAAEEDN10100003118

Product 4	**E meter**
Supplier name	**KAIFA**
Serial number	**UAAEEDN10100024076**
List of test performed	• **5.4.10.1**

Table 1-8: Meter UAAEEDN10100024076

Product 5	**E meter**

Supplier name	**KAIFA**
Serial number	**UAAEEDN10100024258**
List of test performed	• **5.4.13.1**

Table 1-9: Meter UAAEEDN10100024258

Product 6	**Data concentrator (DC)**
Supplier name	**ENEL**
Serial number	**829612000008**
List of test performed	• **5.4.1.2** • **5.4.1.3** • **5.4.1.4** • **5.4.1.5** • **5.4.2.1** • **5.4.3.1** • **5.4.4.1** • **5.4.5.1** • **5.4.5.2** • **5.4.5.3** • **5.4.6.1** • **5.4.6.2** • **5.4.8.2** • **5.4.10.1** • **5.4.13.1**

Table 1-10: DC 829612000008

Product 6	**Data concentrator (DC)**
Supplier name	**ENEL**
Serial number	**829614000116**
List of test performed	• **5.4.8.3**

Table 1-11: DC 829614000116

1.3.2 Results of METERS and MORE field tests

A summary of executed tests is presented in ANNEX B.

1.3.2.1 Results for OM-SR1 Meter registration

TC 5.4.1.2

The test has been performed successfully. In this test the communication with the meter has been checked. The central system (simulated using a test tool) sends a command to reset the meter status word. When the reset is executed, the information is registered on the event log. So, the electricity meter provides functionality to respond to a communication test initiated by the central system. Figures below show information about how the test has been performed and the results.

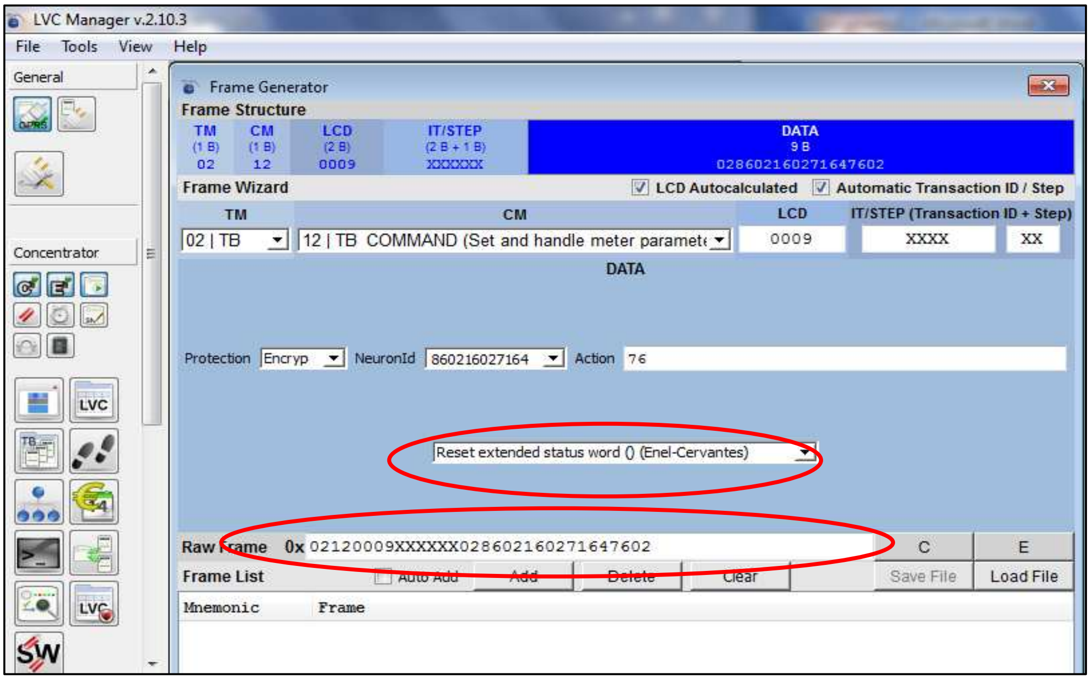


Figure 1-37: A message is sent to reset the meter status word

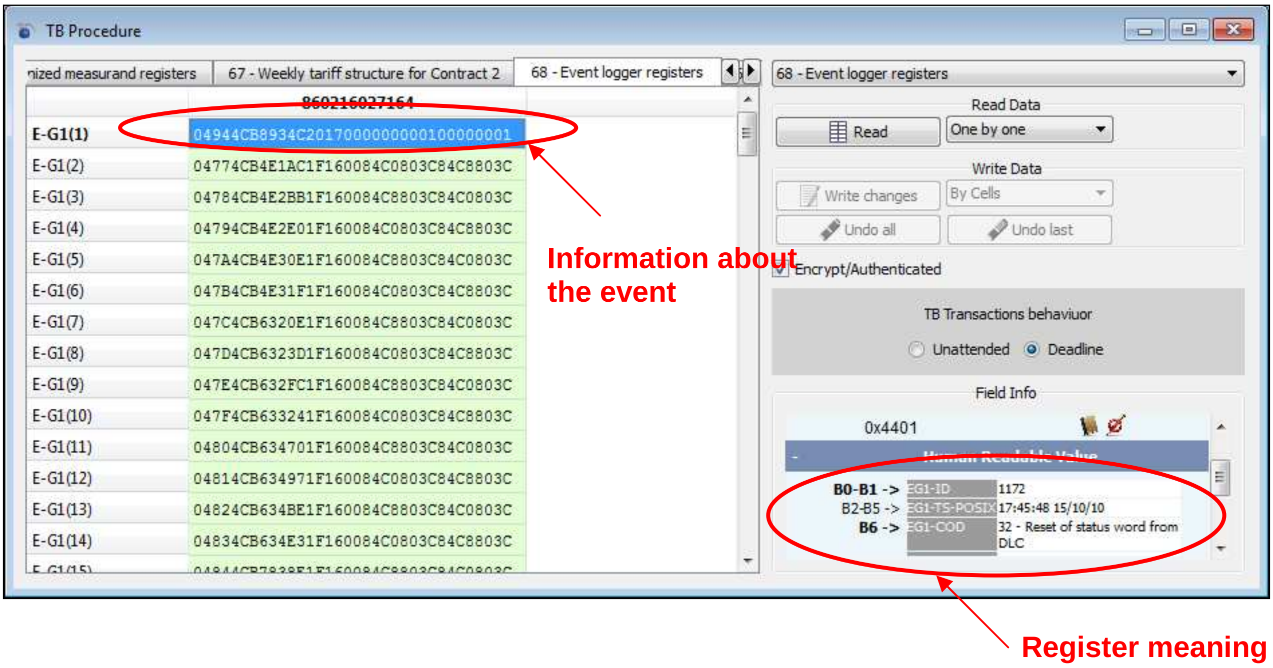


Figure 1-38: Event log table. The event shows test has been performed successfully

TC 5.4.1.3

Test has been performed successfully. During this test case several functionalities have been checked: meter clock and calendar, setting a new contract, meter synchronization, adjusting to daylight savings time and back, setting the tariff structure, public holidays programming.

Figure below shows a new contract in disconnectable mode.

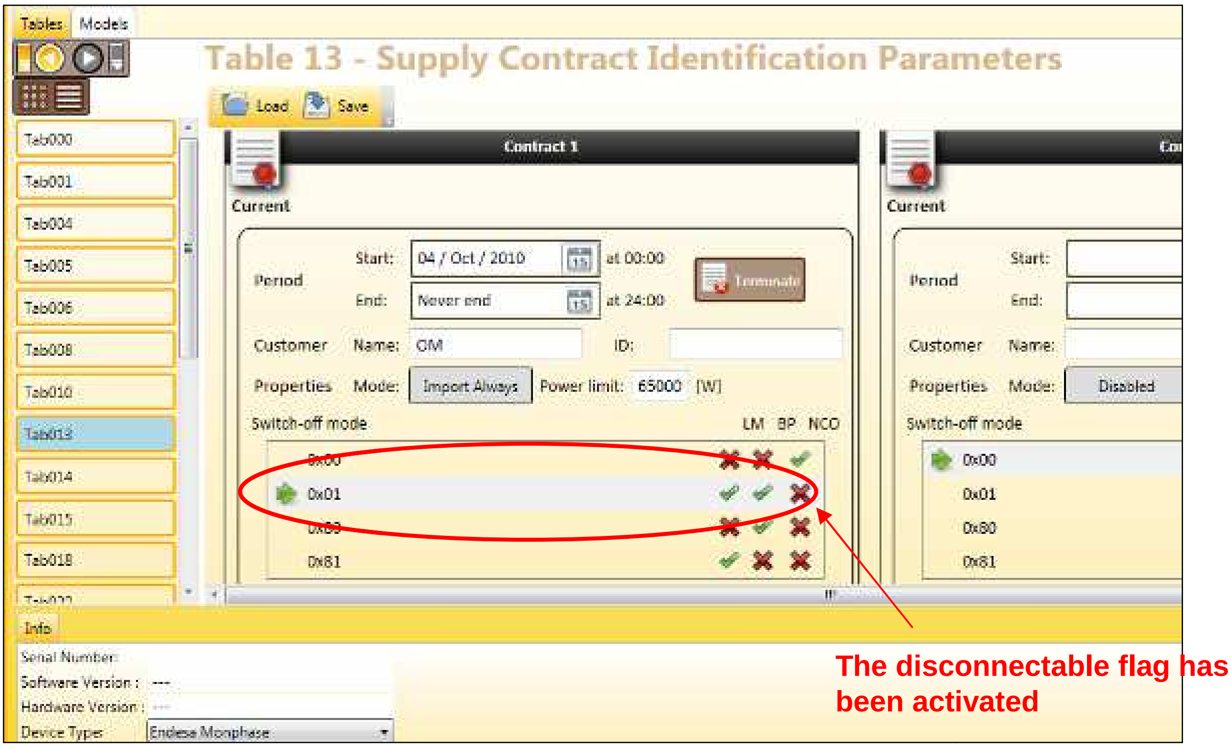


Figure 1-39: New contract setting

Figure below shows it is possible to synchronize the meter.

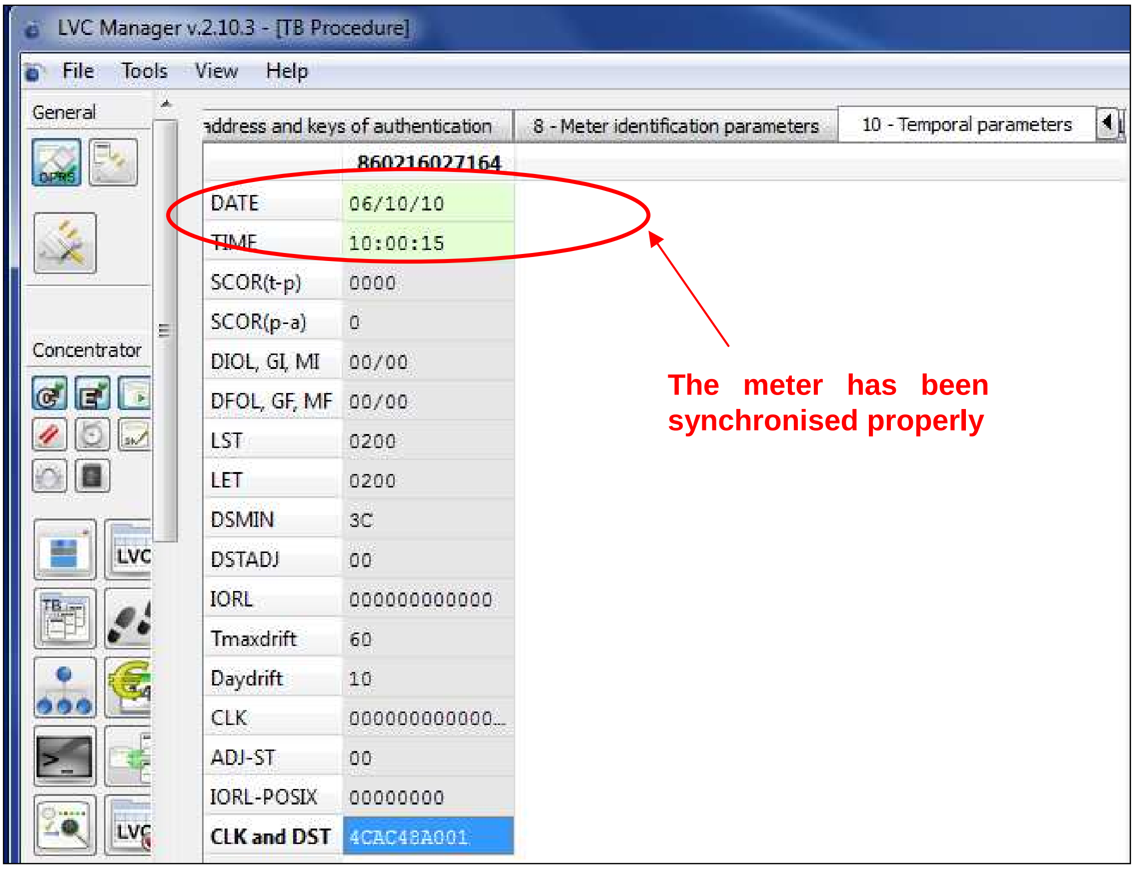


Figure 1-40: Meter synchronization

Then, it has been verified the meter adjusts automatically to daylight savings time and back. The meter was synchronized at 02:58:00 on the 31/10/10 and two minutes later it changed into 02:00:00 on the 31/10/10.

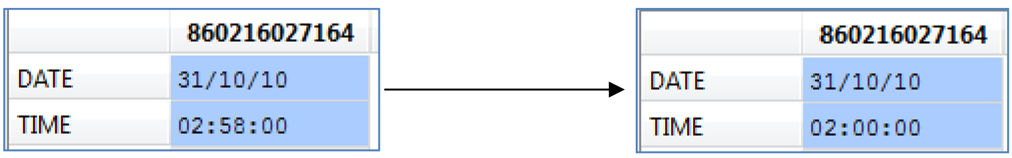


Figure 1-41: Figure shows the meter is able to adjust to daylight savings time and back

Tariff T1 works the whole day and all the days of the week:

Figure 1-42: Weekly tariff structure

Below, it is shown the table that contains the days that will be considered special days.

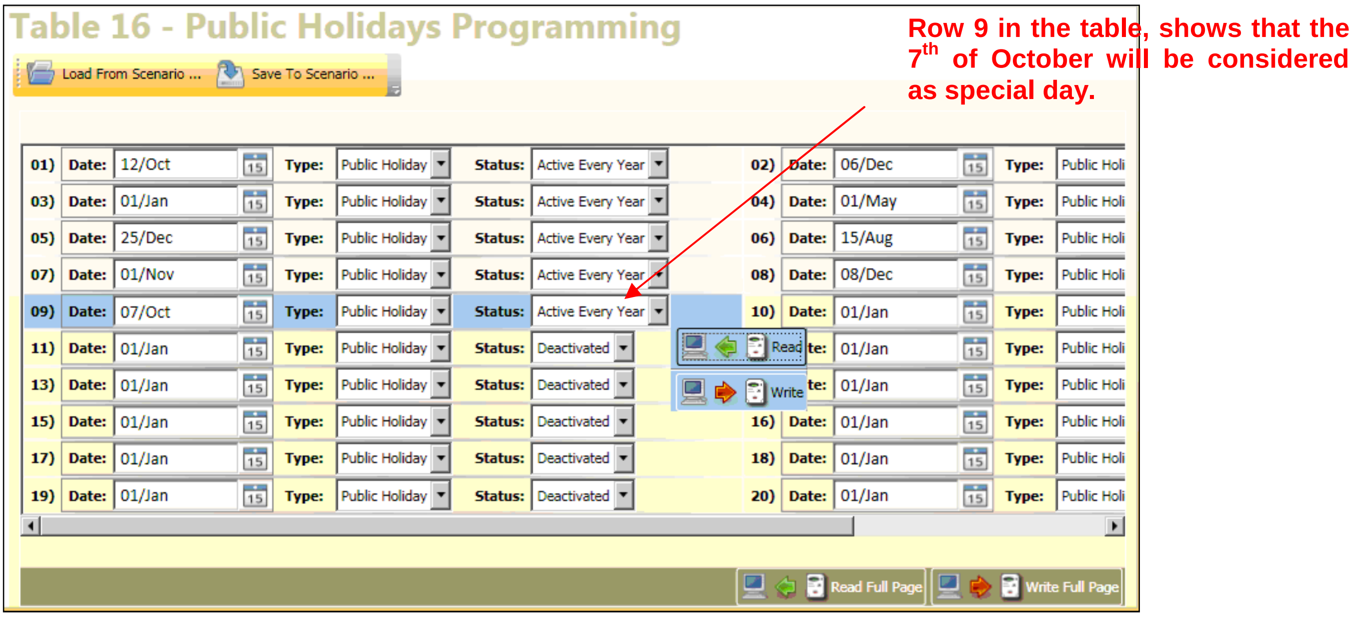


Figure 1-43: Public holidays programming

TC 5.4.1.4

Periodically the meter executes a self-check, the meter issues an error when detects any inconsistency in the memory or in the display. Events log table and status word of meter have been checked to verify there is no error.

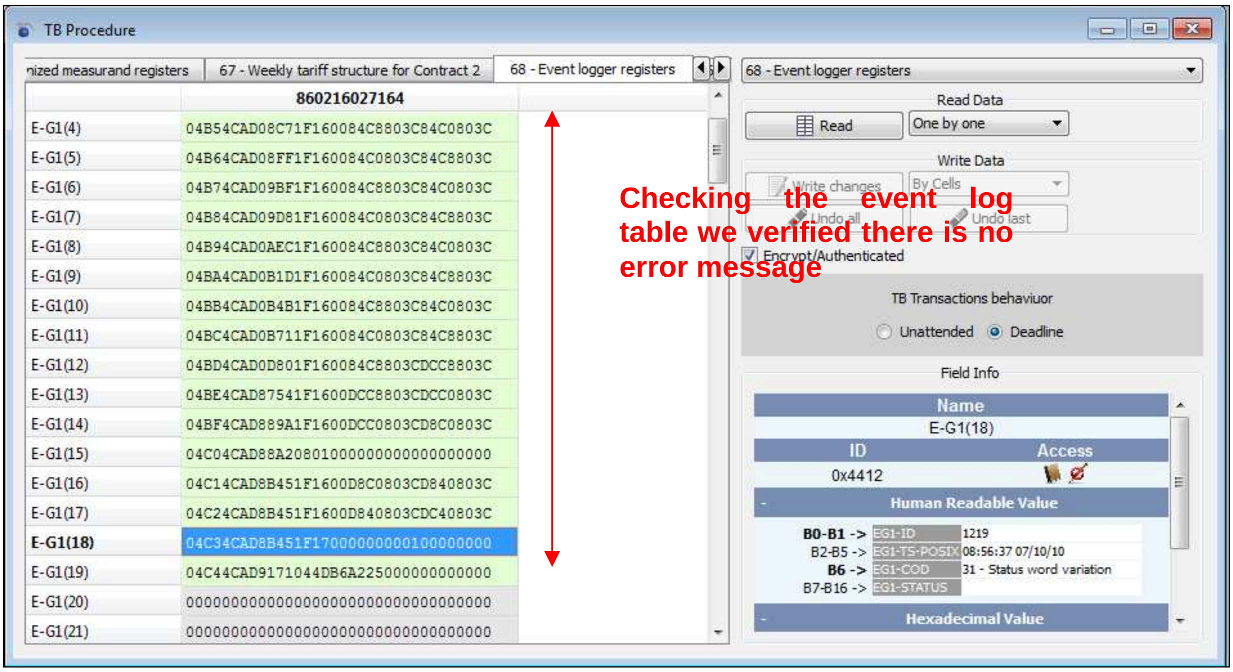


Figure 1-44: Events log table. Meter self-check

All the bits related to display errors and memory errors are zero, which means there is no inconsistency: DDSP (display), DRAM, DEEP,DFLA (memory)

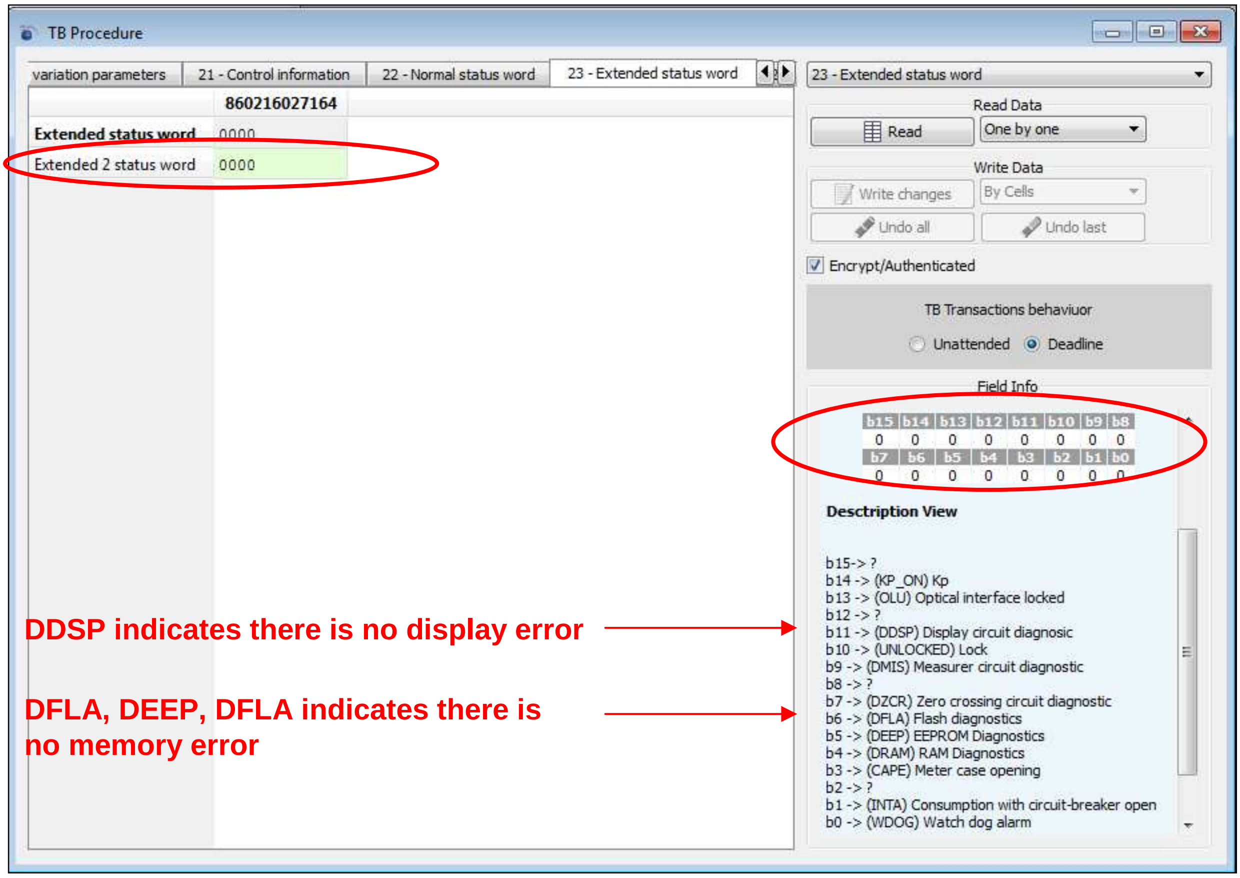


Figure 1-45: Status word table

TC 5.4.1.5

The test has been performed successfully. The meter state has been retrieved, examples of the registers readings are shown bellow:

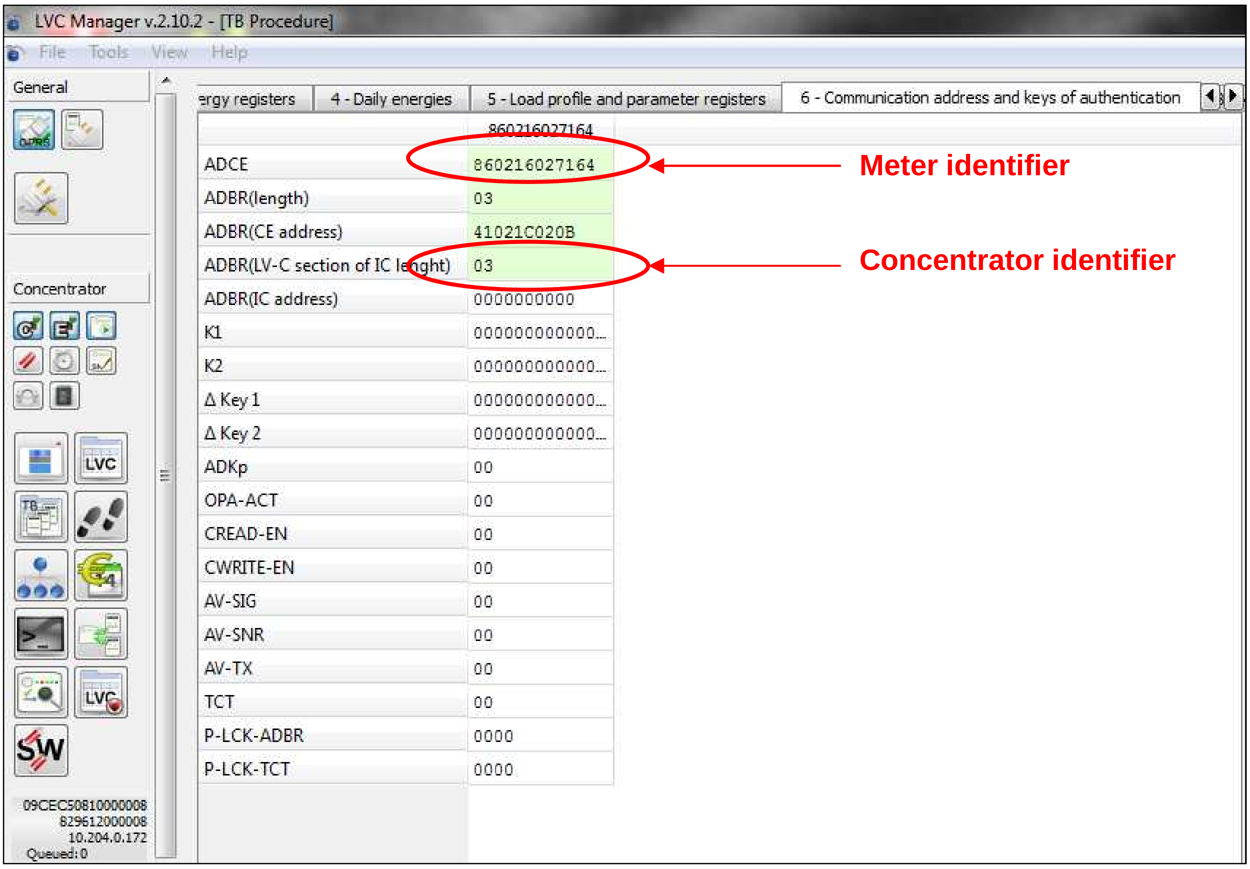


Figure 1-46: Communication address and keys of authentication

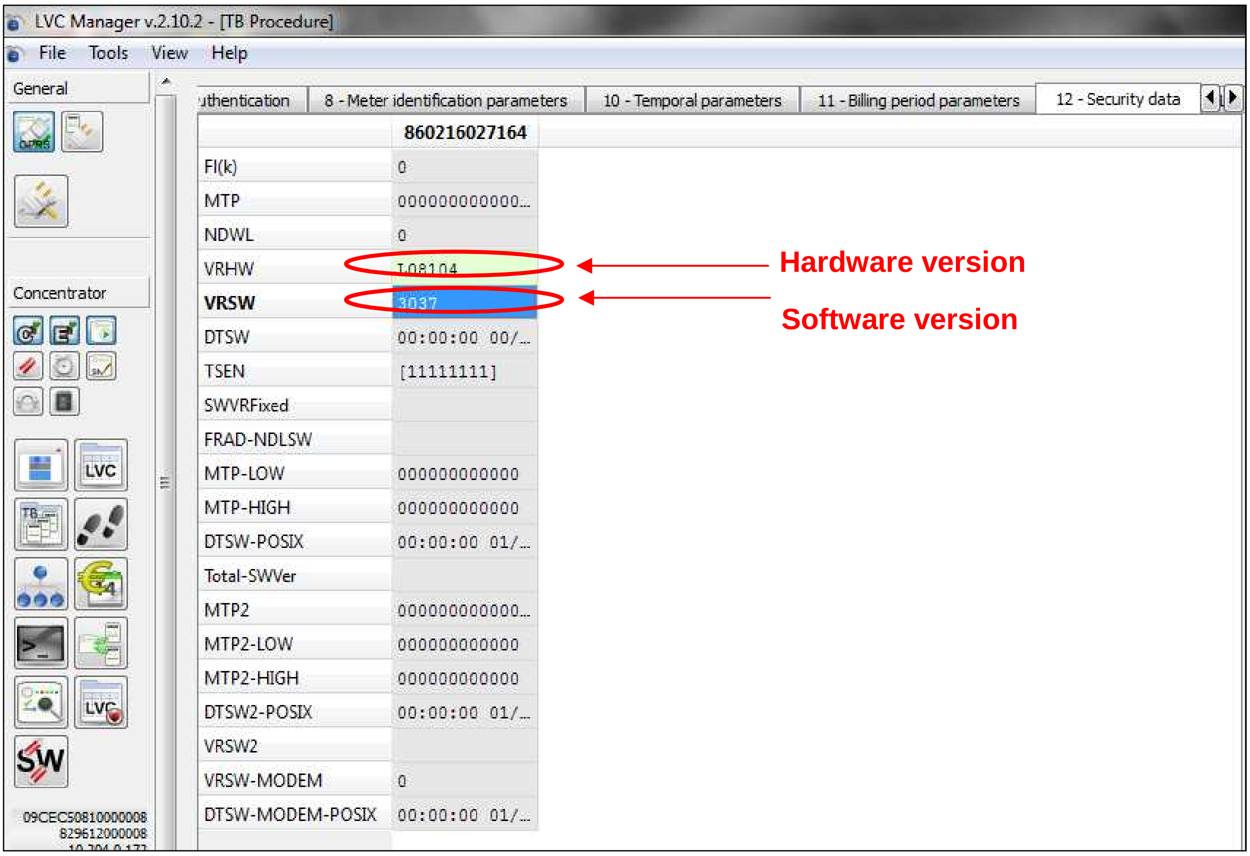


Figure 1-47: Security data: software version and hardware version

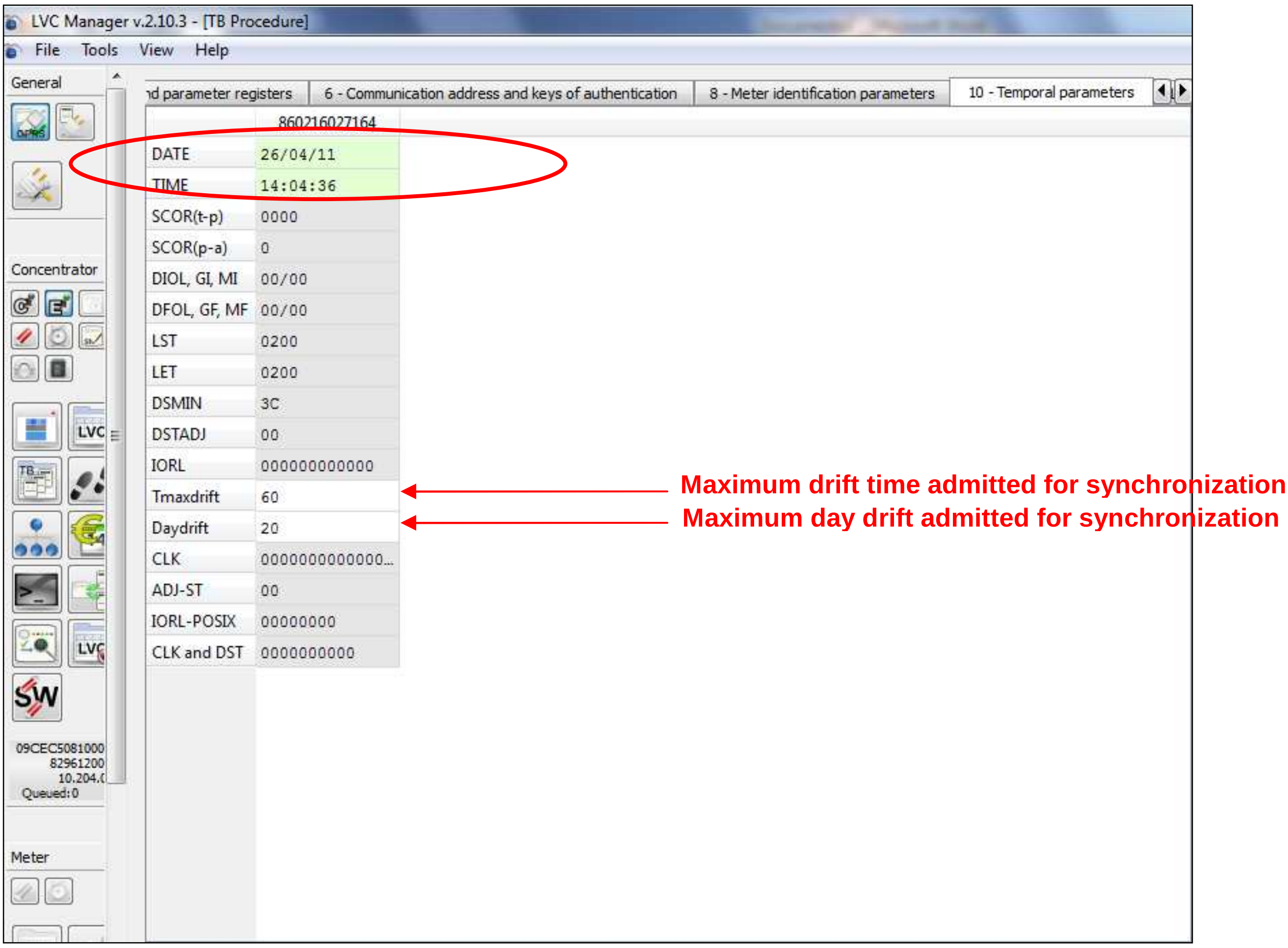


Figure 1-48: Temporal parameters

Figure below shows the parameters that defines the curve of power control

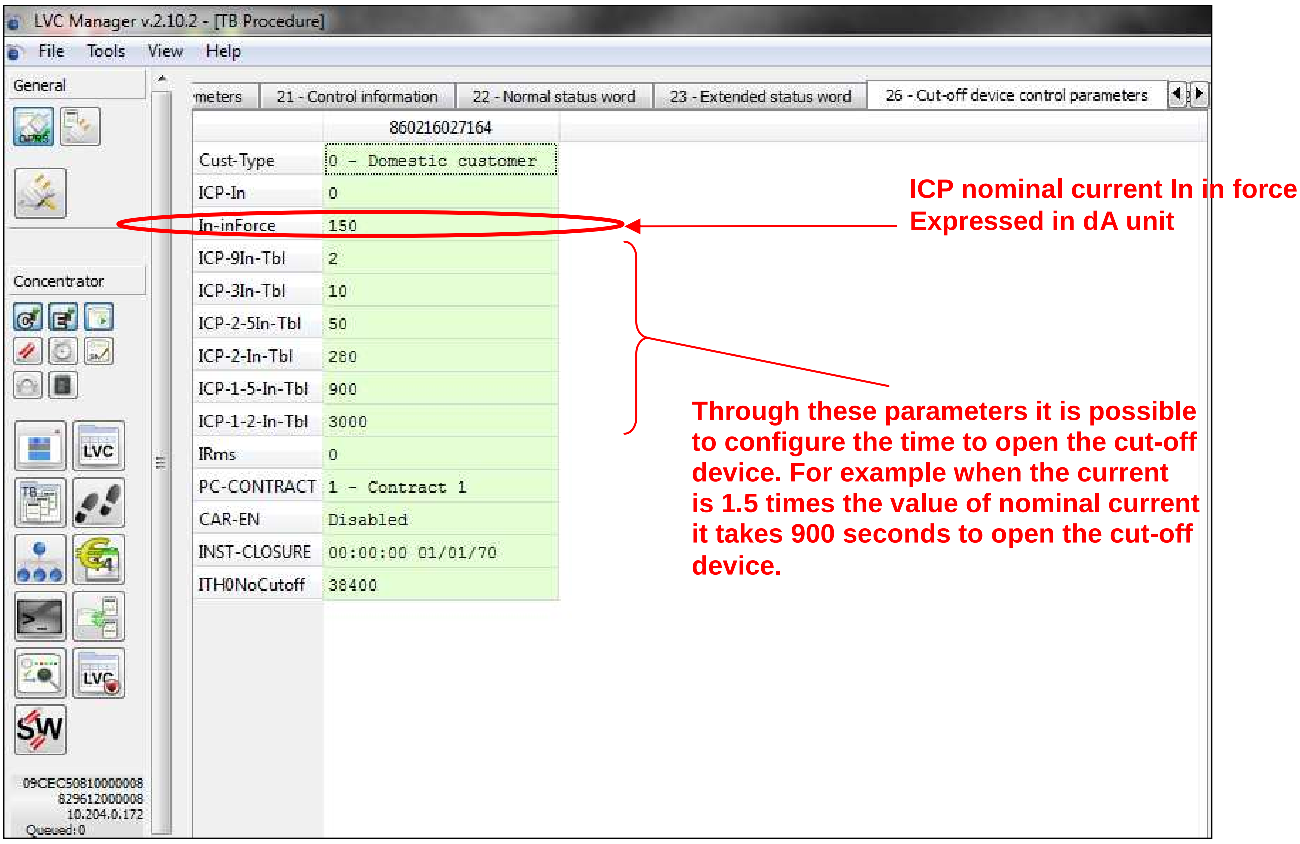


Figure 1-49: Cut-off device control parameters

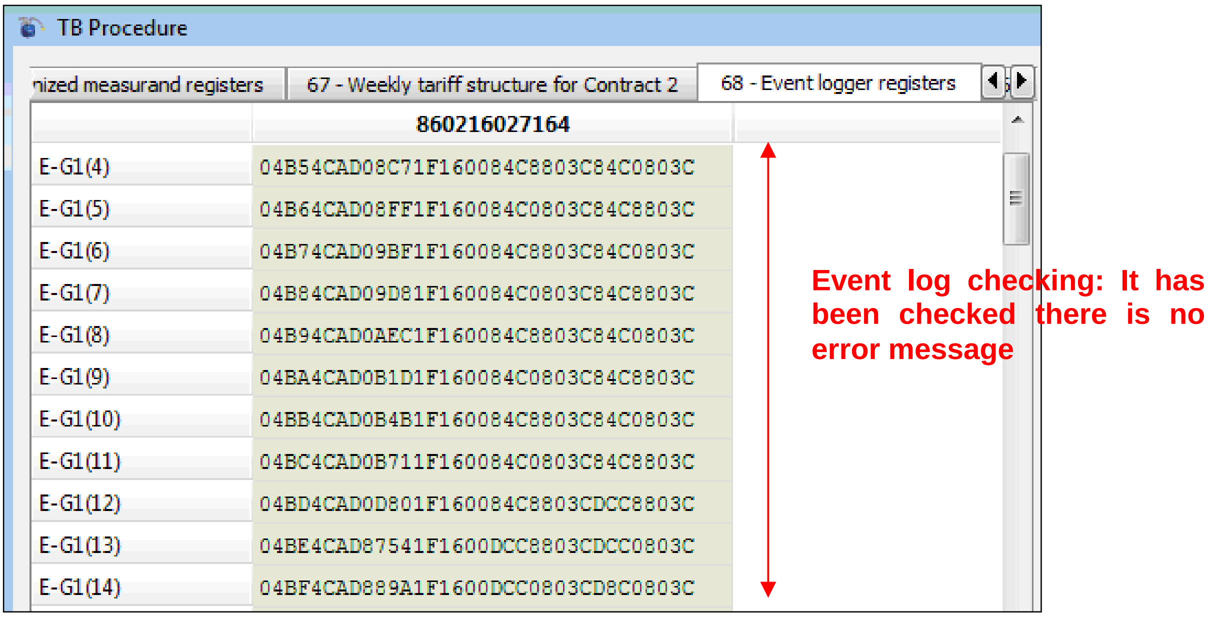


Figure 1-50: Event logger registers

Some of the steps mentioned in deliverable D4.1 for this test case (disconnectable register reading, daylight saving time parameters, table of holidays reading, energy registers reading) have been verified in other tests.

1.3.2.2 Results for OM-SR2 Remote Tariff Programming

TC 5.4.2.1

Test case has been performed successfully.

A new contract has been set, with:

- Two different tariffs for ordinary days: tariff T1 works from 00:00 to 12:00 and tariff T2 works from 12:00 to 24:00

- Tariff T4 for special days

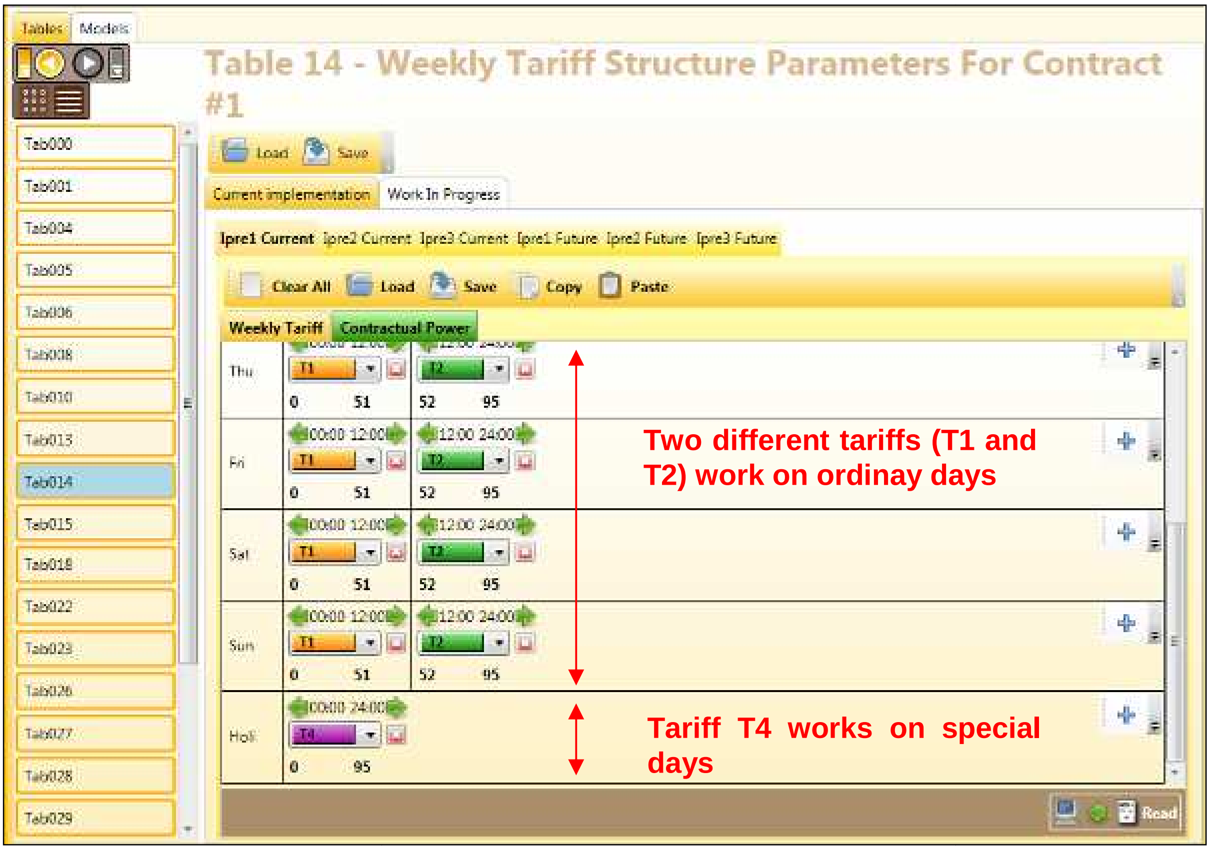


Figure 1-51: Weekly tariff structure

Then, 07.10.2010 was registered as special day/holidays

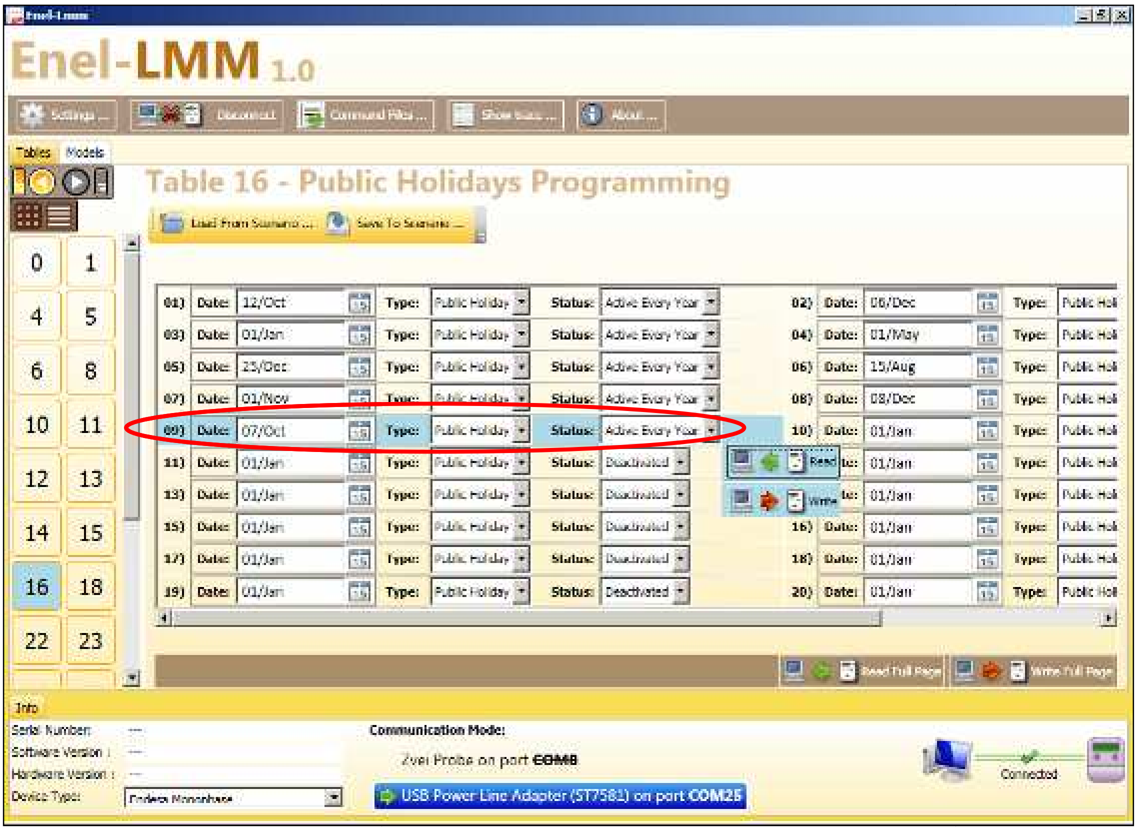


Figure 1-52: Public holidays table

When we synchronized at 10:00 on 06.10.10 we checked T1 is working:

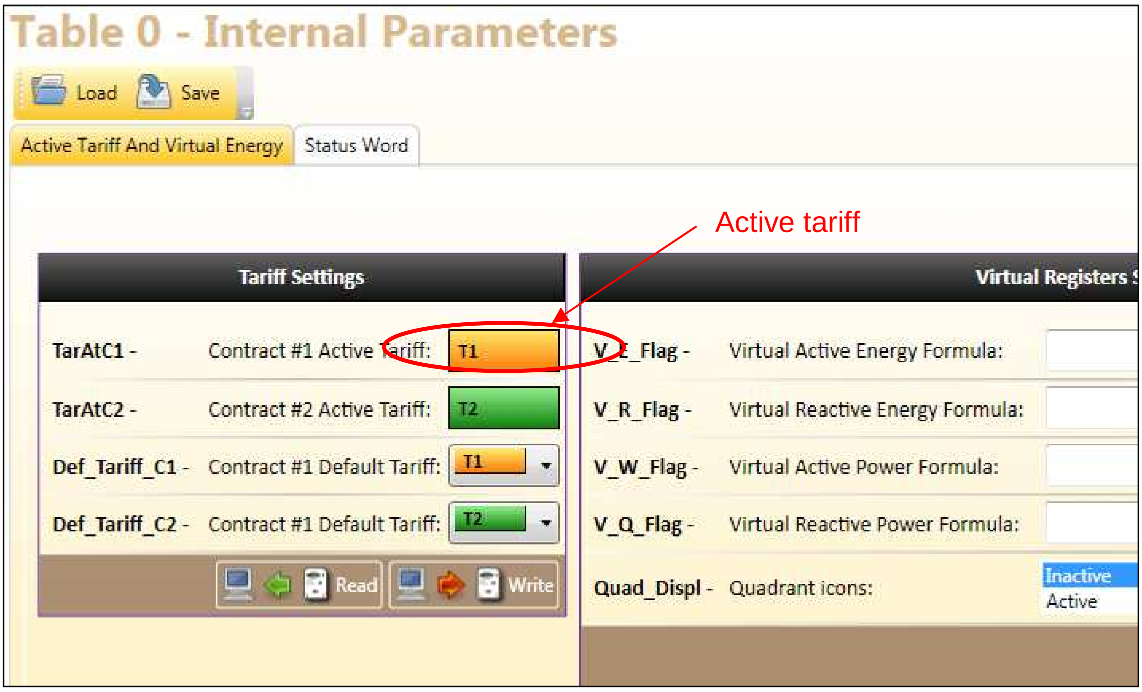


Figure 1-53: Check Active Tariff

Then the tariff shift times have been changed: Tariff T1 works from 00:00 to 13:00 and tariff T2 from 13:00 to 24:00.

Figure 1-54: New weekly tariff structure

The meter was synchronized 5 minutes before the ‘tariff switch times’ (12:55) and it was checked tariff T1 was still used.

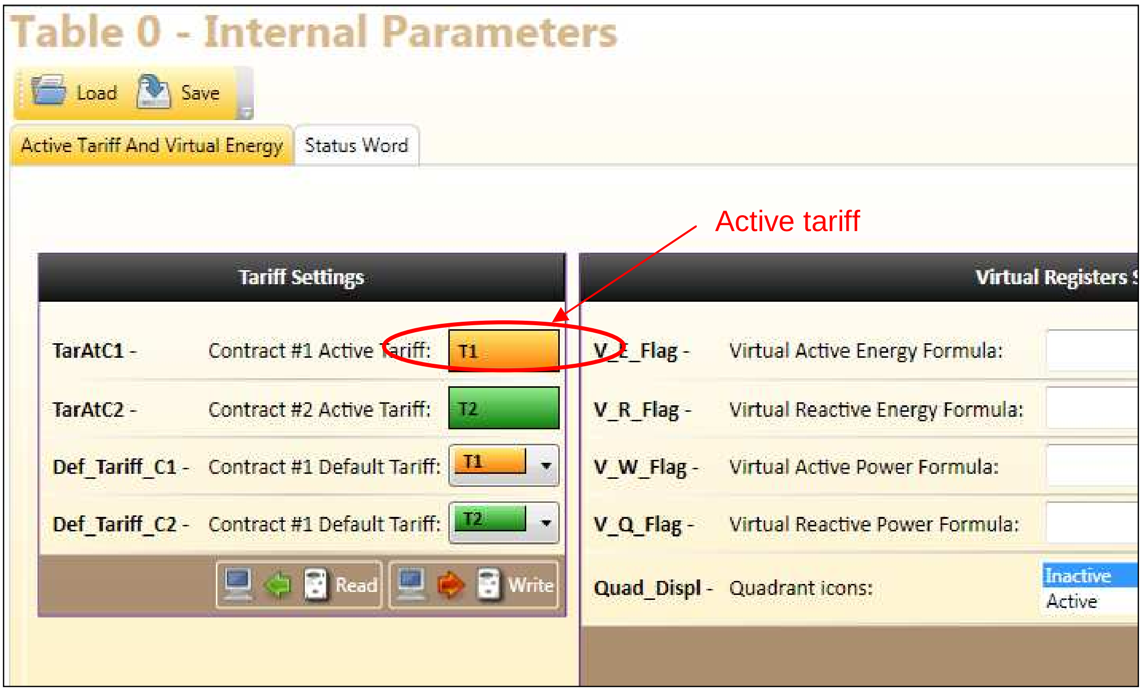


Figure 1-55: Active tariff checking

At 13:05 we checked active tariff has changed and T2 is now working:

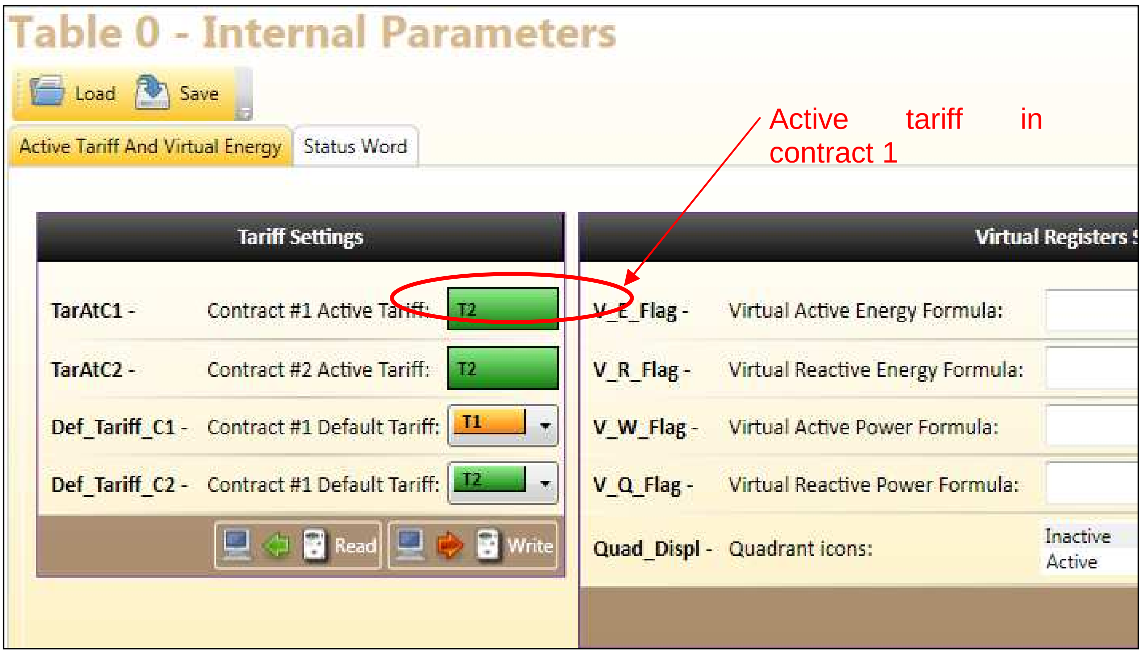


Figure 1-56: New active tariff

When we synchronized the meter on 07.10.2010 (special day) we verified that tariff ‘T4’ is now used.

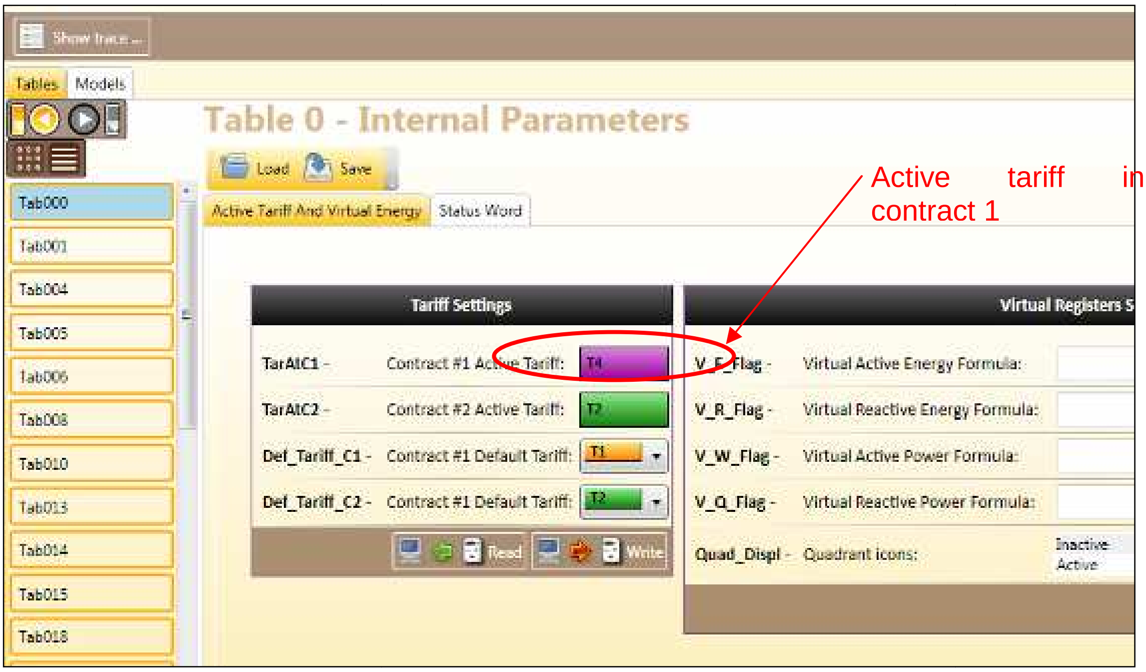


Figure 1-57: New Active tariff checking

When we synchronized the meter again at 10:00 on 08.10.2010 we verified that ‘T1’ is used

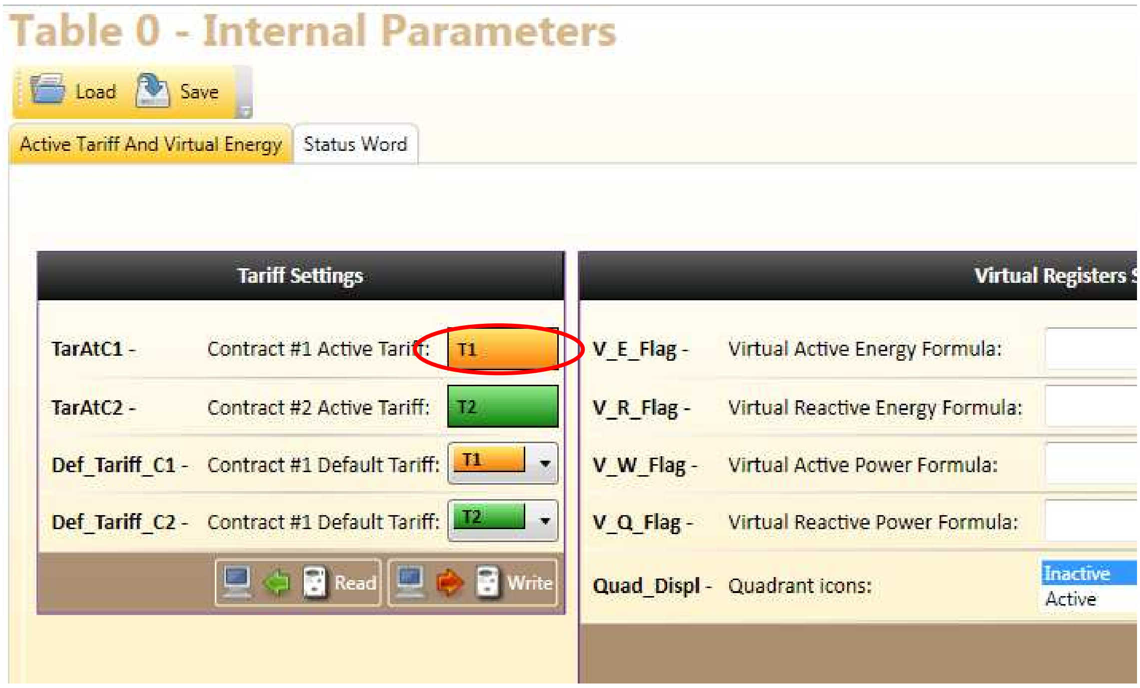


Figure 1-58: T1 Active tariff checking

1.3.2.3 Results for OM-SR3 Meter Reading (on demand)

TC 5.4.3.1

Test case has been performed successfully.

First, instantaneous energy values have been read.

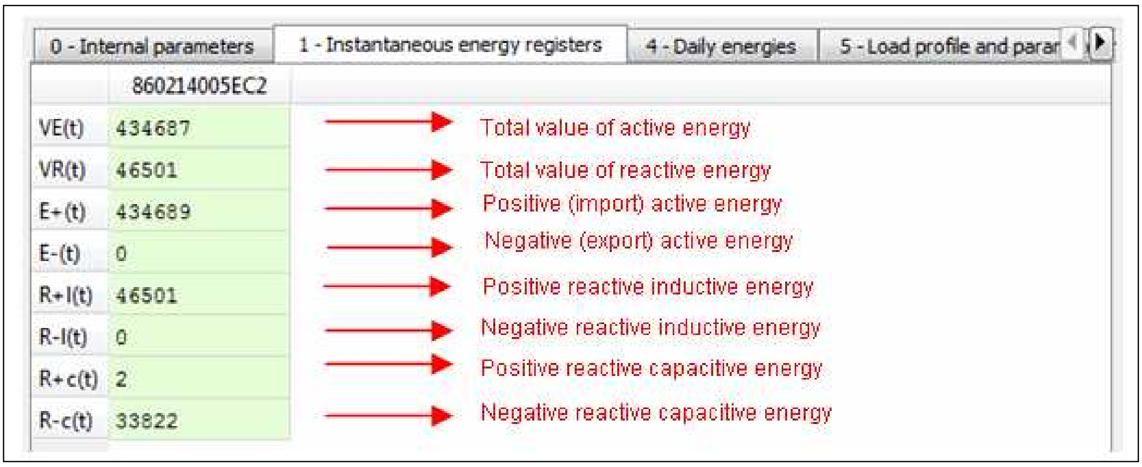


Figure 1-59: Instantaneous energy registers

When a load was connected, energy values increased. Results are shown in the figure below:

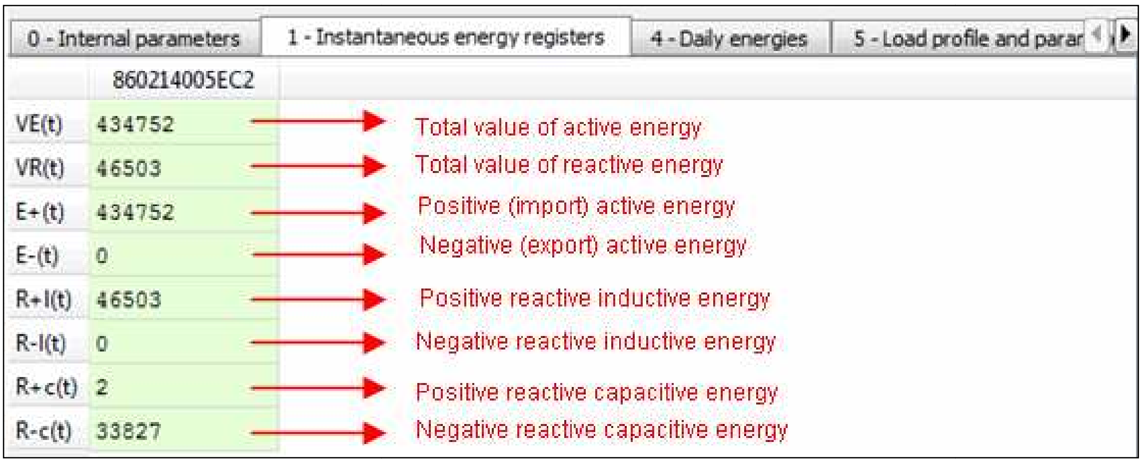


Figure 1-60: New instantaneous energy registers

Next, active and reactive energy values are shown for the current day and for the previous day:

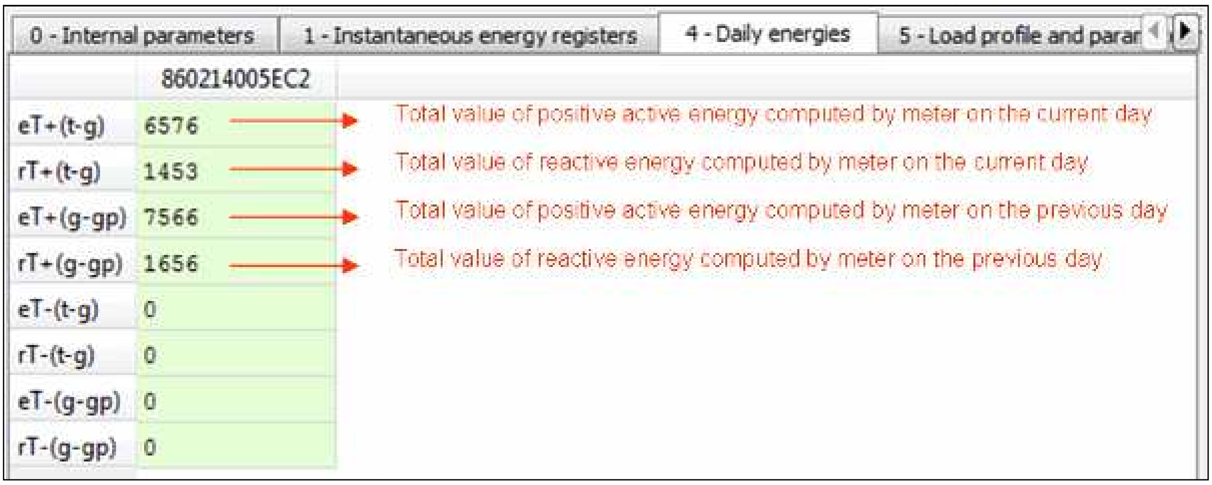


Figure 1-61: Daily energies

1.3.2.4 Results for OM-SR4 Meter Reading (for billing)

TC 5.4.4.1

Test has been performed successfully. The management of the meter readings for different billing period has been checked.

In a first part, a load has been connected and today's energy registers have been read. Next, the load has been disconnected and the meter has been synchronized at 23:58 on 06.10.2010, in order to check that the values change from 'Today Daily Energy' to 'Yesterday Daily Energy'. Two readings have been performed: one on the 06.10.2010 at 23:58 and the other on 07.10.2010 at 00:02.

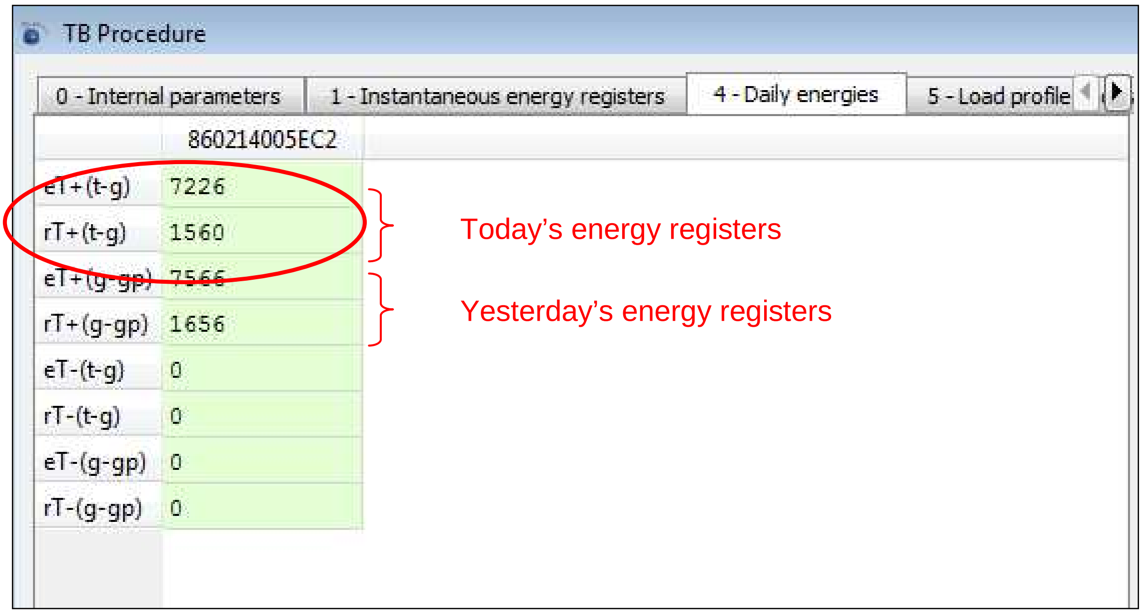


Figure 1-62: Meter reading on the 06.10.2010 at 23:58

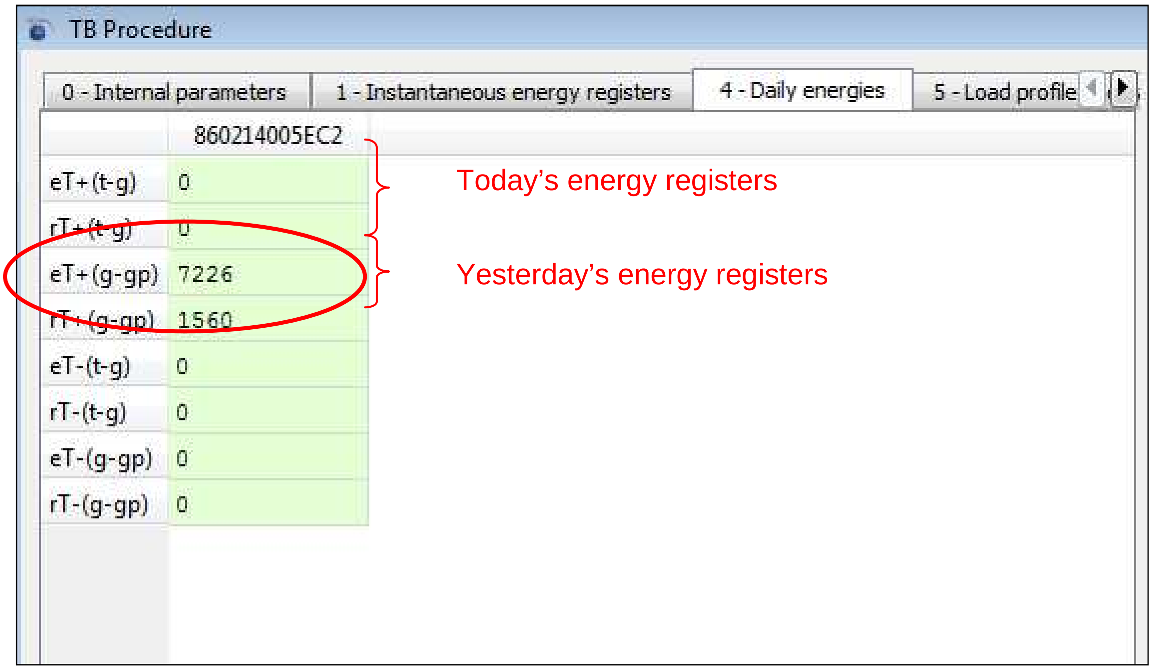


Figure 1-63: Meter reading on the 07.10.2010 at 00:02.

Next, test engineers set the end of the billing period equal to 07th of October and a periodicity equal to 1 month. The meter was synchronized at the end of the billing periods several times (in different months) to verify that the values changed from 'Current Period' to 'Previous Period', then from previous period to last but one billing period and, finally, from last but one billing period to last but two billing period.

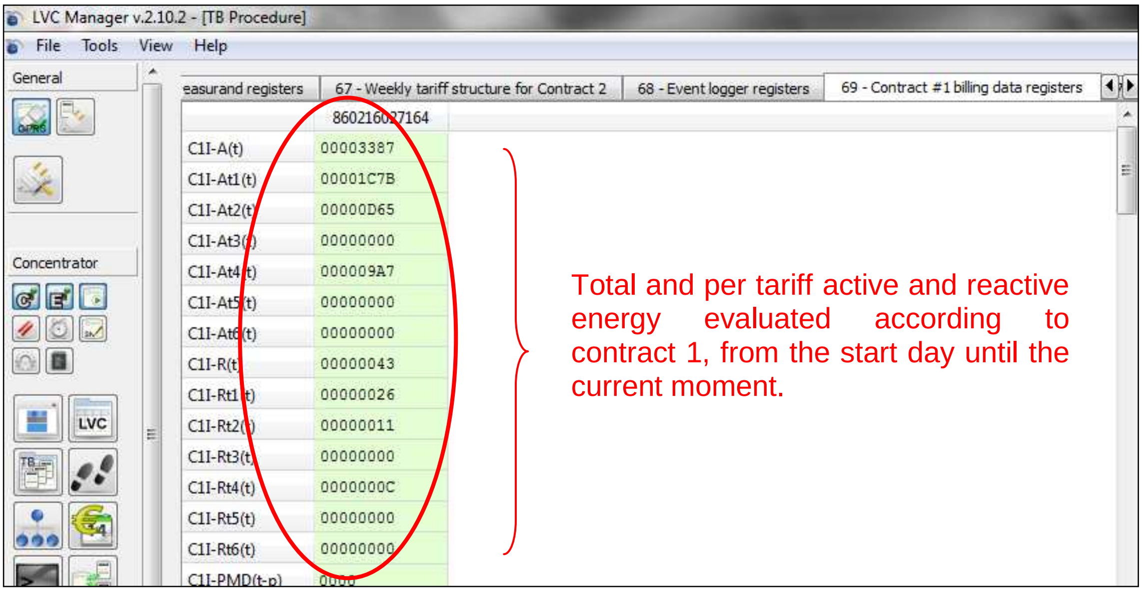


Figure 1-64: Current period readings at 23:55 on 07.10.2010

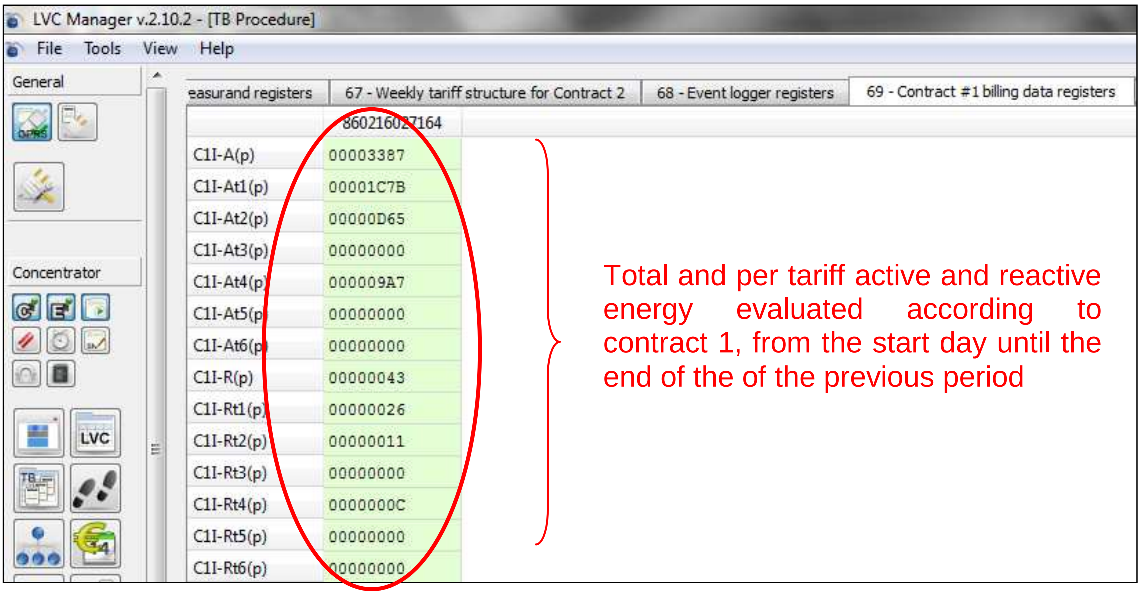


Figure 1-65: Previous period readings at 00:02 on 08.10.2010

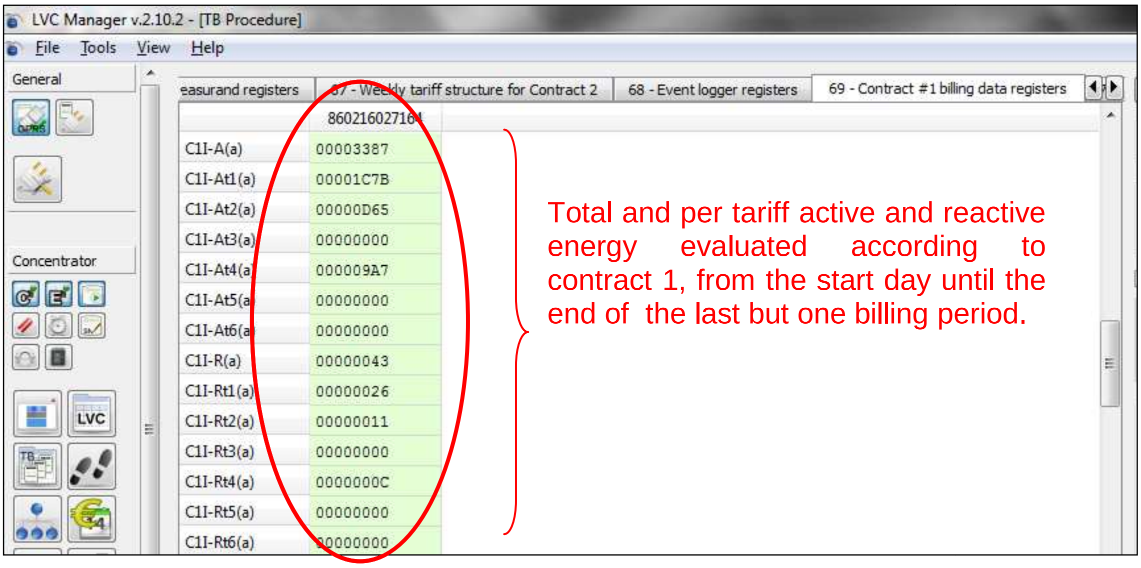


Figure 1-66: The last but one billing period readings at 00:02 on 08.11.2010

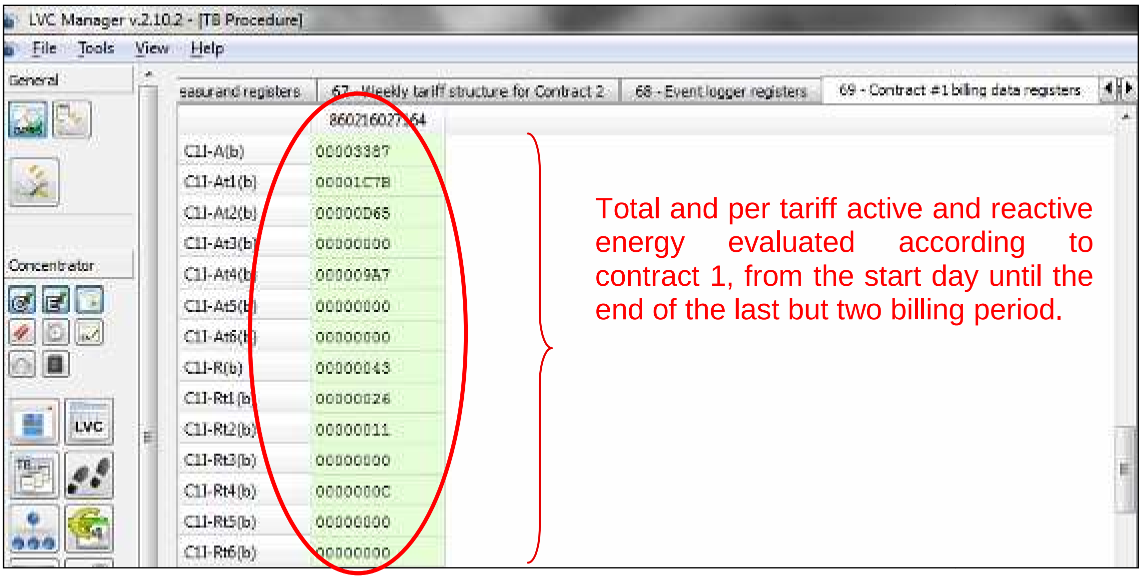


Figure 1-67: Values store in the last but two billing period readings at 00:02 on 08.12.2010

1.3.2.5 Results for OM-SR5 Remote Disconnection and Reconnection

TC 5.4.5.1

Test has been performed successfully. In this test, the meter has been disconnected and reconnected on a designated date at a specified time.

The meter disconnection was scheduled at 13:05 on 06.10.2010 and the meter reconnection at 13:15 on 06.10.2010. The functionally has been verified by checking the event log table, the voltage and current values and by visual inspection on the meter display.

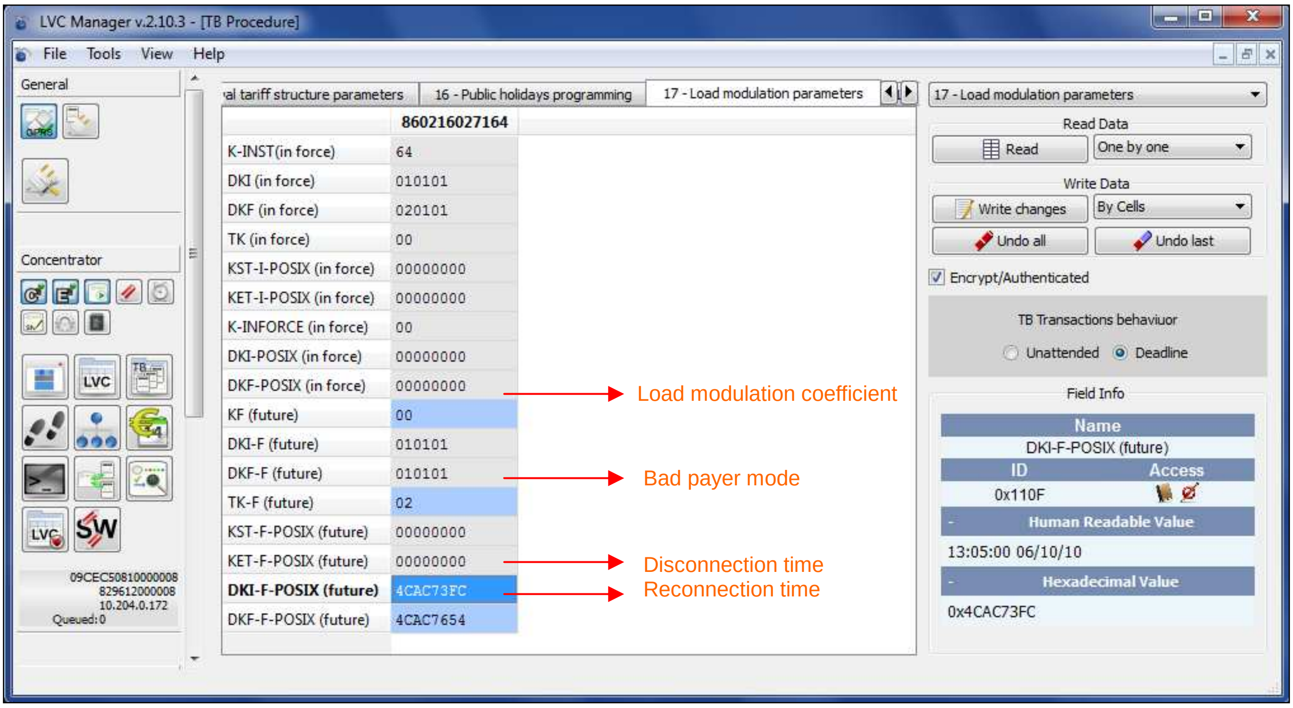


Figure 1-68: TAB 17: Parameter programming to disconnect and reconnect on a designated date at the specified time

Figure below shows that a procedure to disconnect the meter was activated at 13:05.

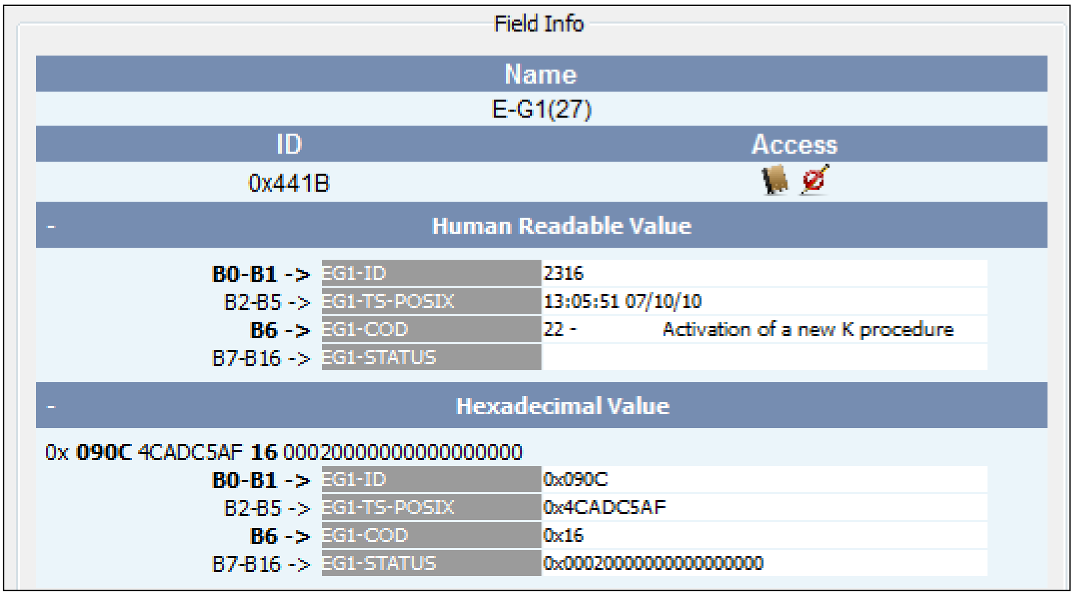


Figure 1-69: Event Log Table. Activation of a new K procedure

Figures below show the power, voltage and current values before and after the meter disconnection and reconnection:

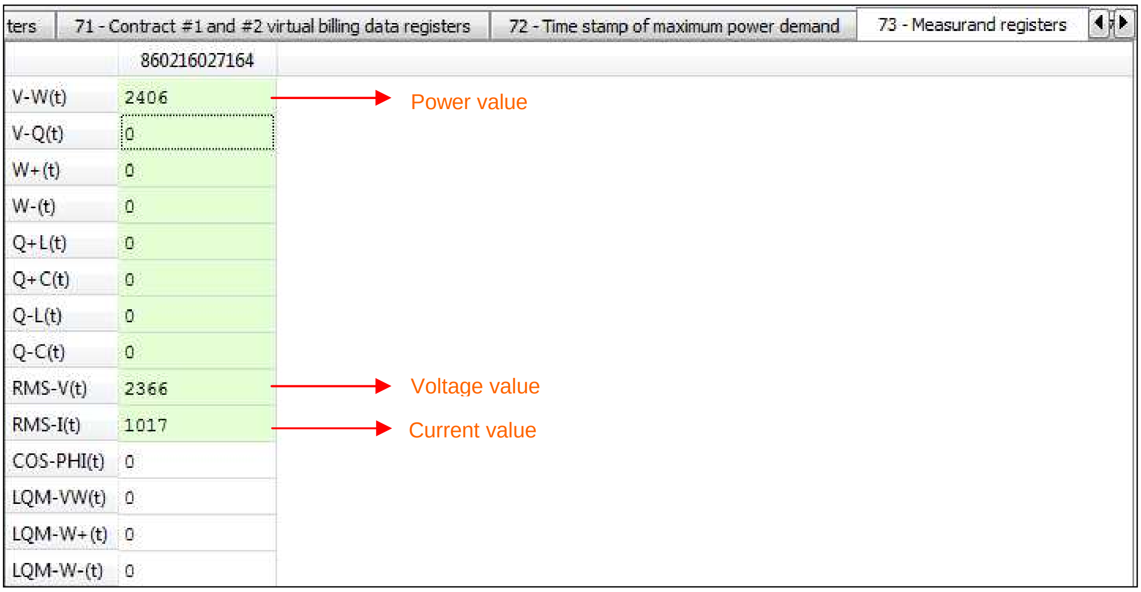


Figure 1-70: Measured registers at 13:04 (before the meter disconnection)

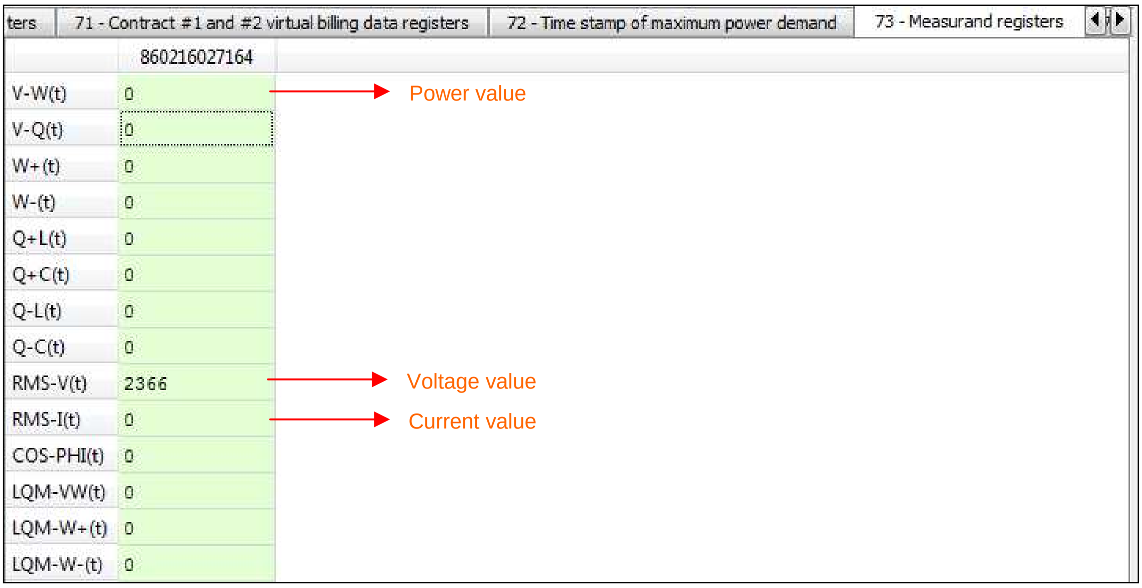


Figure 1-71: Measured registers at 13:06 (after the meter disconnection)

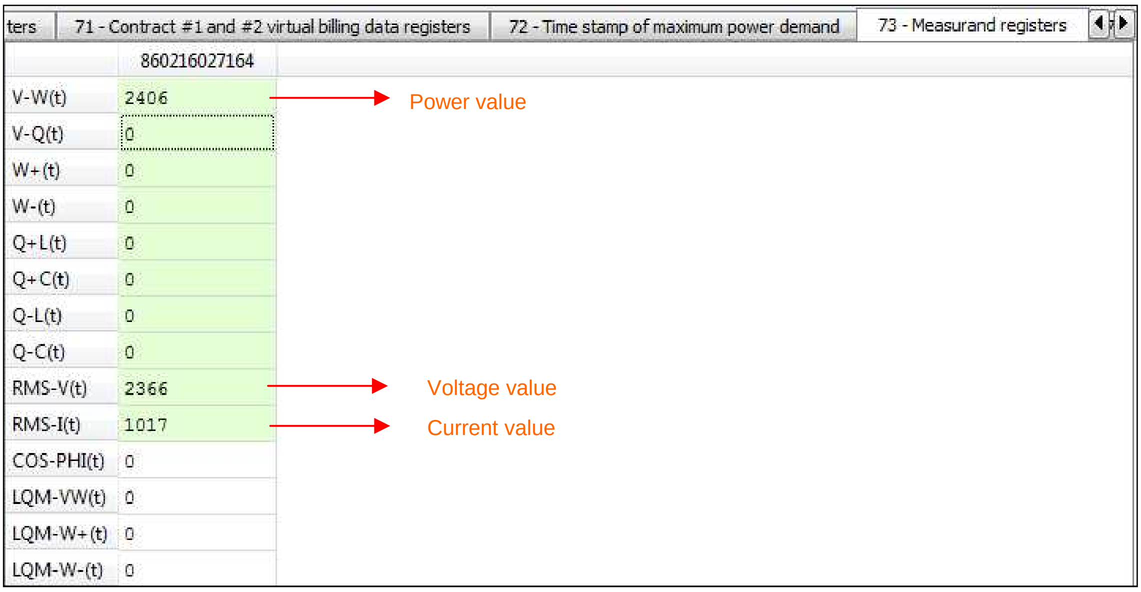


Figure 1-72: Measured registers at 13:16 after the meter reconnection

The following figures are photos of the meter display when the cut-off device was open and closed:

Cut-off device symbol

Figure 1-73: Meter display Cut-off open

Cut-off device symbol has disappeared

disappeared

Figure 1-74: Meter display Cut-off closed

TC 5.4.5.2

Test has been performed successfully. In this test, the meter has been disconnected/reconnected immediately.

First of all, the load modulation table was configured.

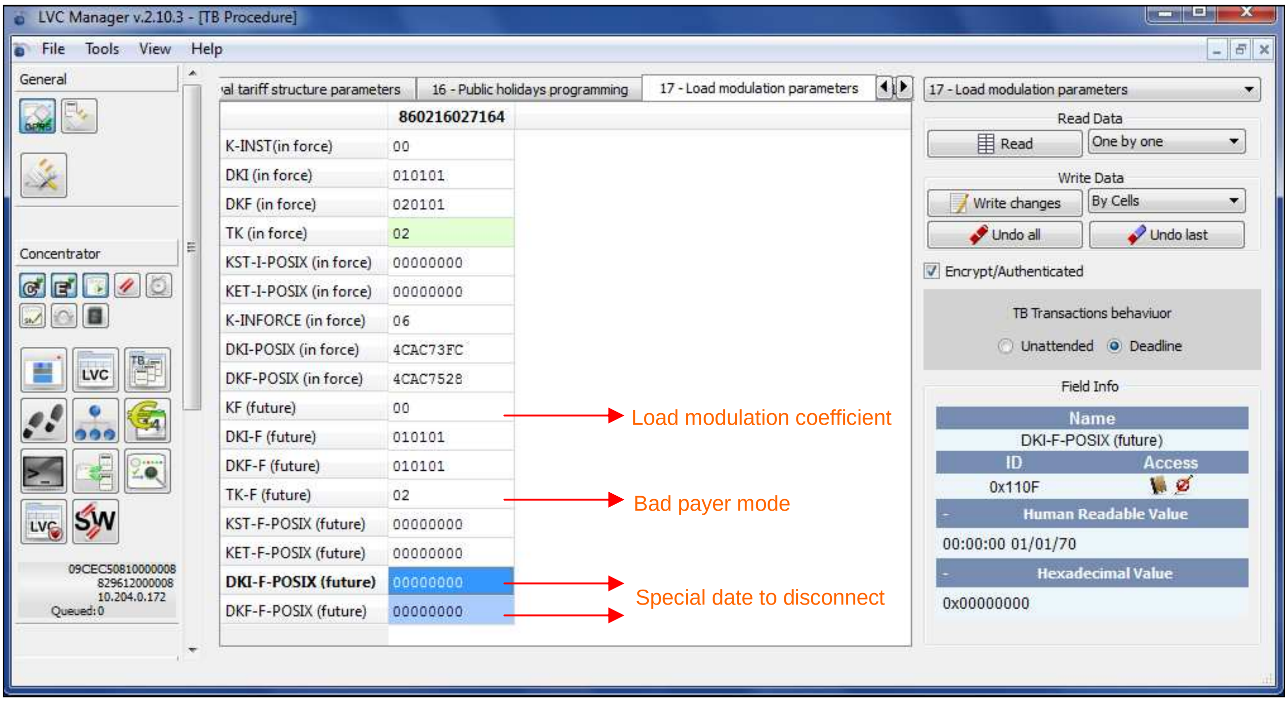


Figure 1-75: TAB 17: Parameter programming for load modulation

Figures below show the events that were generated on the event log because of the meter disconnection.

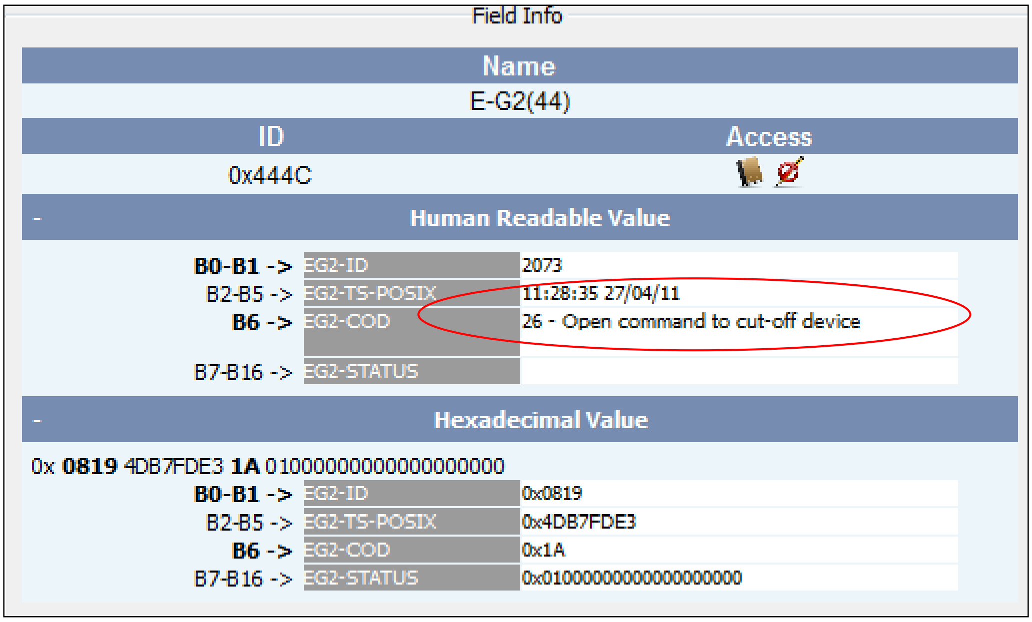


Figure 1-76: Event log table. Open command to cut-off device

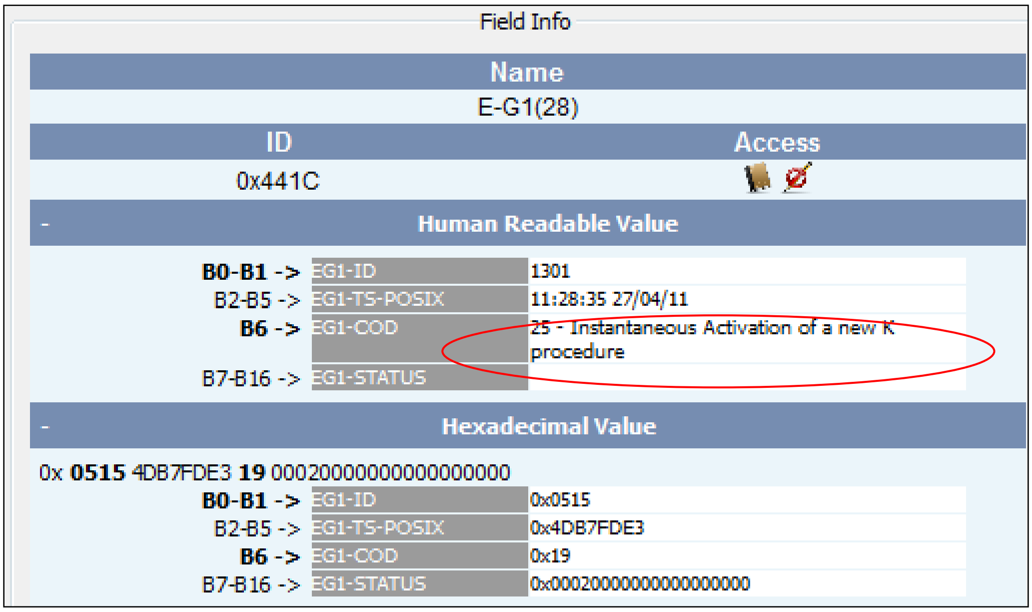


Figure 1-77: Event log table. Activation of a new K procedure

As in the previous test, the functionality also was verified by checking the voltage and current values and through the visual inspection of the meter display.

TC 5.4.5.3

The test case has been performed successfully. It has been checked that the meter issues a logical error if the (dis)connection can't be applied.

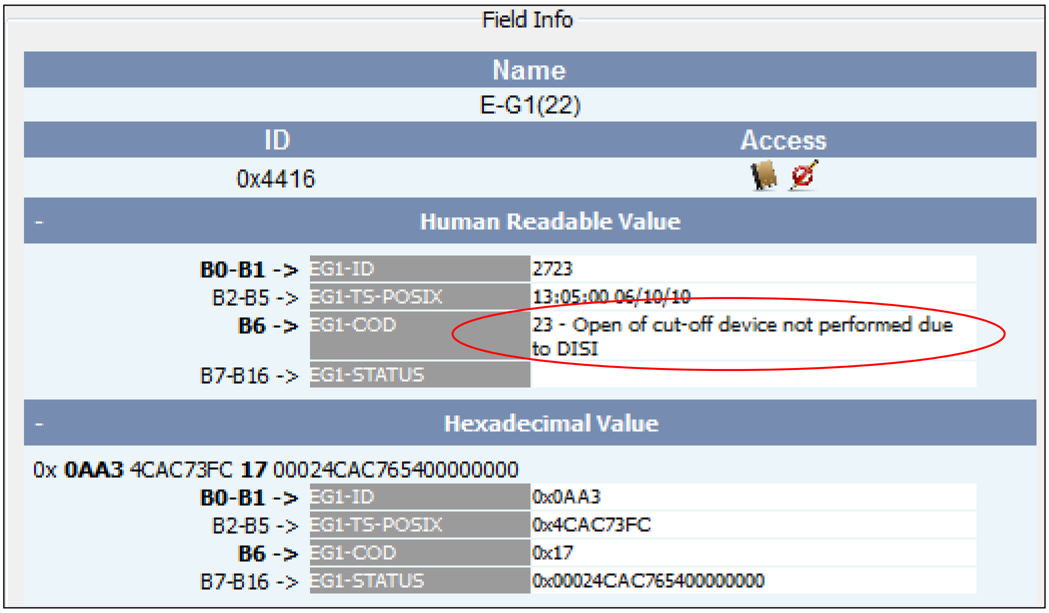


Figure 1-78: Event log table. Open of cut-off device not performed due to DISI

As in the previous test, the functionality also was verified by checking the voltage and current values and through the visual inspection of the meter display.

1.3.2.6 Results for OM-SR6 Power control

TC 5.4.6.1

Power control test has been performed successfully. A limit of power=500 W has been set and when a load bigger than this value has been connected, the meter has opened the cut-off device.

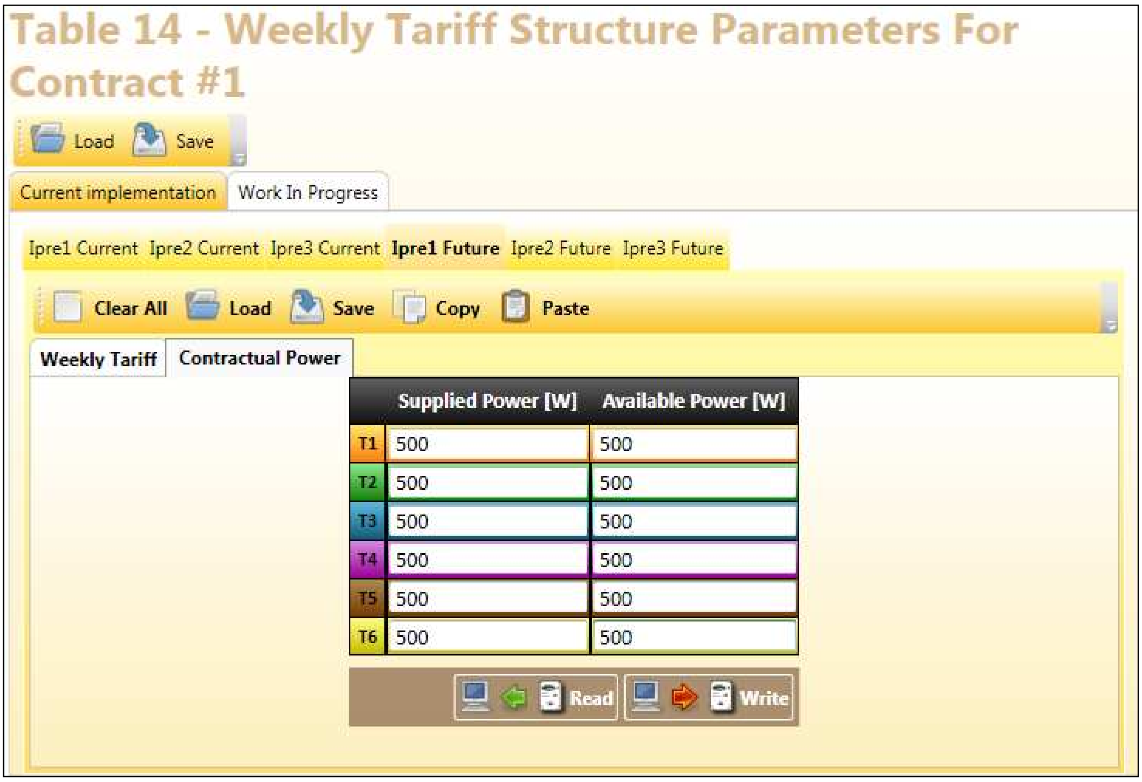


Figure 1-79: Threshold for supplied power

The functionality was verified by checking the voltage and current values and through the visual inspection of the meter display.

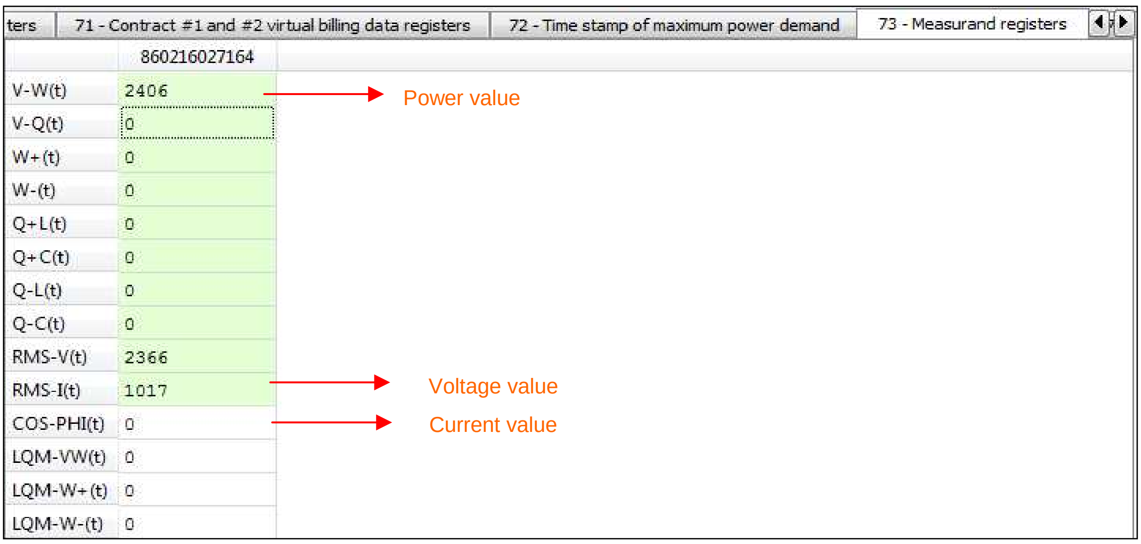


Figure 1-80: Misbrand registers before the meter opened the cut-off device

ters | 71 - Contract #1 and #2 virtual billing data registers | 72 - Time stamp of maximum power demand | 73 - Measurand registers

	860216027164
V-W(t)	0
V-Q(t)	0
W+(t)	0
W-(t)	0
Q+L(t)	0
Q+C(t)	0
Q-L(t)	0
Q-C(t)	0
RMS-V(t)	2366
RMS-I(t)	0
COS-PHI(t)	0
LQM-VW(t)	0
LQM-W+(t)	0
LQM-W-(t)	0

Figure 1-81: Misbrand registers after the meter opened the cut-off device

TC 5.4.6.2

The test case has been performed successfully. A factor K has been used to reduce the available power.

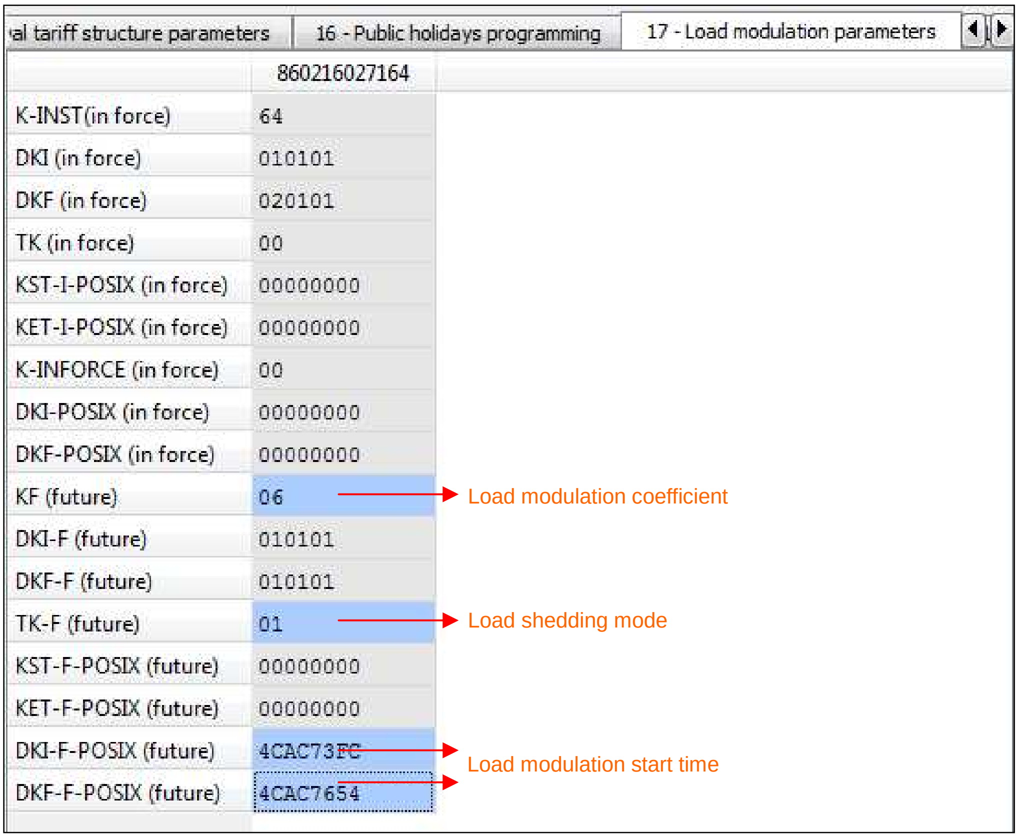


Figure 1-82: Parameter programming for load modulation

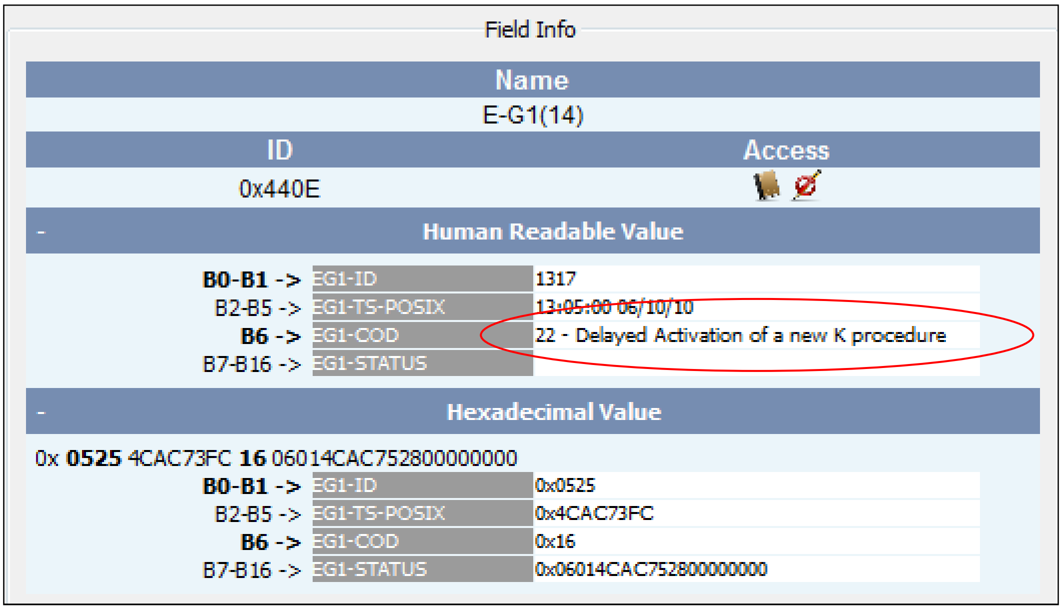


Figure 1-83: Information related to the last event

1.3.2.7 Results for OM-SR8 Remote meter firmware update

TC 5.4.8.2

The firmware update has been performed successfully. Before the update, the software version was 4D37 (human readable: M7) and after 3037 (human readable: 07). The information also has been registered on the event log table. Results are shown in the figures below:

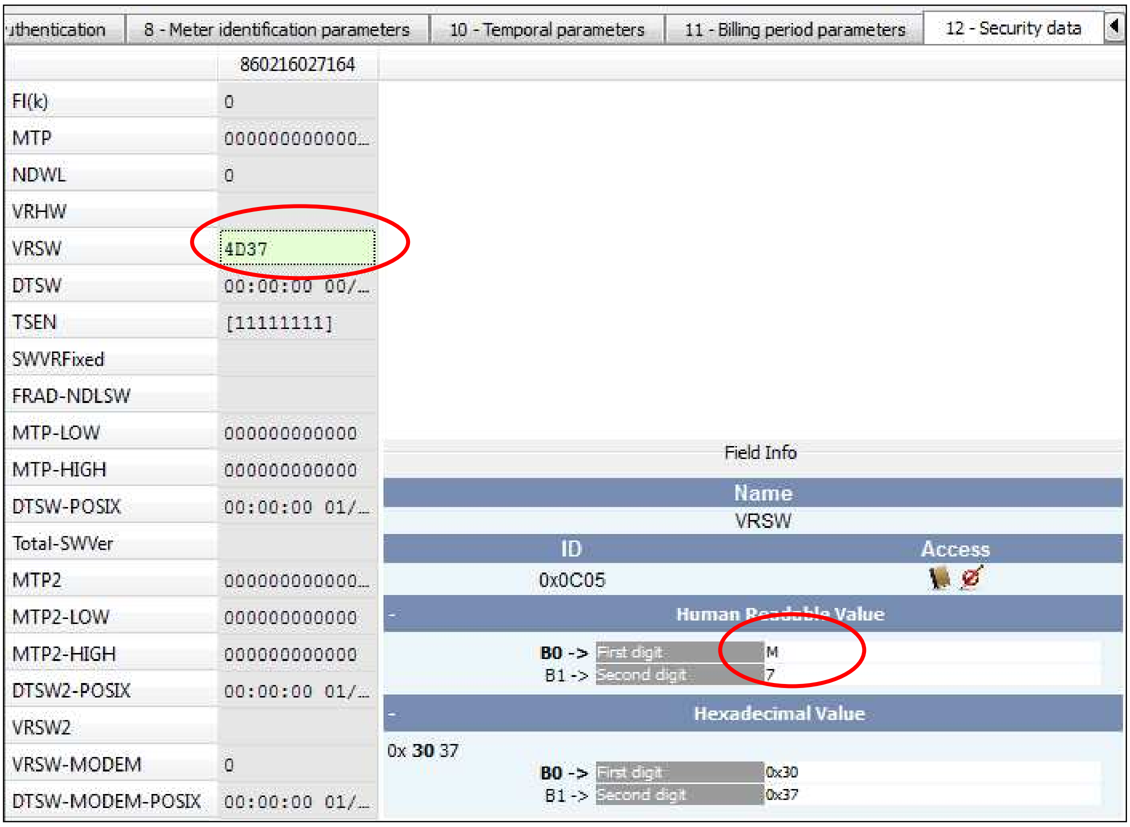


Figure 1-84: Software version before the firmware update

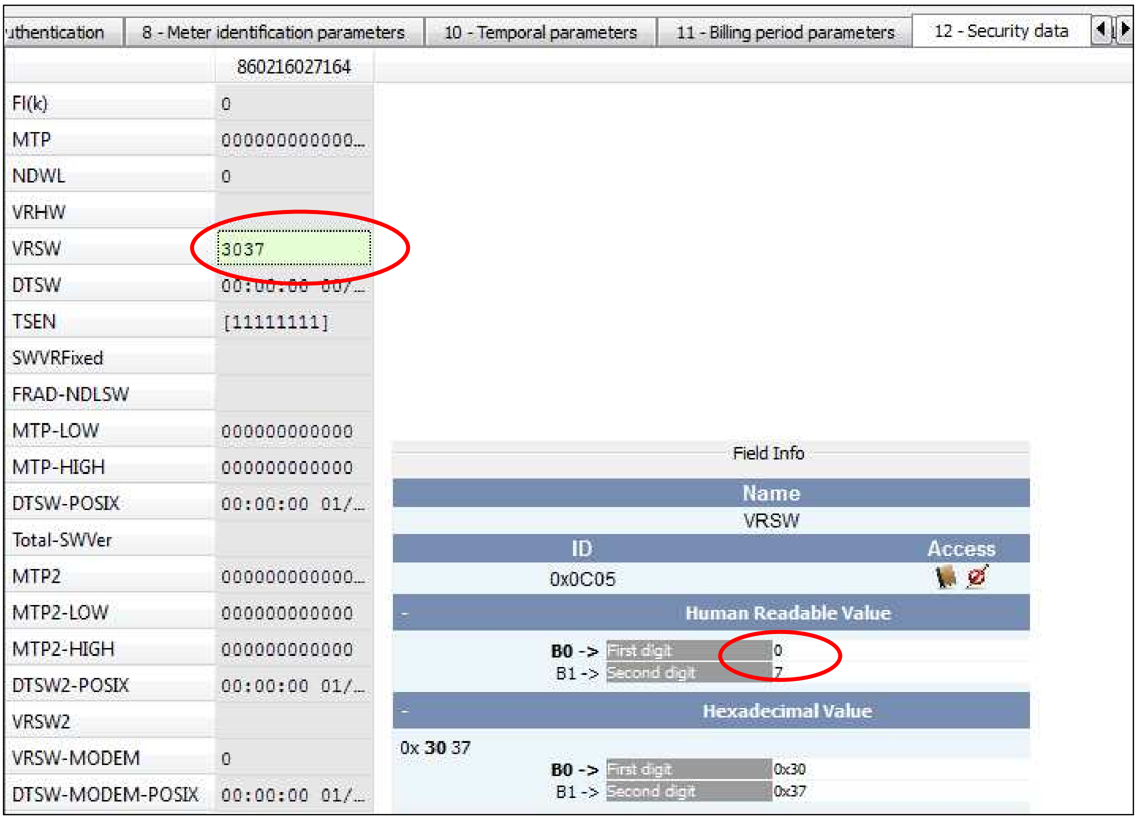


Figure 1-85: Software version after the firmware update

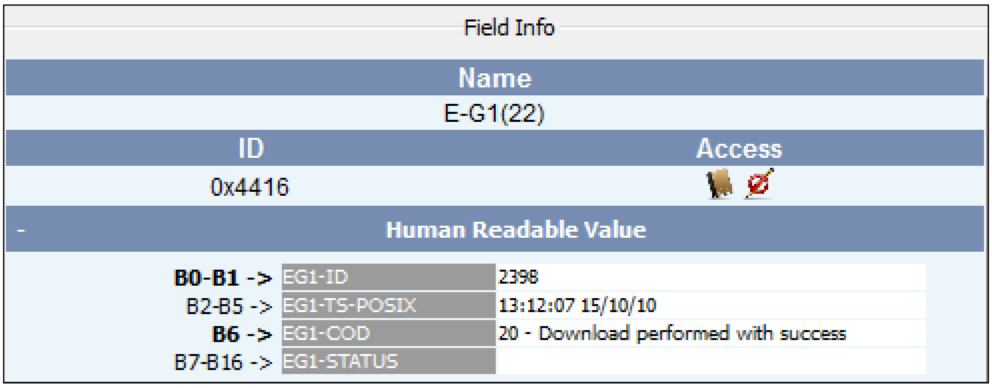


Figure 1-86: Event log table. Download performed with success

TC 5.4.8.3

The test has been performed successfully. After downloading on the meter the corrupted version of the firmware, the meter discards it and issues a logical error.

The logical error is shown on the event log, see table below:

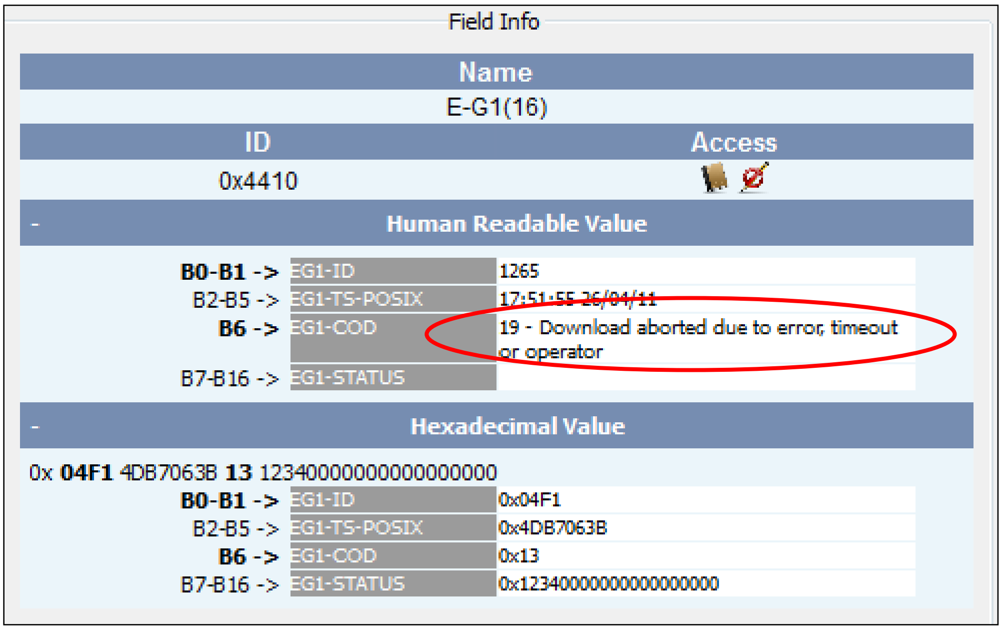


Figure 1-87: Event log table. Download aborted

1.3.2.8 Results for OM-SR9 Alarm and event management

TC 5.4.9.1

Test has been performed successfully. In this test, alarm and events information have been checked.

When the value of maximum day drift admitted was modified, an event was generated on the event log with all the required information: time stamp, activity type and programming information:

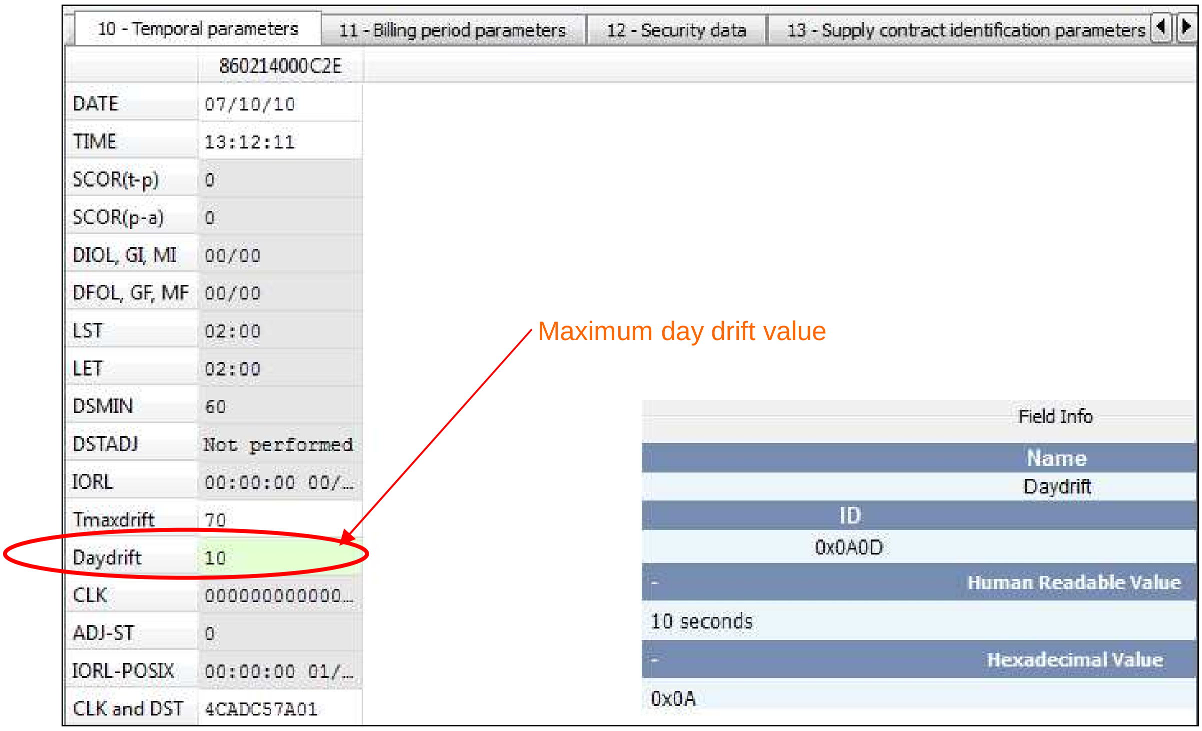


Figure 1-88: Parameter programming on temporal parameters table

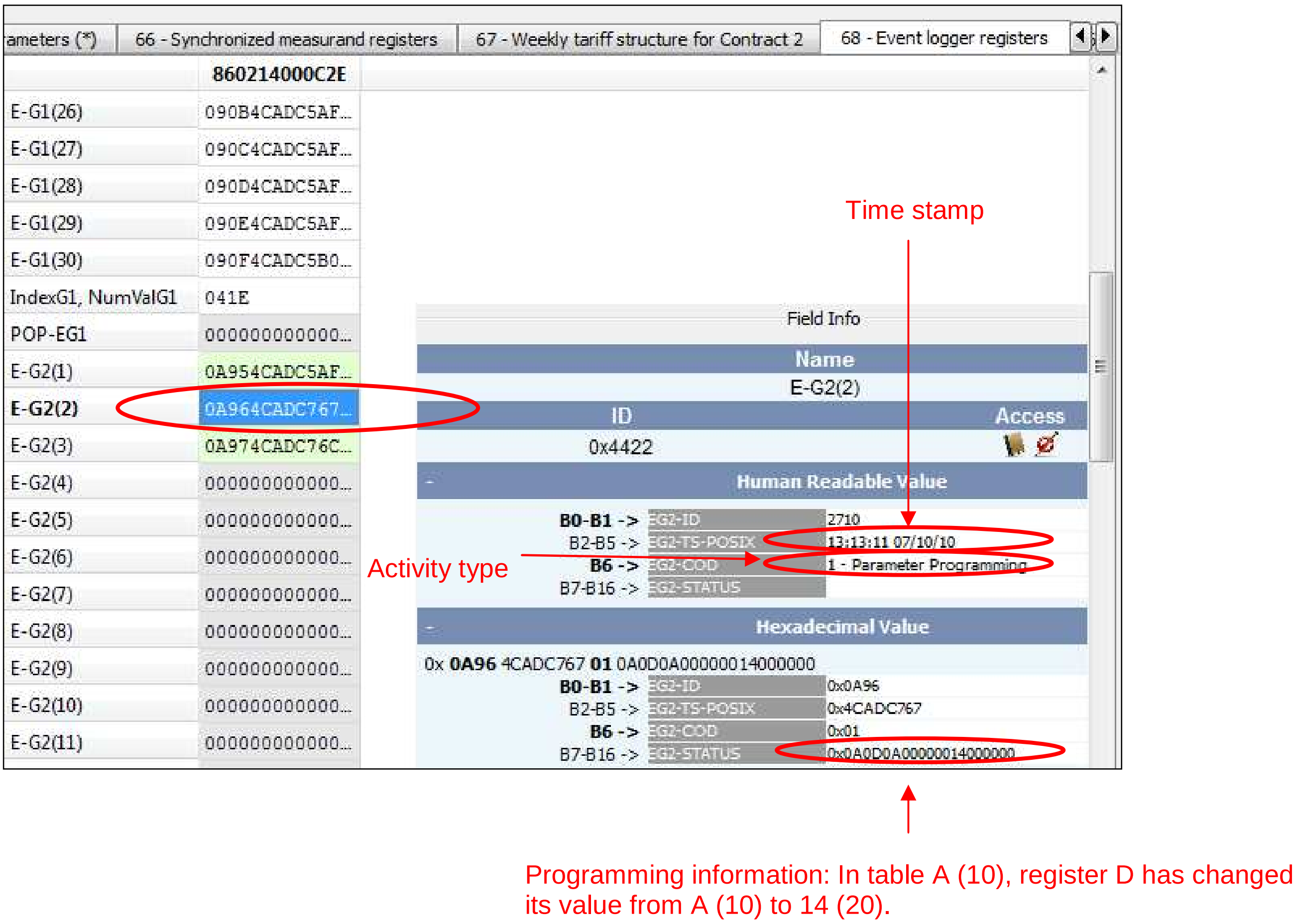


Figure 1-89: Event log table. Parameter programming

When terminal cover was removed, it was generated a variation on the status word, and the bit related to TC_REM Terminal cover removed has been activated. As a consequence of the status word variation a new event is generated on the event log table with all the required information.

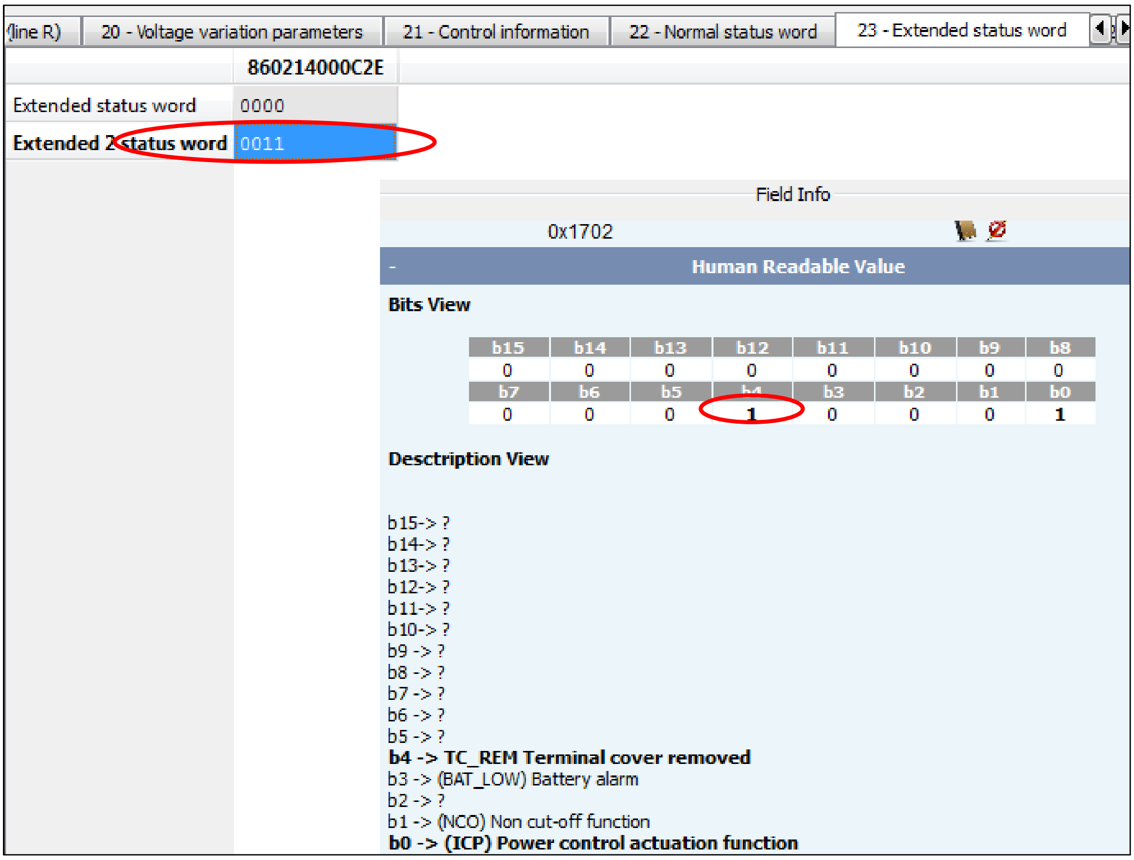


Figure 1-90: Status word table value after the terminal cover was removed

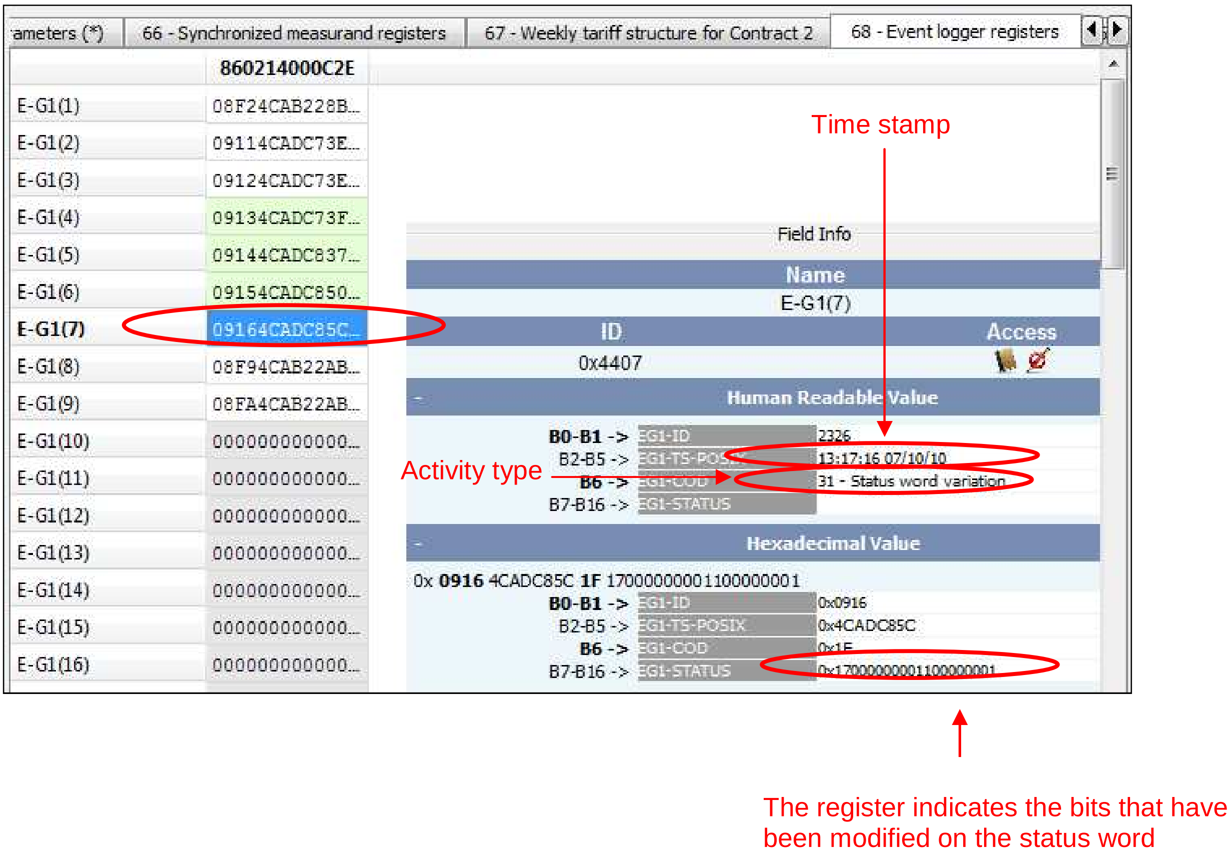


Figure 1-91: Event log table. Terminal cover removed

1.3.2.9 Results for OM-SR10 Interruption Information

TC 5.4.10.1

The test has been performed successfully. It has been checked the information that the meter retrieves related to voltage interruptions, which is stored in table “Voltage interruption parameters”. Figure below shows the values for the two meters that were analyzed during this test:

19 - Voltage interruption parameters (line R) | 20 - Voltage variation parameters | 21 - Control infor

	860214005E04	860214005E0C
THVI	32	32
TSI	0001	0001
TLI	00B4	00B4
TW	02	02
TSHV(t-p)	00000000	00000000
TSHV(p-a)	00000000	00000000
TAVV(t-p)	00000000	00000000
TAVV(p-a)	00000091	000003FB
TLOV(t-p)	00000000	00000000
TLOV(p-a)	00000DCB	00000DDC
NSHV(t-p)	00	00
NSHV(p-a)	00	00
NAVV(t-p)	00	00
NAVV(p-a)	04	80
NLOV(t-p)	00	00
NLOV(p-a)	03	03
NVTD(t-p)	00	00
NVTD(p-a)	00	05

Parameters meanings are explained below:

Figure 1-92: Voltage interruption parameters

Registers meanings are described below:

- THVI: Threshold of voltage for interruption measurement
- TSI: Threshold duration for transient voltage interruptions.
- TLI: Threshold duration for short voltage interruptions.
- TW: Period of time, expressed in 15ms intervals, devoted to measure the voltage for interruptions.
- TSHV (t-p): Totalizer that indicates the duration, in seconds, of voltage interruptions till TSI seconds in the current billing period.
- TSHV (p-a): Totalizer that indicates the duration, in seconds, of voltage interruptions till TSI seconds in the previous billing period.
- TAVV (t-p): Totalizer that indicates the duration, in seconds, of voltage interruptions from TSI to TLI seconds in the current billing period.
- TAVV (p-a). Totalizer that indicates the duration, in seconds, of voltage interruptions from TSI to TLI seconds in the previous billing period
- TLOV (t-p): Cumulative totalizer that indicates the duration, in seconds, of voltage interruptions over TLI seconds in the previous billing period.

- TLOV (p-a): Totalizer that indicates the duration, in seconds, of voltage interruptions over TLI seconds in the current billing period.
- NSHV (t-p): Counter that indicates the number of voltage interruptions till TSI seconds in the current billing period.
- NSHV (p-a): Counter that indicates the number of voltage interruptions till TSI seconds in the previous billing period.
- NAVV (t-p): Counter that indicates the number of voltage interruptions from TSI to TLI seconds in the current billing period.
- NAVV (p-a): Counter that indicates the number of voltage interruptions from TLI to TLI seconds in the previous billing period.
- NLOV (p-a): Counter that indicates the number of voltage interruptions over TLI seconds in the previous billing period.
- NTVD (t-p): Counter that indicates the number of meter power fails in the current billing period.
- NTVD (p-a): Counter that indicates the number of meter power fails in the previous billing period.

1.3.2.10 Results for OM-SR11 Fraud detection

TC 5.4.11.1

Test has been performed successfully. In order to check the tampering management, TSEN (register used to manage the meter's behavior in case of fraud) was set equal to 255 and therefore, CAPE bit is activated when the terminal cover is removed.

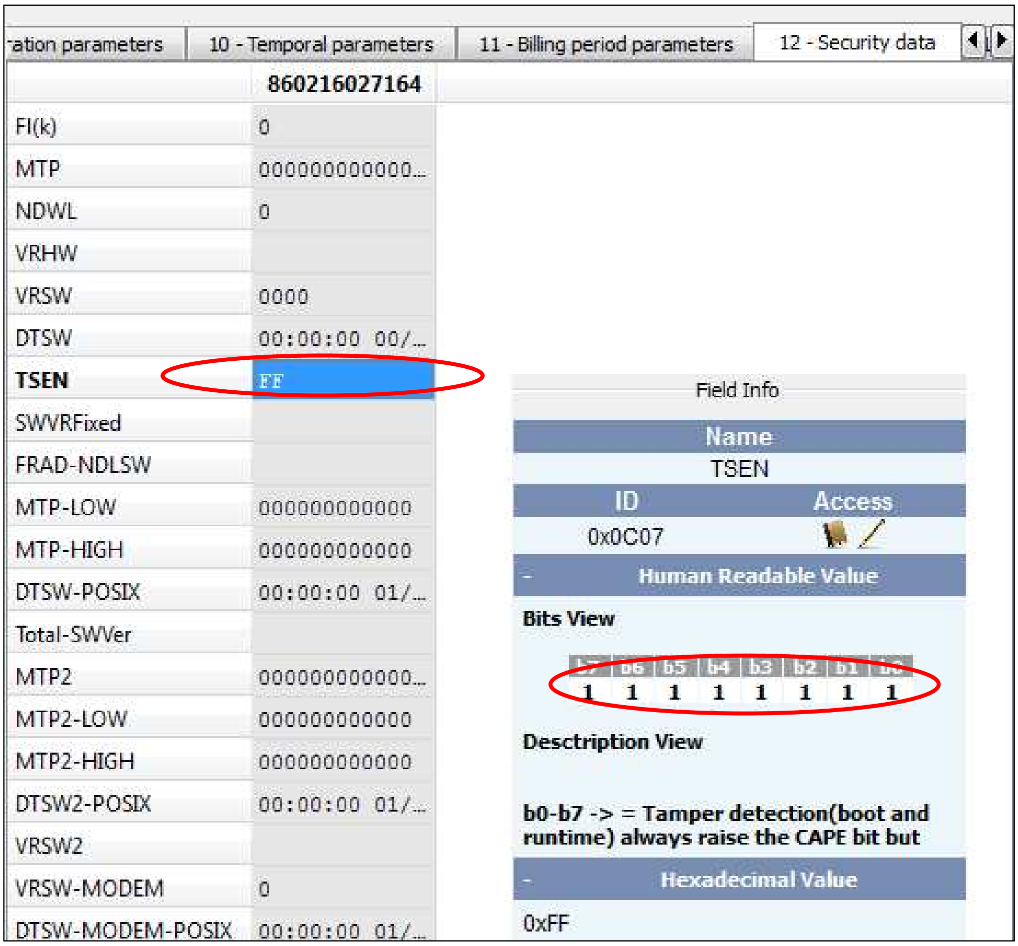


Figure 1-93: Security data. TSEN register was modified

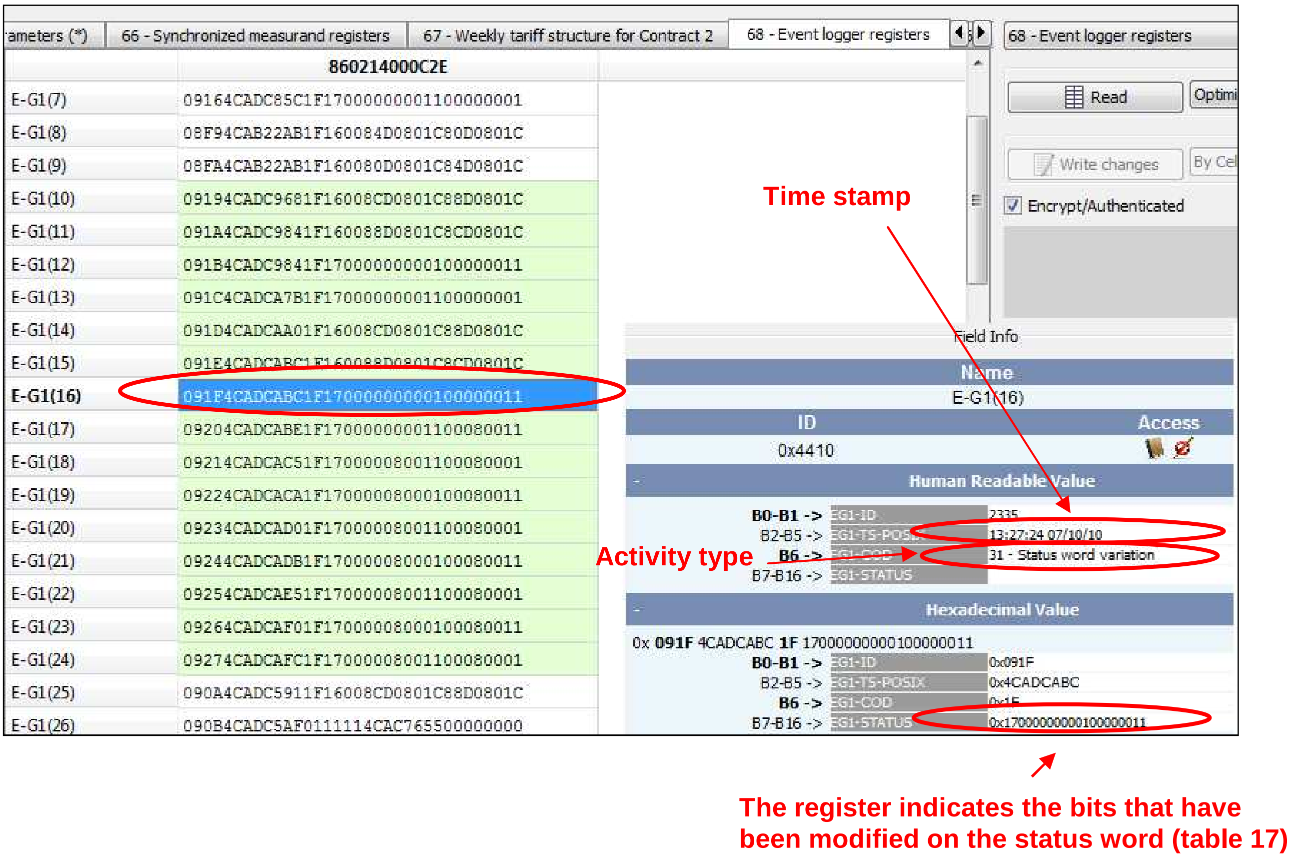


Figure 1-94: Events log table. Fraud detection

1.3.2.11 Results for OM-SR13 Load management

TC 5.4.13.1

Test has been performed successfully.

Figures below show the load profile, for active and reactive energy, stored by the meters from 22/03/2011 to 05/04/2011.

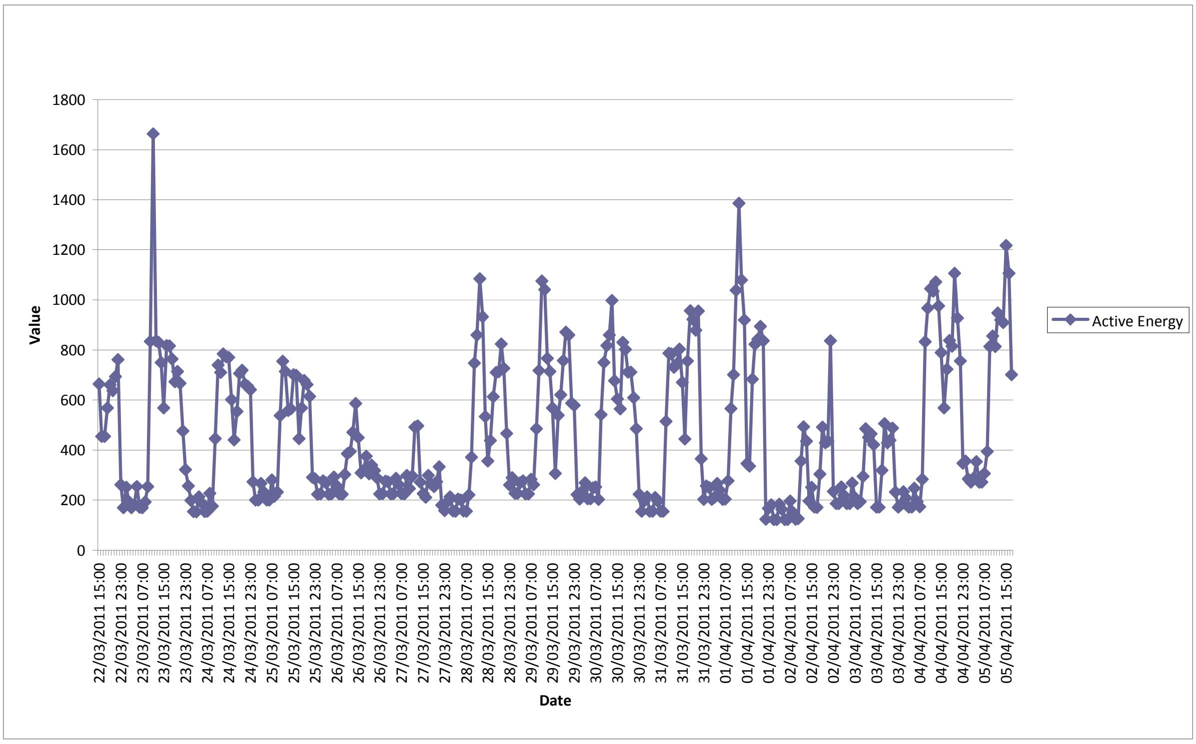


Figure 1-95: Active energy profile

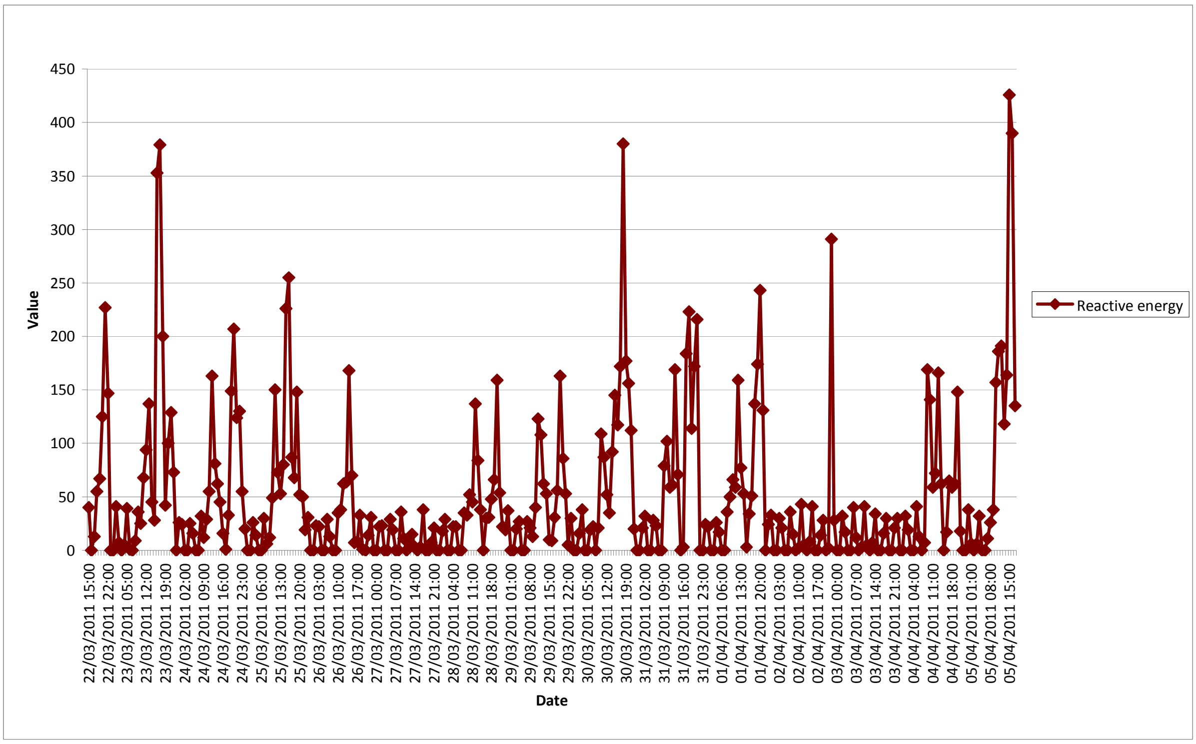


Figure 1-96: Reactive energy profile

1.4 Conclusion & Recommendation

All the tests performed both in the laboratory at RSE SpA in Milan and at the field in the premises of Endesa at Seville were completed successfully showing that METERS and MORE fulfilled the system requirements and functionalities defined in the Open Meter Project, including firmware update mechanism.

During METERS and MORE field tests participants and auditors had the opportunity to visit certified remotely operated meters, which are part of the Endesa's mass roll out, at customers premises to check the results obtained under real world conditions.

Furthermore, the executed interoperability tests have been successful.

The positive results of the detailed technical, functional and interoperability tests, show that the specifications of METERS and MORE technology are mature, implementable and ready for the standardization process. No further recommendations are necessary.

2 REPORT ON TEST RESULTS FOR DLMS - SFSK

2.1 Introduction

In this section, the results for the tests done at EDF, Clamart are described. In the following table, an overview is given of the performed test cases and the tested prototypes.
The test cases that were executed are described in [1] and [3].
The prototypes used are described in [2].

The DLMS-SFSK tests described in this report were performed from 28th February to 7th March 2011, at the premises of EDF Clamart, France.

The following persons were involved:

Test Engineers:	**Reviewers (7th March)**
G. Timmins, L+G	Gyozo Kmethy, DLMS UA
J. Orfeuil, ITRON	Auguste Ankou, ITRON
Support Engineers:	Michael Lanotte, ITRON
V. Godefroy, EDF	Rudy Nicar, ITRON
R. Sebastien, EDF	

Table 2-1: People involved in OPEN Meter testing of SFSK

All tests have been performed on in the period 28th February until 4 March. The following tests were repeated on the 7th March, in the presence of the auditors:

DLMS Functional tests:
- OM-SR2 remote tariff programming
- OM-SR3 meter reading (on demand)
- OM-SR4 meter reading (for billing)
- OM-SR7 clock synchronization
- OM-SR13 load profile management

DLMS Interoperability tests:
- All.

The picture below shows the location of the tests.

Figure 2-1: Testing and auditing SFSK at EDF, Clamart

In the next subsections, overviews are presented for the DLMS Functional tests, the DLMS Interoperability lab tests and DLMS Interoperability Field tests that were executed with the DLMS-SFSK meters.

It should be noted that for each test case performed, the test was performed twice:
- 1 DC from ITRON communicating with 1 meter from ITRON and 1 meter from LANDIS + GYR; + a spy tool from ITRON to catch PLC exchange
- 1 DC communicating with 1 meter from ITRON and 1 meter from LANDIS + GYR; + a spy tool from LANDIS + GYR to catch PLC exchange

This allows to have each time a test of all configurations.
In the next paragraphs, the test report will be organized as follows:
- 1 paragraph with the report of tests done with DC from ITRON (both meters)
- 1 paragraph with the report of tests done with DC from LANDIS + GYR (both meters)

It can occur that for some test, the data is not showed for both meters.
The reason is not that the test failed for one of the meters, but that interesting data is not easily readable, main of the time because it is mixed with other data, due to the fact that initial state of the meter before the test is not clean (Ie: logbook not cleared, load profile not cleared, …). When the test leads to the conclusion that something wrong occurred, it is mentioned in the report and explained in the recommendations.

2.1.1 DLMS lab tests for DLMS-SFSK

We can notice this part of the tests was not designed to be executed as interoperability tests, but only as functional tests.
The tests were however executed with all available combinations of DC and Meter.

Test cases:	Described in sections of [1]	Prototypes:	Described in sections of [2]
DLMS Functional Testing	4.4	LANDIS + GYR Meter	4.3.4.1
		ITRON Meter	4.3.3.1
		LANDIS + GYR DC	4.3.4.2
		ITRON DC	4.3.3.3

Table 2-2: Overview of OPEN Meter functional testing for DLMS-SFSK

NOTE: The DLMS Functional tests were executed according to the testcases as defined in the D4.1 amendment [3] instead of the original testcases in [1].

The table below presents an overview of the DLMS Interoperability test cases that were done for the DLMS – SFSK profile.

Test cases:	Described in sections of [1]	Prototypes:	Described in sections of [2]
DLMS Interoperability Lab tests	6.4.5	LANDIS + GYR Meter	4.3.4.1
		ITRON Meter	4.3.3.1
		LANDIS + GYR DC	4.3.4.2
		ITRON DC	4.3.3.3

Table 2-3: Overview of OPEN Meter interoperability lab testing for DLMS-SFSK

2.1.2 DLMS Field test for DLMS-SFSK

The table below presents an overview of the DLMS Interoperability Field tests that were done for the DLMS-SFSK profile.

Test cases:	Described in sections of [1]	Prototypes:	Described in sections of [2]
DLMS Interoperability Field tests	7.4.1	LANDIS + GYR Meter	4.3.4.1
		ITRON Meter	4.3.3.1
		LANDIS + GYR DC	4.3.4.2
		ITRON DC	4.3.3.3

Table 2-4: Overview of OPEN Meter Interoperability Field testing for DLMS-SFSK

2.1.3 Test tools used for the tests

The following test tools were used to realize both the laboratory- and field tests:

- CR4 administration to manage the DC

This is a web interface that allows to communicate with the DC through Ethernet cable, and allows to send requests to have PLC frames sent on the network.

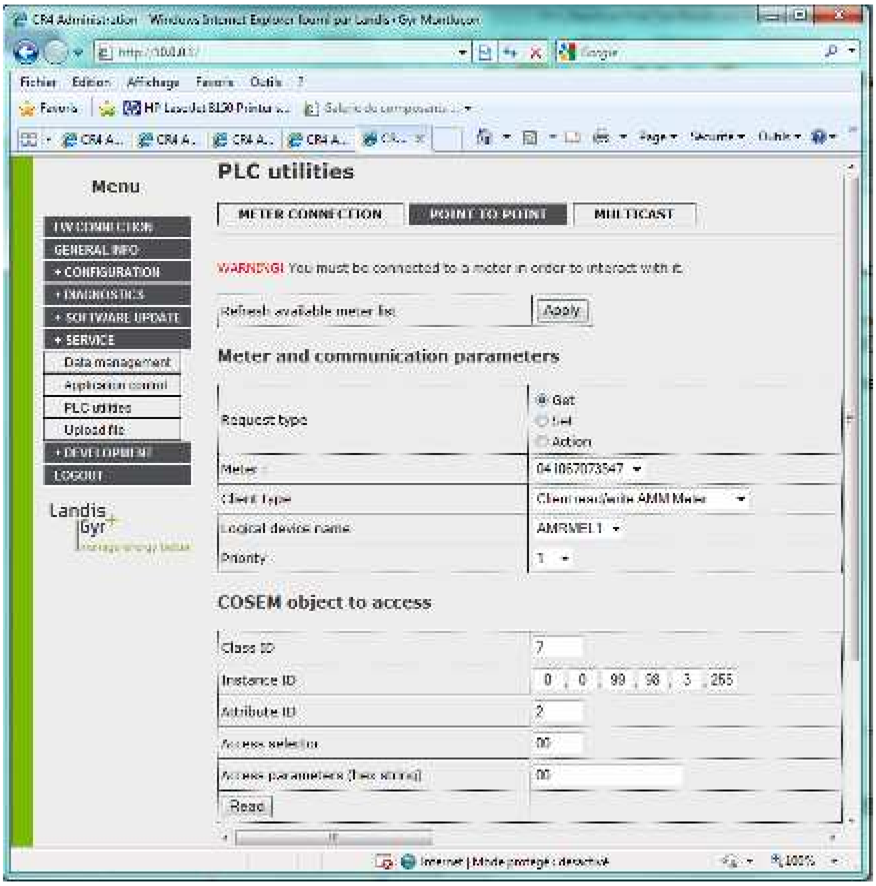


Figure 2-2: CR4 administration web interface

- PLC Spy to spy the PLC communication and capture exchange

The PLC Analyzer apparatus is plugged in parallel of the PLC equipments (DC and meters), and communicates with the computer through RS232 cable. The data is displayed on the main screen of the application, and it is possible to have a copy of the data (feature used in most of the tests). Data is most of the time enough interpreted so that its data is understandable.

Figure 2-3: PLC Analyzer

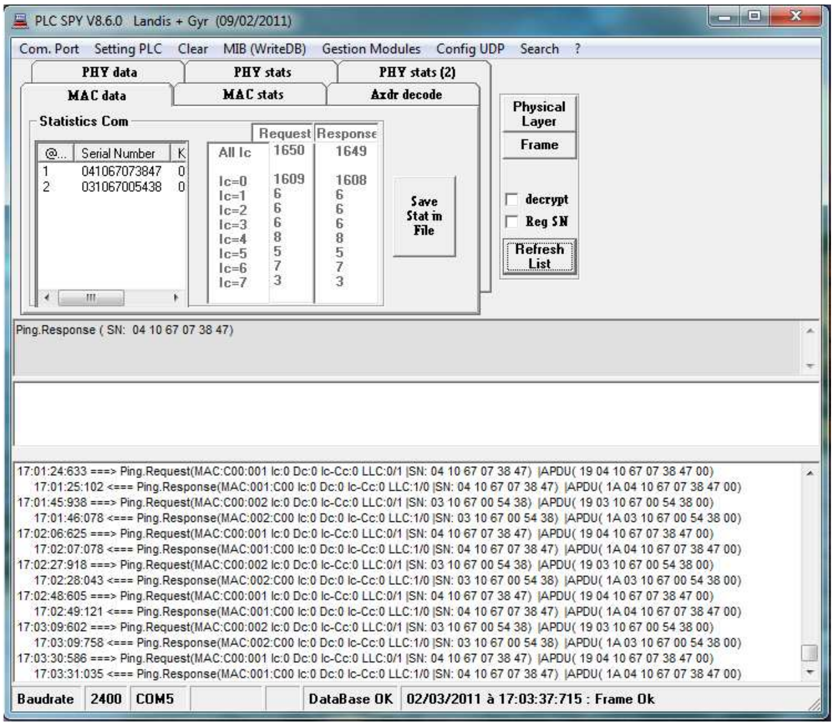


Figure 2-4: PLC Spy software interface

- AMM Euridis tool to communicate with the meters with Euridis port

An Euridis cell allows communication between USB port of the computer and Euridis port of the Linky Meter. The result of the exchange is displayed on the main screen, which we can copy and save. Data is displayed interpreted.

Figure 2-5: Euridis cell

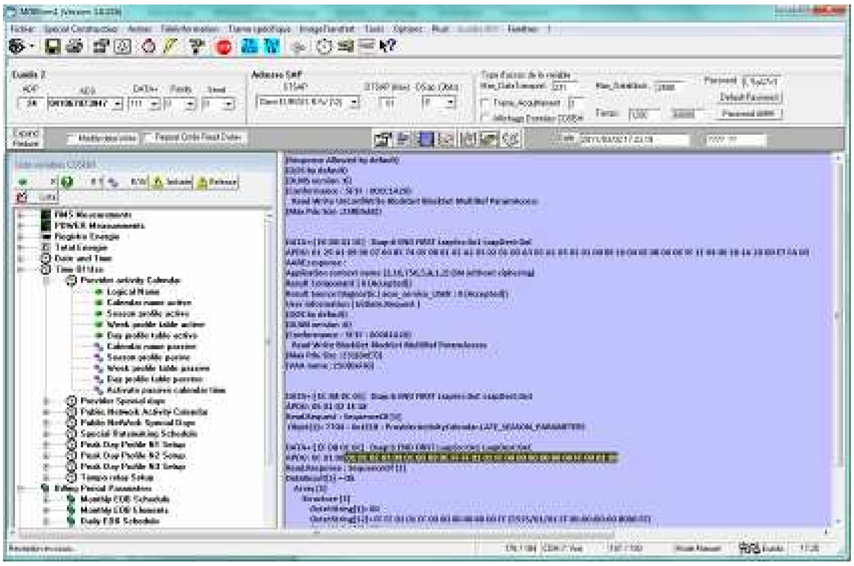


Figure 2-6: AMM Euridis tool PC interface

- LU ITRON to manage the DC

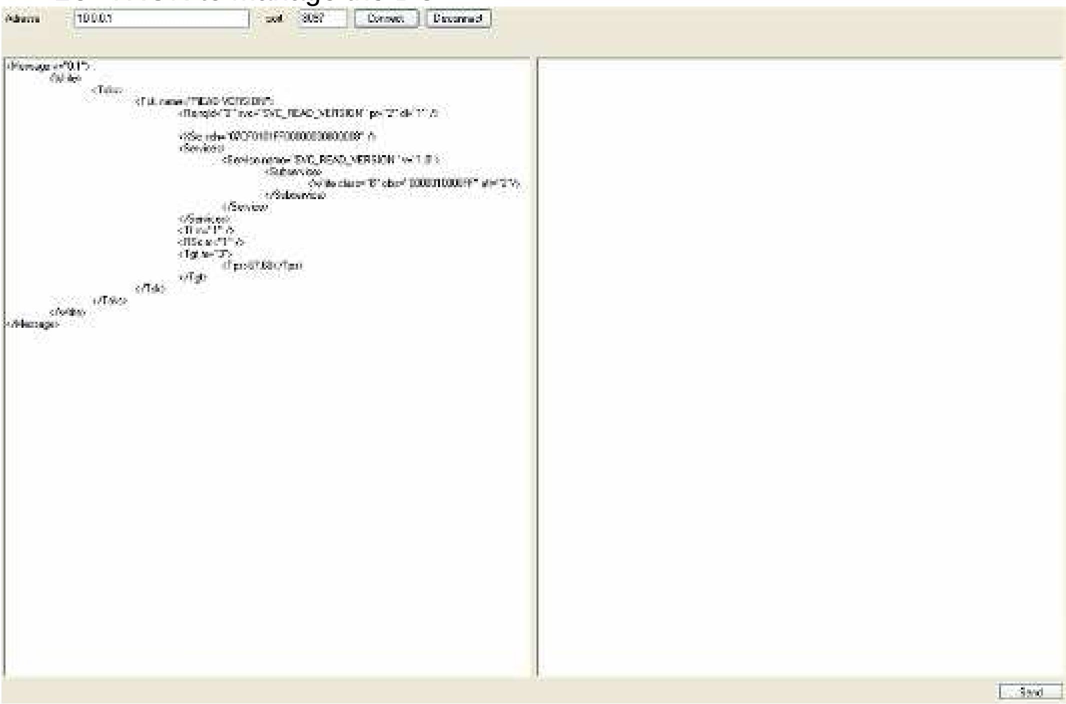


Figure 2-7: XML to LU tool (schedule read/write/action on objects)

- Human Readable Tool (ITRON) to interpret COSEM data

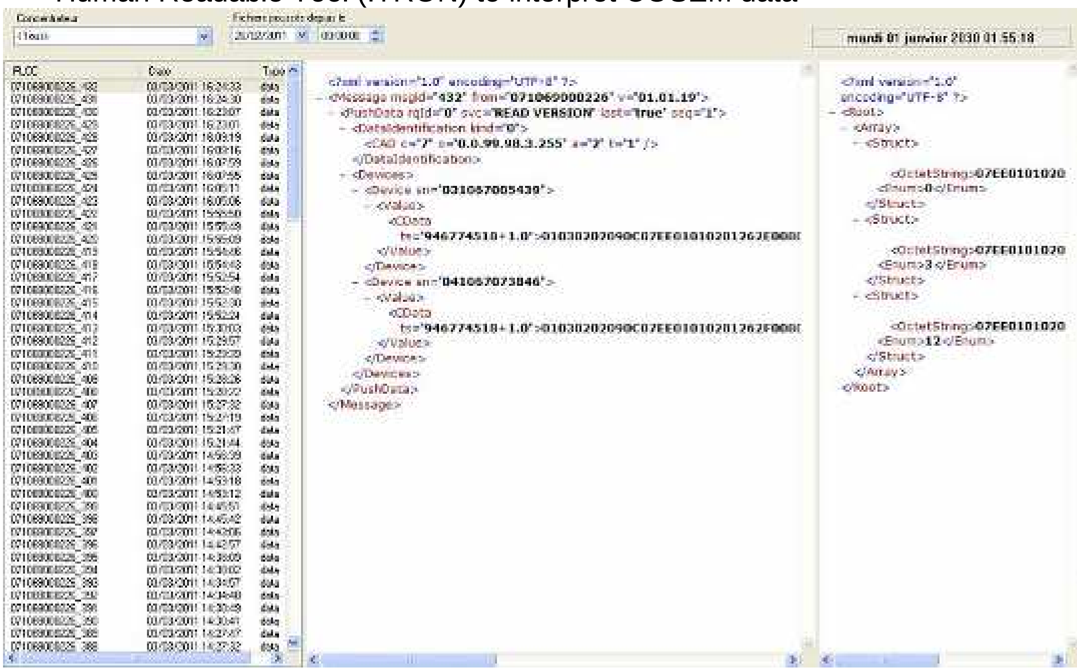


Figure 2-8: LU to "Human" (Decodes data coming back from DCu/meters)

- Cayox (ITRON) to communicate with the meters with Euridis port

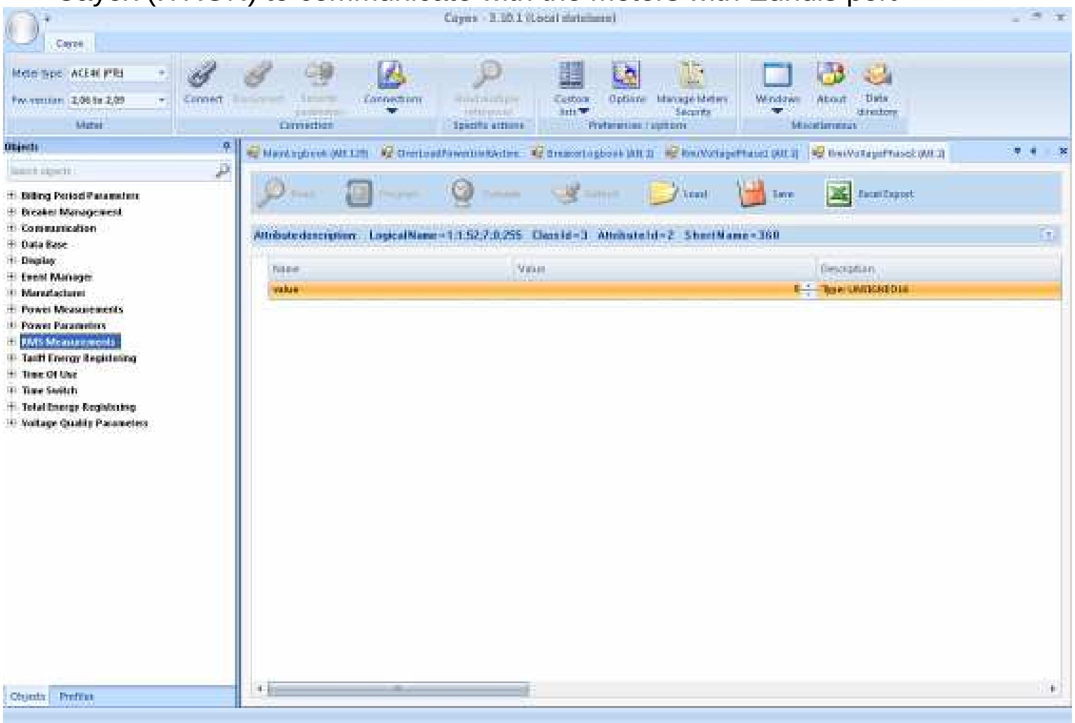


Figure 2-9: Cayox Interface

- PLAN monitor (ITRON) to spy the PLC communication and capture exchange

Figure 2-10: PLAN monitor interface

- Modem Euridis

Figure 2-11: Euridis cell

- Modem "Spy"

Figure 2-12: Modem Spy

2.2 Report on DLMS-SFSK Laboratory tests

In Annex C, the summary of functional test cases are presented.
These results are detailed in the following sections.

2.2.1 Test setup for DLMS and SFSK Interoperability laboratory tests

The test setup used for the DLMS and SFSK Interoperability tests is described in [2], section 3.5.3.
The setup installed for laboratory tests is shown on the pictures below.

Figure 2-13: Lab test facility at EDF, Clamart

2.2.2 Results of DLMS Functional tests

2.2.2.1 Remote Tariff Programming.

2.2.2.1.1 Tests with DC from ITRON

2.2.2.1.1.1 TC4.4.2.1

Step 1: Read actual meters readings
Result for ITRON and LANDIS + GYR meters

Data read:

- Import Active Energy Provider Tariff01
- Import Active Energy Provider Tariff02

Step 2: Shift tariff at designated day
Result for ITRON and LANDIS + GYR meters
Logs of programming of a tariff change the hour after the current one (in our case, 15h00 at 14h30).
Current tariff is 2. Program tariff change is 1.
Conclusion: the test is successful on both meters..

Step 3,4&5:
Result for ITRON and LANDIS + GYR meters
Read of the meter readings after scheduling the tariff change.
Both ITRON and LANDIS + GYR meters go on counting in tariff 2.
Conclusion: the test is successful on both meters.

2.2.2.1.2 Tests with DC from LANDIS + GYR

2.2.2.1.2.1 TC 4.4.2.1

Step 2: Programming a Passedive activity calendar with an activation date:

The old calendar has one single tarification, which is tariff 1.
The new calendar has one single tarification, which is tariff 2.
The new calendar is planned to be activated at 17h50.

Tariff 1 energy register is read with SN 2024.
Tariff 2 energy register is read with SN 2288.

Step 3: Reading of the tariff energy registers before the calendar shift:
Step 4: Reading of the tariff energy registers after the calendar shift:

Evolution of the energy registers before and after the calendar change

	LANDIS + GYR Meter		ITRON Meter	
	Tariff 1	Tariff 2	Tariff 1	Tariff 2
Before calendar	Increase	Don't move	Don't move	Increase

change	(228 ➔ 229)	(value = 0)	(value = 149)	(17 ➔ 18)
After calendar change	Don't move (value = 229)	Increase (0 ➔ 5)	Don't move (value = 149)	Increase (18 ➔ 24)

We can notice here that the behavior of the 2 meters is not exactly the same. See conclusion for this test case below or refer to note in 2.4.2.
Step 5: Reading the active calendar name just after the calendar change:

For both meters:
Old calendar name: 42 41 53 45 20 20 20 20 20 20 20 20 20 20 20 20 00 FF FFFFFF
New calendar name: 42 41 53 45 20 20 20 20 20 20 20 20 20 20 20 **21** 00 FF FFFFFF

2.2.2.1.3 Conclusion for remote tariff programming

ITRON meter doesn't take into account minutes for scheduled tariff change. For example, if you program a tariff change at 14h30 while it's currently 14h20, the change will be effective immediately, because the date taken into account is 14h00 (past date).

For a delayed activation, you have, in this example, to program at least 15h00.
This is a tolerance granted by the client for this meter which is small series.

Apart from that point, the test is successful on both meters. Tariff programming is available,and the meters record the energy in the rate registers as defined by the calendar and the clock.

2.2.2.2 Meter Reading (On Demand).

2.2.2.2.1 Tests with DC from ITRON

2.2.2.2.1.1 TC4.4.3.1

Step 1: Program date to March the 1st of 2011 and time to 00h01

Step 2: Result for both ITRON and LANDIS + GYR Meter

Human Readable results
2482 Wh for ITRON Meter
0 Wh for LANDIS + GYR Meter

Step 3: Load applied: ~250W

Step 4: Read after 10 minutes

ITRON Meter:
Human Readable Results for ITRON Meter
2525 Wh => 43 Wh have been consumed during about 10 minutes

Step 5: Program date to March the 1st of 2011 and time to 23h58 =>Done

Step 6: Read billing period data

Result for LANDIS + GYR and ITRON Meters
ITRON Human Readable values (2010 values removed)

- Conclusion: the test is successful, in both meters. The end of billing data still frozen for all the rates
- The actual data continuously increase according to the rate and the load.

Step 7: Main Logbook reading

For ITRON and LANDIS + GYR Meter

ITRON Human Readable Values (2010 values removed)

Step 8: Confirmed

Conclusion: the test is successful, in both meters.

- No specific additional element in the logbook.
- Actual meter read data are consistent with the display and test conditions.

2.2.2.2.2 Tests with DC from LANDIS + GYR

2.2.2.2.2.1 TC 4.4.3.1

Step 1: Because the time has no influence on the result of the test, it was set to the current date and time and not to 00:01.

Step 2: Reading of instantaneous energy values:

For both meters, all the energy registers value is 0.

Step 4: Reading of instantaneous energy values: value of active energy incremented

Step 5: Program date to March the 1st of 2011 and time to 23h58 =>Done

A load of 150W has been applied to both meters during 12-13 minutes.
The total import active energy register increased by 32Wh, which match the theoretical consumption for this period (150W during 13 minutes is 32,5Wh)

Step 6: Daily EOB Element from 02/03 saved correct value of energy
Daily EOB elements have been saved correctly (32Wh in the active energy register).

Step 7: Logbook contains Reset and time shift events
The logbook contains 3 events:

- Reset
- Old date
- New date

2.2.2.3 Meter Reading (For Billing).

2.2.2.3.1 Tests with DC from ITRON

2.2.2.3.1.1 TC 4.4.4.1

The organization of the test sequence does not allow the entire execution of this test. It will need a specific meter on the load during 3 full days before. Therefore the test was skipped.

2.2.2.3.1.2 TC 4.4.4.2

Step 1: Program Async EOB period to 01/01/2030

Result for ITRON and LANDIS + GYR meters
Step 2: Not necessary

Step 3: Set time to 31/12/2029 23:55

Result for ITRON and LANDIS + GYR meters

Step 4: Read Async EOB

Result for ITRON and LANDIS + GYR meters: An additional end of billing data can be seen inside the EOB profile at the date of 01/01/2030 at 00:00 as expected.

Step 5&6: Impossible with Async EOB

Step 7: Set time to 31/12/2000 23:55

Result for ITRON and LANDIS + GYR meters

Step 8: Set Monthly EOB on the 1st of each month

Result for ITRON and LANDIS + GYR meters

Step 9: Not necessary

Step 10: Set time to 01/01/2002 16:00

Result for ITRON and LANDIS + GYR meters

Step 11: Read Monthly EOB

Result for ITRON and LANDIS + GYR meters: : An additional end of billing data can be seen inside the EOB profile at the date of 01/01/2002 at 16:00 as expected.
Conclusion. The test was successful in both meters.

2.2.2.3.1.3 Test TC 4.4.4.3

Step 1&2: Selective access with specified period before the end of the period

Result for ITRON and LANDIS + GYR meters: No EOB data element is sent back.
Conclusion: The test was successful in both meters

2.2.2.3.2 Tests with DC from LANDIS + GYR

2.2.2.3.2.1 TC 4.4.4.1

This test includes a loading of the meters during 3 days, in addition to all the manipulations to be done on it. Therefore, this test was not realized due to time shortage.

2.2.2.3.2.2 TC4.4.4.2

Step 4: 1 single Asynchronous EOB Element generated on 01/01/2030:
The Asynchronous EOB element has correctly been generated on 01/01/2030.

Step 11: 1 single Monthly EOB Element generated on 01/01/2002:
After the time change, 1 single monthly EOB element has been generated on 01/01/2002.

2.2.2.3.2.3 TC4.4.4.3

Step 2: 1 meter answers with no elements: (NB performed with Euridis tool)

If there is no element stored in the time interval specified, the meter answers with no element.

2.2.2.4 Remote Disconnection and Reconnection.

The depth of the Breaker logbook is 20 events. 1 event is recorded when the breaker logbook is reset, 1 event is recorded when the breaker logbook is disconnected, 1 event is recorded when the breaker logbook is reconnected.

2.2.2.4.1 Tests with DC from ITRON

2.2.2.4.1.1 TC 4.4.5.1

Step 2: Disconnect the meters

Result for ITRON and LANDIS + GYR Meters: Meters are disconnected.

Step 3: Check the breaker logbook

Result for ITRON and LANDIS + GYR Meters: Disconnect logbook available and can be read.

Step 4&5: Read the breaker status (breaker control mode)

Results for ITRON and LANDIS + GYR Meters: Breaker status is DISCONNECTED in both meters

Step 6: Reconnect the breaker

Result for ITRON and LANDIS + GYR: Breaker is connected

Step 7: Check the breaker logbook

Result for ITRON and LANDIS + GYR: one element is inserted inside the breaker logbook at the moment of the operation.

Step 8: Check visual state of the switch

OK (disconnected)

Step 9: Check the switch status (control state)

Result for ITRON and LANDIS + GYR: in both meter, the breaker status is READY FOR CONNECTION

Step 10: Remote connect while breaker connected

Result for ITRON and LANDIS + GYR: successful programming. Nothing change on both meters

Step 11: Check visual state of the switch

OK (output is FALSE)

Step 12: Check the switch status (control state)

Result for ITRON and LANDIS + GYR: Control state still at READY FOR CONNECTION

Step 13: Repeat the tests above

Step 14: Check Breaker Logbook

Result for ITRON and LANDIS + GYR. Same result than above

Total is 7 disconnection/connection. First value (0) is the erase of the logbook before starting the test.

Result in Itron Logging :Logbook contains 7 disco/reco events

Result in Landis+Gyr logging :Logbook contains 7 disco/reco events

Step 15: Read Breaker Opening Counter

Result on ITRON and LANDIS + GYR: Connection and disconnection are correctly recorded inside the logbook. The output state still at FALSE since the manual connection is not performed. Control output switch between DISCONNECTED and READY FOR CONNECTION.

2.2.2.4.1.2 TC 4.4.5.2

Step 1: Synchronize meters on December the 31st of 2009

Results for ITRON and LANDIS + GYR Meters

Step 2: Schedule a disconnection on January the 1st of 2010

Results for ITRON and LANDIS + GYR Meters

OK on LANDIS + GYR
OK on ITRON

Step 3: Visual Check of breaker state

Disconnected on LANDIS + GYR
Disconnected on ITRON

Step 4: Breaker Status (control state)

Results for ITRON and LANDIS + GYR Meters:

- Read Control State

ITRON Meter indicates READY FOR CONNECTION. See conclusion for this test case below or refer to note in 2.4.1. Output state is OK.
OK on LANDIS + GYR Meter

- Read Output State:

OK on ITRON and LANDIS + GYR meters

Step 5&6: Check that you can't connect manually
OK on both meters

Step 7:
OK on both meters

Step 8: Check if measured values do not change
Result for ITRON and LANDIS + GYR meter: Even the load is present nothing is recorded inside the cumulative register.

Step 9: Breaker Status (control state)

Result for ITRON and LANDIS + GYR Meter: OK

Step 10: Check values are changing:

Result for ITRON and LANDIS + GYR meter
The reading values are OK in both meters. The meter moved to connected state and the value of cumulative register restarts increasing.

2.2.2.4.2 Tests with DC from LANDIS + GYR

2.2.2.4.2.1 TC4.4.5.1

Step 2: Disconnect breaker:
Step 3: breaker logbook contains a disconnect event:
A disconnect event has been generated in the breaker logbook with correct timestamp.

Step 5: read breaker state = disconnected:
Breaker state is correct when reading.

Step 6: Reconnect breaker:

Step 7: breaker logbook contains a reconnect event:
A connect event has been generated in the breaker logbook with correct timestamp.

Step 9: read breaker state = connected:
Breaker state is correct when reading.

Step 10: Connect breaker.

Step 12: read breaker state = connected:
Breaker state is correct when reading.

Step 13: We generated a total of 4 disconnections / reconnections

Step 14: After several disconnect and reconnect, the breaker logbook contains that many events:
The breaker logbook contains 9 events:
- Reset of the logbook
- 4 disconnect events with timestamp
- 4 connect event with timestamp.

Step 15: Breaker opening counter contains 4 events:

2.2.2.4.2.2 TC 4.4.5.2

Step 2: Program power cut schedule:
We program a power cut from 01/01/2010 at 0h00 to 01/01/2010 at 0h10.

Step 4: Verify correctness of breaker state:
Result of the reading of the breaker state during scheduled disconnection:

	LANDIS + GYR Meter	ITRON Meter
Breaker control state	Disconnected (0)	Connected (1)
Breaker output state	Opened (0)	Opened (0)

We can notice here that the behavior of the 2 meters is not the same. See conclusion for this test case below or refer to note in 2.4.1.

Step 8: Energy registers do not vary while breaker is opened:
At 1 minute interval, the energy registers do not vary during the scheduled disconnection of the breaker.

Step 10: Energy registers vary while breaker is closed:
At 1 minute interval, the energy registers increase when the scheduled disconnection of the breaker is finished.

2.2.2.4.2.3 Conclusion for this test case

After the breaker opens on power cut schedule, the breaker control state indicates Disconected (0) for LANDIS + GYR meters and Connected (1) for ITRON meters.
The breaker output state, indicating the physical real state of the breaker, indicates Opened (0) for meters of both manufacturers.

2.2.2.5 Power Control.

2.2.2.5.1 Tests with DC from ITRON

2.2.2.5.1.1 TC4.4.6.1

Step 1: Program the OverLoad power limit to 10VA:

Result for ITRON and LANDIS + GYR meters. OK

Step 2&3: N/A

Step 4: Apply a load exceeding the contractual power level: OK

Step 5: Check the supply is stopped within a second:

OK. Disconnection occurs and the meter stops recording the energy

Step 6: Check breaker logbook

Result for ITRON and LANDIS + GYR meters: an additional disconnection element is inserted.

Code 3 means "breaker opened because of overload power"

Step 7: Visual check that the supply can be restarted manually

On ITRON Meter
The light indicating that we are allowed to reconnect manually pops up after a tempo (30 secs). Probably to protect the breaker.
After that, we are able to reconnect manually.

On LANDIS + GYR Meter
The light indicating that we are allowed to reconnect manually pops up immediately.

We can notice here that the behavior of the 2 meters is not the same. See conclusion for this test case below or refer to note in 2.4.3.

Step 8: Check the supply is quickly restored

Both on ITRON and LANDIS + GYR meters, the supply is restored immediately after pushing the button to reconnect.

Step 9: Check the breaker logbook

Result for ITRON and LANDIS + GYR meters:

Code 12 was set: "Manual reconnection of the breaker"

2.2.2.5.1.2 TC 4.4.6.2

Step 1: Restore contractual power level to nominal value (9kVA here)

Result for ITRON and LANDIS + GYR meters. OK

Step 2: N/A

Step 3: Check the threshold is applied on the meter within 5 seconds

OK

Step 4: N/A

Step 5: Apply a load exceeding the previously programmed contractual power level

OK

Step 6: Check the value of the threshold

Result for ITRON and LANDIS + GYR meters

The value is correct (the default reprogrammed one at step 1). We can read in red in the logs the retrieved value. 0101122328 (array of 1 element, integer16 equal to 0x2328 = 9000).

2.2.2.5.1.3 TC 4.4.6.3

As mentioned in [2], this test is not applicable.

2.2.2.5.1.4 TC4.4.6.4

As mentioned in [2], this test is not applicable.

2.2.2.5.2 Tests with DC from LANDIS + GYR

2.2.2.5.2.1 TC 4.4.6.1

Step 1: Program the OverLoad power limit to 10VA:

Step 4: TheOverLoad power limit has been set to 10VA and we apply a load of 100W. The breaker opens automatically after a few seconds.

Step 5: Verify correctness of breaker state:
Result of the reading of the breaker state after breaker opened on overload:

	LANDIS + GYR Meter	ITRON Meter
Breaker control state	Ready to connect (2)	Connected (1)
Breaker output state	Opened (0)	Opened (0)

We notice a different behavior for both meters. See 2.4.1 Breaker output state / control state for more details.

Step 6: Breaker logbook contains disconnect event:
A disconnect event has been generated in the breaker logbook with correct timestamp.

Step 9: Breaker logbook contains reconnect event:
A reconnect event has been generated in the breaker logbook with correct timestamp.
Because the threshold was not modified before the breaker was rearmed, it automatically disconnected a short time after. So, an other disconnect event can be also read in the logbook.

The behavior of the 2 meters after a manual reconnection is not the same. See 2.4.3 Manual reconnection after Overload power.

2.2.2.5.2.2 TC 4.4.6.2

Step 1: Setting the OverLoad power limit to its nominal value:
We write 9000VA to the Overload power limit (nominal value); no method “clear” is available on this object.

Step 5: We apply the same load of 100W than previously, and notice that the breaker remains closed.

Step 6: OverLoad power limit is correct:
The writing of the Overload power limit was successful, we can retrieve 9000VA when reading the register.

2.2.2.5.2.3 TC 4.4.6.3

This test is not applicable.

2.2.2.5.2.4 TC 4.4.6.4

This test is not applicable.

2.2.2.5.3 Conclusion for Power control tests

After the breaker opens on over-power, the breaker control state indicates Disconected (2) for LANDIS + GYR meters and Connected (1) for ITRON meters.
The breaker output state, indicating the physical real state of the breaker, indicates Opened (0) for meters of both manufacturers.

The behavior of the 2 meters after a manual reconnection is not the same:
- ITRON meter reconnects, remain connected for 10-20 seconds, and disconnects after that delay if the load is still present. (the breaker is programmed so that it can't commute very fast, so there is that delay of 10 seconds)
- LANDIS + GYR meter reconnects and disconnects almost immediately, even if the load is not present any more. We have to wait a certain time to be able to reconnect the breaker definitively (the measurement of the overload is programmed as integration of the power, which makes it necessary to wait that the average power measured on the last x seconds PASSED below the overload power threshold).

2.2.2.6 Clock Synchronization.

2.2.2.6.1 Tests with DC from ITRON

2.2.2.6.1.1 TC4.4.7.1

Step 1: Set a date/time into the meter and note the current time of the computer (hh:mm:ss)

Step 2: Wait 12 hours

Step 3: Check the date/time of the meters. Compare to the real date/time of the computer and calculate the deviation.

ITRON meters deviated of 2 seconds.
LANDIS + GYR meters deviated of 4 seconds

Both ITRON and LANDIS + GYR meters deviated of less than 10 seconds. The accuracy of the clock is better but here communication time has to be taken into account for the test results

2.2.2.6.1.2 TC4.4.7.2

Step 1: The maximum deviation is hard coded and set to 10 seconds.

Step 2: Change clock for more than 10 seconds

Initial date of meters: 01/01/2030 00h00
Set date to 01/01/2030 01h00

Step 3,4,5&6: Check an error has been generated

Result for ITRON and LANDIS + GYR meters

2 events are generated in the Main Logbook (43&44).

For both ITRON and LANDIS + GYR meters, we can see the events 43 at 01/01/2030 00h02m and the events 44 at 01/01/2030 01h00

2.2.2.6.2 Tests with DC from LANDIS + GYR

2.2.2.6.2.1 TC 4.4.7.1

Step 1: The maximum deviation from national standard time is a constant of 10 seconds, it can't be programmed on the meters chosen for Open Meter tests.

Step 3: We set the time before leaving the office, and let the meters for all the night (14-15 hours).

Step 4: The time deviation noticed after 1 night was -2 seconds on both meters. This fits the margin of 10 seconds.

2.2.2.6.2.2 TC 4.4.7.2

Step 2: We make sure that the current difference between date and time in the meters and current standard time is more than 10 seconds, and then we set the time.

Step 3: a time change event is recorded in the main logbook:
The main logbook now contains 3 events:

- Reset
- Old date
- New date

2.2.2.7 Remote Firmware Update.

2.2.2.7.1 Tests with DC from ITRON

2.2.2.7.1.1 TC4.4.8.1

Unable to run the test as we were unable to schedule an activation date of the new FW (whatsoever by scheduling a date in the meter or by sending the activate command later)

2.2.2.7.1.2 TC 4.4.8.2

<u>Test only on ITRON Meter</u>

Step 1&2: Please, refer to 6.4.5.2 for details about the transfer.

Step 3: Read event log and check “verification of the firmware” event is inside

The logbook shows:

Structure		Index 64
capture_time	01/01/2010 01:52:12	CurrentDateAndTime (Att. 2)
event_code	Code DSP prêt pour activation (170)	ManufacturerEventCode (Att. 2)

Step 4: Check the new FW is deployed immediately

Check Step 6 screenshot. The DSP is activated 4 minutes after (In case of NLR download, it is activated immediately after)

Step 5: Verify the state of the equipment is not modified after the FW upgrade

Load profile reading before

Step 6: Read event log and check “activation of the firmware” event is inside

Tests are OK. The load profile as already read before the firmware upgrade still readable with the same values back.

2.2.2.7.1.3 TC 4.4.8.3

<u>Test only on ITRON meters</u>

Step 1: Refer to 6.4.5.2

Step 2&3: Check logs and verify an error has been generated

Image Transfer Status shows

Attribute description: LogicalName=0;0;44;0;0;255 ClassId=18 AttributeId=6 ShortName=19736

Name	Value	Description
image_transfer_status	IMAGE_VERIFICATION_FAILED (4)	

, and the main logbook shows

Structure		Index 73
capture_time	01/01/2010 07:37:48	CurrentDateAndTime (Att. 2)
event_code	Erreur de la vérification du code DSP (168)	ManufacturerEventCode (Att. 2)

Step 4: Version discarded by the meter

OK
Status before and status after shows that the firmware update is not performed, and the firmware version still the same.

2.2.2.7.2 Tests with DC from LANDIS + GYR

The tools available for Open Meter tests do not allow to perform tests on Firmware download. This test could not be executed.

2.2.2.8 Interruption Information.

2.2.2.8.1 Tests with DC from ITRON

2.2.2.8.1.1 TC 4.4.10.1

Step 1: Set long power failure threshold to 3 minutes

Result for ITRON and LANDIS + GYR meters: OK.

Step 2: Power off the meters for 2 minutes and 45 secs.

Result for ITRON and LANDIS + GYR meters: OK

Step 3: Check voltage cut profile

Result for ITRON and LANDIS + GYR meters:

The logbook only contains the information of reset (done before the test)

Step 4&5: Power off the meters for 3 minutes and 15 secs and restore the voltage.

OK

Step 6&7&8: Check voltage cut profile

Both LANDIS + GYR and ITRON logbooks contain the following 2 information:

- Power Cut phase 1 (code 51)
- Power Back phase 1 (code 52)

The beginning of the power cut was correctly logged (07DB0303040A073000800000) as long as the end of it (07DB0303040A0B0700800000).

Step 11: OK (at power ON)

2.2.2.8.1.2 TC 4.4.10.2

Step 1: The depth of the Voltage Cut Logbook is minimum 100, 2 events are recorded each time there is a power cut (start/stop)

Step 2: Power Off/On the meters and introduce
7 short outages
12 long outages

Step 3: Read logbook

12 long power outages have been generated.
Therefore, we should have in the logbook 27 elements: 12*2=24 events for this test plus 2 events of the previous test plus 1 event due to the reset of the logbooks before theses tests.

Results for ITRON: 27 entries
Results for LANDIS + GYR: 27 entries

Step 4: Getting the 10 most recent log outages

Performing the same request as the one in step 3 with a selective access gives us the 10 most recent log outages.

Step 5: Not possible

Step 6: OK (at power ON)

2.2.2.8.2 Tests with DC from LANDIS + GYR

2.2.2.8.2.1 TC 4.4.10.1

Step 1: We set the threshold for power failure detection to 3 minutes.

Step 2: We power down the meter during 2 minutes 45 seconds.

Step 3: Voltage cut logbook contains no event:
After a power cut of 2 minutes 45 seconds, no power cut event has been registered in the voltage cut logbook.

Step 4: We power down the meter during 3 minutes 15 seconds.

Step 6: Voltage cut logbook contains a start and a stop of voltage cut:

After a power cut of 3 minutes 15 seconds, A power failure start and a power failure stop have been recorded in the logbook.

2.2.2.8.2.2 TC 4.4.10.2

Step 1: The depth of the voltage cut logbook is 100 events. 1 event is stored at reset, and 2 events are stored each time a power failure occurs (1 for start of power failure, one for stop)

Step 2: after shortening the time detection of power failure to 1 second, we provoke 8 power cut.

Step 3: Reading Voltage cut logbook:
Voltage cut logbook contains data for 8 power failure (a reset event, 8 start of power failure and 8 end of power failure).

2.2.2.9 Fraud Detection.

2.2.2.9.1 Tests with DC from ITRON

2.2.2.9.1.1 TC 4.4.11.1

As mentioned in [2], this test is not applicable.

2.2.2.9.1.2 TC 4.4.11.2

Step 1: The depth of the Cover Logbook is minimum 9, 2 events are recorded each time the cover is moved (open/close).

Step 2: Open the cover

.OK

Step 3: Read the Alarm Register

Result for LANDIS + GYR meter when opening the cover:
The error register and alarm register show the occurrence of the cover opening.

Result for ITRON meter when opening the cover:
The error register and alarm register show the occurrence of the cover opening.

For both ITRON and LANDIS + GYR meters, we can see in the logs that just before the clear alarm request, the alarm register contains the information of a cover opening.

Step 4: Read the Cover Logbook

Step 5: N/A

Step 6: Open cover 2 times
OK

Step 7: Read logbook and check 2 events have been generated
Cover opening events (OPEN & CLOSE) and their occurrence date are inserted in the logbook for both meters.

2.2.2.9.2 Tests with DC from LANDIS + GYR

2.2.2.9.2.1 TC 4.4.11.1

As mentioned in [2], this test is not applicable.

2.2.2.9.2.2 TC 4.4.11.1

Step 2: We simulate a cover removed event, on both meters.

Step 3:Reading of alarm register:
Alarm register contains a Cover removed event. This event is automatically cleared by the DC (internal setting of the DC)

Step 4: Reading Cover logbook:
Cover logbook contains a Cover removed event.

Step 6: we simulate 2 more Cover Opening on one of the meters. Because of the difficulty to generate this kind of events, it was not done on both meters.

Step 7: Reading the cover opening counter:
The cover opening counter has incremented by 3, which corresponds to the number of opening performed on that meter.

2.2.2.10 Load Profile Management

2.2.2.10.1 Tests with DC from ITRON

2.2.2.10.1.1 TC 4.4.13.1

Step 1: Ensure integration period is 10 minutes

Result for ITRON and LANDIS + GYR Meters: the reading provides an integration period of 10mn.

Step 2: Program clocks at 00h00

Result for ITRON and LANDIS + GYR Meters
Clock setting is OK for both.

Step 3: Load meters with ~250W load during 20 minutes =>Done

Step 4: Read Load profile

Result for ITRON and LANDIS + GYR meters with selective access (from 00h00 to 00h21)
Two additional elements are inserted inside the logbook.

Step 5: Sync date/time 2203 integration periods after

2203*10=22030 minutes => 15d07h10m. Therefore, we program 15d07h30m.

Step 6: Read Load Profile

Result for ITRON and LANDIS + GYR meters with selective access (from March the 1st 00h00 to March the 15th 07h40)

Steps 9&10: Check LP format
Results:
There is not 2208 elements in the LP.
There is only the measured values plus marking points indicating the changes of dates.

2.2.2.10.2 Tests with DC from LANDIS + GYR

2.2.2.10.2.1 TC 4.4.13.1

Step 1: The load profile integration time is 10 minutes.

Step 3:we connect a load for nearly 25 minutes.

Step 4: Load Profile Reading:
The load profile contains 4 elements:
- Date (new date, generated at date/time change)
- Time (new time, generated at date/time change)
- Truncated power (power consumption date/time change and between end of period)
- Power (normal power consumption in 10 minutes)

Step 5: we set date time to 15 days in future

Step 6: Load Profile Reading:
The load profile contains 11 elements (4 + 7 new elements)
- Power (normal power consumption in 10 minutes)
- Date (old date, generated at date/time change)
- Time (old time, generated at date/time change)
- Truncated power (power consumption between end of period and date/time change)
- Date (new date, generated at date/time change)
- Time (new time, generated at date/time change)
- Truncated power (power consumption date/time change and between end of period)

Step 7:we connect a load for nearly 25 minutes.

Step 8: Load Profile Reading:
The load profile contains 13 elements (11 + 2 new elements)
- Power (normal power consumption in 10 minutes)
- Power (normal power consumption in 10 minutes)

2.2.2.11 TC 4.4.19 – Power Quality Management

2.2.2.11.1 Tests with DC from ITRON

2.2.2.11.1.1 TC 4.4.19.1

Step 1: Set Nominal voltage to 230V:OK

Steps 2&3: Set integration period for sag and swell to 1 minute: OK

Step 4: Set threshold for voltage sags to 7% below Nominal Voltage: OK

Step 5: Set threshold for long power failure to 3 minutes: OK

Step 6&7&8: Reduce the voltage to 216V. Wait 36 secs (3/5*60s) and restore the voltage to 230V.Read the event log.

Result. No element is inserted in the logbook. The event log is empty

Step 9&10&11: Reduce the voltage to 216V. Wait 84 secs (7/5*60s) and restore the voltage to 230V. Read the event log.

The event log is empty

Step 12: Reduce the voltage to 211V and check the voltage indicators are still active.

OK on ITRON and LANDIS + GYR

Step 13&14: Wait 36 secs (3/5*60s) and restore the voltage to 230V.Read the event log.

Result for ITRON and LANDIS + GYR meters
The event log is empty

Step 15&16&17: Reduce the voltage to 211V. Wait 84 secs (7/5*60s). Read the event log.

Result for ITRON and LANDIS + GYR meters: One event was logged.

Step 18&19&20: Restore the voltage to 230V. Read event Log and check timestamps of event log.
No event was logged when going back to nominal voltage.

2.2.2.11.1.2 TC 4.4.19.2

Step 1: OK

Step 2: Program to 1 minute (lower value allowed). Refer to Test 1 (4.4.19.1) step 2

Step 3: Set Voltage swells threshold to 107%

Result for ITRON and LANDIS + GYR meters

Step 4: Refer to Test 1 (4.4.19.1) step 5

Step 5: Set voltage to 243V.

Voltage indicators are still active

Step 6&7: Wait 48 secs (4/5*60s). Restore voltage to nominal value. Read event log

Result for ITRON and LANDIS + GYR meters:

The event log is empty

Step 8&9&10: Set voltage to 243V. Wait 96 secs (8/5*60s). Restore voltage to nominal values. Read Event log.

Result for ITRON and LANDIS + GYR meters:

The event log is empty

Step 11&12&13: Set voltage to 249V. Wait 48 secs (4/5*60s). Restore voltage to nominal value. Read event log

Result for ITRON and LANDIS + GYR meters:

The event log is empty

Step 14&15&16: Set the voltage to 249V. Wait 96 secs (8/5*60s). Read the event log.

Result for ITRON and LANDIS + GYR meters: OK, one event was logged.

Step 17&18&19: Restore to nominal voltage. Check event log.

No event was logged when going back to nominal voltage.

2.2.2.11.2 Tests with DC from LANDIS + GYR

2.2.2.11.2.1 TC 4.4.19.1

Step 2 and 4: setting and checking the thresholds:
Time threshold for voltage sags and swells is set to 1 minute.
Voltage threshold for voltage sags is set to 93% of nominal voltage.

Steps6 to 14: We generate voltage variations with the following conditions:

- 216V during 50 seconds
- 216V during 90 seconds
- 209V during 50 seconds

Reading of the Abnormal voltage elements:
The events generated do not overPassed the thresholds set previously, so no events have been recorded.

Steps 15 to 20: We generate voltage variations with the following conditions:
- 209V during 90 seconds

Reading of the Abnormal voltage elements:
The event generated overPassed the thresholds set previously, so one event have been recorded. The voltage measured and the timestamp are correct.

2.2.2.11.2.2 TC 4.4.19.2

Step 3 : setting and checking the thresholds:
Time threshold for voltage sags and swells is set to 1 minute.
Voltage threshold for voltage swells is set to 107% of nominal voltage.

Steps 5 to 13: We generate voltage variations with the following conditions:
- 243V during 50 seconds
- 243V during 90 seconds
- 251V during 50 seconds

Reading of the Abnormal voltage elements:
The events generated do not overPassed the thresholds set previously, so no events have been recorded.

Steps 14 to 19: We generate voltage variations with the following conditions:
- 251V during 90 seconds

Reading of the Abnormal voltage elements:
The event generated overPassed the thresholds set previously, so one event have been recorded. The voltage measured (251V) and the timestamp are correct.

2.2.3 Results of DLMS Interoperability Lab tests

In Annex C, the overview of results for the DLMS Interoperability is presented. These results are detailed in the following sections.

2.2.3.1 Read / Write Tests

2.2.3.1.1 Tests with DC from ITRON

2.2.3.1.1.1 TC 6.4.5.1

Discover and register of 2 meters (1 ITRON and 1 LANDIS + GYR)
LANDIS + GYR : @MAC => 12
ITRON : @MAC => 11

Step 1: Connection establishment (LANDIS + GYR and ITRON)
Result is successful in both meters

Step 2: Association establishment
Result is successful in both meters

Step 3: get request:
Result is successful in both meters

Step 4: get request with data block:
Result is successful in both meters.

Step 5: set request
Result is successful in both meters.

Step 6: set request with data block

Result is successful or ITRON and LANDIS + GYR meters

Step 7: association release
Result is successful in both meters

2.2.3.1.2 Tests with DC from LANDIS + GYR

2.2.3.1.2.1 TC 6.4.5.1

Step 1: Connection establishment
Result is successful in both meters

Step 2: association establishment
Result is successful in both meters

Step 3: get request
Result is successful in both meters

Step 4: get request with data block
Result is successful in both meters

Step 5: set request
Result is successful in both meters

Step 6: set request with data block
Result is successful in both meters

Step 7: association release
Result is successful in both meters

2.2.3.2 DLMS Firmware update

2.2.3.2.1 Tests with DC from ITRON

2.2.3.2.1.1 TC 6.4.5.2

<u>On ITRON Meter</u>

Firmware Version before FW Upgrade (DSP actually)

Attribute description: LogicalName=1;0;0;2;0;255 ClassId=1 AttributeId=2 ShortName=15128

Name	Value	Description
Array of 3 elements	3	From 1 to 5 elements
Structure		Index 1
firmware_id	80;95;49;77;69;84;	This data identifies the firmware.
firmware_type	LEGALY_RELEVANT (0)	This data indicates the type of firmware: Legaly R...
major_version	1	Type: UNSIGNED8
minor_version	5	Type: UNSIGNED8
checksum	1236916005	Type: UNSIGNED32
Structure		Index 2
firmware_id	86;95;49;65;80;80;	This data identifies the firmware.
firmware_type	NOT_LEGALY_RELEVANT (1)	This data indicates the type of firmware: Legaly R...
major_version	2	Type: UNSIGNED8
minor_version	15	Type: UNSIGNED8
checksum	3091295800	Type: UNSIGNED32
Structure		Index 3
firmware_id	86;95;49;77;79;68;	This data identifies the firmware.
firmware_type	NOT_LEGALY_RELEVANT (1)	This data indicates the type of firmware: Legaly R...
major_version	6	Type: UNSIGNED8
minor_version	19	Type: UNSIGNED8
checksum	678325516	Type: UNSIGNED32

Firmware Version after FW Upgrade (DSP actually)

Attribute description: LogicalName=1;0;0;2;0;255 ClassId=1 AttributeId=2 ShortName=15128

Name	Value	Description
Array of 3 elements	3	From 1 to 5 elements
Structure		Index 1
firmware_id	80;95;49;77;69;84;	This data identifies the firmware.
firmware_type	LEGALY_RELEVANT (0)	This data indicates the type of firmware: Legaly R...
major_version	1	Type: UNSIGNED8
minor_version	5	Type: UNSIGNED8
checksum	1236916005	Type: UNSIGNED32
Structure		Index 2
firmware_id	86;95;49;65;80;80;	This data identifies the firmware.
firmware_type	NOT_LEGALY_RELEVANT (1)	This data indicates the type of firmware: Legaly R...
major_version	2	Type: UNSIGNED8
minor_version	15	Type: UNSIGNED8
checksum	3091295800	Type: UNSIGNED32
Structure		Index 3
firmware_id	86;95;49;77;79;68;	This data identifies the firmware.
firmware_type	NOT_LEGALY_RELEVANT (1)	This data indicates the type of firmware: Legaly R...
major_version	6	Type: UNSIGNED8
minor_version	147	Type: UNSIGNED8
checksum	678327692	Type: UNSIGNED32

2.2.3.2.2 Tests with DC from LANDIS + GYR

The tools available for Open Meter tests do not allow to perform tests on Firmware download. This test could not be executed.

2.3 Report on DLMS - SFSK Field tests

2.3.1 Test setup for field tests

As indicated in [2], the field test was performed on the MENOFIS platform at EDF site. EDF has installed several meters all around its building, with different equipments plugged behind them (computers, printer, air conditioner, etc…) as shown on the pictures below. With 1 DC installed somewhere in the building, we were able to communicate with some of these meters for field tests.

Figure 2-14: Impression on field test facility at EDF, Clamart

2.3.2 Results of the Interoperability field tests

In Annex C, the overview of results for the DLMS Interoperability Field tests are presented. These results are detailed in the following sections.

Because of the nature of the field tests, the data of the meter can be random at the start of the test, and not necessarily on the default values.

It also occurs that the spy can lose some of the frames (the spy being single phase apparatus and the DC 3-phase apparatus, if the meter is not on the same phase than the spy, the DC can understand the meter and not the spy). It is mainly visible in the read by block, where the spy loose some blocks whereas the DC goes on because it understood the answer.

MAC Addresses of the meters:

For tests performed with DC from ITRON, the MAC addresses of the meters used are

- 041067001102 (@MAC is 005) ➔ Meter from LANDIS + GYR
- 041067028144 (@MAC is 006) ➔ Meter from LANDIS + GYR
- 041067028150 (@MAC is 004) ➔ Meter from LANDIS + GYR
- 041067028139 (@MAC is 007) ➔ Meter from LANDIS + GYR
- 031067005434 (@MAC is 003) ➔ Meter from ITRON

Each command (Set/Get…), except for FW upgrade (only on ITRON), will be executed for all meters registered.

For tests performed with DC from LANDIS + GYR, the MAC addresses of the meters used are

- 041067028150 (@MAC is 003) ➔ Meter from LANDIS + GYR
- 031067005434 (@MAC is 005) ➔ Meter from ITRON
- Others meters have been discovered, but they won't be used during these tests.

2.3.2.1 TC 7.4.1.1 Instantaneous values

2.3.2.1.1 Tests with DC from ITRON

5 meters have been discovered by the concentrator. Here's the result of the _K_GetNetwork command sent via TestAPI.

Step 1&2: Instantaneous Energy values

Results for each meter of:

- Date/time read
- RMS Current Phase 1
- RMS Voltage Phase 1
- Total Reactive Q1 Energy
- Total Reactive Q2 Energy
- Total Reactive Q3 Energy
- Total Reactive Q4 Energy

Results from the meter:
All these data were correctly read from the meter and are consistent with the display and the current load.

2.3.2.1.2 Tests with DC from LANDIS + GYR

Step 1-2: reading of date/time and all Energy values:

Summary of values read on the meters:

Data	LANDIS + GYR Meter	ITRON Meter
Date & time	03/03/2011 11h58:09	03/03/2011 11h57:57
RMS current Phase 1	0A	0A
RMS current Phase 2	Error because we are on a single phase meter	
RMS current Phase 3	Error because we are on a single phase meter	
RMS voltage Phase 1	228V	228V
RMS voltage Phase 2	Error because we are on a single phase meter	
RMS voltage Phase 3	Error because we are on a single phase meter	
Total reactive Q1 energy	0VAr	0VAr
Total reactive Q2 energy	0VAr	0VAr
Total reactive Q3 energy	0VAr	0VAr
Total reactive Q4 energy	0VAr	3VAr

For the items readable on the screen, the coherence could be made.

2.3.2.2 TC 7.4.1.2 Read & Modify clock

2.3.2.2.1 Tests with DC from ITRON

Step 1: Read the clock.
Result: All meters have the current date/time

Step 2: Set the clock
Set the clock to 01/01/2030 00:00

Step 3: Read the clock we just set
Results: All meters synchronized clock correctly at step 2.

2.3.2.2.2 Tests with DC from LANDIS + GYR

Step 1: read the clock:

Step 2: set the clock:

Step 3: read the clock:

Summary of values read on the meters:

Data	LANDIS + GYR Meter	ITRON Meter
Old Date & time	17/12/2010 11h02:57	03/03/2011 11h01:15
New Date & time	03/03/2011 11h02:57	03/03/2011 11h03:06

NB: the difference between the new date read on the meters after the setting of parameters comes from the delay between the moment the data is read.

2.3.2.3 TC 7.4.1.3 – Tarification

2.3.2.3.1 Tests with DC from ITRON

Step 1: Read the currently active tariff
Results: the current active tariff has been successfully read.

Step 2: Program a tariff change
We program the tariff 8 to be used.

Step 3: Read the programmed values after tariff activation.
Result per meter

It can be seen from the data read from the meters that:

- The calendar has been activated at the expected date
- The values have been taken into account correctly by all the meters

2.3.2.3.2 Tests with DC from LANDIS + GYR

Step 1: read active provider activity calendar objects:

Step 2: program a Passedive activity calendar and programm an activation date for Passeive activity calendar:

Step 3: read active provider activity calendar objects:

Summary of values read on the meters:

Data	LANDIS + GYR Meter	ITRON Meter
Active Calendar name	42415345 20202020202020202020202000 FFFFFFFF	42415345 20202020202020202020202020 FFFFFFFF
Active Calendar day table	Array[1] |Struct[2] {U8 = 0x0 (0)} |Array[1] |Struct[3] {OStr[4]= 00 00 00 00} {OStr[6]= FF FFFFFFFF FF} {U16 = 0x8001 (32769) Tariff 1 always active	
Passedive Calendar name	42415345 20202020202020202020202020 FFFFFFFF	
Passedive Calendar day table	Array[1] |Struct[2] {U8 = 0x0 (0)} |Array[1] |Struct[3] {OStr[4]= 00 00 00 00} {OStr[6]= FF FFFFFFFF FF} {U16 = 0x8002 (32770) Tariff 2 always active	

2.3.2.4 TC 7.4.1.4 – Power threshold

2.3.2.4.1 Tests with DC from ITRON

Step 1: Read the reference power active
Results: We can see the value is the same for both meters (9000VA).
Display of the meter is consistent with remotely collected values.

Step 2: Change the threshold
Results: We change the threshold to 8000VA (0x1F40)

Step 3: Check the threshold has been applied
Results: Threshold has been applied. The meters display the right information.

2.3.2.4.2 Tests with DC from LANDIS + GYR

Summary of values read on the meters after the power threshold has been changed:

Data	LANDIS + GYR Meter	ITRON Meter
Old power threshold read on web interface	3000VA	9000VA
Old power threshold read on screen	3kVA	9kVA
New power threshold read on screen	3500VA	8500VA
New power threshold read on screen	3kVA	8kVA

2.3.2.5 TC 7.4.1.5 - Breaker

Because of the nature of the field here present at EDF site, it was not possible to play on the breaker state for field tests. One can easily understand that it is not possible to power down the printer or the computers this way.

2.3.2.6 TC 7.4.1.6 – Read and Check load profile

2.3.2.6.1 Tests with DC from ITRON

Step 1: Check the capture period of the meters
Results: All meters have the same capture period (see highlighted text). 0x708 = 1800s (30 minutes).

Step 2: Read load profiles of the meters
Considering the amount of data read, it is hard to analyze DST points without an advanced system preprocessing the data. Though, the data structure is OK (compact array).

2.3.2.6.2 Tests with DC from LANDIS + GYR

Step 1: Check the capture period of the meters
The load profile period is set to 30 minutes (1800 seconds) for both meters. Because of the time necessary to load a correct amount of data in the buffers, it was not set to 1 hour bet let as is.

Step 2: Read load profiles of the meters
The coherence of data with the specifications is correct (Compact array with elements of kind [structure of 1 element, U32]).
The coherence of data with the load since several month can't be proved.

2.3.2.7 TC 7.4.1.7 – Monthly billing values

2.3.2.7.1 Tests with DC from ITRON

Step 1: Read of monthly EOB
Result per meter: 041067028150

Step 2: Due to the nature of the tests (field tests), we don't know the load applied to meters. Besides, each meter is installed for a different and undefined time.

Step 3: Compare read values with displayed ones

Collected values are the same as displayed values.
Data structure is OK.

2.3.2.7.2 Tests with DC from LANDIS + GYR

Step 1: Reading the monthly EOB elements:
The coherence of data with the specifications is correct (array with elements of kind [structure of 16 elements, OStr(12), 15 * U32]).
The coherence of data with the load since several month can't be proved.

2.3.2.8 TC 7.4.1.8 – Check correct schedule of Daily billing

2.3.2.8.1 Tests with DC from ITRON

Step 1 : Read of Daily EOB

Results:The coherence of data with the load since several month can't be proved.

Step 2: Structure of data is correct
Confirmed

2.3.2.8.2 Tests with DC from LANDIS + GYR

Step 1: Reading the daily EOB elements:
The coherence of data with the specifications is correct (array with elements of kind [structure of 16 elements, OStr(12), 15 * U32]).
The coherence of data with the load since several month can't be proved.

2.3.2.9 TC 7.4.1.9 – Tables of events

2.3.2.9.1 Tests with DC from ITRON

The only generated event supposed to provide an entry in a logbook is the date/time change as the breaker test has not been performed (check 5.5.5 for details). Therefore, we read only the Main Logbook.

2.3.2.9.2 Tests with DC from LANDIS + GYR

Step 1a : Reading the communication logbook

Step 1b : Reading the cover logbook

Step 1c : Reading the main logbook

Step 1d : Reading the voltage cut logbook

The coherence of data with the specifications is correct (array with elements of kind [structure of 2 elements, OStr(12), Enum]).
The coherence of data with the load since several month can't be proved.

2.3.2.10 TC 7.4.1.10 – Firmware update

2.3.2.10.1 Tests with DC from ITRON

<u>Only on ITRON meter</u>

Step 1: Change firmware version
Before changing the version, the version before the download has been read.
After that, a new version is successfully downloaded.

Step 2: Go back to previous firmware version
Result: Same as "version before the download" in step 1.

2.3.2.10.2 Tests with DC from LANDIS + GYR

The tools available for Open Meter tests do not allow to perform tests on Firmware download. This test could not be executed..

2.4 Conclusions and Recommendations

All the tests performed in the laboratory and on the field with the MENOFIS platform showed good results for fulfillments of requirements. Executed interoperability tests have been succesfull.

During the tests, some observations led to a few comments that can be found below, together with recommendations to clarify these points.

2.4.1 Breaker output state / control state

It has been noticed that in case of breaker opening on overpower or power cut schedule, the breaker control state value was not the same for all manufacturers. The breaker output state, indicating the physical real state of the breaker, was correct for all manufacturers.

The SFSK specification does not indicate yet when the breaker control state must be changed. It seems that it should be stated that the breaker control state has to be aligned with the output state. This observation has been taken into account into the standardization program for the SFSK ('LINKY').

2.4.2 Activation of Passedive activity calendar

A different behavior was observed between the meters of the 2 manufacturers, but this issue does not come from incomplete or unclear specification. A tolerance on this point was granted by the client.

2.4.3 Manual reconnection after Overload power

After a disconnection on overload power, the behavior of both meters is not the same when trying to reconnect manually. It appears that the behavior depends on the technology chosen for the breaker.

Although the conditions for opening the breaker are clearly stated in the specifications, some particular case can be seen when trying to open / close the breaker quickly (as seen in this test case). This behavior is therefore considered as an implementation issue, so no further action is taken.

3 REPORT ON TEST RESULTS FOR DLMS - PRIME

3.1 Introduction

3.1.1 DLMS-PRIME Laboratory tests

In this section, the results for the tests done at KEMA Energy, Madrid are described. In the following table, an overview is given of the performed test cases. The test cases were performed on prototypes described in [2].
The test cases are described in the [1].

Test cases:	Described in sections of [1]	Prototypes:	Described in section of [2]
PRIME Interoperability tests	6.4.3	CURRENT DC	4.3.1.2
		ELSTER METER	4.3.2.1
		ITRON METER	4.3.3.2
		LANDIS + GYR METER	4.3.4.1
		ZIV METER	4.3.5.1
		ZIV DC	4.3.5.2
DLMS Interoperability tests	6.4.5	CURRENT DC	4.3.1.2
		ELSTER METER	4.3.2.1
		ITRON METER	4.3.3.2
		LANDIS + GYR METER	4.3.4.1
		ZIV METER	4.3.5.1
		ZIV DC	4.3.5.2
DLMS Functional Testing	4.4.19.1; 4.4.19.2	ELSTER METER	4.3.2.1

Table 3-1: Overview of OPEN Meter testing at KEMA Energy, Madrid

In the next sections, a summary is given of the used test setup.

In section 3.2.3, the test results are presented for the PRIME Interoperability tests.
In section 3.2.4, the test results are presented for the DLMS Interoperability tests.
In section 3.2.5, the results for the DLMS Functional tests are discussed.

In 0, recommendations are presented for the test setup, the test cases as well as for the future standards.

The PRIME and DLMS Interoperability tests described in this report were performed in two runs. These test runs will be indicated with **RUN1** and **RUN2**.

RUN 1 was executed on 8 and 9 February 2011,in the laboratory of KEMA Madrid, Spain. The test was done with ZIV and CURRENT Data Concentrators, and LANDIS + GYR and ZIV meters.

The following people were involved:

Test Engineers:
E. Henriquez, KEMA
B. Roelofsen, KEMA
Support Engineers:
L. Marrón, ZIV
A. Brunschweiler, CURRENT

Table 3-2: Engineers involved in KEMA 'run1'

Auditors:	**Date of participation of auditors**
A. Gómez, ENDESA	8 and 9 of February 2011
D. Mardero, ENEL	8 and 9 of February 2011
G. Mauri, RSE	8 and 9 of February 2011
G. Timmins, L+G	8 and 9 of February 2011
C. Rodriguez, ENDESA	9 of February 2011
A. Signorini, ENEL	8 and 9 of February 2011

Table 3-3: Auditors present at run1

All these tests were executed in the presence of the auditors.[1]

RUN2 was done with ZIV and CURRENT Data Concentrators and with ELSTER and ITRON meters. RUN2 was done at 29 and 30 March 2011, in the laboratory of KEMA Madrid, Spain.

The following persons were involved:

Test Engineers:
E. Henriquez, KEMA
B. Roelofsen, KEMA
Support Engineers:
A. Wohlrath, ELSTER
A. Rabine, ITRON (by email / phone)

Table 3-4: Engineers involved in KEMA 'run2'

Auditors:	**Date of participation of auditors**
M. Arzberger, ELSTER	29 and 30 of March 2011
M. de Tellechea, ENDESA	29 and 30 of March 2011
Francisco Jimenez, ENDESA	29 and 30 of March 2011

Table 3-5: Auditors present at run2

All tests were done in the presence of auditors.

[1] See final remark at testcase 6.4.5.2 in run 1

In 'Annex D – Overview of KEMA Test Results' gives a complete overview of the combinations of equipment that were tested during the interoperability tests of both runs.
The figure below shows the location of the tests. User interfaces from the Data Concentrator and loggings from the sniffer were showed 'live' on the flat-screen, so that the witnesses got direct insight in the test details.

Figure 3-1: Testing and auditing at KEMA Energy, Madrid.

3.1.2 DLMS-PRIME Field tests

The tests were executed on 24th of March 2011,in the laboratory of IBERDROLA located in Bilbao, Spain. All tests were done in the presence of the auditors[2].

The majority of the tests were done with ZIV and CURRENT Data Concentrators, and LANDIS + GYR and ZIV meters.

The following persons were involved:

Test Engineer:	**Auditors (24th March)**
Iñigo Larumbe, IBERDROLA	Amador Gómez, ENDESA
	Vito Rizzo, ENEL
Support Engineers:	Jose María Marro Guerrero, ENDESA
L. Marrón, ZIV	Auguste Ankou, ITRON
J. Hernandez, CURRENT	Alicia Latorre, ITRON
	Jorge Cernandas, KEMA
	E. Henriquez, KEMA

[2] Setup 4 (see section 3.3.2.1) has not been witnessed by all auditors

In the following table, an overview is given of the performed test cases.

The test cases are described in the [1].
The prototypes are described in [2].

Test cases:	**Described in sections of [1]**	**Prototypes:**	**Described in section of [2]**
PRIME + DLMS Interoperability field and lab tests	7.4.1.1 to 7.4.1.10	CURRENT DC	4.3.1.2
		LANDIS + GYR METER	4.3.4.1
		ZIV METER	4.3.5.1
		ZIV DC	4.3.5.2
PRIME + DLMS Interoperability lab tests	7.4.1.1 and 7.4.1.2	SAGEMCOM METER	-
		CIRCUTOR METER	-
		SOGECAM METER	-

Table 3-6: Overview of OPEN Meter testing at IBERDROLA

The Interoperability field tests only affects to the interface MI1-CI1 between the concentrators and the electricity meter.

Note: Although this chapter is called 'Interoperability Field Tests', most of the 'Interoperability Field Tests' have been executed in both the laboratory and the field. (see table 3-26).

3.2 Report on laboratory tests for DLMS - PRIME

3.2.1 Test setup for DLMS - PRIME Interoperability tests

The test setup used for the DLMS - PRIME Interoperability tests is described in [2], section 2.2.2. Deviations and addition on this setup:

1. Instead of the Wireshark analyzer, the KEMA DLMS/PRIME Analyzer was used.
2. The KEMA DLMS/PRIME Analyzer was connected to a PRIME Sniffer hardware module provided by CURRENT.

3.2.2 Test setup for DLMS Functional testing

For the DLMS Functional tests mentioned in Table 3-1, the KEMA DLMS Test Facility as mentioned in [2], section 3.2.3 is used. Some tests were done with a simplified version of this tool, which allowed for a manual execution of readings and writings to a DLMS object instead of full automation.

3.2.3 Results of PRIME Interoperability tests

3.2.3.1 Test Results RUN1: ZIV+CURRENT DC, LANDIS + GYR and ZIV Meters.

In the table below, it is indicated which test cases from the sections 6.4.3 in [1] were executed. The test case id's in the table refer to the test cases in [1] "Definitions of test procedures".

TC id:	Executed (Y/N) RUN 1
6.4.3.1	N
6.4.3.2	N
6.4.3.3	Y
6.4.3.4	Y
6.4.3.5	Y
6.4.3.6	Y
6.4.3.7	Y

Table 3-7: Executed PRIME Interoperability test cases during RUN1.

The test cases that were executed were performed in two setups:

a) Setup 1with the Data Concentrator from CURRENT as the Base Node in the PRIME Network.
b) Setup 2 with the Data Concentrator from ZIV as Base Node in the PRIME Network.

A summary of test executed tests mapped on the prototype combinations is presented in ANNEX D.

The test results for both setups are described in detail in the sections below. In some test cases, the results are 'decorated' with some screenshot, to give the reader an impression on how the test details were obtained.

Detailed loggings of all described tests can be requested from the test engineers.

TC 6.4.3.1 – Performance PHY layer: Modulation schemes

During RUN1 it was decided by KEMA and ZIV representatives to skip this test. It is rather a performance tests than an interoperability test. Special interfacing is required on the DUT to perform the test.

TC 6.4.3.2 – Performance PHY layer: EVM level

During RUN1 it was decided by KEMA and ZIV representatives to skip this test. It is rather a performance tests than an interoperability test. Special interfacing is required on the DUT to perform the test.

TC 6.4.3.3 – Registration and Keep – Alive Process.

This test case was performed successfully in the setup with the CURRENT DC as well as in the setup with the ZIV DC. For both the ZIV and LANDIS + GYR prototype, registration messages were captured. The Keep Alive messages could captured in the sniffer as well, and it has been verified that these messages were sent within the time specified in REG.TIME.

In the web interface, the registration of both devices could be confirmed, like in the picture below:

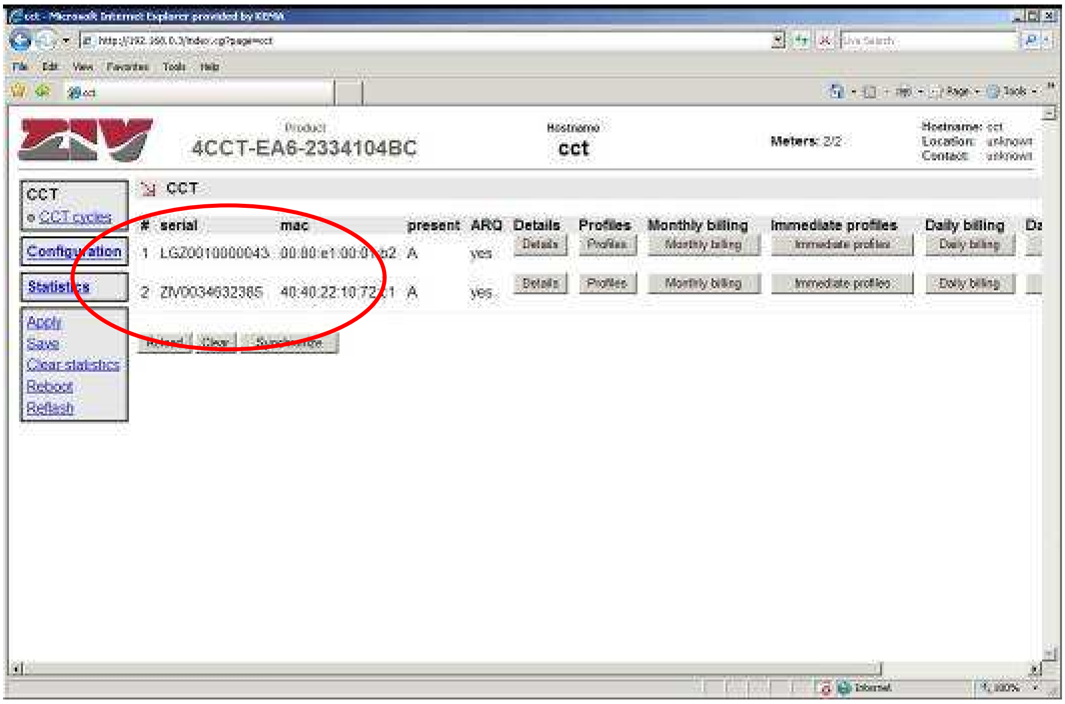


Figure 3-2: Registration of ZIV and LANDIS + GYR prototypes, shown in the ZIV Data Concentrator Web Interface.

In the KEMA PRIME Analyzer, the Registration packages could be analyzed and checked for correctness. Also, the REG.TIME parameter could be read out to check the correct scheduling of the Keep Alive Messages.

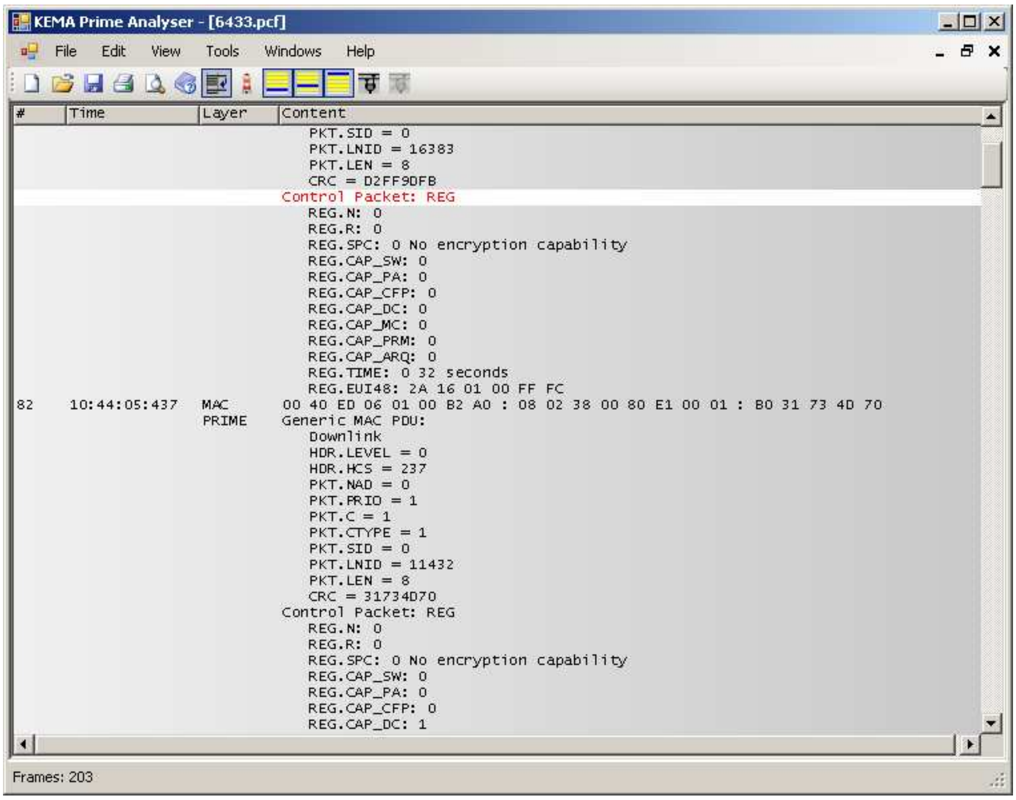


Figure 3-3: KEMA PRIME Analyzer with a captured REG message.

TC 6.4.3.4 – Promotion triggered from Service Node.

By introducing attenuators in the network as described in [2] section 2.2.2., a 3 level network topology was created. In the test book [1], this setup is referred to as 'Setup c' (see [1], par. 6.2).

At level 1, the Base Node is located. At level 2, a Service Node is placed which can receive the beacons from the Base Node. At level 3, another Service Node is placed, which cannot receive the beacons from the Base Node.

In both test setups, the service nodes on level 2 and 3 where the ZIV meter and the LANDIS + GYR meter. The tests were also performed with LANDIS + GYR on level 2 and ZIV on level 3.

In both combinations, the following sequence of messages were captured in the PRIME KEMA Analyzer.

1) BN starts sending beacons.
2) Registration messages from SN prototype on level 2.
3) Promotion Needed PDU's from SN on level 3 (captured with another sniffer module on level 3).
4) Promotion request of prototype on level 2.
5) Registration of prototype on level 3 –via switch on level 2 – at the ZIV Base Node.

In both combinations of prototypes on level 2 and 3, the test ended up with both devices registered at the ZIV Base node. It was observed in the web interface that the connected network was build up of a switch and a terminal, with the mac-adresses of the prototypes.

The figure below is a screenshot of the web interface of the CURRENT Data Concentrator, showing the layered PRIME Network.

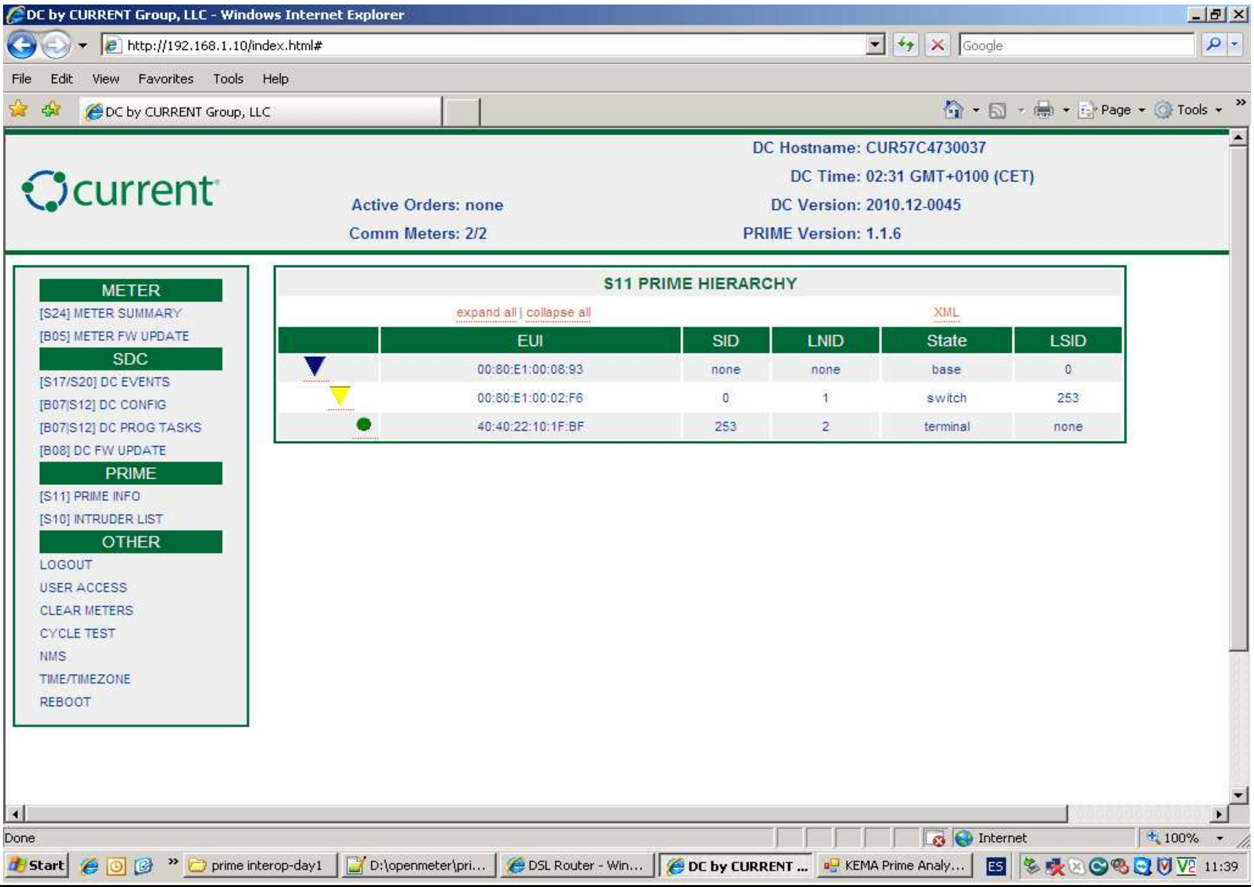


Figure 3-4: CURRENT DC web interface, showing the layered PRIME network.

In both setups, the network was maintained by Keep Alive Messages, that were forwarded by the switch.

TC 6.4.3.5 – Promotion triggered from Base Node.

In this test case, a Service Node in terminal state is triggered by the Base Node to promote to a switch. This functionality of the BN could be used successfully used in the CURRENT Data Concentrator as well as in the ZIV Data Concentrator.

During this test case, the same layered network was created as described in TC 6.4.3.4. After the promotion of the registered prototype at level 2, the prototype of the other manufacturer at level 3 was switched on.

In each setup, the combination with the ZIV prototype on level 2 and the LANDIS + GYR prototype on level 3 was tested, as well as the combination with the LANDIS + GYR prototype on level 2 and the ZIV prototype on level 3.

In both setups and in both combinations, Base Node triggered promotion of the level 2 prototype was observed with the sniffer, as well as registration of the prototype on level 3.

The network was maintained by Keep Alive Messages.

TC 6.4.3.6 – Unicast PRIME Firmware Update.

In the setup with the CURRENT Data Concentrator and the setup with the ZIV Data Concentrator, the PRIME firmware of the ZIV prototype and the LANDIS + GYR prototype was updated via the Base Node in the Data Concentrator.

In the setup with the ZIV Concentrator, the firmware update was done via the PRIME Manager application of which a snapshot is shown below:

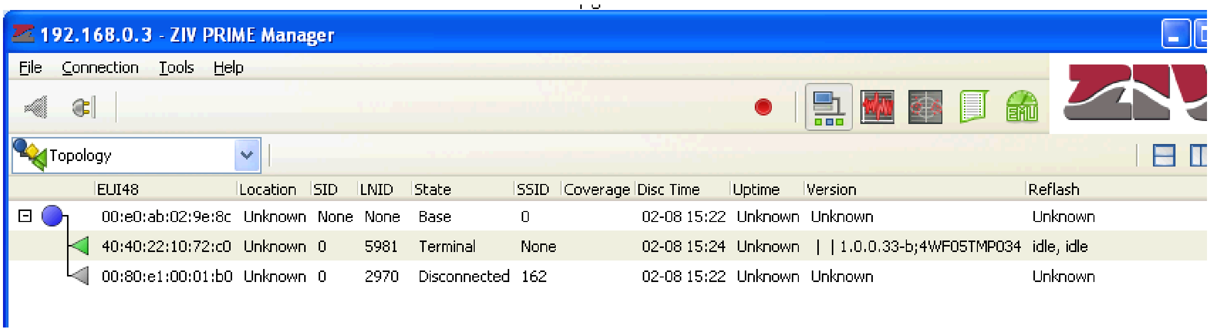


Figure 3-5: Snapshot of the PRIME Manager Application

In the setup with the CURRENT Data Concentrator, the ZIV and LANDIS + GYR updates were done using command line interface.

In the combination CURRENT Data Concentrator with LANDIS + GYR meter, the update did not finish properly for unknown reasons. LANDIS + GYR is provided with the resulting loggings for further analysis.

TC 6.4.3.7 – Multicast PRIME Firmware Update.

The multicast update mechanism was tested at both setups: With the CURRENT Data Concentrator as well as the ZIV Data Concentrator.

With the ZIV Data Concentrator setup, 2 tests were done:
1. The multicast group was formed of 2 ZIV meters. The firmware was updated correctly in the ZIV meters.
2. The multicast group was formed of 2 LANDIS + GYR prototypes and one ZIV prototype. The firmware update was performed correctly in the LANDIS + GYR prototypes. It was checked that one LANDIS + GYR prototype was left out of the multicast group.

At both tests, the firmware was successfully updated in the meters that were part of the multicast group.

With the CURRENT Data Concentrator setup, only the test was done with the 2 ZIV prototypes in the multicast group, and the LANDIS + GYR prototype left outside the group, in order to verify that this meter did not start a firmware update.
At this test, the firmware was successfully updated in the meters that were part of the multicast group.

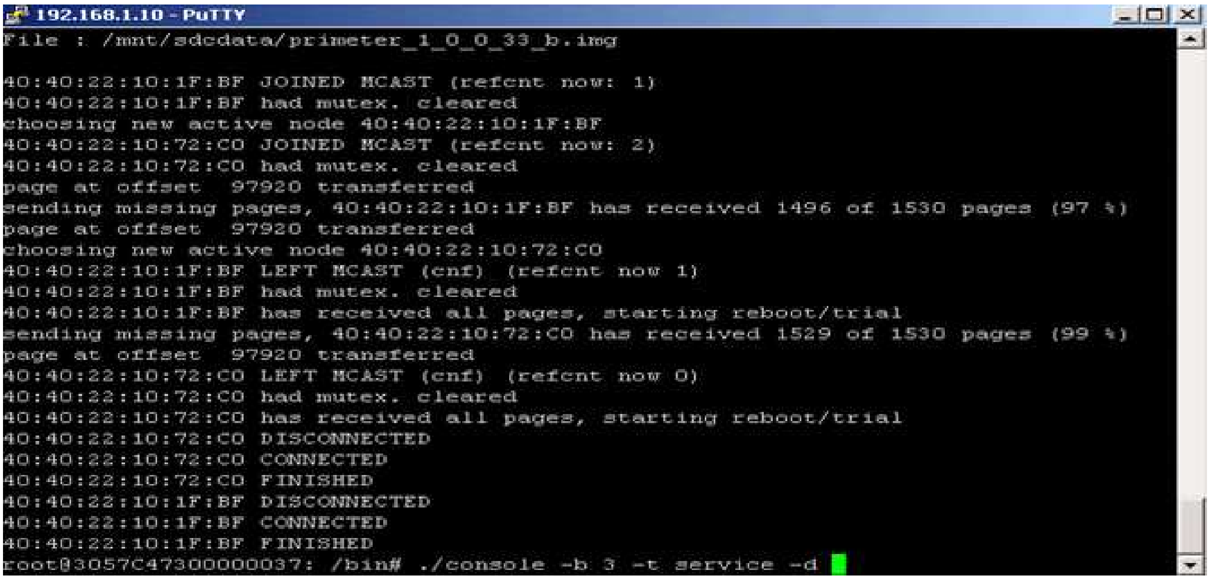


Figure 3-6: Two DISCONNECT - CONNECT - FINISHED sequences at the end of a Multicast FU using the CURRENT Data Concentrator Command Line interface.

3.2.3.2 Test Results RUN2: CURRENT and ZIV DC, ELSTER and ITRON METER

In the table below, it is indicated which test cases from the sections 6.4.3 in [1] were executed. The test case id's in the table refer to the test cases in [1] "Definitions of test procedures".

TC id:	Executed (Y/N) RUN 2
6.4.3.1	N
6.4.3.2	N
6.4.3.3	Y
6.4.3.4	Y
6.4.3.5	Y
6.4.3.6	Y (ELSTER ONLY)
6.4.3.7	N

Table 3-8: Executed PRIME Interoperability test cases during RUN2.

REMARK:
1) Both the ELSTER and the ITRON Meter had not been subject to the official PRIME Conformance test procedure at the date of these OPEN Meter tests.

2) The ELSTER meter did not show up in the Graphical User interface of the CURRENT DC. This problem was investigated during the test but could not be solved during the available test time.

3) It was reported beforehand that the ITRON meter did not yet support firmware updates. The firmware update tests were skipped for the ITRON meter.

TC 6.4.3.1 – Performance PHY layer: Modulation schemes

During RUN1 it was decided by KEMA and ZIV representatives to skip this test. It is rather a performance tests than an interoperability test. Special interfacing is required on the DUT to perform the test. The test was skipped for these reasons at RUN2 as well.

TC 6.4.3.2 – Performance PHY layer: EVM level

During RUN1 it was decided by KEMA and ZIV representatives to skip this test. It is rather a performance tests than an interoperability test. Special interfacing is required on the DUT to perform the test. The test was skipped for these reasons at RUN2 as well.

TC 6.4.3.3 – Registration and Keep Alive Process.

During this test case, correct registration of the service node at the base node is tested. It is also tested that the service node stays registered by a correctly implemented Keep Alive Process. The tests was executed with the CURRENT DC, and the ELSTER and ITRON meters. For both meters, the tests completed successfully.

Figure 3-7 shows registration request from the ELSTER meter, and the response from the CURRENT Data Concentrator.

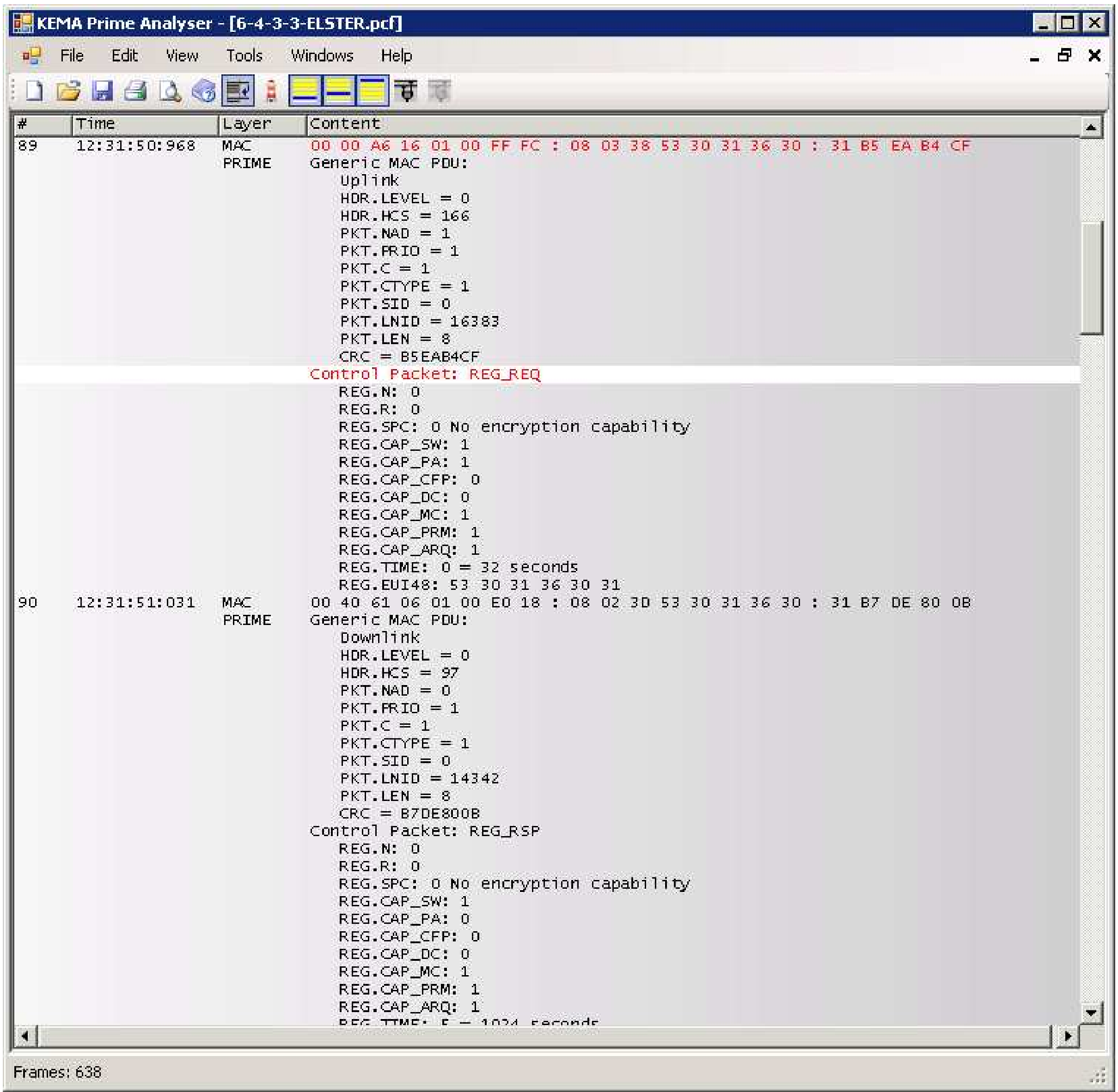


Figure 3-7: Registration Request initiated by ELSTER Meter with EUI 53:30:31:36:30:31 and response from the CURRENT DC.

The test was repeated with the ZIV Data Concentrator. The next figure shows a screenshot of the ZIV PRIME Manager application connected to the ZIV DC, with the ELSTER meter registered to it. It was observed that this registration was stable and that the KEEP ALIVE Messages were exchanged in time.

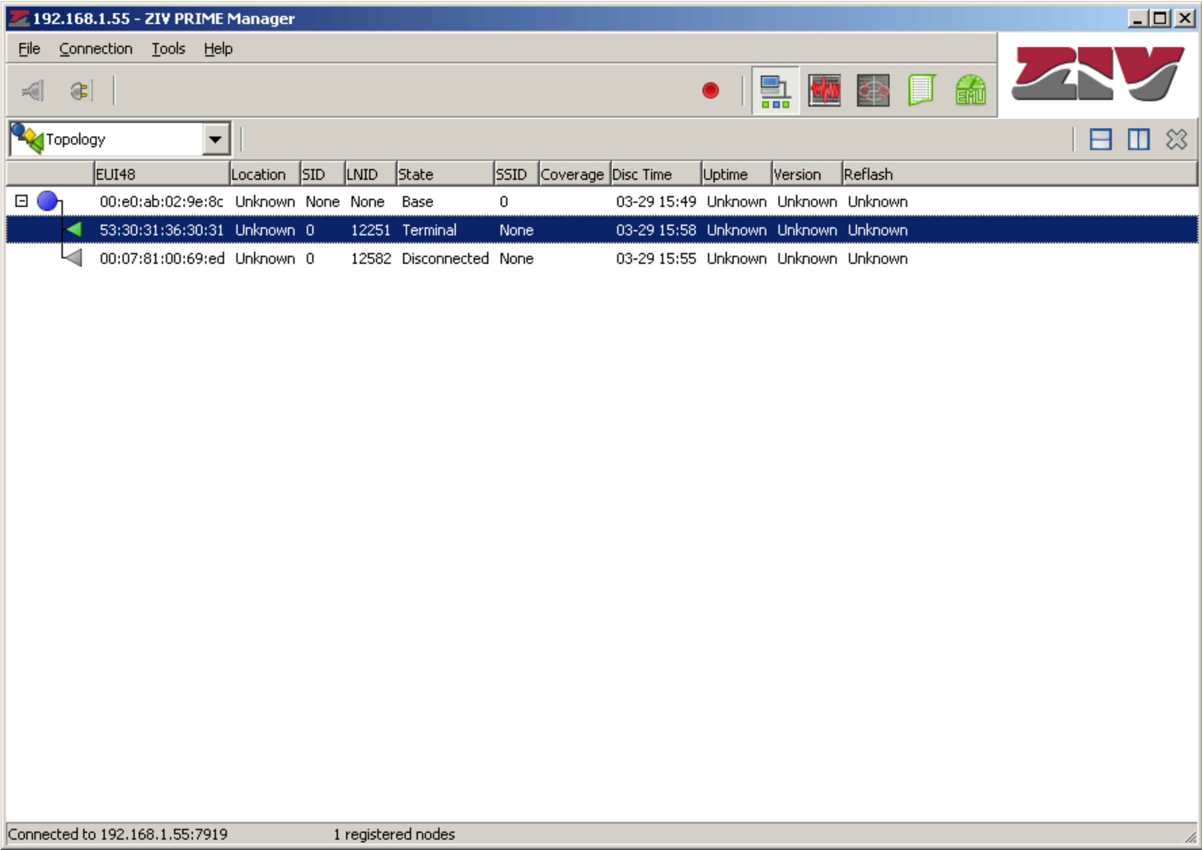


Figure 3-8: ELSTER Meter with EUI 53:30:31:36:30:31 registered to ZIV DC.

With the ITRON meter, a successful registration could be made at the CURRENT DC, as is shown in Figure 3-9. It was observed that this registration was stable and that KEEP ALIVE messages were exchanged in time.

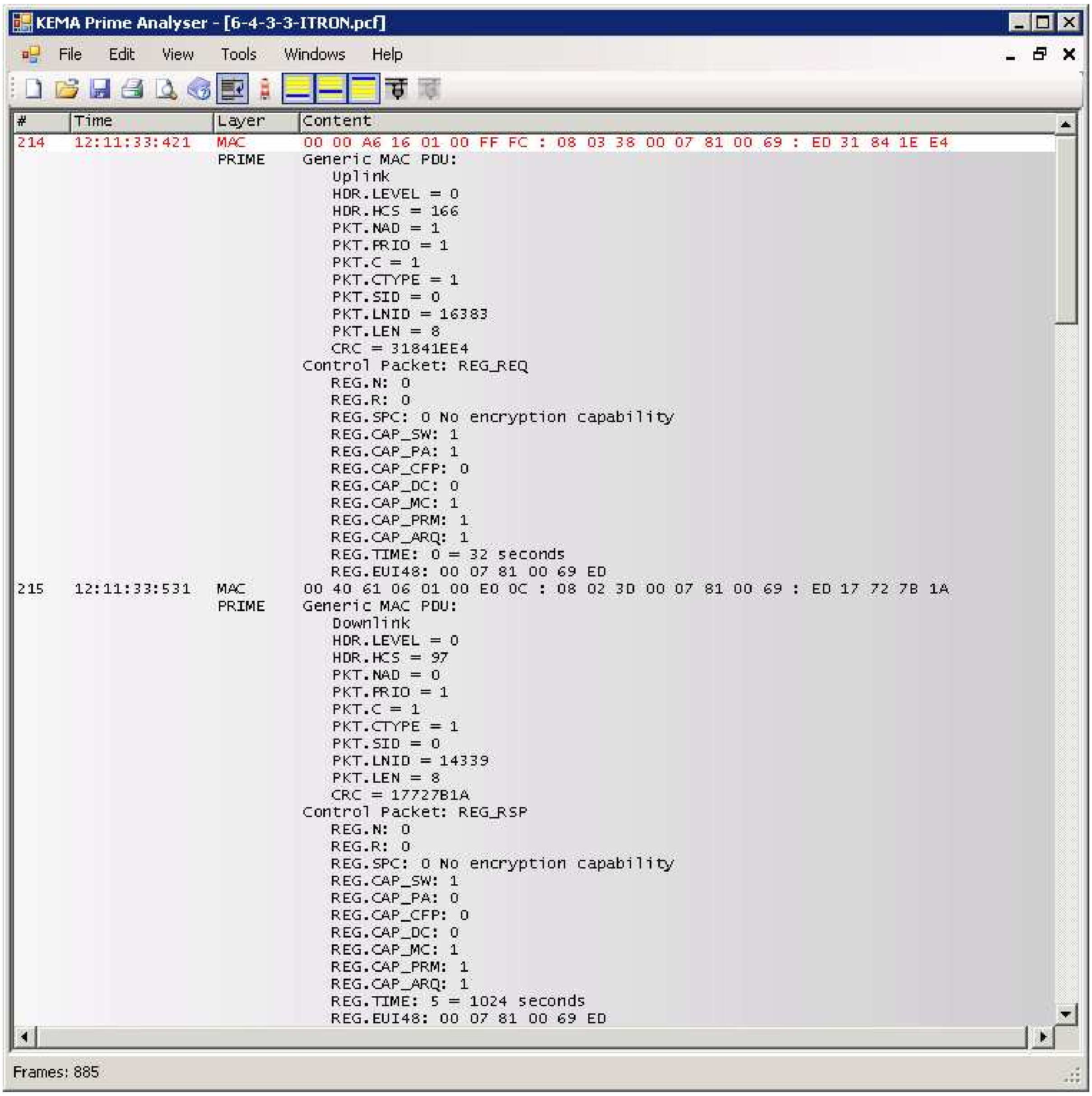


Figure 3-9: ITRON PRIME meter with EUI48 00:07:81:00:69:ED registered at the CURRENT DC.

The combination ITRON Meter registered at the ZIV DC concentrator was also tested. In this combination, the registration procedure was successful and maintained by correct exchange of KEEP ALIVE Messages.

TC 6.4.3.4 – Promotion triggered from Service Node.

By introducing attenuators in the network as described in [2] section 2.2.2., a 3 level network topology was created. In the test book [1], this setup is referred to as 'Setup c' (see [1], par. 6.2).

At level 1, the Base Node is located. At level 2, a Service Node is placed which can receive the beacons from the Base Node. At level 3, another Service Node is placed, which cannot receive the beacons from the Base Node.

This test was done with CURRENT DC and

a) The ELSTER meter as Switch and the ITRON meter as Terminal.

b) The ITRON meter as Switch and the ELSTER Meter as Terminal.

The sniffer was placed at the base node level of the network.

Figure 3-10 shows the Registration Response towards the ELSTER Meter to which the LNID 14342 is assigned.

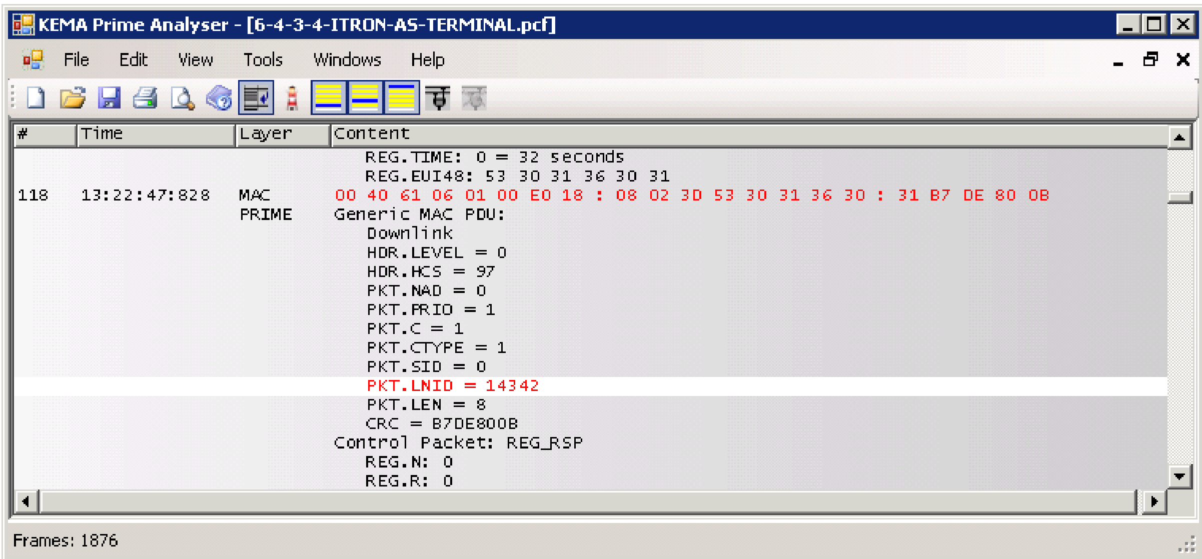


Figure 3-10: REQ_RSP in which the ELSTER meter gets assigned a LNID of 14342. Just above the red line with the time stamp, the EUI48 of the ELSTER meter is shown (as the final line of the REG_RSP message).

Figure 3-11 shows the promotion request of the same meter at a later time stamp during the logging, triggered by PNPDU's (not visible at this location in the PRIME Network).

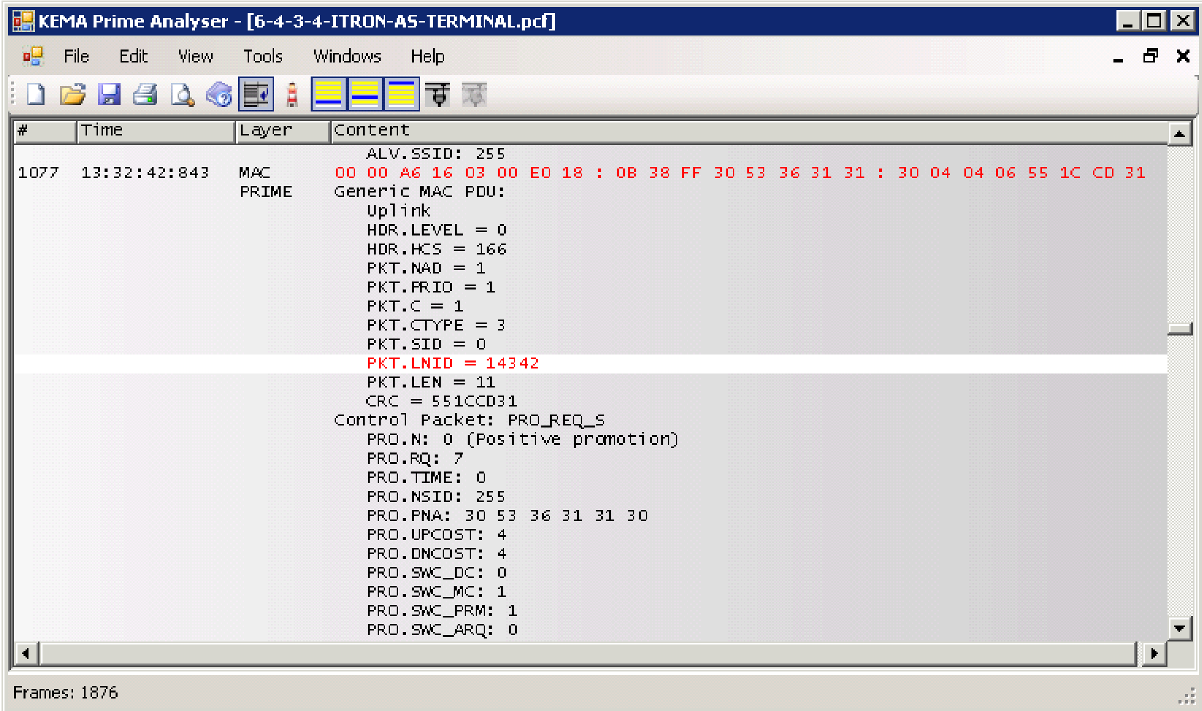


Figure 3-11: Uplink Promotion Request initiated by the ELSTER meter with assigned LNID of 14342.

It was observed that the configuration BN-CURRENT, SWITCH-ELSTER, ITRON-TERMINAL was a stable PRIME Network, with KEEP ALIVE messages exchanged within the configured time limits.

With the same Data Concentrator (from CURRENT), the configuration with ITRON as switch node and ELSTER as terminal node was tested as well, which resulted in a stable PRIME Network and appropriate KEEP ALIVE messages.

Figure shows the assignment of LNID to the ITRON Meter in the REG_RSP.

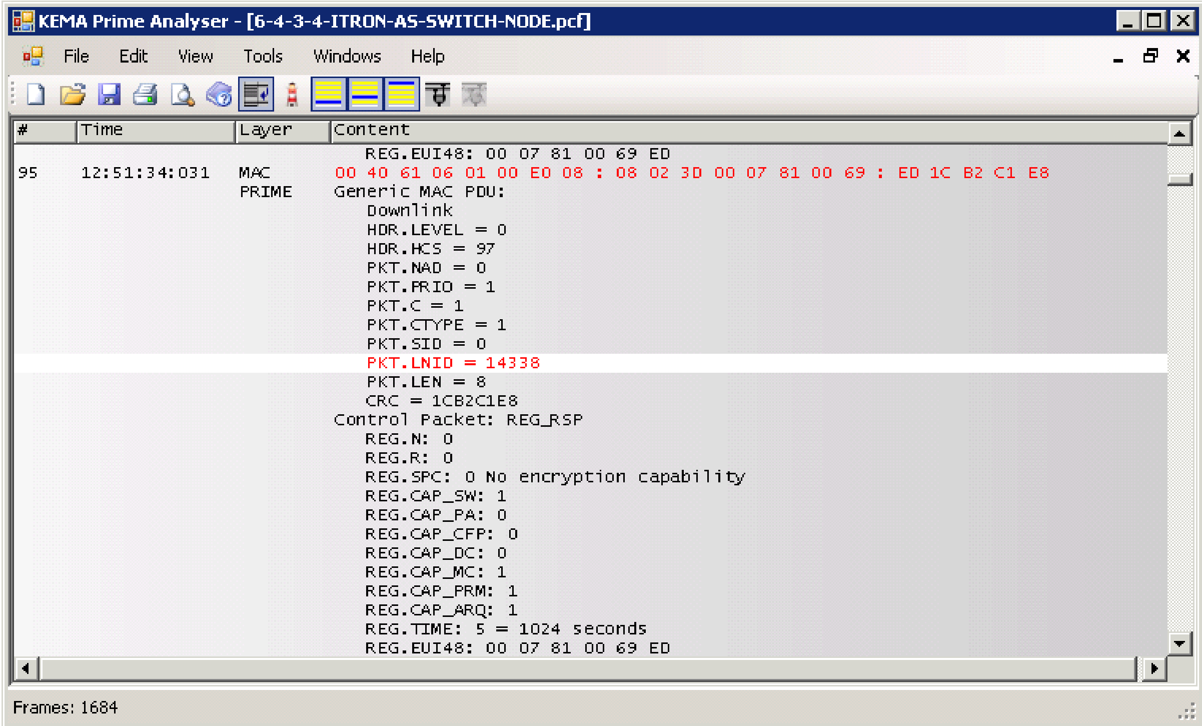


Figure 3-12: LNID 14338 assigned to ITRON Meter with EUI48 of 00:07:81:00:69:ED as shown in the first line of the logging.

The ITRON meter sends a promotion request to the CURRENT DC, as shown in figure

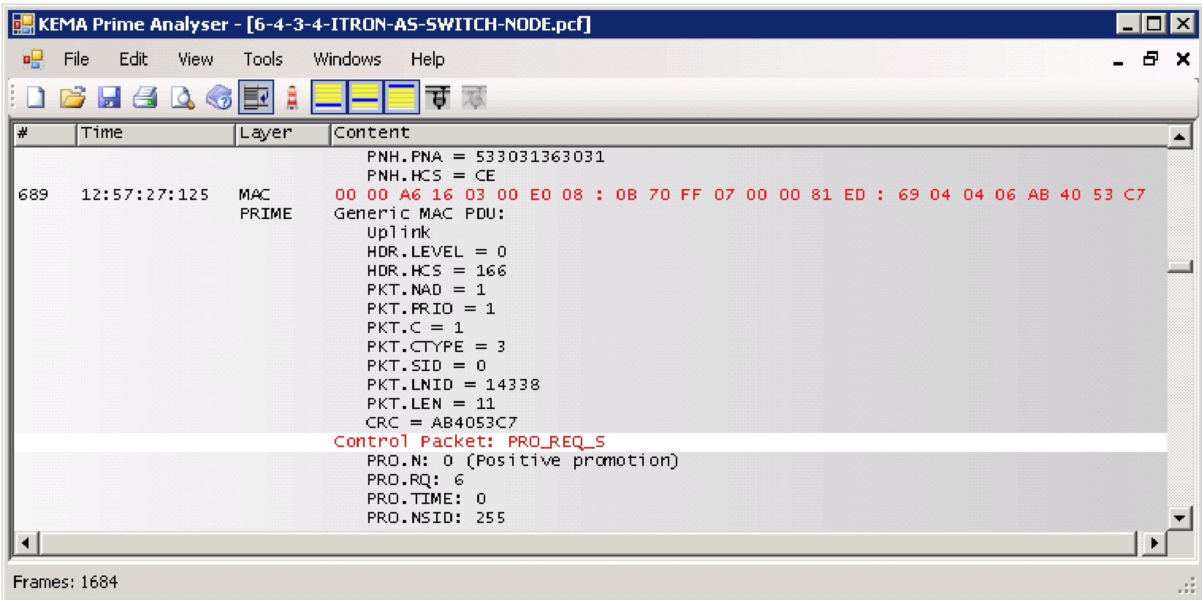


Figure 3-13: Promotion Request initiated from the ITRON meter with the assigned LNID of 14338.

It was observed that also this network, with CURRENT DC as Base Node, ITRON Meter as Switch, and ELSTER meter as Terminal node was stable and maintained by correct exchange of KEEP ALIVE messages.

TC 6.4.3.5

In the contrary to the previous test case, promotion of a service node to a switch is not triggered by the reception of PNPDU's, but by a promotion request sent by the Data Concentrator.

This test case was done with the ZIV Data Concentrator. The results are presented by the following two figures, which shows the ELSTER meter as switch and the ITRON meter as terminal, (Figure 3-14) as well as the ITRON meter as switch and the ELSTER meter as terminal (Figure 3-15). Note the EUI-48's of the switches and the terminal in the 2 pictures, which are exchanged.

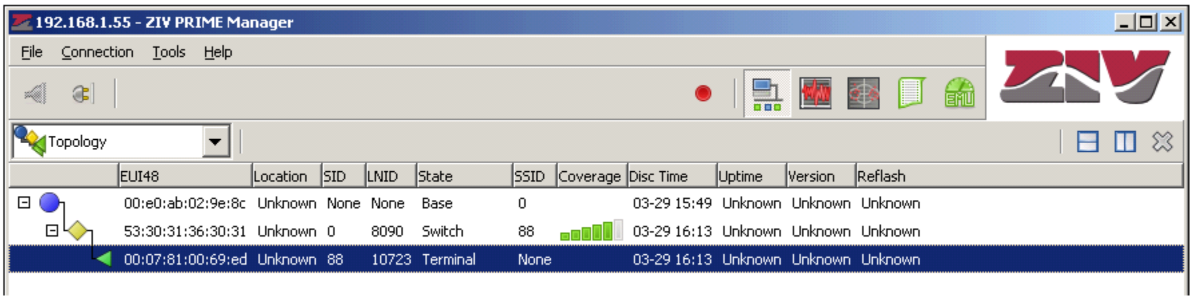


Figure 3-14: Graphical presentation in the ZIV PRIME Manager of a PRIME Network of ZIV Base Node, the ELSTER Switch node and the ITRON terminal node.

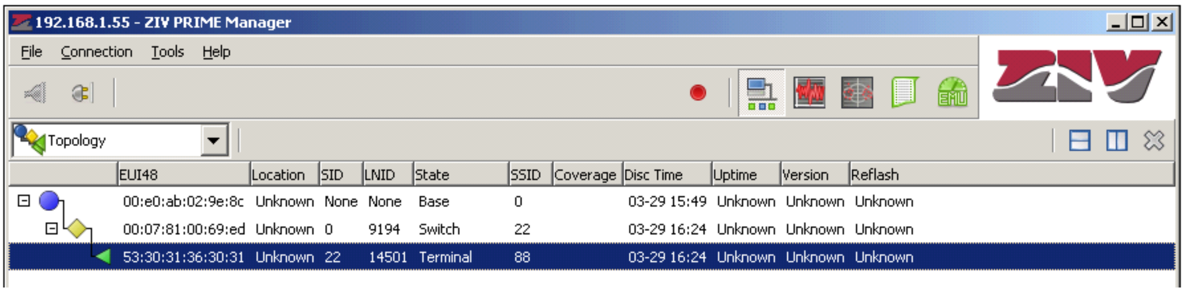


Figure 3-15: PRIME Network of ZIV Base Node, which triggered the ITRON meter to become a switch. The ELSTER meter registered via the ITRON switch node.

In both combinations, a stable network was maintained by KEEP ALIVE messages.

TC 6.4.3.6

The version of the ITRON meter that was available during the test did not support PRIME firmware update.

The ELSTER meter did not show up in the Graphical User interface of the CURRENT DC. A firmware update could not be triggered from the CURRENT DC.

For the ELSTER meter, the firmware update was triggered from the ZIV DC via the PRIME Manager Application. However, after one hour, the process was not completed and the test was stopped. Two attempts were done with the block sizes 64k and 192k. The loggings were sent for investigation to ELSTER and ZIV. Note that the PRIME firmware update procedure is part of the PRIME Conformance Test. See also remark 1 in section 3.2.3.2. It makes sense to repeat the test after the ELSTER meter has proven to be PRIME conformant.

TC 6.4.3.7

Because only one ELSTER meter was available during the tests, the multicast firmware update test could not be performed.

The ITRON meter did not support firmware update.

3.2.4 Results of DLMS Interoperability tests

3.2.4.1 RUN1: ZIV and LANDIS + GYR Data Concentrator, LANDIS + GYR and ZIV Meter

In the table below, it is indicated which test cases from the sections 6.4.3 in [1] were executed during this test run. The test case id's in the table refer to the test cases in [1] "Definitions of test procedures".

TC:	Executed (Y/N)
6.4.5.1	Y
6.4.5.2	Y

Table 3-9: Executed DLMS Interoperability test cases

All prototypes under test implemented the Spanish DLMS Profile.
The test cases that were executed were performed in two setups:

a) Setup with the data concentrator from CURRENT as the data concentrator in the network.
b) Setup with the data concentrator from ZIV as the data concentrator in the network.

A summary of test executed tests mapped on the prototype combinations is presented in ANNEX D.

The test results for both setups are described in detail in the sections below. In some test cases, the results are 'decorated' with some screenshot, to give the reader an impression on how the test details were obtained.

TC 6.4.5.1 – Read / Write Tests

During this test, several read / write mechanisms were tested in both setups, as summarized in the table below:

	ZIV DC with LANDIS + GYR and ZIV meter prototypes	**CURRENT DC with LANDIS + GYR and ZIV meter prototypes**
DLMS Action	**OBIS Code, result**	**OBIS Code, result**
Association Request	NA, success.	NA, success.
GetRequest	{70, 0-0:96.3.10.255,3} Disconnector control, control state. (amongst many others), success.	{ 8,0-0:1.0.0.255,2} Clock - time (amongst many others), success.
GetRequestWith DataBlock	{ 7,0-0:98.1.1.255,2} Monthly billing values.	{ 7,0-0:98.1.1.255,2} Monthly billing values.
SetRequest	{ 8,0-0:1.0.0.255,2} Clock - time	{ 8,0-0:1.0.0.255,2} Clock - time
Association Release	NA, success.	NA, success.

Table 3-10: Details on DLMS Read / Write tests in both setups.

In both setups, the data exchange was triggered from the web interface of the Data Concentrator. For example, the GUI of the ZIV DC is given below, with the buttons to obtain certain DLMS objects.

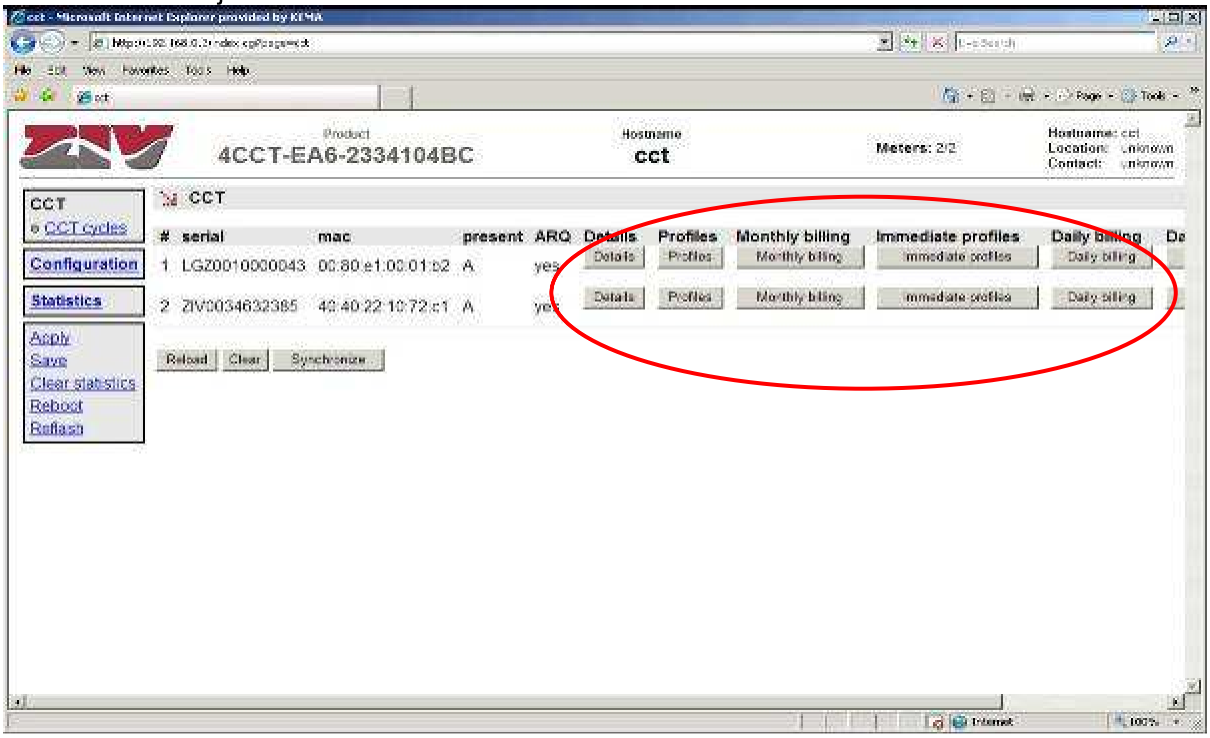


Figure 3-16: GUI of ZIV DC to retrieve DLMS objects from the prototypes.

Inside the data concentrator, the data is stored as XML, a format suitable for exchanging the data with e.g. the Central Access System.

The figure below gives a snapshot of data formatted as XML, which was retrieved from the prototypes by the CURRENT Data Concentrator.


```xml
Report IdRpt="S04" IdPet="0" Version="3.0">
 <Cnc Id="CUR57C4730037">
  <Cnt Id="ZIV0034611135">
   <S04 Fhi="20110131122300000W" Fhf="20110201114000000W" Ctr="1" Pt="0" Mx="0 W" Fx="20110131122300000W">
    <Value AIi="20 Wh" AEi="0 Wh" R1i="0 varh" R2i="0 varh" R3i="0 varh" R4i="0 varh"/>
    <Value AIa="20 Wh" AEa="0 Wh" R1a="0 varh" R2a="0 varh" R3a="0 varh" R4a="0 varh"/>
   </S04>
   <S04 Fhi="20110131122300000W" Fhf="20110201114000000W" Ctr="1" Pt="1" Mx="0 W" Fx="20110131122300000W">
    <Value AIi="19 Wh" AEi="0 Wh" R1i="0 varh" R2i="0 varh" R3i="0 varh" R4i="0 varh"/>
    <Value AIa="19 Wh" AEa="0 Wh" R1a="0 varh" R2a="0 varh" R3a="0 varh" R4a="0 varh"/>
   </S04>
   <S04 Fhi="20110131122300000W" Fhf="20110201114000000W" Ctr="1" Pt="2" Mx="0 W" Fx="20110131122300000W">
    <Value AIi="1 Wh" AEi="0 Wh" R1i="0 varh" R2i="0 varh" R3i="0 varh" R4i="0 varh"/>
    <Value AIa="1 Wh" AEa="0 Wh" R1a="0 varh" R2a="0 varh" R3a="0 varh" R4a="0 varh"/>
   </S04>
   <S04 Fhi="20110131122300000W" Fhf="20110201114000000W" Ctr="1" Pt="3" Mx="0 W" Fx="20110131122300000W">
    <Value AIi="0 Wh" AEi="0 Wh" R1i="0 varh" R2i="0 varh" R3i="0 varh" R4i="0 varh"/>
    <Value AIa="0 Wh" AEa="0 Wh" R1a="0 varh" R2a="0 varh" R3a="0 varh" R4a="0 varh"/>
   </S04>
   <S04 Fhi="20110131122300000W" Fhf="20110201114000000W" Ctr="1" Pt="4" Mx="0 W" Fx="20110131122300000W">
    <Value AIi="0 Wh" AEi="0 Wh" R1i="0 varh" R2i="0 varh" R3i="0 varh" R4i="0 varh"/>
    <Value AIa="0 Wh" AEa="0 Wh" R1a="0 varh" R2a="0 varh" R3a="0 varh" R4a="0 varh"/>
   </S04>
   <S04 Fhi="20110131122300000W" Fhf="20110201114000000W" Ctr="1" Pt="5" Mx="0 W" Fx="20110131122300000W">
    <Value AIi="0 Wh" AEi="0 Wh" R1i="0 varh" R2i="0 varh" R3i="0 varh" R4i="0 varh"/>
    <Value AIa="0 Wh" AEa="0 Wh" R1a="0 varh" R2a="0 varh" R3a="0 varh" R4a="0 varh"/>
   </S04>
   <S04 Fhi="20110131122300000W" Fhf="20110201114000000W" Ctr="1" Pt="6" Mx="0 W" Fx="20110131122300000W">
    <Value AIi="0 Wh" AEi="0 Wh" R1i="0 varh" R2i="0 varh" R3i="0 varh" R4i="0 varh"/>
    <Value AIa="0 Wh" AEa="0 Wh" R1a="0 varh" R2a="0 varh" R3a="0 varh" R4a="0 varh"/>
   </S04>
  </Cnt>
 </Cnc>
/Report>
```


Figure 3-17: XML representation of meter reading from ZIV prototype performed by the CURRENT Data Concentrator.

TC 6.4.5.2 – DLMS Firmware update

In the setup with the ZIV Data Concentrator, the DLMS firmware update was successfully performed on both ZIV Prototypes, as shown by the figure below:

Figure 3-18: Snapshot of ZIV DC Web interface after successful DLMS Firmware Update.

In the setup with ZIV Data Concentrator and LANDIS + GYR, the firmware update could not be completed. The loggings were investigated by the engineers of both involved companies, after which the issue was solved.

The firmware update with the CURRENT DC could not be executed. In the available time, it was not possible to figure out how to correctly operate the Command Line interface of the Data Concentrator in order to initiate DLMS FU.

This test with ZIV DC and ZIV meter was repeated after initial failure. The configuration of the DC was checked and the test has been repeated in a side room in parallel with the audited test that were going on. The final result is PASSED, as indicated in the screenshot shown above.

3.2.4.2 RUN2: ZIV DATA CONCENTRATOR, ELSTER and ITRON METER

In the table below, it is indicated which test cases from the sections 6.4.3 in [1] were executed during this test run. The test case id's in the table refer to the test cases in [1] "Definitions of test procedures".

TC id:	Executed (Y/N)
6.4.5.1	Y
6.4.5.2	Y

Table 3-11: Executed DLMS Interoperability test cases

All prototypes under test implemented the Spanish DLMS Profile.
The test cases that were executed were performed in a setup with the ZIV Data Concentrator.

A summary of test executed tests mapped on the prototype combinations is presented in ANNEX D.

The test results for both setups are described in detail in the sections below. In some test cases, the results are 'decorated' with some screenshot, to give the reader an impression on how the test details were obtained.

TC 6.4.5.1 DLMS – Read / Write Tests.

During this test, several read / write mechanisms were tested, as summarized in the table below:

	ZIV DC + ELSTER Meter	ZIV DC + ITRON Meter
DLMS Action	**OBIS Code, result**	**OBIS Code, result**
Association Request	NA, success	NA, success
GetRequest	{ 8,0-0:1.0.0.255,2} Clock – time, success.	{ 7,0-0:98.1.1.255,2} Clock - time.
GetRequestWith DataBlock	{ 7,0-0:98.1.1.255,2} Monthly billing values, success.	Not in loggings, (monthly billing object was not available).
SetRequest	{ 8,0-0:1.0.0.255,2} Clock - time, success.	Passedword for management client not accepted.
Association Release	NA, success.	NA, success.

The GetRequest(WithDataBlock) were done with the Reading Client (ID=2), whereas the SetRequest was done with the Management Client (ID=1).

The following pictures show some loggings of the DLMS data exchange between the ZIV DC and the ELSTER meter, and the ZIV DC and the ITRON Meter.

The figure below shows the logging of DLMS traffic between the ZIV DC and ELSTER Meter. An Association Request, a Set request and a Get Request are shown.

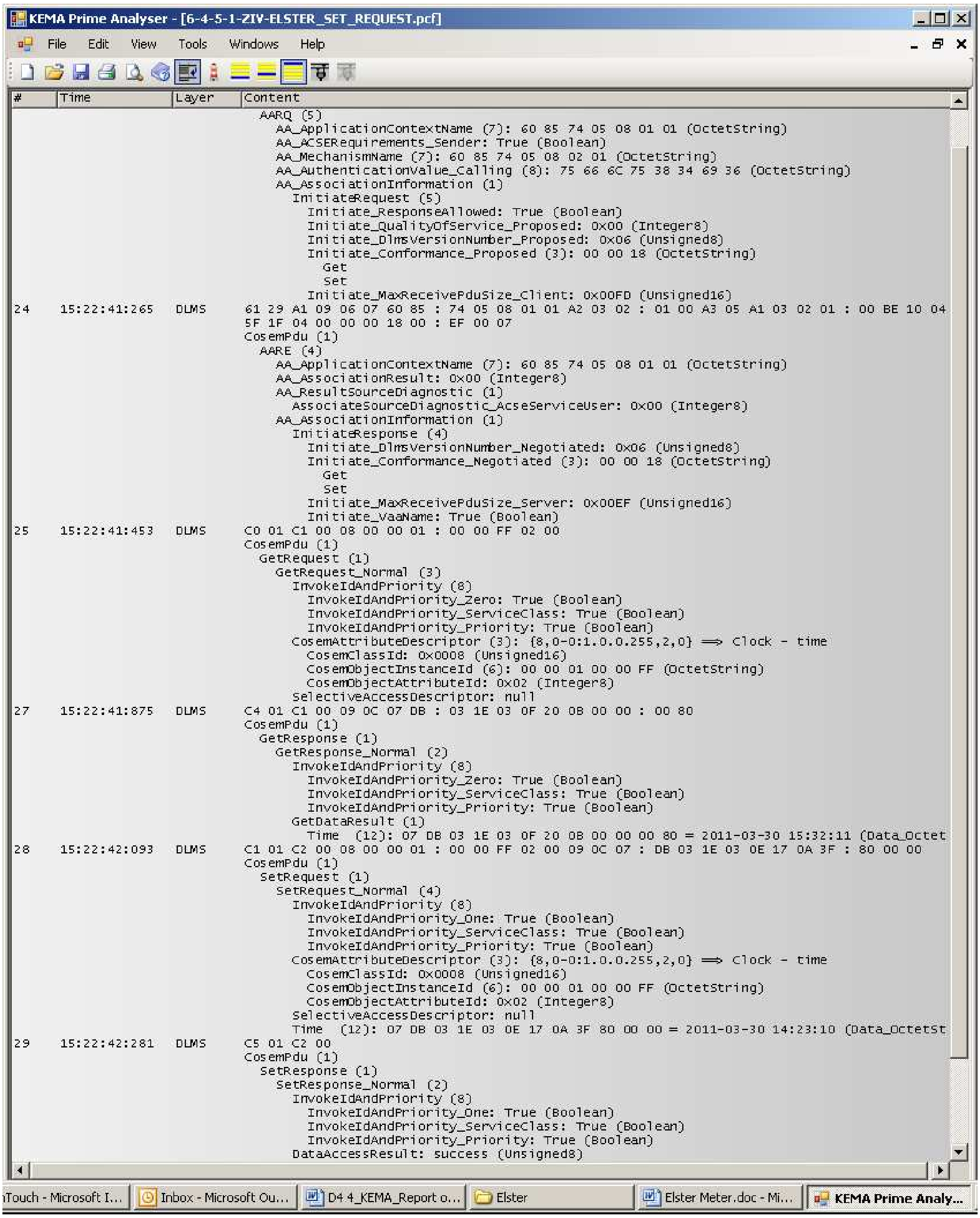


Figure 3-19: Logging of DLMS Traffic between ELSTER Meter and ZIV DC

The figure below shows the result of a GetRequest_withDataBlock.

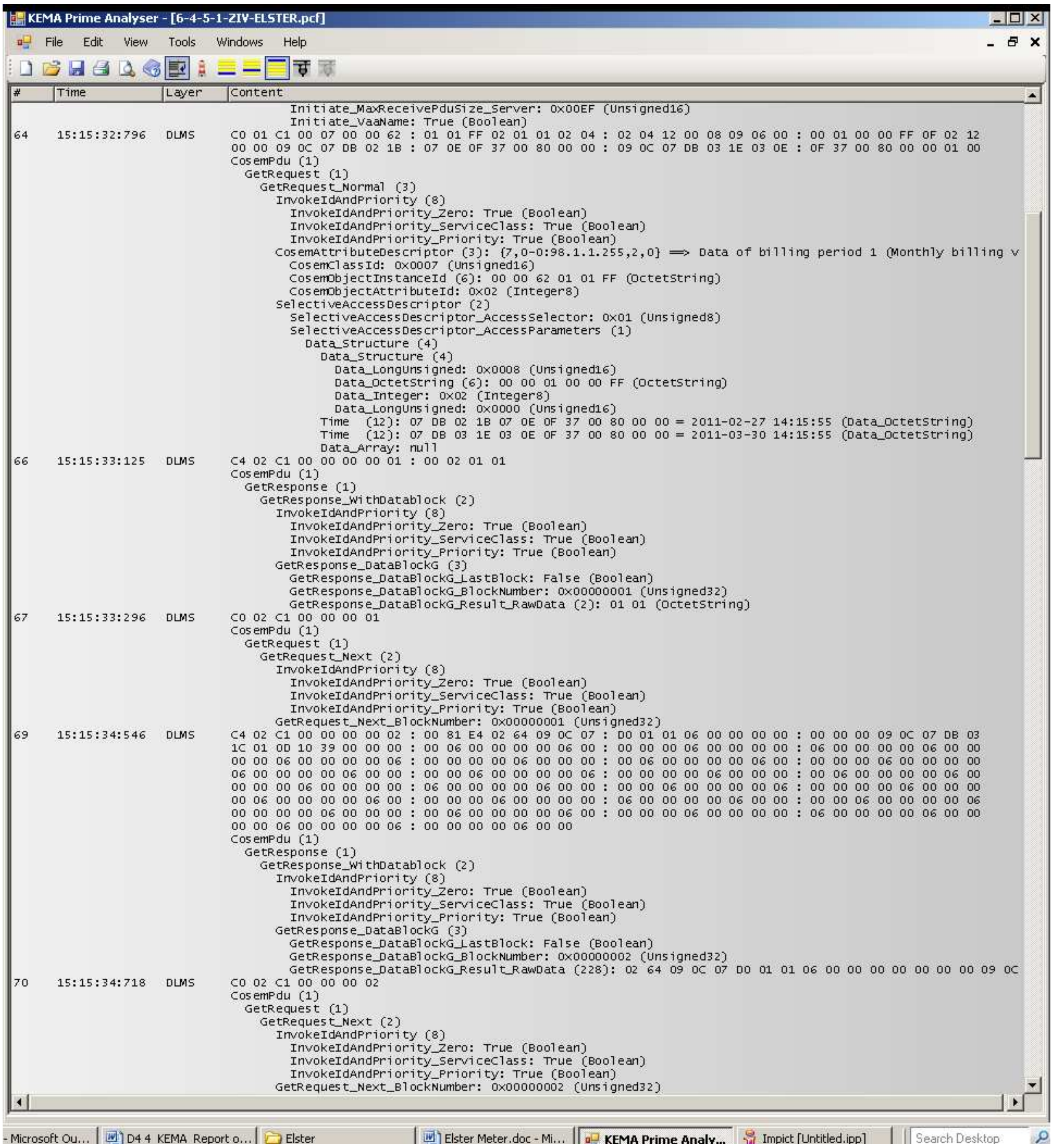


Figure 3-20: Logging of GetRequest_withDataBlock, initiated by the ZIV Data Concentrator and the ELSTER Meter.

The next logging shows the Association Request and Response, and the GetRequest and Response between the ZIV Data Concentrator and the ITRON Meter.

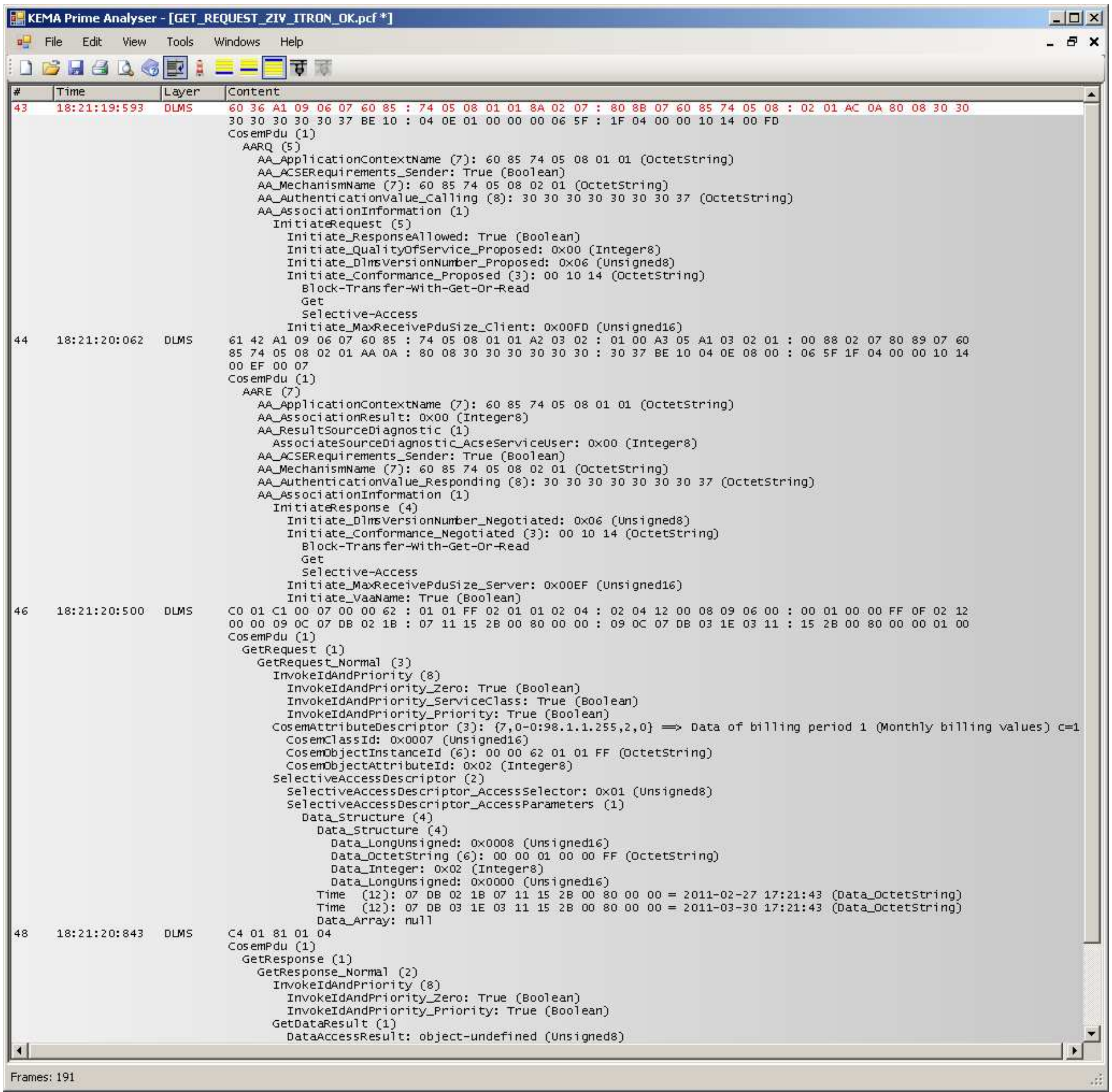


Figure 3-21: Logging of DLMS traffic between ITRON Meter and ZIV Data Concentrator.

Finally a logging is shown of a SetRequest, initiated from the ZIV Data Concentrator and the ITRON Meter. Although the Passedword is OK (ACE00000), the Association is not accepted for the Management client.
Due to the limited time available, no solution could be arranged.

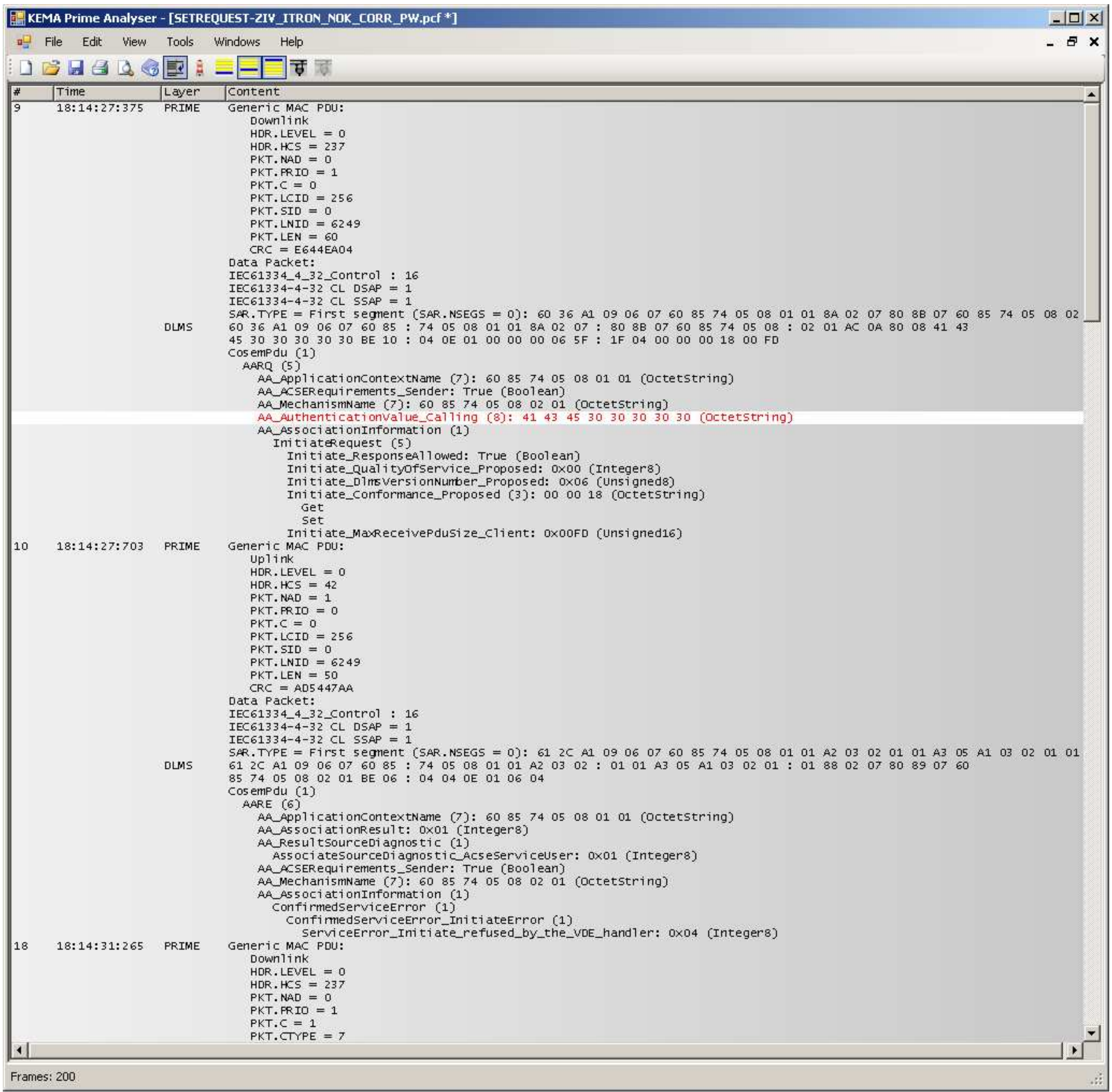


Figure 3-22: Rejected association request (AA_AssocitationResult: 0x01) for management client.

TC 6.4.5.2 DLMS Firmware Update.

ITRON has indicated that no firmware update was supported in the meter that was provided.

During the tests with the ELSTER Meter, the DLMS Firmware Update was triggered from the web interface of the started successfully, however the interface of the DC kept in the state "Receiving" for ~45 minutes. The loggings were stored and sent to ZIV and ELSTER for further investigations.

3.2.5 Results of DLMS Functional Tests

DLMS Functional tests were done at KEMA and ELSTER prototypes. (See Table 3-1). These PRIME meters implement the 'Spanish' DLMS profile.

The tests with the LANDIS + GYR meter was done at KEMA Arnhem, in the period February 2011 – march 2011.

Two DLMS functional tests have been performed with the ELSTER prototype.
These tests were done with at the same date and with the same auditors as in run 2 of the DLMS / PRIME Interoperability tests (see **Error! Reference source not found.**).
The results of these tests are presented in the next sections:

TC 4.4.19.1 Test Voltage Variations – Lower Threshold.
As a preparation, the time and voltage limits as defined in the test case were programmed in the meter via an separate test script.
The test facility checked for the presence of the events at the steps as indicated in the test case description. The test PASSED.

TC 4.4.20.1 Test Voltage Variations – Higher Threshold.
As a preparation, the time and voltage limits as defined in the test case were programmed in the meter via an separate test script.
The test facility checked for the presence of the events at the steps as indicated in the test case description. The test PASSED, although the connection was lost just before step 18 of the test case.
The picture below shows the events that were logged, after a read action with the manufacturer tool:

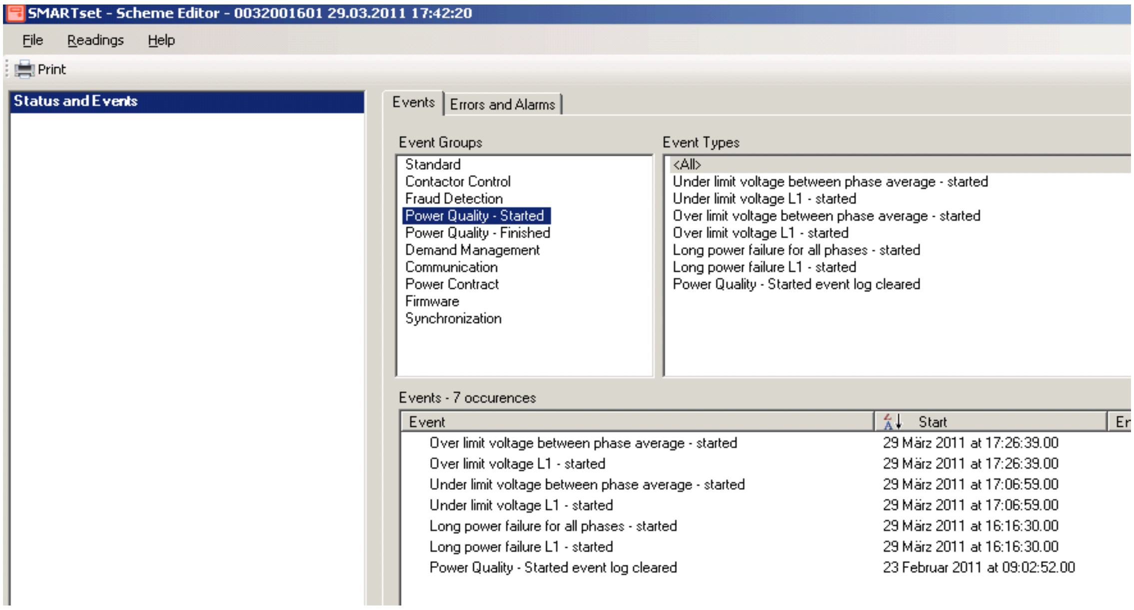


Figure 3-23: Events that were logged, after a read action with the manufacturer tool

3.3 Report on interoperability field test results for DLMS and PRIME

3.3.1 Test Setup

The figure below shows the location of the tests. User interfaces from the Data Concentrator webs were showed 'live' on the monitor-screen.

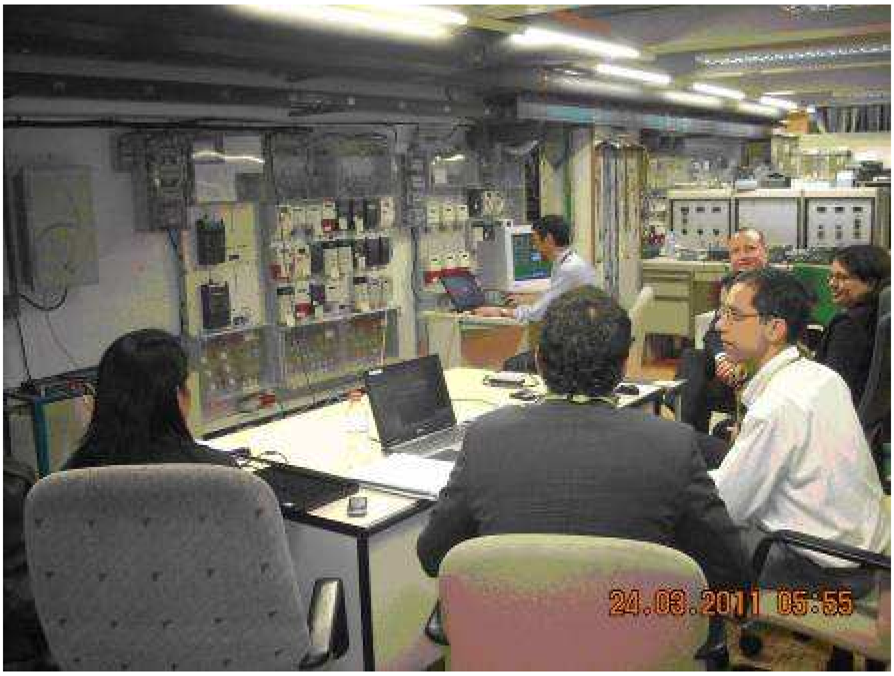


Figure 3-24: Testing and auditing at Iberdrola lab in Bilbao.

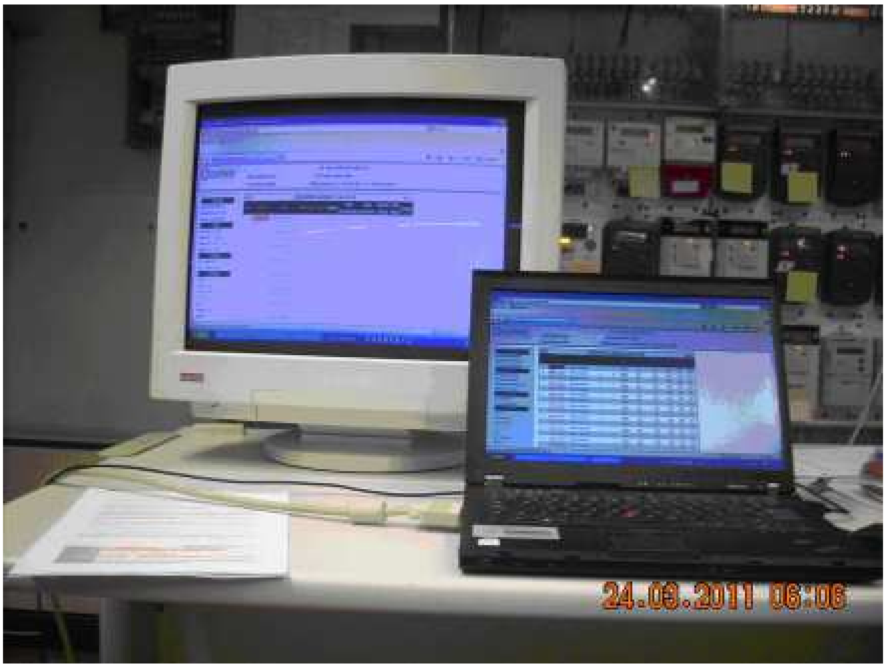


Figure 3-25: Monitor-screen to follow the tests.

3.3.1.1 Test setup for DLMS and PRIME Interoperability tests

The test setup used for the DLMS and PRIME Interoperability tests is described in [2], section 3.3.

The concentrator manufacturers have provided to the lab with a web interface to make possible the performing of the test cases that are described in [1] section 7.4.1.

3.3.2 Results of lab and field Interoperability tests

In this section, the tests done at IBERDROLA offices in Bilbao are described.

3.3.2.1 Test Results: ZIV+CURRENT DC, LANDIS + GYR and ZIV Meters.

In the table below, it is indicated which test cases from the sections 7.4.1 in [1] were executed. The test case id's in the table refer to the test cases in [1] "Definitions of test procedures".

TC id:	Lab Executed (Y/N/NA)	Results (OK/KO)	Field Executed (Y/N/NA)	Results (OK/KO)
7.4.1.1	Y	OK - success	Y	OK - success
7.4.1.2	Y	OK - success	Y	OK - success
7.4.1.3	Y	OK - success	NA	-
7.4.1.4	Y	OK - success	NA	-

7.4.1.5	Y	OK - success	NA	-
7.4.1.6	Y	OK - success	Y	OK - success
7.4.1.7	Y	OK - success	Y	OK - success
7.4.1.8	Y	OK - success	Y	OK - success
7.4.1.9	Y	OK - success	Y	OK - success
7.4.1.10	Y	OK - success	Y	OK - success

Table 3-12: Executed lab and field Interoperability test cases.

The test cases that were executed were performed in five setups:

a) Setup 1 in the lab, with the Data Concentrator from CURRENT as the Base Node in the PRIME Network.

Figure 3-26: PRIME+DLMS prototypes available in Iberdrola lab connected to CURRENT DC

b) Setup 2 in the lab, with the Data Concentrator from ZIV as Base Node in the PRIME Network.

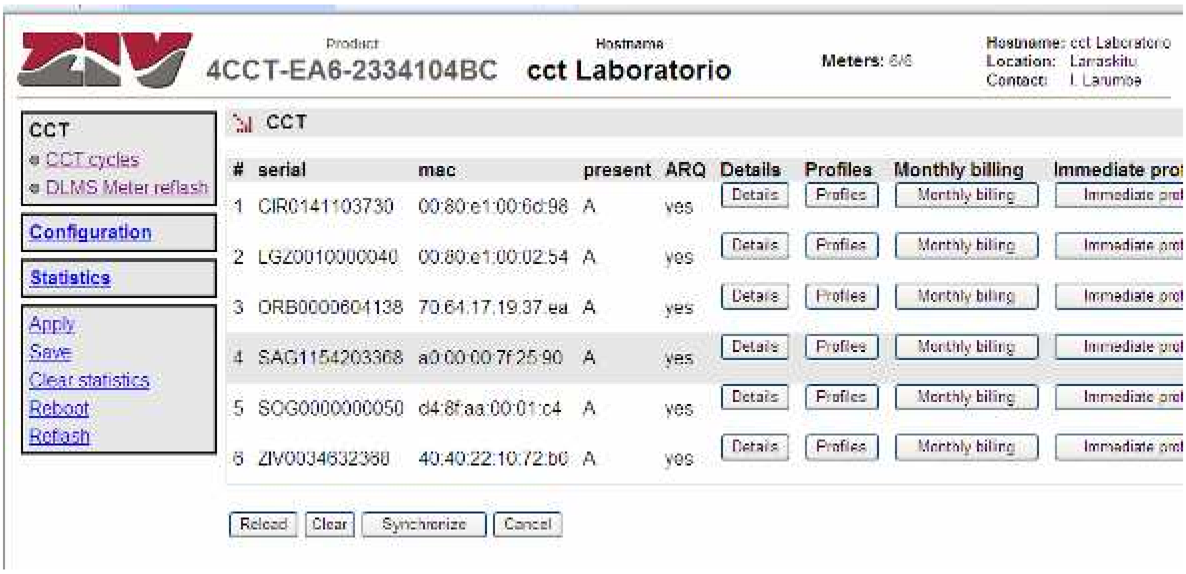


Figure 3-27: PRIME+DLMS meter prototypes available in Iberdrola lab connected to ZIV DC

c) Setup 3 in field, with the Data Concentrator from ZIV as Base Node in the PRIME Network against ZIV meters.

The secondary substation was Alcora 105, that has 39 ZIV meters connected. All of them are active.

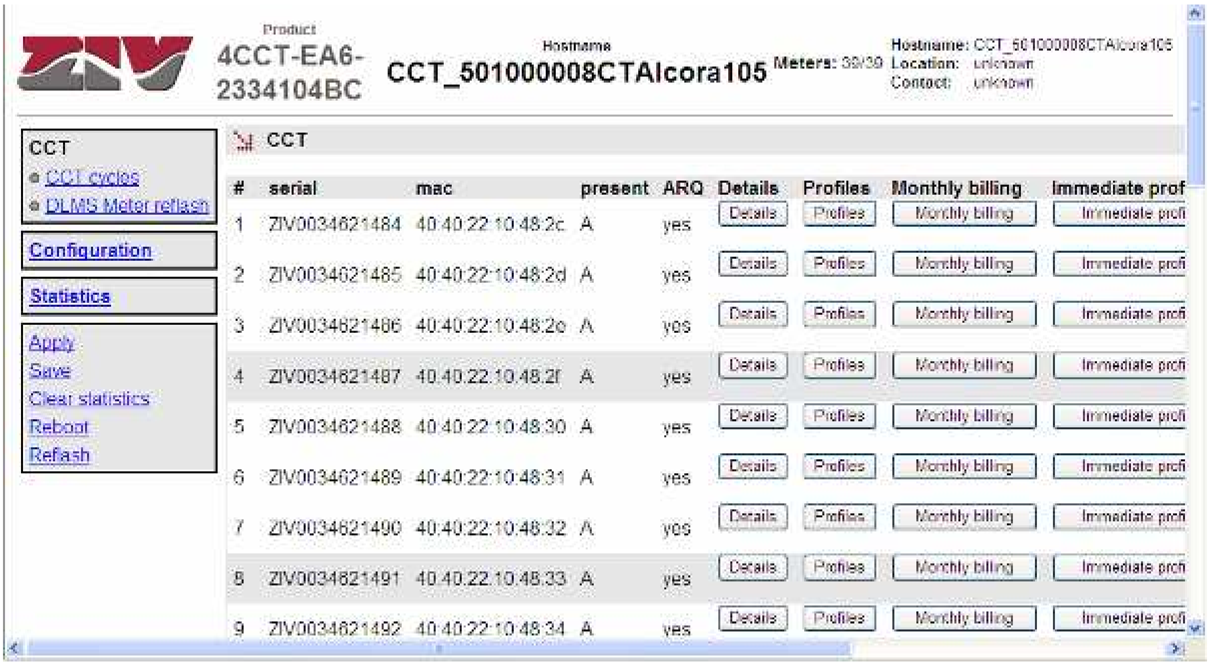


Figure 3-28: PRIME+DLMS ZIV meter prototypes available in field connected to ZIV DC

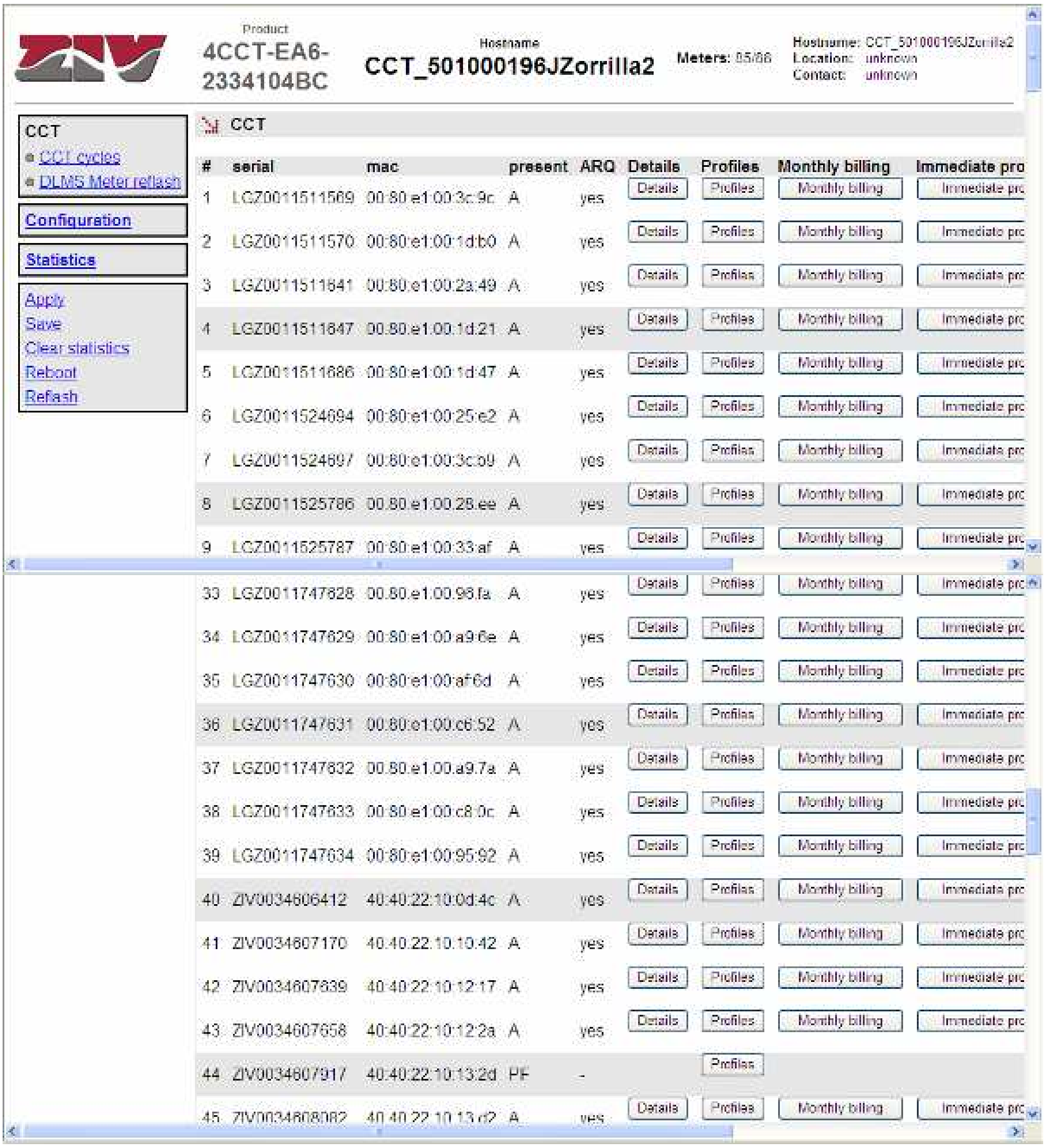
d) Setup 4 in field, with the Data Concentrator from CURRENT as Base Node in the PRIME Network against ZIV meters.

The secondary station was Las Rederas, that has 70 ZIV meters connected.

Figure 3-29: PRIME+DLMS ZIV meter prototypes available in field connected to CURRENT DC

e) Setup 5 in field, with the Data Concentrator from ZIV as Base Node in the PRIME Network against CURRENT meters.

The secondary substation was Jose Zorrilla 2, that has 88 total meters connected, which 39 are LANDIS + GYR meters and 49 are ZIV meters. That means there is **no problem of coexistence of different meters manufacturers** at all.



#	serial	mac	present	ARQ	Details	Profiles	Monthly billing	Immediate pro
1	LGZ0011511569	00:80:e1:00:3c:9c	A	yes	Details	Profiles	Monthly billing	Immediate pro
2	LGZ0011511570	00:80:e1:00:1d:b0	A	yes	Details	Profiles	Monthly billing	Immediate pro
3	LGZ0011511641	00:80:e1:00:2a:49	A	yes	Details	Profiles	Monthly billing	Immediate pro
4	LGZ0011511647	00:80:e1:00:1d:21	A	yes	Details	Profiles	Monthly billing	Immediate pro
5	LGZ0011511686	00:80:e1:00:1d:47	A	yes	Details	Profiles	Monthly billing	Immediate pro
6	LGZ0011524694	00:80:e1:00:25:e2	A	yes	Details	Profiles	Monthly billing	Immediate pro
7	LGZ0011524697	00:80:e1:00:3c:b9	A	yes	Details	Profiles	Monthly billing	Immediate pro
8	LGZ0011525786	00:80:e1:00:28:ee	A	yes	Details	Profiles	Monthly billing	Immediate pro
9	LGZ0011525787	00:80:e1:00:33:af	A	yes	Details	Profiles	Monthly billing	Immediate pro
33	LGZ0011747628	00:80:e1:00:98:fa	A	yes	Details	Profiles	Monthly billing	Immediate pro
34	LGZ0011747629	00:80:e1:00:a9:6e	A	yes	Details	Profiles	Monthly billing	Immediate pro
35	LGZ0011747630	00:80:e1:00:af:6d	A	yes	Details	Profiles	Monthly billing	Immediate pro
36	LGZ0011747631	00:80:e1:00:c6:52	A	yes	Details	Profiles	Monthly billing	Immediate pro
37	LGZ0011747632	00:80:e1:00:a9:7a	A	yes	Details	Profiles	Monthly billing	Immediate pro
38	LGZ0011747633	00:80:e1:00:c8:0c	A	yes	Details	Profiles	Monthly billing	Immediate pro
39	LGZ0011747634	00:80:e1:00:95:92	A	yes	Details	Profiles	Monthly billing	Immediate pro
40	ZIV0034606412	40:40:22:10:0d:4c	A	yes	Details	Profiles	Monthly billing	Immediate pro
41	ZIV0034607170	40:40:22:10:10:42	A	yes	Details	Profiles	Monthly billing	Immediate pro
42	ZIV0034607639	40:40:22:10:12:17	A	yes	Details	Profiles	Monthly billing	Immediate pro
43	ZIV0034607658	40:40:22:10:12:2a	A	yes	Details	Profiles	Monthly billing	Immediate pro
44	ZIV0034607917	40:40:22:10:13:2d	PF	-		Profiles		
45	ZIV0034608082	40:40:22:10:13:d2	A	yes	Details	Profiles	Monthly billing	Immediate pro

Figure 3-30: PRIME+DLMS ZIV and CURRENT meter prototypes available in field connected to ZIV DC

The test results with some additional information not covered in [1] for all the setups are described more in detail in the sections below.

This diagram depicts the different setups tested:

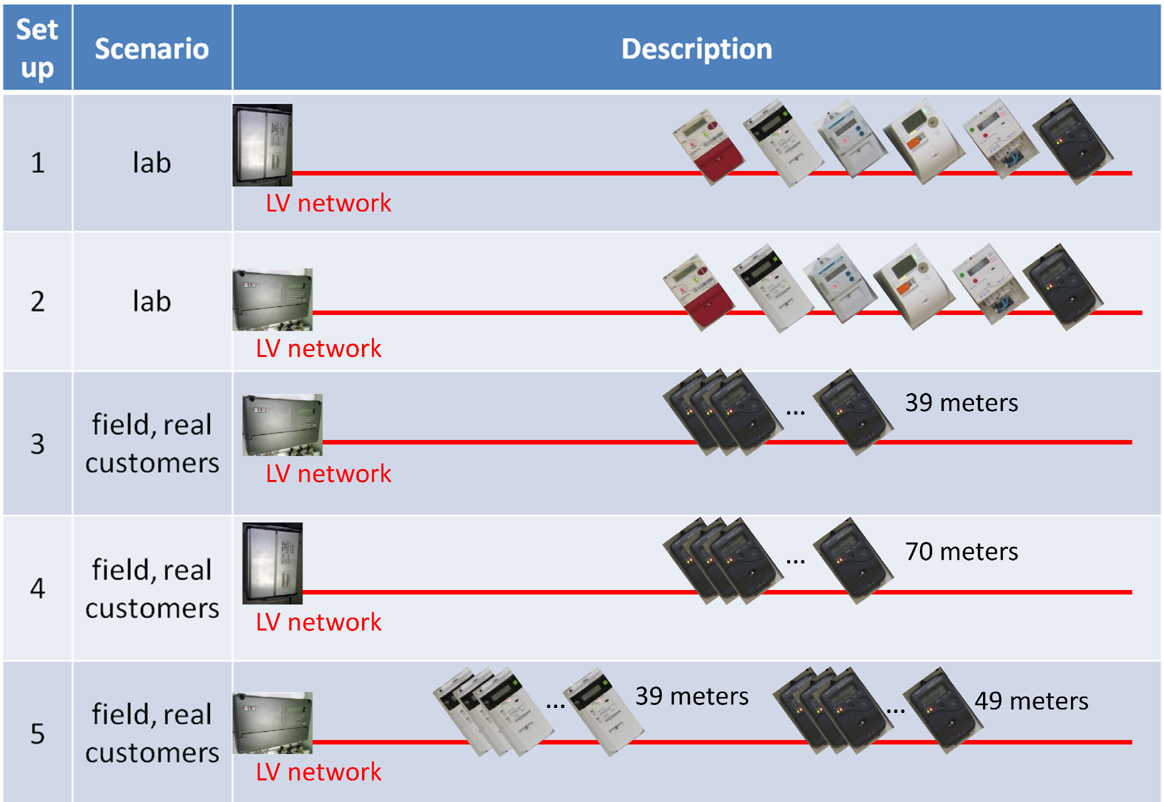


Figure 3-31: Diagram depicting the 5 configurations tested

TC 7.4.1.1 – Read Instantaneous values of the meter
We did the test clicking the "Details" button with ZIV DC and CURRENT DC against all the meter prototypes available in the lab on 24th of March.

TC 7.4.1.9 – Check the correct schedule of the events created.

We also have changed the power programmed to see that the event has been correctly generated.

TC 7.4.1.10 – Check the correct functionality of firmware upgrade.
PRIME ***multicast*** firmware upgrades and DLMS ***broadcast*** upgrades were tested instead of unicast mode as it is described in section 7.4.1.10 of [1]. In setup 3, the DLMS Firmware was transferred via the PRIME update procedure instead of via the DLMS FU procedure.

3.4 Conclusion and recommendations

3.4.1 General conclusions

The PRIME laboratory interoperability tests done at KEMA in both 'run1' and 'run2', have been successfully passed, since the most important PRIME mechanisms have been utilized during those tests.

The DLMS Interoperability Tests in the laboratory, have also been successfully passed showing several ways of data exchange on DLMS level can be performed.

With the prototypes used in both the PRIME and DLMS laboratory tests, it appeared that Firmware Update is the mechanism that is most sensitive to interoperability issues. It is recommended to pay extra attention to the multicast and unicast firmware update procedures in writing specifications, as well as in defining (conformance) test cases.

Within the scope of the OPEN Meter project, it was not mandatory for meters to be 'PRIME Certified'. However, it is highly recommended to have only certified devices in interoperability setups: When all involved devices have PASSED the common set of PRIME Conformance test cases, errors that are resulting from PRIME non-conformance of the prototypes are eliminated from the interoperability test results, so that the focus can be on the real interoperability issues (if present).

Although the PRIME standard (1.3e) describes a security mechanism, none of the prototypes has an implemented security scheme. So far, the PRIME Conformance test cases that are currently available do not contain test cases that verify the correct implementation of the security scheme described in the PRIME standard.

It is highly recommended to add test cases related to security in the set of PRIME Conformance test cases, so that this part of the standard must be implemented in PRIME Compliant devices. Currently, the security topic has attention within the PRIME Alliance and appropriate measures are under development, which will impact near future versions of the PRIME Conformance test procedure and, as a result, on the implementation of security mechanisms in PRIME devices.

The DLMS standard provides several data access security schemes: 'No Security', 'Low Level Security' (authentication of clients by a Passedword) and 'High Level Security' (authentication of client and server by a challenge-mechanism). Beside the data access security, the DLMS standards provides data transport security schemes: No encryption, or encryption of the DLMS APDU.

Choices for the data access security scheme and the data transport security scheme constitute together the security profile in force.

The 'Spanish' DLMS profile has chosen the security policy in which Low Level Security is chosen for data access security, and no-ciphering has been chosen for the data transport security. The current implementation of the prototypes with no encryption in the PRIME layers makes that all data, including Passedwords for management and firmware update clients, are transferred as clear text over the public low voltage network. Note also that the meters are equipped with a Disconnector switch. Attacks of hackers on this part of the meter will be disastrous for the acceptance of smart metering in Europe and beyond. It is therefore recommended to implement an enhanced security policy in the Spanish DLMS profile. This enhancement is currently being defined by the involved parties.

The performed PLC PRIME tests in at IBERDROLA show that PRIME meters fulfill all the functionalities that have been tested and defined in the OPEN meter project. The Devices Under Tests were not just prototypes, but they are already industrialized meters under current MID, what has made possible to run some tests in the field, under real-live operational conditions, and for real-live customers.

Furthermore, interoperability tests have been successfully executed in the field.

3.4.2 Recommendation for the PRIME standard

During the tests and/or the investigation of the results, two issues were found relate to the PRIME standard:

1. The specification of the length of the Device ID. The PRIME standard does not define the length of the 'Device Identifier'. (see section 5.5.3 of the PRIME specification version 1.3e). The Device Identifier is derived from the value of the 'COSEM Logical Device Name', object which is specified in the DLMS Blue book. The companion standard for the Spanish profile does specify the length (16 octets), however for clarity and for future use of PRIME with other DLMS companion standards, it is recommended to specify the length of the Device Identifier in the PRIME Standard itself. Members of OPEN Meter have communicated this to PRIME alliance members for further processing into standardization.

2. Firmware Update

Consider the case that the following conditions are fulfilled:

- The FU pages are larger than the segment size used by the SAR layer.

- ARQ is not used.

If in this situation, a FU page is segmented in two segments, and two segments get lost, the situation as follows can occur:

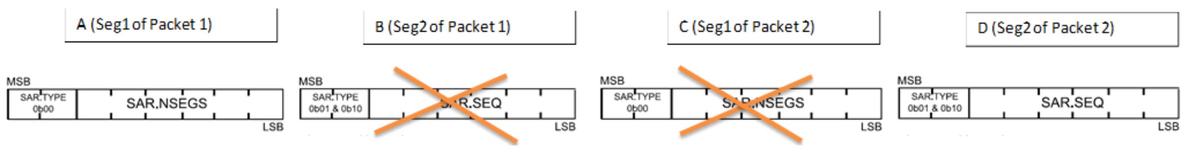


Figure 3-32: Losing some segments in a FU page.

At reassembly at the receiver, the first segment of the first page, and the second segment of the second segment are reassembled, without any information available for the FU application to be notified that something is getting wrong and a corrupt binary is built.

This issue will be further investigated in the PRIME TWG (where it is known as 'ticket 61').

4 REPORT ON TEST RESULTS FOR DLMS - G3

4.1 Introduction

In this section, the results for the PLC G3 tests performed at EDF R&D's Clamart facilities are described.

The PLC G3 tests described in this document were performed from 11th April to 26th April 2011, at the premises of EDF R&D in Clamart, France.
The OPEN METER PLC G3 test audit took place 19th April 2011 in EDF R&D's Clamart facilities. All tests were repeated during the audit. Some tests were repeated again after the audit in order to capture a more suitable logging.

The SAGEMCOM prototypes (DC and Meter) used in the tests are described in [2], section 4.3.7.

The following persons were involved during both the test period and the test audit:

Test Engineers:
A. Pajot, EDF R&D
R. Sebastien, EDF R&D
C. Lavenu, EDF R&D
V. Godefroy, EDF R&D
G. Alberdi, EDF R&D
Support Engineers:
T. Vernet, Sagemcom
Z. Roter, Sagemcom

Table 4-1: People involved in OPEN Meter testing at EDF

Auditors (On the 19th)
Michael ARZBERGER, Elster
Thomas SCHAUB, L+G
Vincent GUILLET, L+G
Auguste ANKOU, Itron
Philippe CHIUMMIENTO, Itron
Bas ROELOFSEN, Kema
Laura MARRON, ZIV
Iulene BEREINCUA, ZIV
Martin SIGLE, KIT Karlsruhe Institute of Technology
Carmelo RODRIGUEZ, Endesa
Francisco Manuel CABEZA TORRES, Endesa

Table 4-2: OPEN Meter Audit Witnesses

4.1.1 DLMS-G3 Laboratory tests

The next table presents an over view of the executed laboratory tests.

Test Cases	Described in section of [1]:	Prototypes:	Described in section of [2]
DLMS-G3 Lab tests	3.4.2.1;3.4.2.2; 4.4.8.1; 4.4.8.2; 4.4.8.3;4.4.11.1; 4.4.5.1;4.4.5.2	SagemCom Meter	4.3.7.1
		SagemCom DC	4.3.7.2

Table 4-3: Overview of the OPEN METER DLMS-G3 testing in the lab.

4.1.2 DLMS-G3 Field tests

The table below provides an overview of the performed test cases in a field setup:

Test Cases	Described in section of [1]:	Prototypes:	Described in section of [2]:
DLMS Field tests	4.4.6.1;4.4.6.2;4.4.13.1;4.4.3.1 4.4.2.1;4.4.4.1;4.4.4.2	SagemCom Meter	4.3.7.1
		SagemCom DC	4.3.7.2

Table 4-4: Overview of the OPEN METER PLC G3 testing in the field.

Figure 4-1: OPEN METER Audit session in Clamart

4.1.3 Test tools used for the tests

In this section, the test tools that were used in both the laboratory and field tests are described.

- **Central information server**

The central information system that grants de communication to the DC through an Ethernet port. In general terms, this central information server is in charge of sending the request and gathering the information coming from the PLC network (network discovery, reconfiguration,…). It configures as well the DC to accomplish the different tasks and manages the scheduler and the configuration within the DC.

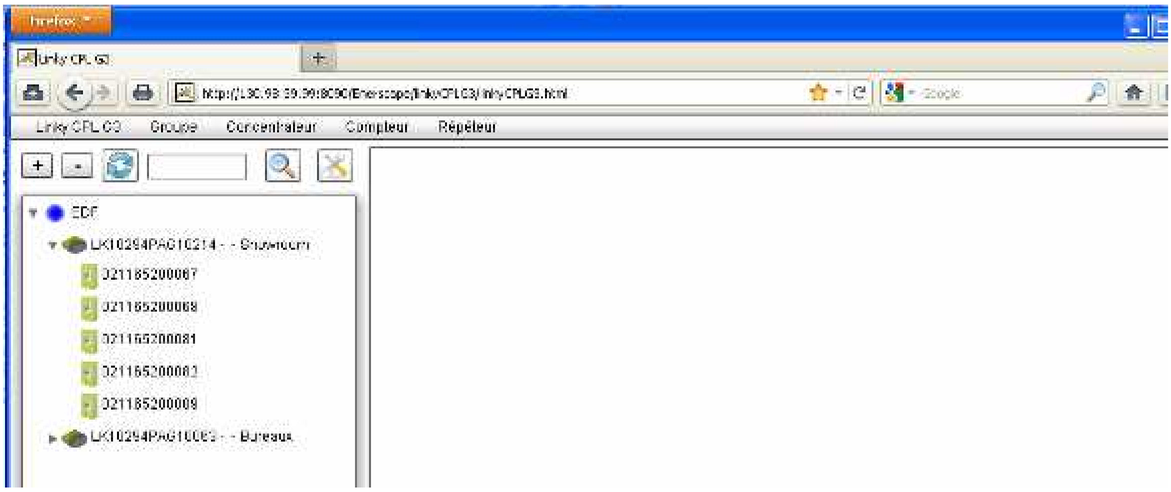


Figure 4-2: Example of the Central Information Server GUI interface based on web services

- **Data concentrator trace analyzer**

Internal software in charge of analyzing and displaying the data concentrator traces

```
CREATE TABLE PlcG3Device(EUI48 text, IPv6 text, Type text, SerailNumber text, VendorId text, HWVersion text, SWVersion text, ProductName text, Status text, DiscoveryDate integer, StatusChangeDate integer, LastConnectionDate integer);
INSERT INTO "PlcG3Device"
VALUES('c0d044abcd200068','fe80000000000000101200fffe000025','1','021165200068','null','null','0.25','021165200068','3',1303376118,1303376118,1303376118);
INSERT INTO "PlcG3Device"
VALUES('c0d044abcd200067','fe80000000000000101200fffe000026','1','021165200067','null','null','0.25','021165200067','3',1303376191,1303376191,1303376191);
INSERT INTO "PlcG3Device"
VALUES('c0d044abcd200082','fe80000000000000101200fffe000027','1','021165200082','null','null','0.25','021165200082','3',1303377701,1303377701,1303377701);
COMMIT;
```

Figure 4-3: Example of the data concentrator trace analyzer outcome

- **XML results files**

The central information server results are described as a XML file. This files includes the whole communication chain results as well as the communication features for a final end user exploitation purposes.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="211" execId="2011-04-21 17:32:20" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="SET">
   <request>
    <set>
     <object>BreakerState.control_state</object>
     <association>RW</association>
     <payload>
      <parameter name="control_state" type="enumerated" value="0"/>
     </payload>
    </set>
   </request>
   <start>2011-04-21 17:35:16 523</start>
   <stop>2011-04-21 17:35:22 993</stop>
   <retries>0</retries>
   <status>1</status>
   <result>
    <raw>0</raw>
    <decoded/>
   </result>
  </transaction>
 </target>
</taskExecution>
```


Figure 4-4: Example of a XML file result

4.2 Report on the DLMS-G3 Laboratory tests

4.2.1 Test Setup

The PLC G3 lab tests were performed in EDF R&D facilities in Clamart, France.

The precise test setup is described below:

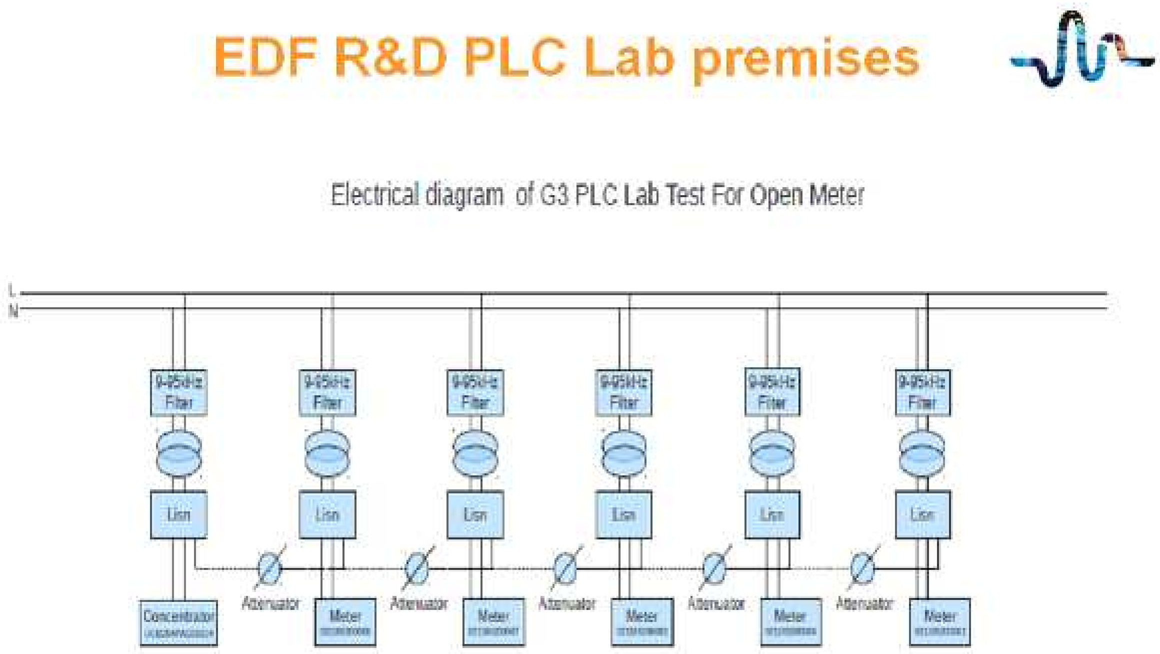


Figure 4-5: DLMS-G3 lab tests electrical diagram. The attenuators were used for the 2 PLC tests cases.

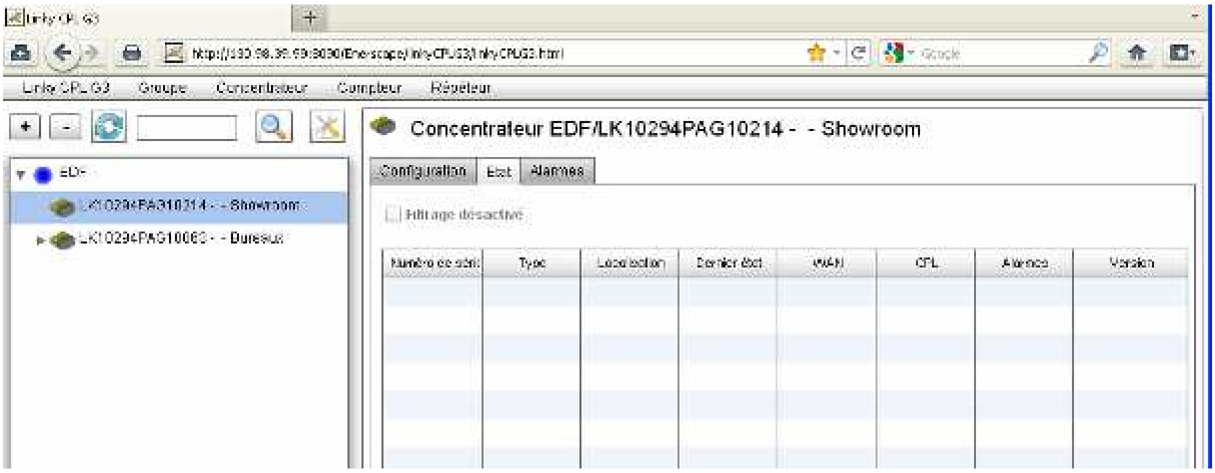
Figure 4-6: DLMS-G3 lab test setup.

4.2.2 Results of DLMS – G3 laboratory tests

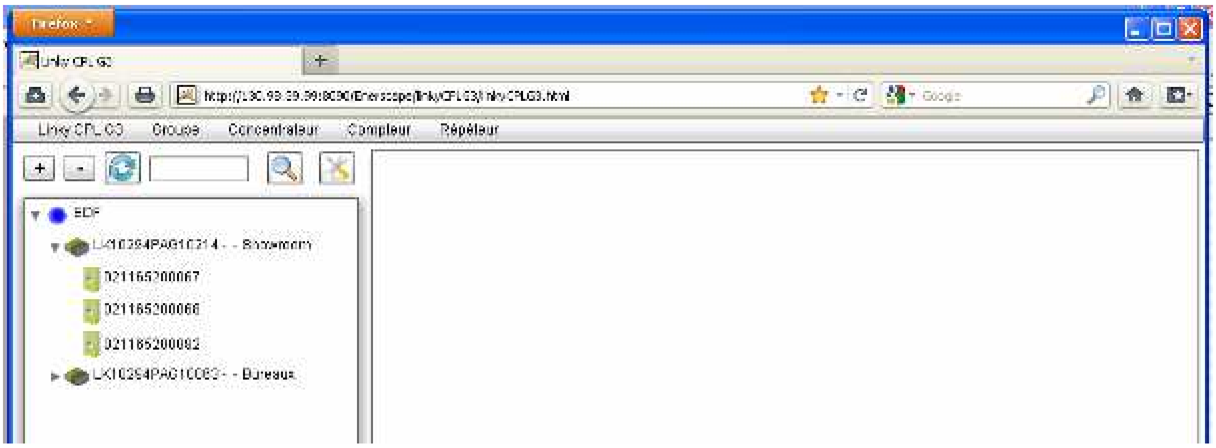
An overview of all test results is given in Annex F. In the next subsection, the results are described in more detail.

4.2.2.1 Test 3.4.2.1 - Plug & Play registering

First, all equipments are reset. As shown in the following figure, no meters are registered by the data concentrator :



meters S1, S2, S3 and S5 are now powered on. As expected, S1, S2 and S3 are discovered, but not S5 (meter S4, which is off, is needed to forward the data concentrator's messages) :

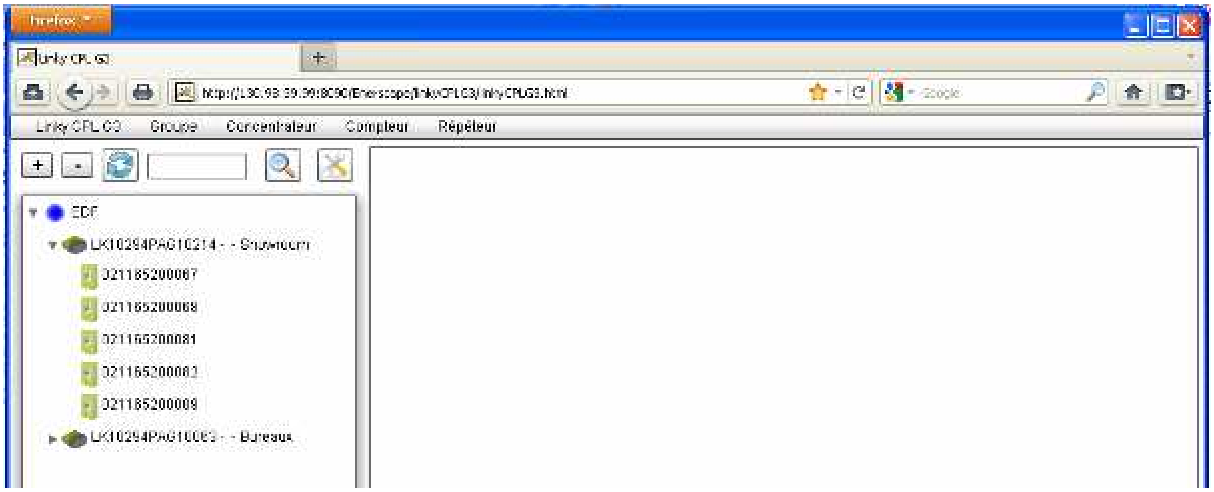


This can also be verified within the data concentrator's traces (only 3 devices identified by their IPv6 addresses) :

```
CREATE TABLE PlcG3Device(EUI48 text, IPv6 text, Type text, SerailNumber text, VendorId text, HWVersion text, SWVersion text, ProductName text, Status text, DiscoveryDate integer, StatusChangeDate integer, LastConnectionDate integer);
INSERT INTO "PlcG3Device"
VALUES('c0d044abcd200068','fe8000000000000101200fffe000025','1','021165200068','null','null','0.25','021165200068','3',1303376118,1303376118,1303376118);
```

```
INSERT INTO "PlcG3Device" VALUES('c0d044abcd200067','fe80000000000000101200fffe000026','1','021165200067','null','null','0.25','021165200067','3',1303376191,1303376191,1303376191);
INSERT INTO "PlcG3Device" VALUES('c0d044abcd200082','fe80000000000000101200fffe000027','1','021165200082','null','null','0.25','021165200082','3',1303377701,1303377701,1303377701);
COMMIT;
```

Then, meter S4 is powered on. Finally S4 and S5 are discovered and the data concentrator is able to send data to all installed meters:



Data concentrator traces :

```
CREATE TABLE PlcG3Device(EUI48 text, IPv6 text, Type text, SerailNumber text, VendorId text, HWVersion text, SWVersion text, ProductName text, Status text, DiscoveryDate integer, StatusChangeDate integer, LastConnectionDate integer);
INSERT INTO "PlcG3Device" VALUES('c0d044abcd200068','fe80000000000000101200fffe000025','1','021165200068','null','null','0.25','021165200068','3',1303376118,1303376118,1303376118);
INSERT INTO "PlcG3Device" VALUES('c0d044abcd200067','fe80000000000000101200fffe000026','1','021165200067','null','null','0.25','021165200067','3',1303376191,1303376191,1303376191);
INSERT INTO "PlcG3Device" VALUES('c0d044abcd200082','fe80000000000000101200fffe000027','1','021165200082','null','null','0.25','021165200082','3',1303377701,1303377701,1303377701);
INSERT INTO "PlcG3Device" VALUES('c0d044abcd200089','fe80000000000000101200fffe000029','1','021165200089','null','null','0.25','021165200089','3',1303378141,1303378141,1303378141);
INSERT INTO "PlcG3Device" VALUES('c0d044abcd200081','fe80000000000000101200fffe00002a','1','021165200081','null','null','0.25','021165200081','3',1303378184,1303378184,1303378184);
COMMIT;
~ #
```

4.2.2.2 Test 3.4.2.2 - Network topology changes behavior

NOTE: The tests regarding the network topology changes behavior was performed in the previous test, ID 3.4.2.1

4.2.2.3 Test 4.4.2.1 - Remote tariff programming

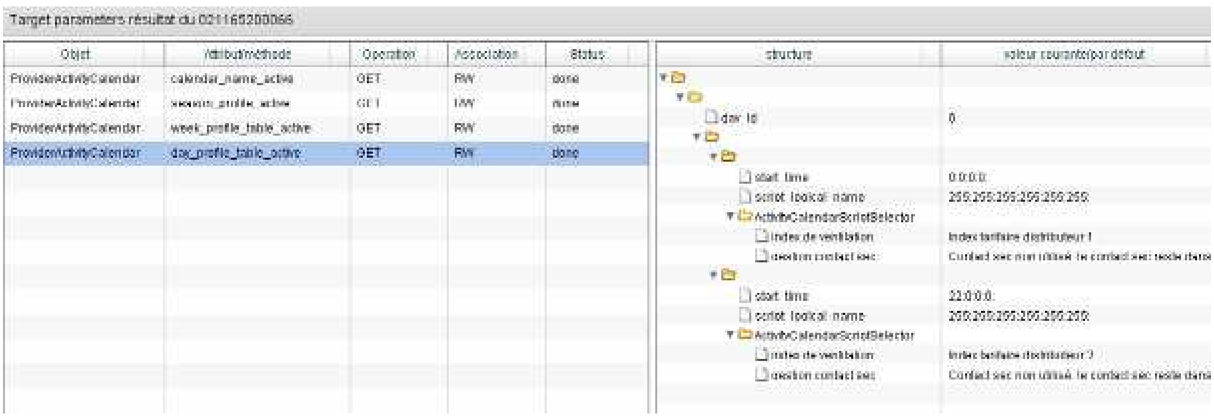
ImportActiveEnergyProvider objects are read for all tariffs :

Then, a PassediveActivityCalendar object is programmed :


```xml
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="190" execId="2011-04-19 14:50:17" date="">
  <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200066" dc="LK10294PAG10214">
    <transaction api="DLMS" id="1" type="SET">
      <request>
        <set>
          <object>ProviderActivityCalendar.day_profile_table_Passedive</object>
          <association>RW</association>
          <payload>
            <parameter name="entries" type="array">
              <parameter name="entry1" type="struct">
                <parameter name="day_id" type="unsigned8" value="0"/>
                <parameter name="subentries" type="array">
                  <parameter name="subentry1" type="struct">
                    <parameter name="start_time" type="octetstring" value=";0;0;0;0"/>
                    <parameter name="script_logical_name" type="octetstring" value="255;255;255;255;255;255"/>
                    <parameter name="script_selector" type="unsigned16" value="1"/>
                  </parameter>
                  <parameter name="subentry2" type="struct">
                    <parameter name="start_time" type="octetstring" value=";22;0;0;0"/>
                    <parameter name="script_logical_name" type="octetstring" value="255;255;255;255;255;255"/>
                    <parameter name="script_selector" type="unsigned16" value="2"/>
                  </parameter>
                </parameter>
              </parameter>
            </parameter>
          </payload>
        </set>
      </request>
      <start>2011-04-19 14:52:46 391</start>
      <stop>2011-04-19 14:52:49 997</stop>
      <retries>0</retries>
      <status>1</status>
      <result>
        <raw>0</raw>
        <decoded/>
      </result>
    </transaction>
    <transaction api="DLMS" id="2" type="SET">
      <request>
        <set>
          <object>ProviderActivityCalendar.activate_Passedive_calendar_time</object>
```

```
        <association>RW</association>
        <payload>
          <parameter name="activate_Passedive_calendar_time" type="octetstring" value="7;219;4;19;255;255;255;255;255;255;255;255;"/>
        </payload>
      </set>
    </request>
    <start>2011-04-19 14:52:50 64</start>
    <stop>2011-04-19 14:52:53 687</stop>
    <retries>0</retries>
    <status>1</status>
    <result>
      <raw>0</raw>
      <decoded/>
    </result>
  </transaction>
  <transaction api="DLMS" id="3" type="SET">
    <request>
      <set>
        <object>CurrentDateAndTime.time</object>
        <association>RW</association>
        <payload>
          <parameter name="time" type="octetstring" value="7;219;4;19;255;21;59;0;0;128;0;0"/>
        </payload>
      </set>
    </request>
    <start>2011-04-19 14:52:53 744</start>
    <stop>2011-04-19 14:52:56 867</stop>
    <retries>0</retries>
    <status>1</status>
    <result>
      <raw>0</raw>
      <decoded/>
    </result>
  </transaction>
 </target>
</taskExecution>
```


Successful programming can be verified afterwards by reading the PassediveActivityCalendar object as shown in the figure below :



4.2.2.4 Test 4.4.3.1 - Meter reading (on demand)

For this test, a field test meter connected to a heating load has been used.
After setting a time frame in the DC's internal scheduler, we have proceed to the register reading by changing the current time.

First of all, we have proceed to define the remote reading tasks:

196	ReadEAIP	METER	Expert	2011-04-19 15:14:33	2011-04-20 15:13:59		DONE
195	Demain	METER	Expert	2011-04-19 15:12:06	2011-04-20 15:11:01		DONE
194	ReadEAIP	METER	Expert	2011-04-19 15:09:28	2011-04-20 15:08:56		DONE

Afterwards, to be sure of the register's value, we have done a first direct read as seen below:

Target parameters résultat du 021165200066

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par défau
ImportActiveEnergyPro	value	GET	RW	done	value	27184
ImportActiveEnergyPro	value	GET	RW	done		

Finally, we change the clock so that the meter reading on demand can be accomplishes by the system:


```xml
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="195" execId="2011-04-19 15:12:06" date="">
  <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200066" dc="LK10294PAG10063">
    <transaction api="DLMS" id="1" type="SET">
      <request>
        <set>
          <object>CurrentDateAndTime.time</object>
          <association>RW</association>
          <payload>
            <parameter name="time" type="octetstring" value="7;219;4;20;255;21;59;0;0;128;0;0"/>
          </payload>
        </set>
      </request>
      <start>2011-04-19 15:14:31 333</start>
      <stop>2011-04-19 15:14:38 521</stop>
      <retries>0</retries>
      <status>1</status>
      <result>
        <raw>0</raw>
        <decoded/>
      </result>
    </transaction>
  </target>
</taskExecution>
```


And the consequent register value:

Target parameters résultat du 021165200066

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par de
ImportActiveEnergyPro	value	GET	RW	done	value	27322
ImportActiveEnergyPro	value	GET	RW	done		

4.2.2.5 Test 4.4.5.1 - Remote disconnection/re-connection

First breaker disconnection is requested :


```xml
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="211" execId="2011-04-21 17:32:20" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="SET">
   <request>
    <set>
     <object>BreakerState.control_state</object>
     <association>RW</association>
     <payload>
      <parameter name="control_state" type="enumerated" value="0"/>
     </payload>
    </set>
   </request>
   <start>2011-04-21 17:35:16 523</start>
   <stop>2011-04-21 17:35:22 993</stop>
   <retries>0</retries>
   <status>1</status>
   <result>
    <raw>0</raw>
    <decoded/>
   </result>
  </transaction>
 </target>
</taskExecution>
```


Once, breaker disconnection is effective, it can be checked by reading BreakerState control state :

Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par défaut
BreakerState	control_state	SET	RW	done	control_state	0

Then BreakerState control state is set back to “connected” :


```xml
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="214" execId="2011-04-21 18:12:42" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="SET">
   <request>
    <set>
     <object>BreakerState.control_state</object>
     <association>RW</association>
     <payload>
      <parameter name="control_state" type="enumerated" value="1"/>
     </payload>
    </set>
   </request>
   <start>2011-04-21 18:15:48 237</start>
   <stop>2011-04-21 18:15:54 600</stop>
   <retries>0</retries>
```

```
    <status>1</status>
    <result>
      <raw>0</raw>
      <decoded/>
    </result>
  </transaction>
 </target>
</taskExecution>
```


Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par défaut
BreakerState	control_state	GET	RW	done	control_state	1

The previous steps are repeated 3 times, what can be checked by reading the BreakerOpeningCounter object :

Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par défaut
BreakerOpeningCount	value	GET	RW	done	value	3

4.2.2.6 Test 4.4.5.2 - Scheduled disconnection/re-connection

First a PowerCutSchedule object is configured that power cut starts on the 20/04/2011 at 00:00 during 5 minutes :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par défaut
PowerCutSchedule	entries	SET	RW	done		


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="171" execId="2011-04-19 10:49:39" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200068" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="SET">
   <request>
    <set>
     <object>PowerCutSchedule.entries</object>
     <association>RW</association>
     <payload>
      <parameter name="entries" type="array">
       <parameter name="entry1" type="struct">
        <parameter name="index" type="unsigned16" value="1"/>
        <parameter name="enable" type="boolean" value="0"/>
        <parameter name="script_logical_name" type="octetstring" value="255;255;255;255;255;255;"/>
        <parameter name="script_selector" type="unsigned16" value="0"/>
        <parameter name="switch_time" type="octetstring" value=";0;0;0;0"/>
        <parameter name="validity_windows" type="unsigned16" value="0"/>
        <parameter name="exec_weekday" type="bitstring" size="7" value="00"/>
        <parameter name="exec_specday" type="bitstring" size="9" value="0000"/>
        <parameter name="begin_date" type="octetstring" value="7;219;4;20;255"/>
        <parameter name="end_date" type="octetstring" value="255;255;255;255;255;"/>
       </parameter>
       <parameter name="entry2" type="struct">
        <parameter name="index" type="unsigned16" value="2"/>
        <parameter name="enable" type="boolean" value="0"/>
```

```
            <parameter name="script_logical_name" type="octetstring" value="255;255;255;255;255;255;"/>
            <parameter name="script_selector" type="unsigned16" value="0"/>
            <parameter name="switch_time" type="octetstring" value=";0;5;0;0"/>
            <parameter name="validity_windows" type="unsigned16" value="0"/>
            <parameter name="exec_weekday" type="bitstring" size="7" value="00"/>
            <parameter name="exec_specday" type="bitstring" size="9" value="0000"/>
            <parameter name="begin_date" type="octetstring" value="7;219;4;20;255"/>
            <parameter name="end_date" type="octetstring" value="255;255;255;255;255;"/>
          </parameter>
        </parameter>
      </payload>
    </set>
  </request>
  <start>2011-04-19 10:52:07 692</start>
  <stop>2011-04-19 10:52:12 153</stop>
  <retries>0</retries>
  <status>1</status>
  <result>
    <raw>0</raw>
    <decoded/>
  </result>
 </transaction>
</target>
</taskExecution>
```


Then, CurrentDateAndTime of meter “021165200068” is set to “19/04/2011 – 23:59” :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par défaut
CurrentDateAndTime	time	SET	RW	done		


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="171" execId="2011-04-19 10:51:41" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200068" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="SET">
   <request>
    <set>
     <object>CurrentDateAndTime.time</object>
     <association>RW</association>
     <payload>
      <parameter name="time" type="octetstring" value="7;219;4;19;255;23;59;0;0;128;0;0"/>
     </payload>
    </set>
   </request>
   <start>2011-04-19 10:54:07 777</start>
   <stop>2011-04-19 10:54:10 872</stop>
   <retries>0</retries>
   <status>1</status>
   <result>
    <raw>0</raw>
    <decoded/>
   </result>
  </transaction>
 </target>
</taskExecution>
```

BreakerState control state is read once disconnection is effective (i.e. on 20/04/2011 at 00:00 meter current time and date) :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status
BreakerState	control_state	GET	RW	done

structure	valeur courante/par défaut
control_state	0

ActiveEnergyImportProvider objects are read two times during the scheduled power cut, and as expected the value does not vary in time :

First reading (value = 622) :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status
ImportActiveEnergyPro	value	GET	public	done
ImportActiveEnergyPro	value	GET	public	done

structure	valeur courante/par défaut
value	622

Second reading (value = 622) :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status
ImportActiveEnergyPro	value	GET	public	done
ImportActiveEnergyPro	value	GET	public	done

structure	valeur courante/par défaut
value	622

After 5 minutes (i.e. on 20/04/2011 at 00:05 meter current time and date), BreakerState control state is checked. As expected, the breaker is back in connected state :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status
BreakerState	control_state	GET	RW	done

structure	valeur courante/par défaut
control_state	1

This is double checked by retrieving ActiveEnergyImportProvider objects within a few minutes time interval :

Reading at t1 (value = 629 Wh) :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status
ImportActiveEnergyPro	value	GET	public	done
ImportActiveEnergyPro	value	GET	public	done

structure	valeur courante/par défaut
value	629

Reading a few minutes after t1 (value = 632 Wh) :

Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status
ImportActiveEnergyPro	value	GET	public	done
ImportActiveEnergyPro	value	GET	public	done

structure	valeur courante/par défaut
value	632

4.3 Report on DLMS-G3 Field tests

4.3.1 Test Setup

The field tests were performed through the MENOFIS platform at EDF R&D site. EDF has installed several meters all around its building, with different equipments plugged behind them (computers, heating system, air conditioning, etc…) as shown on the pictures below.

The different meters as well as the DC were located as defined by the electrical scheme:

Figure 4-7: EDF R&D field test facility, Clamart

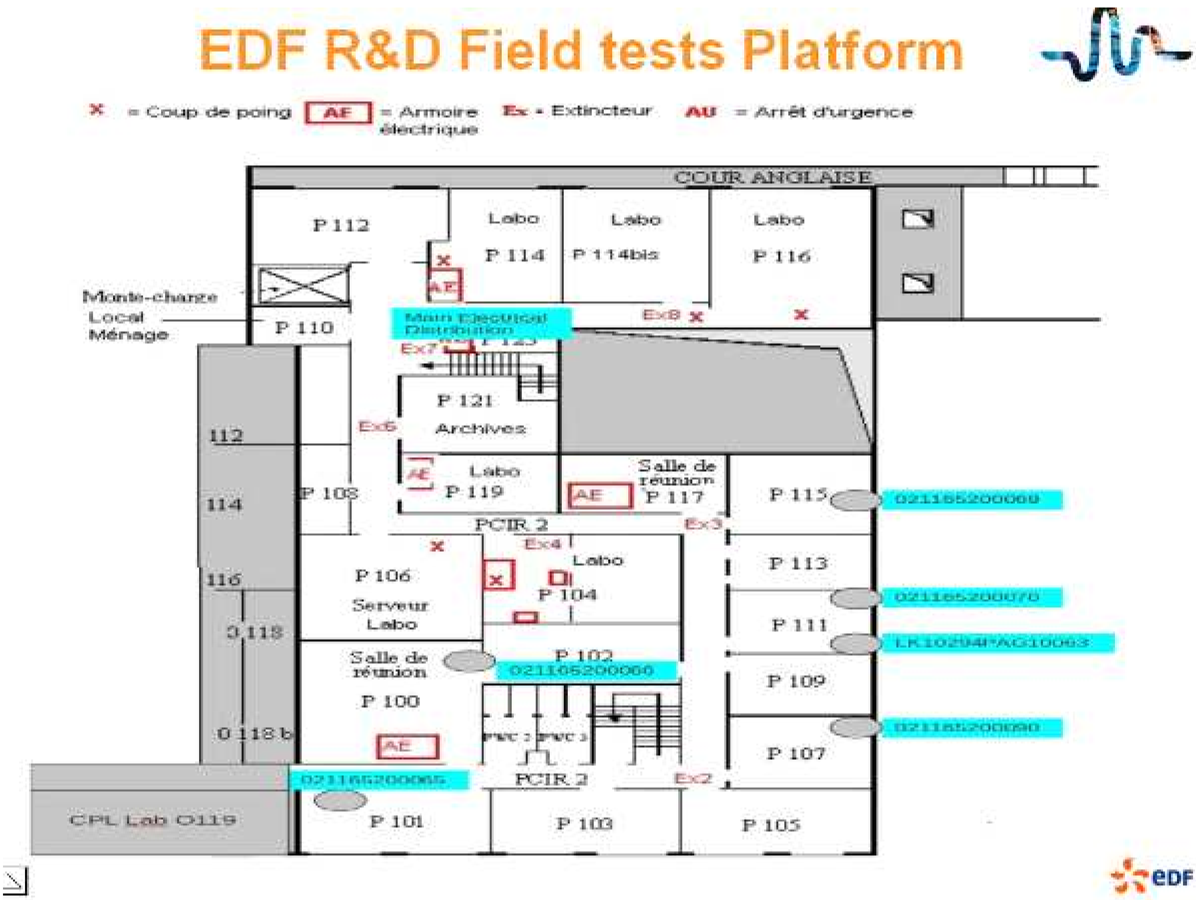


Figure 4-8: EDF R&D PLC G3 field tests electrical scheme

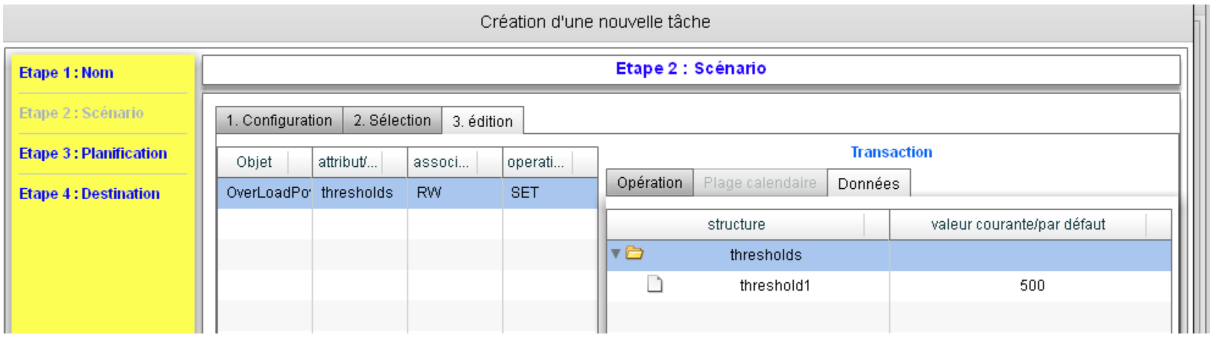
Figure 4-9: Different meter locations with their respective loads

4.3.2 Results of DLMS-G3 field tests.

An overview of all test results is given in Annex F. In the next subsection, the results are described in more detail.

4.3.2.1 Test 4.4.6.1 - Power Control

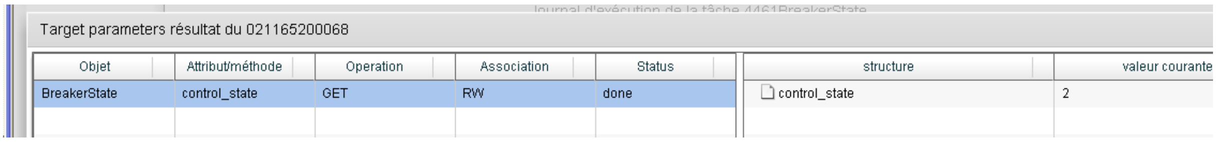
First, the overload power threshold (OverLoadPowerLimitActive object) is set to 500 VA :




```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="235" execId="2011-04-22 09:52:40" date="">
  <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200068" dc="LK10294PAG10214">
    <transaction api="DLMS" id="1" type="SET">
      <request>
        <set>
          <object>OverLoadPowerLimitActive.thresholds</object>
          <association>RW</association>
          <payload>
            <parameter name="thresholds" type="array">
              <parameter name="threshold1" type="unsigned16" value="500"/>
            </parameter>
          </payload>
        </set>
      </request>
      <start>2011-04-22 09:55:49 54</start>
```

```
      <stop>2011-04-22 09:55:52 185</stop>
      <retries>0</retries>
      <status>1</status>
      <result>
        <raw>0</raw>
        <decoded/>
      </result>
    </transaction>
  </target>
</taskExecution>
```

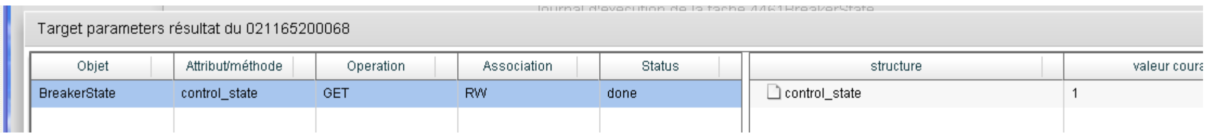

Then a load is applied to the meter (a 1,7 kVA electric heater). After several seconds, power supply is stopped. This can be verified by reading BreakerState control state :


Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante
BreakerState	control_state	GET	RW	done	control_state	2

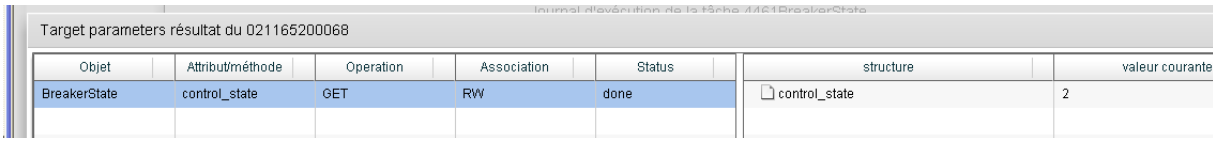
The read value (= 2) means that the breaker is open and can be reconnected manually.

Power supply is re-established by reconnecting the breaker manually. Once manual reconnection is performed, BreakerState control state equals 1 (corresponding to the “connected” state) :


Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur coura
BreakerState	control_state	GET	RW	done	control_state	1

As expected, when the load is applied again, power supply interrupts again, as shown by the following figure (BreakerState control state = 2) :


Target parameters résultat du 021165200068

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante
BreakerState	control_state	GET	RW	done	control_state	2

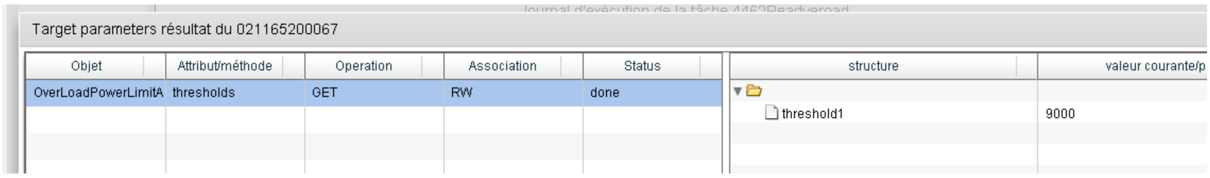
4.3.2.2 Test 4.4.6.2 - Power Control

The contractual power is set to 9 kVA :


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="244" execId="2011-04-22 10:45:12" date="">
  <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200067" dc="LK10294PAG10214">
    <transaction api="DLMS" id="1" type="SET">
      <request>
        <set>
          <object>OverLoadPowerLimitActive.thresholds</object>
          <association>RW</association>
          <payload>
            <parameter name="thresholds" type="array">
```

```
          <parameter name="threshold" type="unsigned16" value="9000"/>
        </parameter>
      </payload>
    </set>
  </request>
  <start>2011-04-22 10:48:23 135</start>
  <stop>2011-04-22 10:48:11 129</stop>
  <retries>0</retries>
  <status>1</status>
  <result>
    <raw>0</raw>
    <decoded/>
  </result>
 </transaction>
</target>
</taskExecution>
```

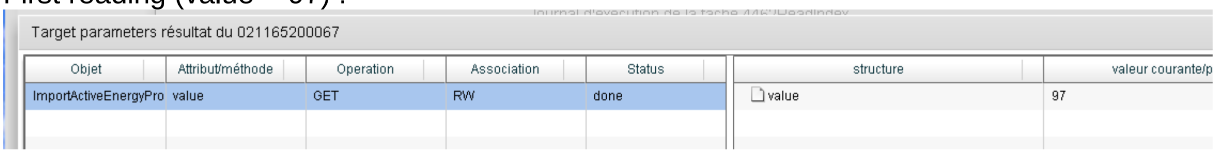

By reading the overload power threshold, successful contractual power setting can be noticed :


Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/p
OverLoadPowerLimitA	thresholds	GET	RW	done		
					threshold1	9000

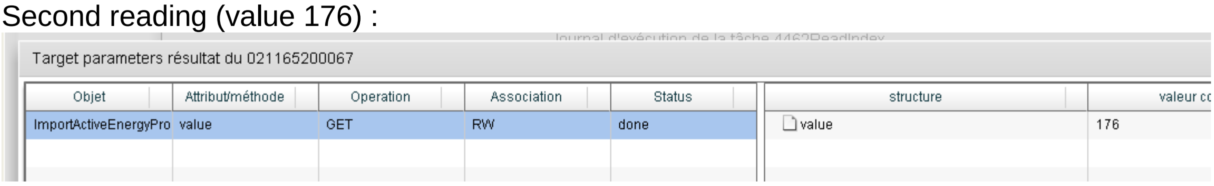
Then, a load is applied to the meter, as shown by the following ImportActiveEnergyProfile reads :

First reading (value = 97) :


Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/p
ImportActiveEnergyPro	value	GET	RW	done	value	97

Second reading (value 176) :


Target parameters résultat du 021165200067

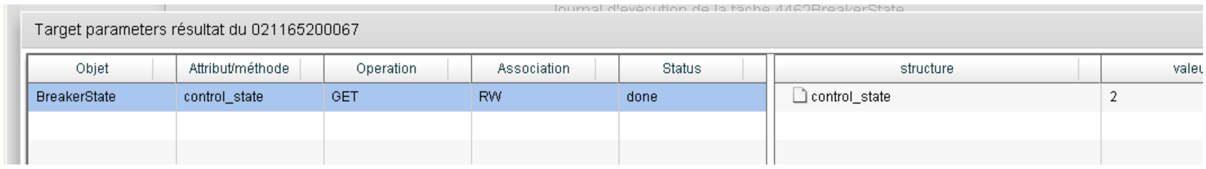
Objet	Attribut/méthode	Operation	Association	Status	structure	valeur co
ImportActiveEnergyPro	value	GET	RW	done	value	176

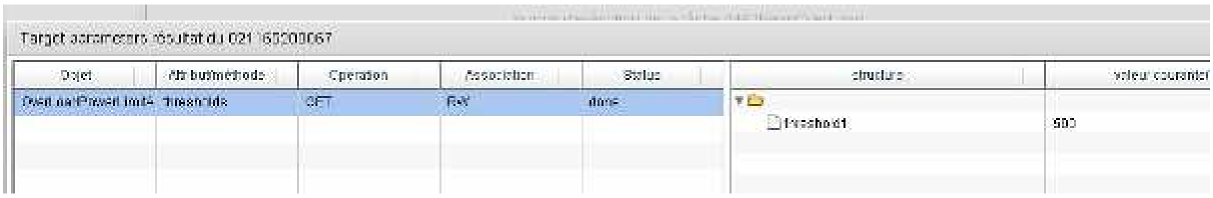
Afterwards, overload power threshold is set down to 500 VA :


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="244" execId="2011-04-22 10:55:20" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="SET">
   <request>
    <set>
     <object>OverLoadPowerLimitActive.thresholds</object>
     <association>RW</association>
     <payload>
      <parameter name="thresholds" type="array">
```

```
          <parameter name="threshold" type="unsigned16" value="500"/>
        </parameter>
      </payload>
    </set>
  </request>
  <start>2011-04-22 10:58:28 115</start>
  <stop>2011-04-22 10:58:31 216</stop>
  <retries>0</retries>
  <status>1</status>
  <result>
    <raw>0</raw>
    <decoded/>
  </result>
 </transaction>
</target>
</taskExecution>
```


Indeed, breaker disconnection is provoked by exceeding overload power limit (breaker ready for manual reconnection) :





Overload power threshold is finally set back to the 9 kVA nominal value :


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="248" execId="2011-04-22 11:05:13" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="SET">
   <request>
    <set>
     <object>OverLoadPowerLimitActive.thresholds</object>
     <association>RW</association>
     <payload>
      <parameter name="thresholds" type="array">
       <parameter name="threshold1" type="unsigned16" value="9000"/>
      </parameter>
     </payload>
    </set>
   </request>
   <start>2011-04-22 11:08:23 465</start>
   <stop>2011-04-22 11:08:26 597</stop>
   <retries>0</retries>
   <status>1</status>
   <result>
```

```
      <raw>0</raw>
      <decoded/>
    </result>
  </transaction>
 </target>
</taskExecution>
```


Breaker is manually reconnected while the previous load (see ImportActiveEnergyProvider value variation below) is still applied to the meter. No automatic disconnection is noticed.

First reading (value = 264) :

Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante
ImportActiveEnergyPro	value	GET	RW	done	value	264

Second reading (value = 314) :

Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur coura
ImportActiveEnergyPro	value	GET	RW	done	value	314

4.3.2.3 Test 4.4.8.1 - Firmware update test

Before the new firmware is uploaded, the E-meter's state is retrieved. As shown below, the firmware version of meter “021165200067” is requested :


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="252" execId="2011-04-22 11:41:46" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="GET">
   <request>
    <get>
     <object>FirmwareVersion.value</object>
     <association>public</association>
    </get>
   </request>
   <start>2011-04-22 11:44:52 213</start>
   <stop>2011-04-22 11:44:55 346</stop>
   <retries>0</retries>
   <status>1</status>
   <result>
```

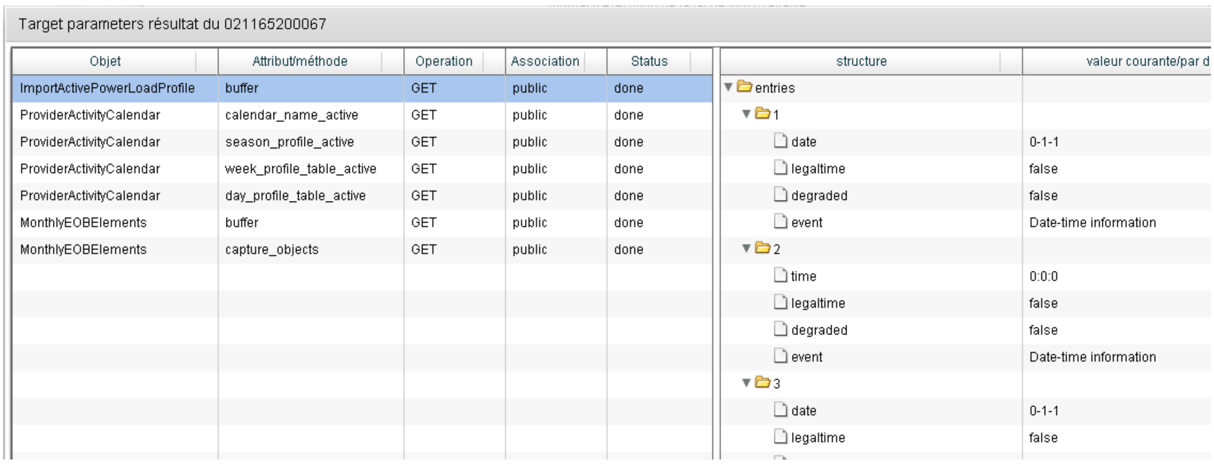

Resulting response :


```
<raw>0104020509064d6574726f0016001100111206eb287138020509064170706c690016011100113706b3f1871402050906426f6f74000016011
10311000694d595920205090600000000000001600110011000600000000</raw>
    <decoded>
     <array>
      <struct size="5">
       <octetString value="77;101;116;114;111;0;" name="firmware_id" edition="asciiText"/>
       <enumerated value="0" name="firmware_type"/>
       <unsigned8 value="0" name="major_version"/>
       <unsigned8 value="18" name="minor_version"/>
```

```
        <unsigned32 value="3945296184" name="checksum"/>
       </struct>
       <struct size="5">
        <octetString value="65;112;112;108;105;0;" name="firmware_id" edition="asciiText"/>
        <enumerated value="1" name="firmware_type"/>
        <unsigned8 value="0" name="major_version"/>
        <unsigned8 value="55" name="minor_version"/>
        <unsigned32 value="3018950420" name="checksum"/>
       </struct>
       <struct size="5">
        <octetString value="66;111;111;116;0;0;" name="firmware_id" edition="asciiText"/>
        <enumerated value="1" name="firmware_type"/>
        <unsigned8 value="3" name="major_version"/>
        <unsigned8 value="0" name="minor_version"/>
        <unsigned32 value="2497025426" name="checksum"/>
       </struct>
       <struct size="5">
        <octetString value="0;0;0;0;0;0;" name="firmware_id" edition="asciiText"/>
        <enumerated value="0" name="firmware_type"/>
        <unsigned8 value="0" name="major_version"/>
        <unsigned8 value="0" name="minor_version"/>
        <unsigned32 value="0" name="checksum"/>
       </struct>
      </array>
     </decoded>
    </result>
   </transaction>
  </target>
</taskExecution>
```


The meter's current firmware version can be read : 0.55 (see red lines of the XML extract above).

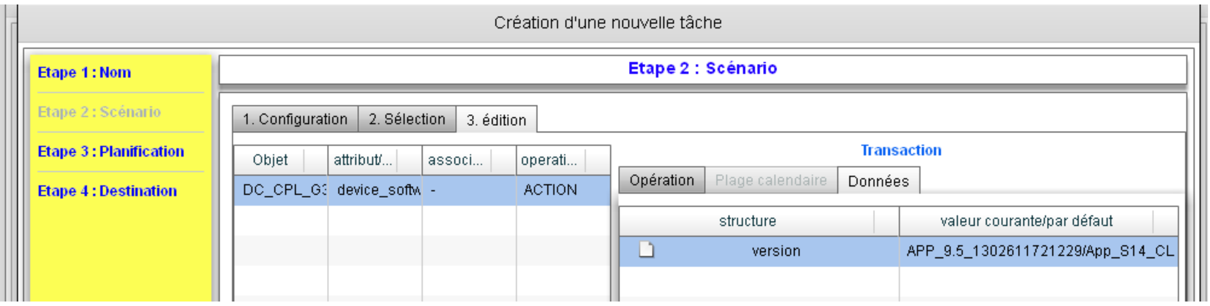
Load profile information is also retrieved as shown in the following figure :


Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status
ImportActivePowerLoadProfile	buffer	GET	public	done
ProviderActivityCalendar	calendar_name_active	GET	public	done
ProviderActivityCalendar	season_profile_active	GET	public	done
ProviderActivityCalendar	week_profile_table_active	GET	public	done
ProviderActivityCalendar	day_profile_table_active	GET	public	done
MonthlyEOBElements	buffer	GET	public	done
MonthlyEOBElements	capture_objects	GET	public	done

structure	valeur courante/par d
entries	
1	
date	0-1-1
legaltime	false
degraded	false
event	Date-time information
2	
time	0:0:0
legaltime	false
degraded	false
event	Date-time information
3	
date	0-1-1
legaltime	false

Then Firmware update is performed :



Data concentrator traces (transfer of 500 byte APDUs) :

```
2011-04-22 09:52:52.8145 kern.info: ##################################################
2011-04-22 09:52:52.8153 kern.warn:
2011-04-22 09:52:52.7005 kern.info: drivers/plc/plc.c:net_frame_tx: plc0 send data
2011-04-22 09:52:52.7008 kern.info: drivers/plc/plc.c:log_frame_tx: REQ:ADPM_DATA
2011-04-22 09:52:52.7009 kern.info: drivers/plc/plc.c:sl_encaps: count= 610
2011-04-22 09:52:52.7010 kern.warn: c0 FF FF FF FF FF FF FF FF FF FF FF FF 60
2011-04-22 09:52:52.7010 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF 10
2011-04-22 09:52:52.7011 kern.warn: 12 FF FF FF FF FF FF FF FF FF FF FF FF 10
2011-04-22 09:52:52.7012 kern.warn: 12 FF FF FF FF FF FF FF FF FF FF FF FF ed
2011-04-22 09:52:52.7013 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF 2c
2011-04-22 09:52:52.7014 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7015 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7025 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7027 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7028 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7028 kern.warn: 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7029 kern.warn: 7d 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7030 kern.warn: 07 00 FF FF FF FF FF FF FF FF FF FF FF FF f4
2011-04-22 09:52:52.7031 kern.warn: 21 00 FF FF FF FF FF FF FF FF FF FF FF FF f4e3
2011-04-22 09:52:52.7032 kern.warn: 6f 00 FF FF FF FF FF FF FF FF FF FF FF FF f42a
………………………………………………………………………………………………………
2011-04-22 09:52:52.7033 kern.warn: b2 3d e1 c1 78 d9 c5 90 b7 e1 2e b9 8f 45 02 b7
2011-04-22 09:52:52.7034 kern.warn: 82 2f 7c 36 9a b6 93 4c 63 c4 3b ff 03 02 0f 62

2011-04-22 09:52:54.1680 kern.info: ##################################################
2011-04-22 09:52:54.4566 kern.info: drivers/plc/plc.c:slip_receive_buf: tty name:ttymxc2 count= 15
```

NOTE: Due to implementation confidentiality issues, the firmware update binary file has been codified in these traces

Afterwards, the image transfer process is verified :

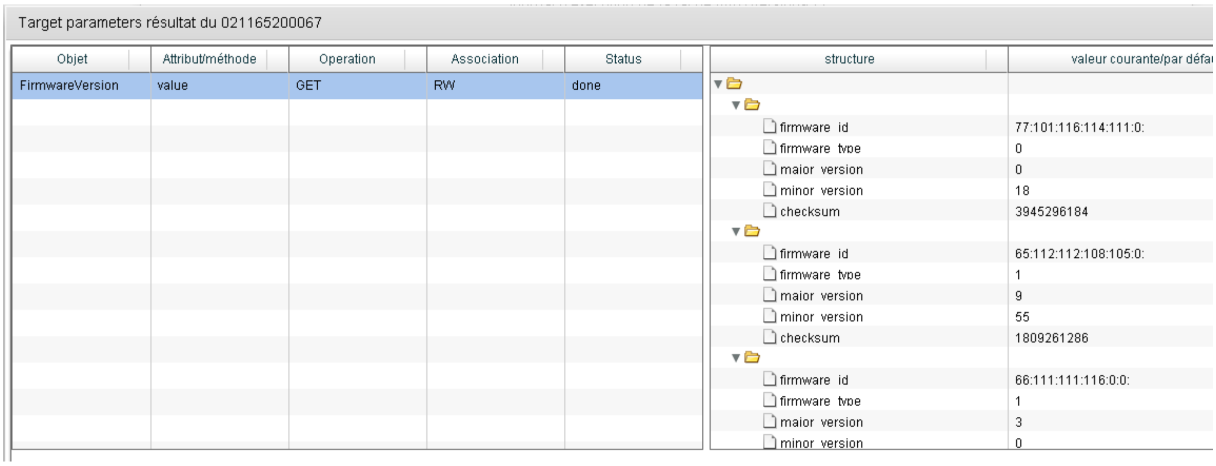

```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="256" execId="2011-04-22 13:54:03" date="">
  <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200067" dc="LK10294PAG10214">
    <transaction api="DLMS" id="1" type="ACTION">
      <request>
        <action>
          <object>ImageTransfer.image_verify</object>
          <association>public</association>
          <payload>
            <parameter name="image_verify" type="integer8" value="0"/>
```

```
        </payload>
      </action>
    </request>
    <start>2011-04-22 13:57:11 504</start>
    <stop>2011-04-22 13:57:17 669</stop>
    <retries>0</retries>
    <status>1</status>
    <result>
      <raw>0</raw>
      <decoded/>
    </result>
  </transaction>
 </target>
</taskExecution>
```


Firmware activation is performed :

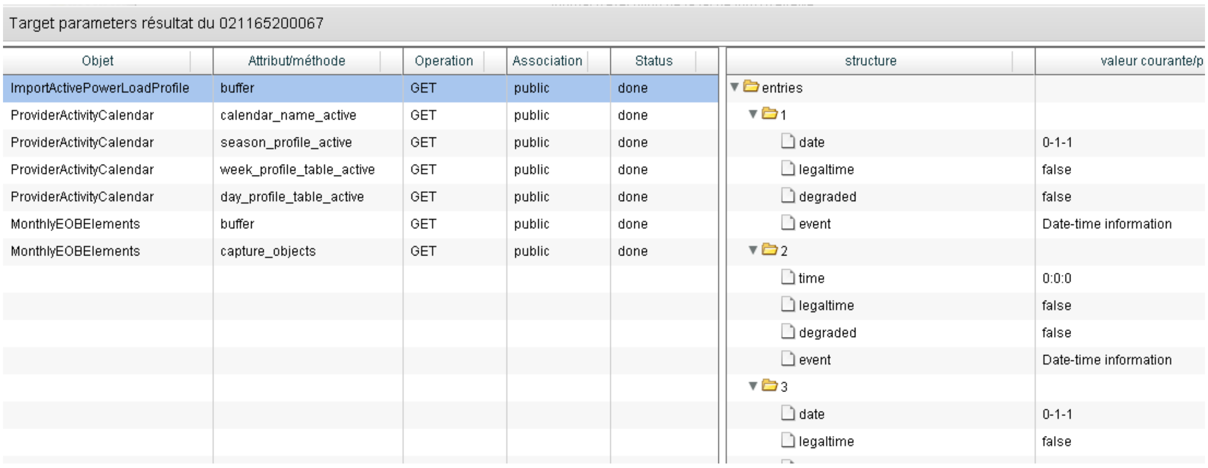

```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="257" execId="2011-04-22 13:56:00" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="ACTION">
    <request>
      <action>
        <object>ImageTransfer.image_activate</object>
        <association>RW</association>
        <payload>
          <parameter name="image_activate" type="integer8" value="0"/>
        </payload>
      </action>
    </request>
    <start>2011-04-22 13:58:55 565</start>
    <stop>2011-04-22 13:58:58 648</stop>
    <retries>0</retries>
    <status>1</status>
    <result>
      <raw>0</raw>
      <decoded/>
    </result>
  </transaction>
 </target>
</taskExecution>
```


And finally new firmware version is checked (9.55 FW version replaces 0.55 FW version) :



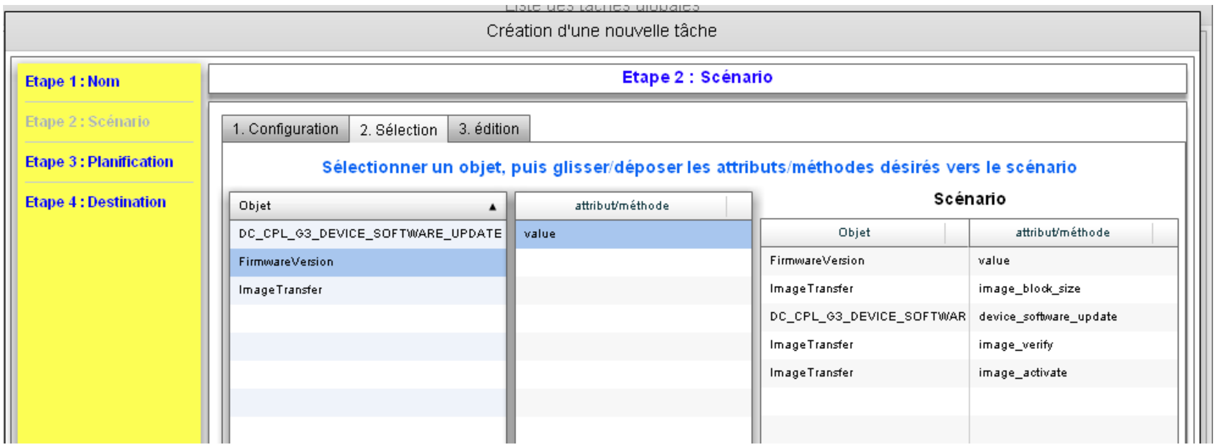
<raw>0104020509064d6574726f0016001100111206eb28713802050906417070 6c690016011109113 7066bd722e602050906426f6f7400001601110311000694d5959202050906000000000000160011001100060000000 0</raw>
<decoded>
<array>
<struct size="5">
<octetString value="77;101;116;114;111;0;" name="firmware_id" edition="asciiText"/>
<enumerated value="0" name="firmware_type"/>
<unsinged8 value="0" name="major_version"/>
<unsinged8 value="18" name="minor_version"/>
<unsigned32 value="3945296184" name="checksum"/>
</struct>
<struct size="5">
<octetString value="65;112;112;108;105;0;" name="firmware_id" edition="asciiText"/>
<enumerated value="1" name="firmware_type"/>
<unsinged8 value="9" name="major_version"/>
<unsinged8 value="55" name="minor_version"/>
<unsigned32 value="1809261286" name="checksum"/>
</struct>
<struct size="5">
<octetString value="66;111;111;116;0;0;" name="firmware_id" edition="asciiText"/>
<enumerated value="1" name="firmware_type"/>
<unsinged8 value="3" name="major_version"/>
<unsinged8 value="0" name="minor_version"/>
<unsigned32 value="2497025426" name="checksum"/>
</struct>
<struct size="5">
<octetString value="0;0;0;0;0;0;" name="firmware_id" edition="asciiText"/>
<enumerated value="0" name="firmware_type"/>
<unsinged8 value="0" name="major_version"/>
<unsinged8 value="0" name="minor_version"/>
<unsigned32 value="0" name="checksum"/>
</struct>
</array>
</decoded>
</result>
</transaction>
</target>
</taskExecution>

At the end, after successful firmware upload, notice that load profile information has been kept by the meter :



4.3.2.4 Test 4.4.8.2 - Firmware update test

Immediate firmware upload (automatic upload, image transfer verification and firmware version application) is done through the following test scenario :



Firmware versions are shown in the following traces (before and after firmware upload) :

Before firmware upload (version 0.55) :


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="260" execId="2011-04-22 14:27:54" date="">
 <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
 <target id="021165200067" dc="LK10294PAG10214">
  <transaction api="DLMS" id="1" type="GET">
   <request>
    <get>
     <object>FirmwareVersion.value</object>
     <association>public</association>
    </get>
   </request>
```

```
    &lt;start&gt;2011-04-22 14:31:00 179&lt;/start&gt;
    &lt;stop&gt;2011-04-22 14:31:03 286&lt;/stop&gt;
    &lt;retries&gt;0&lt;/retries&gt;
    &lt;status&gt;1&lt;/status&gt;
    &lt;result&gt;

&lt;raw&gt;0104020509064d6574726f0016001100111206eb287138020509064170706c6900160111091137066bd722e602050906426f6f7400001601
110311000694d595920205090600000000000016001100110006000000000&lt;/raw&gt;
     &lt;decoded&gt;
      &lt;array&gt;
       &lt;struct size="5"&gt;
        &lt;octetString value="77;101;116;114;111;0;" name="firmware_id" edition="asciiText"/&gt;
        &lt;enumerated value="0" name="firmware_type"/&gt;
        &lt;unsinged8 value="0" name="major_version"/&gt;
        &lt;unsinged8 value="18" name="minor_version"/&gt;
        &lt;unsigned32 value="3945296184" name="checksum"/&gt;
       &lt;/struct&gt;
       &lt;struct size="5"&gt;
        &lt;octetString value="65;112;112;108;105;0;" name="firmware_id" edition="asciiText"/&gt;
        &lt;enumerated value="1" name="firmware_type"/&gt;
        &lt;unsinged8 value="0" name="major_version"/&gt;
        &lt;unsinged8 value="55" name="minor_version"/&gt;
        &lt;unsigned32 value="1809261286" name="checksum"/&gt;
       &lt;/struct&gt;
       &lt;struct size="5"&gt;
        &lt;octetString value="66;111;111;116;0;0;" name="firmware_id" edition="asciiText"/&gt;
        &lt;enumerated value="1" name="firmware_type"/&gt;
        &lt;unsinged8 value="3" name="major_version"/&gt;
        &lt;unsinged8 value="0" name="minor_version"/&gt;
        &lt;unsigned32 value="2497025426" name="checksum"/&gt;
       &lt;/struct&gt;
       &lt;struct size="5"&gt;
        &lt;octetString value="0;0;0;0;0;0;" name="firmware_id" edition="asciiText"/&gt;
        &lt;enumerated value="0" name="firmware_type"/&gt;
        &lt;unsinged8 value="0" name="major_version"/&gt;
        &lt;unsinged8 value="0" name="minor_version"/&gt;
        &lt;unsigned32 value="0" name="checksum"/&gt;
       &lt;/struct&gt;
      &lt;/array&gt;
     &lt;/decoded&gt;
    &lt;/result&gt;
   &lt;/transaction&gt;
  &lt;/target&gt;
&lt;/taskExecution&gt;
```

After firmware upload (version 9.55) :

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;taskExecution taskId="272" execId="2011-04-22 15:53:47" date=""&gt;
  &lt;taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/&gt;
  &lt;target id="021165200067" dc="LK10294PAG10214"&gt;
   &lt;transaction api="DLMS" id="1" type="GET"&gt;
    &lt;request&gt;
     &lt;get&gt;
      &lt;object&gt;FirmwareVersion.value&lt;/object&gt;
      &lt;association&gt;public&lt;/association&gt;
     &lt;/get&gt;
    &lt;/request&gt;
```

```
    <start>2011-04-22 15:56:54 596</start>
    <stop>2011-04-22 15:56:57 728</stop>
    <retries>0</retries>
    <status>1</status>
    <result>

<raw>0104020509064d6574726f0016001100111206eb28713802050906417070 6c6900160111001137 06b3f1871402050906426f6f74000016011
10311000694d59592020509060000000000000016001100110006000 00000</raw>
      <decoded>
       <array>
        <struct size="5">
         <octetString value="77;101;116;114;111;0;" name="firmware_id" edition="asciiText"/>
         <enumerated value="0" name="firmware_type"/>
         <unsinged8 value="0" name="major_version"/>
         <unsinged8 value="18" name="minor_version"/>
         <unsigned32 value="3945296184" name="checksum"/>
        </struct>
        <struct size="5">
         <octetString value="65;112;112;108;105;0;" name="firmware_id" edition="asciiText"/>
         <enumerated value="1" name="firmware_type"/>
         <unsinged8 value="9" name="major_version"/>
         <unsinged8 value="55" name="minor_version"/>
         <unsigned32 value="3018950420" name="checksum"/>
        </struct>
        <struct size="5">
         <octetString value="66;111;111;116;0;0;" name="firmware_id" edition="asciiText"/>
         <enumerated value="1" name="firmware_type"/>
         <unsinged8 value="3" name="major_version"/>
         <unsinged8 value="0" name="minor_version"/>
         <unsigned32 value="2497025426" name="checksum"/>
        </struct>
        <struct size="5">
         <octetString value="0;0;0;0;0;0;" name="firmware_id" edition="asciiText"/>
         <enumerated value="0" name="firmware_type"/>
         <unsinged8 value="0" name="major_version"/>
         <unsinged8 value="0" name="minor_version"/>
         <unsigned32 value="0" name="checksum"/>
        </struct>
       </array>
      </decoded>
    </result>
   </transaction>
  </target>
</taskExecution>
```


As in test 4.4.8.1, the state of the meter has not been modified by the firmware upload procedure.

4.3.2.5 Test 4.4.8.3 - Firmware update test

As shown in the figure below and the associated traces, the corrupted uploaded firmware version is discarded by the meter :

Target parameters résultat du 021165200067

Objet	Attribut/méthode	Operation	Association	Status
FirmwareVersion	value	GET	public	done
ImageTransfer	image_block_size	SET	RW	done
DC_CPL_G3_DEVICE	device_software_upda	ACTION		done
ImageTransfer	image_verify	ACTION	public	failed
ImageTransfer	image_activate	ACTION	public	failed

structure	valeur courante/par défaut

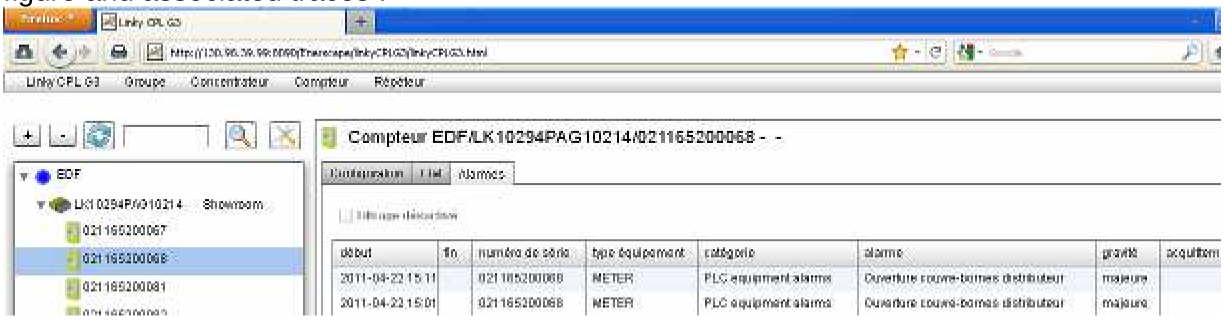

```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="263" execId="2011-04-22 15:03:22" date="">
  <taskExecutionStatus status="FAILED" ko="1(100%)" ok="0(0%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200067" dc="LK10294PAG10214">
    <transaction api="DLMS" id="1" type="GET">
      <request>
        <get>
          <object>FirmwareVersion.value</object>
          <association>public</association>
        </get>
      </request>
      <start>2011-04-22 15:06:31 464</start>
      <stop>2011-04-22 15:06:34 585</stop>
      <retries>0</retries>
      <status>1</status>
      <result>

<raw>0104020509064d6574726f0016001100111206eb287138020509064170706c6900160111091137066bd722e602050906426f6f7400001601
110311000694d5959202050906000000000000016001100110006000000000</raw>
        <decoded>
          <array>
            <struct size="5">
              <octetString value="77;101;116;114;111;0;" name="firmware_id" edition="asciiText"/>
              <enumerated value="0" name="firmware_type"/>
              <unsinged8 value="0" name="major_version"/>
              <unsinged8 value="18" name="minor_version"/>
              <unsigned32 value="3945296184" name="checksum"/>
            </struct>
            <struct size="5">
              <octetString value="65;112;112;108;105;0;" name="firmware_id" edition="asciiText"/>
              <enumerated value="1" name="firmware_type"/>
              <unsinged8 value="9" name="major_version"/>
              <unsinged8 value="55" name="minor_version"/>
              <unsigned32 value="1809261286" name="checksum"/>
            </struct>
            <struct size="5">
              <octetString value="66;111;111;116;0;0;" name="firmware_id" edition="asciiText"/>
              <enumerated value="1" name="firmware_type"/>
              <unsinged8 value="3" name="major_version"/>
              <unsinged8 value="0" name="minor_version"/>
              <unsigned32 value="2497025426" name="checksum"/>
            </struct>
```

```
      <struct size="5">
        <octetString value="0;0;0;0;0;0;" name="firmware_id" edition="asciiText"/>
        <enumerated value="0" name="firmware_type"/>
        <unsinged8 value="0" name="major_version"/>
        <unsinged8 value="0" name="minor_version"/>
        <unsigned32 value="0" name="checksum"/>
      </struct>
    </array>
   </decoded>
  </result>
</transaction>
<transaction api="DLMS" id="2" type="SET">
  <request>
   <set>
    <object>ImageTransfer.image_block_size</object>
    <association>RW</association>
    <payload>
      <parameter name="image_block_size" type="unsigned32" value="500"/>
    </payload>
   </set>
  </request>
  <start>2011-04-22 15:06:34 636</start>
  <stop>2011-04-22 15:06:37 778</stop>
  <retries>0</retries>
  <status>1</status>
  <result>
   <raw>0</raw>
   <decoded/>
  </result>
</transaction>
<transaction api="DC" id="3" type="ACTION">
  <request>
   <action>
    <object>DC_CPL_G3_DEVICE_SOFTWARE_UPDATE.device_software_update</object>
    <association/>
    <payload>
      <parameter name="version" type="enumerated" value="APP_0.5_1302611731100/App_S14_CLI_Encode.dat"/>
    </payload>
   </action>
  </request>
  <start>2011-04-22 15:06:37 808</start>
  <stop>2011-04-22 15:15:57 803</stop>
  <retries>0</retries>
  <status>1</status>
  <result>
   <raw>0</raw>
   <decoded/>
  </result>
</transaction>
<transaction api="DLMS" id="4" type="ACTION">
  <request>
   <action>
    <object>ImageTransfer.image_verify</object>
    <association>public</association>
   </action>
  </request>
  <start>2011-04-22 15:15:57 977</start>
  <stop>2011-04-22 15:15:57 977</stop>
  <retries>0</retries>
```

```
    <status>0</status>
    <result>
      <raw>fail: unsupported transaction type</raw>
      <decoded/>
    </result>
  </transaction>
  <transaction api="DLMS" id="5" type="ACTION">
    <request>
      <action>
        <object>ImageTransfer.image_activate</object>
        <association>public</association>
      </action>
    </request>
```


4.3.2.6 Test 4.4.11.1 - Fraud detection

After the meter's cover has been opened, an alarm is generated, as shown in the following figure and associated traces :




```
[04/22/2011 13:11:35 020]  Decode content.021165200068
[04/22/2011 13:11:35 022]  MeterManager.createAlarmFromDevice begin.
[04/22/2011 13:11:35 022]  MeterManager.createAlarmFromDevice serial number is :021165200068
[04/22/2011 13:11:35 022]  MeterManager.createAlarmFromDevice word :2097152
```

4.3.2.7 Test 4.4.13.1 - Load profile management

As required for this test, the integration time of the load profile equals 30 min (1800 s) :

Target parameters résultat du 021165200082

Objet	Attribut/méthode	Operation	Association	Status	structure	valeur courante/par défaut
ImportActivePowerLoa	capture_period	GET	public	done	capture_period	1800
ImportActivePowerLoa	buffer	GET	public	done		
ImportActivePowerLoa	capture_objects	GET	public	done		

Initially, the following load profile is available (495 entries are available, thus only the last 5 one are shown) :


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="278" execId="2011-04-22 16:44:00" date="">
  <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200082" dc="LK10294PAG10214">
    <transaction api="DLMS" id="1" type="GET">
      <request>
        <get>
```

```
        <object>ImportActivePowerLoadProfile.capture_period</object>
        <association>public</association>
      </get>
    </request>
    <start>2011-04-22 16:47:04 877</start>
    <stop>2011-04-22 16:47:08 16</stop>
    <retries>0</retries>
    <status>1</status>
    <result>
      <raw>0600000708</raw>
      <decoded>
        <unsigned32 value="1800" name="capture_period"/>
      </decoded>
    </result>
  </transaction>
  <transaction api="DLMS" id="2" type="GET">
    <request>
      <get>
        <object>ImportActivePowerLoadProfile.buffer</object>
        <association>public</association>
        <parameters>
          <parameter name="fromDate" value="1970-01-01T00:00:00Z"/>
          <parameter name="toDate" value="1970-01-01T00:00:00Z"/>
        </parameters>
      </get>
    </request>
    <start>2011-04-22 16:47:08 64</start>
    <stop>2011-04-22 16:47:14 384</stop>
    <retries>0</retries>
    <status>1</status>
    <result>

<raw>130201068207bc80000021c000000082000021c20000004080000083000021c300225584200021c42023e840a000008520168cc520f4d240
…20fc0440a0000083201696c32109b182201696c220fc0440a00000</raw>
      <decoded>
        <carray name="entries" edition="LoadProfile">
          <struct name="1">
            <visibleString name="date" value="0-1-1"/>
            <boolean name="legaltime" value="winter"/>
            <boolean name="degraded" value="not degraded"/>
            <visibleString name="event" value="Date-time information"/>
          </struct>
          <struct name="2">
            <visibleString name="time" value="0:0:0"/>
            <boolean name="legaltime" value="winter"/>
            <boolean name="degraded" value="not degraded"/>
            <visibleString name="event" value="Date-time information"/>
          </struct>
          <struct name="3">
            <visibleString name="date" value="0-1-1"/>
            <boolean name="legaltime" value="winter"/>
            <boolean name="degraded" value="not degraded"/>
            <visibleString name="event" value="Event of power outage"/>
          </struct>
          <struct name="4">
            <visibleString name="time" value="0:0:0"/>
            <boolean name="legaltime" value="winter"/>
            <boolean name="degraded" value="not degraded"/>
            <visibleString name="event" value="Event of power outage"/>
```

```
      </struct>
      <struct name="5">
        <boolean name="legaltime" value="winter"/>
        <boolean name="degraded" value="not degraded"/>
        <unsigned8 name="step" value="30min"/>
        <unsigned32 name="power" value="0"/>
      </struct>

        ...

      <struct name="490">
        <boolean name="legaltime" value="summer"/>
        <boolean name="degraded" value="not degraded"/>
        <unsigned8 name="step" value="30min"/>
        <unsigned32 name="power" value="0"/>
      </struct>
      <struct name="491">
        <visibleString name="date" value="11-4-22"/>
        <boolean name="legaltime" value="summer"/>
        <boolean name="degraded" value="not degraded"/>
        <visibleString name="event" value="Event of power recovery"/>
      </struct>
      <struct name="492">
        <visibleString name="time" value="16:38:49"/>
        <boolean name="legaltime" value="summer"/>
        <boolean name="degraded" value="not degraded"/>
        <visibleString name="event" value="Event of power recovery"/>
      </struct>
      <struct name="493">
        <visibleString name="date" value="11-4-22"/>
        <boolean name="legaltime" value="summer"/>
        <boolean name="degraded" value="not degraded"/>
        <visibleString name="event" value="Event of power outage"/>
      </struct>
      <struct name="494">
        <visibleString name="time" value="15:48:4"/>
        <boolean name="legaltime" value="summer"/>
        <boolean name="degraded" value="not degraded"/>
        <visibleString name="event" value="Event of power outage"/>
      </struct>
      <struct name="495">
        <boolean name="legaltime" value="summer"/>
        <boolean name="degraded" value="not degraded"/>
        <unsigned8 name="step" value="30min"/>
        <unsigned32 name="power" value="0"/>
      </struct>
    </carray>
   </decoded>
  </result>
</transaction>
<transaction api="DLMS" id="3" type="GET">
  <request>
   <get>
    <object>ImportActivePowerLoadProfile.capture_objects</object>
    <association>public</association>
   </get>
  </request>
  <start>2011-04-22 16:47:14 444</start>
  <stop>2011-04-22 16:47:17 575</stop>
```

```
      <retries>0</retries>
      <status>1</status>
      <result>
        <raw>0101020412000109060100604001ff0f02120000</raw>
        <decoded>
          <array>
            <struct size="4">
              <unsinged16 value="1" name="Class_id"/>
              <octetString value="1;0;96;64;1;255;" name="Obis_code"/>
              <signed8 value="2" name="Attribute_index"/>
              <unsinged16 value="0" name="Data_index"/>
            </struct>
          </array>
        </decoded>
      </result>
    </transaction>
  </target>
</taskExecution>
```


Approximately 2 hours and 30 minutes later load profile is read again. The following result (500 entries) is available :


```
<?xml version="1.0" encoding="UTF-8"?>
<taskExecution taskId="282" execId="2011-04-22 17:36:11" date="">
  <taskExecutionStatus status="DONE" ko="0(0%)" ok="1(100%)" notAvailable="0(0%)" completion="100%"/>
  <target id="021165200082" dc="LK10294PAG10214">
    <transaction api="DLMS" id="1" type="GET">
      <request>
        <get>
          <object>ImportActivePowerLoadProfile.capture_period</object>
          <association>public</association>
        </get>
      </request>
      <start>2011-04-22 17:39:20 706</start>
      <stop>2011-04-22 17:39:23 816</stop>
      <retries>0</retries>
      <status>1</status>
      <result>
        <raw>0600000708</raw>
        <decoded>
          <unsigned32 value="1800" name="capture_period"/>
        </decoded>
      </result>
    </transaction>
    <transaction api="DLMS" id="2" type="GET">
      <request>
        <get>
          <object>ImportActivePowerLoadProfile.buffer</object>
          <association>public</association>
          <parameters>
            <parameter name="fromDate" value="1970-01-01T00:00:00Z"/>
            <parameter name="toDate" value="1970-01-01T00:00:00Z"/>
          </parameters>
        </get>
      </request>
      <start>2011-04-22 17:39:23 864</start>
      <stop>2011-04-22 17:39:29 615</stop>
      <retries>0</retries>
```

```
<status>1</status>
<result>

<raw>130201068207d880000021c000000082000021c200000040800000083000021c300225584200021c42023e…c32109b182201696c220fc044
0a0000083201696c3210a1c84201696c4210d9840a002e885201698c5210cc0</raw>
  <decoded>
   <carray name="entries" edition="LoadProfile">
    <struct name="1">
     <visibleString name="date" value="0-1-1"/>
     <boolean name="legaltime" value="winter"/>
     <boolean name="degraded" value="not degraded"/>
     <visibleString name="event" value="Date-time information"/>
    </struct>
...
    <struct name="494">
     <visibleString name="time" value="15:48:4"/>
     <boolean name="legaltime" value="summer"/>
     <boolean name="degraded" value="not degraded"/>
     <visibleString name="event" value="Event of power outage"/>
    </struct>
    <struct name="495">
     <boolean name="legaltime" value="summer"/>
     <boolean name="degraded" value="not degraded"/>
     <unsigned8 name="step" value="30min"/>
     <unsigned32 name="power" value="0"/>
    </struct>
    <struct name="496">
     <visibleString name="date" value="11-4-22"/>
     <boolean name="legaltime" value="summer"/>
     <boolean name="degraded" value="not degraded"/>
     <visibleString name="event" value="Event of power recovery"/>
    </struct>
    <struct name="497">
     <visibleString name="time" value="16:40:28"/>
     <boolean name="legaltime" value="summer"/>
     <boolean name="degraded" value="not degraded"/>
     <visibleString name="event" value="Event of power recovery"/>
    </struct>
    <struct name="498">
     <visibleString name="date" value="11-4-22"/>
     <boolean name="legaltime" value="summer"/>
     <boolean name="degraded" value="not degraded"/>
     <visibleString name="event" value="Event of old date-time"/>
    </struct>
    <struct name="499">
     <visibleString name="time" value="16:54:24"/>
     <boolean name="legaltime" value="summer"/>
     <boolean name="degraded" value="not degraded"/>
     <visibleString name="event" value="Event of old date-time"/>
    </struct>
    <struct name="500">
     <boolean name="legaltime" value="summer"/>
     <boolean name="degraded" value="not degraded"/>
     <unsigned8 name="step" value="30min"/>
     <unsigned32 name="power" value="744"/>
    </struct>
   </carray>
  </decoded>
</result>
```

```
  </transaction>
  <transaction api="DLMS" id="3" type="GET">
    <request>
      <get>
        <object>ImportActivePowerLoadProfile.capture_objects</object>
        <association>public</association>
      </get>
    </request>
    <start>2011-04-22 17:39:29 674</start>
    <stop>2011-04-22 17:39:32 766</stop>
    <retries>0</retries>
    <status>1</status>
    <result>
      <raw>0101020412000109060100604001ff0f02120000</raw>
      <decoded>
        <array>
          <struct size="4">
            <unsinged16 value="1" name="Class_id"/>
            <octetString value="1;0;96;64;1;255;" name="Obis_code"/>
            <signed8 value="2" name="Attribute_index"/>
            <unsinged16 value="0" name="Data_index"/>
          </struct>
        </array>
      </decoded>
    </result>
  </transaction>
 </target>
</taskExecution>
```


4.4 Conclusion

The performed PLC G3 tests have proven that the current prototypes cover all the basic functionalities required by the OPEN METER project.

The tests have been performed both in lab and field, providing a wide range of use cases and different configuration/loads.

The DLMS tests enhanced the data exchange between the DC and the meters, including the firmware update mechanism.

Last but not least, highlight the fact that to accomplish the billing tests, the information system has to interface properly with the operators legacy system.

5 CONCLUSIONS

The goal of the OPEN meter project is to establish a set of International open and interoperable standards for the smart meter infrastructure which can be applied in smart meter rollout projects in Europe.

The objectives of workpackage 4 of OPEN meter are:

- To be able to test prototypes of the selected standards/technologies and by doing that,
- To get insight in the compliance, interoperability and maturity of the selected standards/technologies

In task 4.1 test procedures and test cases have been defined in order to test standards/technologies defined in WP3. The deliverable, D4.1, was a set of test procedures and test cases for compliance, functional and interoperability tests.

In task 4.2 manufacturers, utilities and test laboratories prepared prototypes (for some of the field tests end-products) and test facilities. The deliverables (D4.2) were physical prototypes and test facilities. These deliverables have been described in a report (D4.3).

In task 4.3 and 4.4 the prototypes and test facilities have been used in laboratory and field environment (In some of the field tests in which installed meters were involved end-products were used instead of prototypes). Deliverable D4.4 describes the laboratory- and field tests performed and its results, conclusions and recommendations as written in the previous chapters. In these chapters recommendations are provided for further improvement of those standards/technologies.

Based on the results we can conclude that the selected standards/technologies tested in OPEN Meter cover the requirements defined by the OPEN Meter project and thus METERS AND MORE, SFSK, PRIME and PLC G3 will be prepared within WP5 as proposal for International standardization.

6 Copyright

ANNEX A – OVERVIEW OF METERS AND MORE LAB TEST RESULTS (RSE)

Test case	Test Name	Description	Test type	Requirement	Result (PASSED/failed/ not tested)	Meter Manufacturer
5.4.1.1	**OM-SR1 Meter Registration**	**Goal**: - Check that the communication channel is automatically available, immediately after the physical installation of the meter **Preconditions**: - The meter was installed electrically	**Functional**	**OM-FR170** **OM-FR80** **OM-FR139**	**PASSED**	KAIFA
5.4.1.2	**OM-SR1 Meter Registration**	**Goal**: - The meter shall be able to respond to a communication initiated by the central system **Preconditions**: - None	**Functional**	**OM-FR143**	**PASSED**	KAIFA

5.4.1.3	**OM-SR1 Meter Registration**	**Goal**: - Check the functionality of clock and calendar of the equipment. Set a new contract with specific power and just one tariff. **Preconditions**: - None	**Functional**	**OM-FR77** **OM-FR78** **OM-FR67** **OM-FR68** **OM-FR69** **OM-FR72** **OM-FR73** **OM-FR79**	**PASSED**	KAIFA
5.4.1.4	**OM-SR1 Meter Registration**	**Goal**: - The self check runs in background periodically, the absence of error messages means that memories and display hasn't suffered any problem. **Preconditions**: - None	**Functional**	**OM-FR93** **OM-FR94** **OM-FR95** **OM-FR101** **OM-FR104**	**PASSED**	KAIFA
5.4.1.5	**OM-SR1 Meter Registration**	**Goal**: - Retrieve the state of the Electricity meter/Communication hub. **Preconditions**: - None	**Functional**	**OM-FR85** **OM-FR86** **OM-FR87** **OM-FR91** **OM-FR92**	**PASSED**	KAIFA / BITRON

5.4.1.6	**OM-SR1 Meter Registration**	**Goal**: - Retrieve Electricity meter/Communication hub state before un-install it. **Preconditions**: - None	**Functional**	**OM-FR82**	**PASSED**	KAIFA
5.4.2.1	**OM-SR2 Remote Tariff Programming**	**Goal**: - Change of the 'tariff switch times'. **Preconditions**: - When the meter is not synchronized, use a configurable default tariff until its synchronization.	**Functional**	**OM-FR47** **OM-FR48** **OM-FR72** **OM-FR73**	**PASSED**	KAIFA / BITRON
5.4.3.1	**OM-SR3 Meter Reading (on demand)**	**Goal**: - Retrieve an actual meter read from an Electricity Meter **Preconditions**: - Meter reading on demand must be possible without affecting the periodic meter reading process.	**Functional**	**OM-FR9** **OM-FR151**	**PASSED**	KAIFA / BITRON

5.4.4.1	**OM-SR4 Meter Reading (for billing)**	**Goal**: - Verify that the equipment records periodic meter reads, that are available from the concentrator. **Preconditions**: - None.	**Functional**	**OM-FR2** **OM-FR3** **OM-FR4** **OM-FR150**	**PASSED**	KAIFA/BITRON
5.4.5.1	**OM-SR5 Remote Disconnection and Reconnection**	**Goal**: - Remote disconnection and reconnection of the supply of electrical power to a customer, on a designated date and at the specified time. **Preconditions**: - The Electricity Meter must be in connected state and with a load (1 [A]).	**Functional**	**OM-FR28** **OM-FR29** **OM-FR30** **OM-FR158**	**PASSED**	KAIFA/BITRON

5.4.5.2	**OM-SR5 Remote Disconnection and Reconnection**	**Goal**: - Remote disconnection and reconnection of the supply of electrical power. **Preconditions**: - The Electricity Meter must be in connected state.	**Functional**	**OM-FR28** **OM-FR30** **OM-FR31**	**PASSED**	KAIFA
5.4.5.3	**OM-SR5 Remote Disconnection and Reconnection**	**Goal**: - Check that the meter issue a logical error if the (dis)connection can't be applied. **Preconditions**: - The Electricity Meter must be in connected state.	**Functional**	**OM-FR32**	**PASSED**	KAIFA
5.4.6.1	**OM-SR6 Power Control**	**Goal**: - Set a threshold and verify that the meter works as expected. **Preconditions**: - A variable load is available.	**Functional**	**OM-FR69** **OM-FR34** **OM-FR35** **OM-FR36** **OM-FR162** **OM-FR163**	**PASSED**	KAIFA

5.4.6.2	**OM-SR6 Power Control**	**Goal**: - Apply a second threshold for a load shedding on a designated date and at the specified time. **Preconditions**: - A variable load is available.	**Functional**	**OM-FR33**	**PASSED**	KAIFA
5.4.7.1	**OM-SR7 Clock Synchronization**	**Goal**: - Verify that the synchronization functionality works as expected. **Preconditions**: - Verify that is possible wait 25 hours between the first part of the test and the second one.	**Functional**	**OM-FR50** **OM-FR51** **OM-FR167**	**PASSED**	KAIFA

5.4.8.1	**OM-SR8 Remote Frimware Update**	**Goal**: - Download on the meter of a new firmware's version, and then deploy it on a designated date and at the specified time. **Preconditions**: - A new version of the firmware is available.	**Functional**	**OM-FR57** **OM-FR58** **OM-FR60** **OM-FR61** **OM-FR62** **OM-FR63** **OM-FR64** **OM-FR168**	**PASSED**	KAIFA
5.4.8.2	**OM-SR8 Remote Frimware Update**	**Goal**: - Download and immediately deploy of a new firmware's version. **Preconditions**: - A new version of the firmware is available.	**Functional**	**OM-FR57** **OM-FR58** **OM-FR60** **OM-FR63** **OM-FR64** **OM-FR168**	**PASSED**	KAIFA

5.4.8.3	**OM-SR8 Remote Frimware Update**	**Goal**: - Verify that for an incomplete or inconsistent firmware, the meter discards it and issues a logical error. **Preconditions**: - A corrupted version of the firmware is available.	**Functional**	**OM-FR59** **OM-FR65**	**PASSED**	KAIFA
5.4.9.1	**OM-SR9 Alarm and Event Management**	**Goal**: - Check that alarm and event contains all the information required. **Preconditions**: - None	**Functional**	**OM-FR144** **OM-FR145** **OM-FR146** **OM-FR147** **OM-FR148**	**PASSED**	KAIFA
5.4.10.1	**OM-SR10 Interruption Information**	**Goal**: - Retrieve information about voltage interruption from a meter **Preconditions**: - Variable voltage device available	**Functional**	**OM-FR22** **OM-FR23** **OM-FR156** **OM-TR6** **OM-TR7**	**PASSED**	KAIFA

5.4.11.1	**OM-SR11 Fraud Datection**	**Goal**: - Check the activation of fraud detection alarm and the invocation of 'Use case 9: (Dis)connect Electricity', due to cover opened. **Preconditions**: - None	**Functional**	**OM-FR25** **OM-FR27** **OM-FR157** **OM-TR50**	**PASSED**	KAIFA
5.4.13.1	**OM-SR13 Load Profile Management**	**Goal**: - Check that the meter stores load profiles. **Preconditions**: - None	**Functional**	**OM-FR13** **OM-FR15**	**PASSED**	KAIFA

ANNEX B – OVERVIEW OF METERS and MORE FIELD TEST RESULTS (ENDESA)

Test Name	Test Case	Results (PASSED /FAILED /Not Tested)	Test description	Requirements	Meter manufacturer	Concentrator manufacturer
OM-SR1 Meter registration	5.4.1.2	**PASSED**	**The meter shall be able to respond to a communication initiated by the central system**	**OM-FR143**	**KAIFA**	**ENEL**
	5.4.1.3	**PASSED**	**Check the functionality of clock and calendar of the equipment. Set a new contract with specific power and one tariff.**	**OM-FR77 OM-FR78 OM-FR67 OM-FR68 OM-FR69 OM-FR72 OM-FR73 OM-FR79**	**KAIFA**	**ENEL**
	5.4.1.4	**PASSED**	**The self-check runs in background periodically, the absence of error messages means that memory and display had no problem.**	**OM-FR93 OM-FR94 OM-FR95 OM-FR101 OM-FR104**	**KAIFA**	**ENEL**
	5.4.1.5	**PASSED**	**Retrieve the state of the Electricity meter/Communication hub.**	**OM-FR85 OM-FR86 OM-FR87 OM-FR91 OM-FR92**	**KAIFA**	**ENEL**
OM-SR2 Remote Tariff Programming	5.4.2.1	**PASSED**	**Change the 'tariff switch times'.**	**OM-FR47 OM-FR48 OM-FR72 OM-FR73**	**KAIFA**	**ENEL**
OM-SR3 Meter Reading (on demand)	5.4.3.1	**PASSED**	**Retrieve meter reading**	**OM-FR9 OM-FR151**	**KAIFA**	**ENEL**

OM-SR4 Meter Reading (for billing)	5.4.4.1	PASSED	Verify that the equipment records periodic meter readings, that are available from the concentrator.	OM-FR2 OM-FR3 OM-FR4 OM-FR150	KAIFA	ENEL
OM-SR5 Remote Disconnection and Reconnection	5.4.5.1	PASSED	Remote disconnection and reconnection of the supply of electrical power to a customer, on a designated date and at the specified time.	OM-FR28 OM-FR29 OM-FR30 OM-FR158	KAIFA	ENEL
	5.4.5.2	PASSED	Remote disconnection and reconnection	OM-FR28 OM-FR30 OM-FR31	KAIFA	ENEL
	5.4.5.3	PASSED	Check that the meter issues a logical error if the (dis)connection can't be applied.	OM-FR32	KAIFA	ENEL
OM-SR6 Power Control	5.4.6.1	PASSED	Set a threshold and verify that the meter works as expected.	OM-FR69 OM-FR34 OM-FR35 OM-FR36 OM-FR162 OM-FR163	KAIFA	ENEL
	5.4.6.2	PASSED	Apply a second threshold for a load shedding on a designated date and at the specified time.	OM-FR33	KAIFA	ENEL
OM-SR8 Remote Firmware Update	5.4.8.2	PASSED	Download and immediately deploy a new firmware's version.	OM-FR57 OM-FR58 OM-FR60 OM-FR63 OM-FR64 OM-FR168	KAIFA	ENEL

	5.4.8.3	PASSED	**Verify that for an incomplete or inconsistent firmware, the meter discards it and issues a logical error.**	**OM-FR59 OM-FR65**	**KAIFA**	**ENEL**
OM-SR9 Alarm and Event Management	5.4.9.1	PASSED	**Check that alarm and event contains all the information required.**	**OM-FR144 OM-FR145 OM-FR146 OM-FR147 OM-FR148**	**KAIFA**	**ENEL**
OM-SR10 Interruption Information	5.4.10.1	PASSED	**Retrieve information about voltage interruption from a meter**	**OM-FR22 OM-FR23 OM-FR156 OM-TR6 OM-TR7**	**KAIFA**	**ENEL**
OM-SR11 Fraud Detection	5.4.11.1	PASSED	**Check the activation of fraud detection alarm and the invocation of 'Use case 9: (Dis)connect Electricity', due to cover opened.**	**OM-FR25 OM-FR27 OM-FR157 OM-TR50**	**KAIFA**	**ENEL**
OM-SR13 Load Profile Management	5.4.13.1	PASSED	**Check that the meter stores load profiles.**	**OM-FR13 OM-FR15**	**KAIFA**	**ENEL**

ANNEX C – OVERVIEW OF SFSK-DLMS LAB AND FIELD TEST RESULTS (by ITRON/LANDIS at EDF premises)

Functional tests

DC, testcase:		Meter: LANDIS + GYR	Meter: ITRON
LANDIS + GYR	4.4.2.1	**PASSED****	**PASSED****
	4.4.3.1	**PASSED**	**PASSED**
	4.4.4.1	**NT**	**NT**
	4.4.4.2	**PASSED**	**PASSED**
	4.4.4.3	**PASSED**	**PASSED**
	4.4.5.1	**PASSED**	**PASSED**
	4.4.5.2	**PASSED**	**PASSED***
	4.4.6.1	**PASSED*****	**PASSED**
	4.4.6.2	**PASSED**	**PASSED***
	4.4.6.3	**NT**	**NT**
	4.4.6.4	**NT**	**NT**
	4.4.7.1	**PASSED**	**PASSED**
	4.4.7.2	**PASSED**	**Passed**
	4.4.8.1	**NT**	**NT**
	4.4.8.2	**NT**	**NT**
	4.4.8.3	**NT**	**NT**
	4.4.10.1	**Passed**	**Passed**
	4.4.10.2	**Passed**	**Passed**
	4.4.11.1	**NT**	**NT**

	4.4.11.2	**Passed**	**Passed**
	4.4.13.1	**Passed**	**Passed**
	4.4.19.1	**Passed**	**Passed**
	4.4.19.2	**Passed**	**Passed**
ITRON	4.4.2.1	**Passed**	**Passed****
	4.4.3.1	**Passed**	**Passed**
	4.4.4.1	**NT**	**NT**
	4.4.4.2	**Passed**	**Passed**
	4.4.4.3	**Passed**	**Passed**
	4.4.5.1	**Passed**	**Passed**
	4.4.5.2	**Passed**	**Passed***
	4.4.6.1	**Passed*****	**Passed**
	4.4.6.2	**Passed**	**Passed***
	4.4.6.3	**NT**	**NT**
	4.4.6.4	**NT**	**NT**
	4.4.7.1	**Passed**	**Passed**
	4.4.7.2	**Passed**	**Passed**
	4.4.8.1	**NT**	**NT**
	4.4.8.2	**NT**	**Passed**
	4.4.8.3	**NT**	**Passed**
	4.4.10.1	**Passed**	**Passed**
	4.4.10.2	**Passed**	**Passed**
	4.4.11.1	**NT**	**NT**
	4.4.11.2	**Passed**	**Passed**
	4.4.13.1	**Passed**	**Passed**

	4.4.19.1	**Passed**	**Passed**
	4.4.19.2	**Passed**	**Passed**

* see 2.4.1 Breaker output state / control state
** see 2.4.2 Activation of Passedive activity calendar
*** see 2.4.3 Manual reconnection after Overload power

Interoperability lab tests:

		Meter	
DC, testcase:		LANDIS + GYR	ITRON
LANDIS + GYR	6.4.5.1	**Passed**	**Passed**
	6.4.5.2	**NT**	**NT**
ITRON	6.4.5.1	**Passed**	**Passed**
	6.4.5.2	**NT**	**Passed**

Interoperability field tests:

		Meter	
DC, testcase:		LANDIS + GYR	ITRON
LANDIS + GYR	7.4.1.1	**Passed**	**Passed**
	7.4.1.2	**Passed**	**Passed**
	7.4.1.3	**Passed**	**Passed**
	7.4.1.4	**Passed**	**Passed**
	7.4.1.5	**NT**	**NT**
	7.4.1.6	**Passed**	**Passed**
	7.4.1.7	**Passed**	**Passed**

	7.4.1.8	**Passed**	**Passed**
	7.4.1.9	**Passed**	**Passed**
	7.4.1.10	**NT**	**NT**
ITRON	7.4.1.1	**Passed**	**Passed**
	7.4.1.2	**Passed**	**Passed**
	7.4.1.3	**Passed**	**Passed**
	7.4.1.4	**Passed**	**Passed**
	7.4.1.5	**NT**	**NT**
	7.4.1.6	**Passed**	**Passed**
	7.4.1.7	**Passed**	**Passed**
	7.4.1.8	**Passed**	**Passed**
	7.4.1.9	**Passed**	**Passed**
	7.4.1.10	**NT**	**Passed**

ANNEX D – OVERVIEW OF PRIME LAB TEST RESULTS (KEMA)

'Switched' means that the test was done in a layered network, containing a Base Node, a Switch and at least one terminal node. The results for the switched tests are presented in the table on the next page.

DC, testcase:		Meter: LANDIS + GYR	Meter: ZIV	Meter: ELSTER	Meter: ITRON
ZIV	PRIME				
ZIV	6.4.3.1	NT	NT	NT	NT
ZIV	6.4.3.2	NT	NT	NT	NT
ZIV	6.4.3.3.	PASSED	PASSED	PASSED	PASSED
ZIV	6.4.3.4	switched	switched	switched	switched
ZIV	6.4.3.5	switched	switched	switched	switched
ZIV	6.4.3.6	PASSED	PASSED	FAILED(2)	NT
ZIV	6.4.3.7	PASSED	PASSED	NT	NT
ZIV	DLMS				
ZIV	6.4.5.1	PASSED(3)	PASSED(3)	PASSED(3)	PASSED(3,4,5)
ZIV	6.4.5.2	FAILED (6)	PASSED	FAILED(2)	NT
ZIV					
ZIV					
ZIV					
CURRENT	PRIME				
CURRENT	6.4.3.1	NT	NT	NT	NT
CURRENT	6.4.3.2	NT	NT	NT	NT
CURRENT	6.4.3.3.	PASSED	PASSED	PASSED	PASSED
CURRENT	6.4.3.4	switched	switched	switched	switched
CURRENT	6.4.3.5	switched	switched	switched	switched
CURRENT	6.4.3.6	FAILED (1)	PASSED	NT	NT
CURRENT	6.4.3.7	NT	PASSED	NT	NT
CURRENT	DLMS				
CURRENT	6.4.5.1	PASSED	PASSED	NT	NT
CURRENT	6.4.5.2	NT	NT	NT	NT
CURRENT					
CURRENT					
CURRENT					

General: ELSTER and ITRON prototypes were not yet 'PRIME Certified during the tests
(1) The loggings have been analyzed by CURRENT and L+G engineers, after which an issue in the Unicast FU procedure was found and could be fixed. No retest has been done in the OPEN meter project.
(2) After one hour waiting, the test was considered as FAILED.
(3) The step "SetRequest With DataBlock" could not be done from the GUI of the DC's and was skipped for that reason.
(4) Step Get RequestWithData block was skipped (montly billing object was not available)
(5) Step SetRequest was skipped (password for management client rejected).
(6) Loggings were analyzed by engineers from both companies, after which the issue could be solved. The DLMS FU in this combination has been retested during the tests at IBERDROLA, where the tests was passed. (see test case 7.4.1.10 in DLMS PRIME FIELD TESTS RESULT at IBERDROLA, where this tests is repeated in a broadcast setup)

Switched network:

DC, Test Case		LANDIS + GYR		ZIV		ELSTER		ITRON	
		LANDIS + GYR	ZIV	LANDIS + GYR	ZIV	ELSTER	ITRON	ELSTER	ITRON
ZIV	6.4.3.4	**PASSED**	**PASSED**	**PASSED**	**PASSED**	**NT**	**NT**	**NT**	**NT**
	6.4.3.5	**PASSED**	**PASSED**	**PASSED**	**PASSED**	**NT**	**PASSED**	**PASSED**	**NT**
CURRENT	6.4.3.4	**PASSED**	**PASSED**	**PASSED**	**PASSED**	**NT**	**PASSED**	**PASSED**	**NT**
	6.4.3.5	**PASSED**	**PASSED**	**PASSED**	**PASSED**	**NT**	**NT**	**NT**	**NT**

ANNEX E – OVERVIEW OF DLMS- PRIME FIELD TEST RESULTS (IBERDROLA)

Test Name	Test Case ID	Results (PASSED, FAILED, Not Tested (REASON: NOT APPLICABLE)	Test description	Requirements	Meter Manufacturer	Concentrator manufacturer	Which enviroment
Read instantaneous values	7.4.1.1	**PASSED**	The test cases are described in the [1]	**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**ZIV**	**CURRENT**	**LABORATORY**
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51**			
Read and modify tariff in curse	7.4.1.3	**PASSED**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**PASSED**		**OM-SR6;OM-FR33**			
Breaker functionality	7.4.1.5	**PASSED**		**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**			
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			

Read and check the correct schedule of the events created	7.4.1.9	**PASSED**		**OM-SR9;OM-FR34;**			
Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	**PASSED**		**OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65**			
Read instantaneous values	7.4.1.1	**PASSED**		**OM-SR1; OM-SR3;OM-FR1;OM-FR9**			
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51**			
Read and modify tariff in curse	7.4.1.3	**PASSED**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**PASSED**		**OM-SR6;OM-FR33**			
Breaker functionality	7.4.1.5	**PASSED**	The test cases are described in [1]	**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**	**LANDIS + GYR**	**CURRENT**	**LABORATORY**
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			

Read and check the correct schedule of the events created	7.4.1.9	**PASSED**		**OM-SR9;OM-FR34;**			
Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	**PASSED**		**OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65**			
Read instantaneous values	7.4.1.1	**PASSED**	The test cases are described in [1]	**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**ZIV**	**ZIV**	**LABORATORY**
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51;**			
Read and modify tariff in curse	7.4.1.3	**PASSED**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**PASSED**		**OM-SR6;OM-FR33;**			
Breaker functionality	7.4.1.5	**PASSED**		**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**			
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			

Read and check the correct schedule of the events created	7.4.1.9	**PASSED**		**OM-SR9;OM-FR34;**			
Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	**PASSED**		**OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65**			
Read instantaneous values	7.4.1.1	**PASSED**	The test cases are described in [1]	**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**LANDIS + GYR**	**ZIV**	**LABORATORY**
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51**			
Read and modify tariff in curse	7.4.1.3	**PASSED**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**PASSED**		**OM-SR6;OM-FR33;**			
Breaker functionality	7.4.1.5	**PASSED**		**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**			
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			

Read and check the correct schedule of the events created	7.4.1.9	PASSED		OM-SR9;OM-FR34;			
Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	PASSED		OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65			
Read instantaneous values	7.4.1.1	PASSED	The test cases are described in [1]	OM-SR1; OM-SR3;OM-FR1;OM-FR9	SAGEMCOM	CURRENT	LABORATORY
Read and clock modify	7.4.1.2	PASSED		OM-SR7;OM-FR50;OM-FR51			
Read instantaneous values	7.4.1.1	PASSED		OM-SR1; OM-SR3;OM-FR1;OM-FR9	SOGECAM		
Read and clock modify	7.4.1.2	PASSED		OM-SR7;OM-FR50;OM-FR51			
Read instantaneous values	7.4.1.1	PASSED		OM-SR1; OM-SR3;OM-FR1;OM-FR9	CIRCUTOR		
Read and clock modify	7.4.1.2	PASSED		OM-SR7;OM-FR50;OM-FR51			
Read instantaneous values	7.4.1.1	PASSED		OM-SR1; OM-SR3;OM-FR1;OM-FR9	SAGEMCOM	ZIV	
Read and clock modify	7.4.1.2	PASSED		OM-SR7;OM-FR50;OM-FR51			
Read instantaneous values	7.4.1.1	PASSED		OM-SR1; OM-SR3;OM-FR1;OM-FR9	SOGECAM		
Read and clock modify	7.4.1.2	PASSED		OM-SR7;OM-FR50;OM-FR51			

Read instantaneous values	7.4.1.1	**PASSED**		**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**CIRCUTOR**		
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51**			
Read instantaneous values	7.4.1.1	**PASSED**	The test cases are described in [1]	**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**ZIV**	**CURRENT**	**FIELD**
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51**			
Read and modify tariff in curse	7.4.1.3	**NT**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**NT**		**OM-SR6;OM-FR33;**			
Breaker functionality	7.4.1.5	**NT**		**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**			
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			
Read and check the correct schedule of the events created	7.4.1.9	**PASSED**		**OM-SR9;OM-FR34;**			

Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	**PASSED**		**OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65**			
Read instantaneous values	7.4.1.1	**PASSED**	The test cases are described in [1]	**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**LANDIS + GYR**	**CURRENT**	**FIELD**
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51;**			
Read and modify tariff in curse	7.4.1.3	**NT**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**NT**		**OM-SR6;OM-FR33;**			
Breaker functionality	7.4.1.5	**NT**		**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**			
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			
Read and check the correct schedule of the events created	7.4.1.9	**PASSED**		**OM-SR9;OM-FR34;**			

Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	**PASSED**		**OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65**			
Read instantaneous values	7.4.1.1	**PASSED**	The test cases are described in [1]	**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**ZIV**	**ZIV**	**FIELD**
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51;**			
Read and modify tariff in curse	7.4.1.3	**NT**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**NT**		**OM-SR6;OM-FR33;**			
Breaker functionality	7.4.1.5	**NT**		**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**			
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			
Read and check the correct schedule of the events created	7.4.1.9	**PASSED**		**OM-SR9;OM-FR34;**			

Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	**PASSED**		**OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65**			
Read instantaneous values	7.4.1.1	**PASSED**	The test cases are described in [1]	**OM-SR1; OM-SR3;OM-FR1;OM-FR9**	**LANDIS + GYR**	**ZIV**	**FIELD**
Read and clock modify	7.4.1.2	**PASSED**		**OM-SR7;OM-FR50;OM-FR51;**			
Read and modify tariff in course	7.4.1.3	**NT**		**OM-SR2**			
Read and modify power threshold	7.4.1.4	**NT**		**OM-SR6;OM-FR33;**			
Breaker functionality	7.4.1.5	**NT**		**OM-SR5;OM-FR28;OM-FR29;OM-FR30;OM-FR35;OM-FR36**			
Read and check hourly load profile	7.4.1.6	**PASSED**		**OM-SR4;OM-SR13;OM-FR8;OM-FR13;OM-FR14;OM-FR15**			
Read and check monthly billing values	7.4.1.7	**PASSED**		**OM-SR4**			
Read and check daily billing	7.4.1.8	**PASSED**		**OM-SR4;OM-FR2;OM-FR3**			
Read and check the correct schedule of the events created	7.4.1.9	**PASSED**		**OM-SR9;OM-FR34;**			

Check the correct functionality of firmware upgrade (DLMS and PRIME)	7.4.1.10	**PASSED**		**OM-SR8;OM-FR56;OM-FR57;OM-FR59;OM-FR60;OM-FR61;OM-FR62;OM-FR63;OM-FR64;OM-FR65**			

ANNEX F – OVERVIEW OF DLMS – G3 LAB and FIELD TEST RESULTS (EDF)

Test Name	Test Case ID	Results (PASSED, FAILED /NOT TESTED)	Test description	Requirements	Meter Manufacturer	Concentrator manufacturer
Plug & Play registering	3.4.2.1	PASSED	Verify the plug and play registering and network topology changes behavior of the PLC prototypes.	OM-CR 6	SAGEMCOM	SAGEMCOM
Network topology changes behaviour	3.4.2.2	PASSED	Verify the behaviour of the PLC prototypes, when network topology changes	OM-CR 8	SAGEMCOM	SAGEMCOM
Firmware update test	4.4.8.1	PASSED	Verify that the firmware of the E meter can be upgraded remotely with a predefined deployment date. The new version of the firmware shall be deployed in the meter the first calendar day after the test is started.	OM-FR57,OM-FR58,OM-FR61,OM-FR64	SAGEMCOM	SAGEMCOM
Firmware update test	4.4.8.2	PASSED	Verify that the firmware of the E meter can be upgraded remotely immediately (without deployment date).	OM-FR57,OM-FR58,OM-FR61,OM-FR64	SAGEMCOM	SAGEMCOM

Firmware update test	4.4.8.3	PASSED	Verify that the E meter generates an error in case the version of the firmware is incomplete and/or inconsistent.	OM-FR57,OM-FR59	SAGEMCOM	SAGEMCOM
Fraud detection	4.4.11.1	PASSED	Check the activation of the event of fraud detection due to cover opened via DLMS and the absence any indicator visible in display.	OM-FR25,OM-FR27	SAGEMCOM	SAGEMCOM
Remote disconnection/re-connection	4.4.5.1	PASSED	Disconnect (no time parameter) an E meter and re-connect it. (remotely)	OM-FR28	SAGEMCOM	SAGEMCOM
Scheduled disconnection/re-connection	4.4.5.2	PASSED	Scheduled Disconnection of E meter and re-connection. (with time stamp).	OM-FR28	SAGEMCOM	SAGEMCOM
Power Control	4.4.6.1	PASSED	Set the contractual power level for an E meter immediately (i.e. without providing a timestamp)	OM-FR33, OM-FR35	SAGEMCOM	SAGEMCOM
Power Control	4.4.6.2	PASSED	Clear the contractual power level for an E meter immediately (i.e. without providing a timestamp)	OM-FR33, OM-FR35,OM-FR160	SAGEMCOM	SAGEMCOM
Load profile management	4.4.13.1	PASSED	Check the existence of the defined load profiles.	OM-FR13, OM-FR14,OM-FR15	SAGEMCOM	SAGEMCOM
Meter reading (on demand)	4.4.3.1	PASSED	Retrieve an actual meter read from an existing E meter	OM-FR9, OM-FR151	SAGEMCOM	SAGEMCOM

Remote tariff programming	4.4.2.1	PASSED	Set the tariff times for a E meter for the next calendar day for a given meter	OM-FR 47	SAGEMCOM	SAGEMCOM
Meter reading (for billing)	4.4.4.1	PASSED*	The objective of this test is to verify the End of billing data. During this test 4 type of EOB data will be verified: The monthly, the daily, and the asynchronous EOB data for a preprogrammed date and immediate.	OM-FR2, OM-FR3,OM-FR4, OM-FR150	SAGEMCOM	SAGEMCOM
Meter reading (for billing)	4.4.4.2	PASSED*	Read Monthly billing Values Profile	OM-FR4	SAGEMCOM	SAGEMCOM
*EOB still to be taken into account by the legacy system						