Integrated ICT-platform based Distributed Control in electricity grids with a large share of Distributed Energy Resources and Renewable Energy Sources

Deliverable D5.2

Simulation test report

PowerMatching City

Integration Test Plan

Jörgen van der Velde, Eric Bakker,

Gerben Venekamp, René Kamphuis, Sjaak Kaandorp

Identifier:	Integral_D5.2C_1.0
	ICT Doc ID: ICT 2008938286.2.5-001
Date:	25-02-2009
Class:	Deliverable
Responsible Partners:	GET, ECN, ICT
Annexes:	
Distribution:	PU

Overview:

This project is funded by the European Commission
Under the 6th Framework Programme
(Project FP6-038576)

<u>The INTEGRAL consortium consists of:</u>

ECN	Principal Contractor & Coordinator	The Netherlands
NTUA/ICCS	Principal Contractor	Greece
IDEA	Principal Contractor	France
Blekinge University of Technology	Principal Contractor	Sweden
Gasunie Engineering&Technology	Principal Contractor	The Netherlands
WattPic Intelligent	Principal Contractor	Spain
EnerSearch AB	Principal Contractor	Sweden
INPGrenoble	Principal Contractor	France
ICT	Principal Contractor	The Netherlands

Control Versions:

Version	Date	Author	Description of Changes
0.1	25-02-2009	Jörgen van der Velde	Document template
0.1	27-02-2009	Eric Bakker	Test cases added
0.2	19-03-2009	Eric Bakker	review comments processed
0.3	21-04-2009	Eric Bakker	Added test cases (AMR/Heatpump/DB sync)
0.4	21-04-2009	Eric Bakker	Added Agent integration test cases (from Gerben Venekamp)
0.5	07-05-2009	Gerben Venekamp	Added additional testcases
0.6	08-05-2009	Eric Bakker	some textual issues
1.0	09-06-2009	René Kamphuis, Sjaak Kaandorp	Finalized

Distribution:

Version	Date	To	Company
1.0	10-06-2009	Partners	
1.0	15-6-2009	Public via Website	

Table of Contents

List of Figures

References

Reference	Document description
[DoW]	INTEGRAL Description of Work
[D4.4]	A Common Experiment Design Framework leading to an
	Integrated ICT-platform for Distributed Control
[SYSSPEC]	PowerMatching City
	System Specification
	D5.1a
[ADD]	PowerMatching City
	D5.1b
[TEMB]	Testing Embedded Software
	Broekman / Notenboom
[NOMENCLATURE]	PowerMatching City
	Terms and Abbreviations
	D5.1c
[PLAN]	Planning and milestones of WP5
[D8.1]	Integral Deliverable D8.1
	Definition of test and evaluation procedures
[IDD_HS]	Heat system Control Design
[SIM_TEST_SPEC]	Describes the test cases for the simulation tests (Field test)
	D5.2
[DB_PARAM_DEF]	Lists the parameters (configuration, control and measurements) defined in the database

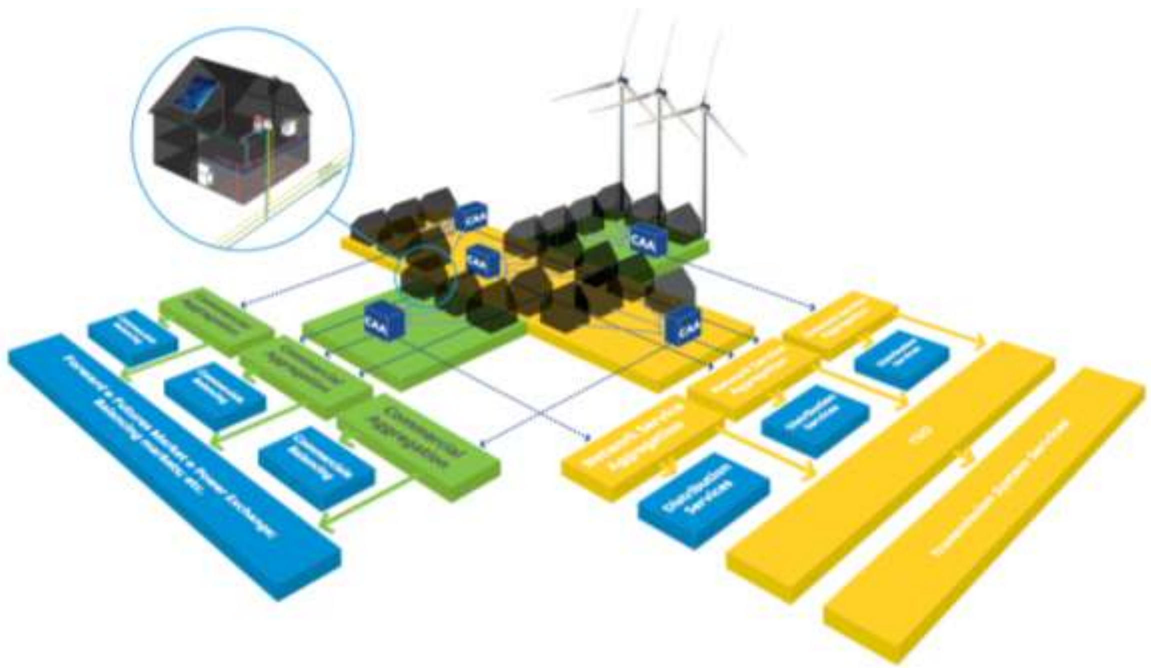
Acronyms and Abbreviations

See [NOMENCLATURE]

Executive Summary

The INTEGRAL project aims to build and demonstrate an industry-quality reference solution for Distributed Energy Resources (DER) aggregation-level control and coordination, based on commonly available ICT components, standards and platforms.

PowerMatching City, also known as Demonstrator A of the INTEGRAL project, aims to demonstrate control of DER under normal operational circumstances. The control concept will be PowerMatcher, developed by ECN. The Commercial Aggregator (CA) delivers electricity to the Prosumers or buys it. The Distribution System Operator (DSO) offers distribution capacity.



This document describes the way the PowerMatching City system is tested prior to roll out in the (friendly user) field test.

Following the general IIDC-approach sketched in D4.4, integration testing is split up in three types of tests:

- The Software Integration Test
  This test focuses on correct functioning of the software after integration

- The System Integration Test
  This test focuses on correct functioning of the entire system after integration

- The Lab Testing
  This test focuses on testing the system in a number of scenarios which are simulated

In this subdocument the first testing strategies are covered.

1 Introduction

1.1 Purpose

This document describes the test designs, the test cases and procedures used to verify that the system components are integrated correctly and perform in the intended way. It applies to various steps in the INTEGRAL WP5, which is described below.

This document is intended for the project team members that are involved in the Software Integration Test, the System integration Test and the Lab Testing.

This test document is written in the format of an acceptance test. The system (hardware and software) under test will be as close as possible to the end product. Mainly black box testing is used. To some extent, grey box testing is performed, in order to determine specific test results.

1.2 Scope

The scope of this document are the Integration Tests. These tests encompass

- The Software Integration Test
  This test focuses on correct functioning of the software after integration

- The System Integration Test
  This test focuses on correct functioning of the entire system after integration

The actual simulation tests (Field test) are described in [DB_PARAM_DEF]. This test focuses on testing the system in a number of scenarios. Part of these test may be executed by simulation.

For each of the tests a separate chapter will focus on the details.

The system will be tested against the requirements as laid down in the System Specification [SYSSPEC] and Architectural Design Document [ADD]. A test should be defined for every essential user requirement for as far as such a requirement has not been (can not be) tested in the individual application or module tests.

Execution is a responsibility of the project team and will be performed at the GET laboratory.

1.3 Deliverable 5.2

This document forms the basis of deliverable 5.2 'Simulation results report'. As described in [DoW] the simulation test will test and fine-tune the control concepts and strategies of the PowerMatcher in a simulation environment. Simulations will start in a software environment, using the PowerMatcher concept, and will be carried through to the GET laboratory, where realistic behavior of houses and their occupants can be emulated.

2 Context of this Integration Test

2.1 Scope of the Integration test

The PowerMatching City System is set-up and tested in a laboratory environment. The Integration Tests check

- Functionally: do all components collaborate adequately? Does the system behave like it was intended? Does it control the MicroCHP like it was meant to be?
- Comfort: does the user always have hot tap water? Does the system respond adequately to heat demand from the household?
- Quality: is the system sufficiently stable? Does the system use resources (like the GPRS connection) in the proper way?

The tests are executed during development of the system and as final acceptation test for the system for deployment in the friendly-user test phase.

As stated this test document is split in two parts (see 1.2). This document encompasses

- The Software Integration Test
- The System integration Test

2.2 Test basis

The bases of the content of this test document are project plan, product specifications and interface description. In Table 1 an overview is given of the corresponding documentation that forms the test basis. For formal references see the paragraph References (page 6) for details. Of the documents the versions listed in that paragraph are used.

Document Reference	Description
[DoW]	The description of work describes the goals of the INTEGRAL project
[SYSSPEC]	The System Specification defines the system and the requirements
[ADD]	This document describes the system and sub-systems.
[D8.1]	This document describes the execution of the field test. It defines the goals to be achieved with the PowerMatching City system

Table 1 Test basis documentation

2.3 Subdivision of test cases

Since the PowerMatching City system is a large system consisting of a number of sub-systems a subdivision will be made of the test cases. Following levels are foreseen:

- Household/Laboratory level
  Testcases at this level concern the domestic sub-systems and components. At this level the GET laboratory is seen as one or more households.

- Wind park level
  Testcases at this level concern

- Central control level
  Testcases concern centralised sub-systems, like the Central Storage, the Commercial Aggregator sub-system, the DSO sub-systems and the portals.

- System level
  Testcases at this level concern operation of the PowerMatching City as a whole

2.4 Incremental integration

Integration takes place in a number of increments as described in the planning. Chapters in this document describe test cases in a structure that does not take into account these increments. Therefore, for each test case the increment shall be indicated in which the test case is executed for the first time.

3 Test Procedures

3.1 Purpose

The test procedure describes how all test cases are executed. Deviations from this procedure will be described in the concerned test case specification.

3.2 Procedure steps

To execute the test cases, the next steps should be taken care of:

Step	Action
Log	Describe how the results of the test execution are reported
Set up	Build the system set-up before starting with the test cases.
For each test case:	
Prepare	Prepare the system for each test case, so it starts within a predefined state.
Time	***Register the date & time at which a test case or input to the system is applied!*** Because parts of the system may be constantly monitoring system values it is important to know what has caused the changes on the ‘output’ values. This is in particular important for the Lab Test test cases.
Execute	Execute the test case as described.
Wrap up	Restore the environment when the test case has to be restarted, repeated etc.
Report	Fill in the test log.

3.2.1 Log test results

For each test case the results are logged in a test log. For each executed test case the results are recorded. The following results are possible:

Pass	:	the test case is executed successfully
Fail	:	the test case failed, submit a PR and record the PR id
Not executed	:	the test case was not executed, indicate the reason why.

3.2.2 Feature pass/fail criteria

A test case passed when the actual result of the test matches the expected result. If not it fails. If the expected result is inaccurately described and the outcome of the test is different then the test fails and a PR is submitted on the Test Spec description.

3.2.3 Report

At completion of the tests the test logs will be summarized in test reports. The test reports are made according to template D5.2B. The test reports will be used by the project leader and representatives of the customer to determine if the software and system is mature enough for the Field test phase.

3.3 Test case identification

Each test case is labeled by a unique identifier. This number is used for submitting PRs and referencing. The format of the identifiers is as follows:

Identifier	Description
SWI-xx-yyy	SoftWare Integration test case. xx subject code yyy individual test spec number
SI-xx-yyy	System Integration test case. xx subject code yyy individual test spec number

Following sub-codes are used

Identifier	Description
01	Household/laboratory level
02	Wind turbine level
03	Central Control level
04	System level

3.4 Test case description

Test cases in this document will be described in following format:

<Identification>	
Objectives	<Objectives>
Increment	<Increment>
Test environment	<Test environment>
Preconditions	<Preconditions>
Inputs	<Inputs>
Expected result	<Result>
Wrap up	<Wrap up>
Remarks	<Remarks>

<Identification>	Identification of the testcase (see 3.3)
<Objectives>	This item describes the objectives of the test case, i.e. what is going to be tested
<Incremement>	The first integration increment at which this test case is executed
<Test environment>	The test environment used to execute the test
<Preconditions>	Lists the pre conditions before starting the test.
<Inputs>	All additional artefacts, equipment, data that is required for execution of the test, apart from the test set-up.
<Expected result>	The expected result, i.e. the exact condition which must be fulfilled in order for the test case to pass
<Wrap up>	Additional steps to be taken when the test case has been executed (e.g. collection of test data required for other tests)
<Remarks>	Additional remarks for the executor of the test case

4 Software Integration Test

4.1 Scope of the Software Integration Test

This test is executed when all relevant software components have been integrated.

4.1.1 Software to be tested

The software to be tested are the software components that are relevant for the integration step.

4.1.2 Software not to be tested

-

4.2 Objective

The objective of this test is to verify correct functioning of the software.

4.3 Preconditions

-

4.4 Assumptions

-

4.5 Infrastructure

-

4.6 Test environment

The tests will be executed in the GET laboratory during the integration phase. A number of tests might be executed prior and on other locations. Next paragraphs describe the test environment specific for testing a device or a combined test setup.

4.6.1 CHP

Next figure shows the test setup for testing the CHP device, defined as **CHP01**:

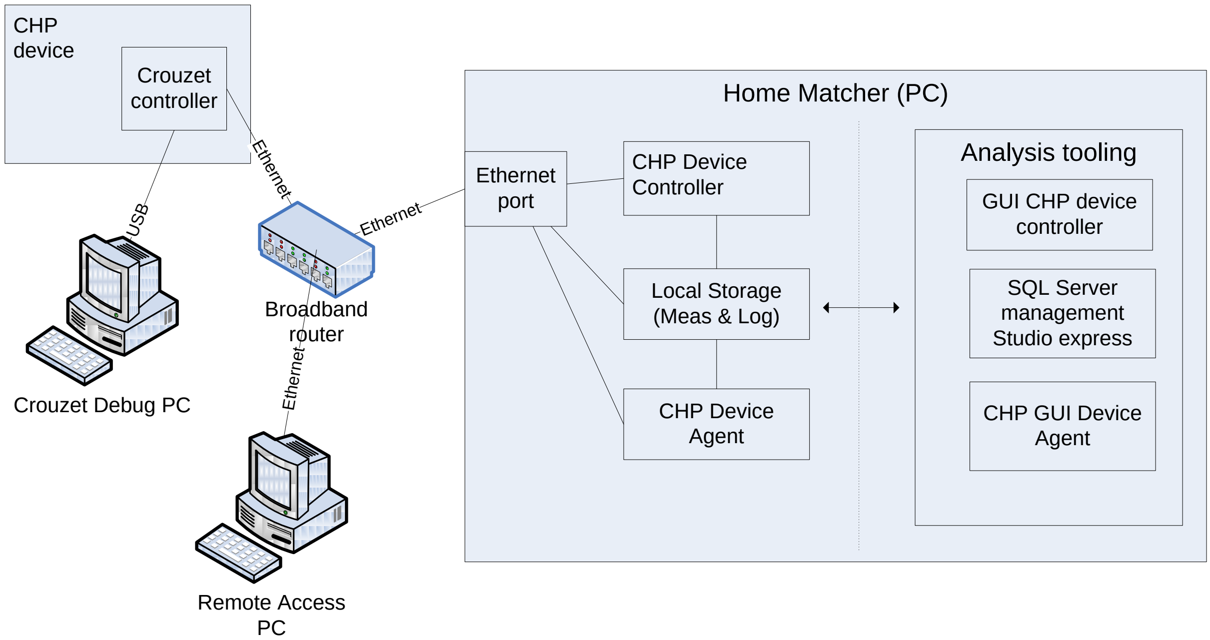


Figure 1 Test setup for CHP.

Next components can be identified :

Subsystem	Description
CHP device	CHP unit to be tested. The CHP unit is controlled with the Crouzet controller (PLC).
Crouzet PLC	The controller of the CHP. The controller is connected to a Crouzet debug PC (using an USB connection). The CHP is connected to the Homematcher PC using an ethernet connection.
Homematcher	Domestic PC running the software for controlling, measuring and communication with devices and network (central storage) as will be used on site: • Heat system device controller (as CHP) • Local storage • CHP device Agent
Analysis tooling	On the Homematcher PC test tooling (software) will be installed / available for testing: • GUI for heat system device controller • GUI for CHP device agent • SQL server management studio express
Broadband router	Using the switch functionality of the router to interconnect the hardware device (CHP) with the Homematcher.
Remote Access PC	PC used for gaining remote access to the Homematcher PC.

4.6.2 Heat pump

Next figure shows the test setup for testing the heat pump device, defined as **HP01**:

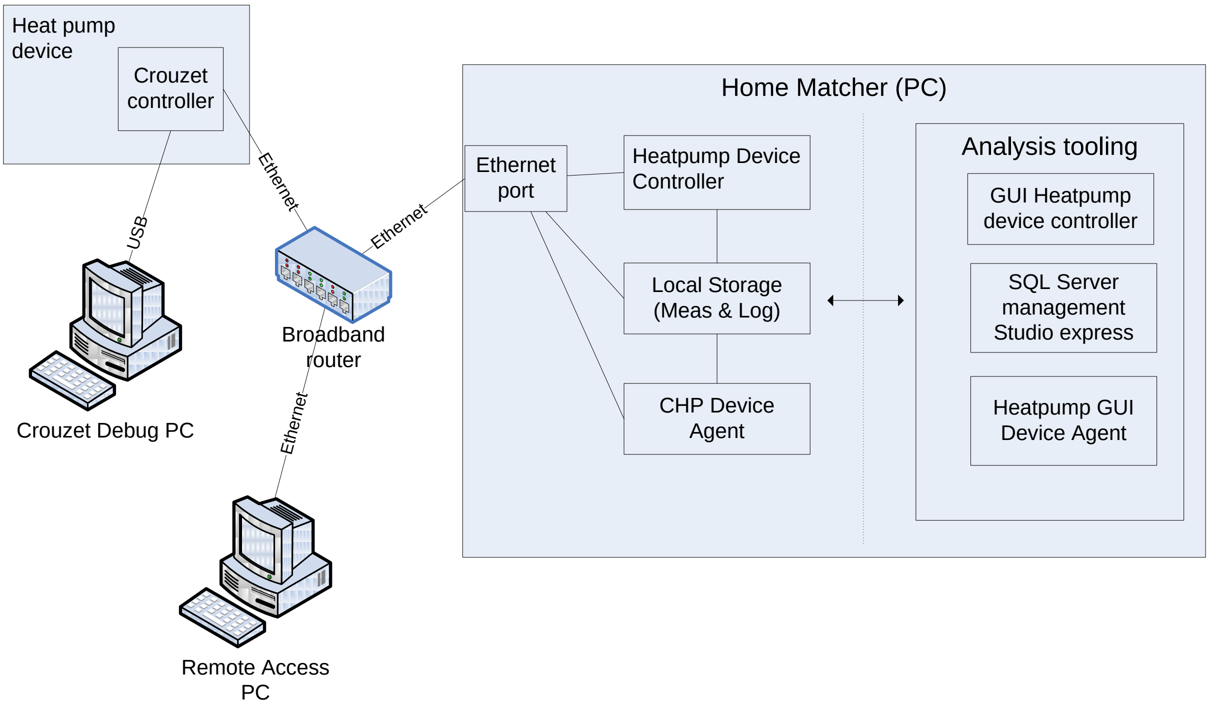


Figure 2 Test setup for heat pump.

Next components can be identified :

Subsystem	Description
Heat pump device	Heat pump unit to be tested. The heat pump unit is controlled with the Crouzet controller (PLC).
Crouzet PLC	The controller of the CHP. The controller is connected to a Crouzet debug PC (using an USB connection). The heat pump is connected to the Homematcher PC using an ethernet connection.
Homematcher	Domestic PC running the software for controlling, measuring and communication with devices and network (central storage) as will be used on site: • Heat system device controller (as heat pump) • Local storage • Heat pump device Agent
Analysis tooling	On the Homematcher PC test tooling (software) will be installed / available for testing: • GUI for heat system device controller • GUI for CHP device agent • SQL server management studio express
Broadband router	Using the switch functionality of the router to interconnect the hardware device (heat pump) with the Homematcher.
Remote Access PC	PC used for gaining remote access to the Homematcher PC.

4.6.3 AMR

Next figure shows the test setup for testing the AMR device (the home meter), defined as **AMR01**:

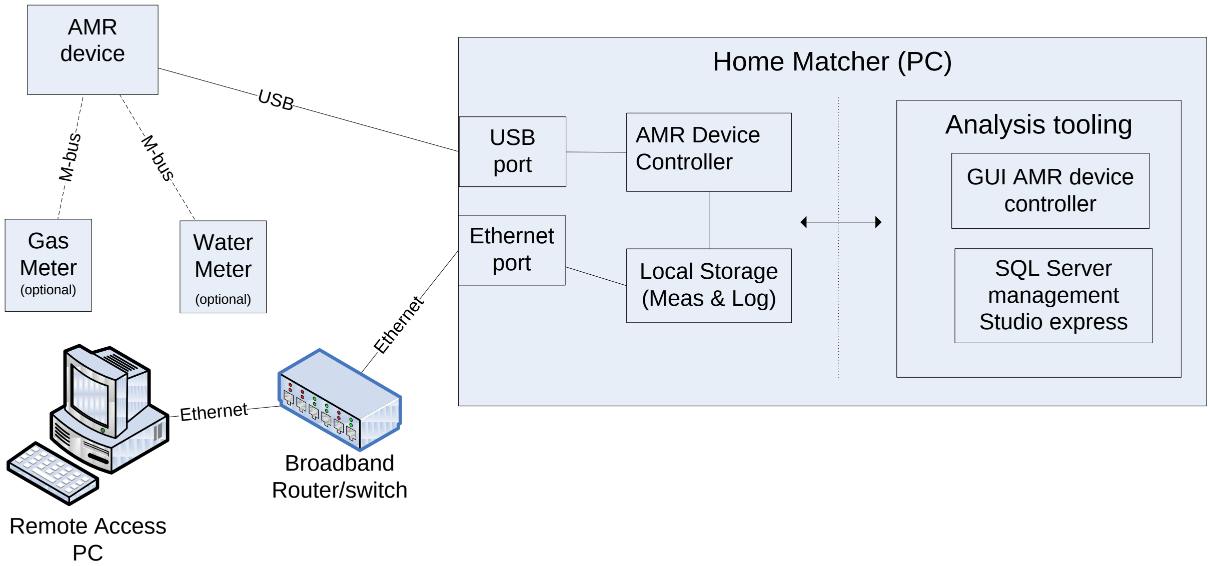


Figure 3 Test setup for AMR (home meter).

Next components can be identified :

Subsystem	Description
AMR device	The AMR unit to be tested. The Actaris ACE4000GPR smart meter is used. optional the home metering devices for gas and water can be connected using the M-bus (port P2) of the Actaris meter.
Homematcher	Domestic PC running the software for controlling, measuring and communication with devices and network (central storage) as will be used on site: AMR device controller (AMR Home) Local storage
Analysis tooling	On the Homematcher PC test tooling (software) will be installed / available for testing: GUI for AMR device controller (AMR Home) SQL server management studio express
Broadband router	Using the switch functionality of the router to interconnect other hardware (remote PC) with the Homematcher.
Remote Access PC	PC used for gaining remote access to the Homematcher PC.

4.6.4 AMR extended (M-bus devices)

Next figure shows the test setup for the M-bus meters that will be read using a fast poll-rate, defined as **AMR02**:

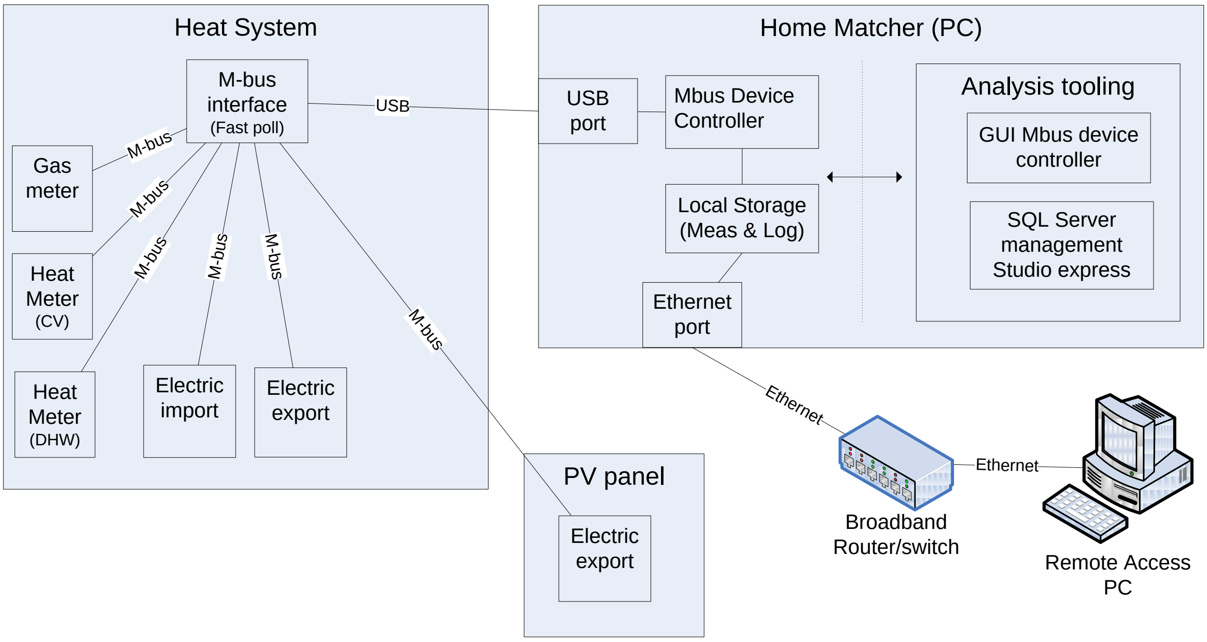


Figure 4 Test setup for AMR extended reading (fast poll M-bus devices).

Next components can be identified :

Subsystem	Description
M-bus Interface (fast poll)	The M-bus unit to be tested. It is not yet clear what hardware device is used.
Gas meter	Gas meter in the Heat system registering the actual gas usage
Heat meters	Heat meter registering the actual delivered heat (both CV and DHW)
Electric Heat System (import/export)	The electricity meters registering import and export (CHP) of the electrical energy.
Electric PC panel (export only)	Electricity meter registering the export of the PV panel.
Homematcher	Domestic PC running the software for controlling, measuring and communication with devices and network (central storage) as will be used on site:
	AMR extended device controller.
	Local storage
Analysis tooling	On the Homematcher PC test tooling (software) will be installed / available for testing:
	GUI for AMR extended device controller
	SQL server management studio express
Broadband router	Using the switch functionality of the router to interconnect other hardware (remote PC) with the Homematcher.
Remote Access PC	PC used for gaining remote access to the Homematcher PC.

4.6.5 Central Storage

Next figure shows the test setup for testing the Central storage, defined as **CS01**:

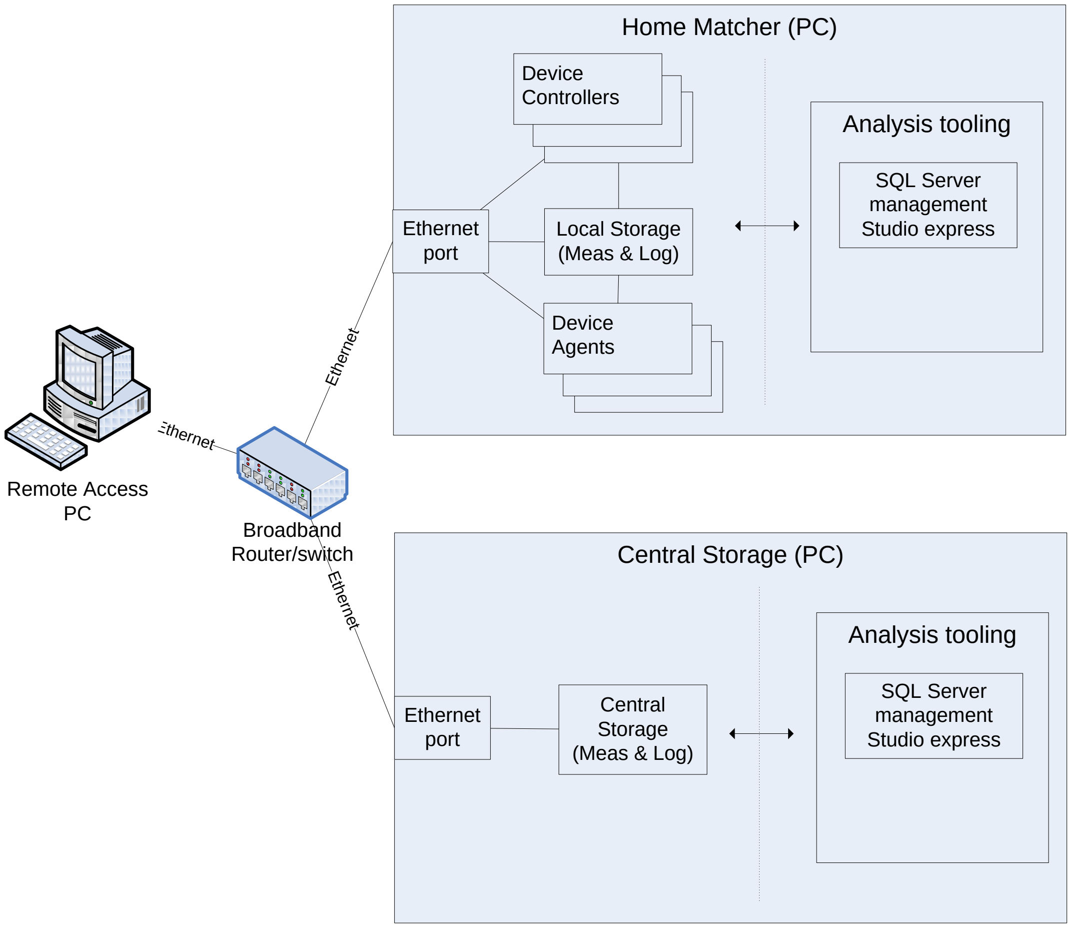


Figure 5 Test setup for Central Storage.

Next components can be identified :

Subsystem	Description
Homematcher	Domestic PC running the software for controlling, measuring and communication with devices and network (central storage) as will be used on site: • Device controller(s) • Local storage • Device Agents
Central storage	PC running the central storage functionality.
Analysis tooling	On both the Homematcher PC and Central storage server (PC) Microsofts “SQL server management studio express“ is installed / available for analysing test results.
Broadband router	Using the switch functionality of the router to interconnect the hardware devices with the Homematcher
Remote Access PC	PC used for gaining remote access to the Homematcher PC.

4.6.6 Complete system

Next figure shows the test setup for testing the Complete system, defined as **ALL01**:

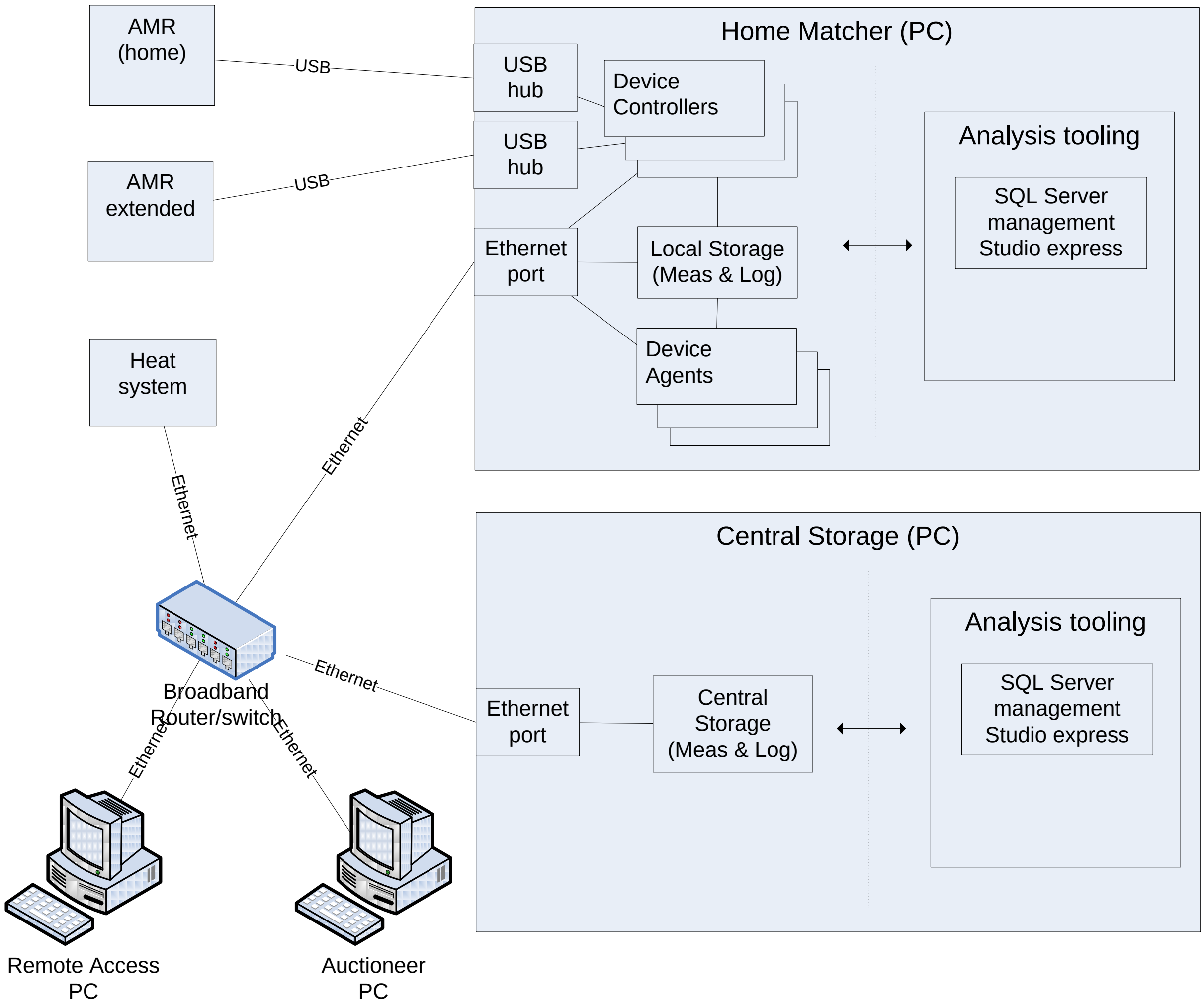


Figure 6 Test setup with all components.

Next components can be identified :

Subsystem	Description
Homematcher	Domestic PC running the software for controlling, measuring and communication with devices and network (central storage) as will be used on site: Device controller(s) Local storage Device Agents
AMR (home)	The actual home meter of the householdd
AMR extended	The extended AMR for reading the heat system parameters and optional a PV panel
Heat system	Heat pump or micro CHP
Central storage	PC running the central storage functionality.
Analysis tooling	On both the Homematcher PC and Central storage server (PC) Microsofts "SQL server management studio express" is installed / available for analysing test results.
Broadband router	Using the switch functionality of the router to interconnect the hardware devices with the Homematcher
Auctioneer PC	PC used for running the Auctioneer (simulation) software.
Remote Access PC	PC used for gaining remote access to the Homematcher PC.

4.7 Test Case Specifications

This chapter lists the actual test cases for the Software Integration Tests. Test environments are described in section 4.6.

4.7.1 Household / laboratory level test cases

4.7.1.1 System-startup

SWI-01-001	
Objectives	Tests correct start-up of the system when Homematcher PC is started , concerning next software components and associated hardware device: - Measurements & Logging (PC) - microCHP (HeatSystem) device **controller** (PC) Establishing connection between components: microCHP device controller and Measurements & Logging Establishing connection between components: microCHP device controller and Crouzet controller (based on the configuration parameters defined in lMeasurements & Logging)
Increment	1
Test environment	**CHP01**
Preconditions	Homematcher PC is not running microCHP Crouzet controller (PLC) is running
Inputs	Start Homematcher PC
Expected result	Homematcher starts and connections are established between components: - microCHP (HeatSystem) device controller and Measurements & Logging. - microCHP (HeatSystem) device controller and Crouzet controller
Wrap up	-
Remarks	-

SWI-01-002	
Objectives	Tests correct start-up of the system when Homematcher PC is started , concerning next software components: - Measurements & Logging (PC) - microCHP device **agent** (PC) Establishing connection between components: microCHP device agent and Measurements & Logging.
Increment	1
Test environment	**CHP01**
Preconditions	Homematcher PC is not running Heatsystem Crouzet controller (PLC) is running
Inputs	Start Homematcher PC
Expected result	Homematcher starts and connections are established between components: - microCHP device agent and Measurements & Logging.
Wrap up	-
Remarks	-

SWI-01-003	
Objectives	Tests correct start-up of the system when Homematcher PC is started , concerning next software components and associated hardware device: - Measurements & Logging (PC) - AMR Home device **controller** (PC) Establishing connection between components: AMR device controller and Measurements & Logging, based on the configuration parameters defined. Establishing connection between components: AMR and the actual E-meter in the cupboard.
Increment	2
Test environment	**AMR01**
Preconditions	Homematcher PC is not running, E-meter is running and connected to PC
Inputs	Start Homematcher PC
Expected result	Homematcher starts and connection are established between components: - AMR device controller and Measurements & Logging. - AMR device controller and E-meter
Wrap up	-
Remarks	-

SWI-01-004	
Objectives	Tests correct start-up of the system when Homematcher PC is started , concerning next software components and associated hardware device: - Measurements & Logging (PC) - AMR Extended device **controller** (PC) Establishing connection between components: AMR extended device controller and Measurements & Logging, based on the configuration parameters defined. Establishing connection between components: AMR extended and the actual M-bus devices (as part of the heat system).
Increment	2
Test environment	**AMR02**
Preconditions	Homematcher PC is not running, M-bus devices are running and connected to PC
Inputs	Start Homematcher PC
Expected result	Homematcher starts and connection are established between components: - AMR device controller and Measurements & Logging. - AMR extended device controller and the M-bus devices
Wrap up	-
Remarks	-

SWI-01-005	
Objectives	Tests correct start-up of the system when Homematcher PC is started , concerning next software components and associated hardware device: - Measurements & Logging (PC) - Heat pump (HeatSystem) device **controller** (PC) Establishing connection between components: Heat pump device controller and Measurements & Logging Establishing connection between components: Heat pump device controller and Crouzet controller (based on the configuration parameters defined in lMeasurements & Logging)
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is not running Heat pump Crouzet controller (PLC) is running
Inputs	Start Homematcher PC
Expected result	Homematcher starts and connections are established between components: - Heat pump (HeatSystem) device controller and Measurements & Logging. - Heatpump (HeatSystem) device controller and Crouzet controller
Wrap up	-
Remarks	-

SWI-01-006	
Objectives	Tests correct start-up of the system when Homematcher PC is started , concerning next software components: - Measurements & Logging (PC) - Heat pump device **agent** (PC) Establishing connection between components: Heat pump device agent and Measurements & Logging.
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is not running Heatsystem Crouzet controller (PLC) is running
Inputs	Start Homematcher PC
Expected result	Homematcher starts and connections are established between components: - Heat pump device agent and Measurements & Logging.
Wrap up	-
Remarks	-

4.7.1.2 MicroCHP

SWI-01-010	
Objectives	Test interface between: - MicroCHP (HeatSystem) device controller (PC) - MicroCHP Crouzet controller (PLC) All signals defined in the IDD_HS (Heat System Control Design) must be tested for correct functioning. Changes made to Crouzet outputs should be reflected in the Device Controller GUI. Also changing Crouzet inputs (using the Device controller GUI) should be reflected within the Crouzet PLC.
Increment	1
Test environment	**CHP01**
Preconditions	Homematcher PC is running and connection between Heatsystem device controller (PC) and Crouzet controller (PLC) is active. The Crouzet is monitored using the Crouzet monitoring tooling
Inputs	signals as defined in IDD (Heat system Control Design)
Expected result	All signals must be correctly reflected within the counterpart device.
Wrap up	-
Remarks	-

SWI-01-011	
Objectives	Test measurement interface between: - Measurements & Logging (PC) - MicroCHP (HeatSystem) device controller (PC) The defined measurement parameters in Measurements & Logging should actually be logged to the local Database. Check next items for every measured parameter: - Measured values - Measurement interval
Increment	1
Test environment	
Preconditions	Homematcher PC is running and connection between microCHP device controller (PC) and Measurements & Logging (PC) is active. The microCHP device controller runs in simulation mode (input of measured data using simulation data file). Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management).
Inputs	Measurement parameters as defined in IDD (Heat system Control Design). Use either simulation file for generating data or life system where values can be set.
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for given parameter.
Wrap up	-
Remarks	-

SWI-01-012	
Objectives	Test controlling interface between: - Measurements & Logging (PC) - MicroCHP (HeatSystem) Device controller (PC) The defined control parameters in Measurements & Logging should be reflected on microCHP device controller software (GUI) regarding the control update interval. Check next items for every control parameter: - Control value is read correctly from DB - Reading interval is correct
Increment	1
Test environment	**CHP01**
Preconditions	Homematcher PC is running and connection between microCHP device controller (PC) and Measurements & Logging (PC) is established
Inputs	Control signals as defined in IDD (Heat system Control Design). Control values are changed using a DB management tool.
Expected result	Changed control values should be reflected in the GUI of the microCHP device controller GUI.
Wrap up	-
Remarks	-

SWI-01-013	
Objectives	Test total interface between: - Measurements & Logging (PC) - MicroCHP Crouzet controller (PLC) The actual Crouzet measurement parameters should be logged in Measurements & Logging. Changed control parameters should be send to the Crouzet controller.
Increment	1
Test environment	**CHP01**
Preconditions	Homematcher PC is running and all connections are active. Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management). Also making changes to control parameters is executed using a DB management tool.
Inputs	Parameters as defined in IDD (Heat system Control Design)
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for that parameter. The changed) Control parameters must be send regularly to the Crouzet controller.
Wrap up	-
Remarks	-

SWI-01-014	
Objectives	Stress testing interface between: - Measurements & Logging (PC) - MicroCHP Crouzet controller (PLC) Measurements and controlling parameter should be updated with the highest possible frequency (at least once every second).
Increment	1
Test environment	**CHP01**
Preconditions	Homematcher PC is running and connection between microCHP device controller (PC) and Crouzet controller (PLC) is active. The Crouzet is monitored using the Crouzet monitoring tooling and the Device Controller GUI. Actual entries made for the control parameters are executed using a database monitoring tool (e.g. SQL server management).
Inputs	parameters as defined in IDD (Heat system Control Design)
Expected result	All variables must be handled correctly and within the defined time interval.
Wrap up	-
Remarks	-

SWI-01-015	
Objectives	Test measurement interface between: - Measurements & Logging (PC) - MicroCHP device agent (PC) Stored measurements (local DB) must be able to be read by the microCHP agent. Check for the actual value and date time stamp.
Increment	1
Test environment	**CHP01**
Preconditions	Homematcher PC is running and connection between microCHP device agent (PC) and Measurements & Logging (PC) is active.
Inputs	Measurement parameters as defined in IDD (Heat system Control Design.
Expected result	All measured values must correctly be retrieved from the local DB.
Wrap up	-
Remarks	-

4.7.1.3 Heat pump

SWI-01-020	
Objectives	Test interface between: - Heat pump (HeatSystem) device controller (PC) - Heat pump Crouzet controller (PLC) All signals defined in the IDD_HS (Heat System Control Design) must be tested for correct functioning. Changes made to Crouzet outputs should be reflected in the Device Controller GUI. Also changing Crouzet inputs (using the Device controller GUI) should be reflected within the Crouzet PLC.
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is running and connection between Heatsystem device controller (PC) and Crouzet controller (PLC) is active. The Crouzet is monitored using the Crouzet monitoring tooling
Inputs	signals as defined in IDD (Heat system Control Design)
Expected result	All signals must be correctly reflected within the counterpart device.
Wrap up	-
Remarks	-

SWI-01-021	
Objectives	Test measurement interface between: - Measurements & Logging (PC) - Heat pump (HeatSystem) device controller (PC) The defined measurement parameters in Measurements & Logging should actually be logged to the local Database. Check next items for every measured parameter: - Measured values - Measurement interval
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is running and connection between heat pump device controller (PC) and Measurements & Logging (PC) is active. The heat pump device controller runs in simulation mode (input of measured data using simulation data file). Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management).
Inputs	Measurement parameters as defined in IDD_HS (Heat system Control Design). Use either simulation file for generating data or life system where values can be set.
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for given parameter.
Wrap up	-
Remarks	-

SWI-01-022	
Objectives	Test controlling interface between: - Measurements & Logging (PC) - Heat pump (HeatSystem) Device controller (PC) The defined control parameters in Measurements & Logging should be reflected on heat pump device controller software (GUI) regarding the control update interval. Check next items for every control parameter: - Control value is read correctly from DB - Reading interval is correct
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is running and connection between heat pump device controller (PC) and Measurements & Logging (PC) is established
Inputs	Control signals as defined in IDD (Heat system Control Design). Control values are changed using a DB management tool.
Expected result	Changed control values should be reflected in the GUI of the heat pump device controller GUI.
Wrap up	-
Remarks	-

SWI-01-023	
Objectives	Test total interface between: - Measurements & Logging (PC) - Heat pump Crouzet controller (PLC) The actual Crouzet measurement parameters should be logged in Measurements & Logging. Changed control parameters should be send to the Crouzet controller.
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is running and all connections are active. Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management). Also making changes to control parameters is executed using a DB management tool.
Inputs	Parameters as defined in IDD (Heat system Control Design)
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for that parameter. The changed) Control parameters must be send regularly to the Crouzet controller.
Wrap up	-
Remarks	-

SWI-01-024	
Objectives	Stress testing interface between: - Measurements & Logging (PC) - Heat pump Crouzet controller (PLC) Measurements and controlling parameter should be updated with the highest possible frequency (at least once every second).
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is running and connection between heat pump device controller (PC) and Crouzet controller (PLC) is active. The Crouzet is monitored using the Crouzet monitoring tooling and the Device Controller GUI. Actual entries made for the control parameters are executed using a database monitoring tool (e.g. SQL server management).
Inputs	parameters as defined in IDD (Heat system Control Design)
Expected result	All variables must be handled correctly and within the defined time interval.
Wrap up	-
Remarks	-

SWI-01-025	
Objectives	Test measurement interface between: - Measurements & Logging (PC) - Heat pump device agent (PC) Stored measurements (local DB) must be able to be read by the heat pump agent. Check for the actual value and date time stamp.
Increment	1
Test environment	**HP01**
Preconditions	Homematcher PC is running and connection between heat pump device agent (PC) and Measurements & Logging (PC) is active.
Inputs	Measurement parameters as defined in IDD (Heat system Control Design.
Expected result	All measured values must correctly be retrieved from the local DB.
Wrap up	-
Remarks	-

4.7.1.4 AMR

SWI-01-030	
Objectives	Test interface between: - AMR device controller (PC) - AMR meter (Actaris, smart meter) All signals available (electrical) on the P1 port of the AMR meter must be tested for correct functioning. Values of the AMR meter should be reflected in the AMR device controller GUI.
Increment	1
Test environment	**AMR01**
Preconditions	Homematcher PC is running and connection between the AMR device controller (PC) and the actual AMR meter (Actaris) is active.
Inputs	Signals as defined in NTA 8130 document (specified for P1 port).
Expected result	All signals must be shown correctly in the AMR device controller GUI.
Wrap up	-
Remarks	-

SWI-01-031	
Objectives	Test measurement interface between: - Measurements & Logging (PC) - AMR device controller (PC) The defined measurement parameters in Measurements & Logging should actually be logged to the local Database. Check next items for every measured parameter: - Measured values - Measurement interval
Increment	1
Test environment	**AMR01**
Preconditions	Homematcher PC is running and connection between AMR device controller (PC) and Measurements & Logging (PC) is active. The AMR device controller runs in simulation mode (input of measured data using simulation data file). Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management).
Inputs	Measurement parameters as defined in NTA 8130 document (specified for P1 port). Use simulation file for generating data.
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for given parameter.
Wrap up	-
Remarks	-

SWI-01-032	
Objectives	Test total interface between: - Measurements & Logging (PC) - AMR meter (smart meter, Actaris) The actual Actaris measurements (electrical) should be logged in Measurements & Logging.
Increment	1
Test environment	**AMR01**
Preconditions	Homematcher PC is running and all connections are active. Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management). Also making changes to control parameters is executed using a DB management tool.
Inputs	Parameters as defined in IDD (Heat system Control Design)
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for that parameter.
Wrap up	-
Remarks	-

4.7.1.5 AMR extended

AMR extended interface handles all connected M-bus devices available within the household. The AMR extended device controller therefore logs data for different available device components (e.g. Heatsystem / PV AMR).

SWI-01-040	
Objectives	Test interface between: - AMR extended device controller (PC) - AMR M-bus meters All measured values available through the M-bus components on the m-bus must be tested for correct functioning. Values of the AMR extended meters should be reflected in the AMR extended device controller GUI.
Increment	1
Test environment	**AMR02**
Preconditions	Homematcher PC is running and connection between the AMR extended device controller (PC) and the actual AMR M-bus meters are active.
Inputs	Signals (measurements) as defined in DB_PARAM_DEF (AMR Extended part)
Expected result	All signals must be shown correctly in the AMR extended device controller GUI.
Wrap up	-
Remarks	-

SWI-01-041	
Objectives	Test measurement interface between: - Measurements & Logging (PC) - AMR extended device controller (PC) The defined measurement parameters in Measurements & Logging should actually be logged to the local Database. Check next items for every measured parameter: - Measured values - Measurement interval
Increment	1
Test environment	**AMR02**
Preconditions	Homematcher PC is running and connection between AMR extended device controller (PC) and Measurements & Logging (PC) is active. The AMR extended device controller runs in simulation mode (input of measured data using simulation data file). Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management).
Inputs	Measurement parameters as defined in DB_PARAM_DEF. Use simulation file for generating data.
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for given parameter.
Wrap up	-
Remarks	-

SWI-01-042	
Objectives	Test total interface between: - Measurements & Logging (PC) - AMR extended meters The actual m-bus measurements should be logged in Measurements & Logging.
Increment	1
Test environment	**AMR02**
Preconditions	Homematcher PC is running and all connections are active. Actual entries in the local storage are verified using a database monitoring tool (e.g. SQL server management). Also making changes to control parameters is executed using a DB management tool.
Inputs	Parameters as defined in DB_PARAM_DEF.
Expected result	All measured values must be correctly stored in the local DB and updated with the defined measurement interval for that parameter.
Wrap up	-
Remarks	-

4.7.1.6 Local and Central storage synchronization

SWI-01-080	
Objectives	Test synchronization between local storage and central storage. Measurements are synchronized to the central server (according selected compacting method).
Increment	1
Test environment	**CS01**
Preconditions	Homematcher PC and central server PC are running and connected.
Inputs	
Expected result	All measured values stored in the local DB are synchronized (and compacted when selected) to the central server.
Wrap up	-
Remarks	-

SWI-01-081	
Objectives	Test synchronization between local storage and central storage. Configuration settings are send from central storage to local storage.
Increment	1
Test environment	**CS01**
Preconditions	Homematcher PC and central server PC are running and connected.
Inputs	
Expected result	Changed configuration parameters must be send to the local storage (Homematcher).
Wrap up	-
Remarks	-

4.7.2 Wind turbine level test cases

SWI-02-011	
Objectives	Test real-time contingency and error messaging
Increment	1
Test environment	**DDW01**
Preconditions	Kreileroord PC-node is connected and running in the network; there is a failure on the turbine or maintenance actions after 15 minutes the failure is repaired
Inputs	Windturbine agent is running in single agent system; auctioneer is operating with a 30 second bidding interval
Expected result	The agent is notified on the error condition; removed from the cluster and added again after reestablishing the connection
Wrap up	-
Remarks	-

4.7.3 Central control level test cases

SWI-06-003	
Objectives	The Commercial Agent decreases the market price when demand decreases.
Increment	1
Test environment	CA01
Preconditions	–
Inputs	Decrease the demand.
Expected result	Market price goes down.
Wrap up	Nothing.
Remarks	–

SWI-06-004	
Objectives	The Agent decreases the market price when supply increases.
Increment	1
Test environment	CA01
Preconditions	–
Inputs	Increase the demand.
Expected result	Market price goes down.
Wrap up	Nothing.
Remarks	–

4.7.4 System level test cases

SWI-04-010	
Objectives	Check congestion management approach
Increment	1
Test environment	**SYS01**
Preconditions	Electricity Storage node, micro-CHP, PV-node and transformer agent node are connected to the cluster based on SWI-03-012 operation during a day; transformer agent performs bids using a very constrained capacity
Inputs	The agents are running in a multi-agent system; the transformer agent bid
Expected result	Operation of devices to diminish the load on the transformer; the ;load should be spread
Wrap up	-
Remarks	-

5 System Integration Test

5.1 Scope of the System Integration Test

The purpose of this test is to verify the correct operation of the system as a whole. The test cases defined for the 'System Integration Test' are only executed when the 'Software Integration Test' is completed and passed. What is expected from the 'System Integration Test' is that from an overall functional point of view, do the sensor values end up correctly in the database and does the system react correctly to given stimuli? To put is shortly: does the whole system technically operate as designed?

5.2 Objective

The objective of this test is to verify correct functioning of the entire system from technical point of view.

5.3 Preconditions

Although not a necessary precondition, it is expected that the test cases are executed in order. This has to do with the fact that the test cases are modeled in such a way that the fill level of the buffers are correct for the next test case, i.e. there is not need to wait for the correct fill level and thus time is saved.

Read the Comfort Control document at:
https://www.integraldc.eu/exec/documentread/beajaieiccdc?id=18882-766f66646f63

5.4 Assumptions

It is assumed that the test cases are executed in order.

5.5 Infrastructure

The infrastructure defined using ADSL-connections to the central database, provided by an external party is used. Tests are performed using the remote desktop facility of Windows.

5.6 Test environment

The test cases are ordered somewhat logically. First the system is put in a known state by emptying the buffer and setting a low market price. Now we can start to gradually fill the buffer and test the behaviour of the system for four different market prices: $c < C_1$, $C_1 < c < C_a$, $C_a < c < C_2$ and $c > C_2$. At the end of this series, the SoC is at its maximum and now we change the market price in reverse order for the same intervals. This ensures that the system does not start to fill the buffer for the intervals. At this point all the changes in market price are covert by the test cases and all that needs to be tested now is the emptying of the buffer. Since the buffer is full, we need to start emptying the buffer at the four intervals, starting at high prices going to the lower ones. Finally there are two test cases that test the stopping and starting of an Agent.

5.7 Test Case Specifications

Constant	Value	Definition
TW_{min}	62°C	Lowest set point of the tap water at which point the Crouzet does not intervene yet.
L_{min}	65°C	
C_1	5	Lower cutoff point. As defined in the micro-CHP configuration file.
C_a	10	Auxiliary cutoff point. As defined in the micro-CHP configuration file.
C_2	40	Upper cutoff point. As defined in the micro-CHP configuration file.
C_1	10	Lower cutoff point. As defined in the Heat Pump configuration file.
C_a	30	Auxiliary cutoff point. As defined in the Heat Pump configuration file.
C_2	40	Upper cutoff point. As defined in the Heat Pump configuration file.
C_{max}	50	Upper limit of the cost price
T_{wu}	30 s	Warm up time of primary burner, i.e. minimal time required to be on after start up.
T_{cd}	180 s	Cool down time of primary burner, i.e. minimal time required to be off after shutdown.

5.7.1 Household/laboratory level test cases

5.7.1.1 Micro-CHP

SI-01-001	
Objectives	Auxiliary burner is switched on while the primary burner stays off when the current market price is below the C_1 point and the requested SoC is higher then the current one.
Increment	1
Test environment	CHP01
Preconditions	If the micro-CHP agent is running, stop it first. Using the interface on the Auctioneer, set the price to 0. Set the fill level of the tap water buffer to its lowest allowed value such that the Crouzet does not take control, i.e. TW_{min}. Verify that neither burner is switched on. When the primary burner was switched off, wait at least T_{cd} seconds before making any changes to the system.
Inputs	Using the interface on the Auctioneer, set the price to 3, i.e. $c < C_1$ and start the micro-CHP Agent.
Expected result	The auxiliary burner is switched on, while the primary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-002	
Objectives	Auxiliary burner is switched on while the primary burner stays off when the current market price is between C_1 and C_a and the requested SoC is higher then the current one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-001* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 8, i.e. $C_1 < c < C_a$.
Expected result	The auxiliary burner is switched on, while the primary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-003	
Objectives	Primary burner is switched on while the auxiliary burner stays off when the current market price is above C_a and the requested SoC is higher then the current one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-002* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 30.
Expected result	The primary burner is switched on while the auxiliary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system and wait at least T_{cd} seconds before making any changes to the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-004	
Objectives	Primary burner is switch on while the auxiliary burner stays off when the current market price is above C_2 and the requested SoC is higher then the current one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-003* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 45, i.e. $c > C_2$.
Expected result	The primary burner is switched on while the auxiliary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system and wait at least T_{cd} seconds before making any changes to the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-005	
Objectives	Both the primary and auxiliary burners stay off when the current market price is above C_2 and the requested SoC is below the current one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-004* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 42, i.e. $c > C_2$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is.
Remarks	–

SI-01-006	
Objectives	Both the primary and auxiliary burners stay off when the current market price is between C_a and C_2 and the requested SoC is below the current one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-005* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 30, i.e. $C_a < c < C_2$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is.
Remarks	–

SI-01-007	
Objectives	Primary and auxiliary burners stay off when the current market price is between C_1 and C_a and the requested SoC is below the current one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-006* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 8, i.e. $C_1 < c < C_a$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is.
Remarks	–

SI-01-008	
Objectives	Primary and auxiliary burners stay off when the current market price is below C_1 and the requested SoC is below the current one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-007* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 3, i.e. $c < C_1$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is.
Remarks	–

SI-01-009	
Objectives	Primary is switched on while the auxiliary burner stays off when the current market price is above C_2 while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-008* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 48, i.e. $c > C_2$.
Expected result	As the current SoC drops below the desired SoC, the primary burner is switched on until the desired SoC is reached.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system and wait at least T_{cd} seconds before making any changes to the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-010	
Objectives	Primary burner is switched on while the auxiliary burner stays off when the current market price is between C_a and C_2 while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-009* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 15, i.e. $C_a < c < C_2$.
Expected result	As the current SoC drops below the desired SoC, only the primary burner is switched on until the desired SoC is reached.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-011	
Objectives	Auxiliary burner is switched on while the primary burner stays off when the current market price is between C_1 and C_a while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-010* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 6, i.e. $C_1 < c < C_a$.
Expected result	As the current SoC drops below the desired SoC, only the auxiliary burner is switched on until the desired SoC is reached.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-012	
Objectives	Auxiliary burner is switched on while the primary burner stays off when the current market price is below C_1 while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-010* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 2, i.e. $c < C_1$.
Expected result	As the current SoC drops below the desired SoC, only the auxiliary burner is switched on until the desired SoC is reached.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-01-013	
Objectives	Crouzet takes over control in case the agent is killed.
Increment	1
Test environment	CHP01
Preconditions	–
Inputs	Kill the agent.
Expected result	The Crouzet takes control and the system continuous to operate.
Wrap up	Leave the system as it is.
Remarks	–

SI-01-014	
Objectives	Agent takes over control in case the agent is restarted.
Increment	1
Test environment	CHP01
Preconditions	Take the end state of test case *SI-01-013* as starting point for this test case.
Inputs	Start the micro-CHP agent.
Expected result	The agent takes control and the system continuous to operate.
Wrap up	Nothing.
Remarks	–

5.7.1.2 Heat pump

SI-02-001	
Objectives	Primary burner is switched on while the auxiliary burner stays off when the current market price is above the C_2 point and the requested SoC is higher then the current one.
Increment	1
Test environment	HP01
Preconditions	If the Heat Pump agent is running, stop it first. Using the interface on the Auctioneer, set the price to C_{max}. Set the fill level of the tap water buffer to its lowest allowed value such that the Crouzet does not take control, i.e. TW_{min}. Verify that neither burner is switched on. When the primary burner was switched off, wait at least T_{cd} seconds before making any changes to the system.
Inputs	Using the interface on the Auctioneer, set the price to 45, i.e. $c > C_2$ and start the Heat pump Agent.
Expected result	The auxiliary burner is switched on, while the primary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-002	
Objectives	Auxiliary burner is switched on while the primary burner stays off when the current market price is between C_a and C_2 point and the desired SoC is higher then the current one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-001* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 35, i.e. $C_a < c < C_2$.
Expected result	The auxiliary burner is switched on, while the primary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-003	
Objectives	Primary burner is switched on while the auxiliary burner stays off when the current market price is between C_1 and C_a and the desired SoC is higher then the current one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-002* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 20, i.e. $C_1 < c < C_a$.
Expected result	The primary burner is switched on, while the auxiliary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system and wait at least T_{cd} seconds before making any changes to the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-004	
Objectives	Primary burner is switched on while the auxiliary burner stays off when the current market price for below C_1 and the desired SoC is higher then the current one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-003* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 4, i.e. $c < C_1$.
Expected result	The primary burner is switched on, while the auxiliary burner stays off.
Wrap up	Wait for the system to reach the determined SoC, i.e. both burners will be tuned off by the system and wait at least T_{cd} seconds before making any changes to the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-005	
Objectives	Both burners are stay off when the current market price is below C_1 and the desired SoC is below the current one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-004* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 2, i.e. $c < C_1$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-006	
Objectives	Both burners are stay off when the current market price is between C_1 and C_a and the desired SoC is below the current one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-005* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 25, i.e. $C_1 < c < C_a$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-007	
Objectives	Both burners are stay off when the current market price is between C_a and C_2 and the desired SoC is below the current one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-006* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 33, i.e. $C_a < c < C_2$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-008	
Objectives	Both burners are stay off when the current market price is above C_2 and the desired SoC is below the current one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-007* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 47, i.e. $c < C_2$.
Expected result	Both the primary and auxiliary burners stay off.
Wrap up	Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-009	
Objectives	Primary burner is switched on while the auxiliary burner stays off when the current market price is below C_1, while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-008* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 6, i.e. $c < C_1$.
Expected result	As the current SoC drop below the desired SoC, only the primary burner is switched on un till the desired SoC is reached.
Wrap up	Wait for the system to reach the desired SoC, i.e. both burners will be turned off by the system and then wait for at least T_{cd} seconds before making any changes to the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-010	
Objectives	Primary burner is switched on while the auxiliary burner stays off when the current market price is between C_1 and C_a, while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-009* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 18, i.e. $C_1 < c < C_a$.
Expected result	As the current SoC drop below the desired SoC, only the primary burner is switched on un till the desired SoC is reached.
Wrap up	Wait for the system to reach the desired SoC, i.e. both burners will be turned off by the system and then wait for at least T_{cd} seconds before making any changes to the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-011	
Objectives	Auxiliary burner is switched on while the primary burner stays off when the current market price is between C_a and C_2, while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-010* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 32, i.e. $C_a < c < C_2$.
Expected result	As the current SoC drop below the desired SoC, only the auxiliary burner is switched on un till the desired SoC is reached.
Wrap up	Wait for the system to reach the desired SoC, i.e. both burners will be turned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-012	
Objectives	Auxiliary burner is switched on while the primary burner stays off when the current market price is above C_2, while the buffer is emptied, i.e. the current SoC is decreased to a point below the desired one.
Increment	1
Test environment	HP01
Preconditions	Take the end stat of test case *SI-02-011* as starting point for this test case.
Inputs	Using the interface on the Auctioneer, set the price to 42, i.e. $c > C_2$.
Expected result	As the current SoC drop below the desired SoC, only the auxiliary burner is switched on un till the desired SoC is reached.
Wrap up	Wait for the system to reach the desired SoC, i.e. both burners will be turned off by the system. Leave the system as it is as it can be used for the next test case.
Remarks	–

SI-02-013	
Objectives	Crouzet takes over control in case the agent is killed.
Increment	1
Test environment	HP01
Preconditions	–
Inputs	Kill the agent.
Expected result	The Crouzet takes control and the system continuous to operate.
Wrap up	Leave the system as it is.
Remarks	–

SI-02-014	
Objectives	Agent takes over control in case the agent is restarted.
Increment	1
Test environment	HP01
Preconditions	Take the end state of test case *SI-02-013* as starting point for this test case.
Inputs	Start the Heat Pump agent.
Expected result	The agent takes control and the system continuous to operate.
Wrap up	Nothing.
Remarks	–

5.7.2 Battery level test cases

SI-03-001	
Objectives	The Prius batteries start charging some time after the Prius is plugged in and the batteries are not full.
Increment	1
Test environment	BAT01
Preconditions	The batteries of the Prius are not full.
Inputs	Plug in the Prius
Expected result	After sometime the Prius stats to charge.
Wrap up	Nothing.
Remarks	–

SI-03-002	
Objectives	The Agent takes the balancing time of the batteries into account
Increment	1
Test environment	BAT01
Preconditions	The Prius is charging.
Inputs	–
Expected result	When the batteries have been charge the waiting time for the balancing of the batteries are observed.
Wrap up	Nothing.
Remarks	–

SI-03-003	
Objectives	The Agent does not want to charge the batteries when the Prius is not plugged in to the grid.
Increment	1
Test environment	BAT01
Preconditions	The Prius is unplugged.
Inputs	–
Expected result	The Agent does not want to charge the batteries.
Wrap up	Nothing.
Remarks	–

5.7.3 Wind turbine level test cases

SI-04-001	
Objectives	The Agent matches supply and demand
Increment	1
Test environment	WIND01
Preconditions	There is a demand for power.
Inputs	–
Expected result	The Agent matches the demand by offering the total production at the lowest price
Wrap up	Nothing.
Remarks	–

SWI-02-010	
Objectives	Test real-time realization output of wind turbine
Increment	1
Test environment	**CS01**
Preconditions	Kreileroord PC-node is connected and running in the network; Wind agent bids all output power at lowest price
Inputs	Windturbine agent is running in single agent system; auctioneer is operating with a 30 second bidding interval
Expected result	A sequence of bids leads to a sequence of power outputs; this has to correspond with the database stored value
Wrap up	-
Remarks	-

5.7.4 Photo Voltaic level test cases

SI-05-001	
Objectives	The Agent matches supply and demand
Increment	1
Test environment	PV01
Preconditions	There is a demand for power.
Inputs	–
Expected result	The Agent matches the demand by offering the total production at the lowest price level.
Wrap up	Nothing.
Remarks	–

5.7.5 Commercial Aggregator level test cases

SI-06-001	
Objectives	The Agent increases the market price when demand increases.
Increment	1
Test environment	CA01
Preconditions	–
Inputs	Increase the demand.
Expected result	Market price goes up.
Wrap up	Nothing.
Remarks	–

SI-06-002	
Objectives	The Agent increases the market price when supply decreases.
Increment	1
Test environment	CA01
Preconditions	–
Inputs	Increase the demand.
Expected result	Market price goes up.
Wrap up	Nothing.
Remarks	–

SI-03-010	
Objectives	Test incorporation of wind expected production from wind production forecasts
Increment	1
Test environment	**COM01**
Preconditions	Commercial aggregator PC-node and wind parkPC-node are connected and running in the network; the system processing the HIRLAM-meteorological model to local terrain specific predictions is connected
Inputs	Commercial agent and wind agent are running in a single agent system
Expected result	The CA-agent will issue a bid as a seller or a buyer depending on the difference between the expected production and the realized production derived from the current aggregated bid only from the windpark
Wrap up	-
Remarks	-

SI-03-011	
Objectives	Test incorporation of solar expected production from solar radiation production forecasts
Increment	1
Test environment	**COM01**
Preconditions	Commercial aggregator PC-node and solar PC-node are connected and running in the network; the system processing the Meteoconsult-meteorological model to local terrain coordinates is connected
Inputs	Commercial agent and PV agent are running in a single agent system
Expected result	The CA-agent will issue a bid as a seller or a buyer depending on the difference between the expected production and the realized production derived from the current aggregated bid from the PV
Wrap up	-
Remarks	-

SI-03-012	
Objectives	Get an estimate of the expected internal electricity price in the cluster and check with the expected price development (e.g. from ARIMA-model)
Increment	1
Test environment	**COM01**
Preconditions	Initial pattern for the ARIMA-model known, Electricity Storage node, micro-CHP node with a heat demand pattern, PV-node with a radiation production pattern and household agent node with a lighting demand pattern are connected to the cluster.
Inputs	The agents are running in a multi-agent system and exchange their predictions
Expected result	A composite pattern of the expected price development to be used to optimize buffer charging
Wrap up	-
Remarks	-

SI-03-013	
Objectives	Inspect control capability to a power demand or supply push situation from the trade-floor
Increment	1
Test environment	**COM01**
Preconditions	Electricity Storage node, micro-CHP, PV-node and household agent node are connected to the cluster based on SWI-03-012 operation during a day; trade floor agent adds sequence of bids with alternating for supply and demand
Inputs	The agents are running in a multi-agent system; the aggregated bidding curve is available
Expected result	A composite pattern of the expected price development
Wrap up	-
Remarks	-

5.7.6 DSO level test cases

SI-07-001	
Objectives	The Agent increases the market price when load on the substation is too high in order to reduce its load.
Increment	1
Test environment	CA01
Preconditions	A balanced substation.
Inputs	Increase load on the substation.
Expected result	Market price goes up and the load is decreased within tolerance.
Wrap up	Nothing.
Remarks	–

5.7.7 Central control level test cases

These are incorporated in the use cases from the field test

5.7.8 System level test cases

See 5.7.5 and 5.7.6. Note the AMR-infrastructure has to be in place in order to verify operation at the system level. The first successive use case is optimization behind the meter.