D6.6.32 Assessment of Energy Management System

Assessment of Energy Management System

Revision History

Edition	Date	Status	Editor
v0.01	16.2.2015	Created	Pekka Savolainen
v0.5	18.2.2015	Added Apros models	Kari Mäki
v1.0	24.2.2015	Finalized document	Pekka Savolainen

Authors: Pekka Savolainen (VTT), Kari Mäki (VTT)

Abstract

This deliverable describes the work done in SGEM-project Phase 5, also known as funding period five (FP5). The work done in FP5 is a direct continuation to the work in SGEM funding periods two (FP2), three (FP3), and four (FP4). In deliverable D6.12.7 "*Survey Report on KPIs for the Control and Management of Smart Grids*" [1], written in FP2, an idea of a microgrid with an Energy Management System (EMS) was introduced and defined. Deliverable D6.6.21 "*A Prototype of an Intelligent Decision-making Component for Smart Grids*" [2], written in FP3, contains the description for a Simulink/Matlab simulation model for the microgrid's Energy Management System (EMS) and a microgrid model containing various generation, storage and load devices. In the fourth funding period, work consisted of defining a completely new microgrid simulation platform, which would allow us to test different EMS algorithms in a much more realistic model than what was possible with the Simulink microgrid model implemented in FP3. This work was documented in deliverable D6.6.31 "Enhanced prototype of EMS decision-making component" [3].

In the fifth funding period work concentrated on building the new simulation platform. It contains Apros models for a microgrid and a Microgrid Controller (MGC) and also an Energy Management System (EMS) program implemented in Python and C++. The Apros models and the EMS program communicate using OPC UA, an interoperability standard for the secure and reliable exchange of data in industrial automation. The new platform combines VTT's expertise from electrical and telecommunications areas and it was developed in co-operation with SGEM task 5.1. The completed work made as part of WP6.6 is described in this deliverable.

Table of Contents

1 Preface

This report was written as a part of the Finnish national research project "Smart Grid and Energy Market" SGEM Phase 5 and it was funded by Tekes – the Finnish Funding Agency for Technology and Innovation and the project partners.

The authors would like to thank colleagues, CLEEN Oy, SGEM project partners, and SGEM project management for the ideas, support, and advice on tackling the challenges encountered in shaping the future of electric networks.

2 Introduction

This deliverable describes the work done in SGEM-project Phase 5, also known as funding period five (FP5). The work done in FP5 is a direct continuation to the work in SGEM funding periods two (FP2), three (FP3) and four (FP4). In deliverable D6.12.7 "*Survey Report on KPIs for the Control and Management of Smart Grids*" [1], written in FP2, introduced and defined an idea for a microgrid with an Energy Management System (EMS). Deliverable D6.6.21 "*A Prototype of an Intelligent Decision-making Component for Smart Grids*" [2], written in FP3, contains the description for a Simulink/Matlab simulation model for the microgrid's Energy Management System (EMS) and a microgrid model containing various generation, storage and load devices. Deliverable D6.6.31 "*Enhanced prototype of EMS decision-making component*" [3], defined a completely new microgrid simulation platform, which would allow us to test different EMS algorithms in a much more realistic microgrid simulation model than what was possible with the Simulink microgrid model implemented in FP3.

In FP5, the work has concentrated on implementing the microgrid simulation platform defined in FP4 deliverable [3]. The new simulation platform combines VTT's expertise from electrical and telecommunications side and it was developed in co-operation with SGEM task 5.1. This deliverable documents the implementation details of the microgrid simulation platform, which includes an Apros microgrid simulation model with distributed generation (DG) components (diesel generator and PV unit), demand components (electric vehicles (EV) and controllable and non-controllable loads), power storage component, and a microgrid controller block. We also developed an Energy Management System (EMS) that is a separate program for managing the microgrid simulation models' higher level functions. The microgrid simulation model and components were implemented with Apros, which is an electricity network and component simulation tool designed by VTT and the Energy Management System was implemented with Python. The OPC UA interface used in the data exchange between the Apros software and the Python software was implemented in C++.

This deliverable is organized in the following manner. Chapter 3 introduces the scenario that the microgrid simulation platform is simulating. Chapter 4 describes the microgrid simulation platform. Chapter 5 explains the Apros microgrid and controller simulation models. Chapter 6 describes the features of the EMS program. Chapter 7 describes the data exchange between components and finally conclusions in chapter 8 conclude the deliverable.

3 Microgrid Scenario

The studied scenario consists of a microgrid, which is a low voltage (LV) distribution network, with a secondary substation and a medium voltage (MV) network connection. It is assumed that the state of the MV network remains stable. The microgrid includes various components, which affect the network status and can be controlled. These components include DG generation components (a diesel generator and a PV unit), electric vehicles, a storage unit, and both controllable and non-controllable loads. A breaker switch resides at the Point of Common-Coupling (PCC) to connect the LV network to the MV network. The breaker is on the LV side of the transformer and it can be controlled to run the microgrid connected to the MV network or disconnected for islanded operation mode.

The microgrid contains a Microgrid Controller (MGC), which has the responsibility to make sure that individual devices in the microgrid and the microgrid as a whole operates smoothly. It does this by controlling the voltage levels. In the future, also frequency control will be integrated to the MGC. An Energy Management System (EMS) controls the power balance characteristics of the microgrid. EMS features include:

- Setting the microgrid operational mode (islanded or connected to MV)
- Connecting/disconnecting devices in the microgrid
- Selecting the amount of power generated by the DG components
- Selecting the amount of power used by the controllable loads (demand response)
- Setting the duration of load control
- Setting electric vehicle and storage status (charge, idle or discharge)
- Setting EV and storage (dis)charge power
- Setting the duration of dis(charge) power (e.g. charge EV1 4h @ 6kW)
- Calculation of power sold to or bought from the DSO
- Calculation of microgrid demand/generation balance

The EMS does not control network devices directly but sends all control signals to the MGC, which relays them to the network devices. MGC checks the signals, adjusts them according to network status and relays them to device blocks. In normal operation, the network status causes no constraints and MGC simply forwards the EMS signals to the device blocks but sometimes it is necessary to adjust the control signals in case microgrid voltage levels are not proper. After the MGC has adjusted the control signals, it sends the adjusted signal to the device block and the same signal back to the EMS as feedback to indicate the control that actually went to the device block. Figure 1 depicts the simulation scenario. The microgrid contains 2 EVs, 1 storage, 1 PV unit, 1 diesel generator, 5 controllable loads, 5 non-controllable loads and the MGC.

SGEM WP5.1 deliverable D5.1.46 [4], lists use cases for microgrid simulations. The use cases were divided into two categories: Control & Management and Business use cases. The microgrid simulation platform presented in this deliverable is capable of simulating the following use cases:
Control & Management Use Cases:

- Balancing supply and demand on different time-scales
- Demand-side management
- Supply-side management
- Storage management
- Planning DG and storage resources depending on the size of the microgrid
- Island operation (when frequency control is integrated to MGC)

Business Use Cases:

- Microgrid operator sells/buys energy on external market

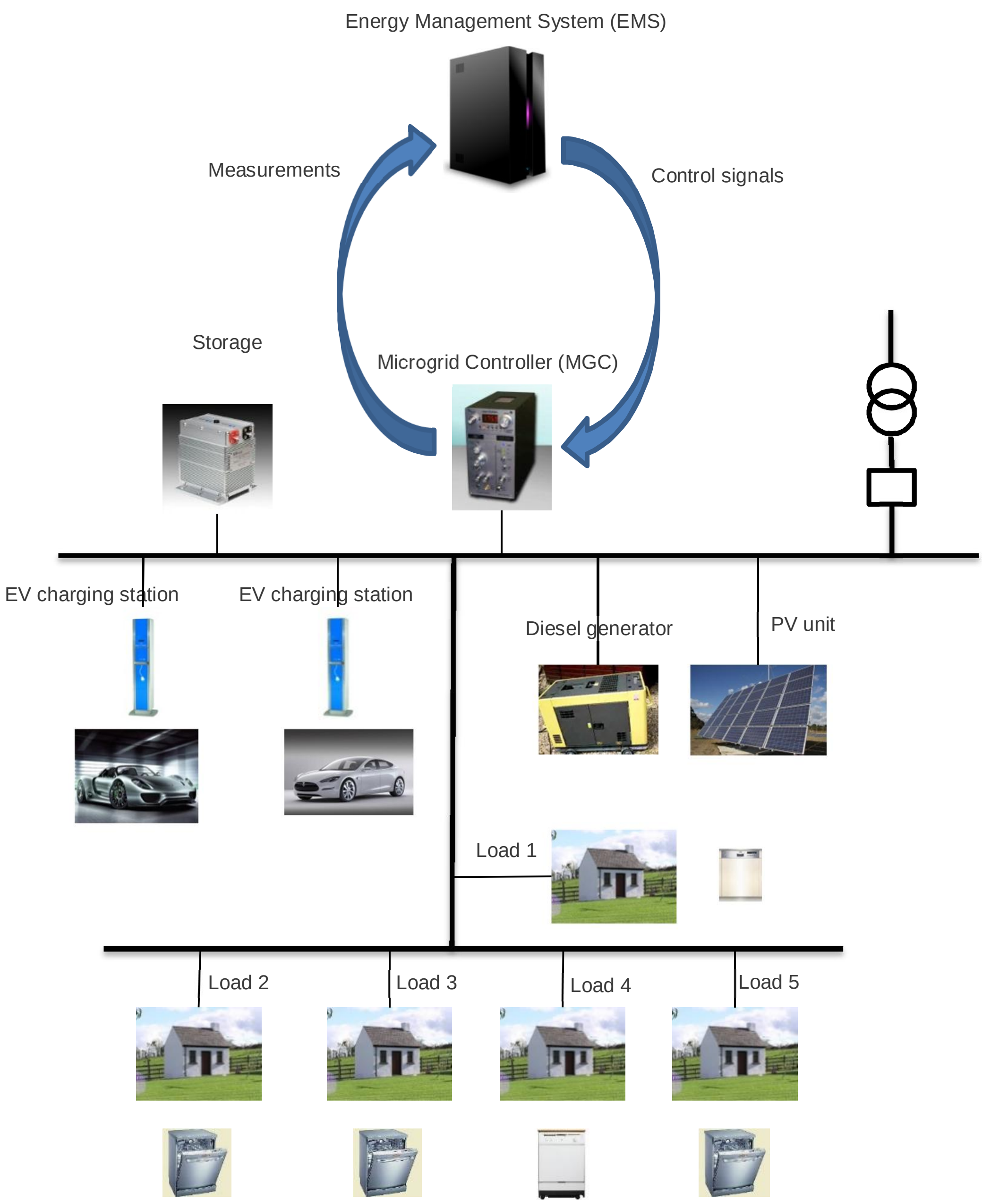

Figure 1. Simulation scenario

4 Microgrid Simulation Platform

The microgrid simulation platform consists of a microgrid model, a microgrid controller, an energy management system (EMS), and an OPC UA client-server based data exchange scheme. The microgrid model and controller have been implemented with Apros. The Energy Management System is a Python program, which includes a Graphical User Interface (GUI) designed with the PyQt4 library. Figure 2 depicts the simulation platform, which consists of two host computers; Host A and Host B. Host B is a virtual machine running Windows 7 OS, capable of running the 64-bit version of the Apros software. Host A can be any Windows 7 computer e.g. a laptop. Host A is running an Energy Management System process and Host B is running an Apros process, which includes the microgrid simulation model i.e. the microgrid model and the microgrid controller. The microgrid model and the microgrid controller have been designed so that there is a clear interface between them. This means that it is possible to separate the two quite easily into their own distinct Apros models. This way, we could use a third host computer that runs the microgrid controller in its own Apros process, while the microgrid model runs on another host in a separate Apros process. However, in this implementation, the microgrid model and controller are running on the same host and in the same Apros process.

Hosts A and B are exchange data via an industrial M2M client-server based communication protocol, OPC UA. It is a successor to OPC, which is an open standard for data transmissions, used frequently in industrial automation to integrate control and diagnostic applications to process devices [5]. OPC UA (OLE for Process Control – Unified Architecture) uses a service-oriented architecture for process control and offers a more secure, configurable, and scalable protocol than the original OPC.

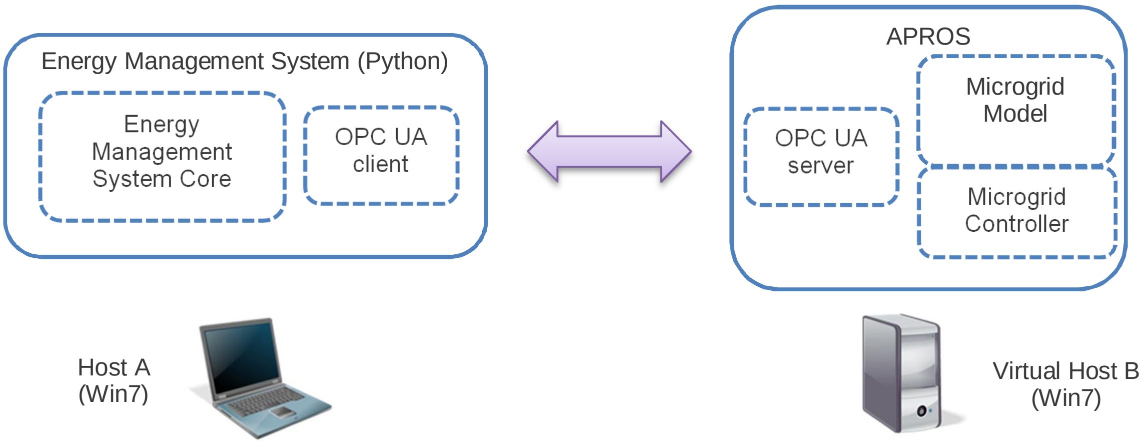

Figure 2. Microgrid Simulation Platform

In the microgrid simulation platform, the hosts are using OPC UA to realize connectivity. The latest releases of Apros include the OPC UA server so hosts A and B can be connected if the Python program implements the OPC UA client interface. We used a C++ implementation of the OPC UA library, which has been integrated to the Python program in order to connect to the Apros server, subscribe to events, monitor variables and write values from the Python program into Apros. With OPC UA, the communication is realized in a way that the OPC UA server can create signals or events that are sent periodically or instantaneously when they have changed. Other OPC UA servers or clients can then monitor variables they are interested in. OPC UA client can also create its own signals and transmit them to the OPC UA server.

Figure 3 depicts the data communication between the EMS, the microgrid controller and the microgrid model. The EMS sends control messages to the MGC, which then relays them to the microgrid model. The EMS receives feedback messages from the MG controller and measurement messages from the microgrid model. When the MGC receives control messages from EMS, it adjusts them if needed and then relays them to the microgrid model. Relaying all control messages to microgrid model via the MGC is done in order to simulate a more realistic microgrid use case, where the EMS has a communication link to the MGC but has no link to individual components. This way, the EMS can control the microgrid remotely from anywhere. MGC has the responsibility to control individual components and make sure the microgrid is operating as expected. MGC receives measurements directly from the MG components. MG components in the microgrid model receive control messages from the MGC. Microgrid model sends measurements (e.g. current generated power) and status messages to both MGC and EMS. The measurements from the microgrid model to the EMS could be relayed through the MGC but in this implementation, we have a direct uplink from the microgrid model to the EMS but no downlink. All messages between the EMS and the MGC are exchanged with the OPC UA protocol. Also measurements from microgrid model to the EMS are sent with the OPC UA protocol. In addition, the EMS reads initialization data from a configuration file when the simulation is started.

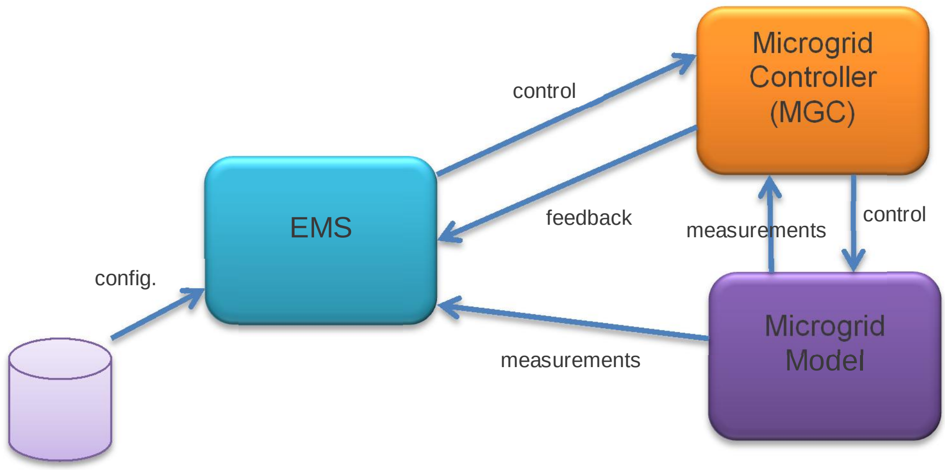

Figure 3. Data communication between the EMS, the MGC and the microgrid model

The Energy Management System concentrates on the energy balance of the microgrid from the economical point of view. The Microgrid controller makes quick decisions to keep the microgrid running. It communicates directly with the microgrid components. The two algorithms make separate decisions. It can be thought, that the EMS makes suggestions and the MG controller decides if those suggestions can be executed without harming or damaging the microgrid. The following chapter presents the Apros models for the Microgrid Controller and microgrid. The EMS is presented in chapter 6.

5 Apros Microgrid and Microgrid Controller Models

The case studied includes a low voltage (LV) distribution network with secondary substation and medium voltage (MV) network connection feeding it. It is assumed that the situation remains stable on the MV side. The LV level includes various components, which affect the network status and can also be controlled. These components include storage unit, EV charging, controllable loads, diesel generator and PV unit. The breaker on the LV side of the transformer can be controlled so that the network could be run as an island.

The control is based on measuring values from LV busbar, making control decisions upon them and distributing the controls across active components located on the LV level. Currently, the system is intended for control actions in grid-connected mode only; thus it uses voltage measurements for control. Voltage control is based on a simple droop-based control, which affects the power balance of the LV network. A further development will be to include frequency control, which enables islanded operation mode.

In the Apros simulation environment, measurement and control signals can be transferred via emulated communication channels. This enables studying the impacts of delays and bandwidth on measurement-control loops. While integration of communication emulation has been developed in SGEM WP5, the model presented here can be extended to communication-related studies. The scenario behind the model was presented in Figure 1. Figure 4 presents the overall configuration of the model in Apros.

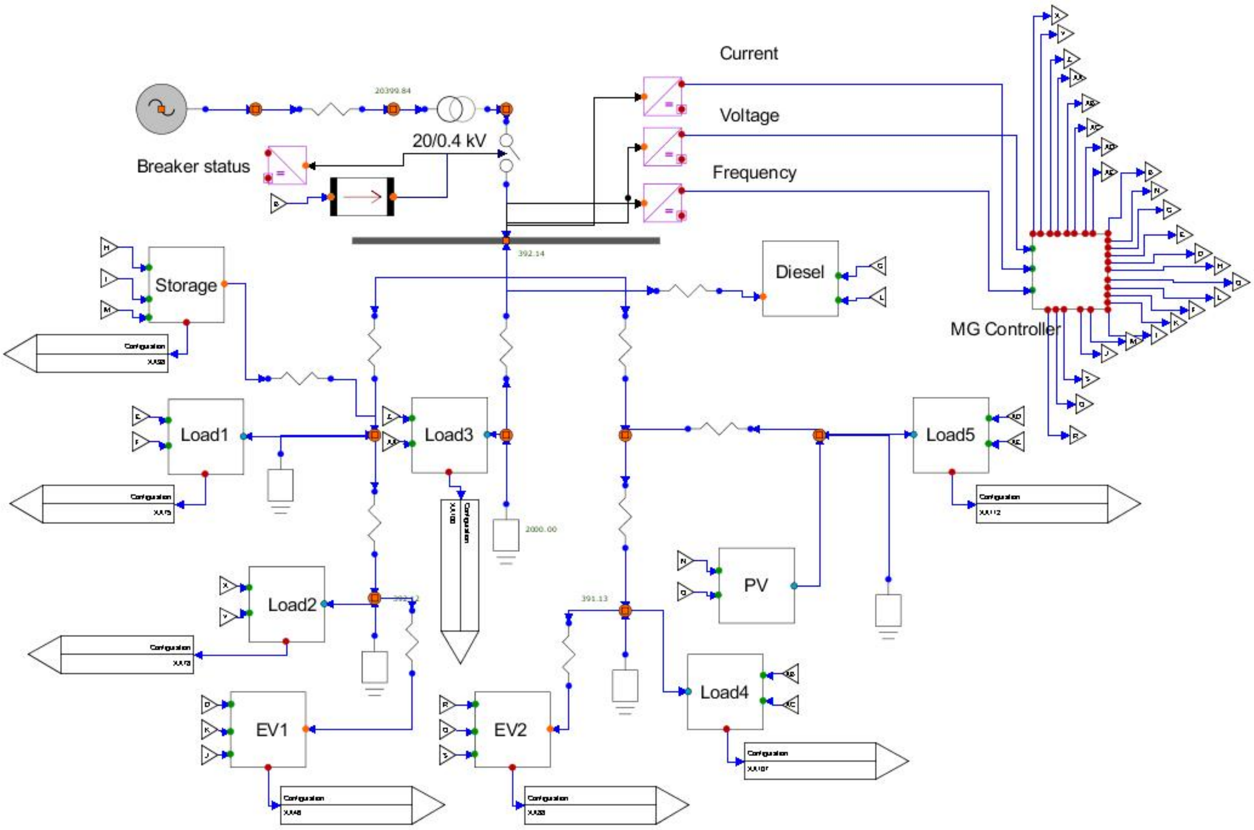

Figure 4. Model layout in Apros

The layout presented in Figure 4 shows the main characteristics of the model. A 20 kV medium voltage network is modelled as a source including source background impedance. Distribution transformer (20 kV / 400 V) is included. Controllable breaker is located between the transformer and LV busbar. Component models included are: 1 diesel generator, 1 storage unit, 1 PV unit, 2 controllable EV charging units, 5 controllable loads and 5 static loads. These models are described with more details in the following subchapters.

Measurements are conducted on LV busbar level, including voltage, frequency and current values. These measurements are used as an input for the MG Controller (MGC) block. MGC also takes control signals provided by the EMS system as an input. MGC overrides EMS signals in case network status requires such actions. Where network status allows controls, MGC simply forwards the controls proposed by EMS.

The signals provided by MGC (shown as small triangles) are used as an input to component models. At this point, these signals can be transferred via emulated communication channels as discussed.

5.1 Controllable Load

The controllable load block is based on a normal Apros load model, which can be adjusted with an external signal. The block has also a switch that can be used to indicate the presence of load momentarily. The controllable load block can thus be adjusted in two ways; by adjusting the setpoint of active power or by disconnecting/connecting the load with fixed power consumption.

The implementation of the block is simple; as inputs it takes control of the switch and active power setpoint. As measured outputs it provides measured power, status of the switch and controllability of load. Controllability of load is deduced based on the switch position and current power; load must be connected and it must have some control capacity in order to mark controllability as true. Figure 5 presents the Apros simulation model of the controllable load.

Table 1. Controllable load I/O signals (Apros)

Signal	Direction	Format
controlLoad	Input	Int
powerSetLoad	Input	Real [W]
powerLoad	Output	Real [W]
statusLoad	Output	Int
controllabilityLoad	Output	Int

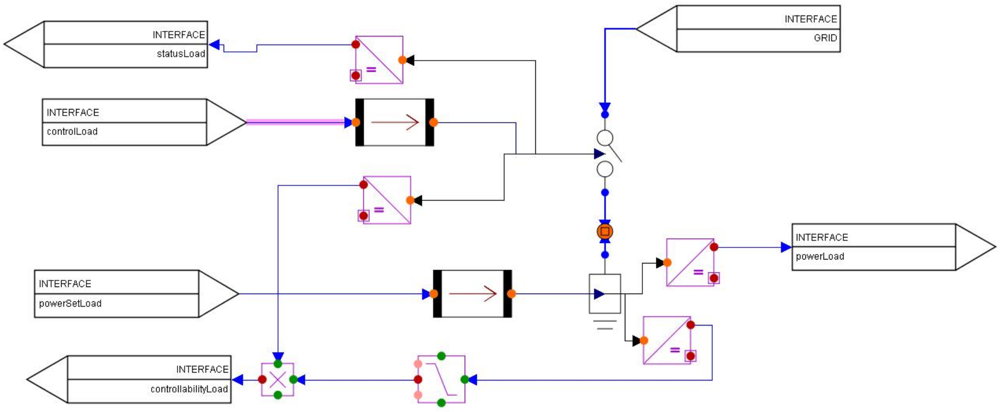

Figure 5. Layout of controllable load block

5.2 Storage Unit and EV Charging Blocks

The storage unit block combines Apros models for battery and DC side equipment with adjustable load components on the AC side. As the Apros inverter models are difficult to run over longer time periods, the storage unit block is implemented as separate DC and AC side models. AC side resembles the charging and discharging towards grid, whereas the DC side fully simulates the battery charging/discharging and related battery variables such as state of charge (SOC).

The system takes as inputs charging control (charge/idle/discharge), charging power setpoint and control of unit switch. As measured outputs it provides current operation status, measured power, SOC, battery capacity, status of the unit switch and controllability. Controllability is deduced based on the switch position, battery SOC and current power.

Storage unit and EV charging systems are implemented as similar blocks.

Table 2. Storage & EV I/O signals (Apros)

Signal	Direction	Format
controlStorage	Input	Int
powerSetStorage	Input	Real [W]
presenceSetStorage	Input	Int
powerStorage	Output	Real [W]
presenceStorage	Output	Int
statusStorage	Output	Int
socStorage	Output	Real [%]
capacityStorage	Output	Real [Ah]

controllabilityStorage	Output	Int

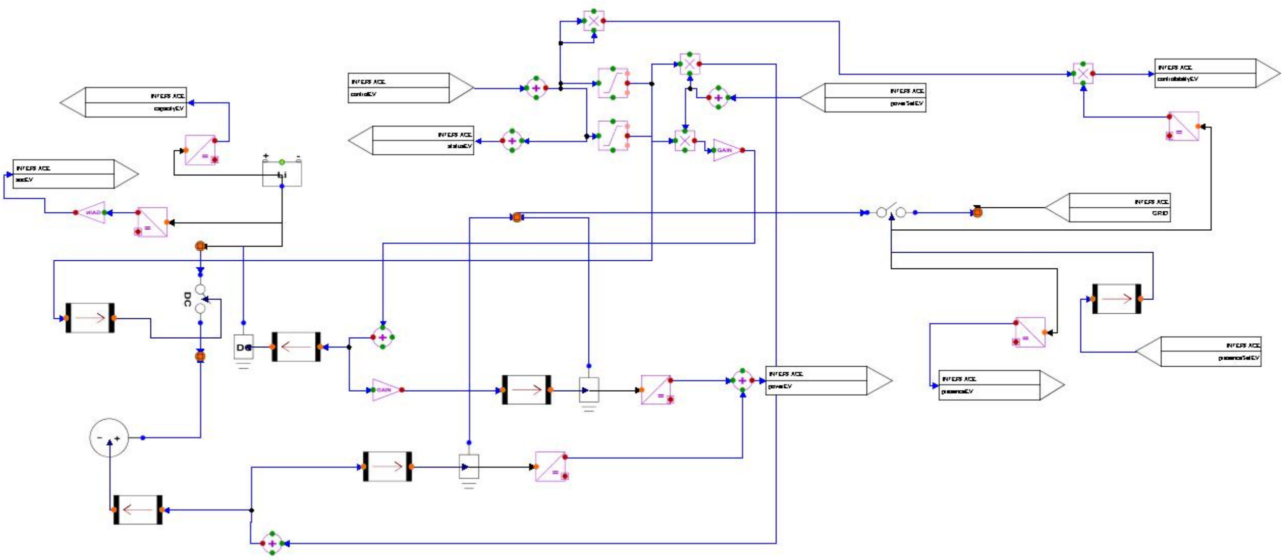

Figure 6. Layout of EV/Storage charging block

5.3 PV Unit

PV unit block utilizes Apros models for solar radiation, PV panel output, maximum power point tracking (MPPT), DC-DC converter and DC-AC inverter. As a result, the model represents the chain from radiation to actual power fed in to distribution grid. The controls are mainly on DC-AC inverter level, where output power curtailment and power factor can be adjusted. PV system is connected to the grid via a controllable switch.

The system takes as inputs active power maximum setpoint, power factor and control of PV unit switch. As measured outputs, it provides measured PV power, used power factor setting, and status of the switch. Apros generates solar radiation data for the PV model according to the given parameters. These parameters include e.g. latitude, longitude, elevation and solar constant, so the PV model can be used to simulate generated power anywhere on the globe.

Table 3. PV unit I/O signals (Apros)

Signal	Direction	Format
controlPV	Input	Int
powerSetPV	Input	Real [W]
powerFactorSetPV	Input	Real [0…1]
powerFactorPV	Output	Real [0…1]
powerPV	Output	Real [W]
statusPV	Output	Int

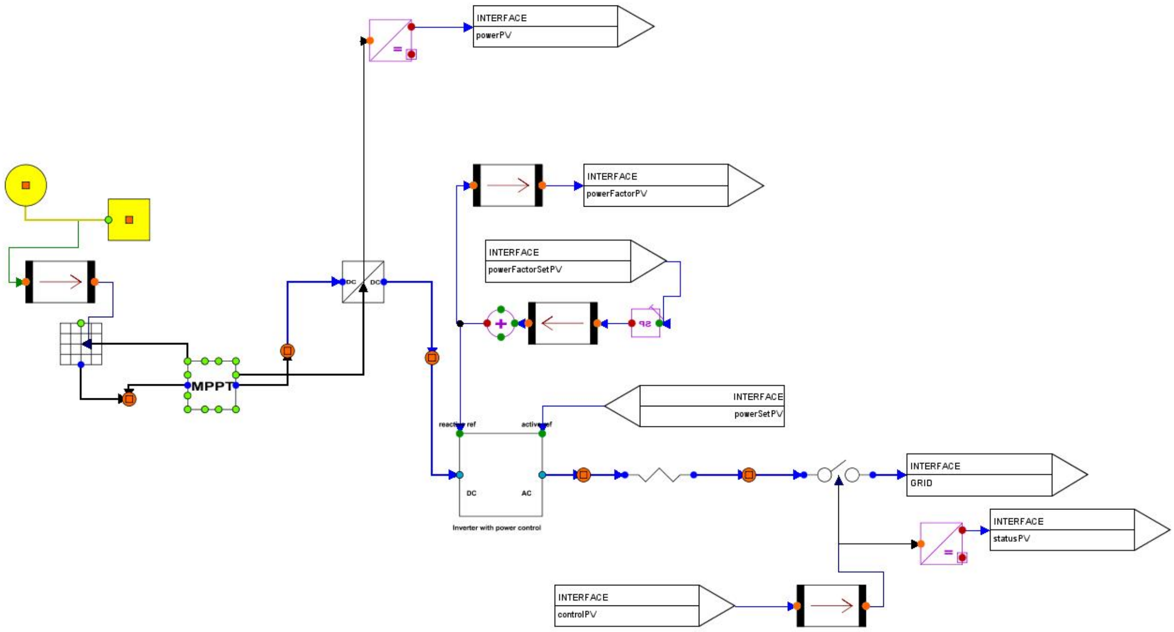

Figure 7. Layout of PV unit

5.4 Diesel Generator

Diesel generator is simply modelled as a standard generator with power control possibilities. The generated active output power as well as the power factor can be controlled.

The system takes as inputs, active power maximum setpoint, power factor and control of unit switch. As measured outputs, it provides measured output power, used power factor setting and status of the switch.

Table 4. Diesel generator I/O signals (Apros)

Signal	Direction	Format
controlGen	Input	Int
powerSetGen	Input	Real [W]
powerFactorSetGen	Input	Real [0…1]
powerFactorGen	Output	Real [0...1]
powerGen	Output	Real [W]
statusGen	Output	Int

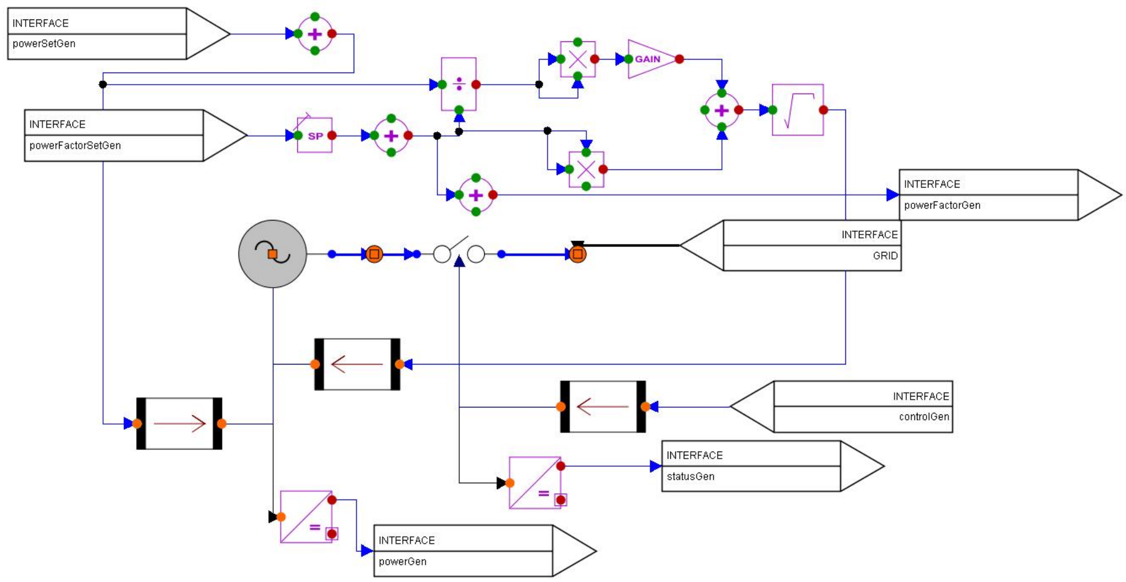

Figure 8. Layout of diesel generator block

5.5 Microgrid Controller (MGC)

The MGC block is mainly implemented as logic circuits. It takes inputs from the EMS system, adjusts them according to network status where needed and then transfers them to device blocks. Currently MGC routine includes voltage control based on droop control characteristics. Figure 9 depicts the applied droop curve. In short, the MGC adjusts the control signals assigned for loads, storage units, EVs and PV in case LV network voltage levels are not proper. For majority of time network status causes no constraints and MGC simply forwards the controls provided by the EMS.

The system receives input from the EMS for all parameters described in previous chapters. Additionally, there are other parameters such as control time step, which are also provided by the EMS. The output of the MGC is the confirmed control signal, which is transferred to component models. These signals also serve as feedback signals to the EMS, stating the eventual values of the signal.

Some measurements from component models are used as input for the MGC. These include measured power values of controllable loads and EVs. Figure 10 depicts the MG Controller block.

All Apros internal and external signals including the ones for MGC has been collected to tables. The complete tables can be found in the appendices.

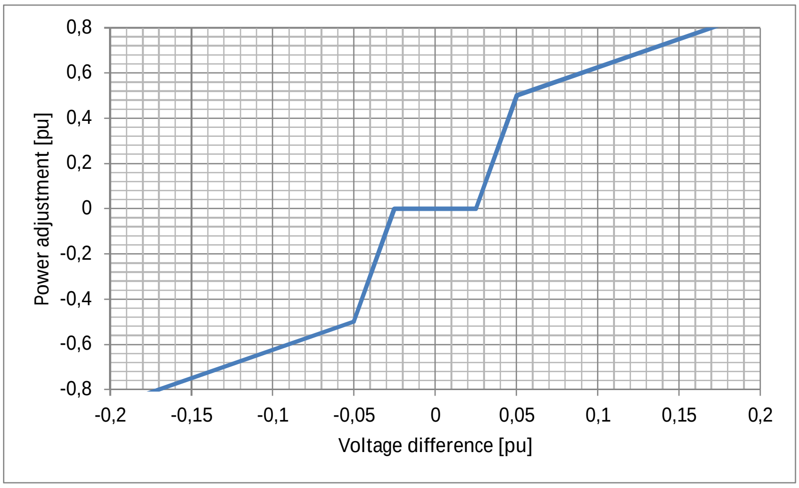

Figure 9. Voltage droop characteristics used in MGC logics

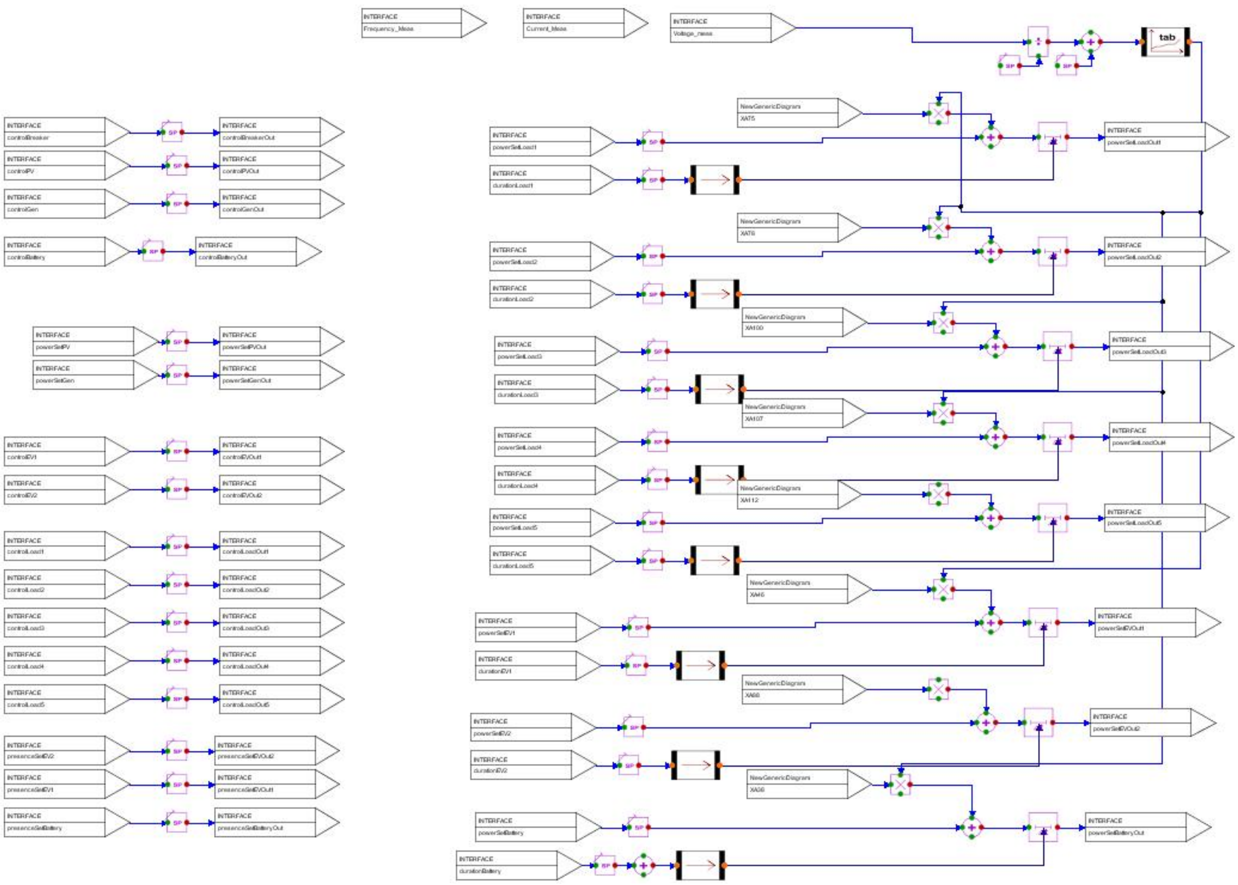

Figure 10. Layout of MGC logics

6 Energy Management System (EMS)

The Energy Management System has been implemented with Python 3.3 and the Graphical User Interface (GUI) has been designed using the PyQt4 library. The control panel, shown in Figure 11, is designed to show all essential information about the microgrid so the user can make informed control decisions. One of the advantages of using the control panel is that the state of the microgrid can be deduced with a single glance to the program. Deducing the state of the microgrid in Apros by clicking through dozens of components by hand would take a lot of time and the user would only see the parameters of a single component at a time, so the control panel gives a huge advantage.

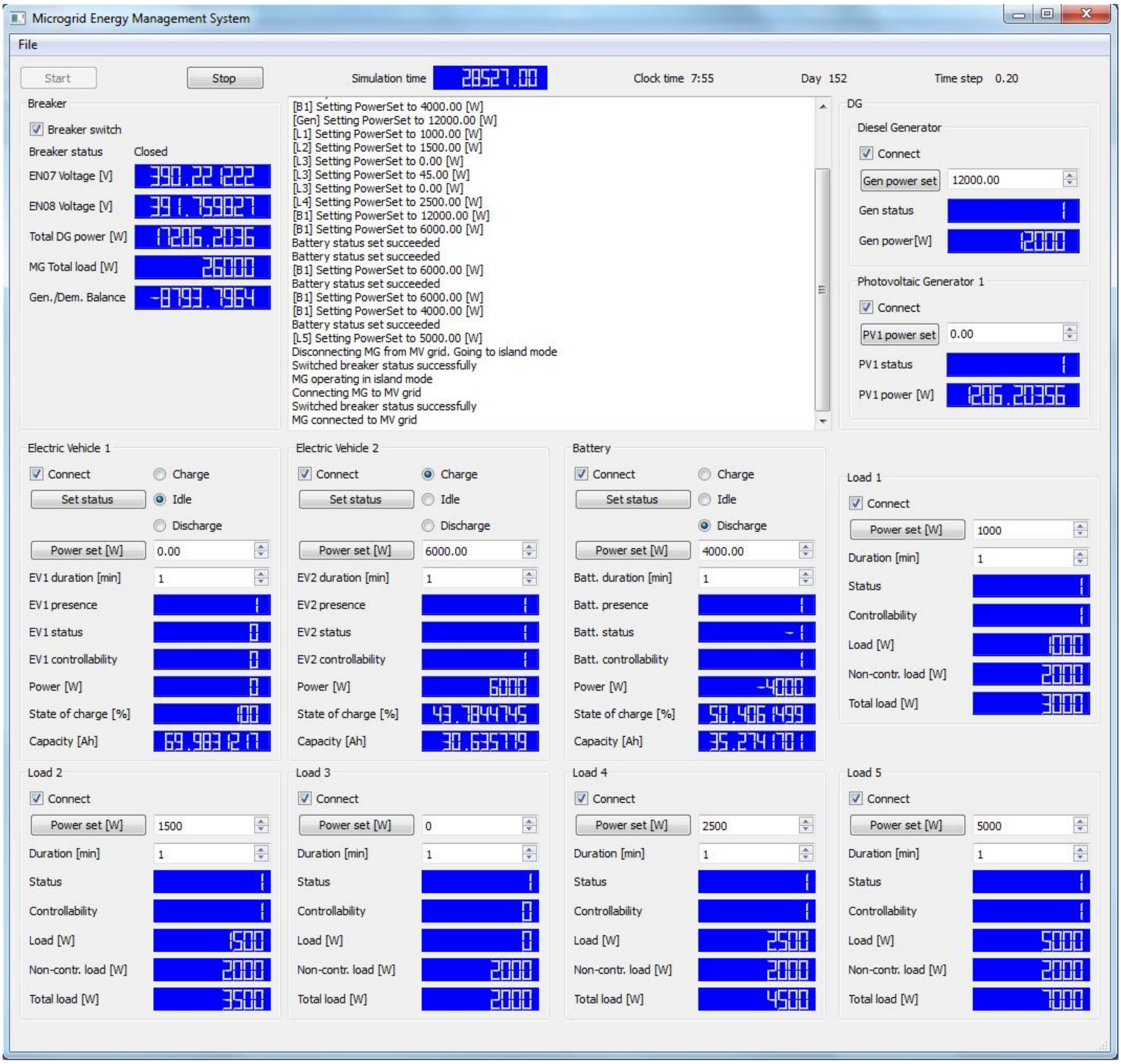

Figure 11. Energy Management System control panel

Another advantage is that errors during long simulations, which would go undetected, can be seen instantly so the user can restart the simulation and not waste time in waiting results that were calculated with failed models. The control panel also gives the user an interface to test individual Apros models. This speeds up testing and helps in detecting problems with the logic of Apros models.

Pressing start button initializes the OPC UA connection to the Apros model. After a few seconds, the simulation starts and all values are refreshed as soon as new values are ready in Apros. The simulation can be stopped and restarted in Apros at any time without the need to restart EMS.

The top of the control panel shows the current simulation time [s], simulation time in hours and minutes (7:55 am), day number and the simulation time step [s]. The day number currently displayed (152) denotes the first of June and it is read from the solar radiation component in the PV unit. The day number is used, among other parameters, to determine the amount of power generated by the PV unit. The components in the Apros model are represented in the UI and their relevant parameters have been grouped together. There is a designated parameter group for each component in the Apros model (Diesel generator, PV unit, EV1 & 2, Battery, and Loads 1 to 5). The upper left corner of the UI has been designated for the MG breaker switch and other general parameters. The upper middle part of the UI contains a textbox, which is used to give feedback to the user about the controls given to the model. All components and the breaker have a user-checkable checkbox indicating their presence in the microgrid. All ‘Connect’ checkboxes control a switch inside the component model connecting the component to the microgrid. The ‘Breaker switch’ checkbox controls the Point of Common-Coupling (PCC). If the breaker switch is closed, the microgrid is connected to the MV grid and if the switch is open, the MG is operating in island mode. There is a ‘Power set [W]’ button and a spinbox for the powerset value for every component in the MG. The user can type or use the arrows of the spinbox to set the desired amount of power/load. By pressing the ‘Power set [W]’ button the value in the spinbox is written to the Apros model. For the diesel generator, powerset value is directly the wanted output power of the diesel generator. After pressing the ‘Power set [W]’ button it takes approximately 10-50 simulation steps for the Apros model to reach the new setpoint. The current output power can be seen in the blue LCD number box e.g. Gen power [W] is the diesel generators current generated power. The ‘Power set [W]’ button for the PV unit is not activated in the current implementation. The output power given by the PV unit is always the maximum amount that can be generated (1206 W at 7:55 am). The ‘Power set [W]’ button could be used to limit the power given to the MG in the next version of the EMS.

The EVs and storage are always in one of three states. Their status can either be charge (1), idle (0) or discharge (-1). As can be seen from their status fields, the status of EV1 is 0 (idle), the status of EV2 is 1 (charging) and the status of storage is -1 (discharging). Their status can be changed by selecting the desired radiobox (charge, idle or discharge) and then pressing the ‘Set status’ button. The amount of power used for charging or discharging can be set with the ‘Power set [W]’ button. EV2 charge setpoint is 6 kW and storage discharge setpoint is 4kW, which is negative to denote that the storage is generating power to the MG. EVs can also be used as backup storages for the MG as long as they are connected to the grid.

Duration button can be used to limit the amount of time that the power setpoint is active, for example we can charge (or discharge) EVs and storage for 2 hours with 6kW power, after which the power is set to zero and status to idle. If duration is zero, the power setpoint is active indefinitely.

Each load contains a controllable (Load [W] field) and a non-controllable (Non-contr. load [W] field) load. For example, load five, has 5 kW controllable load and 2 kW non-controllable load for a total load of 7 kW. The user can set the desired amount of controllable load by using the 'Power set [W]' button. The non-controllable load is fixed to simulate the part of load that is unavailable for demand response.

The total generated power in the microgrid can be seen in the upper left corner (Total DG power [W]). This includes the power generated by the diesel generator and the PV unit but also the power that is discharged from EVs or storage if they are connected. MG Total load [W] field is the sum of all loads in the microgrid. This includes all controllable and non-controllable loads and also the current charging powers of EVs and/or storage. Below is the Gen./Dem. Balance field, which is the total power load subtracted from the total generated power. If balance is negative, the absolute value is the amount of power that the microgrid needs from the MV but if the balance is positive, the surplus generated power could be sold to the energy market.

All signals exchanged via OPC UA are found in tables in the appendices. In addition, all Apros signals can be found there.

7 Microgrid Simulation Platform Functional Sequence Diagrams

This chapter describes how communication operates between the EMS, the MGC and the microgrid components. When the EMS starts the OPC UA client, it needs to subscribe to the variables on the OPC UA server (located in the Apros model) it wants to monitor. These are called monitored variables. The EMS via the OPC UA client can write values to *SetPoint* blocks in the Apros model. The signals in Figure 12, Figure 13, Figure 14 and Figure 15 are presented to give an example on how data input and output works. The figures do not include all used signals in the simulation platform. All signals used in the simulation platform have been collected to tables, which can be found in the appendices. EMS block input signals represent monitored variables and output signals represent signals that write values to the destination components setpoint block. Figure 12 depicts the initialization process of the EMS. When the simulation starts, generators, EVs, the storage, the breaker, and loads send status updates periodically to the MG Controller and the EMS. The length of the update period can be selected in the EMS configuration file and it can either be a fixed value (e.g. 200 ms) or zero, as in instantaneous, meaning that new variable values are sent as soon they are available. Status messages update the connection status (connected/disconnected) of components in the microgrid model. Power generators also send measurements on the amount of the current output power. In addition, electric vehicles and batteries send measurements on their state of charge, power and capacity. Load components send measurements on the current power load in addition to the status messages.

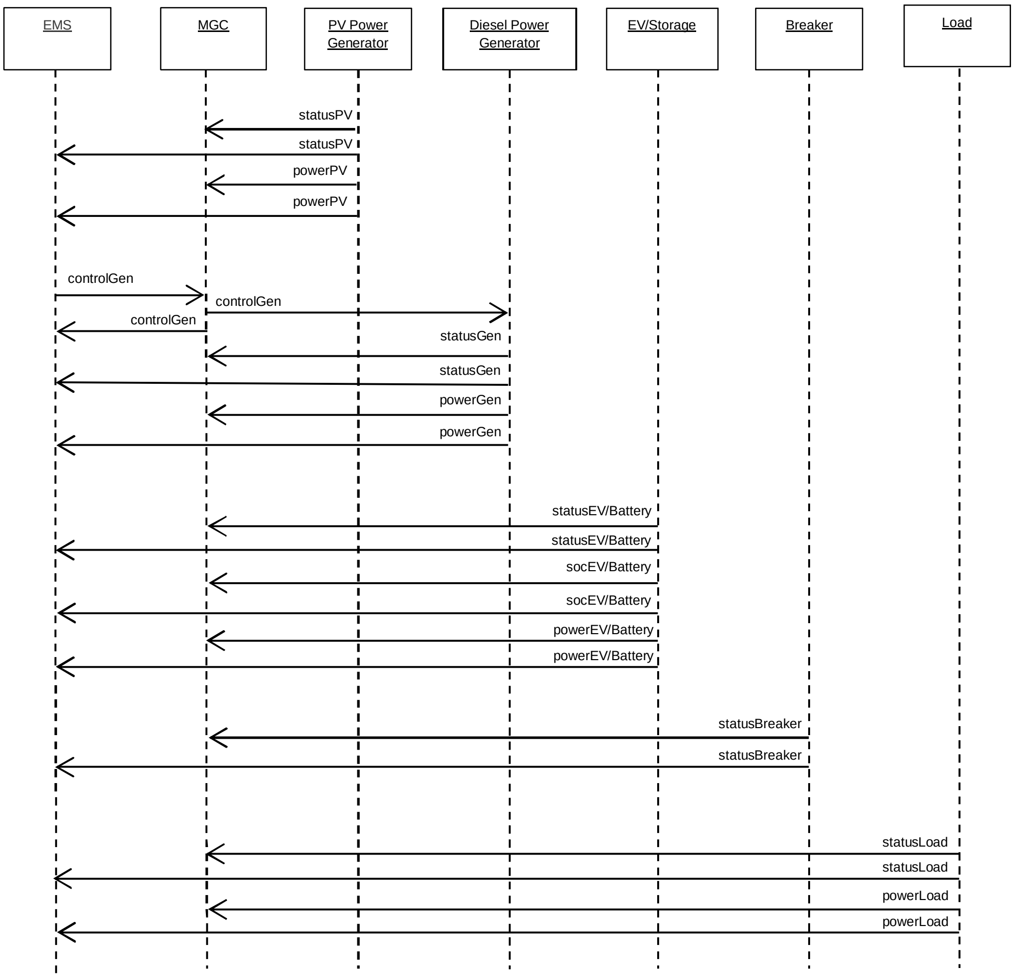

Figure 12. Initialization sequence diagram

Figure 13 depicts the input and output messages of the microgrid model's diesel generator. When the EMS decides to request more or less power from the generation component, it first sends a *powerSetGen* signal to the MG controller. The MG controller checks the request and decides if it is feasible or not. If the request is ok, it is sent to the generation component as is, but if the MG controller decides to modify the request, then the modified request is sent to the generation component instead. The *powerSetGen* value is also sent back to the EMS as feedback because it needs to know the actual command that was sent to the generation component. The generation component sends measurements (*powerGen*) about the current generated power to the MG controller and the EMS.

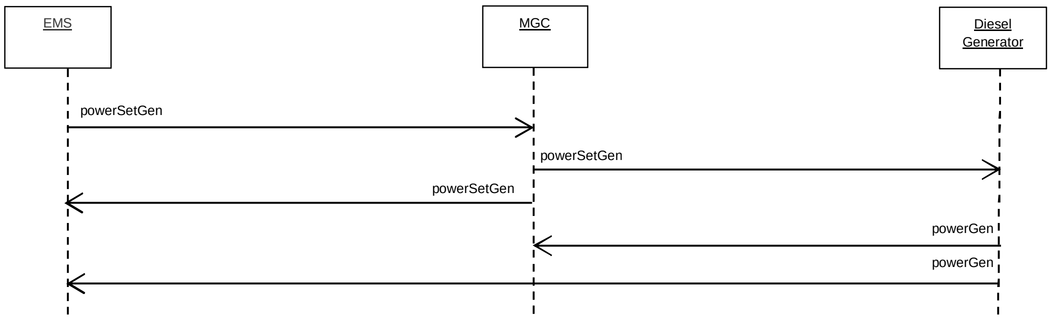

Figure 13. Control of the Diesel Generator

Figure 14 depicts the input and output messages of the load components in the microgrid model. During the simulation, the load components send a *controllabilityLoad* signal to the MG Controller and the EMS. This signal tells whether the load component participates in demand response or not. The loads that can be controlled may receive *powerSetLoad* signals from the EMS via the MG Controller. The logic with *powerSetLoad* signals is similar to the way generation components are controlled with *powerSet* signals. The EMS sends the *powerSetLoad* request to the MGC, which adjusts it or sends it as is to the load component. Either way the EMS receives the signal sent to the load component as feedback. The EMS also controls the duration of the *powerSetLoad* command with the *durationLoad* signal. Load components also send measurements on the current power load with the *powerLoad* signal.

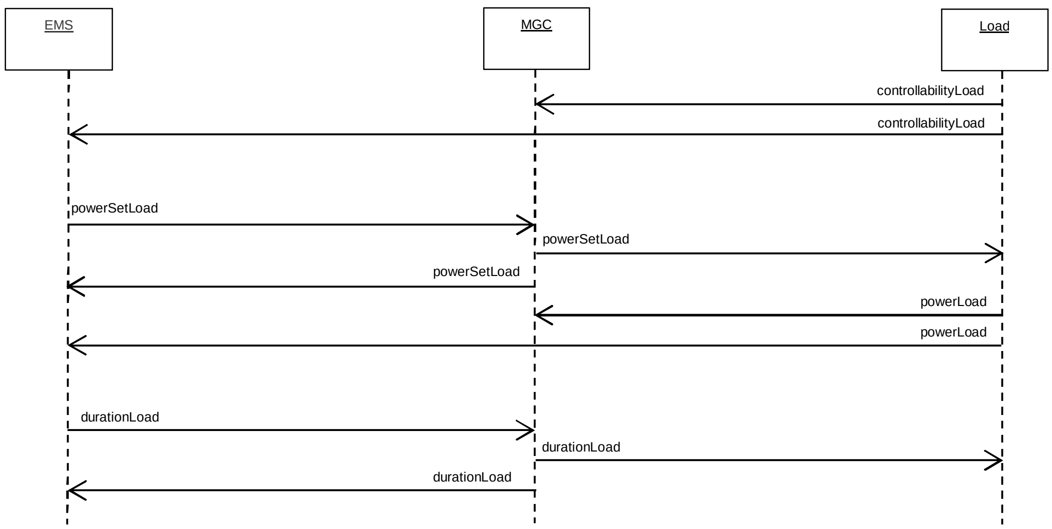

Figure 14. Load Control / Demand Response

Figure 15 depicts the input and output messages of the electric vehicle and storage (battery) models. The behavior of the EVs and battery models is similar in the simulation. Their interfaces are similar, but the difference is that EVs are not always present while the battery is. The presence of EVs is indicated by the *presenceEV* signal. If the EV is not present in the simulation then it naturally cannot be charged or discharged. However, EVs can of course discharge when they are not plugged in. The control signals available for the EMS are *controlEV/Battery, presenceSetEV/Battery,* p*owerSetEV/Battery* and *durationEV/Battery*. Status of EVs or battery can be changed by using the *controlEV/Battery* signal, *presenceSetEV/Battery* connects or disconnects the component to microgrid and the (dis)charging power and duration can be set with *powerSetEV/Battery* and *durationEV/Battery* signals. The EVs and the battery send *statusEV/Battery* signal telling whether they are charging, discharging or idle. A *socEV/Battery* signal tells the state of charge and the *powerEV/Battery* signal tells the current discharging or charging power depending on the control state. *powerEV/Battery* is negative if discharging.

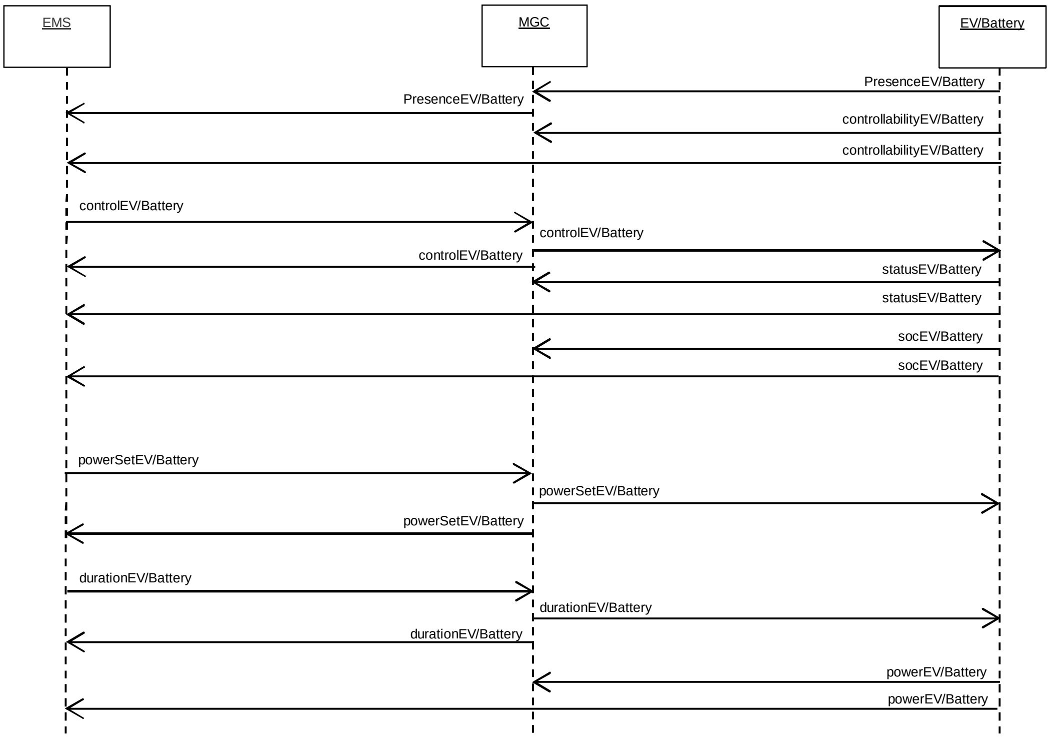

Figure 15. Control of the EV/Battery models

8 Conclusions

In SGEM FP5, we have implemented a Microgrid simulation platform, which consists of a microgrid model, microgrid controller and an Energy Management System. The microgrid model and the microgrid controller are modelled with the Apros simulation software and the Energy Management System is a program implemented in Python and C++. The microgrid model contains various LV components including a diesel generator and a photovoltaic generator for MG energy generation (DG), a battery component for storages, electric vehicle model for storage and/or loads, and controllable and non-controllable loads for demand response modelling. The microgrid controller (MGC) has the responsibility to make sure that the microgrid model runs fluently and adjusts the state of the microgrid dynamically. It does this by controlling the voltage of the LV network components. The voltage control method is based on droop characteristics. The Energy Management System controls either individual microgrid components or all of them toward a common goal. With a little extra development, the EMS can be used to optimize microgrid power management. The optimization goal for the microgrid could be for example to decrease operational costs. This can be done by using as much DG generated power as possible when energy purchased from DSO is expensive or to purchase energy from DSO when it is cheap. Another optimization goal could be to use as much renewable energy as possible, and not pay so much attention to the cost.

Advantages of using the EMS include:

- Detecting the status of the microgrid model
- Detecting the status of individual microgrid components
- Provide information to make informed control decisions
- Instantly detecting faults in long simulations
- Testing the logic of components and models is faster and easier

There are still a lot of possibilities to improve the system. For example, frequency control could be integrated to the Apros microgrid controller to enable the simulation of realistic islanded operation mode. More DG components could be modelled. For example, one useful renewable energy source could be a wind power generator. Also, the EMS offers many possibilities for further development. For example, the next step could be to make a simulation case that lasts for a year or more.

9 References

[1] Pekka Savolainen and Heli Kokkoniemi-Tarkkanen: SGEM project deliverable (FP2): D6.12.7 “Survey Report on KPIs for the Control and Management of Smart Grids”. 2012.

[2] Pekka Savolainen, Heli Kokkoniemi-Tarkkanen, Jouni Hiltunen, and Pekka Ruuska: SGEM project deliverable (FP3): D6.6.21 “A Prototype of an Intelligent Decision-making Component for Smart Grids”. 2013.

[3] Pekka Savolainen, Jouni Hiltunen, Heli Kokkoniemi-Tarkkanen, Riku Pasonen: SGEM project deliverable (FP4): D6.6.31 ”Enhanced prototype of EMS decision-making component”. 2014.

[4] Robert Weiss: SGEM project deliverable (FP3): D5.1.46 “Requirement documentation on selected promising use-cases for Smart Grid communication simulator environment”. August 2013.

[5] http://www.opcfoundation.org/ link checked on 22.2.2015.

10 Appendices

Interfaces of EMS, MG Controller and microgrid model

Description of the columns in the following tables:

- **Name**: Parameter name
- **Type [unit]**: Data type and unit for the parameter
- **Source**: Component(s) which provides the parameter
- **Destination**: Component(s) which uses the parameter
- **Note**: Short description of the parameter and other possible comments

Microgrid configuration (EMS)

Name	Type [unit]	Source	Destination	Note
generator	struct	conf. file	EMS	Static information of windmill, PV, and diesel gen. units
load	struct	conf. file	EMS	Static information of houses and electric vehicles
storage	struct	conf. file	EMS	Static information of batteries

Weather forecasts

Name	Type [unit]	Source	Destination	Note
solarRadiation	time series - double [W/m²]	Apros Solar Radiation block	PV unit	Average solar radiation at certain location. Used by APROS PV generator.

MG busbar <-> MG Controller

Name	Type [unit]	Source	Destination	Note
voltageMeas1	double [V]	MG busbar	MG controller	Phase A voltage measurement
voltageMeas2	double [V]	MG busbar	MG controller	Phase B voltage measurement
voltageMeas3	double [V]	MG busbar	MG controller	Phase C voltage measurement
currentMeas1	double [amp]	MG busbar	MG controller	Phase A current measurement
currentMeas2	double [amp]	MG busbar	MG controller	Phase B current measurement

currentMeas3	double [amp]	MG busbar	MG controller	Phase C current measurement

Breaker <-> MG Controller

Name	Type [unit]	Source	Destination	Note
controlBreaker	int [0/1]	MG controller	Breaker	Open/close breaker (command)
statusBreaker	int [0/1/2]	Breaker	MG controller	Breaker status (open/closed/fail)

Breaker <-> EMS

Name	Type [unit]	Source	Destination	Note
statusBreaker	int [0/1/2]	Breaker	EMS	Breaker status (open/closed/fail)

PV unit <-> MG Controller

Name	Type [unit]	Source	Destination	Note
controlPV	int [1/0]	MG controller	PV unit	Connect/disconnect (command)
statusPV	int [1/0]	PV unit	MG controller	Connection status (connected/dis-connected)
powerSetPV	double [W]	MG controller	PV unit	Setpoint for power production (command)
powerFactorSetPV	double [-1..1]	MG controller	PV unit	Setpoint for power factor
powerPV	double [W]	PV unit	MG controller	Current output power
powerFactorPV	double	PV unit	MG controller	Current power factor

PV unit <-> EMS

Name	Type [unit]	Source	Destination	Note
statusPV	int [1/0]	PV unit	EMS	Connection status (connected/dis-connected)
powerPV	double [W]	PV unit	EMS	Current output power
maxControlLimitPV	double [W]	Apros conf.	PV unit	Maximum limit for the power generation

minControlLimitPV	double [W]	Apros conf.	PV unit	Minimum limit for the power generation

Diesel generator (Gen) <-> MG Controller

Name	Type [unit]	Source	Destination	Note
controlGen	int [1/0]	MG controller	Gen	Connect/disconnect unit (command)
statusGen	int [1/0]	Gen	MG controller	Connection status (connected/dis-connected)
powerSetGen	double [W]	MG controller	Gen	Setpoint for power production (command)
powerFactorSetGen	Double	MG controller	Gen	Setpoint for power factor
powerGen	double [W]	Gen	MG controller	Current output power
powerFactorGen	Double	Gen	MG controller	Current power factor

- NOTE: Diesel generator is relevant for MG Controller only when running. Startup cannot be used. statusGen and powerGen can be used to know whether the generator is running.

Diesel generator (Gen) <-> EMS

Name	Type [unit]	Source	Destination	Note
statusGen	int [1/0]	Gen	EMS	Connection status (connected/dis-connected)
powerGen	double [W]	Gen	EMS	Current output power
maxControlLimitGen	double [W]	Apros conf.	Gen	Maximum limit for the power generation
minControlLimitGen	double [W]	Apros conf.	Gen	Minimum limit for the power generation

Controllable load (Load) <-> MG Controller

Name	Type [unit]	Source	Destination	Note
controlLoad	int [1/0]	MG controller	Load	Load on / off (command)
statusLoad	int [1/0]	Load	MG controller	Load status (on/off)
controllabilityLoad	int [1/0]	Load	MG controller	Available for control

powerSetLoad	double [W]	MG controller	Load	Power setpoint for local controller (command)
durationLoad	double [s]	MG controller	Load	Control duration (command)
powerLoad	double [W]	Load	MG controller	Current power consumption

- NOTE: Local override during the load control is needed. controllabilityLoad can be used for this when.

Controllable load (Load) <-> EMS

Name	Type [unit]	Source	Destination	Note
statusLoad	int [1/0]	Load	EMS	Load status (on/off)
controllabilityLoad	int [1/0]	Load	EMS	Available for control
powerLoad	double [W]	Load	EMS	Current power consumption
maxControlLimitLoad	double [W]	Apros conf.	Load	Maximum limit for the power consumpion
minControlLimitLoad	double [W]	Apros conf.	Load	Minimum limit for the power consumpion

Electric vehicle (EV) <-> MG Controller

Name	Type [unit]	Source	Destination	Note
presenceEV	int [1/0]	EV	MG controller	Connected/disconnected
controlEV	int [1/0/-1]	MG controller	EV	Charge/idle/discharge (command)
statusEV	int [1/0/-1]	EV	MG controller	Charging status (charge/idle/discharge)
controllabilityEV	int [1/0]	EV	MG controller	Available for control
powerSetEV	double [W]	MG controller	EV	Power set for control depending on the control state (command)
durationEV	double [s]	MG controller	EV	Control duration (command)
powerEV	double [W]	EV	MG controller	Current discharging or charging power depending on the control state
socEV	double [%]	EV	MG controller	Battery state of charge
presenceSetEV	int [1/0]	EV	MG controller	Set connection status (connected/disconnected)

capacityEV	double [Ah]	EV	MG controller	Current stored energy

Electric vehicle (EV) <-> EMS

Name	Type [unit]	Source	Destination	Note
presenceEV	int [1/0]	EV	EMS	Connected/ disconnected
statusEV	int [1/0/-1]	EV	EMS	Charging status (charge/idle/discharge)
controllabilityEV	int [1/0]	EV	EMS	Available for control
powerEV	double [W]	EV	EMS	Current discharging or charging power depending on the control state
socEV	double [%]	EV	EMS	Battery state of charge
capacityEV	double [Ah]	EV	EMS	Current stored energy
maxDischargeLimitEV	double [W]	Apros conf.	EV	Maximum limit for the discharging power
minDischargeLimitEV	double [W]	Apros conf.	EV	Minimum limit for the discharging power
maxChargeLimitEV	double [W]	Apros conf.	EV	Maximum limit for charging power
minChargeLimitEV	double [W]	Apros conf.	EV	Minimum limit for charging power

Storage (Battery) <-> MG Controller

Name	Type [unit]	Source	Destination	Note
presenceBattery	int [1/0]	Battery	MG controller	Connected/disconnected
controlBattery	int [1/0/-1]	MG controller	Battery	Charge/idle/discharge (command)
statusBattery	int [1/0/-1]	Battery	MG controller	Charging status (charge/idle/discharge)
controllabilityBattery	int [1/0]	Battery	MG controller	Available for control
powerSetBattery	double [W]	MG controller	Battery	Power set for control depending on the control state (command)
maxPowerBattery	double [W]	MG controller	Battery	Maximum charging power (command)
durationBattery	double [s]	MG controller	Battery	Control duration (command)
powerBattery	double [W]	Battery	MG controller	Current discharging or charging power depending on the control state

socBattery	double [%]	Battery	MG controller	Battery state of charge
presenceSetBattery	int [1/0]	Battery	MG controller	Set connection status (connected/disconnected)
capacityBattery	double [Ah]	Battery	MG controller	Current stored energy

Storage (Battery) <-> EMS

Name	Type [unit]	Source	Destination	Note
presenceBattery	int [1/0]	Battery	EMS	Connected/disconnected
statusBattery	int [1/0/-1]	Battery	EMS	Charging status (charge/idle/discharge)
controllabilityBattery	int [1/0]	Battery	EMS	Available for control
powerBattery	double [W]	Battery	EMS	Current discharging or charging power depending on the control state
socBattery	double [%]	Battery	EMS	Battery state of charge
capacityBattery	double [Ah]	Battery	EMS	How much energy can the battery store
maxDischargeLimit-Battery	double [W]	Apros conf.	Battery	Maximum limit for the discharging power
minDischargeLimit-Battery	double [W]	Apros conf.	Battery	Minimum limit for the discharging power
maxChargeLimit-Battery	double [W]	Apros conf.	Battery	Maximum limit for charging power
minChargeLimit-Battery	double [W]	Apros conf.	Battery	Minimum limit for charging power

MG Controller <-> EMS

Name	Type [unit]	Source	Destination	Note
controlBreaker	int [0/1]	EMS	MG controller	Open/close breaker (recommendation)
controlBreaker	int [0/1]	MG controller	EMS	Open/close breaker (feedback)
controlPV	int [1/0]	EMS	MG controller	Connect/disconnect (recommendation)
controlPV	int [1/0]	MG controller	EMS	Connect/disconnect (feedback)
controlGen	int [1/0]	EMS	MG controller	Connect/disconnect unit (recommendation)

controlGen	int [1/0]	MG controller	EMS	Connect/disconnect unit (feedback)
controlLoad1	int [1/0]	EMS	MG controller	Load1 on / off (recommendation)
controlLoad1	int [1/0]	MG controller	EMS	Load1 on / off (feedback)
controlLoad2	int [1/0]	EMS	MG controller	Load2 on / off (recommendation)
controlLoad2	int [1/0]	MG controller	EMS	Load2 on / off (feedback)
controlLoad3	int [1/0]	EMS	MG controller	Load3 on / off (recommendation)
controlLoad3	int [1/0]	MG controller	EMS	Load3 on / off (feedback)
controlLoad4	int [1/0]	EMS	MG controller	Load4 on / off (recommendation)
controlLoad4	int [1/0]	MG controller	EMS	Load4 on / off (feedback)
controlLoad5	int [1/0]	EMS	MG controller	Load5 on / off (recommendation)
controlLoad5	int [1/0]	MG controller	EMS	Load5 on / off (feedback)
controlEV1	int [1/0/-1]	EMS	MG controller	Charge/idle/discharge (recommendation)
controlEV1	int [1/0/-1]	MG controller	EMS	Charge/idle/discharge (feedback)
controlEV2	int [1/0/-1]	EMS	MG controller	Charge/idle/discharge (recommendation)
controlEV2	int [1/0/-1]	MG controller	EMS	Charge/idle/discharge (feedback)
controlBattery	int [1/0/-1]	EMS	MG controller	Charge/idle/discharge (recommendation)
controlBattery	int [1/0/-1]	MG controller	EMS	Charge/idle/discharge (feedback)
connectEV1	int [1/0]	EMS	MG controller	Connect/disconnect EV1 (presenceSetEV1)
connectEV1	int [1/0]	MG controller	EMS	Connect/disconnect EV1 (presenceSetEV1 feedback)
connectEV2	int [1/0]	EMS	MG controller	Connect/disconnect EV2 (presenceSetEV2)
connectEV2	int [1/0]	MG controller	EMS	Connect/disconnect EV2 (presenceSetEV2 feedback)
connectBattery	int [1/0]	Battery	MG controller	Connect/disconnect Battery (presenceSetBattery)
connectBattery	int [1/0]	MG controller	Battery	Connect/disconnect Battery (presenceSetBattery feedback)
powerSetPV	double [W]	EMS	MG controller	Setpoint for power production (recommendation)
powerSetPV	double [W]	MG controller	EMS	Setpoint for power production (feedback)

powerSetWind	double [W]	EMS	MG controller	Setpoint for power production (recommendation)
powerSetWind	double [W]	MG controller	EMS	Setpoint for power production (feedback)
powerSetGen	double [W]	EMS	MG controller	Setpoint for power production (recommendation)
powerSetGen	double [W]	MG controller	EMS	Setpoint for power production (feedback)
powerSetLoad1	double [W]	EMS	MG controller	Power setpoint for local controller (recommendation)
powerSetLoad1	double [W]	MG controller	EMS	Power setpoint for local controller (feedback)
powerSetLoad2	double [W]	EMS	MG controller	Power setpoint for local controller (recommendation)
powerSetLoad2	double [W]	MG controller	EMS	Power setpoint for local controller (feedback)
powerSetLoad3	double [W]	EMS	MG controller	Power setpoint for local controller (recommendation)
powerSetLoad3	double [W]	MG controller	EMS	Power setpoint for local controller (feedback)
powerSetLoad4	double [W]	EMS	MG controller	Power setpoint for local controller (recommendation)
powerSetLoad4	double [W]	MG controller	EMS	Power setpoint for local controller (feedback)
powerSetLoad5	double [W]	EMS	MG controller	Power setpoint for local controller (recommendation)
powerSetLoad5	double [W]	MG controller	EMS	Power setpoint for local controller (feedback)
powerSetEV1	double [W]	EMS	MG controller	Power set for control depending on the control state (recommendation)
powerSetEV1	double [W]	MG controller	EMS	Power set for control depending on the control state (feedback)
powerSetEV2	double [W]	EMS	MG controller	Power set for control depending on the control state (recommendation)
powerSetEV2	double [W]	MG controller	EMS	Power set for control depending on the control state (feedback)
powerSetBattery	double [W]	EMS	MG controller	Power set for control depending on the control state (recommendation)
powerSetBattery	double [W]	MG controller	EMS	Power set for control depending on the

				control state (feedback)
durationLoad1	double [s]	EMS	MG controller	Control duration (recommendation)
durationLoad1	double [s]	MG controller	EMS	Control duration (feedback)
durationLoad2	double [s]	EMS	MG controller	Control duration (recommendation)
durationLoad2	double [s]	MG controller	EMS	Control duration (feedback)
durationLoad3	double [s]	EMS	MG controller	Control duration (recommendation)
durationLoad3	double [s]	MG controller	EMS	Control duration (feedback)
durationLoad4	double [s]	EMS	MG controller	Control duration (recommendation)
durationLoad4	double [s]	MG controller	EMS	Control duration (feedback)
durationLoad5	double [s]	EMS	MG controller	Control duration (recommendation)
durationLoad5	double [s]	MG controller	EMS	Control duration (feedback)
durationEV1	double [s]	EMS	MG controller	Control duration (recommendation)
durationEV1	double [s]	MG controller	EMS	Control duration (feedback)
durationEV2	double [s]	EMS	MG controller	Control duration (recommendation)
durationEV2	double [s]	MG controller	EMS	Control duration (feedback)
durationBattery	double [s]	EMS	MG controller	Control duration (recommendation)
durationBattery	double [s]	MG controller	EMS	Control duration (feedback)