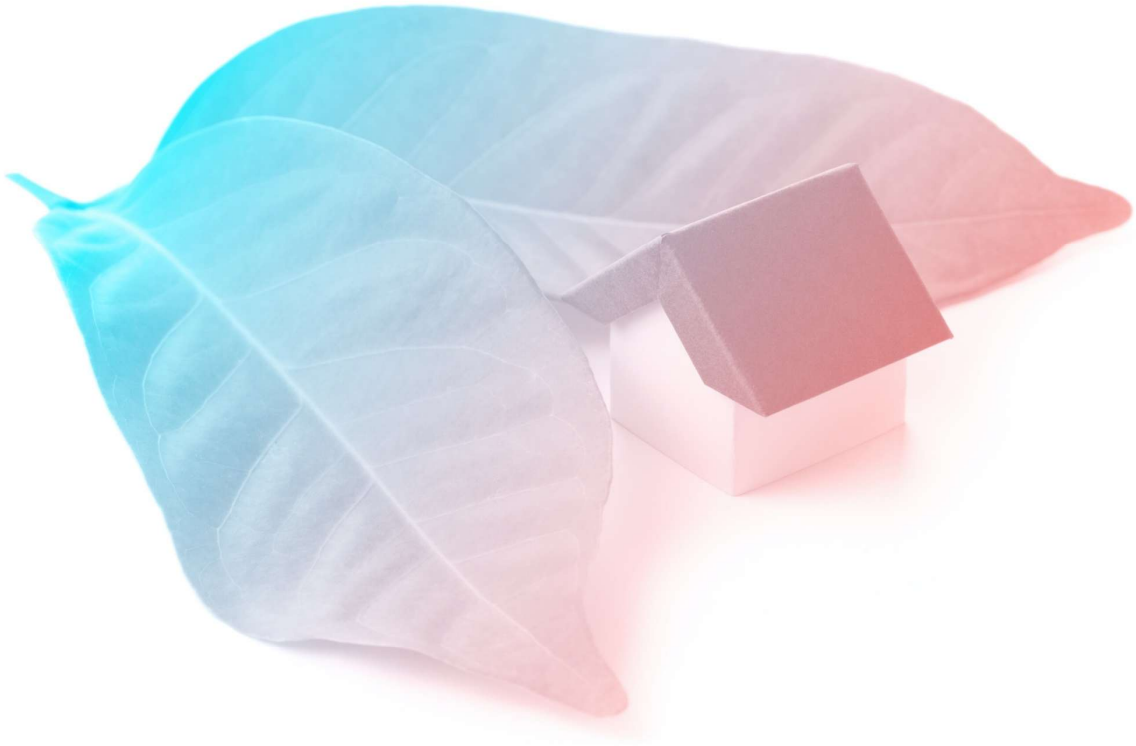
D5.2 Detailed design of the Energy Management System



Authors: Luis Jimeno (CAR), Alberto Belda (CAR), Roberto Arnanz (CAR)

This project has received funding from the European Union´s Horizon 2020 research and innovation programme under the grant agreement No 869821

D5.2 - Detailed design of the Energy Management System

Summary
Deliverable D5.2 presents the finalized design of the MiniStor Smart Home Energy Management System (SHEMS) platform, building on the initial architecture and control strategies reported in D5.1 and the operation mode definitions of D2.4. It documents the detailed control-algorithm specifications, modular software architecture, and integration with cloud-based forecasting and optimization services. Field implementation at the five European demo sites of the project (Thessaloniki, Kimmeria, Sopron, Santiago de Compostela and Cork) is described, highlighting site-specific adaptations, commissioning results and continuous fine-tuning. Analytical models for heat pumps, phase-change material and thermochemical storage validate performance against real data. A robust IoT backend, Grafana alerting, fault-tolerant data pipeline and auxiliary safety PLC ensure operational transparency and reliability. The final design meets MiniStor's energy-efficiency, self-optimization and user-comfort objectives, demonstrating readiness for full deployment.

Deliverable Number	Work Package
D5.2	WP5
Lead Beneficiary	**Deliverable Authors**
CARTIF Technology Centre (CARTIF)	Luis Jimeno (CAR), Alberto Belda (CARTIF), Roberto Arnanz (CAR)
Beneficiaries	**Deliverable Reviewers**
Woodspring IERC	Zoltan Pasztory Carlos Ochoa
Planned Delivery Date	**Actual Delivery Date**
30/06/2025	28/07/2025

Type of deliverable	R	Report	X
	Demo	Demonstrator	
Dissemination Level	PU	Public	X
	CO	Confidential	

Index

List of tables

List of figures

List of acronyms and abbreviations

API	Application Programming Interface
BACnet	Building Automation and Control Networks
COP	Coefficient Of Performance
CSV	Comma-Separated Values
DB	Database
DHW	Domestic Hot Water
DER	Distributed Energy Resources
EMS	Energy Management System
GMT	Greenwich Mean Time
HEMS	Home Energy Management System
HLC	High-Level Controller
HP	Heat Pump
HTF	Heat Transfer Fluid
HVAC	Heating, Ventilation and Air Conditioning
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MQTT	Message Queuing Telemetry Transport
MPC	Model Predictive Control
NH_3	Ammonia (Hydrogen nitride)
PID	Piping and Instrumentation Diagram
PLC	Programmable Logic Controller
PCM	Phase-Change Material
PV	Photovoltaics
PVT	Photovoltaic-Thermal
RMSE	Root-Mean-Square Error
SCADA	Supervisory Control And Data Acquisition
SHEMS	Smart Home Energy Management System
SOC	State Of Charge
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TCM	Thermochemical Material
TRNSYS	Transient System Simulation software
WP	Work Package

Executive summary

Deliverable D5.2 “Detailed design of the Energy Management System” finalizes the MiniStor Smart Home Energy Management System (SHEMS) by consolidating architecture refinements, control algorithms, and system integrations developed since D5.1. The document details a four-layer EMS structure which includes acquisition, control, supervision, and high-level strategy, and describes how rule-based logic running in Node-RED balances photovoltaic-thermal (PVT) input, Phase-Change Material (PCM) and thermochemical (TCM) storage, heat pump (HP) operation and user demands. An MQTT/InfluxDB/Grafana pipeline provides real-time monitoring and alerting, while a secondary PLC ensures safe shutdown under communication failures.

Field commissioning across the five European sites participating in the project validated key operating modes such as pressurization, charging, discharging and dynamic heat-pump interactions, revealing necessary site-specific tweaks such as safety modes, preheating thresholds in cold climates and BACnet integration for building networks. Analytical models for the HP, PCM and TCM subsystems were calibrated against live data, showing reasonable errors and confirming model fidelity for use in high-level optimization.

A Python-based cloud service on Azure hosts forecasting and genetic-algorithm-based evolutionary optimization modules, exchanging data via REST APIs to propose setpoints that the low-level controller enforces if feasible. Overall, D5.2 delivers a fully validated, modular and scalable EMS platform that achieves MiniStor’s goals of maximizing self-consumption, reducing costs, and maintaining occupant comfort under real-world conditions.

1. Introduction

1.1. Scope and objectives of the deliverable

Deliverable D5.2, titled *Detailed design of the Energy Management System*, presents the final design of the MiniStor Smart Home Energy Management System (SHEMS); the platform responsible for automated control and self-optimization of the integrated MiniStor energy storage solution. This document follows and builds upon the initial design presented in deliverable D5.1. Whereas D5.1 provided the conceptual architecture and first implementation of the control system, deliverable D5.2 now consolidates all developments and refinements into a comprehensive detailed design ready for deployment at the project's demonstration sites. In essence, the scope of deliverable D5.2 is to report on the completed EMS design, incorporating all modules and improvements introduced since the initial design phase. The objectives of this deliverable are fourfold:

- Finalize the control system design: provide a full and detailed specification of the MiniStor control platform (SHEMS), including its architecture and control algorithms at both local device level and high-level supervisory level. This includes any refinements made after prototype evaluations and integration of new components, ensuring the design is robust and fully aligned with the final system configuration.
- Integrate optimization features: the EMS includes optimized control strategies based on the generation and demand predictions shown in D5.3 *Design of the forecasting modes* that contain the specific user preferences and operation restrictions of every site.
- Incorporate site-specific requirements: ensure the control strategies and operation modes account for particular preferences and feedback gathered during the project, as well as insights from site evaluations and preliminary tests at the demo installations. Thus, the final design reflects any needed adjustments to setpoints, operating modes or the practical realities observed at each demonstration site.
- Align with MiniStor project objectives: demonstrate that the final EMS design meets the overarching goals of the MiniStor project. The SHEMS is tasked with contributing to supplying the building's demand using the different subsystem constituting the prototype, including the integrated thermal Phase-Change Material (PCM)-Thermochemical Material (TCM) and electrical storage in an optimal way, while maximizing the use of local renewable energy sources and maintaining comfortable indoor conditions. Deliverable D5.2 confirms that the control platform's design accomplishes these objectives. It shows how the EMS coordinates all subsystems under various operating scenarios to efficiently utilize local renewable sources, thereby contributing to the project's energy and environmental targets.

1.2. Structure of the deliverable

For clarity and completeness, this deliverable is organized into several sections, each covering a specific aspect of the detailed design. In particular, the structure of D5.2 is as follows:

- **Section 1 - Introduction:** outlines the scope and objectives of D5.2, describes the document's structure and explains how this deliverable relates to other tasks and outputs of the MiniStor project.
- **Section 2 - SHEMS final design:** presents the overall design of the SMES architecture, highlighting all components and their interactions in the final configuration. This includes the final architecture diagram of the MiniStor control system and a description of the main components, with emphasis on any refinements made since the initial design. The section also explains how the SHEMS interfaces with the project's IoT platform for data exchange and user interaction (as initially outlined in D5.1). It specifies the final configuration of controllers, communication networks/protocols and the software algorithms implemented.

- **Section 3 - Implementation and validation of control strategies and operational modes:** presents the results of applying the operational and functional requirements and implementing the control strategies and operation modes of the MiniStor system defined in deliverables D2.4 and D5.1 in the different demo sites. Examples of the main operation modes in the demonstration environments, therefore validating the low-level control strategies under real conditions. Additionally, specific adaptations, additional modes and fine-tuning actions implemented in the demo sites are described in detail, ultimately ensuring the safe and efficient operation of the MiniStor system. In this section, the results of the development of the high-level control algorithms are also presented.
- **Section 4 - Conclusions:** summarizes the key achievements of the detailed design phase and highlights how the objectives set out for the MiniStor EMS have been met. It reflects on the progress since the initial design (D5.1) and emphasizes the new capabilities and improvements introduced in this final design deliverable, such as full integration of forecasting, enhanced control stability, etc. Overall, the conclusion reaffirms the readiness of the MiniStor EMS for operation and its impact on the project's goals.

This structured approach ensures that the reader can easily follow the development from high-level concepts down to implementation details of the MiniStor EMS. Each section provides a logical piece of the overall picture, making the deliverable comprehensive yet accessible. Technical descriptions are presented in a clear manner, with supporting figures where appropriate, so that readers who are less familiar with certain technical aspects can understand the key points.

1.3. Relation to other tasks and deliverables

Deliverable D5.2 is closely interconnected with several other tasks and deliverables across the MiniStor project. As the culmination of Task 5.1 (*Design of the MiniStor control and self-optimization platform*) under Work Package (WP) 5, this report both utilizes inputs from earlier work and provides a foundation for other activities. Key relationships include:

- **WP2 and WP3 – System specifications and operation modes:** the design of the EMS in D5.2 is built upon the technical specifications and operation modes developed in previous work packages. For instance, Task 2.4 (deliverable D2.4) provided the system characterization and operational parameters of the MiniStor solution, including definitions of baseline operating modes and conditions. Likewise, Task 3.1 (D3.1) and Task 3.2 (D3.2) delivered the initial system dimensioning and peripheral equipment designs, respectively, which defined the overall layout, subsystem connections and device characteristics that the control system must handle. Task 3.5 (deliverable D3.8) focused on the electrical storage subsystem design, supplying details on the battery system operation and its control modes. All these inputs have been considered and incorporated into the EMS design. The final control algorithms explicitly account for the characteristics and constraints of each component as defined in those deliverables (e.g., charge/discharge conditions, temperature limits, capacities), ensuring consistency with the system designed by WP2–WP4.
- **WP5 – Forecasting module and IoT platform integration:** within the same work package as this deliverable, two other tasks have direct interactions with the EMS design and deployment. Task 5.2, documented in deliverable D5.3, has developed the renewable energy production and demand forecasting models to be used in MiniStor. D5.2 takes these forecasting modules as critical inputs, and the final SHEMS design integrates them to enable predictive control (for example, using forecasted PV generation and heat demand to plan charging of storage). Additionally, Task 5.3/5.4, documented in deliverable D5.4, and D5.5 designed the IoT-based user interface platform for MiniStor. The EMS must interface with this IoT platform to exchange data and receive user inputs; therefore, D5.2 aligns with D5.4 by defining how the control system connects to the cloud platform or home automation interface for user interaction and remote monitoring. The close relation between D5.2 and

D5.4 ensures that the control algorithms and the user interface operate seamlessly together, providing a cohesive smart home energy management experience.

- WP6 – Monitoring system and demonstration: the MiniStor EMS design is also linked to the project's monitoring and demonstration activities. Task 6.1 (deliverable D6.1) defined the monitoring system and Key Performance Indicators (KPIs) for the MiniStor demo sites. Those KPIs (e.g. renewable self-consumption ratio, energy cost savings, comfort levels) and the specified sensor measurements are directly relevant to the EMS: they guide what performance metrics the control system should optimize and how success is measured. In D5.2, the control design incorporates the capability to collect data from the monitoring devices (temperature sensors, power meters, etc. as described in D6.1) and to output the necessary control and status signals for monitoring. This ensures that when the system is deployed, all required data for performance evaluation will be available and the EMS can be assessed against the defined KPIs. Thus, D5.2 effectively bridges the gap between the design phase and the execution phase: it translates the various research and development outcomes from WP2–WP5 into a unified control platform to be used in WP6 to evaluate the performance of the MiniStor system.

In summary, deliverable D5.2 is a converging point of many project threads. It integrates inputs from prior technical deliverables (system models, component designs, forecasting tools, user interface design, etc.) into the final control system. This interconnectedness guarantees that the EMS design is fully informed by all relevant knowledge generated in the project and that it functions cohesively with the MiniStor hardware, software and user-oriented components.

2. SHEMS final design

The SHEMS is the central control platform for the MiniStor residential energy solution. It builds on the initial architecture defined in deliverable D5.1 and incorporates refinements identified during prototype tests and user feedback. The SHEMS coordinates all energy subsystems to meet user demand efficiently and maximize self-consumption of renewables.

2.1. SHEMS architecture

The SHEMS developed in MiniStor acts as the brain of the residential energy system. Its modular, scalable structure can monitor, coordinate and optimize the following subsystems constituting MiniStor:

- **Photovoltaic generation**, which manages solar production and routes energy to consumption or storage;
- **Thermal storage** (latent PCM and thermochemical TCM), managed according to demand and resource availability;
- **Heating and cooling**, through coordinated control of HPs, heat exchangers and hydraulic circuits;
- **Monitoring and forecasting**, which uses weather and energy-production predictions and consumption models for an efficient energy management.

The SHEMS architecture follows a four-layer model (see Figure 1):

- **Acquisition and communication layer:** to collect data from sensors, actuators, energy meters and weather stations via protocols such as Modbus, BACnet and MQTT. This layer normalizes measurements and forwards them to the control engine.
- **Processing and control layer:** to host the energy management and optimization algorithms, including predictive models or model predictive control (MPC), with strategies to prioritize self-consumption or thermal comfort.
- **Interface and supervision layer:** to provide dashboards (for example, Grafana) and APIs for end users, energy managers and remote platforms to monitor system status and adjust preferences in real time.
- **High-level control layer:** to define the overall operating strategies, aligning local control with global objectives such as energy balance targets and applicable policies.

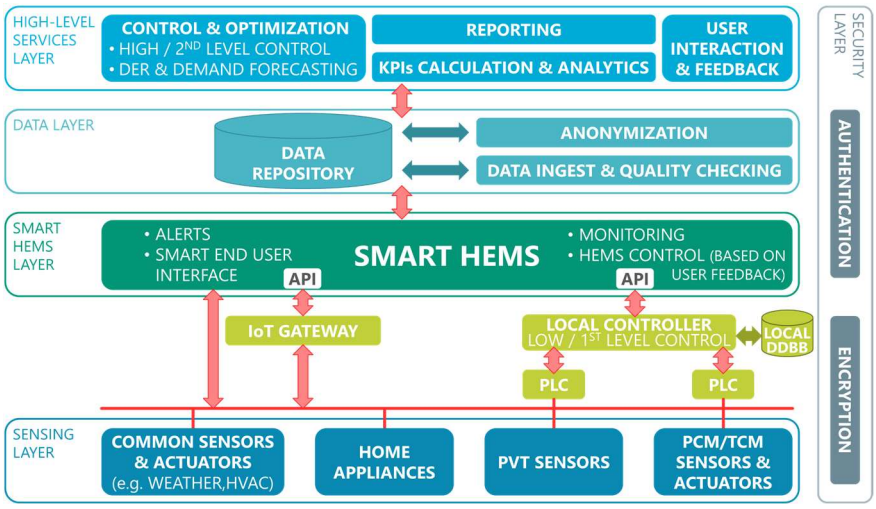


Figure 1: SHEMS and IoT platform architecture design

The processing and control layer constitute the intelligent core of the EMS. It hosts the algorithms and rules that govern the system's energy behaviour. The control was implemented using a rule-based logic in the visual environment Node-RED and the logic itself is a comprehensive set of rules to balance and efficiently manage the solar generation, thermal storage, instantaneous thermal demand and weather forecasts within a robust operational framework.

Each decision evaluates multiple real-time variables and anticipates their evolution (for example, responding to an upcoming temperature drop, prioritizing PCM versus TCM storage, or disconnecting loads to avoid inefficiencies). Structuring this logic through visual nodes required detailed functional modelling, thorough validation of logical sequences and extensive exception handling.

2.1.1. Submodules

The control logic has been structured into clearly defined submodules, as represented in Figure 2 and detailed below:

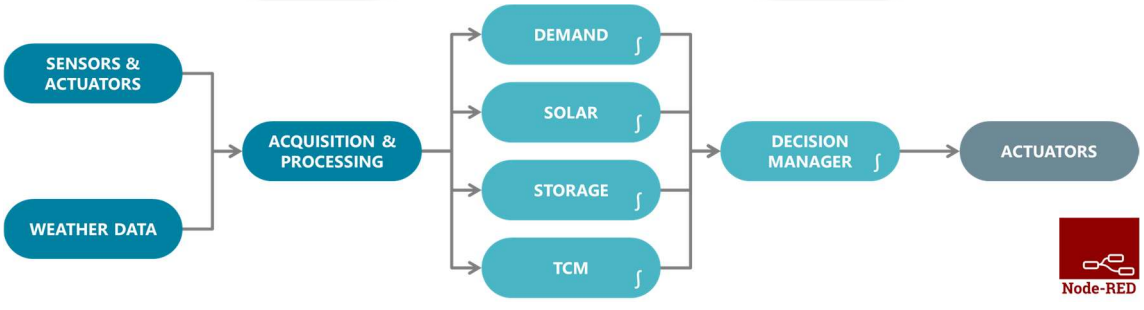


Figure 2: SHEMS control submodules and data flow in Node-RED

Thermal demand management ("Demand" in Figure 2)

This submodule decides whether the system can supply heat or cooling based on current and forecasted conditions. Two strategies are available:

- Scheduled operation: the user defines time slots during which heating or cooling may run, provided that thermal storage has sufficient energy. This simplifies the control logic and enables predictable consumption planning.
- Demand-signal operation: for environments with a higher level of automation, the system receives real-time demand signals from each home's automation infrastructure. Full integration with existing smart-home systems allows the EMS to respond flexibly to actual usage patterns.

Solar thermal supervision ("Solar" in Figure 2)

This submodule governs the activation of the solar circuit. Despite the basic control of the solar loop is managed by its own system, this submodule monitors the inverter and thermal battery status and applies adaptive thresholds (with seasonal hysteresis) to avoid unnecessary cycling. The system ensures the inertia tank has enough temperature margin to feed the TCM steadily and it triggers early start-up when generation exceeds predefined limits, therefore maximizing the energy use and reducing the need for thermal dissipation through fan coils.

Thermal battery management ("Storage" in Figure 2)

Responsible for charging and discharging both heating and cooling PCM units, this submodule processes discharge requests and determines the optimal timing for charging according to the user's thermal demand and the status of the devices. It considers state-of-charge, source availability and local PCM controller signals. The logic prioritizes solar heat when available, prevents inefficient overlap (for example, charging PCMs with TCM heat in summer) and enforces charge-order rules (such as filling the heating PCM before the DHW PCM).

TCM reactor management ("TCM" in Figure 2)

This submodule controls TCM charging and discharging cycles using key signals (temperatures, state-of-charge, operating status, energy availability). It prioritizes thermal consistency by avoiding interruptions and inefficient start-ups and it adapts to seasonal conditions to decide whether the TCM should act as a heat source or sink. Low-level safety control of the reactor's valves and pumps remains delegated to the TCM's dedicated controller.

All submodules are themselves finite-state machines, each defining the logic and transitions for a specific domain (e.g. thermal-battery management, solar-loop control, TCM reactor cycling). These submodules generate status signals that feed into a top-level state machine (implemented in JavaScript). Each state in this hierarchical architecture represents a system configuration, explicitly setting every actuator (valves, pumps, heat exchangers, fans). Therefore, these states are not simple switches, but functional blocks that ensure all system components are properly aligned with the intended operational mode. Figure 3 shows an example of one such global state definition.

```
async function Transition_NOHEATDemandTCMCHARGING()
{
    global.set("Request.TCMCharge",1);
    global.set("Request.TCMDischarge",0);
    global.set("estado_aux",0);
    global.set('ffg',5)
    if (global.get("global.TCMstate")=== "precharging")
    {
    global.set("Valves.TCMrecirc_in_3wv_out",0);
    global.set("Valves.TCMin_3wv_out",0);
    global.set("Valves.TCMrecirc_out_3wv_out",1);
    global.set("Pumps.TCMcondenser_pump_out",1);
    global.set("Valves.TCMcondenser_out_valve_out",1);
    global.set('ffg',1)
    global.set("Valves.TCMcondenser_in_valve_out",1);
    global.set("Pumps.TCMevaporator_pump_out",0);
    global.set("Valves.TCMevaporator_in_valve_out",0);
    global.set("Valves.FANCOILcold_in_3wv_out",0);
    global.set("Fancoils.FANCOILcold_out",0);
    global.set("Valves.HPevaporator_in_valve_out",0);
    global.set("Pumps.HPcondenser_pump_out",0);
    global.set("Valves.MANIFOLDreturn_valve_out",0);
    global.set("Valves.FANCOILheat_in_valve_out",1);
    global.set("Fancoils.FANCOILheat_out",1);
    global.set("Valves.MANIFOLDsupply_valve_out",0);

      global.set("estado_aux",1);
    if (global.get ('HP') === 1)
    {global.set("HP",0);
    await delay(60000)// wait 1 minute to turn off HP
    }
    else{global.set("HP",0);}
    global.set("global.Heat_available",1);
    global.set("Valves.PVTtoColdPCM_valve_out",0);
    global.set("Valves.PVTtoPRODpump_valve_out",1);
        if (global.get("NH3_PT_1") > 10)
    {global.set("Pumps.TCMreactor_pump_out",0)}
    else if (global.get("NH3_PT_1") < 9)
    {global.set("Pumps.TCMreactor_pump_out",1);}

    }
    global.set("string_signals.State","Transition_NOHEATDemandTCMCHARGING");
}
```

Figure 3: Example of state definition in the SHEMS controller, in JavaSript

Transitions between states are defined by logical conditions combining submodule outputs, sensor readings and weather data (see an example in Figure 4). Within each state, additional branches that manage specific variants depending on the operational context and enforce minimum on/off intervals and boundary-condition checks (for example, timers, temperature thresholds), ensuring safe and efficient operation.

```
//C1 && c23
if ((global.get("global.PVTstate") === "SolarHot")  && global.get("string_signals.State") === "OFF" && global.get("global.PCM_heat_REQUIRED")=== 0   &&
(global.get("inputs.TCMcondenser_T")> ( global.get("inputs.Demo_outdoor_T") + global.get("parameters.FANCOIL_Tdiff") + (global.get("parameters.Hystheresis_T")/2)))
&& (global.get("global.TCMstate") === "discharged" || global.get("global.TCMstate") === "chargingPAUSE") )

{
    if ((global.get("inputs.TCMcondenser_T") +global.get("parameters.HP_Tdiff")) < global.get("parameters.HP_T_max") )

    A_SA_Transition_NOHEATDemandTCMCHARGING()
}
```

Figure 4: Example of transition in the SHEMS controller, in JavaScript

This modular, hierarchical approach promotes the coordination between subsystems, avoids conflicts during simultaneous charging or discharging, maintains safety margins and allows new rules or scenarios to be added with minimal code changes, making the system a flexible and adaptable platform for different configurations and operating conditions.

2.2. Intermediate database and data monitoring tool

The MiniStor monitoring architecture centralizes acquisition, transmission and processing of data from both system equipment (temperatures, flow rates, valve and pump states) and external sources (solar generation, weather conditions). It builds on the commissioning and start-up requirements of each demonstrator to provide real-time visibility and historical analysis.

2.2.1. Data acquisition with Node-RED

Field data collection is implemented in Node-RED. This environment acquires sensor signals and internal system states via industrial protocols such as Modbus and MQTT. Figure 5 shows the Node-RED flow used to publish measurements to the MQTT broker.

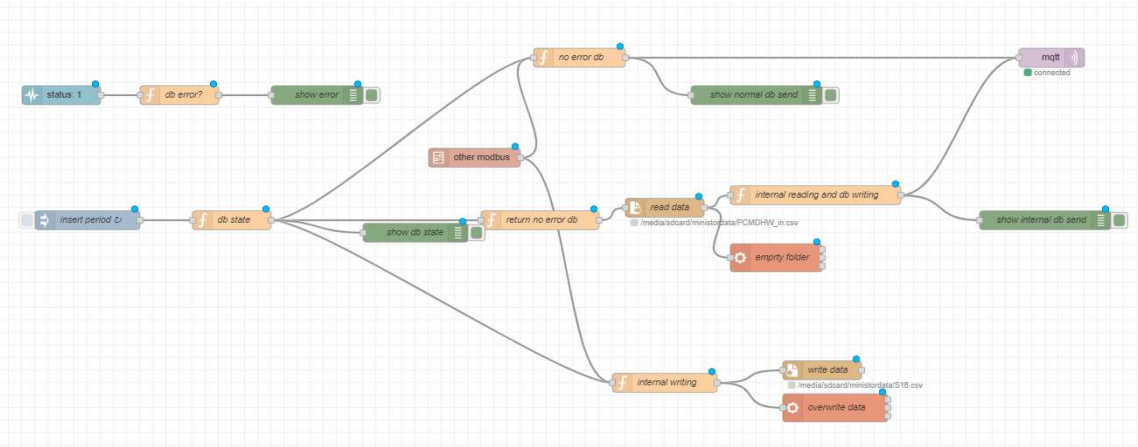


Figure 5: Node-RED flow for MQTT data publishing

2.2.2. MQTT broker, message aggregation and time-series storage with Telegraf and InfluxDB

Collected data are published in real time via the MQTT protocol to a centralized Mosquitto broker. This broker aggregates information from all demonstrators into a single-entry point, simplifying the communication architecture and enabling remote sites to publish and subscribe securely.

The broker's messages are consumed by the Telegraf plugin, which filters, structures and forwards them to an InfluxDB time-series database. This database forms the core of the monitoring system and supports efficient storage and retrieval of high-resolution temporal data (Figure 6).

Figure 6: Data explorer influxDB

2.2.3. Visualization with Grafana

Grafana serves as the primary visualization interface for the monitoring and analysis of the operational data collected from the demonstrators for technical activities, including the commissioning and fine-tuning of the demonstration sites. It connects directly to InfluxDB to display real-time and historical trends of key variables, facilitating diagnosis, validation and optimization of thermal and electrical cycles.

This end-to-end pipeline (Node-RED - MQTT broker - Telegraf - InfluxDB - Grafana, see Figure 7) provides a robust, scalable monitoring tool that supports commissioning, performance analysis and ongoing optimization of the MiniStor demonstrators.

Figure 7: Monitoring data transmission architecture

This solution proved to be key during the commissioning and verification phases of each demonstrator, providing a centralized and customizable view of the behaviour of subsystems such as the TCM, PCM, hydraulic network and solar generation.

As an example, Figure 8 and Figure 9 display the complete operating cycle of the MiniStor system; from the gradual temperature increase in the TCM sensors to the cyclical behaviour of PCM temperatures during their charging and discharging phases. Similarly, Figure 8 on the bottom graph shows the instantaneous solar power, allowing for direct analysis of the relationship between renewable energy production and the activation of thermal subsystems.

This type of visualization also enables comparison between actual performance and simulation models. Additionally, Grafana provides mechanisms for alert generation, data quality control and user- or device-level segmentation, making it a comprehensive monitoring tool within the IoT ecosystem proposed by MiniStor.

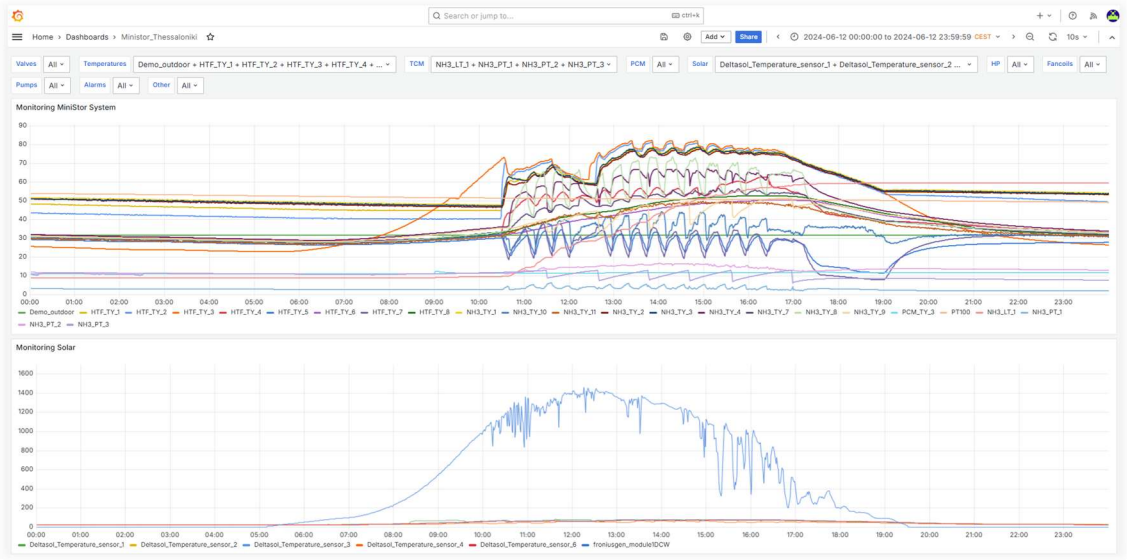


Figure 8: Example of Grafana visualization platform

Figure 9: Zoomed-in screenshot of the Grafana panel

2.3. SCADA system

The Supervisory Control and Data Acquisition (SCADA) system provides a graphical user interface for real-time supervision and control of the MiniStor installation. Built on an SVG (Scalable Vector Graphics) diagram synchronized with the PLC, the SCADA panel enables operators to visualize system status, issue commands and monitor alarms in a single view.

2.3.1. Interface overview

Figure 10 illustrates the main SCADA screen, where all key subsystems are represented with standardized icons and piping: solar field, inertia tank, TCM reactor, PCM batteries (heating, DHW and cooling), heat pump (HP) and building load. Pale lime-green value boxes overlay each component to show live measurements (temperatures, flow rates, pressures) and actuator states (valve positions, pump status) through communication with the PLC. This allows for a direct visual correlation between each physical component and its real-time operation.

In the upper-left corner of the SCADA (see Figure 10), a status panel displays:

- Global state and operating mode (manual, automatic or semi-automatic)
- State of charge of TCM and PCM units
- Active charge/discharge requests

Below, a user panel indicates current season mode (winter/summer) and any heating or cooling demands. An alarm section at the bottom left corner reports critical warnings (for example, compressor faults or communication errors).

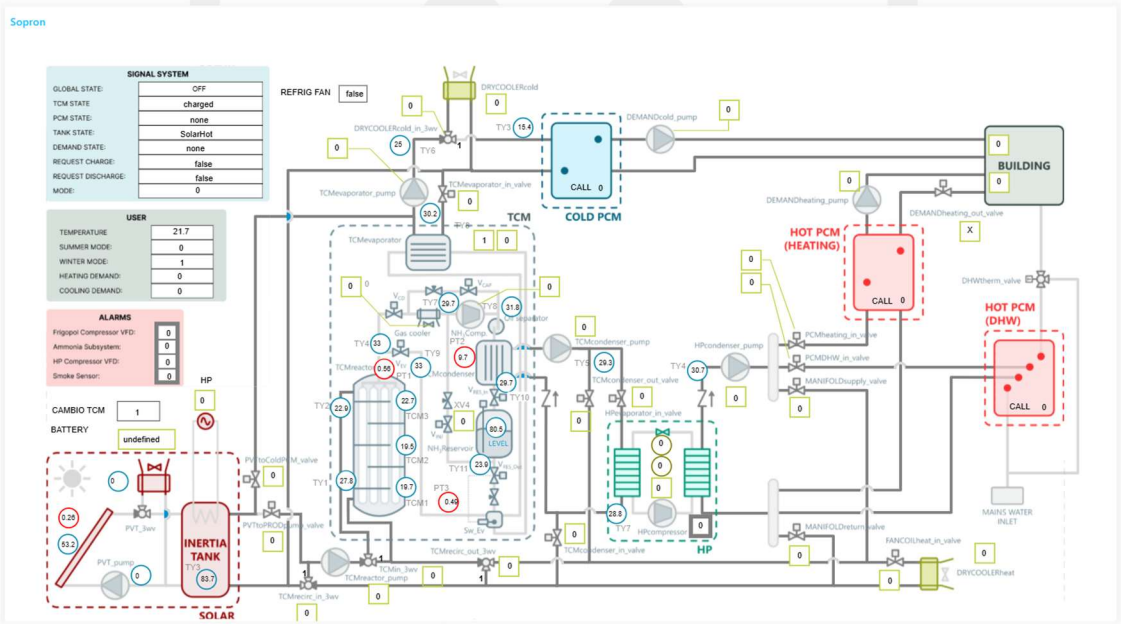


Figure 10: SCADA overview of the MiniStor system – Example of Sopron demo site

2.3.2. Dynamic updates and control modes

The functional blocks are organized to logically reflect thermal flows: from generation sources (solar, HP) to storage systems (inertia tank, TCM, PCM) and finally to the building. The connections through pipelines are depicted with grey lines that include motorized valves, pumps and sensors, all represented using standardized icons. These pipelines turn red when actuators are open, providing immediate visual feedback on active flow paths.

The SCADA supports three control modes that allow operation to be adapted based on environmental needs or the user's profile:

i. Manual mode: operators toggle each actuator (pumps, valves) directly via on-screen switches, taking full control over the system's actions and decisions. This setup allows each component to be manually activated or deactivated (see Figure 11), regardless of the system's automatic conditions. This mode is ideal for commissioning, maintenance and functional testing.
ii. Automatic mode: the EMS control engine manages all operations without user intervention, executing the defined rule-based logic and therefore enabling continuous and autonomous performance.
iii. Semi-automatic mode: the advantages of the manual and automatic approaches are combined. The system analyses conditions and recommends state transitions. Operators review the suggestion on the console and approve it by clicking "Continue", enabling guided control that balances automation with human supervision. It is ideal for validating system behaviour in a real plant or for enabling use by operators without advanced technical expertise.

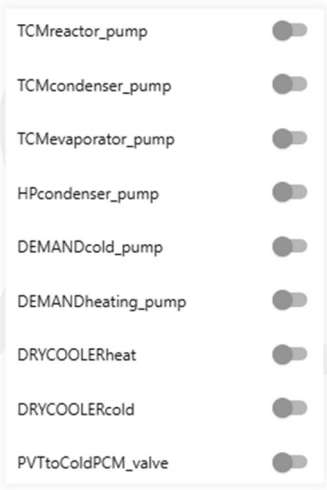


Figure 11: Manual control actuators

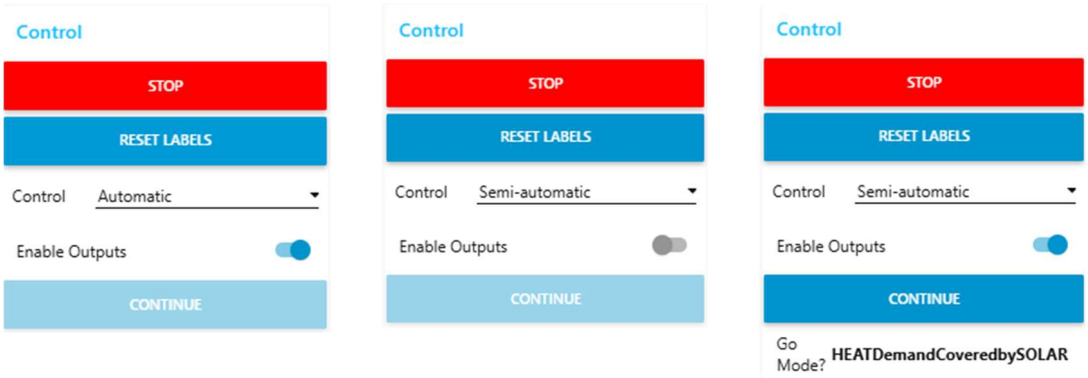


Figure 12: Control mode examples in the SCADA interface

The following images (Figure 13, Figure 14 and Figure 15) illustrate real-time examples of the SCADA interface during system operation, showcasing how the platform behaves under various working conditions.

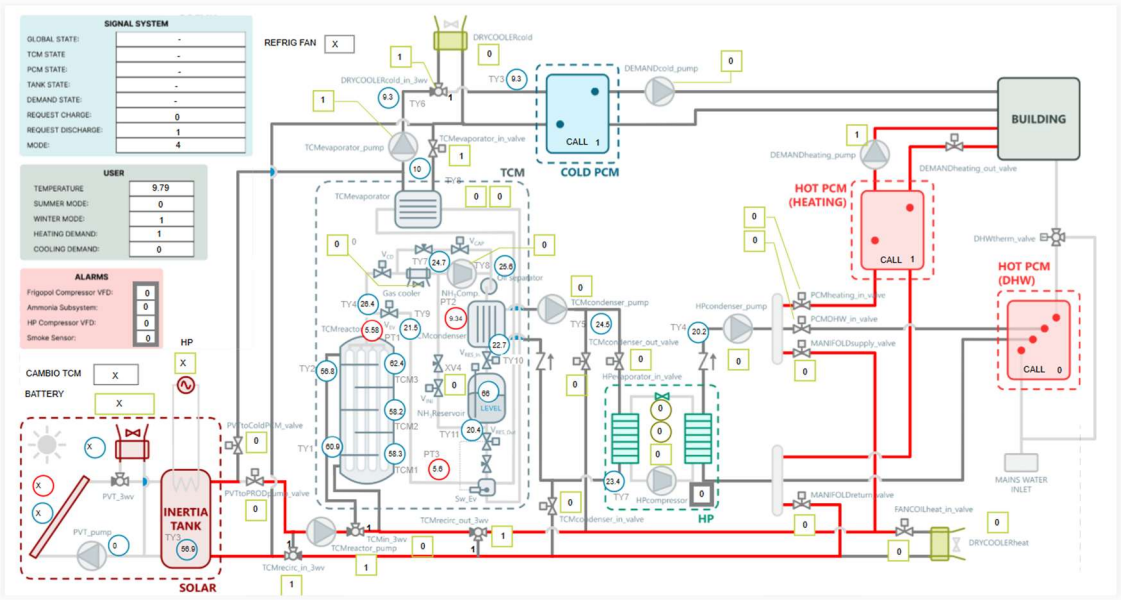


Figure 13: Example of use of the SCADA system – MiniStor state: charging PCM from solar and discharging

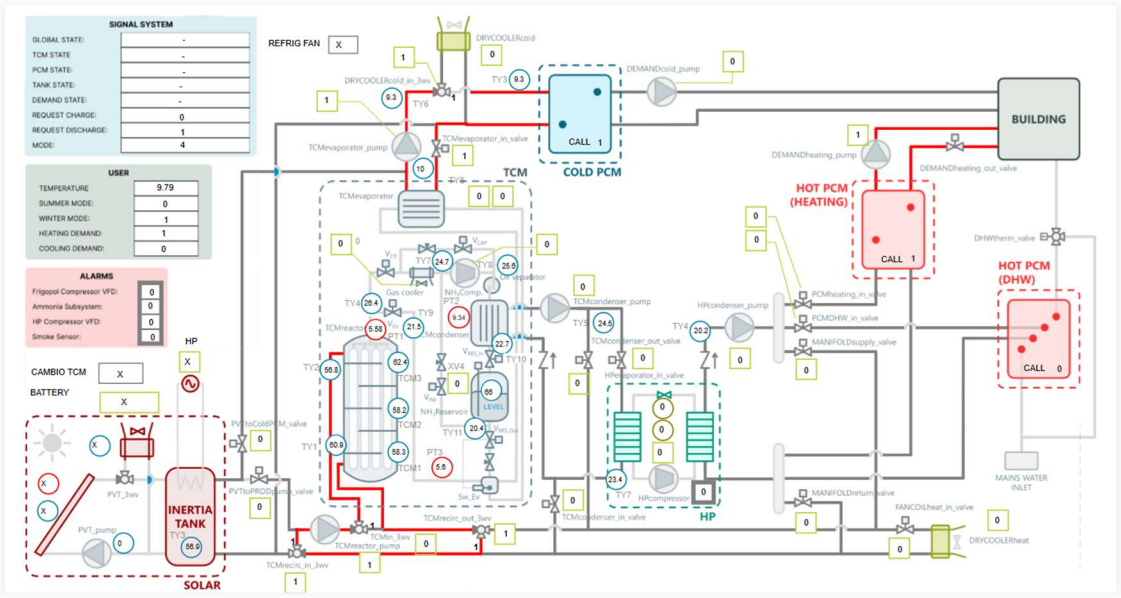


Figure 14: Example of use of the SCADA system – MiniStor state: discharging TCM, charging cold PCM and discharging heat PCM

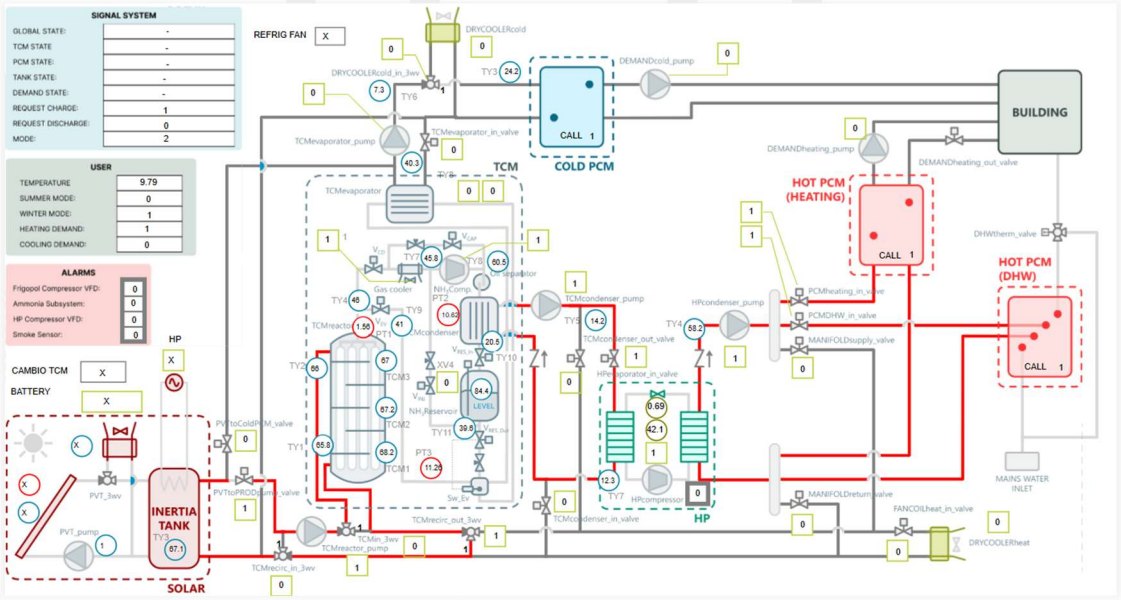


Figure 15: Example of use of the SCADA system – MiniStor state: charging TCM and PCMs

By combining real-time visualization with flexible control modes, the SCADA system:

- Enhances situational awareness, showing how each subsystem interacts;
- Speeds up fault diagnosis through integrated alarms and status panels;
- Facilitates safe commissioning by allowing step-by-step manual overrides;
- Supports training of operators via semi-automatic guidance.

Together with the Grafana dashboards (section2.2.3), the SCADA interface completes the MiniStor monitoring ecosystem, delivering both instantaneous operational insight and long-term performance analysis.

2.4. Continuous monitoring and tailored alarming system

A robust monitoring and alerting framework are critical to ensure reliable operation of the distributed MiniStor demonstrators. This framework combines Grafana's real-time visualization and alerting capabilities with a fault-tolerant data-acquisition pipeline in Node-RED and a secondary PLC-based safety shutdown. Together, these elements guarantee continuous data integrity, rapid anomaly detection and secure fall-back control.

2.4.1. Grafana-based alerting

Grafana connects directly to the InfluxDB time-series database to display live and historical trends of all key variables. Its alerting engine runs queries at configurable intervals and triggers notifications when predefined conditions are met. For example, the "Data Loss Sopron" rule uses a Flux query against the last 10 minutes of the *HTF_TY_3* temperature series. If no new data arrive in that window, the alert fires automatically (Figure 16).

Figure 16: Example of Flux-based alert definition in Grafana for data-loss detection

Each alert shows its current status (Normal, Paused, or Firing), health metrics and next evaluation time. Operators can view details, silence notifications, or edit rule parameters directly from the Grafana UI. An alarms list (Figure 17) provides at-a-glance visibility into all active and configured alerts across sites and subsystems.

Figure 17: Sample of the Grafana alarms dashboard

When an alert fires, for instance, “Data Loss Cork”, Grafana includes rule name, affected instances and evaluation timestamps in its notifications. These alerts integrate with external channels (email, SMS, or messaging platforms) to drive immediate response to data-collection failures or sensor disconnections (Figure 18).

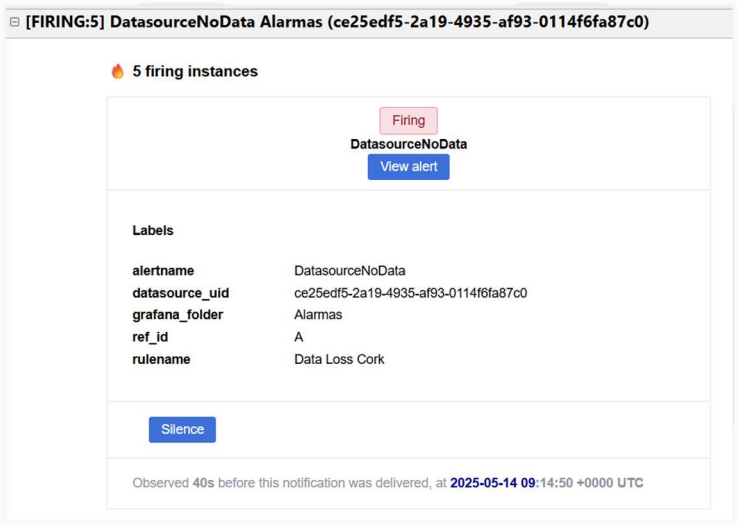


Figure 18: Example of an active Grafana alert message

The advantages of Grafana, including its clear interface, capability to edit rules dynamically or compatibility with multiple data sources, has made it a key tool for the continuous monitoring of distributed systems by simplifying error detection and accelerating operational decision-making.

As an example, this alerting setup proved essential during the Kimmeria summer tests, where it detected early signs of overheating in the thermal circuit and enabled preventive interventions before any critical failures. This allowed for preventive actions before critical system failures could develop.

2.4.2. Fault-tolerance and data integrity

This entire data acquisition and transmission methodology is complemented by a fault-tolerance and data integrity strategy designed to prevent data loss in the event of temporary unavailability of the InfluxDB database service. The system implements a verification mechanism that checks the database's availability during each data transmission cycle. Depending on the result, three different operating modes are applied:

- Mode 0 (Online): when InfluxDB is available, data are sent directly to the MQTT broker and onward to the database.
- Mode 1 (Recovery): if InfluxDB has just returned online, previously buffered data are recovered and resent.
- Mode 2 (Offline): if InfluxDB remains offline, data are stored locally in internal memory until connectivity is restored.

This approach, visually represented in Figure 19, ensures the integrity of time series data and avoids information gaps, which is essential for continuous monitoring processes, performance analysis and technical validation of the demonstrators.

Node-RED coordinates this logic with asynchronous functions and retry cycles , ensuring continuous, gap-free time-series records.

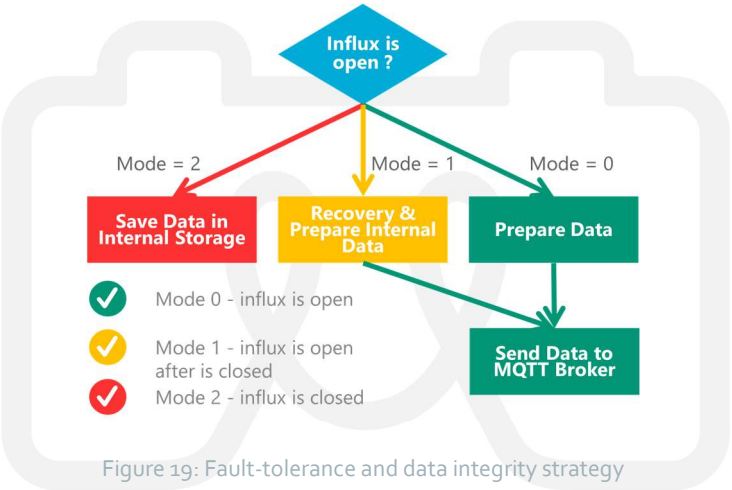


Figure 19: Fault-tolerance and data integrity strategy

2.4.3. Auxiliary safety shutdown

In parallel with the development of the main system, a backup subsystem has been implemented through a secondary low-level PLC which provides an autonomous safety shutdown in the event of main PLC failure or disconnection of the main PLC (Figure 20). This backup functions by continuously monitoring the communication link with the high-level PLC, which is responsible for the system's primary control logic. If the backup PLC detects that a heartbeat or valid messages have not been received within a predefined timeout period, it interprets this as a failure of the main PLC and initiates a programmed shutdown sequence that takes operational priority over any previously received commands.

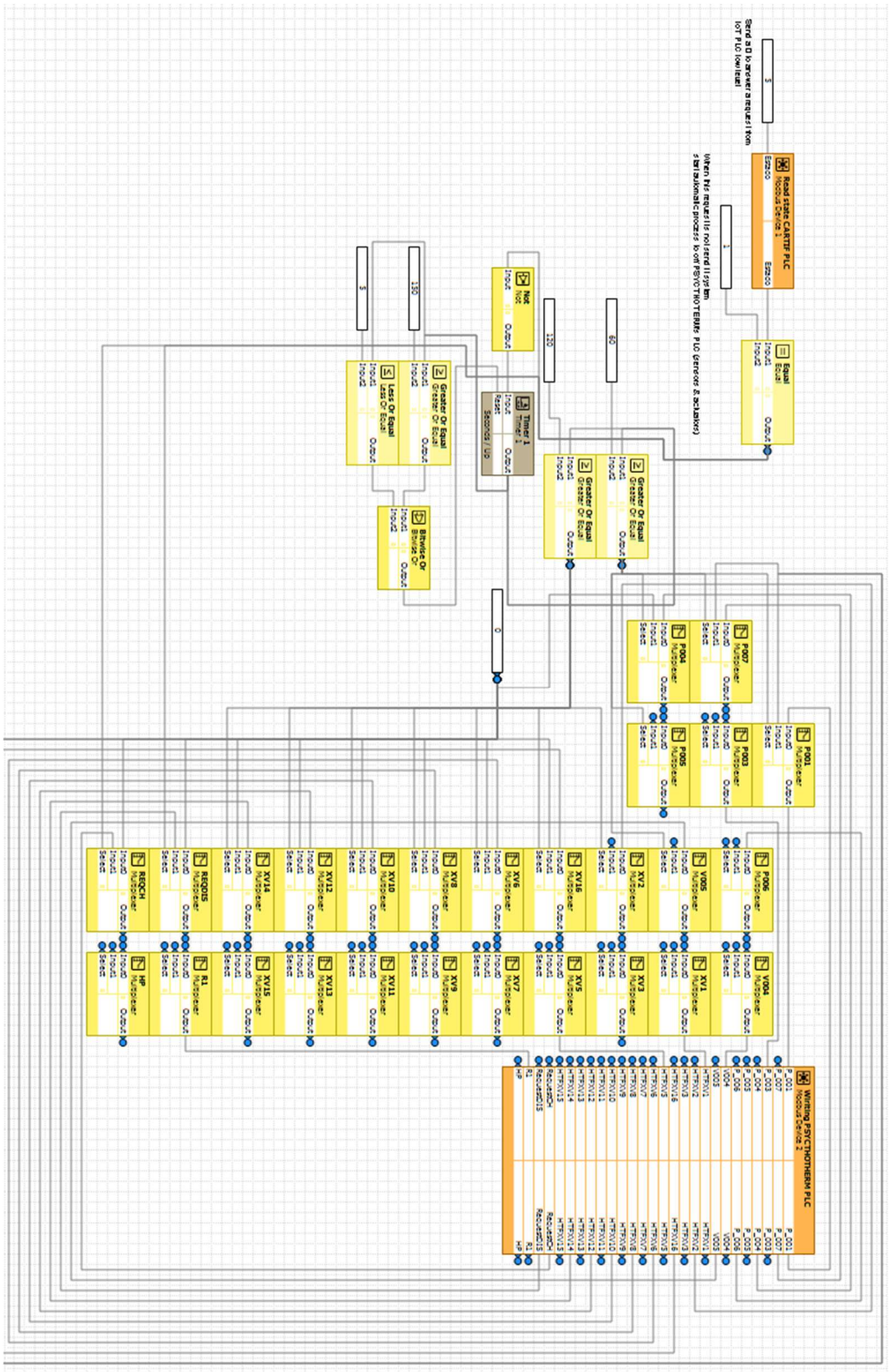


Figure 20: Function-block diagram of auxiliary PLC shutdown logic

The defined shutdown protocol consists of two phases:

1. **Shutdown of active components:** Immediately upon detecting the failure, all dynamic actuators such as circulation pumps, fan coils, or other active thermal exchange systems are deactivated. This action prevents uncontrolled operation of the system and ensures that energy exchange processes do not continue unsupervised.
2. **Delayed valve closure:** After a one-minute safety delay, during which systems such as the HP are allowed to circulate residual fluids and dissipate remaining energy, all motorized valves associated with the thermal circuit are safely closed. This delay is designed to avoid hydraulic blockages or unwanted thermal build-up, allowing a smooth transition to a passive state.

Once the shutdown sequence is completed, the backup PLC enters a passive monitoring state. It remains in this mode until communication with the main PLC is re-established. At that point, the failure condition is cleared and full control is handed back to the original system, disabling the auxiliary shutdown logic.

It is important to note that this auxiliary PLC communicates directly with the PLC which is responsible for real-time sensor readings and actuator control. This enables the auxiliary system to intervene effectively and in real time in the physical system, even in the absence of the main controller, thereby ensuring a robust and secure fall-back strategy in the event of communication loss or malfunction of the primary PLC.

This auxiliary control logic was **validated in the laboratory through multiple tests** that simulated real fault conditions, replicating scenarios previously observed during similar system shutdowns. In particular, situations involving power losses to the main PLC were reproduced, which resulted in the interruption of control logic and the unprotected state of the thermal subsystems. During these tests, the auxiliary system demonstrated a reliable and rapid response, executing the programmed shutdown sequence as expected, without causing conflicts with actuators or leaving the system in unstable or unsafe states.

To ensure the effectiveness of this backup mechanism, it was identified as essential to provide a robust and independent power supply for the auxiliary PLC. Since its purpose is precisely to intervene in the event of a failure of the main system, it must operate in an electrically stable environment, isolated from the rest of the control infrastructure.

In short, by integrating high-resolution visualization, automated alerts, a resilient data pipeline and an independent safety controller, the monitoring system described in Section 2.4 ensures that MiniStor demonstrators achieve both operational transparency and robust fault management, meeting the stringent reliability requirements of demonstration projects.

2.5. Integration with the IoT platform

The platform previously described unifies data collection, forecasting and high-level optimization into a single, coherent service. It exposes a REST API that both the field PLCs and cloud services use to exchange information every minute.

Field-to-cloud data ingestion

Each demonstrator's PLC packages its operating variables such as sensor readings (temperatures, flow rates, pressures), component states (valve and pump positions) and calculated thermal flows into a JSON payload with timestamps and standardized variable names. The PLC then issues an HTTP POST to the platform API endpoint at one-minute intervals. This guarantees a continuous, up-to-date stream of real-time and near-real-time data for monitoring and analytics.

Forecast and optimization outputs

In parallel, the Azure-hosted forecasting and optimization modules push their results into the same API:

- Forecasting service retrieves weather data from Weatherbit[1], runs pre-trained neural-network models for each site and generates hourly predictions of solar irradiance, electrical power and thermal demand.
- High-level optimization module consumes both historical measurements and forecast data to compute optimal setpoints and schedules that minimize costs or maximize self-consumption, subject to system constraints.

Both tools use historical data and external meteorological information, along with real-time sensor readings, to anticipate solar production, thermal demand and system conditions. Their outputs are posted via the same REST interface, ensuring the platform database always contains synchronized measured, forecast and optimized values.

Cloud-to-field coordination

The PLC can also retrieve high-level recommendations through HTTP GET requests from the Azure-based service through forecasts and optimization outputs. At each control cycle, the low-level controller queries the API to fetch the latest setpoints or operating strategies. It then locally evaluates whether current system conditions allow execution, thereby enforcing a hierarchical control scheme in which:

- the cloud proposes global strategies and
- the PLC implements only those commands that are feasible given real-time states and safety constraints.

This integration ensures intelligent, efficient and adaptive operation of the demonstrator, in alignment with the modular and hierarchical architecture of the HEMs previously described.

Azure service architecture

The forecasting and optimization services run on a virtual machine in Microsoft Azure, orchestrated by a main scheduler script, which also launches the different secondary processes (Figure 21). Its operation is based on a modular architecture programmed to run periodically, according to time configurations defined by the user.:

- Weather data ingestion: periodic downloads from the Weatherbit API.
- Forecast generation: execution of neural-network models for each demonstrator, with hourly results (irradiance, electrical power and thermal power) stored in SQL Server or forwarded to the IoT API.
- Forecast deduplication: query module prevents duplicate entries by checking existing timestamps before inserting new forecasts.
- Optimization runs: batch simulations that compute setpoints, then publish results to both SQL Server and the IoT API. The module evaluates optimal energy operation strategies based on the forecasts and the current state of the system. It simulates the entire process: from data reading, through the execution of the optimization,

[1] Weatherbit.io, "Weather API Documentation," 2024 [Online]. Available: https://www.weatherbit.io/

All modules log their activities for traceability and debugging. Azure's scalable infrastructure ensures the system can handle additional demonstrators, expanded tasks or more frequent scheduling with minimal configuration changes. This architecture ensures that the system operates in a robust, adaptable and continuous manner, with the capability to integrate with other energy and control platforms.

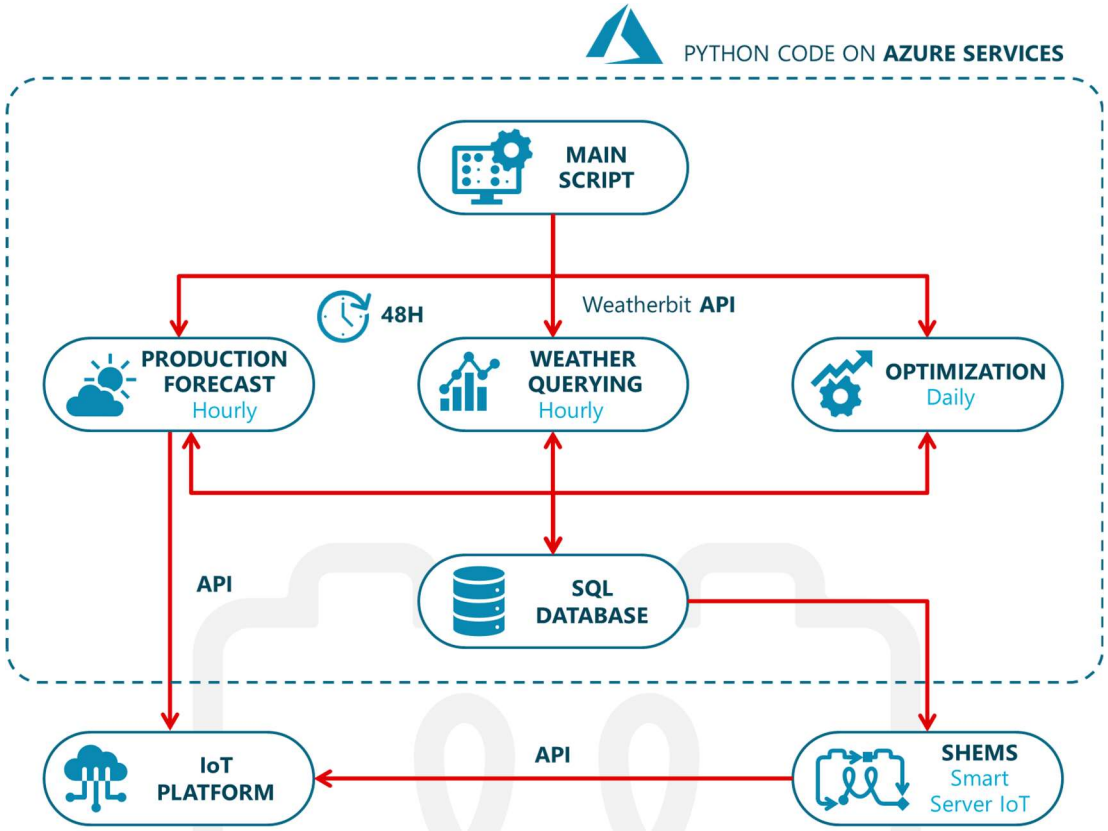


Figure 21: Cloud service architecture for IoT data, forecasting and optimization

3. Implementation and validation of control strategies and operational modes

Section 3 presents selected examples of how the MiniStor control strategies and operational modes originally defined in deliverable D2.4 and elaborated in D5.1 were deployed, tested and refined across all demonstration sites. Rather than repeating every mode in full, this section focuses on representative scenarios that illustrate the system's behaviour in factory acceptance tests, on-site commissioning and early operation. These activities were carried out continuously in close collaboration with system integrators, installers and software developers: beginning at the factory's final assembly stage, extending through each site's commissioning phase, and evolving via iterative fine-tuning to address the unique technical and environmental conditions encountered at Thessaloniki, Kimmeria, Sopron, Santiago de Compostela and Cork. By highlighting these hands-on validation steps and site-specific adaptations, this section demonstrates how the SHEMS achieved robust, reliable performance under real-world conditions. For a complete list of baseline modes and control logic details, readers are referred to deliverables D2.4 and D5.1.

3.1. Low-level control

3.1.1. Validation of control strategies and operation modes in the pilots

The low-level controller executes rule-based logic to manage solar generation, thermal storage (PCM/TCM), heat pumps and loads according to predefined modes. Validation relied on real-time SCADA and Grafana data, confirming that each mode transitions correctly and the system responds safely to varying environmental and demand conditions. As mentioned, due to the vast number of modes and transitions between them, only key operational modes are illustrated here.

Initial validation took place in CARTIF's lab, using a setup to cycle through all modes and transitions. The validation was later carried out in PSYCTOTHERM's premises during the finalisation of the prototypes' manufacturing. Subsequent site deployments repeated these tests under real conditions and included fine-tuning procedures to adapt to the specific conditions of every demo, as explained in next subsections (3.1.2.1-3.1.2.4). Commissioning activities related to the SHEMS included functional and fault tests and performance tuning to verify mode transitions, system's robustness and safe fall-back.

It is to be noted that the figures displaying monitoring results included in following pages have been directly taken from the Grafana platform and therefore contain the notation system of variables as defined in the MiniStor P&ID (Piping and Instrumentation Diagram). In all cases, the first row of legend entries (left-aligned) correspond to those represented in the left vertical axis. Similarly, the second row of legend variables (right-aligned) correspond to those represented in the right vertical axis.

Despite the meaning of the key variables in each case are explained in the text descriptions, a complete version of the variable map has been included in the Annex (section 5) of the present document.

TCM CHARGING CYCLE

Figure 22 shows the start of the TCM pressurization sequence. When the buffer tank temperature (*HTF_TY_3*) reaches 65 °C, the system enters Mode 1 (*MODE_TCM_UNIT*). At this point, the logic opens the appropriate valves and starts the transfer pumps, driving heat into the thermochemical reactor. As the reactive material desorbs ammonia, the reactor pressure (*NH3_PT_1*) climbs sharply.

The figure captures the coordinated interplay of temperature, pressure and control-signal trajectories during this critical start-up phase.

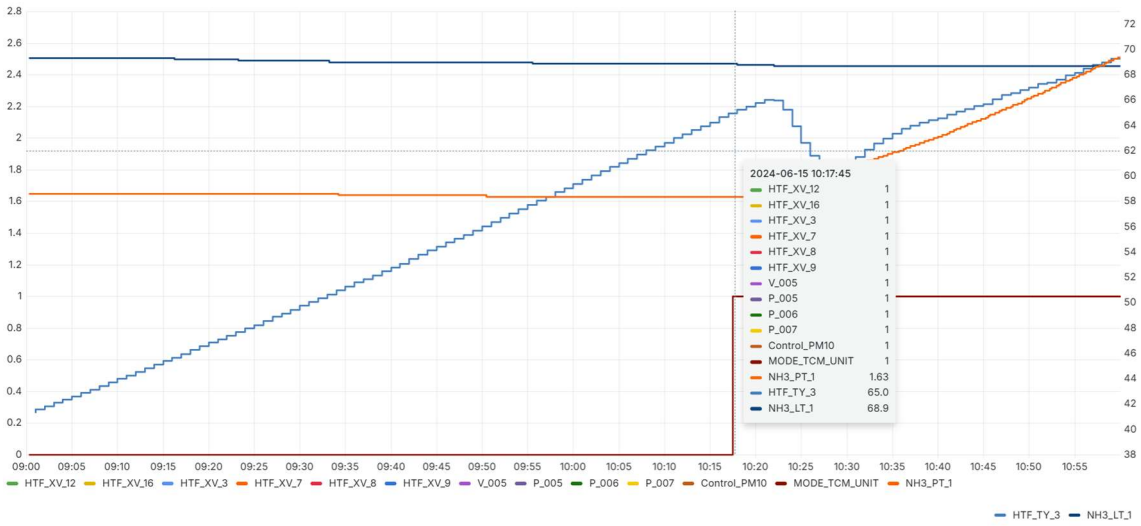


Figure 22: Example of the first stage of the TCM charging start-up visa solar heat – Thessaloniki demo site

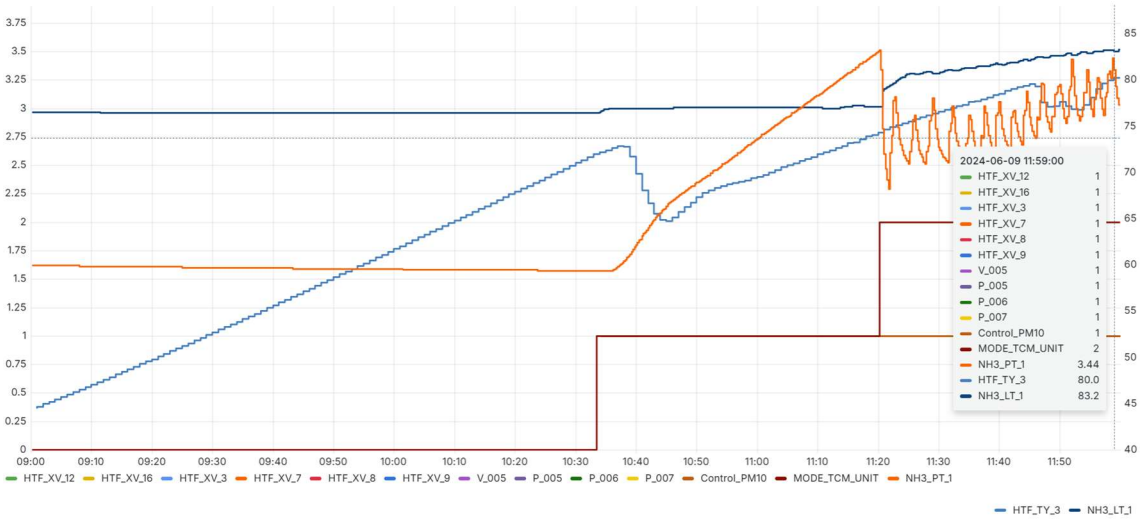


Figure 23: Example of the TCM charging process (ammonia cycle) – Thessaloniki demo site

Figure 23 illustrates the transition into Mode 2 (*MODE_TCM_UNIT*) once the reactor pressurization phase ends. The relevant actuators remain active, with the main difference being that during this phase the heat pump is activated (*Control_PM10*[2]) to supply additional thermal energy to also charge the PCM-based storage systems for space heating and DHW. Upon compressor start-up, reactor pressure (*NH3_PT_1*) experiences a sharp drop as ammonia is driven toward the condenser and reaches the reservoir, in turn increasing the ammonia level (*NH3_LT_1*). The controller continuously monitors this pressure and enforces predefined upper and lower pressure bounds: if pressure falls below the minimum threshold, the system automatically reverts to Mode 1 to re-pressurize the

[2] It is to be noted that, in Grafana, depending on the query conditions the shift from one value to a different one is not visually displayed, as in the case of *Control_PM10* (from 0 to 1), where the trigger of the control signal corresponds exactly with the change from Mode 1 to Mode 2.

reactor and restore safe operating conditions. Continuous SCADA and Grafana traces confirm that all actuators remain correctly sequenced and that pressure excursions stay within allowable limits throughout the charging cycle.

TCM DISCHARGING CYCLE

When thermal energy stored in the TCM is required to charge the cold PCM units, the controller switches to Mode 3, initiating the reactor depressurization phase, and subsequent shift to Mode 4. During this process

- Heat release from the TCM: valves directing hot ammonia vapour from the reactor to the DHW PCM battery or to the ambient reject loop open, allowing the exothermic reaction to transfer heat as needed.
- Cold delivery to PCM: simultaneously, the evaporator circuit redirects the chilled ammonia flow toward the cold-PCM loop, lowering its temperature until the PCM reaches its phase-change setpoint of 11 °C.

Figure 24 displays a typical sequence of TCM discharging at the Thessaloniki site. The mode signal (*MODE_TCM_UNIT* in dark green) rises to 3 when depressurization begins, and the reactor pressure decreases (*NH3_PT_1*, dark blue). Subsequently, the mode changes to 4, the ammonia level starts being released which results in the decrease of the reservoir level (*NH3_LT_1*), and the cold-PCM temperature sensor (*PCM_TY_3*, purple) drops toward the 11 °C threshold.

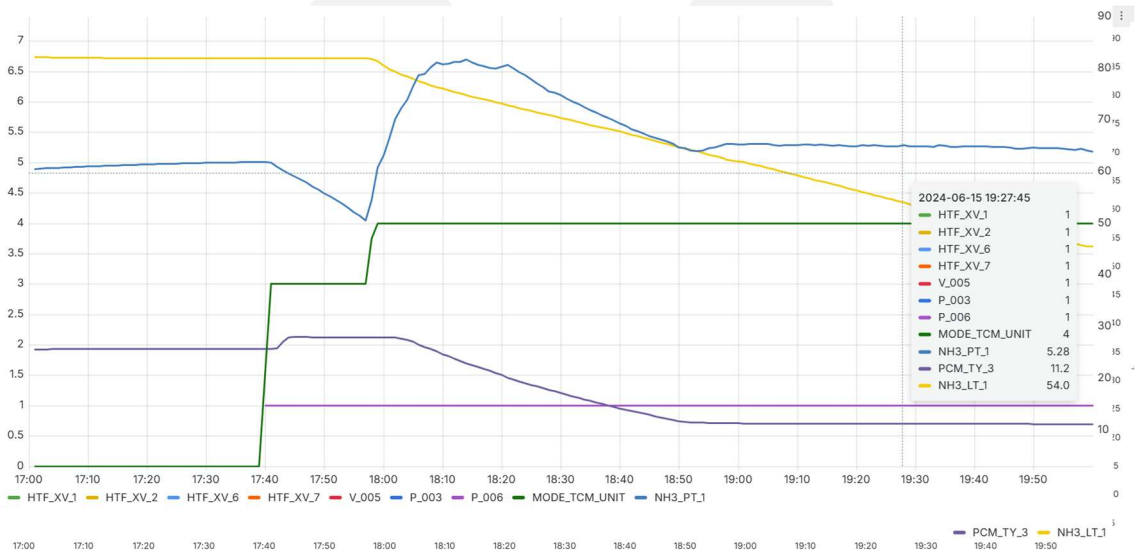


Figure 24: Example of the TCM discharging process (ammonia cycle) – Thessaloniki demo site

HP USE DURING THE TCM CHARGING CYCLE & PCM DYNAMIC USE

As previously mentioned, during TCM charging processes the heat pump is commonly used to increase the condenser outlet temperature, thereby boosting the thermal gradient and facilitating the charging of the heat PCM storage. At the same time, the PCM batteries may be charged or discharged simultaneously depending on the users' thermal demands. Figure 25 illustrates this combined operation at the Kimmeria site. The system initially operates with the PCM being discharged; i.e. with heat being demanded in the building for space hating, and therefore the corresponding pump is activated (*P_001*). This causes the decrease of the temperature in the heating PCM (*PT100*). Subsequently, as soon as the TCM charging process begins (*MODE_TCM_UNIT* = 2),

the controller switches on the HP (*Control_PM10*) and activates the necessary actuators to redirect the thermal flow towards the PCM, initiating its charging (valves *HTF_XV_10*, *HTF_XV_11*, pumps *P_005*, *P_007*). The temperature at the HP condenser outlet (*HTF_TY_4*) exhibits a hysteresis of approximately 5 °C, enabling stable operation around the PCM's phase change point. This figure also shows a common dynamic sequence during the first charging stage, where the TCM mode may briefly revert to Mode 1 and back to Mode 2 in response to small pressure or temperature excursions in the reactor, illustrating the controller's real-time adaptation to TCM conditions.

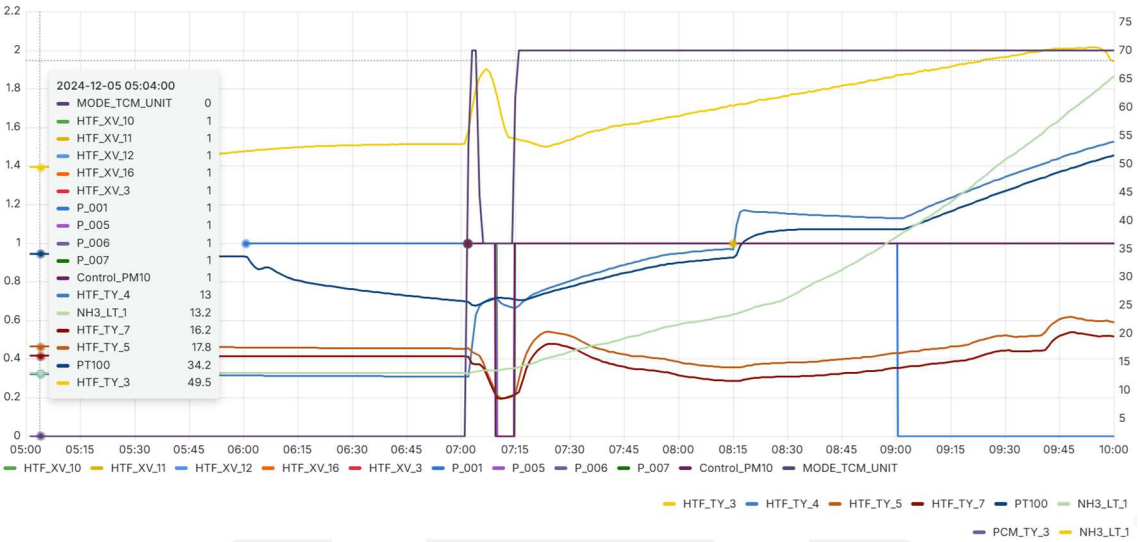


Figure 25: Combined PCM discharge, TCM charging and heat pump operation for PCM charge – Kimmeria demo site

HP USE IN THE SOLAR LOOP (SANTIAGO DE COMPOSTELA)

In Santiago de Compostela demo site, the particular configuration of the system includes a heat pump operated in combination with the solar loop. In this case, the control is based on a combined strategy of scheduling and temperature setpoints. Activation is programmed for the early hours of the day and depends on the solar panel temperature reaching a suitable threshold. Once this condition is met, the heat pump is activated to support the system's thermal charging (*Yutaki_Control_Circuit_RS*). In Figure 26 it can be observed that, upon activation with a 60 °C setpoint, the system reacts quickly, the temperature increases (*Yutaki_Water_outlet_unit_temperature*) and the TCM charging process is initiated as described in previous cases. Additionally, the solar-loop's heat pump control considers the pressure in the TCM (*NH3_PT_1*), dynamically adjusting the temperature setpoint: if pressure increases sharply, especially at the beginning of the process, the temperature setpoint (*Yutaki_Control_Circuit_1_W*) is reduced to prevent overpressure and maintain operation within safety limits.

Figure 26: Example of operation of the HP in combination with the solar loop. Ministor's container (top) and solar field's (bottom) monitoring – Santiago de Compostela demo site

HOT PCM BATTERY CHARGE VIA SOLAR HEAT

There are situations in which, due to overpressure, the TCM being already fully charged or high temperatures coming from the solar installation, the controller decides to charge the PCMs directly via solar heat. During periods of high solar irradiance, by activating the corresponding actuators, the thermal energy stored in the buffer tank is redirected to the PCM units. The pump responsible for this transfer (*P_006*) operates intermittently, depending on the buffer tank temperature (*HTF_TY_3*). The figure below (Figure 27) shows characteristic spikes caused by the on/off operation of the pump recirculating heat from the solar panels (*Deltasol_Temperature_sensor_1*) to the tank. This pump is managed by a dedicated solar controller, separate from the main MiniStor system controller. As a result of this direct charging process, the PCM temperature reaches its phase change point, around 58 °C (*PT100*).

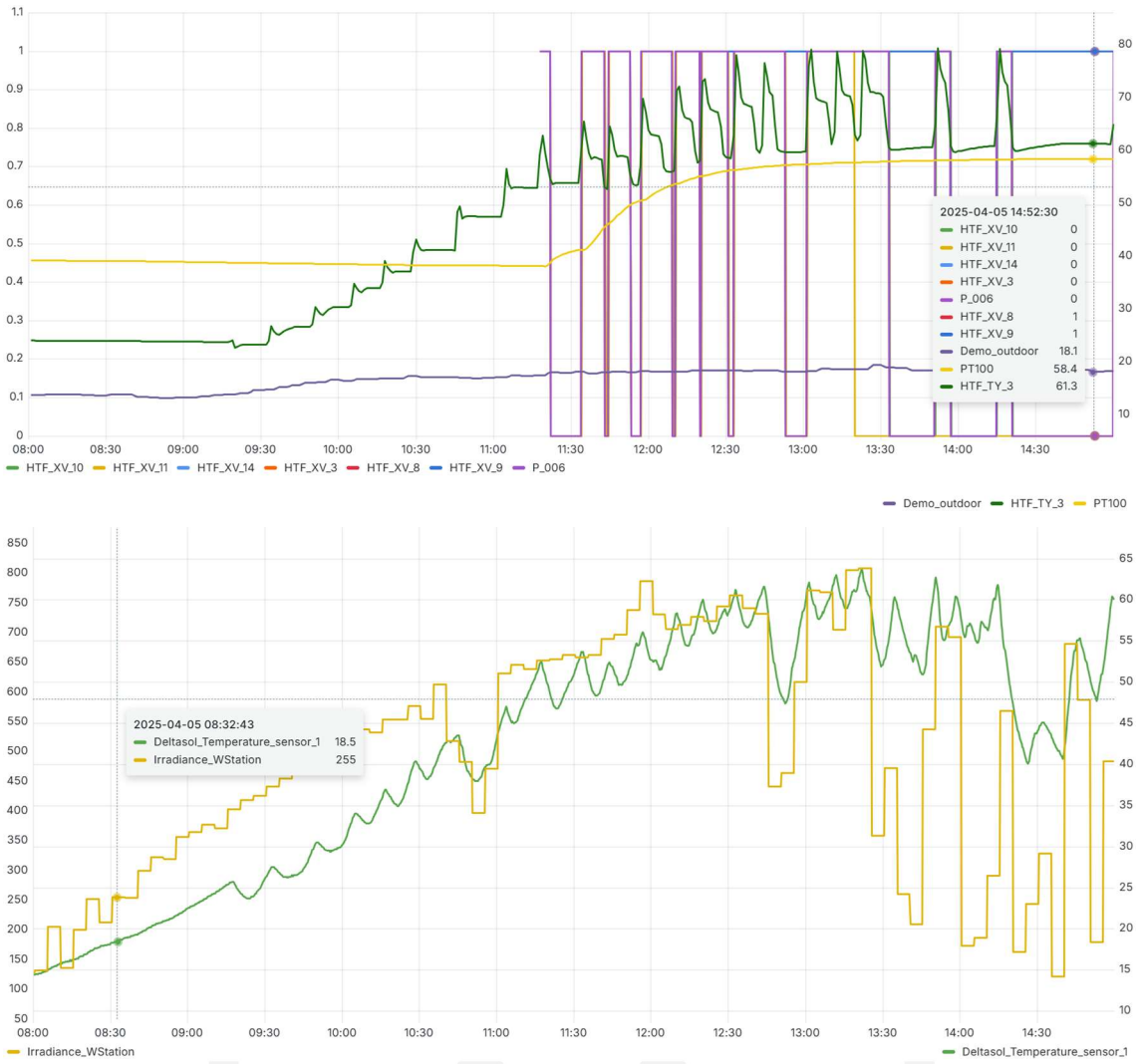


Figure 27: Example of hot PCM battery charge via solar heat. Ministor's container (top) and solar field's (bottom) monitoring – Sopron demo site

HEAT DISSIPATION USING OUTDOOR FAN COILS

In situations where heat dissipation is required, particularly when the temperature exceeds the operational limits of the system components, the controller resorts to the use of fan coil external units. In the specific case depicted in Figure 28, during the charging process of the PCM the material has already reached its phase change temperature but is not yet fully charged. The heat supplied by the HP (*HTF_TY_4*, redirected to the PCM through the corresponding actuators) is too high to be efficiently used, reaching values up to 66 °C. Since this exceeds the usable limit, the system automatically diverts part of the thermal energy to the fan coil (*V_005*). In this way, the excess heat is dissipated and the outlet temperature is regulated, allowing the PCM charging process (*PT100*) to continue in a controlled manner and under more stable, suitable conditions.

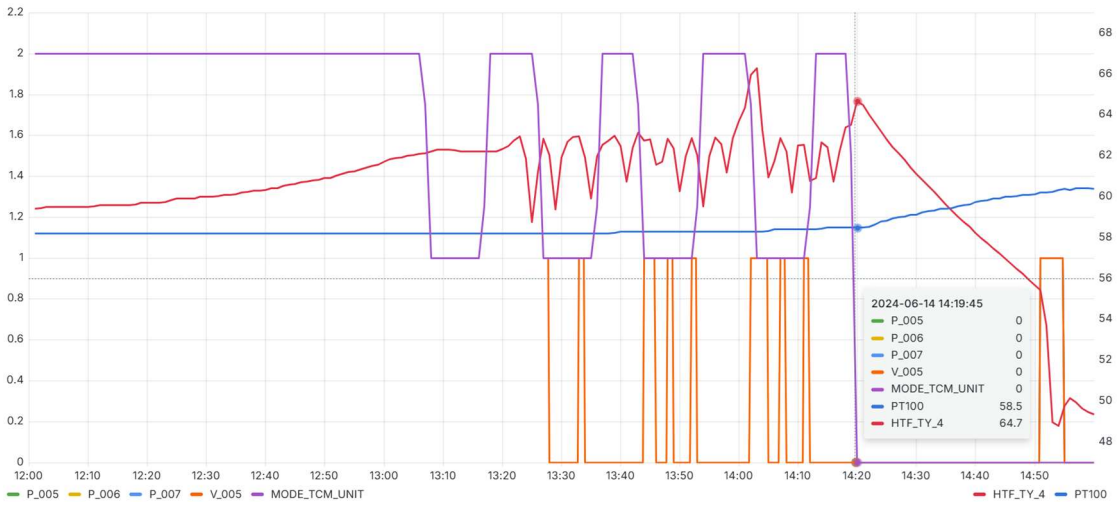


Figure 28: Example of heat dissipation using outdoor fan coils – Thessaloniki demo site

3.1.2. Specific control adjustments for each demo site

Commissioning tests of the various MiniStor system demonstrators revealed that low-level control strategies needed to be adapted to the specific conditions of each site. Although the system is based on a shared control architecture, specific adjustments were implemented at each location to ensure reliable, safe and optimized performance, tailored to the local climate, demand patterns and the configuration of installed components.

These adaptations involved refining control modes to better suit each use case, recalibrating key sensors to improve measurement accuracy and reinforcing the system's resilience through enhanced fault-handling routines. Operating schedules were also customized to reflect the actual usage patterns of each building. In parallel, additional protection mechanisms were embedded directly within the PLC to safeguard critical components such as compressors, pumps and valves.

The following subsections describe in detail the technical specifications and the modifications applied to each demonstrator.

3.1.2.1. Specific control adjustments for Thessaloniki pre-pilot

The Thessaloniki demonstrator was the first demo site to receive a fully integrated MiniStor system, becoming a key environment for the initial functional tests under real-world conditions. This installation enabled early evaluation of operating modes, sensor reliability and control logic, serving as a foundational reference for the other pilot sites.

One of the main challenges identified during this phase was the need to incorporate a **new safety control mode for the solar loop**. This mode was designed as a fall-back mechanism in case of failure of the primary controller for the solar installation, automatically activating to prevent dangerous overheating of system components. This critical functionality was implemented directly in the PLC and was triggered upon detection of excessive temperatures in the solar loop, significantly enhancing the operational safety of the system. Figure 29 displays a real situation where this mode had to be activated to release excess heat into the ambient.

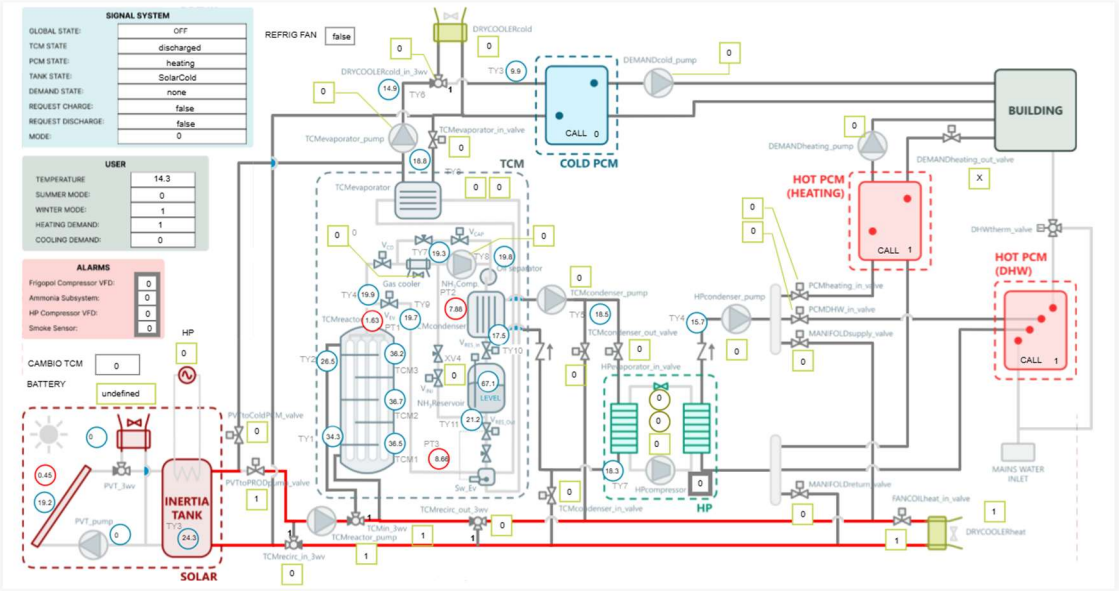


Figure 29: Example of the additional safety mode for the solar loop

During this stage, it also became necessary to recalibrate the ammonia level sensors, as inaccurate readings were hindering the completion of the thermochemical reaction. This intervention proved essential to ensure the system could operate efficiently. Additionally, the logic governing the operating modes was fine-tuned to better respond to the variability of available solar resources. This experience was later shared with other demonstrators to improve system responsiveness across different environments.

Various operational issues were also identified and resolved, leading to improved control robustness. Specific alarms were implemented to detect compressor lockouts due to overpressure and pressure and temperature constraints were adjusted during transitions between operating modes. These improvements were applied directly in the PLC to minimize failures and protect critical components.

Fixed thermal demand schedules (for heating and cooling) were defined based on the building's usage profile, optimizing interaction with the PCM modules. On a seasonal control level, a dedicated variable was introduced to distinguish between summer and winter operating modes, improving the system's response to changing HVAC needs.

3.1.2.1. Specific control adjustments for Sopron demo site

The Sopron demonstrator was developed under unique conditions, shaped by its prior validation phase carried out at the facilities of project partner EMI. At this location, functional and quality tests were conducted before the system was transferred to its final installation site. These tests made it possible to identify and resolve various operational issues, serving as a reference for validating the low-level control logic.

During the testing phase, the system was configured with a simulated solar input panel and a thermal load designed to mimic the behaviour of a real building. The low-level control was adjusted to execute multiple charging and discharging cycles, both in manual and automatic modes. Subsequent analysis confirmed that the thermal and energy behaviour trends matched the design expectations.

At this stage, adjustments were also made to the control code and SCADA interface to adapt them to the specific test parameters defined by EMI. These modifications provided a more flexible and detailed system configuration, allowing it to respond effectively to the various testing scenarios.

Once validated, the prototype was transferred to its final location in Sopron. At this demonstrator, it was necessary to adapt the system parameters to more extreme outdoor temperatures, significantly lower than those at other sites. The start-up thresholds for the HP and compressor were modified, along with the PLC's operational limits, to prevent low-temperature lockouts and ensure reliable performance under harsh winter conditions.

Additionally, a new mode was introduced to preheat the inlet and outlet of the HP in cases where temperatures fall outside its optimal operating range. This mode is essential to restore the system to suitable conditions, especially before initiating the TCM charging, which relies on the proper functioning of the HP.

In Figure 30, a typical activation of this mode can be observed: the system halts its normal operation and enters a preheating phase, during which it uses available energy from the solar buffer tank and the PCM storage to raise the temperature of the fluid entering and exiting the HP. This ensures that the HP can operate within safe and efficient parameters. Without this preheating step, the system would be unable to charge the TCM, as the HP would remain locked out due to suboptimal thermal conditions.

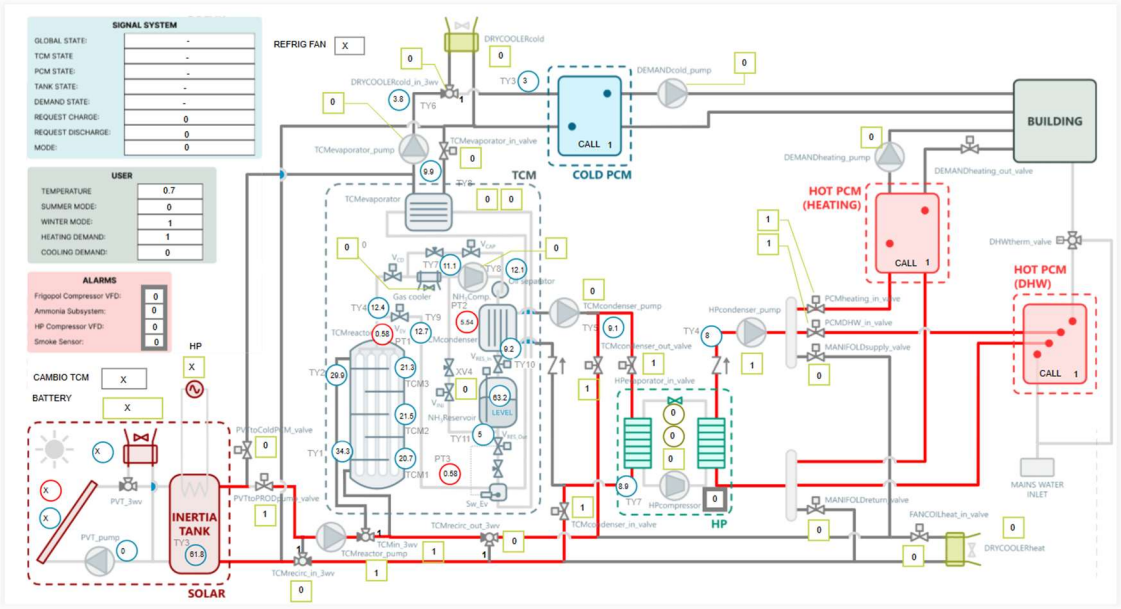


Figure 30: Example of the additional preheating mode of the HP

3.1.2.2. Specific control adjustments for Kimmeria demo site

Unlike other demonstrators, in the case of Kimmeria no PVT or solar thermal panels provided by the consortium were installed; instead, the system was directly connected to the building's existing energy centre.

From a low-level control perspective, one of the main adjustments made was the specific management of solar production. A fixed schedule was defined for the operation of the solar pumps,

restricting their activity to hours of high solar incidence. This strategy aimed to maximize thermal capture and avoid inefficient cycles during periods of low radiation, thereby contributing to a more stable and efficient system operation.

As shown in the following graph (Figure 31), which illustrates normal system operation during a typical day, a significant increase of the temperatures can be observed around 7:00 AM (*HTF_TY-1-8*), corresponding to the activation of the pumps used to charge the solar buffer tank.

This sharp rise reflects the moment when solar energy becomes available and the system begins actively transferring heat into the thermal inertia tank. The timing of this event is aligned with both the sunrise and the predefined control schedule, ensuring that the system takes advantage of the first hours of solar radiation.

It is to be noted that, during the commissioning phase it was decided to apply a seasonal temperature setpoint, so that the temperature threshold that triggers the system (from the solar system, *HTF_TY-1-8* in the figure) varies depending on the season: in winter, it is lowered to 57.5 °C, while in summer it is raised to 65 °C. This seasonal adjustment is intended to increase the system's operating hours, particularly in colder months when solar input may not reach the standard setpoint. Although the system is less efficient at a lower temperature, it can still operate and contribute to heat storage under these conditions.

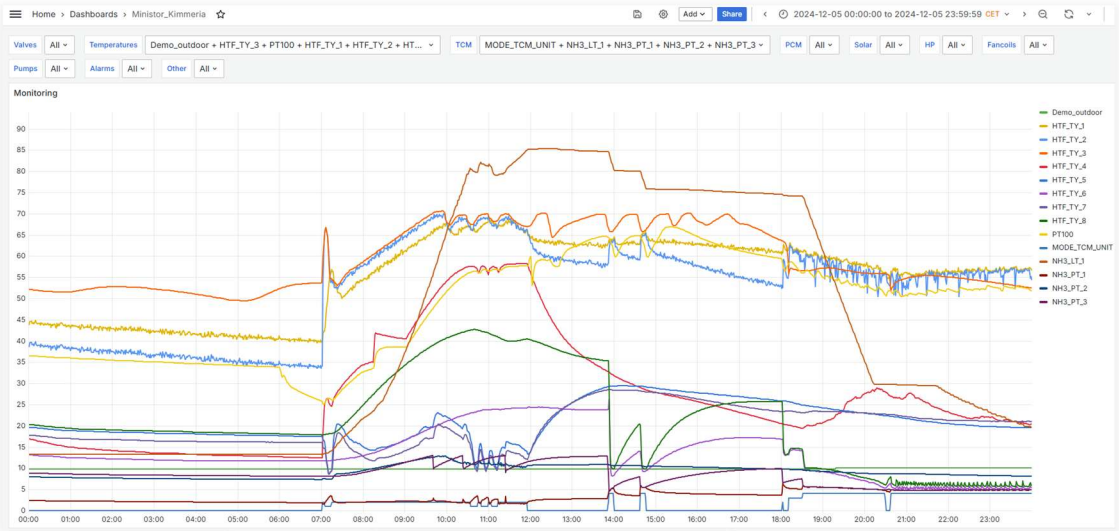


Figure 31: Example of the Grafana dashboard at Kimmeria demo site

Regarding thermal demand, differentiated schedules were established based on the season and time of day, adapting energy delivery to the building's actual usage patterns.

Although the Kimmeria demonstrator's setup differed, lacking some of the external communications and advanced generation hardware present elsewhere due to the pre-existence of a solar thermal system, it provided a clean, well-defined environment for verifying core control modes. Its focused scope enabled rapid validation of the hierarchical, modular approach of the MiniStor system without detracting from the overall rigor applied across all sites.

3.1.2.3. Specific control adjustments for Santiago de Compostela demo site

The demonstrator located at the University of Santiago de Compostela (USC) features one of the most complex configurations of the MiniStor system, integrating not only thermal storage but also a commercial air-to-water HP and a dedicated solar field installed specifically as part of the project.

During the commissioning phase, multiple adaptations were implemented at the low-level control layer, tailored to the system's specific requirements and operational context. One of the most significant developments was the inclusion of autonomous control logic for the HP, programmed directly into the PLC. This logic allows the HP to activate only when solar conditions are favourable, meaning that the PVT solar collectors reach a temperature setpoint that varies throughout the year.

The defined operational goal is to heat the solar buffer tank up to 60 °C, thus establishing a safe condition for charging the system. If the temperature drops below this setpoint minus a hysteresis margin, the HP shuts off. Additionally, this control condition is restricted to central hours of the day, ensuring the setpoint is met under sustained solar conditions and preventing undesired activation during night-time hours.

Figure 32: Example of the monitoring logic in Grafana for the additional HP in Santiago demo site

A unique thermal demand control strategy was also introduced, setting this demonstrator apart from the others. Thus, at USC demo site the MiniStor system does not respond directly to user demand, but instead, it only delivers energy when the PCM batteries are fully charged or exceed a certain temperature threshold. This demand remains continuously active through an external device, ensuring the system operates only when it is ready to supply energy, thereby reducing wear and improving efficiency.

Another key aspect was the implementation of BACnet communication (Figure 33), which made all system data accessible to the building's internal network. This not only facilitates real-time monitoring, but also enables the system to integrate with other existing control platforms at the USC facilities, meeting interoperability requirements.

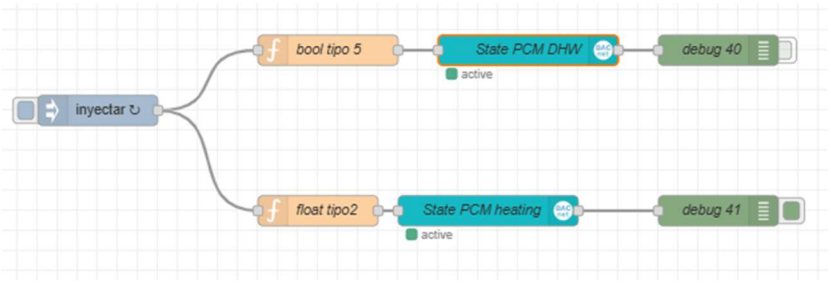


Figure 33: Node-RED flow for BACnet demand signal parsing and routing to state nodes

At the protection level, the system inherited and implemented all the measures developed in the demonstrators previously commissioned, including alarms for compressor lockouts (such as those caused by overpressure in the reactor), temperature and pressure restrictions during mode transitions, as well as delay routines and automatic reset mechanisms in response to repeated faults.

3.1.2.4. Specific control adjustments for Cork demo site

The Cork demonstrator was the last to be commissioned, which allowed it to incorporate all the improvements, updates and adjustments previously implemented in the other demonstrators. During the initial operating phase, the control system had to function without the availability of the TCM subsystem due to ongoing repair work on the ammonia compressor. Nevertheless, the system was able to store the necessary thermal energy using only solar production and the PCM storage system, successfully covering the heat demand required for system preheating.

During this phase, network issues were identified with the household router, specifically related to blocking remote access to the PLC and port forwarding management. To resolve this, the router installed inside the MiniStor container was reconfigured to operate as a network switch, while the building's main domestic router was used as the primary network access point.

In terms of operation, the demonstrator follows a predefined schedule coordinated with the auxiliary heating system, which in this case is a gas boiler, working in an integrated manner with the MiniStor system. The gas boiler is scheduled to turn on at 07:00 (GMT) and off at 10:00 (GMT) and then back on at 16:00 (GMT) and off again at 22:00 (GMT). As part of the control strategy, MiniStor has been programmed to supply heat during the two hours preceding each gas boiler operating period. This is designed to maximize the use of renewable sources and minimize gas consumption. This operating logic enables efficient integration between locally available renewable energy and the auxiliary thermal generation system, optimizing both energy performance and sustainability.

3.2. High-level control

The High-Level Controller (HLC) is a key component of the MiniStor EMS, designed to plan and optimize the operation of household-scale energy devices such as thermal storage systems (PCM and TCM), the HP, the solar system and building loads. Its main objective is to maximize the use of local renewable sources, reduce operational costs and ensure thermal comfort for users.

During the initial phase of the project, as documented in deliverable D5.1, it was not possible to complete the software design of the HLC, since real operational data was required to analyse the system's degrees of freedom and define optimization algorithms tailored to the actual dynamic behaviour. Given that the MiniStor system features complex and constrained operational limits, the

HLC design had to identify which conditions could be relaxed without compromising operational safety, enabling effective optimization.

In this context, two technical approaches were studied for the development of the HLC, depending on the type of model available:

- Optimization using exact physical model: the plan includes the use of optimization algorithms developed in Python, integrated with energy models built in TRNSYS[3]. The intention behind using TRNSYS is to leverage its established thermal simulation capabilities to generate input-output datasets under realistic operating conditions. These simulations serve as a basis for tuning and validating control strategies in the absence of complete physical equations.
- Optimization using simplified and machine-learning models: based on the data and knowledge acquired during the normal operation of the system, simplified (linear or non-linear) mathematical models are identified obtaining equations to calculate and model the main variables required for controlling the Ministor system.

In both cases the optimization will use genetic-based heuristic algorithms that run the existing models to calculate the value of the objective function to be minimised.

Based on the MiniStor system conditions, it was ultimately found that the first option was not suitable for MiniStor's intelligent controller. It requires a parametrization of the TRNSYS model based on the monitored data, and it has been proven a not feasible approach for the TCM due to the absence of a clear and consistent functional relationship between observable variables and the system's actual dynamics. Thus, although efforts were made to correlate the integral of the temperature difference with the ammonia charge level, the results revealed high dispersion and lack of coherence, particularly during the charging mode. As can be seen in Figure 34, Figure 35, Figure 36 and Figure 37, where the thermal energy injected to the TCM is represented against the NH_3 (ammonia) level, multiple thermal trajectories lead to similar charge states, indicating that the relationship is neither unique nor deterministic.

The behaviour of the TCM is inherently complex, influenced by thermal, chemical and hysteresis phenomena that cannot be fully captured by external sensors alone. The temperatures of the thermal fluid provide limited information unless internal variables such as pressure or the thermal distribution within the reactor are also considered. Additionally, the charging and discharging cycles occur under very different boundary conditions, which prevents data homogenization and reduces its statistical value.

[3] Klein, S.A. et al, 2017, *TRNSYS 18: A Transient System Simulation Program*, Solar Energy Laboratory, University of Wisconsin, Madison, USA, http://sel.me.wisc.edu/trnsys

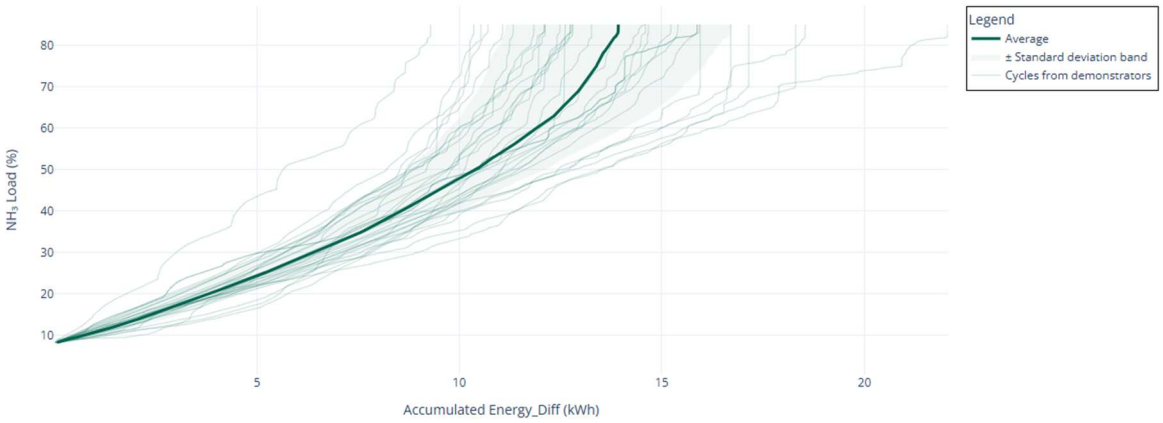


Figure 34: Correlation analysis of multiple charging processes – Accumulated energy difference in the TCM reactor vs. NH_3 storage level

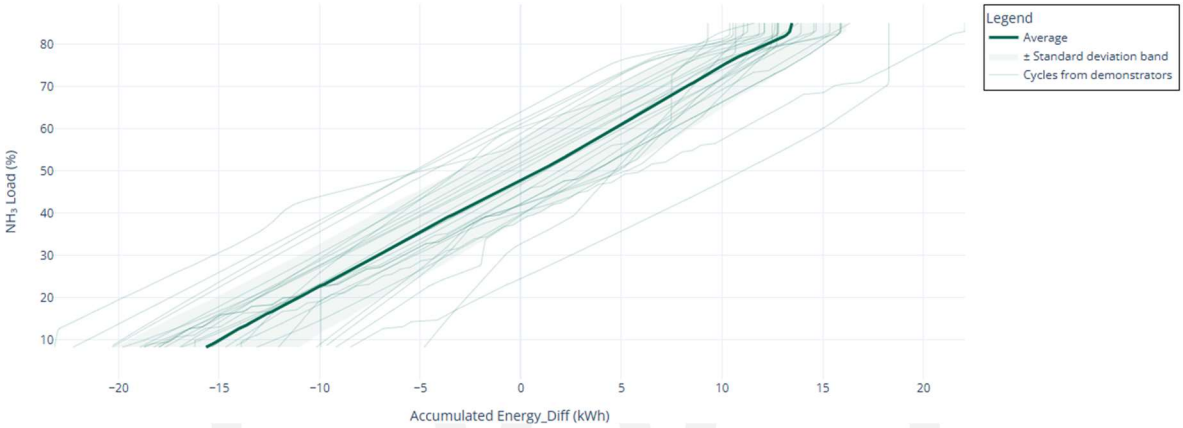


Figure 35: Correlation analysis of multiple discharging processes – Accumulated energy difference in the TCM reactor vs. NH_3 storage level

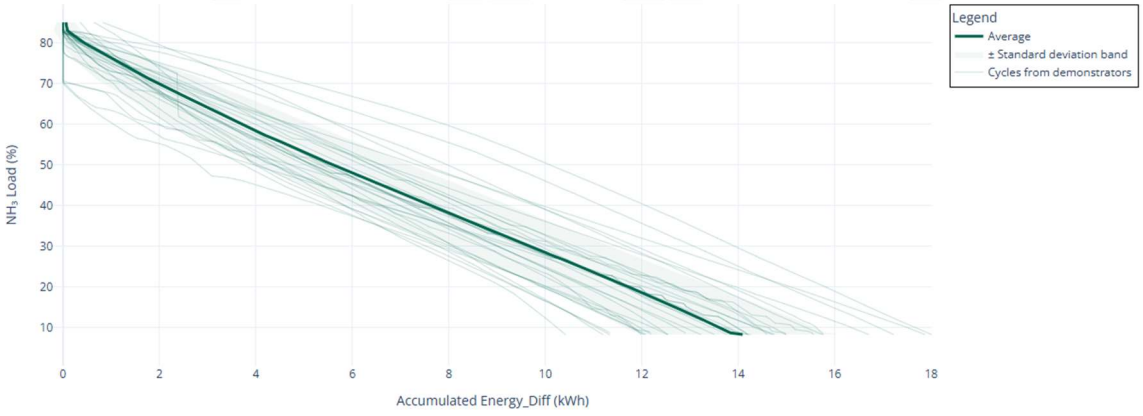


Figure 36: Correlation analysis of multiple discharging processes – Accumulated energy difference in the TCM evaporator vs. NH_3 storage level

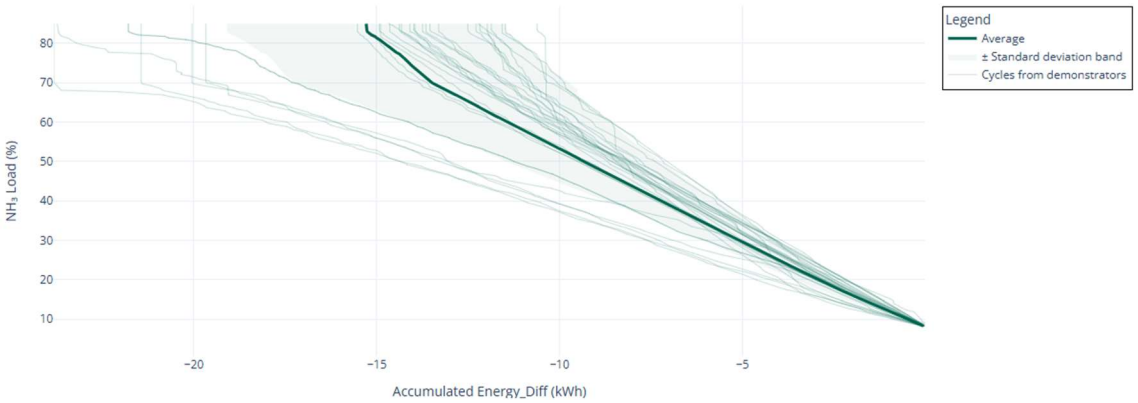


Figure 37: Correlation analysis of multiple charging processes – Accumulated energy difference in the TCM condenser vs. NH_3 storage level

Finally, although a considerable number of cycles has been processed, it is not sufficient to train a robust and generalizable model. The lack of data volume, the heterogeneity of operating conditions and the limited resolution of some signals prevent the reliable application of machine learning techniques.

Taken together, these factors demonstrate that a purely data-driven strategy is not suitable for parametrizing and modelling the TCM in TRNSYS. The system requires a more structured approach, based on analytical or hybrid models that incorporate physical knowledge of the process, allowing its real behaviour to be captured more accurately and reliably.

As a result, a grey-box model-based approach has been chosen. This strategy allows for a more faithful representation of the internal dynamics of the MiniStor system's components. It combines simplified physical equations with parameters calibrated from available technical knowledge, facilitating integration into control and optimization schemes. The following sections describe the models developed for the main thermal components; i.e. the HP, the PCM storage system and the TCM storage, as well as the formulation of the energy optimization problem that coordinates them.

3.2.1. HP grey-box model

The model developed for the HP represents its thermal behaviour within the MiniStor system, with the objective of predicting energy exchanges and interactions with the thermal storage subsystems (TCM and PCM). The approach is based on a simplified yet physically grounded energy balance that simulates the HP's operation in transient mode, accounting for the evolution of temperatures and thermal flows over time.

The model distinguishes two main states: the active operation of the HP ('on' mode) and its shutdown phase. In the active phase, thermal power at the condenser is calculated based on an estimated Coefficient of Performance (COP), which depends on the saturation temperatures of the condenser and evaporator, assuming a limited thermodynamic efficiency. Based on this power, the temperature increase in the condenser circuit is estimated, considering thermal losses to the environment, modelled as proportional to the temperature difference between the condenser and outdoor air.

On the other hand, the evaporator outlet temperature is continuously adjusted toward a target that depends on the inlet temperature of the thermal fluid and the absorbed thermal load. This adjustment is governed by a relaxation model representing the system's thermal inertia and the resistance to heat exchange in the evaporator.

In the shutdown phase, the model simulates the thermal dissipation of the condenser through natural heat losses to the surroundings. This dissipation is modelled over a limited number of time steps, after which it follows an exponential relaxation toward ambient temperature, reflecting passive cooling after the cycle ends.

The model also enables the calculation of energy flows absorbed by the evaporator, delivered by the condenser and electrical consumption, using the temperature difference between the inlet and outlet of the thermal fluid, its mass flow rate and specific heat capacity. In this way, the model remains consistent with the physical principles of a HP and is suitable for integration into a full energy simulation of the system.

As exemplified in Figure 38, the energy results obtained with the analytical model of the HP show a high level of agreement with the actual values measured in the demonstrator. The thermal energy consumed by the HP during active operating periods is nearly identical in both cases, indicating that the model accurately reproduces the switching behaviour and the electrical power associated with the compressor.

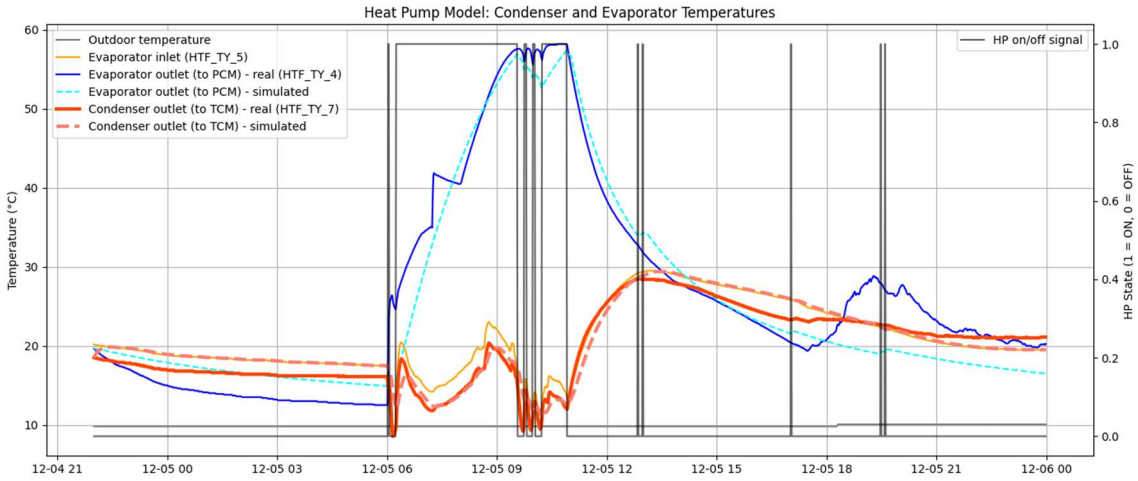


Figure 38: Validation example of the HP model using real measurements

Regarding the energy absorbed at the evaporator, the model correctly captures the thermal transfer from the environment to the refrigeration circuit, with only a small deviation from the actual value. This reflects an accurate estimation of the temperature difference between the evaporator's inlet and outlet, as well as the mass flow rate used.

The energy delivered at the condenser, representing the heat transferred to the PCM subsystem, was evaluated using two approaches within the model: one based on the direct balance between absorbed heat and electrical consumption and another derived from the estimated COP at each time step. Both methods closely match the measured value, with slight deviations above or below, which is expected given the COP's sensitivity to small temperature variations.

Overall, the model satisfactorily reproduces the global energy exchanges of the HP. The deviations from the real data are small and remain within an acceptable margin, confirming the model's validity for integration into full system simulations and optimization processes.

The model results show excellent agreement with the actual data from the demonstrator, as presented in Table 1. The estimated electrical consumption along one day of the HP presents a relative error of approximately 1%, indicating a highly accurate reproduction of the compressor's

energy behaviour. Regarding the energy absorbed at the evaporator, the model slightly overestimates the actual value, with an error close to 4%. This difference remains acceptable, considering the sensitivity of the evaporation process to input conditions and thermal losses.

Table 1: Numerical results of the HP's model validation – Example for one-day sample period

Quantity	Measured value (kWh)	Simulated value (kWh)	Relative error
HP electrical consumption	2.54	2.57	+1.18 %
Energy absorbed at the evaporator	-4.07	-4.25	+4.43 %
Energy delivered at the condenser (method 1: Evap + Cons)	6.43	6.81	+5.91 %
Energy delivered at the condenser (method 2: COP × Cons)	6.43	6.21	-3.42 %

As for the energy delivered at the condenser, two different estimates were obtained: one based on a direct energy balance and another calculated from the COP. Both fall within an error margin of 3% to 6% compared to the actual value, showing good consistency with the measured behaviour. These moderate and consistent deviations reinforce the model's validity as a simulation tool, with sufficient accuracy for use in the energy optimization of the MiniStor system.

3.2.2. PCM grey-box model

The model developed to represent the behaviour of the PCM subsystem in MiniStor has been implemented in Python, too. It follows a physically discretized approach that reproduces the thermal evolution of the material over time and space. The model is based on the general principles presented in deliverable D4.1 and adapts the theoretical thermal properties of the Sunamp SU58 system and similar sources, including specific heats, latent heat of fusion, thermal conductivity and system geometry.

The PCM is modelled as a one-dimensional block divided into layers, each associated with an enthalpy value and a temperature. Temperature evolution is determined through an energy balance that considers thermal exchanges with various sources (solar, HP and TCM output), as well as heat losses to the environment. Phase change is incorporated using an enthalpy-based model, in which the transition from solid to liquid and vice versa occurs within a defined fusion temperature range. The use of enthalpy allows the model to accurately capture the system's thermal inertia during charging and discharging phases without explicitly solving the phase change equations.

The system's operating conditions are retrieved from actual demonstrator data (valves, pumps and TCM mode) and are used to determine whether the PCM is in a charging or discharging phase. Depending on the heat source (solar, direct charging from the TCM, or from its discharge), thermal flows into the PCM are calculated using a constant heat transfer coefficient, an estimated surface area and the temperature difference between the thermal fluid and the PCM. A maximum temperature threshold is introduced to limit thermal charging, along with a maximum energy storage capacity equivalent to the unit's nominal limit (7 kWh), which prevents artificial overcharging in the simulation.

The model dynamics are solved using explicit time and space discretization, incorporating internal thermal diffusion between layers and external energy accumulation by convection. Both the layered

thermal evolution and the total stored energy are tracked, including continuous thermal losses to the environment.

Figure 39 represents the thermal evolution of the PCM system during a complete daily operating cycle, comparing the results of the simulated model with real data measured by the *PT100* sensor. The curves corresponding to the individual PCM layers show how the thermal front progresses through the material, reflecting heat propagation during the charging and discharging processes. The blue curve represents the average temperature of the model, obtained as the mean of all simulated layers, while the green curve corresponds to the real temperature measured inside the PCM. A good correlation between both curves is observed, indicating that the model can accurately reproduce the system's thermal dynamics.

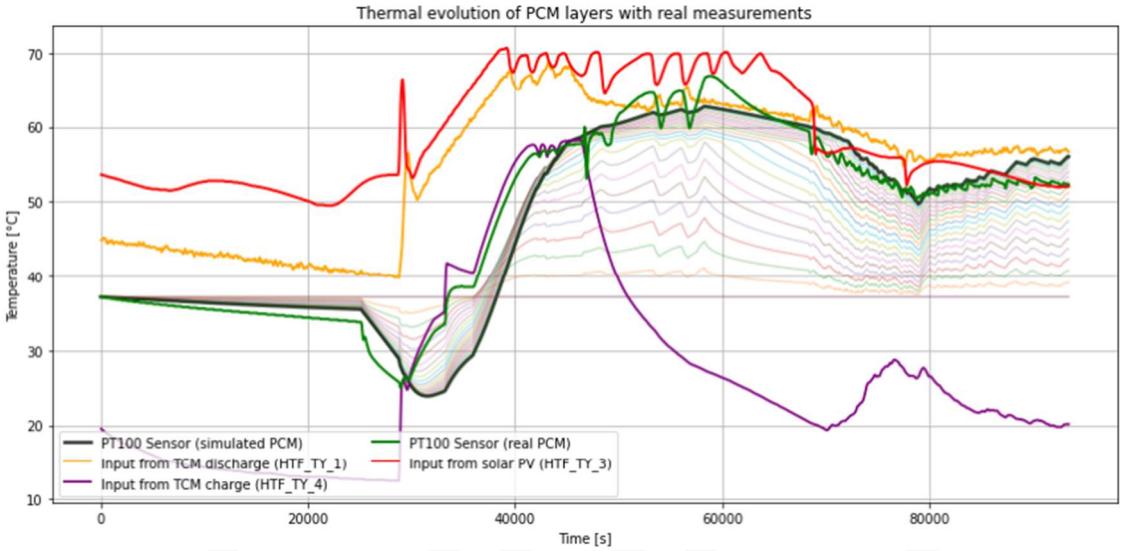


Figure 39: Validation example of the PCM model using real measurements

The simulation considers multiple sources of thermal load toward the PCM, allowing a more realistic representation of actual operating conditions. The red curve corresponds to the solar input (*HTF_TY_3*), which shows a significant thermal peak during the central hours of the day. The purple curve represents the thermal load from the TCM subsystem (*HTF_TY_4*), while the orange curve reflects the thermal discharge from the TCM to the PCM (*HTF_TY_1*). The combination of these input sources generates a complex thermal response in the PCM, which is correctly captured by the model in both the heat accumulation and release phases.

The simulated model reproduces the thermal transitions observed in the real system, showing a consistent response to varying loading conditions. Minor discrepancies observed during rapid temperature changes can be attributed to simplifications in the thermal exchange modelling or to the temporal resolution used, but do not compromise the overall validity of the model.

Quantitative results show a Root-Mean-Square Error (RMSE) of 3.6°C between the simulated average PCM temperature and the real *PT100* sensor measurement, with a relative error of 7.6% compared to the observed thermal range. This level of accuracy is acceptable for energy simulation applications, although it suggests potential areas for improvement. In particular, model parameters related to effective thermal capacity, internal conductivity, or boundary conditions at heat exchange interfaces could be refined as more real data curves become available for calibration.

3.2.3. TCM grey-box model

The model developed to simulate the behaviour of the TCM system and its interaction with the condenser and evaporator subsystems has been also implemented in Python, based on technical specifications from deliverable D4.1 of the MiniStor project, which defines the physicochemical characteristics and operational logic of the devices. The approach is based on a simplified phenomenological model that represents the fraction of ammonia absorbed/desorbed through the evolution of the reactor temperature (*HTF_TY_2*), compared with the equilibrium temperature *Teq* derived from the system pressure.

To reflect the different system states (charging, discharging and pause), a synthetic pressure profile is generated based on the operating mode (*MODE_TCM_UNIT*), which allows the dynamic derivation of the equilibrium temperature. The model calculates the absorbed fractions in two distinct reactor regions (*X1* and *X2*) using energy balances with adjusted heat transfer coefficients and reaction enthalpies. From the average conversion, the ammonia content in the tank is estimated, as well as the instantaneous mass flow of the refrigerant. This allows quantification of the energy exchanged during the sorption process, which is then compared to the thermal energy measured in the condenser (*HTF_TY_5*) taking into account also the evaporator energy (*HTF_TY_8*).

The model also incorporates thermal sub-models of the condenser and evaporator, estimating the outlet temperature variation as a function of the refrigerant flow rate and the associated condensation/evaporation heat. This thermohydraulic model enables the simulation of *HTF_TY_5* (condenser) signal, which is compared with real measurement to evaluate the simulation's accuracy.

The graph below (Figure 40) shows the evolution of the TCM system and its thermal exchange subsystem (condenser and evaporator) throughout a day, as a function of time expressed in minutes. Both simulated and real variables are compared, with special attention to the ammonia level in the tank (*NH3_LT_1*), key temperatures and the estimated ammonia flow.

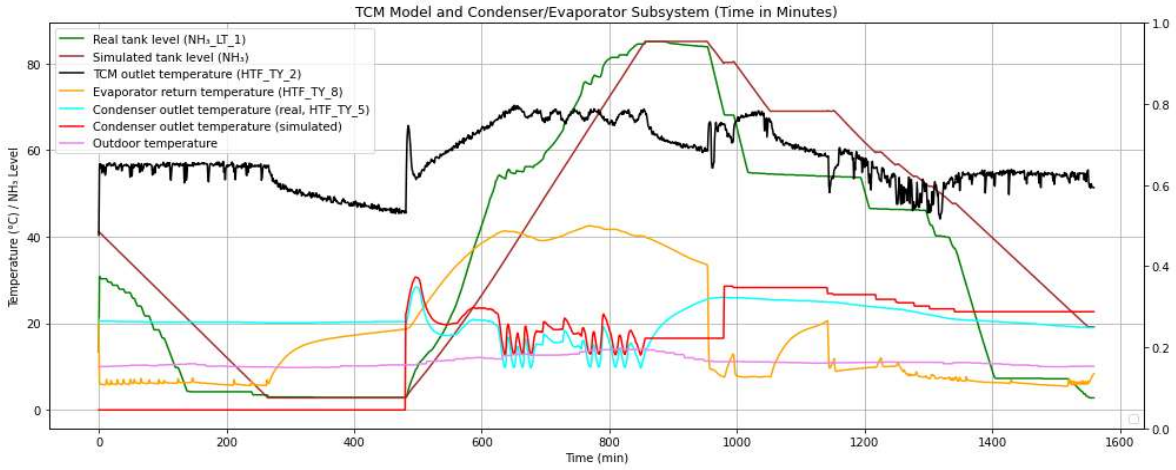


Figure 40: Validation example of the TCM model using real measurements

The model presents several important strengths. First, the simulated profile of NH_3 level (brown line) appropriately fits the general shape of the real level (green), especially during charging and discharging periods. This suggests that the thermodynamic absorption/desorption model based on energy balances and the artificial pressure profile reasonably reproduces the NH_3 storage dynamics. Likewise, the simulated temperature at the condenser outlet (red) consistently follows the

operational phases, particularly during charging modes 1 and 2, showing a realistic thermal transition. The integration of the evaporator subsystem also shows consistent thermal behaviour.

Among the challenges identified, the model's sensitivity to pressure steps and abrupt changes in operational mode stands out, which affects the accuracy of the NH_3 flow (blue line) during certain transitions. Some jumps in the flow profile indicate that the model could still benefit from refinement in detecting brief pauses or handling intermediate states. Additionally, the simulated condenser output tends to underestimate real temperatures (cyan), which could be related to an overestimation of the overall heat transfer coefficient or a simplification in the assumed thermal properties of the working fluid and system.

Figure 40 shows a good correspondence between the real ammonia level in the tank (*NH3_LT_1*) and the simulated level, especially during the charging and discharging cycles. This behaviour is reflected in a Pearson correlation coefficient of 0.95, indicating a strong linear relationship between both signals and suggesting that the model correctly reproduces the dynamics of absorption and desorption processes. The normalized State of Charge (SOC) offers an RMSE of 0.13 indicating a good ability of the model to represent cumulative energy use over time.

3.2.4. MiniStor's optimization algorithm

The mathematical problem developed consists of optimizing the operation of a hybrid thermal system that integrates three main components: a HP, a TCM storage system and a PCM storage system. The objective is to efficiently meet a daily thermal demand profile, simultaneously minimizing the error with respect to that demand and avoiding non-physical or undesirable system configurations. To this end, a nonlinear, integer, block-based optimization problem has been formulated and solved using a genetic-algorithm-based evolutionary optimizer (differential evolution).

The system is operated in 1-minute intervals over 2 days (2880-time steps), grouped into decision blocks of 60 minutes, resulting in 48 decision variables for controlling the operation modes. Each block can take an integer value between 0 and 3, representing the possible actions: idle, solar charging of the PCM, charging the PCM from the TCM, or discharging the PCM. The system operates from an initial solution derived via low-level control rules, furnishing a valid yet adaptable starting point to support the subsequent optimization process.

These decisions are translated into binary signals (solar charging, charging/discharging from TCM and discharging to meet demand). From these, a control signal is generated for the TCM (*MODE_TCM_UNIT*), which is used to feed the physical model of the TCM.

The objective function of the problem focuses on **maximising the self-consumption and use of local renewables** by minimising the squared error between the thermal demand and the energy delivered by the PCM to fulfil the users' requirements. This optimises the use of the storage capacities of TCM and PCMs. To complete the optimization problem additional conditions were added to penalize the following undesirable behaviours:

- Charging the TCM when it is already saturated (SOC ≥ 1).
- Exceeding the physical storage limits of the PCM (energy outside the range [0, 7] kWh).
- Charging of TCM when temperatures are out of the operation limits [50,85] °C).

Each evaluation of the objective function involves chaining together the three analytical models previously described:

- **TCM model**, which simulates the evolution of ammonia content in the reactor based on the operating mode and the thermal fluid temperature (*HTF_TY_2*), generating the NH_3 flow and thermal output of the reactor.

- HP model, which calculates the behaviour of the condenser and evaporator based on input/output temperatures and the TCM mode and provides the useful temperature to the PCM subsystem.
- PCM model, which simulates layer-by-layer the thermal state of the storage and determines the energy delivered to the system based on external conditions, charge and discharge modes and the accumulated energy balance.

The optimal solution is obtained by applying the genetic-algorithm-based evolutionary optimizer with fixed swarm size and number of iterations. The inputs come from the generation and demand forecasting algorithms (external temperature and solar generation and demand profiles) and the decision outputs can be used by the low-level controller when its flexibility allows it.

To illustrate the high-level optimization model, a two-days example is included below with the aim of demonstrating the system's ability to behave as a flexible system capable of adapting to demand. This case allows observing how the system responds under various operating conditions and constraints, assessing its ability to meet thermal demand through the optimal management of available resources.

Thus, Figure 41 illustrates the operational behaviour of the MiniStor system over this two-day cycle, including the thermal evolution of the PCM subsystem (*Energy PCM [kWh]* and *Average temperature PCM [°C]*, in the first subplot), the dynamics of the TCM (*NH_3 produced* and *SOC TCM*, in the second subplot), the control decisions implemented by the HLC (*Mode TCM*, *Charge from solar*, *Charge from TCM*, *Discharge TCM*, *Discharge PCM*, in the third subplot) and the coverage of thermal demand (*Demand* and *Delivered*, in the fourth subplot). The optimization is applied to Sopron demo site where the solar production supplying Ministor is stored in a buffer tank.

It is observed that the optimization initially uses the energy contained in the PCM to meet the energy demand (hours 7 to 8) and charges the TCM using solar heat to charge the PCM for the following days (hours 8-12). Consequently, the temperature and energy levels in the PCM (and *Average temperature PCM [°C]* and *Energy PCM [kWh]*) and the TCM SOC and NH_3 produced (*SOC TCM* and *NH_3 produced*) behave accordingly (first and second subplots, respectively). Once the temperature in the solar loop reaches a high temperature, the system switches to using the heat from the solar loop directly in the PCM due to the operational conditions given in the MiniStor system (hours 12 to 16). During the night, the energy demand is covered by discharging the PCM (hours 20 to 22).

On the second day, the system operates in a similar way: first, the thermal demand is covered using the PCM (hour 31 to 32), and then the TCM is charged when the temperature in the solar loop is high enough (hours 36 to 38). In the middle of the day, the energy from the solar loop is directly supplied to the PCM because the TCM is charged, and once the radiation cannot provide enough temperature, the TCM is discharged to charge the PCM (hours 42 to 44), which is in turn discharged to cover the demand (hours 44 to 46).

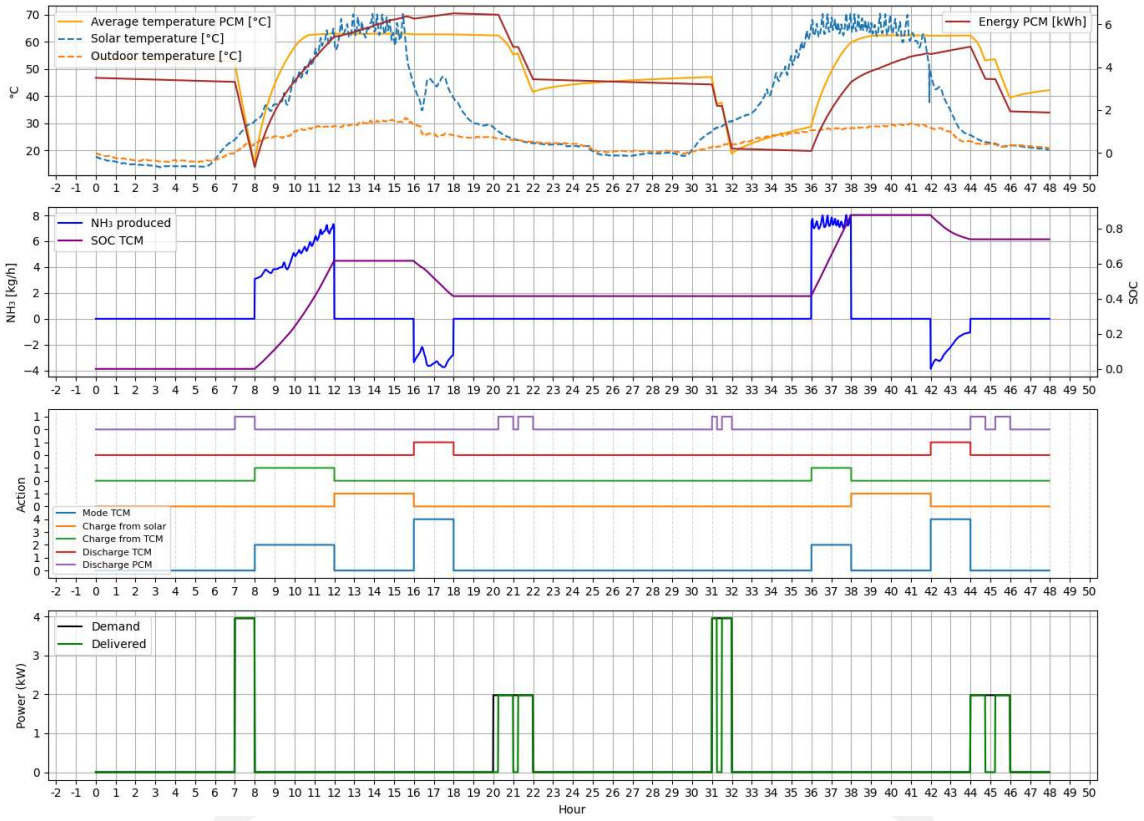


Figure 41: Example of HLC optimization for a 48-hour cycle

As can be seen, the system operates in a predictive manner, anticipating periods of low solar generation and preparing the TCM and PCM as thermal reserves. This hybrid strategy offers greater robustness against renewable variability and ensures better coverage of the user's thermal demand, while also respecting the physical storage limitations of both the PCM and the TCM.

4. Conclusions

Deliverable D5.2 presents the definitive, end-to-end design of the MiniStor SHEMS, bringing together all architectural refinements, control algorithms and system integrations developed since the initial design in D5.1. The low-level controller's rule-based logic, which includes reactor pressurization, charging, depressurization or hybrid PCM/HP operation, was rigorously validated in tests at CARTIF, factory acceptance trials at PSYCTOTHERM and on-site commissioning across the five European demo sites within MiniStor system: Thessaloniki, Sopron, Kimmeria, Santiago de Compostela and Cork. These tests confirmed safe and reliable mode transitions under a wide range of environmental and load conditions, while iterative fine-tuning accounted for site-specific hydraulics, climatology and user patterns.

A robust IoT backbone underpins the SHEMS platform: Node-RED acquires sensor and actuator data, publishes via MQTT to a centralized broker, and streams to InfluxDB; Grafana dashboards and alert rules provide live visualization and anomaly detection; and a secondary PLC implements an autonomous safety shutdown in the event of primary-PLC failure. Concurrently, analytical models of key components such as the HP, PCM and thermochemical storage modules were calibrated against measured data, achieving reasonable errors and embedded within the HLC. At the high level, Python services on Azure execute weather-driven forecasts and genetic algorithm-based optimization routines using the analytical models, exchanging setpoints via REST APIs and enabling proactive, hierarchical coordination between predictive strategies and real-time execution.

Site-specific adaptations such as solar-loop safety interlocks in Thessaloniki, preheating thresholds in cold climates, BACnet integration at Santiago, and tailored pump schedules in Kimmeria and Cork demonstrate the SHEMS platform's modularity and flexibility. By integrating validated control, comprehensive monitoring, accurate modelling and cloud-based optimization, D5.2 delivers a scalable, fully documented EMS platform that meets MiniStor's objectives of maximizing renewable self-consumption, reducing energy costs and preserving occupant comfort.

5. Annex

MiniStor system's variable map

Table 2: Variable map of the MiniStor system

PID variables	Description	Unit	Data type
Alarms			
DI1	Frigopol Compressor VFD alarm	-	Bit (1/0)
GL	Ammonia SubSystem Alarm	-	Bit (1/0)
HP_AL	Heat Pump Alarm Status	-	Bit (1/0)
S10	Smoke Sensor Alarm	-	Bit (1/0)
X10	Heat Pump Compressor VFD alarm	-	Bit (1/0)
Fancoils			
V_004	Fan Coil 1	-	Bit (1/0)
V_005	Fan Coil 2	-	Bit (1/0)
HP			
Control_PM10	HP Run Command	-	Bit (1/0)
Control_PM11	HP Setpoint °C	°C	Integer
HP_A	Heat Pump Compressor Ampere	A	Integer
HP_Cap	Heat Pump Heating Capacity kW	kW	Integer
HP_Hz	Heat Pump Compressor Hz	Hz	Integer
HP_kW	Heat Pump Power Consumption kW	kW	Integer
HP_V	Heat Pump Compressor Volt	V	Integer
M_001	Heat Pump Compressor Run	-	Bit (1/0)
Refrig_flow	Refrigerant flow kg/s	kg/s	Integer
RF_PAH_1	High Press Safety Switch	-	Bit (1/0)
RF_PAL_1	Low Press Safety Switch	-	Bit (1/0)
RF_PT_1	Refrigerant Discharge Pressure bar	bar	Integer
RF_PT_2	Refrigerant Suction Pressure bar	bar	Integer
Other			
-	Period mode (Summer/Winter) from CARTIF	-	0-1
-	State System from CARTIF	-	0-17
-	State TCM from CARTIF	-	0-7
-	State PCM from CARTIF	-	0-7
-	State Demand from CARTIF	-	0-3
-	State Solar Tank from CARTIF	-	0-1
APC_PH1_kW	Apparent Power Consumption Ph1 kVA	kW	Integer
APC_PH2_kW	Apparent Power Consumption Ph2 kVA	kW	Integer
APC_PH3_kW	Apparent Power Consumption Ph3 kVA	kW	Integer
APC_tot_kW	Apparent Power Consumption Total kVA	kW	Integer
C_PH1_A	Current Ph1 A	A	Integer
C_PH2_A	Current Ph2 A	A	Integer
C_PH3_A	Current Ph3 A	A	Integer
C_tot_A	Current Total A	A	Integer
Control_PM12	Remote Emergency Stop Command	-	Bit (1/0)
COS_PH1	Cos Ph1	°	Integer
COS_PH2	Cos Ph2	°	Integer
COS_PH3	Cos Ph3	°	Integer
F_PH1	Frequency Ph1	Hz	Integer
F_PH2	Frequency Ph2	Hz	Integer
F_PH3	Frequency Ph3	Hz	Integer
RPC_PH1_kW	Real Power Consumption Ph1 kW	kW	Integer

PID variables	Description	Unit	Data type
RPC_PH2_kW	Real Power Consumption Ph2 kW	kW	Integer
RPC_PH3_kW	Real Power Consumption Ph3 kW	kW	Integer
RPC_tot_kW	Real Power Consumption Total kW	kW	Integer
S1	Emergency Stop Switch	-	Bit (1/0)
S14	Universal Input 1	-	Bit (1/0)
S15	Universal Input 2	-	Bit (1/0)
S16	Universal Input 3	-	Bit (1/0)
S2	Local ON/OFF switch	-	Bit (1/0)
S3	Reset alarm button	-	Bit (1/0)
U0	Universal Output 1	-	Bit (1/0)
U1	Universal Output 2	-	Bit (1/0)
U2	Universal Output 3	-	Bit (1/0)
U3	Universal Output 4	-	Bit (1/0)
U4	Universal Output 5	-	Bit (1/0)
V_PH1_V	Voltage Ph1 V	V	Integer
V_PH2_V	Voltage Ph2 V	V	Integer
V_PH3_V	Voltage Ph3 V	V	Integer
PCM			
S12	Hot Water PCM Controller Command	-	Bit (1/0)
S13	DHW PCM Controller Command	-	Bit (1/0)
Pumps			
P_001	PCM Pump 1	-	Bit (1/0)
P_003	NH_3 Evaporator Pump	-	Bit (1/0)
P_004	PCM Pump 2	-	Bit (1/0)
P_005	HP Condenser Pump	-	Bit (1/0)
P_006	Solar Buffer Pump	-	Bit (1/0)
P_007	NH_3 Condenser Pump	-	Bit (1/0)
Solar			
Deltasol_Err_HC_cooling	Error: HC cooling below flow minimum temperature	-	Bit (1/0)
Deltasol_Err_line_broke	Error: Sensor line broken	-	Bit (1/0)
Deltasol_Err_line_short	Error: Sensor line short-circuited	-	Bit (1/0)
Deltasol_Err_lowpresurre	Error: Low pressure	-	Bit (1/0)
Deltasol_Err_overpressure	Error: Overpressure	-	Bit (1/0)
Deltasol_Flow_rate_sensor_13_H	Flow rate sensor 13	l/h	Integer
Deltasol_Flow_rate_sensor_14_H	Flow rate sensor 14	l/h	Integer
Deltasol_Imp_1_H	Counter reading	-	Integer
Deltasol_Imp_2_H	Counter reading	-	Integer
Deltasol_Output_A	Output A	%	Integer
Deltasol_Pressure_sensor_19	Pressure sensor 19	bar	Integer
Deltasol_Pump_1	Pump speed relay 1	%	Integer
Deltasol_Pump_2	Pump speed relay 2	%	Integer
Deltasol_Temperature_sensor_1	Temperature sensor 1	°C	Integer
Deltasol_Temperature_sensor_10	Temperature sensor 10	°C	Integer
Deltasol_Temperature_sensor_11	Temperature sensor 11	°C	Integer
Deltasol_Temperature_sensor_12	Temperature sensor 12	°C	Integer
Deltasol_Temperature_sensor_2	Temperature sensor 2	°C	Integer
Deltasol_Temperature_sensor_3	Temperature sensor 3	°C	Integer
Deltasol_Temperature_sensor_4	Temperature sensor 4	°C	Integer
Deltasol_Temperature_sensor_5	Temperature sensor 5	°C	Integer
Deltasol_Temperature_sensor_6	Temperature sensor 6	°C	Integer
Deltasol_Temperature_sensor_7	Temperature sensor 7	°C	Integer
Deltasol_Temperature_sensor_8	Temperature sensor 8	°C	Integer
Deltasol_Temperature_sensor_9	Temperature sensor 9	°C	Integer
Deltasol_Warning_AT	Warning: ΔT too high	-	Bit (1/0)

PID variables	Description	Unit	Data type
Deltasol_Warning_Night_circulation	Warning: Night circulation	-	Bit (1/0)
froniusgen_AphA	Phase A Current	A	Integer
froniusgen_AphB	Phase B Current	A	Integer
froniusgen_AphC	Phase C Current	A	Integer
froniusgen_ChaSt	Charge status of storage device. Enumerated value.	-	Integer
froniusgen_ChaState	Currently available energy as a percent of the capacity rating.	-	Integer
froniusgen_InBatV	Internal battery voltage.	V	Integer
froniusgen_module1DCA	DC Current	A	Integer
froniusgen_module1DCV	DC Voltage	V	Integer
froniusgen_module1DCW	DC Power	W	Integer
froniusgen_module2DCA	DC Current	A	Integer
froniusgen_module2DCV	DC Voltage	V	Integer
froniusgen_module2DCW	DC Power	W	Integer
froniusgen_PF	AC Power Factor	-	Integer
froniusgen_PhVphA	Phase Voltage AN	V	Integer
froniusgen_PhVphB	Phase Voltage BN	V	Integer
froniusgen_PhVphC	Phase Voltage CN	V	Integer
froniusgen_PPVphAB	Phase Voltage AB	V	Integer
froniusgen_PPVphBC	Phase Voltage BC	V	Integer
froniusgen_PPVphCA	Phase Voltage CA	V	Integer
froniusgen_StorAval	SOC ChaState minus storage reserve MinRsvPct times capacity rating AhrRtg.	-	Integer
froniusgen_TmpCab	Cabinet Temperature	°C	Integer
froniusgen_VA	AC Apparent Power	VA	Integer
froniusgen_VAr	AC Reactive Power	Var	Integer
froniusgen_W	AC Power	W	Integer
froniusgen_WChaMax	Setpoint for maximum charge	W	Integer
froniusmeter_PF	Power Factor	-	Integer
froniusmeter_VA	AC Apparent Power	VA	Integer
froniusmeter_VAR	Reactive Power	VAR	Integer
froniusmeter_W	Total Real Power	W	Integer
froniusmeter_WphA	Watts phase A	W	Integer
froniusmeter_WphB	Watts phase B	W	Integer
froniusmeter_WphC	Watts phase C	W	Integer
R1	Buffer Resistors Backup Heater 2kW	-	Bit (1/0)
Inverter_Active_power	N/A	W	Integer
Inverter_Internal_temperature	N/A	°C	Integer
Inverter_Phase_A_voltage	When the output mode is L/N, L1/L2/N, or L1/L2, the information is invalid.	V	Integer
Inverter_Phase_B_current	When the output mode is L/N, L1/L2/N, or L1/L2, the information is invalid.	A	Integer
Inverter_Phase_B_voltage	When the output mode is L/N, L1/L2/N, or L1/L2, the information is invalid.	V	Integer
Inverter_Phase_C_current	When the output mode is L/N, L1/L2/N, or L1/L2, the information is invalid.	A	Integer
Inverter_Phase_C_voltage	When the output mode is L/N, L1/L2/N, or L1/L2, the information is invalid.	V	Integer
Inverter_Power_factor	N/A	-	Integer
Inverter_Power_grid_current	When the output mode is L/N, L1/L2/N, or L1/L2, Power grid current is used.	A	Integer
Inverter_PV1_current	A maximum of 24 PV strings are supported. The number of PV strings read by the host is defined by the Number of	A	Integer
Inverter_PV1_voltage		V	Integer
Inverter_PV2_current		A	Integer

PID variables	Description	Unit	Data type
Inverter_PV2_voltage	PV strings signal. The voltage and current register addresses for each PV string are as follows: PVn voltage: 32014 + 2n PVn current: 32015 + 2n n indicates the PV string number, which ranges from 1 to 24.	V	Integer
Inverter_Reactive_power	N/A	W	Integer
Yutaki_ ECO_Offset_Temperature	Control Circuit 1: Heat ECO Offset Temperature	-	Integer
Yutaki_Alarm_number	Alarm number	-	Integer
Yutaki_Ambient_Average_Temp	Ta2: Outdoor Unit Ambient Average Temp.	°C	Integer
Yutaki_Compressor_curr	R134a Compressor current value	A	Integer
Yutaki_Compressor_f	R134a Compressor frequency	Hz	Integer
Yutaki_Contro_Unit_Mode	Control Unit Mode	-	Integer
Yutaki_Control Circuit_1_W	Control Circuit 1: Water heating Fix Setting Temp	°C	Integer
Yutaki_Control_Block_menu	Control Block menu *6	-	Integer
Yutaki_Control_BMS_Alarm	Control BMS Alarm *4	-	Integer
Yutaki_Control_Circuit_RS	Control Circuit 1 Run/Stop	-	Bit (1/0)
Yutaki_Control_Heat	Control Heat. OTC Circuit 1.	-	Integer
Yutaki_Control_Unit	Control Unit Run/Stop	-	Bit (1/0)
Yutaki_Defrosting	Defrosting	-	Integer
Yutaki_Discharge_Pressure	R134a Discharge Pressure	Mpa	Integer
Yutaki_Discharge_Temperature	R134a Discharge Temperature	°C	Integer
Yutaki_Eco_mode	Control Circuit 1: Eco mode	-	Integer
Yutaki_Exp_Valve	R134a Indoor Expansion valve 2 opening	%	Integer
Yutaki_Expansion_valve_opening	EVI: Indoor Expansion valve opening	%	Integer
Yutaki_H4	H4: Inverter Operation frequency	Hz	Integer
Yutaki_Operation	Operation System	-	Integer
Yutaki_Out_Expansion_valve	EVO: Outdoor Expansion valve	%	Integer
Yutaki_Outdoor_ambient_temperature	Outdoor ambient temperature	°C	Integer
Yutaki_P	P1: Compressor running current	A	Integer
Yutaki_Retry_Code	R134a Retry Code	-	Integer
Yutaki_Software_LCD	Software LCD Liquid Crystal Display	-	Integer
Yutaki_Software_PCB	Software PCB Printed Circuit Board	-	Integer
Yutaki_stoppage	DI: Cause of stoppage	-	Integer
Yutaki_Suction_Pressure	R134a Suction pressure	MPa	Integer
Yutaki_Suction_temperature	R134a Suction temperature	°C	Integer
Yutaki_System_Configuration	System Configuration	-	Integer
Yutaki_System_status_2	System status 2	-	Integer
Yutaki_Td	Td: Discharge Gas temp °C	°C	Integer
Yutaki_Te	Te: Evaporation temp °C	°C	Integer
Yutaki_Tg	Tg: Gas Temperature THMg °C	°C	Integer
Yutaki_TI	TI: Liquid Temperature THMI °C	°C	Integer
Yutaki_Unit_Power_consumption	Unit Power consumption	W	Integer
Yutaki_Water_flow_level	Water flow level	-	Integer
Yutaki_Water_Inlet_unit_temperature	Water Inlet unit temperature	°C	Integer
Yutaki_Water_outlet_hpT	Water outlet hp T° Water outlet hp outlet unit temperature	°C	Integer
Yutaki_Water_outlet_Temp3	O3: Water outlet Temp. 3 Two3 Linked to Inertia Tank	°C	Integer
Yutaki_Water_outlet_unit_temperature	Water outlet unit temperature	°C	Integer
Yutaki_Water_pump_speed	Water pump speed	%	Integer

PID variables	Description	Unit	Data type
TCM			
Contro_PM8	TCM Command to Charge Request to start charging	-	Bit (1/0)
Control_PM1	Ready for charging fully discharged	-	Bit (1/0)
Control_PM2	Ready for discharging fully charged	-	Bit (1/0)
Control_PM3	Stand by ON/OFF	-	Bit (1/0)
Control_PM4	Pressurization/Depresssurization	-	Bit (1/0)
Control_PM6	TCM Event Discharging Discharging ON/OFF	-	Bit (1/0)
Control_PM7	TCM Event Charging Charging ON/OFF	-	Bit (1/0)
Control_PM9	TCM Command to Discharge request to start discharging	-	Bit (1/0)
Frigopol_A	Frigopol Compressor Ampere	A	Integer
Frigopol_Hz	Frigopol Compressor Hz	Hz	Integer
Frigopol_kW	Frigopol Compressor kW	kW	Integer
Frigopol_V	Frigopol Compressor Volt	V	Integer
M1	NH_3 Compartment Exhaust Fan	-	Bit (1/0)
MODE_TCM_UNIT	MODE TCM UNIT	-	0-4
NH3_LS_1	NH_3 Liquid Level Switch on Separator	-	Bit (1/0)
NH3_LT_1	Liquid Level Sensor %	%	Integer
NH3_PH_1	NH_3 High Press Safety Switch	-	Bit (1/0)
NH3_PT_1	TCM Pressure bar	bar	Integer
NH3_PT_2	Discharge Pressure bar	bar	Integer
NH3_PT_3	Suction evaporating Pressure bar	bar	Integer
NH3_TS_1	Oil thermal safety switch	-	Bit (1/0)
NH3_XV1	Suction Solenoid Valve	-	Bit (1/0)
NH3_XV2	Suction separator Solenoid Valve	-	Bit (1/0)
NH3_XV3	Hot Gas Solenoid Valve	-	Bit (1/0)
NH3_XV4	Injection Solenoid Valve	-	Bit (1/0)
NH3_XV5	Liquid Line Solenoid 1	-	Bit (1/0)
NH3_XV6	EEV Solenoid	-	Bit (1/0)
NH3_XV7	Oil Solenoid	-	Bit (1/0)
S1	Emergency Stop Switch	-	Bit (1/0)
S11	Ammonia Detection Sensor Level 2	-	Bit (1/0)
S3	Local ON/OFF switch for TCM	-	Bit (1/0)
S8	Ammonia Detection Sensor Level 1	-	Bit (1/0)
V_001	Frigopol Compressor Run	-	Bit (1/0)
V_002	Desuperheater Fan	-	Bit (1/0)
Temperatures			
Demo_outdoor	Temperature offsite DEMO	°C	Integer
HTF_TY_1	HTF Reactor OUT °C	°C	Integer
HTF_TY_2	HTF Reactor IN °C	°C	Integer
HTF_TY_3	Solar Buffer °C	°C	Integer
HTF_TY_4	HTF HP Condenser OUT °C	°C	Integer
HTF_TY_5	HTF NH_3 Condenser OUT °C	°C	Integer
HTF_TY_6	HTF NH_3 Evaporator OUT °C	°C	Integer
HTF_TY_7	HTF HP Evaporator OUT °C	°C	Integer
HTF_TY_8	HTF NH_3 Evaporator IN °C	°C	Integer
NH3_TY_1	TCM Reactor Temp1 °C	°C	Integer
NH3_TY_10	Receiver IN Temp °C	°C	Integer
NH3_TY_11	Receiver OUT Temp °C	°C	Integer
NH3_TY_2	TCM Reactor Temp2 °C	°C	Integer
NH3_TY_3	TCM Reactor Temp3 °C	°C	Integer
NH3_TY_4	TCM OUT Temp °C	°C	Integer
NH3_TY_7	Suction Temp °C	°C	Integer

PID variables	Description	Unit	Data type
NH3_TY_8	Discharge Temp °C	°C	Integer
NH3_TY_9	Separator OUT Temp °C	°C	Integer
PCM_TY_3	Cold PCM °C	°C	Integer
PT100	Spare Temp Sensor Inlet °C	°C	Integer
Valves			
HTF_XV_1	Solenoid - Fancoil_cold_in_3wv	-	Bit (1/0)
HTF_XV_2	Solenoid - TCMevaporator_in	-	Bit (1/0)
HTF_XV_3	Solenoid - PVTtoPRODpump	-	Bit (1/0)
HTF_XV_5	Solenoid - PVTtoColdPCM	-	Bit (1/0)
HTF_XV_6	Solenoid - TCMrecirc_in_3wv	-	Bit (1/0)
HTF_XV_7	Solenoid - Fancoilheat	-	Bit (1/0)
HTF_XV_8	Solenoid - MANIFOLDreturn	-	Bit (1/0)
HTF_XV_9	Solenoid - MANIFOLDsupply	-	Bit (1/0)
HTF_XV_10	Solenoid - PCMHW	-	Bit (1/0)
HTF_XV_11	Solenoid - PCMDHW	-	Bit (1/0)
HTF_XV_12	Solenoid - HPevaporator	-	Bit (1/0)
HTF_XV_13	Solenoid - TCMcondenser_out	-	Bit (1/0)
HTF_XV_14	Solenoid - TCMin_3wv	-	Bit (1/0)
HTF_XV_15	Solenoid - TCMcondenser_in	-	Bit (1/0)
HTF_XV_16	Solenoid - TCMrecirc_out_3wv	-	Bit (1/0)