Delivery no.: 4.4d
Design, development and testing of low-cost controllers for fuel-shift technologies

Photo: By & Havn / Ole Malling

DTU/METROTHERM
Author, Andreas Thingvad
Date April, 28, 2017

Public deliverable
Confidential deliverable

Preface

EnergyLab Nordhavn – New Urban Energy Infrastructures is an exciting project which will continue until the year 2019. The project will use Copenhagen's Nordhavn as a full-scale smart city energy lab, which main purpose is to research as well as to develop and demonstrate future energy solutions for renewable energy systems.
The goal is to identify the most cost-effective smart energy systems, which can contribute to solve the major climate challenges the world are facing.

Budget: The project has a total budget of DKK 143 m (€ 19 m) of this, DKK 84 m (€ 11 m) is funded in two rounds by the Danish Energy Technology Development and Demonstration Programme (EUDP).

Forord

EnergyLab Nordhavn er et spændende projekt der løber til og med år 2019. Projektet vil foregå i Københavns Nordhavn, og vil fungere som et fuldskala storby-energilaboratorium, der skal undersøge, udvikle og demonstrerer løsninger for fremtidens energisystemer.
Målet er at finde fremtidens mest omkostningseffektive energisystemer, der desuden kan bidrage til en løsning på de store klimaudfordringer verden står overfor nu og i fremtiden.

Budget: Projektets totale budget er DKK 143 mio. (€19 mio.), hvoraf DKK 84 mio. (€11 mio.) er blevet finansieret af Energiteknologisk Udviklings- og Demonstrationsprogram, EUDP.

Project Information

Deliverable no.: 4.4d
Deliverable title: Design, development and testing of low-cost controllers for fuel-shift technologies
WP title: Smart Network Services
Task Leader: Torben Schifter-Holm
WP Leader: Christian Brandt Rasmussen
Comment Period: May 03 2017 to June 02 2017]

For further information on this specific deliverable, please contact:

Andreas Thingvad - athing@student.dtu.dk or
Christian Brandt Rasmussen – chrbr@elektro.dtu.dk
Section 4: Anders Bro Pedersen – abp@elektor.dtu.dk

For other information regarding EnergyLab Nordhavn, please contact:

EnergyLab Nordhavn Secretariat
Center for Electric Power and Energy, DTU Electrical Engineering
Elektrovej
Building 325
DK-2800 Kgs. Lyngby
Denmark

E-mail eln@dtu.dk
Tlf. +45 45 25 35 54

www.energylabnordhavn.dk

Table of Contents

List of Abbreviations

ELN – Energylab Nordhavn
DH – District Heating
DHW – Domestic Hot Water
CHP - Combined Heat and Power
PV - Photovoltaics
ECL310 - ECL Comfort 310 (Danfoss)
FS - Fuel Shift
TSO – Transmission System Operator

Executive Summary

Summary

This report presents the design and development of a novel heating system that can switch between electrical power and district heating as the source for heating up domestic hot water in residential apartments in Nordhavn. The presented way of switching between heating sources is called fuel shift and is used to perform ancillary services to the electrical grid.

Version Control

Version	Date	Author	Description of Changes
18	2017-27-07	Kasper Korsholm Østergaard	Minor changes from review
17	2017-29-06	Thomas Henrik Pedersen	Added comments from review
16	2017-24-05	Christian Brandt Rasmussen	Returned from review
15	2017-02-05	Christian Brandt Rasmussen	Finalizing document setup, ready for WPL approval
14	[2017-02-05]	Andreas Thingvad	Implementing comments and suggestions from internal review

Quality Assurance

Author	Reviewer	Approver
Andreas Thingvad	Jan Eric Thorsen, Danfoss	WPL group

Status of deliverable		
Action	**By**	**Date/Initials**
Sent for review	Christian Brandt Rasmussen, DTU CEE	**2017-05-10 CHRBR**
Reviewed	Jan Eric Thorsen, Danfoss	**2017-05-24 JET**
Verified	Kasper Korsholm Østergaard, METRO THERM	**2017-07-27 KKO**
Approved	WPL group	**2017-07-27**

1. Introduction

Today, most buildings heat requirements are supplied by District Heating (DH). Via two heat exchangers, the DH heats both the room heating water, circulating in the radiators and floor heating, and the Domestic Hot Water (DHW), consumption water used in the showers and the tap water.
The distribution of DH is a subject to some losses but since heat is a cogeneration when generating electricity using Combined Heat and Power (CHP) units, it historically has been very cheap.
Going towards a more sustainable electricity production, however, means that an increasing part of the production will come from renewable sources, such as wind and photovoltaics (PV), which are not based on heat generation. As modern buildings are continuously better insulated, they require less energy to maintain a comfortable room temperature. When less energy is transferred to the house the relative distribution losses of the DH will be larger.
A part of the project, EnergyLab Nordhavn will therefore investigate a concept called fuel shift, where the domestic heat can come from other sources than DH, such as electrical heating. By this the DH temperature can be lowered to reduce the distribution loss and in some situations, the DH supply can be turned off. Electrical heating can be made using a resistive load, which can instantaneously transform large amount of electrical power into heat, it can also be made using a heat pump, which with a higher efficiency but poorer flexibility, can absorb heat from the surroundings to heat water or air. A resistive load is used in this project because it is cheaper and more controllable. The efficiency is less important than a flexible power consumption as the electrical grid sometimes, has a surplus of power production causing an imbalance between production and consumption. To aid in balancing the active power in the grid is one of many ancillary services offered to the Transmission System Operator (TSO). This balancing of the consumption with the production can be on the short seconds-to-second scale, but also on an hourly basis, where large wind production can cause the electricity price to fluctuate and make it more feasible to consume electricity.
This project introduces a real-time controlled system that can perform fuel-shift in domestic apartments to shift between the two heat sources for heating DHW.

2. System description

The fuel shift system is based upon an external controller, which can disable a conventional DH system and enable the electrical heating element installed in the boiler for DHW when required, controlled by an external signal e.g. Smart Network Services. Regardless of the input, the internal state machine of the ELN controller will continuously operate, in order to guarantee that the given temperature thresholds are not exceeded and the home owners never experience cold water. This controller is named after the project and called EnergyLab Nordhavn (ELN) controller. Besides the basic functions, the ELN controller can be expanded with internal algorithms for enabling e.g. autonomous smart network services and other related features during the project.

2.1 ECL310 and the normal application

The DH system is based on a DH controller, namely the ECL Comfort 310 (ECL310) made by Danfoss. ECL310 is an electronic heat controller that can regulate the usage of DH for heating and cooling systems in up to 3 circuits. A typical application utilizing ECL310 can be seen in Fig. 1. The DH comes in on the left side of the figure and the heat exchangers heats up the water in the top circuit (circuit 1) for the room heating and in the bottom circuit (circuit 2) for the DHW. Room heating and DHW runs in their own isolated circuits with heat exchangers between them and the DH supply.
The system is running completely autonomously and can change the temperature set-point of the room heating measurement (S3), based on variables such as the outdoor temperature measurement (S1), or a schedule that reduces the indoor temperature at night, or during e.g. a holiday period. ECL310 can be used in multiple configurations with standardised setups, defined by the availability and placement of the different sensors. The setup shown in Fig. 1 is the setup used for this project, called 266.10.

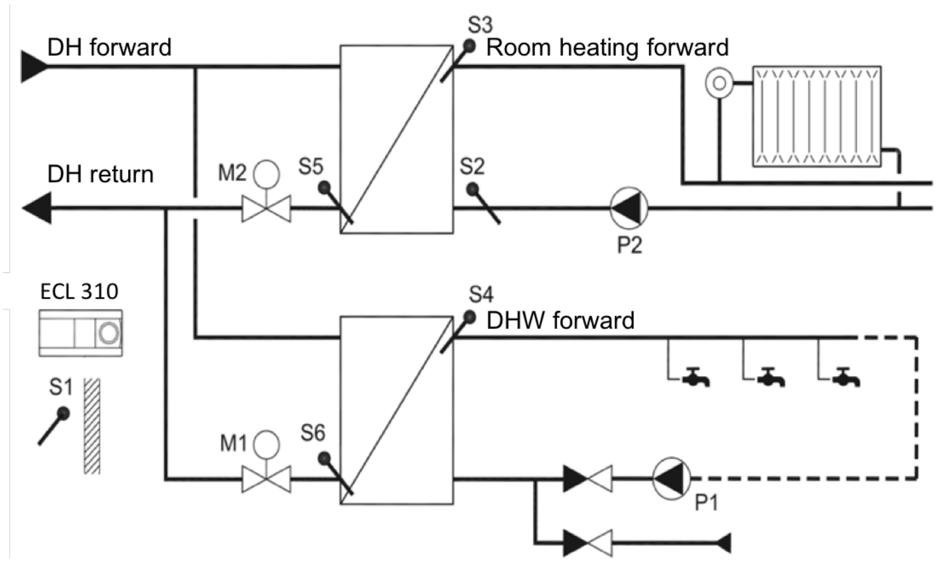

Figure 1 ECL310 application 266.10

There is a motorised valve (M1, M2) at the return point of the two heat exchangers that controls how much of the DH water that will be allowed through the heat exchanger. When the DHW temperature sensor, (S4) measures a temperature of the DHW below the desired set-point, M1 will open more and vice versa when the temperature is too high. The ECL310 controller actuates the DH input valves, M1 and M2, and the circulation pumps, P1 and P2 on the secondary side, based on the measurements, S1 to S6.

The different measurements in the ECL310 application:

- S1 Outdoor temperature
- S2 Room heating return temperature
- S3 Room heating forward temperature
- S4 Forward temperature, (DHW)
- S5 DH return temperature (Room heating)
- S6 DH return temperature (DH)
- P1 & P2 Circulation pumps
- M1 & M2 Motorized control valve

2.2 System description including ELN controller

The ELN controller is JAVA based software platform running on a Raspberry Pi that can read sensor data through the ELN310 and change its settings.
The ECL310 controls everything in Fig. 1, regarding the heating of circuit 1 and 2 with DH, without any external implications.
In order to perform a fuel shift from DH to electrical heating for heating of the DHW, it is necessary to close M1 as well as a relay for enabling the electrical heating element. ECL310 has a number of electrical relays that can be used for that purpose and it can be accessed via a Modbus-TCP connection. Via this connection, the external ELN controller can access the temperature measurements, close M1 and enable the electrical heating.
The system implemented in the households in Nordhavn, consists of a typical ECL310 setup where the DHW is stored in a thermally insulated water tank of around 95 and 140 Litres. Inside this tank is placed

both the heat exchanger which is a spiral for the DH and an electrical heating element that can be turned on by a relay.
In Fig. 2, the augmented system with the storage tank can be seen, including the electrical heating as well as two new sensors that are necessary to regulate between the two energy sources.
When using DH, the ECL310 will run the system in normal operation; when using electrical heating the ELN controller will send a command to the ECL310 to close valve M1 and enable relay R6, which is available via the ECL310.

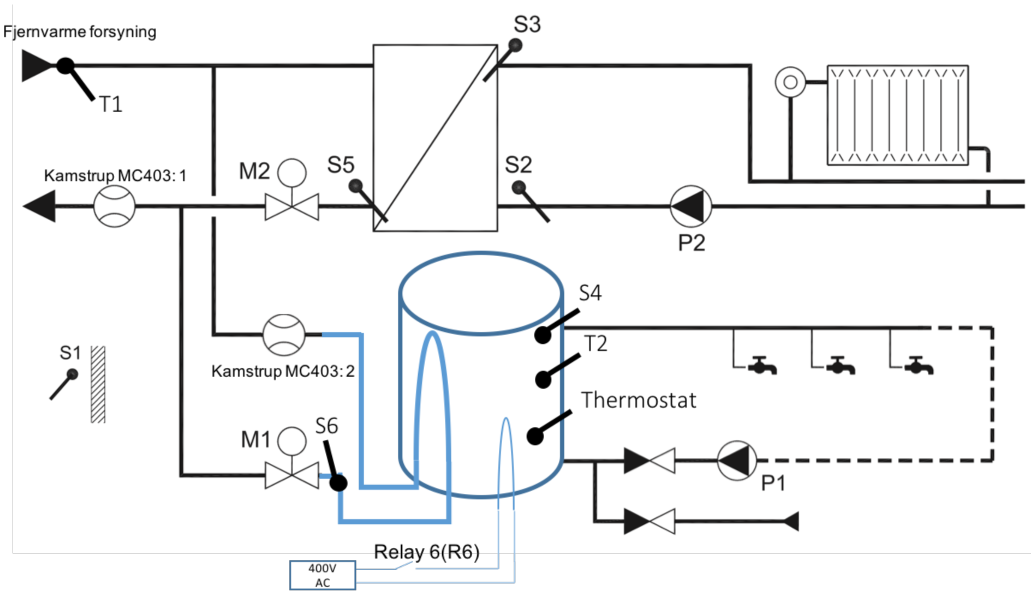

Figure 2 System including heat storage and electrical heater

The added sensors T1 and T2 are used by the ELN controller only, and not the ECL310. The sensor T1, is used to determine if the supply of DH is too cold, together with the temperature T2, measured in the tank, are used to determine whether electrical heating should be used to compensate. How the ELN controller acts on the measurements from T1 and T2 is described in a later section.

Two flow energy meters called Kamstrup MC403:1-2 on Fig. 2, are added to the system. The purpose is to differentiate between the energy used for DHW and for room heating. When using electrical heating, the DH consumption for DHW is expected to be correspondingly lower.
The first energy meter measures the forward- and return temperatures of the DH water as well as the flow. Based on this, it calculates the household's total energy consumption in kWh, which is used for billing. The measurement T1, the DH forward temperature, is accessed through Kamstrup MC403:1 to avoid using an additional temperature sensor. The second energy meter calculates the energy consumption for heating DHW. This additional second energy meter is installed purely for analyses of demonstration data and is not required for controller operation.

T2 is an additional sensor used by the ELN controller, to determine the water temperature in the tank. Instead of using S4, used by the ELC310, T2 is placed lower in the tank and therefore measures the temperature at another water depth. In the tank is also a thermostat, which limits the electrical heating to 65°C in that height.

2.3 Components and communication

In Fig. 3, the different active components of the whole fuel shift system, is shown. The ELN controller will run on a Raspberry Pi 3, connected to a Meraki router (from Cisco), the ECL310 and a power meter. The Meraki router will facilitate an encrypted internet connection (wired or wireless) to the data warehouse,

developed for this project. All data will be logged in the data warehouse and through this connection, the control signals will be sent. An electrical power meter of the model Kamstrup 382M, is used to measure the power consumption, at all times, as well as the total energy consumption used for electrical heating. The power meter is connected to the Raspberry Pi through an RS232/USB converter. The energy flow meters are of the type Kamstrup MultiCal 403. They use a 2-wire communication standard called M-bus which can be read by the ECL310. All measurements and flow energy meters are therefore read through the ECL310 with Modbus-TCP except the electrical power meter that is read directly with RS232/USB.
Fig. 3 shows that the ECL310 controls Relay 6 (electrical relay), M1, M2, P1 and P2, but M2, P1 and P2 are not used by the ELN controller but is a part of the ELC310 standard setup. M1 is used to close the DH supply to the DHW when using electrical heating.

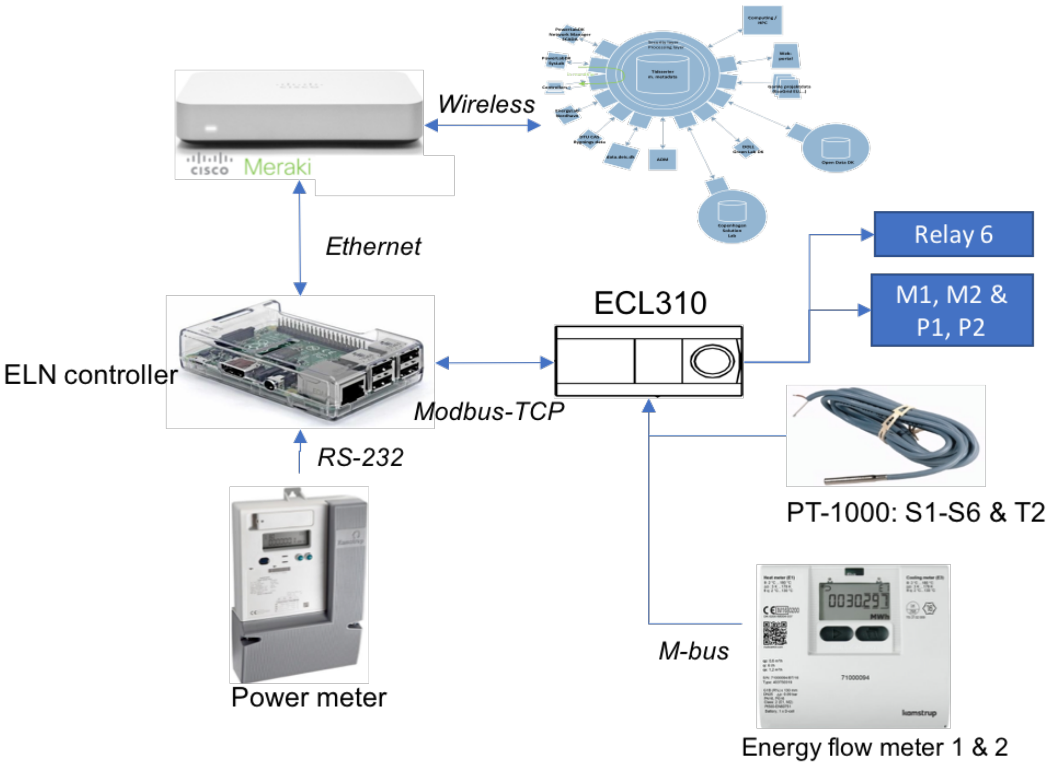

Figure 3 Active components in the ELN controller setup

The location of the new sensors can be seen in Fig. 4.

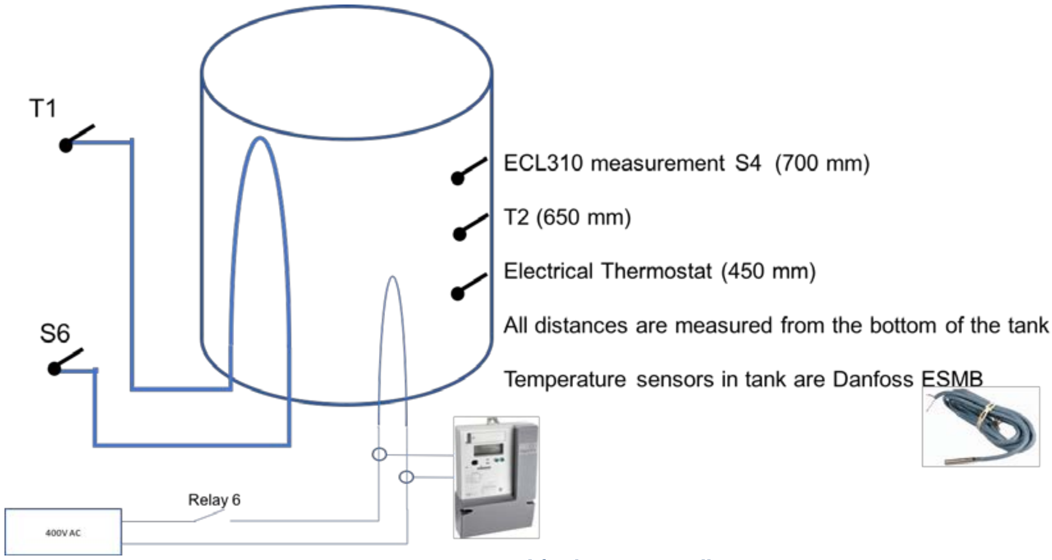

Figure 4 Sensors used for the ELN controller

The following measurements are logged by the platform, and the interval shows how often the values are updated.

Sensor	**Type**	**interval**	**Description**
S1	temperature	1s	Outdoor temperature
S2	temperature	1s	Domestic heating return
S3	temperature	1s	Domestic heating forward
S4	temperature	1s	Inside boiler (700 mm)
S5	temperature	1s	District heating return from heat exchanger
S6	temperature	1s	District heating return from boiler
T2	temperature	1s	Inside boiler (650 mm)
MC403:1	flow	1s	District heating
MC403:1	temperature	1s	District heating forward
MC403:1	temperature	1s	District heating return
MC403:1	power	1s	District heating
MC403:1	acc. volume	1s	District heating
MC403:1	acc. energy	1s	District heating
MC403:2	flow	1s	Boiler forward
MC403:2	temperature	1s	Boiler forward
MC403:2	temperature	1s	Boiler return
MC403:2	power	1s	Boiler
MC403:2	acc. volume	1s	Boiler
MC403:2	acc. energy	1s	Boiler
382M	V1	1s	Electric heater voltage phase1
382M	V2	1s	Electric heater voltage phase2
382M	V3	1s	Electric heater voltage phase3
382M	W1	1s	Electric heater active power phase1
382M	W2	1s	Electric heater active power phase2
382M	W3	1s	Electric heater active power phase3
382M	VAR1	1s	Electric heater reactive power phase1

382M	VAR2	1s	Electric heater reactive power phase2
382M	VAR3	1s	Electric heater reactive power phase3
382M	W	1s	Electric heater total active power
382M	VAR	1s	Electric heater total reactive power
382M	kWh	1s	Electric heater accumulated energy

3. Description of ELN controller functionality

This section describes the additional functionality that is added to the standard DH system by the ELN controller. The overall goal is to heat up the DHW to 55°C when using DH and 65°C when using electric heating. ECL310 uses S4 as the controller variable and will try to achieve 55°C measured at S4.
The ELN controller chooses its course of action based on 4 factors, which are the Boolean inputs FUEL_SHIFT, SINGLE_BOOST and the measurements T1 and T2. The input SINGLE_BOOST is a tuning parameter that can be set differently for different installations, and not changed at runtime. The following shows an overview of how the ELN controller acts, based on the 4 inputs,

- The ELN controller acts differently for hot DH (T1=>60°C) or cold DH (T1<60°C).
- If DH is cold, it must be at least 10°C warmer than the water in the tank at T2 to be able to transfer the necessary amount of energy. DH is therefore only used if T2<T1-10°C.
- When the DH is cold and DHW has been heated to the point T2>=T1-10°C, the electrical heating will be used.
- If SINGLE_BOOST is enabled, electrical heating will only heat the water to 65°C one time, until the thermostat deactivates the heating element, and then wait until the water in the tank become cold enough to use DH again. Otherwise will the electrical heating maintain the temperature measured at the thermostat at 65°C.
- If FUEL_SHIFT is enabled electrical heating will be used solely.

The functionality described above can be made into 4 cases, representing the possible combinations of the inputs.

3.1 CASE 1: Hot DH

If the temperature of the DH, measured at T1, is higher than or equal to 60°C, the target temperature at S4 will be set to $S4_{target}$=55°C in the ECL310. This is called the comfort set-point and the default value is 55°C. Relay 6 will not be enabled, M1 is set to auto and the ECL310 will heat the DHW autonomously. This is seen in Fig. 5.

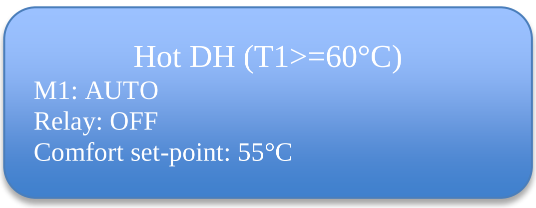

Figure 5 Actions during hot district heating

3.2 CASE 2: Cold DH

If the temperature of the incoming DH, measured at T1, is below 60°C, it will not be able to heat up the DHW to 55°C, but can heat the water to a temperature 10°C lower than the DH temperature, equal to T1-10°C. To avoid that the ECL310 completely opens the valve, M1, trying to achieve 55°C, the comfort set-point is lowered to T1-5°C. When the sensor T2 reaches the temperature T1-10°C, the ECL310 is asked to close valve M1 and enable relay 6, where the electrical heater is connected.
The electrical heater will maintain the temperature close to 65°C, via its thermostat. This is done by disabling the power when the temperature is above 65°C and enabling it again when it drops below. If a large consumption of the DHW occurs, and the temperature drops below T2<T1-14°C, the controller will disable

the relay and set the valve M1 on AUTO so DH again can be used to heat the water. The temperature differences are chosen as a hysteresis avoiding the controller to change state too often. This is seen in Fig. 6.

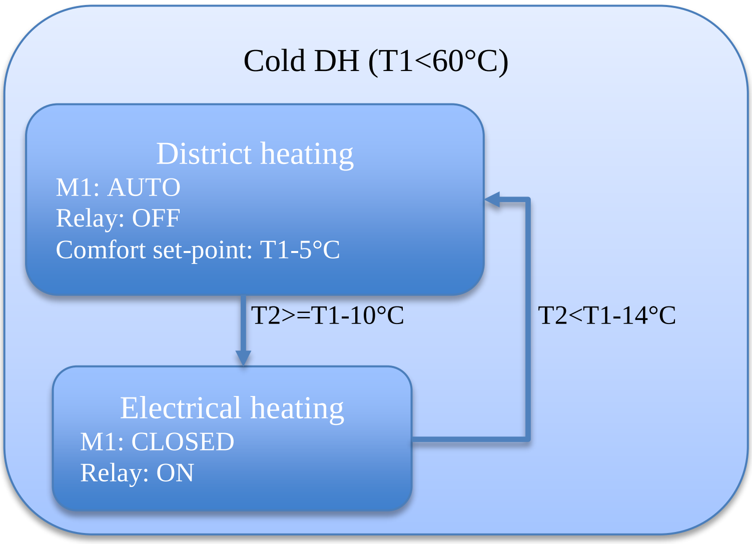

Figure 6 Actions during cold district heating

3.3 CASE 3: Cold DH Single Boost

The same as in case 2, except, the electric heater only boost the water temperature to 65°C once and then waits until the water is cold enough to use DH again.

When the electrical heater has increased the water temperature to 65°C, the thermostat will temporarily disable the relay, so the power consumption becomes equal to zero. When the measured power consumption is equal to zero, the ELN controller moves to a waiting state. In this state, the electric relay is disabled by the ELN controller, while the valve M1 is also closed. There will not be any input energy to the system until the temperature DHW is cold enough to use DH, namely when T2<T1-14°C. In cases where the consumption is low and the cooling takes a very long time, the controller should always go back to the boost state after 48 hours, to avoid growth of Legionella in the water. By the recommendation of health authorities, the water need to be heated to a temperature above 50°C to deal with the Legionella bacteria. This can be seen in Fig. 7.

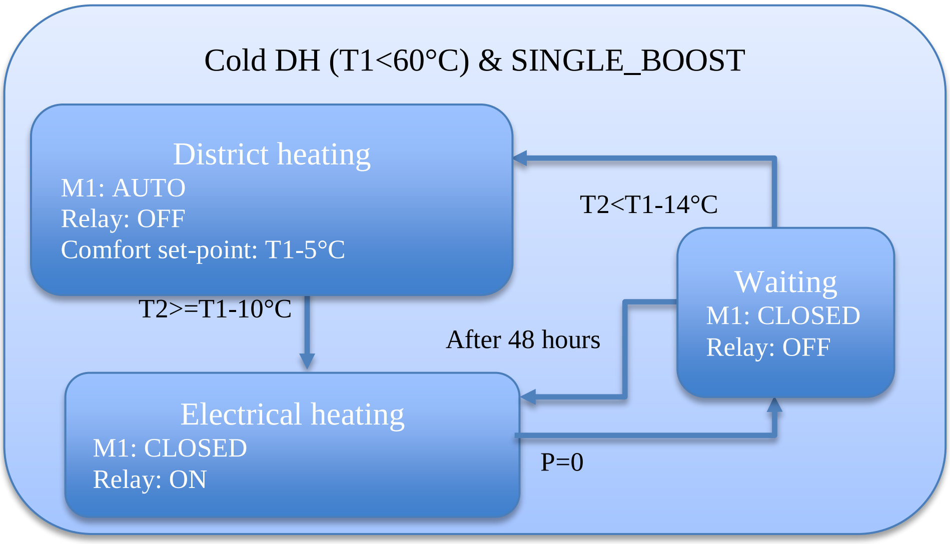

Figure 7 Actions during cold district heating when single boost is enabled

3.4 CASE 4: Fuel shift/Only electric heating

When the input, FUEL_SHIFT (FS) is enabled, valve M1 will be closed and relay 6 will be enabled. The heating elements own thermostat will maintain the DHW temperature around 65°C. This temperature can be changed manually by adjusting the setting on the water tank.
When FUEL_SHIFT is disabled, the controller will go back to case 1 to 3, depending on the other variables. The ELN controller will first disable relay 6 and set M1 back to auto. If the DHW is heated to 65°C, it is too hot to use DH and will result in to hot (high) return flow. This is handled by the ELC310 which limits the DH return water temperature to 50°C, so it will not use DH before the water is cooled. This is see in Fig. 8.

Figure 8 Actions when FUEL_SHIFT is enabled

3.5 From cases to a state machine

To enable the controller to go from one case to another during operation when the DH temperature or the request for fuel shift changes, it is implemented as a state machine.
The state machine consists of 4 states, defined by their actions. The relation to the cases are determined by the AND-statements that needs to be true to allow it to change state. This state machine is the one implemented in the software that makes up the ELN controller. This is see in Fig. 9.

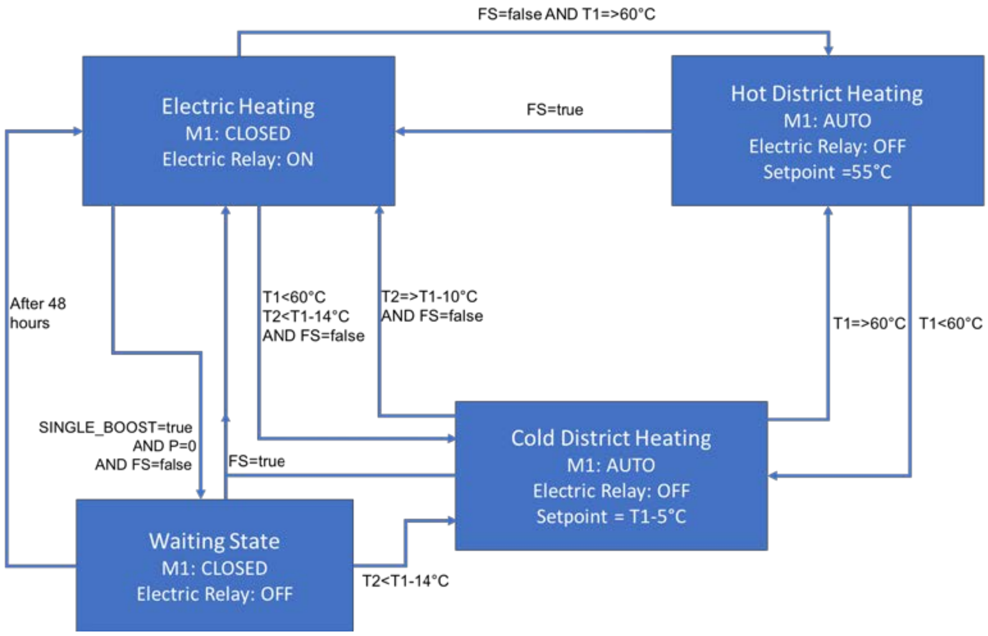

Figure 9 State Machine

The system is running autonomously without outside control, except when the FUEL_SHIFT status is changed. The people living in the apartments in Nordhavn will not experience a lack of heat supply even if the internet communication is lost. The worst case would be that FUEL_SHIFT is enabled and the internet connection is lost for a longer period, because electricity is a more expensive source for heating. The system will therefore always set FUEL_SHIFT=false if it has not received a command from the centralised control in the last hour.

4. ELN controller software implementation

The ELN controller has been implemented as a highly configurable- and modular platform, written from the ground up in Java. The choice of language stems from existing software platforms developed- and employed by CEE, as well as the inherent cross-platform friendliness of the Java Virtual Machine (JVM).

An illustration of the overall platform architecture, as well as the various modules developed to fulfil the requirements for deployment in Nordhavn, can be seen in Figure 10. Although the modules all share the same basic interface, in order to be discoverable- as well as loadable by the platform core, they can be viewed as belonging to separate layers of the same hierarchy: the hardware access layer, the control layer and the communication layer.

When started, the platform core loads the modules specified in its configuration and performs the necessary bindings. These bindings, which are specified in the configuration for the individual modules, is what allows the modules to communicate with each other. The bindings can, in other words, be seen as a form of data channel. Though there are different types of modules supported by the platform, they all have the ability to support binding to one another. The modules loaded- as well as their mutual bindings, is what dictates the capabilities of the deployed platform.

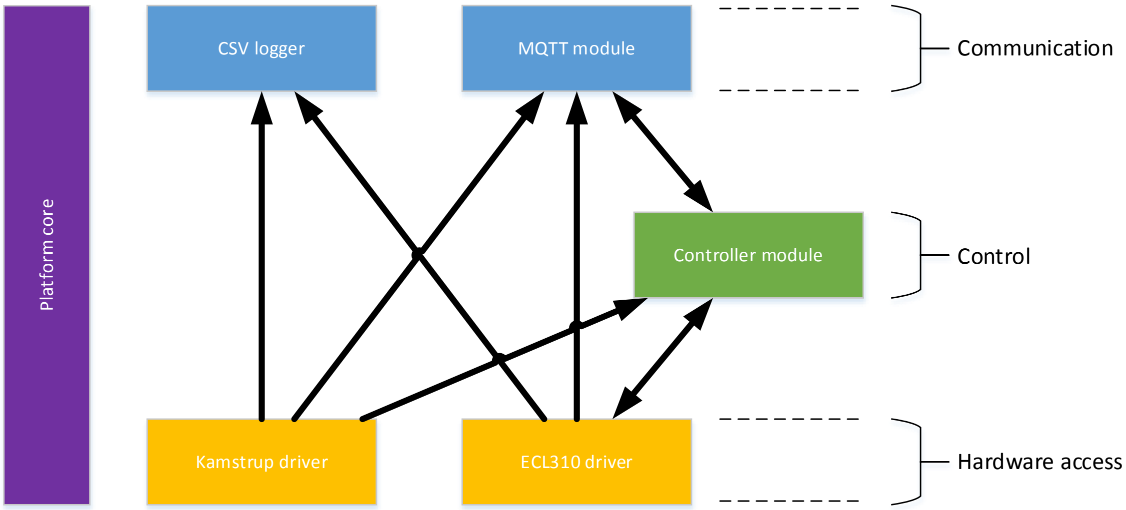

Figure 10 ELN controller – architectural overview

The bindings is one of the strengths of the platform architecture, and allows for a fully event driven data flow, from when the data is read from hardware- to when it leaves the platform through whatever modules is currently configured to communicate it.

4.1 Hardware

As previously discussed, the ELN controller setup has the platform retrieve measurements from- and directly control two pieces of external hardware: the Kamstrup 382M energy meter and the ECL310. The communication with these are implemented in the *Kamstrup*- and *ECL310 driver* modules respectively and do, as such, not bind to any other modules. Since neither the Kamstrup meter nor the ECL310 support event based updates, both modules implement a polling mechanism that keep their internal process images updated. Values can be updates by other modules bound to the hardware layer, allowing the control to update set-points at runtime. The measurements that are retrieved from hardware, is subscribed to by the modules that need them.

4.2 Control

The core logic of the ELN controller setup, as described in section 3, is implemented in the *Controller* module. When loaded- it is bound to both hardware modules and subscribes to the specified measurements needed to run the previously discussed state machine. The current implementation keeps the state machine updated using a timer, but could as well be implemented asynchronously, triggering on any change in the data coming from either the underlying hardware- of upper communication layer.

As previously discussed, the flexibility of the platform allows for numerous combination of modules, even to achieve the same goal. Presently, the core logic is implemented in a module, but that logic could also be implemented externally, controlling the hardware directly through e.g. the *MQTT client* module (or some other future communication channel). The control algorithms scheduled for development in T4.2, could also be made to run separate from the platform. It has, however, been decided that the most production-like scenario, would be to implement them in another control module, which is bound to the one running the core logic described above and in section 3.

4.3 Communication

There are two additional modules in the ELN controller setup, namely the *CSV logger* and the *MQTT client* modules. The *CSV logger* does not technically communicate, but has been included here to simplify the platform architecture. It is bound to both hardware modules and is, through the bindings, able to auto-enumerate all available measurements, which are then logged to a file. The basic implementation of the logger will ensure that the logs are rotated at midnight as well as maintain the integrity of the column format. If any of the bound modules are updated during the same day (which currently requires a platform restart), the logger will compare the column format and create a new log file. This file is automatically appended with a numeric, auto-increasing postfix, which will reset again at midnight, when the next rollover happens.

The *MQTT client* modules, as the names implies, establishes a connection to a configured MQTT message broker. At the time of this writing, the official data warehouse for Nordhavn had not yet been completed, so a temporary broker was setup. To avoid communication mishaps, the data sent via the *MQTT client* module, is serialized using Apache Thrift. It is important to note, that other serialization schemes were considered (e.g. Google’s Protocol Buffers and basic JSON) and while Thrift is currently used, it may be replaced when the data warehouse comes online.

5. ECL310 communication

The benefit of most sensors and all actuators being connected to ECL310, is that the communication can be done using a single Modbus-TCP connection.
In the Modbus TCP protocol, it is necessary to know the IP address and the unit ID of the slaves, the latter of which is mainly matter when multiple slaves are present at the same IP/port. The units have a number of registers where it is possible to either read or write information. The standard allows for reading from multiple consecutive registers at the same time.
This section contains information about the different quantities that can be read or controlled through the ECL310.

5.1 Read Flow Energy meters

The ECL310 can read from up to 5 energy meters using the European standard for electricity and gas meters, EN 13757-2, called Meter bus or M-bus. The M-bus is a two-wire protocol that requires specific hardware to read. The ECL310 does not use the information from the meters but is functioning as a gateway. The energy meters have a specific ID which can be found by running the ECL310 scan function. When the meters are found, they can be accessed via the interface on ECL310, as seen in Fig. 11.

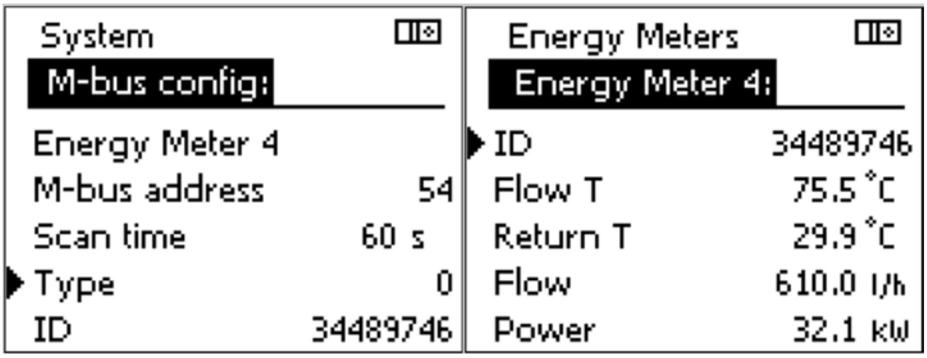

Figure 11 Example of M-bus configuration (left) and data from energy meter (right)

The Modbus registers for accessing the meter readings, are shown in Fig. 12. The communication documentation uses a Parameter Number (PNU), which is the Modbus register minus 1. As indicated in the figure, the first address for energy meter 1 is 6000. The measurements from energy meter 2 is accessed by adding 50 to the registers used for energy meter 2.

ECL Comfort Parameter	Description	PNU	Access	Scale
M-bus Energy meter 1				
Address		6000		
Type		6001		
Scan time		6002		
ID / Serial		6003		
- low part		6004		
Reserved		6005		
Flow temperature		6006		
Return temperature		6007		
Flow		6008		
- low part		6009		
Power		6010		
- low part		6011		
Accumulated Volume		6012		
- low part		6013		
Accumulated Energy		6014		
- low part		6015		
Dynamic part		...		
Last pnu		6049		
M-bus Energy meter 2		6050		
..				

Figure 12 ECL310 PNU regarding Energy meter 1 and 2

5.2 Read sensors

The ECL310 can read from 10 PT1000 temperature sensors and the values can be read from the register addresses seen in Fig. 13. When reading the temperatures, the value should be divided with 100 to get the real temperature. The scaling is the right column and the second right column means that the measurements are read only.

ECL Comfort Parameter	Description	PNU	Access	Scale
S1 sensor Outdoor temperature		*11201*	*R*	*100*
S2 sensor Room heating return temperature		*11202*	*R*	*100*
S3 sensor Room heating forward temperature		*11203*	*R*	*100*
S4 sensor Forward temperature, (DHW)		*11204*	*R*	*100*
S5 sensor DH return temperature (Room heating)		*11205*	*R*	*100*
S6 sensor DH return temperature (DH)		*11206*	*R*	*100*
S7 sensor unused		*11207*	*R*	*100*
S8 sensor unused		*11208*	*R*	*100*
S9 sensor T2		*11209*	*R*	*100*
S10 sensor unused		*11210*	*R*	*100*

Figure 13 ECL310 PNU regarding S1 to S10

5.3 Set DHW setpoint

When T1<60°C, the comfort temperature is set to T1-5°C, otherwise it should be set to 55°C. This is done by writing to the comfort temperature register 12189. The set-point is multiplied by 10 as seen in Fig. 14.

ECL Comfort Parameter	Description	PNU	Access	Scale
Comfort DHW setpoint		**11190**	R/W	10

Figure 14 ECL310 PNU regarding Comfort DHW setpoint

5.4 Valve control

The motor valve M1, controlling the DH flow to heat the DHW, is connected to TRIAC 1 and 2, which is AC connection 6 and 7, as seen in Fig 15.

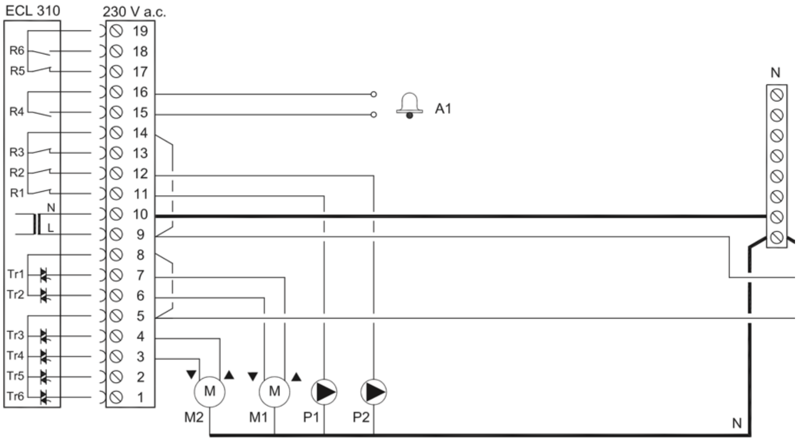

Figure 15 Connection of actuators to ECL310

TRIAC 1 is controlled via the R/W register 4059 in the ECL310. To control the valve, M1, the values that can be written to 4059, is as follows:

- '0'=Auto
- '1'=Stop
- '2'=Close
- '3'=Open.

The status of M1 can be read from the same register and is used to check if the command needs to be sent again. Writing to TRIAC 2 would result in the valve moving in the opposite direction. The PNUs of the different TRIACS are seen in Fig. 16.

ECL Comfort Parameter	Description	PNU	Access	Scale
Override status, Triac 1		4060	RW	
Override status, Triac 2		4061	RW	
Override status, Triac 3		4062	RW	
Override status, Triac 4		4063	RW	
Override status, Triac 5		4064	RW	
Override status, Triac 6		4065	RW	

Figure 16 PNU regarding triac control

5.5 Relay control

The relay used to enable the electrical heater is relay 6 with the address 4069. The relay is either ON or OFF. The relay is set to ON by writing 2 to 4069 and to OFF by writing 1 to 4070. '0'=Auto, '1'=OFF, '2'=ON. The status of the relay can be read from the same register and is used to check if the command needs to be sent again. The PNU address can be seen from Fig. 17.

ECL Comfort Parameter	Description	PNU	Access	Scale
Override status, Relay 1		4066	RW	
Override status, Relay 2		4067	RW	
Override status, Relay 3		4068	RW	
Override status, Relay 4		4069	RW	
Override status, Relay 5		4070	RW	
Override status, Relay 6		4071	RW	

Figure 17 PNU regarding relay control

6. Physical setup in Nordhavn

The fuel shift system will be tested in newly build apartments in Nordhavn, where up to 24 apartments are participating in the experiment. The left picture in Fig. 18 shows the standard DH installation in the apartments and the right picture shows what is changed for the residents participating in the ELN demonstration. The water tank on the picture is a model 110 (~95 litre), but in some apartments, it will be a model 160 (~140 litre) tank. In both cases, the standard district heating tanks have been pre-fitted with a corresponding tank containing an additional electric heater with thermostatic control. The DH system is normally controlled by an ECL210 unit, but it is replaced with the more advanced ECL310. which have Ethernet communication.

The electrical power meter, Raspberry Pi and 5 V DC power supplies will be placed in a water proof container under the standard ventilation unit together with the Cisco Meraki router, responsible for the external communication. The Meraki router acts as a local switch connecting the devices locally.

Figure 18 PNU Residential DH system in Nordhavn

The layout of the water proof container is described in the following section.

7. Control box for Nordhavn

For the setup is required a space for the additional hardware delivered by the project.

A dust and water resistant box is chosen for this purpose. The dimensions are 130x560x285mm DxWxH and it is to be installed in Nordhavn under the air ventilation unit.

Internally is following: electrical connection points, fuse, relay, electrical power measurement, the Raspberry Pi and a 5V power supply. Outside is the Meraki router and in front of the box is mounted a 230V plug supplying this.

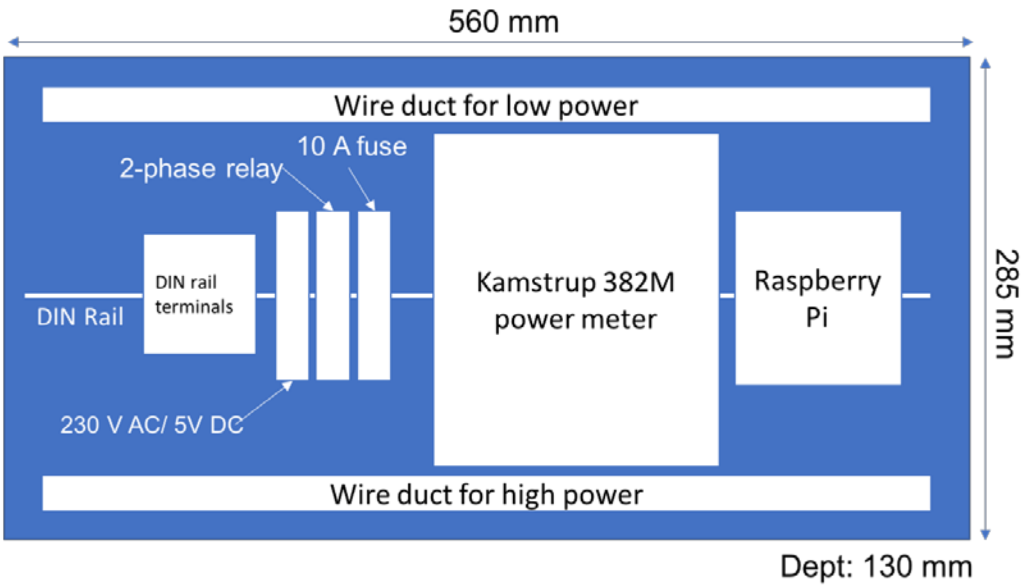

Figure 19 Box containing measurement equipment

Fig. 20 shows the electrical diagram for the different components. It specifies how the power meter and relay is connected to two of the phases, as the heating element is 2-phased.

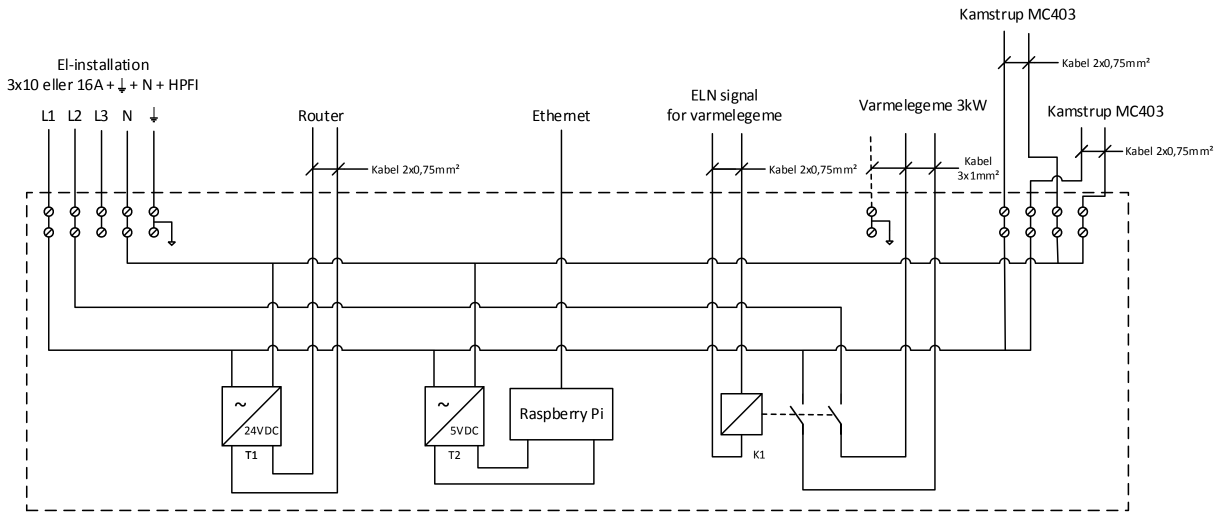

Figure 20 Electrical diagram

8. Physical setup in SYSLAB

A complete copy of the Fuel shift setup is made at Risø Campus in the test system, SYSLAB. The fuel shift setup controls the water heating of Power FLexhouse 2. The layout of the building is seen in Fig. 21.

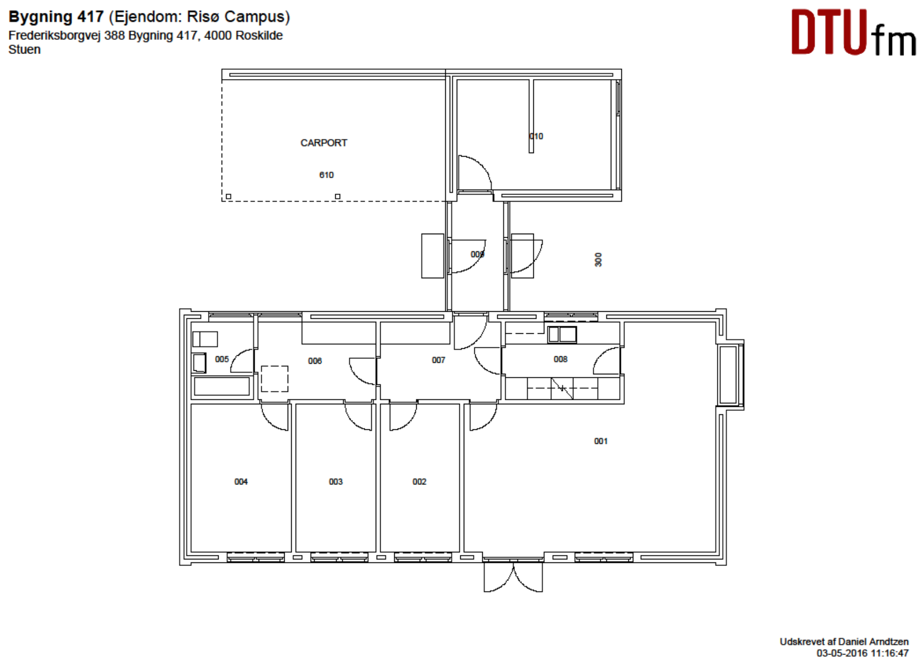

Figure 21 Power Flex House 2

The location of the units is Power Flex house 2 can be seen in Fig. 22.

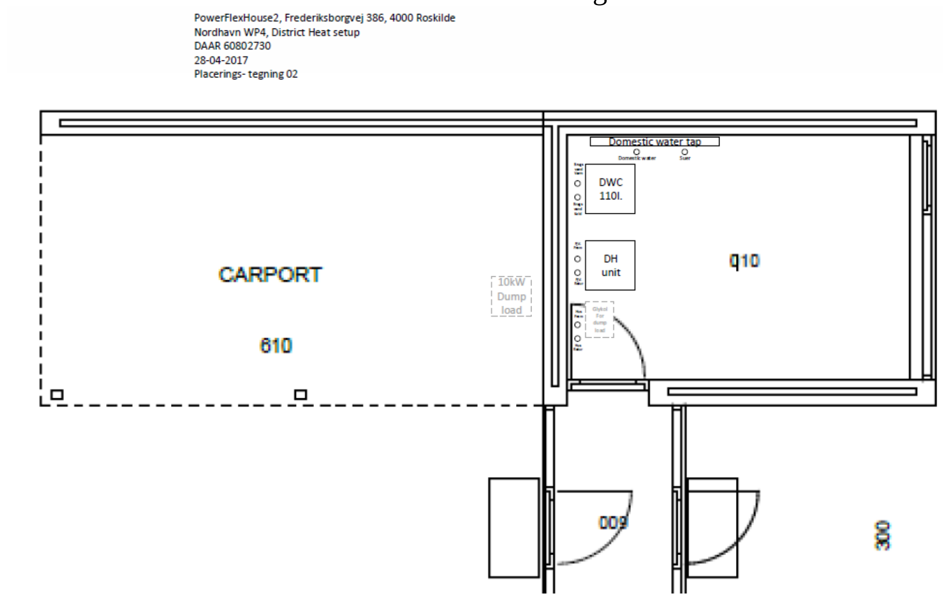

Figure 22 Location of components

The fuel shift setup in Power Flexhouse 2 with all the components is seen in Fig. 23.

Figure 23 Setup with component names

9. Test against the Danish Standard for water requirements

Danish Standard 439 (DS439) describes a pattern of water consumption for different types of settlements. In this case we use the pattern for a flat with shower only. During which the DH system should be able to maintain a certain temperature. If the system lives up to the DS439, are the heating capabilities satisfying enough that it can be used in the residential apartments. Any fuel shift algorithm will be tested against the DS439 before it is deployed in Nordhavn. In the following figure, the system installed in SYSLAB is seen together with the test system and the additional sensors that only are in the SYSLAB installation and not the Nordhavn installation.

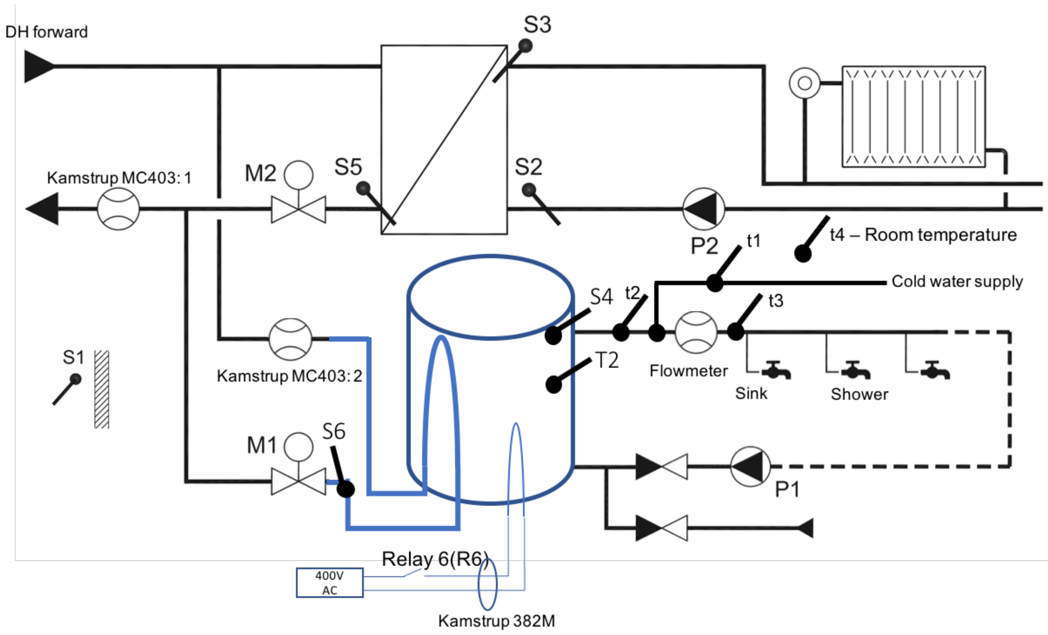

Figure 24 SYSLAB setup including test setup

The water consumptions algorithm is run on a PLC and the additional measurements are measured by that. The following table shows all the additional measurements that are made for testing the system.

Measured through PLC			
t1	temperature	1s	Cold water supply for drain
t2	temperature	1s	Hot water supply for drain
t3	temperature	1s	Mixed water supply for drain
t4	temperature	1s	Room temperature
Flowmeter	flow	1s	Whole liters drained
Sink	ON/OFF	1s	Position of valve
Shower	ON/OFF	1s	Position of valve

10. System Acceptance Test

In order to ensure correct system behaviour according to requirements described in section 3, a number of scenarios has been designed for validation. The scenarios are designed such that the state machine experiences all possible state changes. The tests were performed on the final application of the ELN controller as a part of the platform.
As the ELN controller is installed in the SYSLAB laboratory at PowerLabDK DTU Risø, it is possible to control the temperature of the DH supply to the house. A situation with cold DH forward flow temperature (T1<60°C) and a situation with hot DH forward flow temperature (T1>=60°C), are both recreated in the SYSLAB laboratory. The tank temperature T2 can be manipulated by draining the hot water, which is replaced with cold water, like when the residents take a bath. DHW temperatures must fulfil the Danish standard of drawing requirements DS439 for acceptance.

10.1 Cold DH, does it use electric heating after heating with DH?

Prerequisites:
T1 < 60 (58), T2 < T1-10 (45). FS = off, Single_boost = off
Expected:
Initially ➔ flow on DH and electrical relay is off. Change comfort set-point to T1-5 (53°C).
When T2>T1-10 (48°C) ➔ M1 closed and electrical relay on.
While T2 < 65 ➔ heating coil will use power.
Conclusion:
Everything happened as expected. DH flow is verified from the flow meter. Comfort set-point is verified from ECL310 display.

10.2 Does controller use DH again after a large water drain?

Prerequisites:
T1 < 60 (58), electrical=ON, Thermostat=OFF. FS = off, Single_boost = off. Use so much water that T2<T1-14.
Expected:
Initially ➔ M1=CLOSED and electrical relay=ON.
When T2<T1-14 ➔Relay=OFF and M1= AUTO.
When T2>T1-10 (48) ➔M1=CLOSE and Electrical relay= ON.
While T2 < 65 heating coil will use power.
Conclusion:
After the large drain did the system start using DH again, as long as T2 was cold enough. Then It went back to electrical heating.

10.3 Do we go back to HOT district heating?

Prerequisites:
T1 < 60 (58), FS = off, Single_boost = off. Increase DH forward flow temperature to T1 >= 60.
Expected:
Initially ➔ M1=AUTO and relay OFF. Comfort set-point=T1-5.
When T1>=60 ➔ Change comfort set-point to 55°C. M1=AUTO, electrical relay=OFF
Conclusion:
The system went back to the hot DH state and changed the comfort set-point accordingly.

10.4 Enable/disable FUEL_SHIFT(FS) during HOT DH

Prerequisites:
T1 >= 60, FS = off, Single_boost = off. Enable FS.
Expected:
Initially ➔ M1=AUTO and Electrical relay=OFF. Comfort set-point=55°C.
When FS=ON ➔ M1=CLOSED, Electrical Relay=ON
When FS=OFF and T1>=60°C ➔ M1=AUTO and Electrical relay=OFF. Comfort set-point=55°C.
Conclusion:
Fuel shift was enabled and disabled successfully during hot DH.

10.5 Enable/disable FUEL_SHIFT(FS) during COLD DH

Prerequisites:
T1 < 60, FS = off, Single_boost = off. Enable FS.
Expected:
Initially ➔ M1=AUTO and Electrical relay=OFF. Comfort set-point=T1-5.
When FS=ON ➔ M1=CLOSED, Electrical Relay=ON
When FS=OFF and T1<60°C ➔ M1=AUTO and Electrical relay=OFF. Comfort set-point=T1-5
Conclusion:
Fuel shift was enabled and disabled successfully during cold DH.

10.6 DATA readings

All data is sent to a MQTT broker and are displayed as graphs through the Grafana visualisation software. All measurements have been crosschecked with display readings on the individual meters to ensure correct reading interpretation and mapping.

10.7 Conclusion

All the described tests had the expected output. The ELN controller logs its current state which makes it easier to identify that expected outcome has been accomplished. The functionality SINGLE_BOOST has not been tested, as it will not be utilized during the demonstrations in Nordhavn.