System documentation of a heating control system

This document has been produced, within the context of software engineering courses at the University of Kaiserlslautern by Marco Heide, Wolfgang Wagenbichler, Andreas Schank, Angela Schmidt and Axel Seck, then reworked by Christiane Differding and Matthias Gutheil. It has been translated in english and adapted by Mario Negro Ponzi of Politecnico di Torino. It has subsequently undergone extensive rework by Vincenzo D'Elia, Alfredo Pironti and Massimiliano Schillaci of Politecnico di Torino.

V 5.0 February 2015

Abstract

A heating control system monitors and controls temperature in a building composed of several rooms. It is a good example of a system with software and non-software parts. This document contains the system requirements, the system design, the software requirements and the software design

1.1 Table of contents

Part 2 Problem description

The heating control system should be developed, to control heating in a house, optimizing energy efficiency and comfort. Below the idea of this system is described.

2.1 Problem

Up until today the temperature of every room is regulated by a thermostat in the heater in the room. A target temperature for each room can be set. An automatic management of the temperature is based only on heaters. A window also is useful to control temperature in a single room. But it should be opened or closed by someone. Water in heaters has a fixed temperature set by a boiler.
Problems in temperature management are:

1. Heater’s temperature cannot be differently set basing on heating demand or time. There is no regulation linked with people presence
2. Thermal regulation is limited to heating. The use of windows to cool down is based on user intervention
3. There is no optimization of boiler temperature. The boiler’s temperature is set to the maximum value regardless whether the temperature is too low or too high. That produces an energy loss.

The above control system is not satisfying both because of energy loss and because it is not comfortable.

2.2 Expected functionalities

The user wants to be able to individually set temperature for each room using a terminal. So he should be able to provide described parameters.
First it should be possible to set a temperature that could be maintained when there is someone in the room. In doing so the system should not react immediately when someone comes in. It should be provided a time span that indicates how long a person should stay in the room before the system takes control. The user should also be able to set the time span in which the desired temperature should be reached. The provided time span can be maintained heating up heaters for a small period of time over the desired temperature.
Also the user should be able to set a minimum temperature that is acceptable when there is no one in the room. In this case, too, a time span should be provided, after everyone has left the room, before the system takes control.
There should also be default values available for every parameter.
Windows, too, should be controlled by the system. If a window has been completely opened by a user, the system should maintain only the minimal temperature. If the window is open and it starts raining the system should put it in tilt position. More, the windows should be usable by the system to cool down when the outside temperature is cooler, by tilting them.

2.3 Environment characteristics

Below are described the characteristics of the environment inside which the system will operate.

2.3.1 House and rooms

The house is composed of a number of rooms.
The house also has:

- an outside temperature sensor
- a rain switch (closed if it rains)

Each room is so configured (as a result of the whole system concept):

- 1 window with two switches and two actuators. A switch gets opened when the window is completely open (open when the window is open / NC = normally closed). The other when is tilted (open when the window is tilted / NC = normally closed). When the window is completely open the tilt switch is closed. An actuator swings the window while the other tilts it.
- 1 heater (hot water) with a thermostat

- 1 presence switch (closed when someone is in the room)
- 1 room temperature sensor

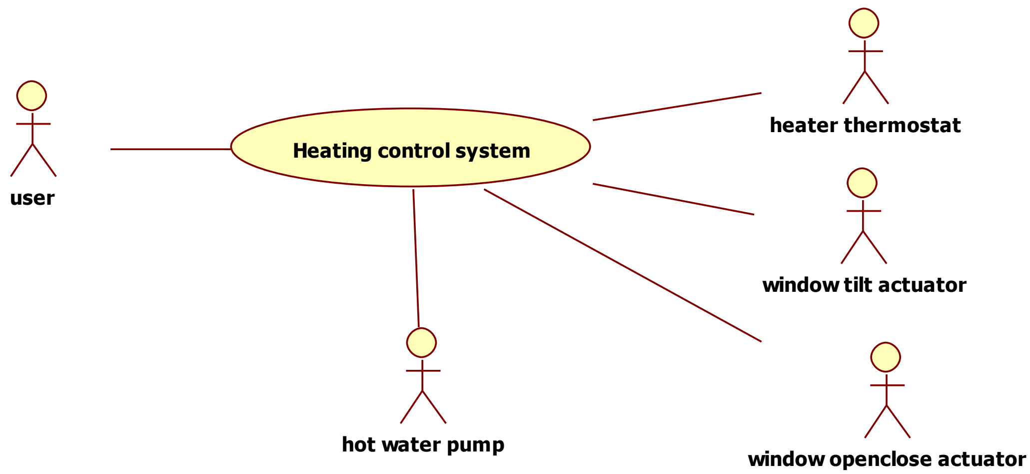
2.3.2 Boiler and water pump

Heat is produced by a centralized boiler, its control is an already deployed autonomous subsystem. The maximum temperature is fixed at 70 degrees. The house receives heat through a single pipe that brings hot water to the heaters. It can be assumed that there are no dispersions, so water is at 70 degrees. The heating control system controls the intake flow of hot water through a pump.

Part 3 System requirements

3.1 *Context diagram and interfaces*

The context diagram specifies what is meant by ‘System’ and what are its interfaces with the external world. See System design to know what is inside the system.



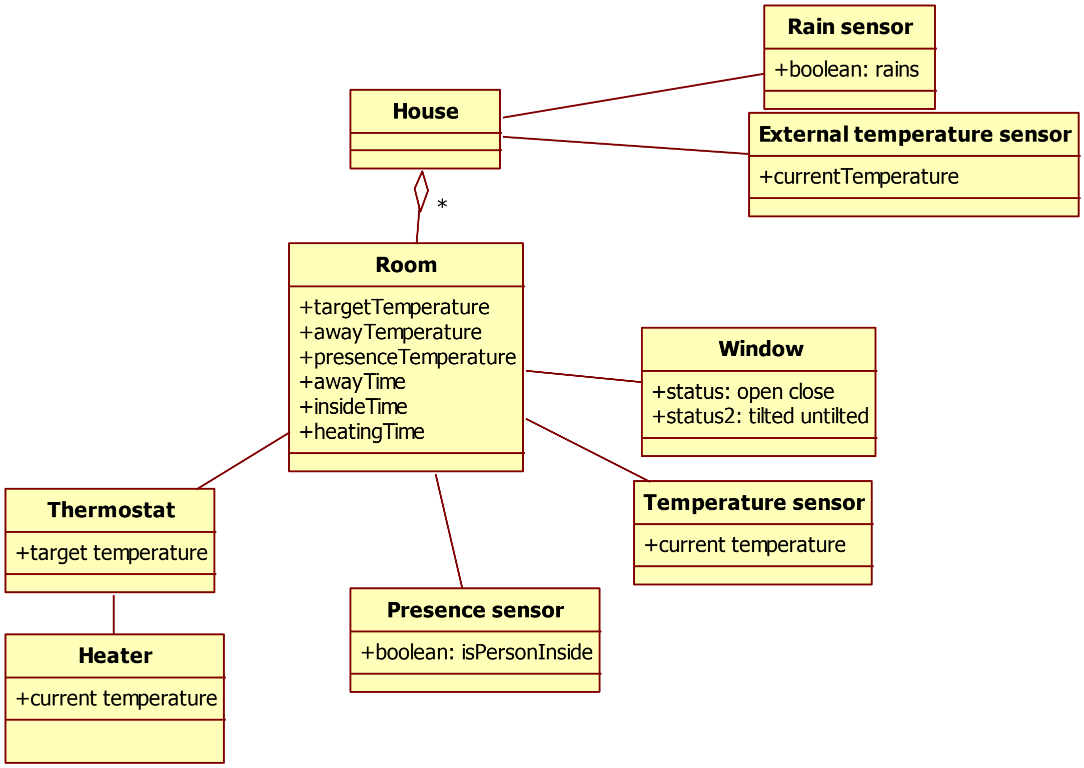
Interfaces

With object	Physical	Logical
User	Touch screen	Screens and buttons
Window tilt actuator	RS232	command protocol, see vendor tech spec main commands are read position, set position
Window open close actuator	RS232	command protocol, see vendor tech spec, main commands are read position, set position
Hot water pump	Bipolar wire, 12 V DC	close = pump ON, open = pump OFF
Heater thermostat	RS232	command protocol, see vendor tech spec, main commands are read temperature, set temperature

Remark that there is only one hot water pump, but one heater thermostat and window actuator per room.

3.2 *Glossary*

Here we describe the key parts and concepts in the case considered. The class diagram below does NOT discriminate what is inside from what lies outside the system.



A house has several rooms, a rain sensor (tells if it rains or not) and an external temperature sensor (reads temperature outside). A room has a heater (radiates heat from water to air) controlled through a thermostat (open closes a valve that controls water intake in the radiator, trying to keep a set target temperature), a window (can be open closed, or tilted), a presence sensor (tells if a person is inside or not), a temperature sensor (reads the current temperature in the room).

To control temperature in the room, a room defines

- Target temperature: desired temperature in the room
- Away temperature: what should be target temperature when no person is inside
- Presence temperature: what should be target temperature when a person is inside
- Away time: time to wait, after a person exits the room, before reducing target temperature to away temperature
- Inside time: time to wait, after a person enters the room, before rising target temperature to presence temperature
- Heating time: maximum time before which the room should get to target temperature after a person enters

3.3 Functional Requirements

FR ID	Requirement's description
FR 1	The user shall be able to set *PresenceTemp* in °C
FR 2	The user shall be able to set *AwayTemp* in °C
FR 3	The user shall be able to set a time span *HeatingTime* in minutes
FR 4	The user shall be able to set a time span *AwayTime* in minutes
FR 5	The user shall be able to set a time span *InsideTime* in minutes
FR 6	The temperature of FR 1 shall be maintained when there is someone in the room and the *InsideTime* from FR 5 has elapsed
FR 7	The temperature of FR 2 should be maintained in the room when the room is empty

	and *AwayTime* from FR 4 has elapsed
FR 8	The time span from FR 3 indicates after how many minutes the *PresenceTemp* from FR 1 after the entrance of a person in the room shall be reached.
FR 9	The time span from FR 4 indicates for how many minutes the *PresenceTemp* from FR 1 shall be maintained after the last person has left the room
FR 10	The time span from FR 5 indicates for how long shall someone stay in the room for the system to get in control
FR 11	The *HeatingTime* from FR 3 must be at least 5 minutes longer than the *InsideTime* from FR 5
FR 12	For each quantity from FR 1, FR 2, FR 3, FR 4, FR 5 default values shall exist.
FR 13	If a window gets opened the system maintains the room temperature at *AwayTemp*, whether or not there is someone in the room.
FR 14	If it starts raining with the window open, the system will set the window in a rain secure position.
FR 15	When the temperature of a room with someone inside exceeds that from FR 1 and the outer temperature is lower than the room temperature, the window in this room gets tilted
FR 16	The control of the temperature is admitted with an error of +/-1 °C
FR 17	The temperature of the boiler shall be set to the maximum required by any heater, not to waste energy.

Table 1

3.4 Non-functional requirements

The system should complain also with the following requirements.

NFR	Requirement's description
NFR 1	The system should be easily changeable.

Table 2

3.5 Inverse user requirements

Following situations should be avoided.

IFR	Requirement's description
IFR 1	When there is nobody in the room and the room is hotter than target temperature, the window does not get tilted to cool.
IFR 2	If the user manually opens the window, the system shall not tilt it to the rain secure position.

Table 3

3.6 Design decisions

Following decisions have been taken.

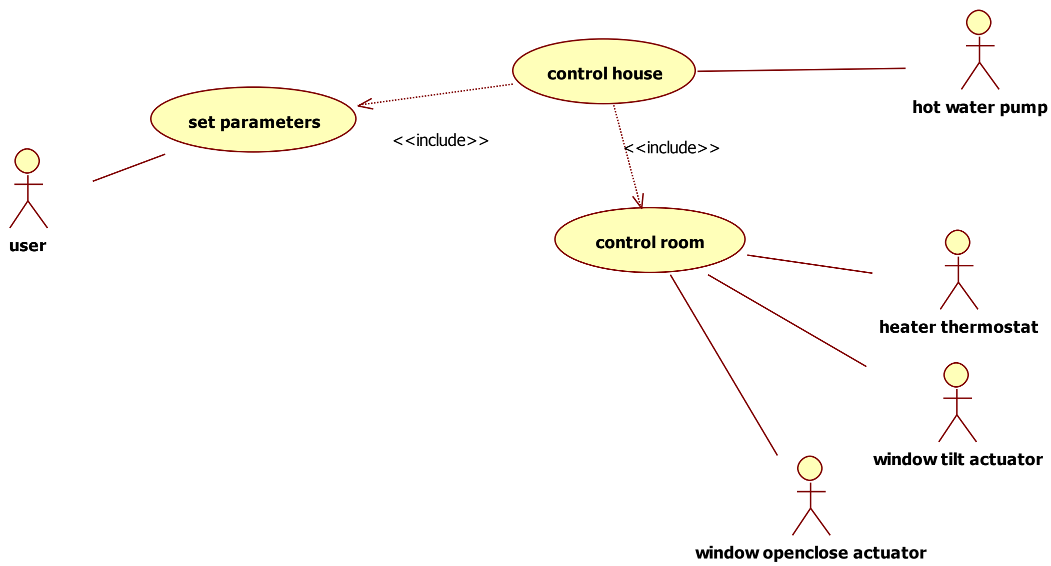
DD	Requirement's description
DD 1	The system should be object oriented.

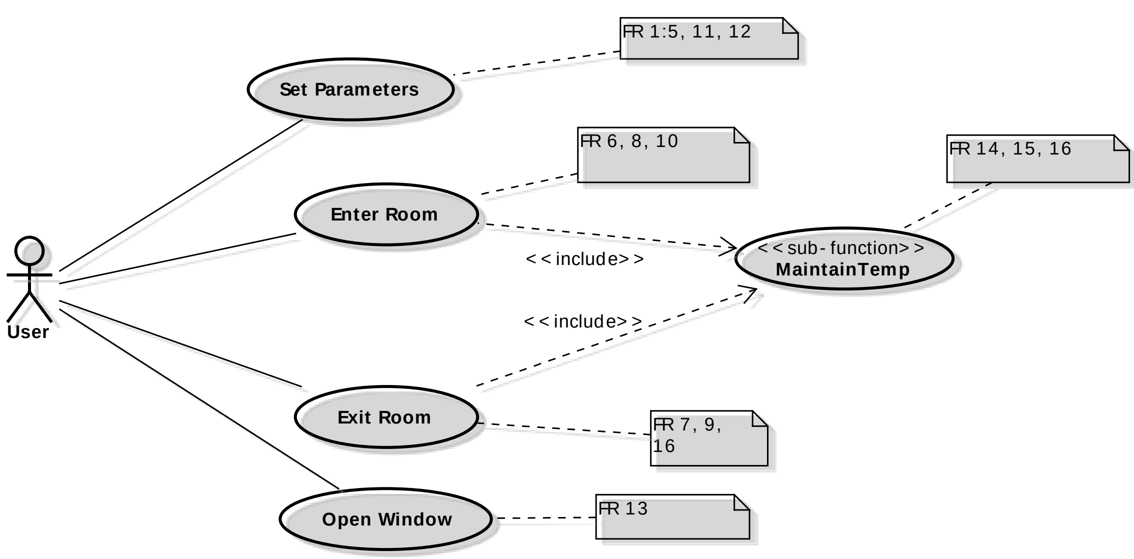
Table 4

3.7 Use cases

3.7.1 Use case diagrams

The main use cases are summarized in the following use case diagram.





Use Case: control house

It deals with overviewing all rooms and interacting with the water pump. It is composed of

- Scenario 12
- Scenario 13

3.7.2 Detailed use cases

Use case:	**Set Parameters**
Level:	User goal
Scope:	HCS User Interface
Intention in context:	The user wish to define the working parameters for the system
Stakeholder's concerns:	
Minimum Guarantees:	The parameter values are consistent
Success Guarantees:	The system's parameters set by the user become effective for the system.
Primary actor:	User
Support actors:	-
Precondition:	
Trigger:	
Main success scenario:	1. The user sets a value for InsideTime (FR5) 2. The system validates and confirms 3. The user sets a value for HeatingTime (FR3) 4. The system validates and confirms (FR11) 5. The user sets a value for AwayTemp (FR2) 6. The system validates and confirms 7. The user sets a value for AwayTime (FR4) 8. The system validates and confirms 9. The user sets a value for PresenceTemp (FR1) 10. The system validates and confirms
Extensions	1.a. :The user sets InsideTime to default 3.a. :The user sets HeatingTime to default 5.a. :The user sets AwayTemp to default 7.a. :The user sets AwayTime to default 9.a. :The user sets PresenceTemp to default 2.a. the InsideTime time is less then 5 minutes smaller than HeatingTime: the system ignores new time set, and throws an error message (FR11) 4.a. the HeatingTime is less then 5 minutes larger than InsideTime: the system ignores new time set, and throws an error message (FR11)

Use case:	**Enter room**
Level:	User-goal
Scope:	Room
Intention in context:	The user enters the room and the HCS must set temperature to PresenceTemp
Stakeholder's concerns:	*: avoid immediate reaction when just passing by user: have the room achieve required temp.
Minimum Guarantees:	Heating is not started before InsideTime has passed
Success Guarantees:	The heating is started The room achieves the required PresenceTemp within HeatingTime The temp is then maintained at the required temp.

Primary actor: User

Support actors: InternalThermostat WindowTiltSensor, OutsideTempSensor

Precondition: No user in the room since at least AwayTime

Trigger:

Main success scenario:
1. The users enters the room[1]
2. After InsideTime, the target temp is set to PresenceTemp
3. HCS maintains the temperature (UC MaintainTemp)

Extensions: 2a. The user exits the room before InsideTime has elapsed: the use case terminates in failure.
2b: The user opens the window: the target temperature is set to AwayTemp

Use case: **Exit room**
Level: User-goal

Scope: Room

Intention in context: The user exited the room and the HCS must set temperature to AwayTemp

Stakeholder's concerns: *: avoid immediate reaction when just passing by
user: have the room achieve required temp.

Minimum Guarantees: Target temp. is not changed before AwayTime has passed

Success Guarantees: The AwayTemp target is set
The room achieves the required AwayTemp
The temp is then maintained at the required temp (see UC maintain temp)

Primary actor: User

Support actors:
Precondition: One user is in the room and the room has been occupied for at least InsideTime

Trigger:

Main success scenario:
1. The users exits the room[2]
2. After AwayTime, HCS set the temperature to AwayTemp
3. HCS maintains the temperature (see UC Maintain Temp)

Extensions:

Use case: **Maintain Temp**
Level: Sub-function

Scope: Room

[1] A presence sensor is the interface in this case
[2] A presence sensor is the interface in this case

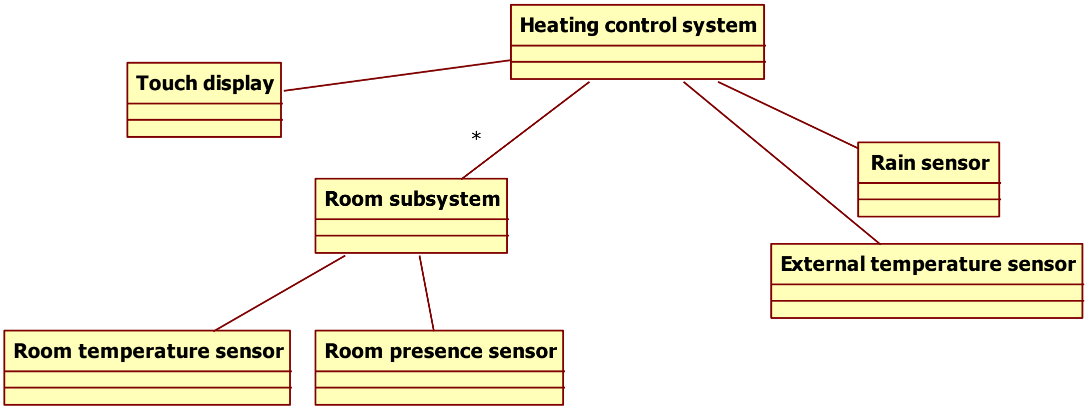
Intention in context:	Maintain a target temperature in the room
Stakeholder's concerns:	user: have the room maintained at the required temp.
Minimum Guarantees:	
Success Guarantees:	The room temperature is maintained at the target temperature +- tolerance
Primary actor:	
Support actors:	
Precondition:	A target temperature has been set
Trigger:	
Main success scenario:	1. HCS activate heating in the room 2. As soon as the room temperature achieves the target temperature HCS deactivate heating Steps 1,2 are repeated until a new target temperature is set.
Extensions:	1,2a: Room temperature raises above target temperature (+tolerance) AND windows is closed AND outside temperature is lower than target temperature: HCS opens window and use case continues 1,2b: HCS has previously opened window AND room temperature achieves target temperature: HCS closes window and use case continues 1,2c: Starts raining: HCS set window into rain safe position

Use case:	**Open window**
Level:	User-goal
Scope:	Room
Intention in context:	The user opens a windo and the HCS must set temperature to AwayTemp
Stakeholder's concerns:	
Minimum Guarantees:	
Success Guarantees:	The AwayTemp target is set The temp is then maintained at the required temp (see UC maintain temp)
Primary actor:	User
Support actors:	
Precondition:	Window is closed
Trigger:	
Main success scenario:	4. The users opens the window[3] 5. Immediately, HCS set the temperature to AwayTemp 6. HCS maintains the temperature (see UC Maintain Temp)
Extensions:	

[3] A window open sensor is the interface in this case

Part 4   System design

This UML class diagrams shows the main components of the system, as defined by the system designers. Remark that the class 'Heating control system' corresponds to the bubble 'Heating control system' in the context diagram. See the context diagram for interfaces of the system with the external world.
A key choice is on having one interaction point between the user and the system. (Another, more expensive, option is to have an interaction point in each room).
Another choice is to have only one computer (as above, another option is to have a computer in each room).
Another choice is wired connections (instead of wireless).



Components

Heating control system	Dedicated computer with touch screen, Cpu, rom, ram, AD converters, DA converters; type of cpu and size of memory to be defined, as compromise between performance and cost At least 12 AD converters Runs Android OS.
Room presence sensor	Infrared sensor, battery powered, analog output (3V)
Rain sensor	Self powered (rechargeable battery and solar cell), analog output (0V dry, 10V rain)
External temperature sensor, room temperature sensor	Self powered (rechargeable battery and solar cell), analog output (0V, 3.5V inverse prop to temperature)

Internal Connections

From	to	type
Temperature sensor (external, room)	Computer (AD converter)	Bipolar electrical wire
Rain sensor	Computer	Bipolar electrical wire
Room presence sensor	Computer	Bipolar electrical wire

External connections
See section 3.1 for connection to be managed with external devices.

Part 5 Software requirements

The software will be written in Java (to run on Android platform)

Set up and configuration

FR-ID	SW- R ID	Description
FR 1	SW1	The user shall be able to set *PresenceTemp* in °C
FR 2	SW2	The user shall be able to set *AwayTemp* in °C
FR 3	SW3	The user shall be able to set a time span *HeatingTime* in minutes
FR 4	SW4	The user shall be able to set a time span *AwayTime* in minutes
FR 5	SW5	The user shall be able to set a time span *InsideTime* in minutes
	SW31	The system should automatically recognize the number of rooms and configure accordingly
	SW32	The system should perform a self test (verify availability and functionality of each sensor and actuator connected)

Control logic

FR-ID	SW- R ID	Description
FR6	SW6	The temperature of FR 1 shall be maintained when there is someone in the room and the *InsideTime* from FR 5 has elapsed
FR7	SW7	The temperature of FR 2 should be maintained in the room when the room is empty and *AwayTime* from FR 4 has elapsed
FR 17	SW8	The *HeatingTime* from FR 3 must be at least 5 minutes longer than the *InsideTime* from FR 5
FR 18	SW9	For each quantity from FR 1, FR 2, FR 3, FR 4, FR 5 default values shall exist.
FR 19	SW10	If a window gets opened the system maintains the room temperature at *AwayTemp*, whether or not there is someone in the room.
FR 20	SW11	If it starts raining with the window open, the system will set the window in a rain secure position.
FR 21	SW12	When the temperature of a room with someone inside exceeds that from FR 1 and the outer temperature is lower than the room temperature, the window in this room gets tilted
FR17	SW13	The temperature of the boiler shall be set to the maximum required by any heater, not to waste energy

GUI (on touch screen, see system design)

FR-ID	SW- R ID	Description
	SW14	Display parameters (from SW1 SW5)
	SW15	Read parameters (from SW1 SW5)
	SW16	Show status of each room (current temperature, window position, heater status, person inside or not)
	SW17	Show status of house (number of rooms, rain, external temperature, boiler)

Devices - external

FR-ID	SW- R ID	Description
	SW18	Read status external thermometer
	SW19	Read rain sensor
	SW20	Read status hot water pump
	SW21	Set hot water pump
	SW22	Read boiler thermostat
	SW23	Set boiler thermostat

		Test boiler

Devices, in room

FR-ID	SW- R ID	Description
	SW24	Read status window in room
	SW25	Control (tilt) window in room
	SW26	Control (open close) window in room
	SW27	Read status presence sensor in room
	SW28	Read status room temperature sensor
	SW29	Read status room thermostat
	SW30	Set room thermostat