Design of Embedded Systems

ESSTA, Energy Saving Smart-home distributed Temperature control Application, Requirements

Falzone Giovanni

jointly M.Sc Embedded Computing Systems

Sant'Anna School of Advanced Studies

University of Pisa

June 9, 2019

Contents

1 Introduction

The purpose of this project is to realize a smart-home application to control the heating system of a building based on the temperature of each room, in order to minimize the consumption of the building each room apply an energy saving function reducing the desired temperature when it is not needed. The system is composed by two differ modules

- central unit module
- room module

1.1 Central Unit

The *Central Unit* has the role of coordinator that retrieve the status of each room and computes the average values for the building.

1.1.1 Graphical user interface

Using a graphical User Interface the module represents the average values of the building and the values for each room, the graphical User Interface is composed by:

- *Main page* to represent the overview of the building
- *Room page* to represent the status of each room
- *Settings page* to set the desired temperature

Whenever the *Main page* is selected the module shall represent the average values among all the rooms for *Temperature*, *Humidity* and *Usage*.
Whenever the *Main page* is selected the module shall represent the *Energy Saving* if at least one room is set to **Energy Saving mode**.
Whenever the *Main page* is selected the module shall represent the *Warning* if at least one room is set to **crashed**.
Whenever the *Main page* is selected the module shall allow the user to move in the *Settings page*, next and previous *Room page*.

Whenever the *Settings page* is selected the module shall represent the *Desired Temperature* and shall allow the user to increase or deacrease it by a factor of 0.5 C°in the range of 15 C°and 30 C°.
Whenever the *Settings page* is selected the module shall represent the average values among all the rooms for *Humidity* and *Usage*.
Whenever the *Settings page* is selected the module shall represent the *Energy Saving* if at least one room is set to **Energy Saving mode**.
Whenever the *Settings page* is selected the module shall represent the *Warning* if at least one room is set to **crashed**.
Whenever the *Settings page* is selected the module shall allow the user to move in the *Main page*.

Whenever the *Room page* is selected the module shall represent the average values among all the rooms for *Temperature*, *Humidity* and *Usage*.
Whenever the *Room page* is selected the module shall represent the *Energy Saving* if at least one room is set to *Energy Saving mode*.
Whenever the *Room page* is selected the module shall represent the *Warning* if at least one room is set to **crashed**.
Whenever the *Room page* is selected the module shall allow the user to move in the *Main page*, *Settings page*, next and previous *Room page*.

The graphical user interface shall represent the following information as reported in the table 1.

Information	**Format**
Temperature	C°
Humidity	%
Usage	%
Energy Saving	boolean
Warning	boolean

Table 1: Display Information

1.1.2 Communication

Whenever a *Room Request message* is sent and the *Room Status message* is not received within 20s the module shall mark the room as **crashed**. The module shall send the *Room Request message* for each room at least every 30s.

1.2 Room Module

The purpose of this module is to control the temperature of the room acting on a valve in order to reach and maintain the *GoalTemperature*.

1.2.1 Energy Saving mode

In order to minimize the consumption the module keep tracks of the presence of motion inside the room using a motion sensor.

If a motion is detected in the last 30s, the module shall set the *GoalTemperature* to the one set by the user,otherwise the module shall set the *GoalTemperature* to:

$$GoalTemperature = DesiredTemperature - EnergySavingTemperatureOffset \quad (1)$$

Whenever the module is in *Energy Saving mode* it shall notify it through the *Interface*.

1.2.2 Valve control

In order to control the heating of the room the valve is moved to different positions based on the temperature error (*ActualTemperature* - *GoalThemperature*)

as in the following table, whenever one of these rules is valid the module shall move the valve in the correspondent position described in the third column of the table 2 as percentage of maximum flow.

rule	**valve position**	**Flow in %**
$error < -HIGH$	OPEN_POSITION	100
$error \in [-HIGH, -APPROCHING)$	HIGH_POSITION	75
$error \in [-APPROCHING, +APPROCHING]$	MIDDLE_POSITION	50
$error \in (APPROCHING, HIGH]$	LOW_POSITION	25
error $>$ HIGH	CLOSED_POSITION	0

Table 2

Whenever the valve is in *OPEN_POSITION* or *CLOSED_POSITION* the module shall check the position and set **Valve Error** if it is not valid.

In the following table3 are reported the temperature thresholds:

HIGH	2 C°
APPROCHING	1 C°

Table 3

1.2.3 Communication

Whenever a *Room Request message* is not received within 60s the module shall send the *Room Status Message* and set the **Communication error**.

1.2.4 Errors

Whenever one of the errors are set, the module shall notify it through the *Interface*. In the following table 4 are reported the possible errors.

Valve error
Communication error
Sensor error

Table 4

2 User Requirements

2.1 Central Unit

2.1.1. Graphical User Interface on Central Unit module

2.1.1.1. Main Page

2.1.1.1.1. Whenever the *Main page* is selected the module shall represent the average values among all the rooms for *Temperature*, *Humidity* and *Usage*.

2.1.1.1.2. Whenever the *Main page* is selected the module shall represent the *Energy Saving* if at least one room is set to **Energy Saving mode**.

2.1.1.1.3. Whenever the *Main page* is selected the module shall represent the *Warning* if at least one room is set to **crashed**.

2.1.1.1.4. Whenever the *Main page* is selected the module shall allow the user to move in the *Settings page*, next and previous *Room page*.

2.1.1.2. Settings Page

2.1.1.2.1. Whenever the *Settings page* is selected the module shall represent the *Desired Temperature* and shall allow the user to increase or deacrease it by a factor of 0.5 C°in the range of 15 C°and 30 C°.

2.1.1.2.2. Whenever the *Settings page* is selected the module shall represent the average values among all the rooms for *Humidity* and *Usage*.

2.1.1.2.3. Whenever the *Settings page* is selected the module shall represent the *Energy Saving* if at least one room is set to **Energy Saving mode**.

2.1.1.2.4. Whenever the *Settings page* is selected the module shall represent the *Warning* if at least one room is set to **crashed**.

2.1.1.2.5. Whenever the *Settings page* is selected the module shall allow the user to move in the *Main page*.

2.1.1.3. Room Page

2.1.1.3.1. Whenever the *Room page* is selected the module shall represent the average values among all the rooms for *Temperature*, *Humidity* and *Usage*.

2.1.1.3.2. Whenever the *Room page* is selected the module shall represent the *Energy Saving* if at least one room is set to **Energy Saving mode**.

2.1.1.3.3. Whenever the *Room page* is selected the module shall represent the *Warning* if at least one room is set to **crashed**.

2.1.1.3.4. Whenever the *Room page* is selected the module shall allow the user to move in the *Main page*, *Settings page*, next and previous *Room page*.

2.2 Room

2.2.1. Energy Saving mode

2.2.1.1. Whenever a motion is detected in the last 30s the module shall move in **Enery Saving** mode

3 Functional requirements

3.1 Central Unit

3.1.1. Communication

3.1.1.1. The *Central Unit* shall send a *Room Request message* polling among the rooms.

3.1.1.2. The *Room Request message* shall include the Id of the room and the *Desired Temperature* in Celsius°.

3.1.1.3. The *Room Status message* shall include the Id of the room, the *Energy Saving mode* one if active zero otherwise, the *Temperature* in Celsius°, the *Humidity* in % and the *Valve position* in %.

3.1.1.4. Whenever a *Room Status message* is corrupted or dosen't arrive within 5s from the sent of *Room Request message*, the same *Room Request message* shall be resend until 3 times before marking the room as **crashed**.

3.2 Room

3.2.1. Initialization

3.2.1.1. During the initialization phase the module shall check the valve moving it from the DAFULT_CLOSED_POSITION to the DAFULT_OPEN_POSITION

3.2.1.2. During the initialization phase the module shall set the CLOSED_POSITION, LOW_POSITION, MIDDLE_POSITION, HIGH_POSITION, OPEN_POSITION

3.2.1.3. During the initialization phase the module shall check the temperature sensor until a correct value is received

3.2.1.4. During the initialization phase the module shall check the humidity sensor until a correct value is received

3.2.2. Energy Saving mode

3.2.2.1. Whenever the module is in energy saving mode the ENERGY_SAVING_LED shall be turn on

3.2.3. Valve Control

3.2.3.1. The module shall change the position of the valve every VALVE_PERIOD seconds

3.2.3.2. The valve shall be in OPEN_POSITION whenever the difference between the actual temperature and the desired temperature is below -COLD_THRESHOLD C°

3.2.3.3. The valve shall be in HIGH_POSITION whenever the difference between the actual temperature and the desired temperature is greater then -COLD_THRESHOLD C°and below -APPROACHING_THRESHOLD C°

3.2.3.4. The valve shall be in MIDDLE_POSITION whenever the difference between the actual temperature and the desired temperature is greater or equal then -APPROACHING_THRESHOLD C°and below or equal then APPROACHING_THRESHOLD C°

3.2.3.5. The valve shall be in LOW_POSITION whenever the difference between the actual temperature and the desired temperature is greater then APPROACHING_THRESHOLD C°and below or equal then HOT_THRESHOLD C°

3.2.3.6. The valve shall be in CLOSED_POSITION whenever the difference between the actual temperature and the desired temperature is greater then HOT_TEMP C°

3.2.4. **Communication**

3.2.4.1.

3.2.5. **Errors**

3.2.5.1. If the **communication error** or **valve error** or **sensor error** is set the module shall turn on the ERROR_LED.

4 SySML Functional model

In the picture 1 is reported the functional Block Definition Diagram that describes the composition of the system, composed by one Central Unit and up to eight Rooms, the two modules are connected via two FlowPort as shown in 2. The Central Unit send a *RoomRequest* message composed as follows:

parameter	**type**	**[Min,Max]**
Id	Natural	[1,8]
DesiredTemperature	Float	[15.00, 30.00]

Table 5: Room Request variables

The Room module send a *RoomStatus* message composed as follow:

parameter	**type**	**[Min,Max]**
Id	Integer	[1,8]
Eco	Boolean	[0, 1]
Temperature	Float	[15.00, 30.00]
Humidity	Float	[0.00, 100.00]
Valve	Integer	[0, 100]

Table 6: Room Status variables

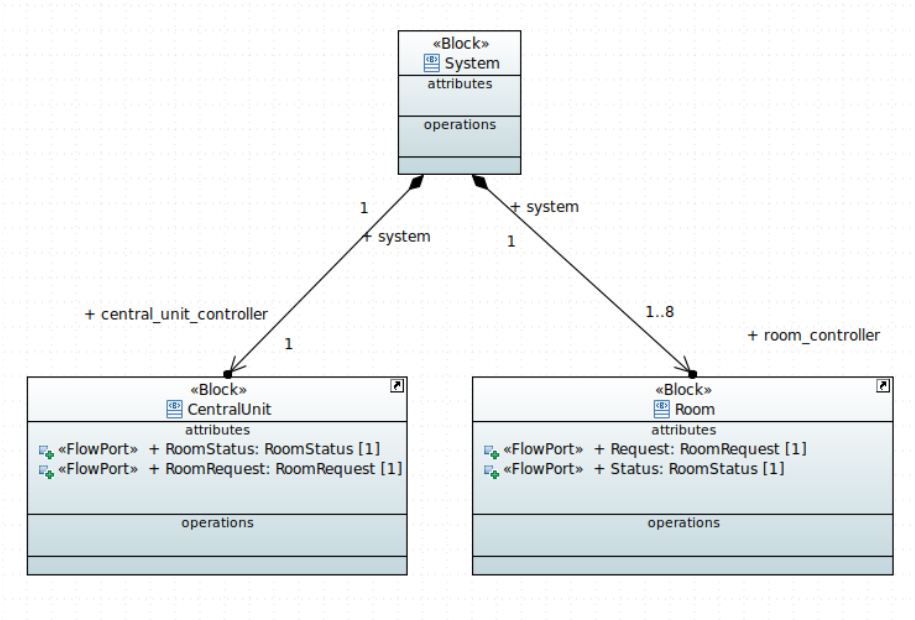

Figure 1: System Components

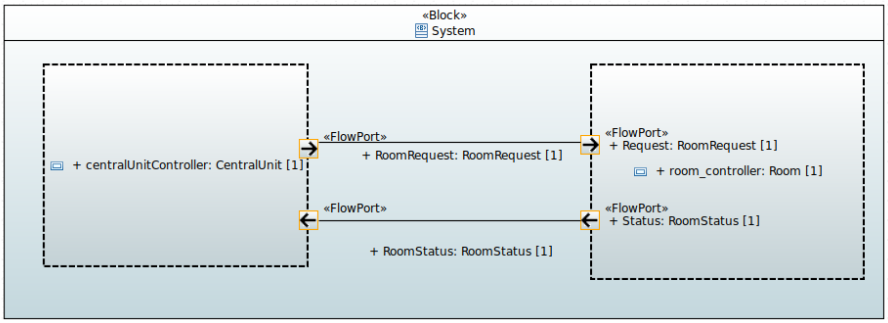

Figure 2: System Internals

4.1 Central Unit

The *Central Unit* is composed by two modules, the *RoomsManager* and the *UserInterfaceManager*. The *RoomsManager* implements the functionalities related to the status of each room. The *UserInterfaceManager* that implements the functionalities related to represent the status of the system. The two components exchange data as shown in 4.

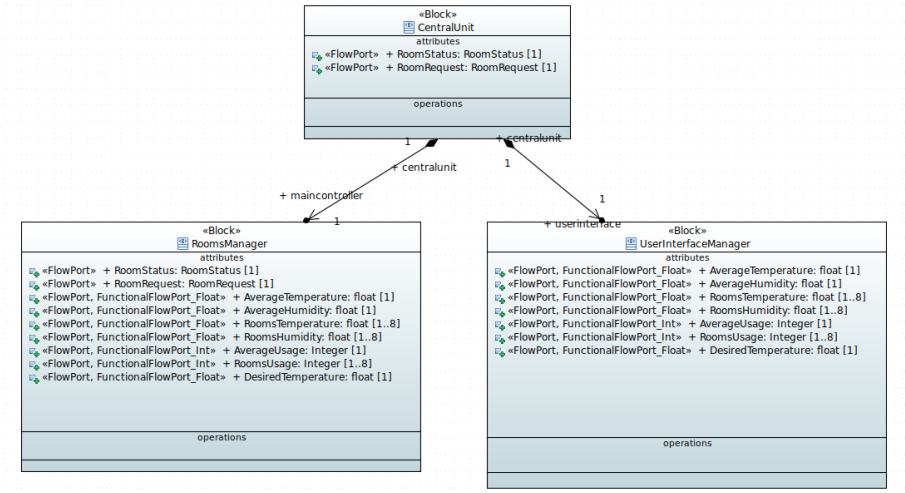

Figure 3: Central Unit components

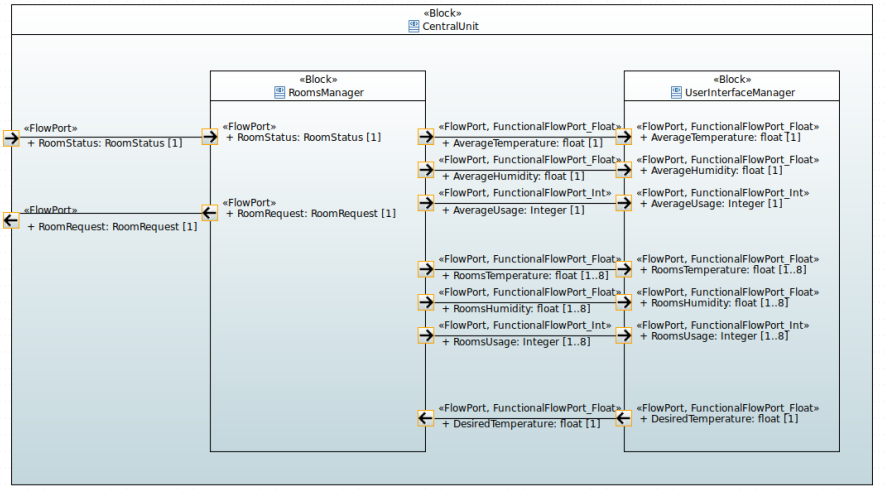

Figure 4: Central Unit internals

4.2 Room module

The main component of this module is the *MainController* composed by different functions as shown in 6.

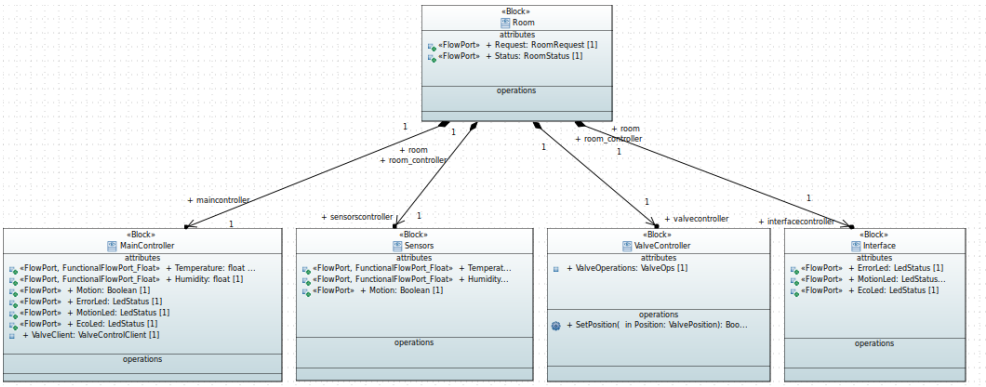

Figure 5: Room Components

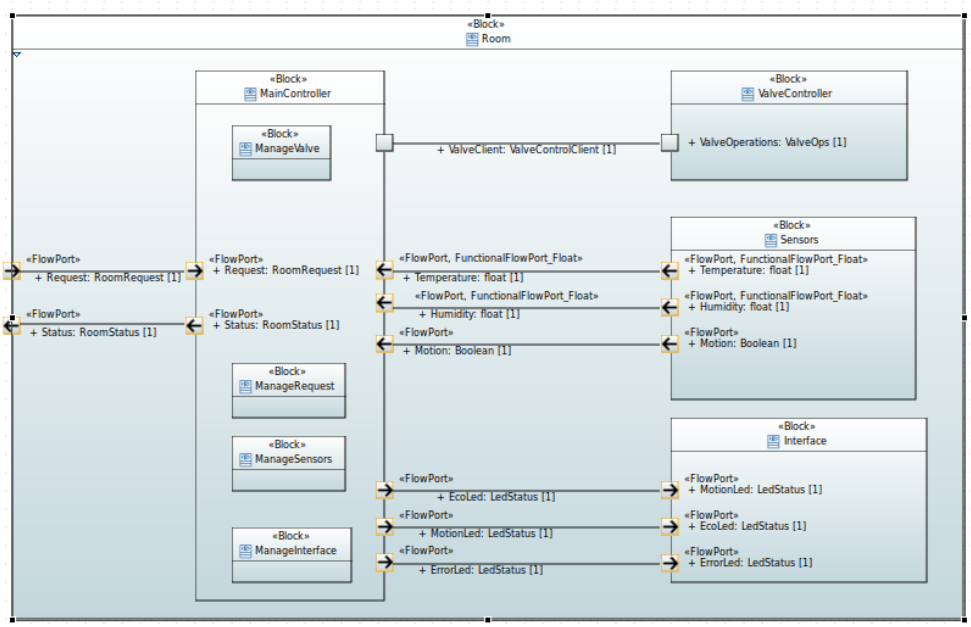

Figure 6: Room Internals

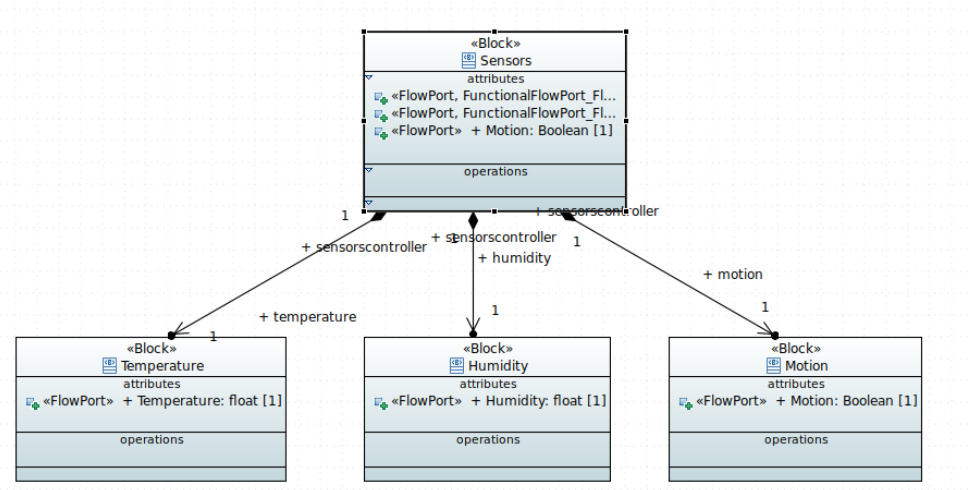

Figure 7: Room sensors components

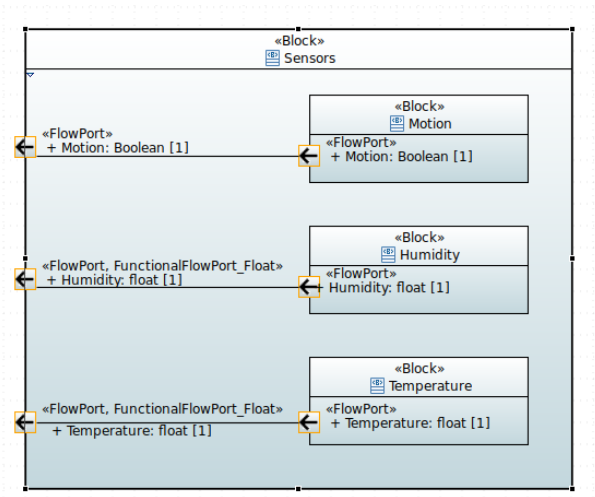

Figure 8: Room sensors internals

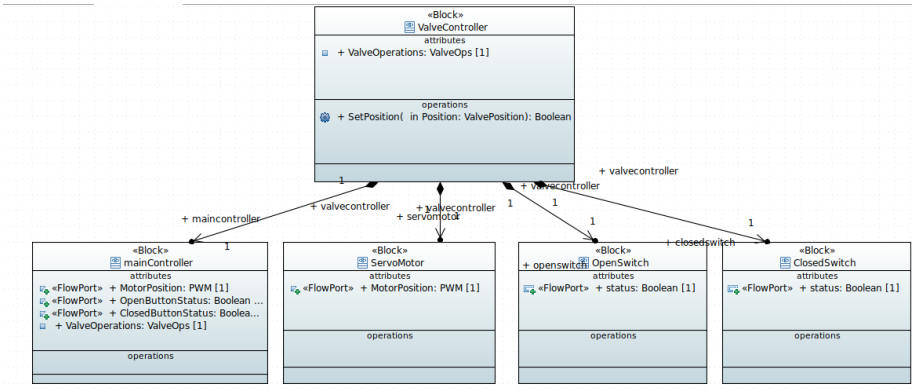

Figure 9: Valve Controller components

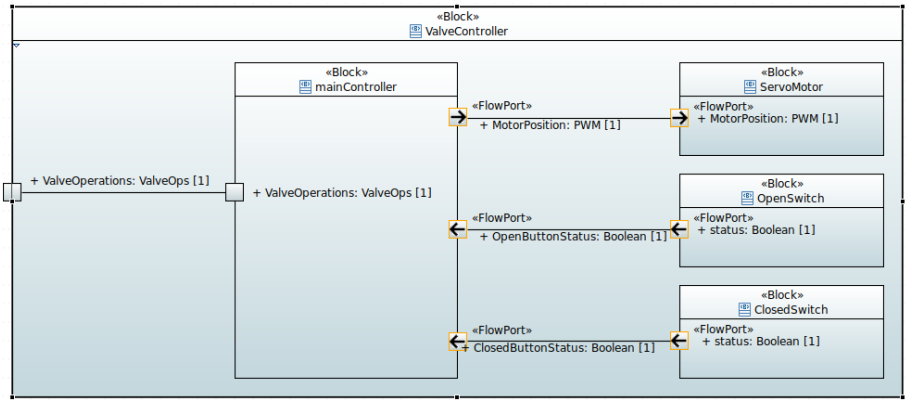

Figure 10: Valve Controller internals