Design of Embedded Systems

ESSTA, Energy Saving Smart-home distributed Temperature control Application

Falzone Giovanni

jointly M.Sc Embedded Computing Systems

Sant'Anna School of Advanced Studies

University of Pisa

May 23, 2019

Contents

1 Introduction

The goal of this project is to realize a smart-home application to control the temperature of different room in order to minimize the consumption of the entire building.

The system is composed by two differ modules

- A central unit
- A room module

2 Room Module

The aim of this module is to control the temperature of the room and act on the valve in order to achieve the desired temperature set by the user through the *Central Unit* but minimizing the consumption reducing the desired temperature when the room is not used.
The module is composed by:

- Temperature sensor
- Humidity sensor
- Motion sensor
- Valve actuator
- Wireless communication module
- Error led
- Eco mode led
- Motion detection led

2.1 Temperature control

The module adjust the temperature of the room acting on a valve. The valve is set to different positions based on the temperature error (difference between the actual temperature and the desired themperature):

- if the temperature error is below then -COLD_THRESHOLD the valve is moved in OPEN_POSITION
- if the temperature error is below then -WARM_THRESHOLD the valve is moved in HIGH_POSITION
- if the temperature error is between a [-APPROCHING_THRESHOLD, APPROCHING_THRESHOLD] the valve is moved in HALF_POSITION
- if the temperature error is greater then WARM_THRESHOLD the valve is moved in LOW_POSITION
- if the temperature error is greater then HOT_THRESHOLD the valve is moved in CLOSED_POSITION

2.2 Energy Saving mode

In order to minimize the consumpion the module keep tracks oF the presence of motion inside the room using a motion sensor. If a predefined number of motion is detected in a time slot, the module set the desired temperature to the one set by the user. If there is the number of motions counted is greater then a predefined thresholf the module set the desired temperature to a value equal to the user desired temperature minus a default value ENERGY_SAVING_TEMPERATURE_DIFFERENCE.

2.3 User Requirements

2.3.1. Whenever a motion is not detected the module shall move in Eco mode

2.3.2. Whenever difference between the actual temperature and the desired temperature is included in the range [-APPROACHING_THRESHOLD, APPROACHING_THRESHOLD] the valve shall be in HALF_POSITION

2.4 Functional requirements

2.4.1. Initialization

2.4.1.1. Whenever the module is turn on it shall send an initialization message to the *Central Unit* and wait for the response

2.4.1.2. During the initialization phase the module shall continue blinking the ERROR_LED

2.4.1.3. During the initialization phase the module shall check the valve moving it from the CLOSED_POSITION to the OPEN_POSITION

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

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

2.4.2. Communication

2.4.2.1. The module shall move in COMMUNICATION_ERROR status if does not receive the check message from the *Central Unit* within 1 minute

2.4.2.2. The module shall send its status to the *Central Unit* every 10 seconds

2.4.2.3. The module shall send its status in conformance with JSON format

2.4.2.3.1. The status message shall include its ID in the status message

2.4.2.3.2. The status message shall include the Eco mode status

2.4.2.3.3. The status message shall include its sensors list

2.4.2.3.4. The status message shall include its actuators list

2.4.2.3.5. The status message shall include the name of every sensor and actuator

2.4.2.3.6. The status message shall include the format for every numerical value

2.4.2.4. Whenever a check message from the *Central Unit* is not received whitin 1 minute the module shall go in COMMUNICATION_ERROR

2.4.3. Valve management

2.4.3.1. The module shall change the position of the valve every 30 seconds

2.4.3.2. The valve shall be in one of the allowed positions

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

2.4.3.2.2. 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°

2.4.3.2.3. The valve shall be in HALF_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°

2.4.3.2.4. 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°

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

2.4.4. **Sensors management**

2.4.4.1. The module shall update the actual temperature every 10 seconds

2.4.4.2. Whenever the actual temperature is below 15 C°or greater then 40 C°the module shall go in temperature error state

2.4.4.3. The module shall update the actual humidity every 10 seconds

2.4.4.4. The module shall update the presence of motion every 5 seconds

2.4.4.5. Whenever a motion is detected the module shall notify it turning on the MOTION_LED

2.4.4.6. Whenever a motion is detected the module shall increase a MOTION_COUNTER value

2.4.4.7. Whenever a motion is not detected and the MOTION_COUNTER is greater then zero the module shall decrease the MOTION_COUNTER

2.4.5. **Energy Saving management**

2.4.5.1. Whenever the MOTION_COUNTER reaches the MOTION_THRESOLD value the module shall move in normal mode

2.4.5.2. Whenever the MOTION_COUNTER is below the MOTION_THRESOLD value the module shall move in Eco mode

2.4.5.3. Whenever the module is in Eco mode the module shall notify it turning on the ECO_MODE_LED

2.4.5.4. Whenever the module is in Eco mode the module shall set the desired temperature with the difference between the desired temperature and the ENERGY_SAVING_TEMPERATURE_DIFFERENCE

2.4.6. **Error handling**

2.4.6.1. Whenever an error case is achieved the module shall notify the precense of errors turning on the ERROR_LED

2.4.6.2. Whenever an error case is achieved the module shall continue sending a status message with the error field set to 1 to the *Central Unit*

3 Central Unit

4 Hardware implementation

In this chapter it is shown the hardware implementation of each module, the hardware is chosen to be compliant with the requirements of the project.

4.1 Room module

The room module is implemented on a ATM328P

- Atmega328P
- Xbee Pro S1
- DHT22 Temperature and Humidity sensor
- PIR motion sensor
- Red led
- Green led
- Yellow led

4.2 Central Unit moduke

- STM32F407VG DISCOVERY
- 3"5 LCD board Touchscreen
- Relè switch
- Fan

4.3 Network configuration

PAN ID:

Device	**Role**	**ATID**	**ATDL**	**ATMY**
Central Unit	Coordinator	1	87837	787
Room 1	End device	2	78	5415
Room 2	End device	3	78	5415

Appendices

References

[1]