System Requirements Specification

ECSE 421 - Embedded Systems

Group 1

Sary Amatouri
Carlos Asmat
Yevgeniy Goldenberg
François Grondin
Stephen Hopkins
David Lopez
Ritvik Mudur
Maram Qouqa

February 12, 2008

Table of contents

Page

Introduction

Global warming has become a major issue during the past few years. Many individuals and corporations are now trying to decrease their power consumption. As a result, more and more people are interested in the idea of a “green” house – an environmentally friendly structure that would significantly reduce power consumption by making use of renewable energy and efficiently managing the available resources.

This project is about designing a real-time embedded system that would manage the power consumption in an apartment occupied by two students. The apartment consists of two bedrooms, a kitchen, a living room and a bathroom. The apartment is located in Montreal and the system is designed to operate during winter, when the average temperature outside is -10°C. The temperature in each room depends on the temperature outside and varies according to laws of heat conduction. For simplicity, it is assumed that the temperature of each room does not affect that of the other rooms. Each room is equipped with different devices and its own heating unit. The house is connected to two different sources of energy: solar cells connected to rechargeable batteries and the Hydro-Quebec network.

Both students have a fixed schedule: they go to school from Monday to Friday during the day and they sleep from 11PM to 7AM.

The system must minimize the amount of power bought from Hydro-Quebec. Solar cells provide power only during the day and at a random rate while batteries have a maximum charging capacity. The system will minimize power consumption by turning off unused devices and scheduling some tasks that are performed over a long period of time. Also, as a part of the system, a scheduler is configured to cut power consumption to its minimum during the night, especially on certain electrical appliances no longer being used, and at times when residents of the apartment are not home.

The document is divided into five sections. The inputs and outputs of the system are listed and described. Then the functional requirements are provided. In this section, the internal working of the system such as data manipulation and processing is described. Moreover, two instances of an error that might occur in the system are defined and the default behavior of the system for these cases is described. Finally, the technical requirements are described. This explains what hardware and software will be use to implement the system.

Inputs

There are different switches and sensors in each room that allow the user to interact with the system. Each room has different buttons and sensors which allow the user to interact with the system. Moreover, when a device can operate independently of the system, there is a status signal which indicates if the device is currently on or off. The local button for each independent device is not considered as part of the embedded system.

The following list shows the switches, sensors and status signals along with each device (with their respective power consumption). Also, since this system must be aware of the time, there is an input for an external time base. This is also convenient for testing purposes, since the time can be accelerated or slowed down according to the time base signal frequency.

Bedroom 1

Room Light Button	Released: 0 / Pushed: 1
Bed Lamp 1 Button	Released: 0 / Pushed: 1
Bed Lamp 2 Button	Released: 0 / Pushed: 1
Room temperature	4-bit sensor: 0b0000 = 13°C, 0b1111 = 28°C
Motion detector	No motion: 0 / Motion: 1

Bedroom 2

Room Light Button	Released: 0 / Pushed: 1
Bed Lamp 1 Button	Released: 0 / Pushed: 1
Bed Lamp 2 Button	Released: 0 / Pushed: 1
Room temperature	4-bit sensor: 0b0000 = 13°C, 0b1111 = 28°C
Motion detector	No motion: 0 / Motion: 1

Living-room

Television status	OFF: 0 / ON: 1
Room Light 1,2,3 Button	Released: 0 / Pushed: 1
Room temperature	4-bit sensor: 0b0000 = 13°C, 0b1111 = 28°C
Motion detector	No motion: 0 / Motion: 1
Luminosity detector	4-bit sensor: 0b0000 = no light, 0b1111 = max light

Bathroom

Room Light Button	Released: 0 / Pushed: 1
Fan status	OFF: 0 / ON: 1
Hair blow dryer status	OFF: 0 / ON: 1
Room temperature	4-bit sensor: 0b0000 = 13°C, 0b1111 = 28°C

Motion detector	No motion: 0 / Motion: 1
Kitchen	
Oven status	OFF: 0 / ON: 1
Refrigerator status	OFF: 0 / ON: 1
Extra accessory status	OFF: 0 / ON: 1
Microwave oven status	OFF: 0 / ON: 1
Room Light 1,2,3 Button	Released: 0 / Pushed: 1
Room temperature	4-bit sensor: 0b0000 = 13°C, 0b1111 = 28°C
Motion detector	No motion: 0 / Motion: 1
Luminosity detector	4-bit sensor: 0b0000 = no light, 0b1111 = max light
Power sources	
Power produced by solar cell	8-bit sensor: 0b00000000 = 0 W, 0b11111111 = 3000 W
Energy stored in the batteries	8-bit sensor: 0b00000000 = 0 kWh, 0b11111111 = 2.2 kWh
Electricity failure signal	Normal: 0 / Failure: 1
Clock	
Real-time clock	1-bit signal: each pulse corresponds to 1 second

Total inputs: 67 bits

Outputs

Each room has its own devices and a heating unit. Some devices are strictly controlled by the embedded system while others can be controlled independently. When a device is only controlled by the system, it is turned on (1) or off (0) according to the status of its control signal. However, when a device is not only controlled by the system, the control signal is different. Each independent device has its own local button. When the control signal is in the "allow to be ON or OFF" mode, only the local button controls whether or not the device is ON. On the other hand, if the control signal is in the "force to be OFF" mode, the device cannot be turned on even if the local button is pressed. Finally, if a device is ON (the control signal is at 0 and the button has been pressed), it is turned off if the control signal changes from 0 to 1. If the control signal comes back to 0 afterwards, the device stays off unless the local button is pressed again.

The following list shows the control signals and the device (with their respective power consumption) for each room. Also, a signal controls the batteries' action (charging or discharging).

Bedroom 1

Room Light 1	60 W	0: OFF / 1: ON
Bed Lamp 1	40 W	0: OFF / 1: ON
Bed Lamp 2	40 W	0: OFF / 1: ON
Heating Unit	1000 W	0: OFF / 1: ON

Bedroom 2

Room Light 1	60 W	0: OFF / 1: ON
Bed Lamp 1	40 W	0: OFF / 1: ON
Bed Lamp 2	40 W	0: OFF / 1: ON
Heating Unit	1000 W	0: OFF / 1: ON

Living-room

Television	200 W	0: Allow to be ON or OFF / 0 to 1: Turn OFF / 1: Force to be OFF
Room Light 1	60 W	0: OFF / 1: ON
Room Light 2	60 W	0: OFF / 1: ON
Room Light 3	60 W	0: OFF / 1: ON
Heating Unit	1000 W	0: OFF / 1: ON

Bathroom

Room Light	60 W	0: OFF / 1: ON
Fan	500 W	0: Allow to be ON or OFF / 0 to 1: Turn OFF / 1: Force to be OFF
Hair blow dryer	1000 W	0: Allow to be ON or OFF / 0 to 1: Turn OFF / 1: Force to be OFF
Heating Unit	500 W	0: OFF / 1: ON

Kitchen

Oven	2500 W	0: Allow to be ON or OFF / 0 to 1: Turn OFF / 1: Force to be OFF
Refrigerator	600 W	0: OFF / 1: ON
Extra accessory	1000 W	0: Allow to be ON or OFF / 0 to 1: Turn OFF / 1: Force to be OFF
Microwave oven	1100 W	0: Allow to be ON or OFF / 0 to 1: Turn OFF / 1: Force to be OFF
Room Light 1	60 W	0: OFF / 1: ON
Room Light 2	60 W	0: OFF / 1: ON
Room Light 3	60 W	0: OFF / 1: ON
Heating Unit	1000 W	0: OFF / 1: ON

Batteries

Charging/Discharging		Batteries are charged: 0 / Batteries feed power to the house: 1

Board

LED		LED is OFF: 0 / LED is ON: 1

Total Outputs: 27 bits

Functional requirements

1. Room Lights (Bedroom 1, Bedroom 2, Living-room, Kitchen, and Bathroom)

When the room lights are initially turned off and a pulse signal (from the room light button) shorter than 2 seconds is sent, the room lights are turned on.

When the room lights are initially turned on and a pulse signal (from the room light button) shorter than 2 seconds is sent, the room lights are turned off.

When the button is pressed, the signal sent is a logical 1. When the button is released, the signal sent is a logical 0. The duration of the pulse therefore corresponds to the time the button was kept pressed.

The lights cannot be turned on or off independently of the house management system. Moreover, when a light is turned on, it can also be turned off by the house management system in some specific cases. These cases are described in the next requirements.

Inputs: Room light button signal
Outputs: Room light control signal

2. Room Lights (Living-room and Kitchen)

When the room lights are initially turned off and a pulse signal (from the room light button) longer than 2 seconds is sent, one, two or three room lights are turned on. One light is turned on. The output signal of the luminosity detector is checked. If the luminosity obtained is not sufficient, then the second light is turned on. Again, if the luminosity level achieved is not sufficient, the third light is turned on. At this point, nothing more can be done if the luminosity level is still not reached.

The luminosity level can be configured by the students via the GUI interface. Moreover, if the measured level changes over time (due to sunlight), the system will turn on or off lights to adjust luminosity. However, this feedback is performed only each 5 minutes. The reason for this is to avoid having a light turning on and off quickly. This could happen if the measured luminosity level requires a given number of lights but is on the edge of turning on one extra light. Small changes in the perceived luminosity would cause the extra light to be turned on or off. This would be annoying for the person in the room.

Inputs: Room light button signal
Outputs: Room lights 1, 2 and 3 control signals

3. **Bed Lamps**

When a bed lamp is initially turned off and a pulse signal (from the bed lamp button) is sent, the bed lamp is turned on.

When a bed lamp is initially turned on and a pulse signal (from the bed lamp button) is sent, the bed lamp is turned off.

When the button is pressed, the signal sent is a logical 1. When the button is released, the signal sent is a logical 0. The duration of the pulse therefore corresponds to the time the button was kept pressed.

The lights cannot be turned on or off independently of the house management system. Moreover, when a light is turned on, it can also be turned off by the house management system in some specific cases. These cases are described in the next requirements.

Inputs: Bed lamps 1 and 2 button signals
Outputs: Bed lamps 1 and 2 control signals

4. **Motion detection**

When someone moves in a room, the motion detector signal is set to 1. If there is no motion in a given room for 5 minutes (meaning the signal was kept to 0), the following devices in this room are turned off:

Bedrooms

Room light
Bed lamps 1 and 2

Living-room

Room lights 1, 2 and 3
Television

Bathroom

Room light
Hair blow dryer
Fan

Kitchen

Room lights 1, 2 and 3

Once a device has been turned off, it stays off even if there is motion in the room. As mentioned before, some devices are independent of the house management system. The television, hair blow dryer and the fan can be turned back on by the user. This is so because the system only turns off the devices but does not force them to stay off afterwards. The room lights and bed lamps can be turned back on by sending a request signal with the push buttons (as explained previously).

Inputs: Motion detectors signals
Outputs: Control signals for all devices mentioned above

5. Refrigerator

The refrigerator has to be operating at least 30 minutes during an hour in order to preserve food. The refrigerator is controlled only by the system. For this reason, the user cannot turn on or off the refrigerator manually.

The refrigerator could be turned off more than 30 minutes in the case no one opens the door. However, there is no sensor to detect that the door is opened or not. For this reason, 30 minutes is considered a conservative choice to ensure that the food is preserved even if the door is opened many times during a given period.

All refrigerators have their own thermostat to ensure that the temperature inside is kept to a desired level. When this is the case, the compressor is not working and thus no power is consumed by the device. For this reason, when the refrigerator is said to be turned on, this means the device is allowed to regulate its temperature. However, if there is no need for such regulation, the refrigerator will not consume power. This means that turning on the refrigerator for 30 minutes does not necessary implies power consumption during the whole interval of time. However, for simplicity, it will be assumed that the refrigerator consumes power all the time when it is turned on.

Inputs: Real-time clock signal
Outputs: Refrigerator control signal

6. Power consumption - excess

When power consumption exceeds the power produced by the solar cell, some actions are taken in order to reduce as much as possible the use of power from the Hydro-Quebec network.

- The refrigerator is turned off (but for no more than 30 minutes as explained above)
- The heating system allows the temperature to drop by 3°C in the room where there is no motion
- If charged, batteries are discharged to feed power to the house

The power consumption level is constantly checked by the system. This level is obtained since the system receives status signal from devices and directly control all the lights. Since the power consumption of each device is known, the system knows the total power consumption. When this level exceeds the power produced, the refrigerator is first turned off. This is done because the solar cell power production varies during the day according to sunlight. Since the refrigerator does not need to be powered all the time, this allows reducing power consumption when needed.

If this is not enough to bring back the consumption level below the produced level, then the temperature is allowed to drop in rooms where there is no motion. However, if there is motion in a room, the temperature must be set back to the desired level.

If lowering the temperature in empty rooms is still not enough, then the batteries are discharged in order to provide more power. Finally, if more power is still needed, then it is provided from the Hydro-Quebec network.

Inputs: Real-time clock signal
Power produced by solar cell signal
Energy stored in the batteries signal
Room temperature sensors signals
Outputs: Refrigerator control signal
Heating units control signals
Charging/Discharging control signal

7. Power consumption - shortage

When the power produced by the solar cell exceeds the power consumed by the devices, the battery is charged. Obviously the battery cannot be discharged and charged at the same time. If the battery was being discharged, the system must ensure that the power produced is still greater than the power being consumed when the battery is not discharging. If not, the battery continues discharging.

The battery can be charged at a rate of 1100 Watts. It can be discharged at the same rate. The battery can store up to 2.2 kWh. However, it is never discharged under 100 Wh. If this limit is reached, the system stops discharging the battery and waits for an excess of power produced in order to charge it. When the battery is fully charged, the system stops charging the battery. If the battery is charged and the power produced exceeds the power consumed, the refrigerator is turned on. This allows storing energy since the refrigerator will be able to be turned off for 30 minutes later on. If the power produced still exceeds the power consumed, then the extra power is lost.

Inputs: Power produced by solar cell signal
Energy stored in the batteries signal
Outputs: Charging/Discharging control signal
Refrigerator control signal

8. Temperature

The students can configure the desired temperature for each room. This is done from the GUI interface on the PC. The configuration is saved on the hard drive of the PC. Therefore, if the PC is restarted, the configuration is not lost.

The system regulates the temperature for all rooms in order to keep it at the desired level. When the temperature is above the desired level in a given room, the heating unit of this room is turned off. When the temperature is below the desired level, the heating unit is turned on.

Since the temperature varies slowly, temperature checks are not time critical and therefore are performed each minute.

Inputs: Room temperature sensors signals
Outputs: Heating units control signals

9. Scheduler

The students can configure the time they are not home. This configuration is done from the GUI interface on the PC. Once again, the configuration is saved on the hard drive of the PC. When students are not home, the temperature is allowed to drop by 5°C in all rooms under the desired level.

The same temperature regulation is performed except that the level is lower. This is done because heat transfer is greater when the temperature difference is greater. Since the temperature outside is maintained at -10°C, lowering the indoor temperature reduces this difference. The heat transfer is thus reduced and less power is needed for heating.

Since the temperature varies slowly, the heating system restarts one hour before the students come back home. This ensures that the usual desired level is reached when they get home.

Inputs: Real-time clock signal
Room temperature sensors signals
Outputs: Heating units control signals

10. Night power saving mode

The students can activate night power saving mode. This is done from the GUI interface on the PC. As usual, the configuration is saved on the hard drive of the PC. The temperature is then allowed to drop by 5 °C in the kitchen, the living room and the bathroom from 11PM to 6AM. Again, this is done in order to save power regarding heat transfer as explained above. Temperature is increased to the desired level between 6AM and 7AM. This ensures that the desired temperature level is reached when students wake up. During the night, only the bedrooms are kept at the desired temperature.

Inputs: Real-time clock
Room temperature sensors signals
Outputs: Heating units control signals

Error Cases

Case 1: Electricity failure

The electricity failure signal has priority over all other signals. This signal is set when no more power is available from the Hydro-Quebec network. The power available is now limited but the house temperature must be maintained at a safe level. Moreover, the lights must keep working in case this happens at night. For these reasons, all devices are turned off and forced to stay off in all rooms except lights, heating units and the refrigerator. The power produced by the batteries and the solar cell is used to power up these devices. The usual power consumption strategies described in functional requirements (sections 6 and 7) are still applied. The only difference is that the Hydro-Quebec network can no longer supply power when the power consumed exceeds the power produced.

If the battery is half discharged, then all lights are closed in order to save energy for heating. If the battery is completely discharged and no power is produced by the solar cells, then the house can no longer maintain its temperature. Students then have to leave the house for their own safety.

For simplicity, it is assumed that the house management system has its own backup power source in case of electricity failure.

<u>Inputs</u>: Electricity failure signal
Power produced by solar cell signal
Energy stored in the batteries signal
Room temperature sensors signals
<u>Outputs</u>: All output signals (except the LED on the board)

Case 2: PC failure

The system is made of a board with a logic circuit and a PC. The PC runs under Windows and may crash. Moreover, the serial link between the PC and the board can be broken or disconnected. For this reason, there is a protection system on the board to ensure that basic functions are still handled without the PC. If this happens, there is no more power consumption optimization. Heating units are turned on or off in order to maintain the same temperature in all rooms at 21°C. Moreover, all devices that are independent are no longer forced to be OFF. All lights can be turned ON or OFF. There is also a LED on the board that blinks to indicate that the PC is no longer responding.

In the case of a PC crash, the PC must be restarted. While this is done, the students can still make full use of the difference devices within the house.

However, if the PC is not responding and there is an electricity failure, the board will not handle this specific case. The PC needs to be restarted before the system can handle this case.

<u>Inputs</u>: None (PC failure is detected from the connection between the PC and the board)
<u>Outputs</u>: All output signals (including the LED on the board)

Technical requirements

The house management system is made of a PC and a board. This PC is where the entire decision making takes place, and it also displays the systems outputs after processing. In order to make simulation possible, a house simulator system will also be designed with another PC and another board. This one will be used as the front-end of the system, where all inputs can be provided and all outputs can be visualised.

This system is designed for interacting with the different house inputs and outputs as shown in figure 1.

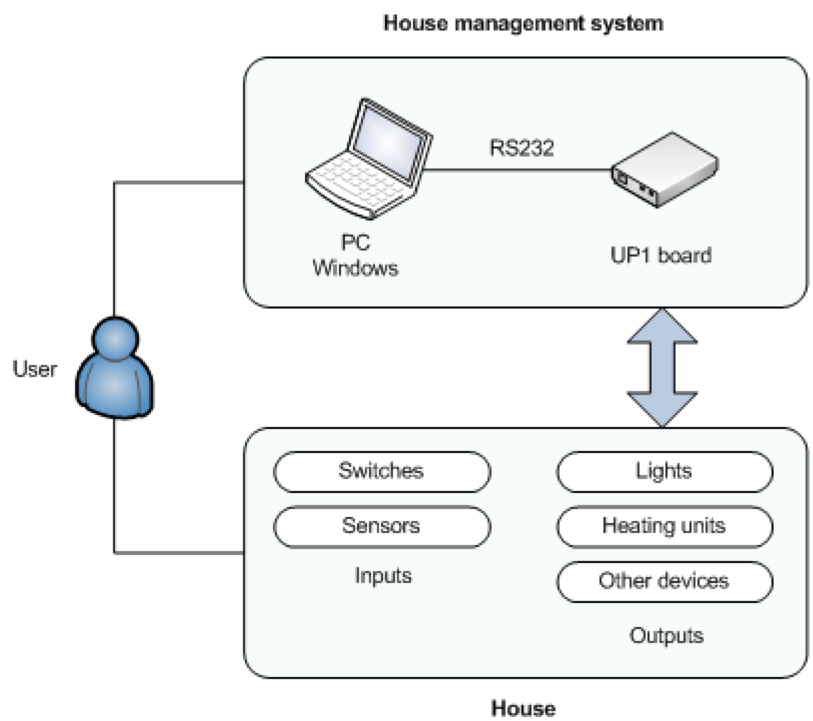

Figure 1: Overview of the management system and the house

However, as mentioned, this is not practical for simulation purpose. For this reason, the house inputs and outputs are replaced by a simulator as shown on figure 2. The inputs can then be produced by a GUI on the PC and the outputs are displayed by the same GUI.

•

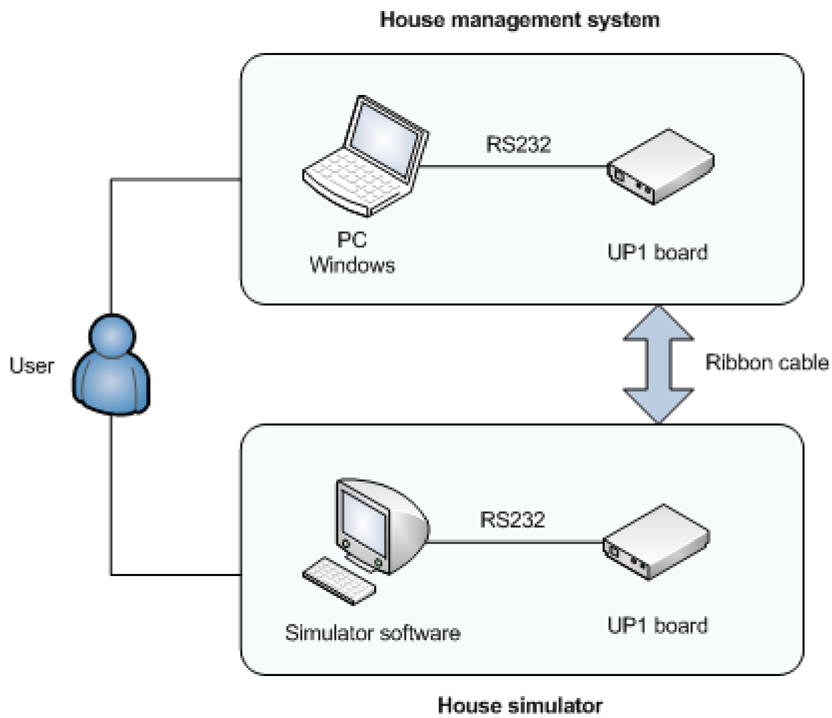

Figure 2: Overview of the management system and simulator

House management system

PC

- **Processor**
  - o An x86 CPU running Windows XP will be responsible for all processing and decision making.
  - o Based on the inputs and current power consumption, the PC will control devices and adjust temperatures accordingly.
- **Front End**
  - o The Front-end of the system will be in the form of a GUI running on a Windows XP platform.
  - o The interface will permit students to change settings regarding scheduling, desired temperature and night power saving mode
  - o The reason for this is that everyone has access to a PC and they are easy to program.
- **Software**
  - o As for the programming language, C programming language will be used to aid in the communication between the PC and the board. The reason behind this choice is that C is a fast, powerful, flexible and efficient language and is best used for applications that need to be small, fast and efficient. C++ may also be used for the GUI interface since it is a higher level language and is object-oriented.

Board

- Since the system has 94 Inputs/Outputs, a FLEX chip which has 126 I/O pins is used. The system will use an Altera UP1 board to communicate with the processor.
- The logical circuit on this board will be implemented with VHDL

Communication

- The FLEX chip will communicate with a PC using an asynchronous RS-232 protocol. A serial-to-USB cable is required for this communication for PCs that do not have a serial input.

<u>House simulator system</u>

Inputs are provided to system through the first PC, via the use of a mouse and keyboard to enter required data. This permits users to predefine settings on the levels of temperature, lighting or power consumption which are to be regulated and maintained at all times.

Any output generated by the system due to processing (which occurs as a result of the input entered) and any detected changes in the temperature or power consumption will be displayed on the screen of the second PC as opposed to having both inputs and outputs displayed on the same PC. This is done in order to illustrate and ensure distinguishable communication between the output screen and controller.

PC

- **Processor**
  - An x86 CPU running Windows XP will be responsible for generating simulation inputs and reading house management system outputs.
- **Front End**
  - The Front-end of the system will be in the form of a GUI running on a Windows XP platform.
  - The interface will permit the user to change the inputs sent to the house management system in real-time.
  - The interface will permit non-technical users to enter inputs with the use of simple mouse clicks and alter the conditions of the rooms in the house as well as to change settings for the maximum power consumption.
  - The user will also be able to run a preprogrammed vector of different test cases
- **Software**
  - As for the programming language, C sharp or BASIC programming languages will be used to aid in the communication between the PC and the board and to allow the user to interact with the simulation software. The reason behind this choice is that these languages are object oriented and a user-friendly GUI is already available.

Boards

- For testing purposes, a second FLEX chip will be used to simulate the inputs and outputs.
- A ribbon cable will be used to connect the 94 I/O pins across the 2 boards.

Communication

- The FLEX chip will communicate with a PC using an asynchronous RS-232 protocol. A serial-to-USB cable is required for this communication for PCs that do not have a serial input.