1 Introduction

The first modern air conditioning system was developed in 1902 by a young electrical engineer named Willis Haviland Carrier.

Air conditioning units came a long way in this hundred years that have passed. They are smaller, more efficient, more reliable and more intelligent.

The purpose of this project is to design an embedded system responsible for controlling an air conditioning unit.

2 Objective

Design an embedded system to control an air conditioning (AC) unit. That embedded system should be intelligent enough to detect patterns in user usage and thus efficiently allow the cooling of the room where the system is installed.

3 Introduction to Air Conditioners

Air conditioners use refrigeration to chill indoor air, taking advantage of when a liquid converts to a gas (in a process called phase conversion), it absorbs heat. Air conditioners exploit this feature of phase conversion by forcing special chemical compounds to evaporate and condense over and over again in a closed system of coils.

Air conditioners also contain fans that move warm interior air over these cold, refrigerant-filled coils. When hot air flows over the cold, low-pressure evaporator coils, the refrigerant inside absorbs heat as it changes from a liquid to a gaseous state. To keep cooling efficiently, the air conditioner has to convert the refrigerant gas back to a liquid again. To do that, a compressor puts the gas under high pressure, a process that creates unwanted heat. All the extra heat created by compressing the gas is then evacuated to the outdoors with the help of a second set of coils called condenser coils, and a second fan. As the gas cools, it changes back to a liquid, and the process starts all over again.
The cold side of an air conditioner contains the evaporator and a fan that blows air over the chilled coils and into the room. The hot side contains the compressor, condenser and another fan to vent hot air coming off the compressed refrigerant to the outdoors. In between the two sets of coils, there’s an expansion valve. It regulates the amount of compressed liquid refrigerant moving into the evaporator. Once in the evaporator, the refrigerant experiences a pressure drop, expands and changes back into a gas. The compressor is actually a large electric pump that pressurizes the refrigerant gas as part of the process of turning it back into a liquid.

In Figure 1, there’s a basic configuration of an AC unit. In this configuration, inside the room stays the cold coils and the expansion valve, and outside the room stays the compressor and the hot coils. The fans are not represented here since in this project we are only interested in controlling the compressor. In Figure 1, it’s also represented the main modules that control the system.
4 System specifications

The embedded system must be designed according to the functional specifications and non-functional specifications.

Functional specification

The embedded system must work according to the specifications below.

The AC unit must be able to measure the temperature of the room. Depending on the measured temperature and the desired temperature it should control the amount of cold air that is produced. The actual temperature must be visible in a display in the AC unit in Celsius units.

The AC unit should be intelligent to detect patterns in the user usage and act accordingly. Thus, the AC unit will have a motion detector to detect activity in the room where it is installed and thus decide if it should be working or not. When in intelligent mode a light should be turned on in the AC unit.

Powering on/off and manually controlling the temperature will be done using a remote control. The remote control should work using infrared communication. The remote control always overrides the intelligent mode.

Non functional specifications

The display to show the temperature should be installed in the AC unit and be comprised of only two digits.

The remote control should be as simple as possible with the least number of keys.

The system must be designed according to the best practices studied in Software for Embedded Systems.

5 Simulator

As the first step in the design of the AC controller, an application that simulate the system must be designed in a host machine. This simulation must model as accurately as possible the physical implementation. The application will consist of multiple modules:

- inside unit;
- remote command;
- outside unit;
- debugging.

The application must be implemented on top of a Windows or Linux platform using C++, C# or Java language.
Inside Unit

The inside unit will consist of one (or more) process(es) running in the host machine. In this process each individual module of the inside unit will consist of an object in the software language chosen. Thus, there will be at least the following modules/objects:

- CPU
- memory
- motion sensor
- temperature sensor
- display that shows the current temperature as measured by the temperature sensor
- led to show the intelligent mode
- IR module to communicate with the remote control
- module to communicate with the outside unit using serial data communication

The inputs and outputs of these modules/objects must be clearly defined.

The processes must adhere to the specifications described in Section 4.

Remote command

The remote command will be simulated as a process running in the host machine. In this process each individual module of the remote control will consist of an object in the software language chosen. Thus, there will be at least the following modules/objects:

- IR module to communicate with the inside unit
- keypad to power on/off the AC unit and to set the temperature

The process must adhere to the specifications described in Section 4.

Outside Unit

The outside unit will be simulated as a process running in the host machine. In this process each individual module of the outside unit will consist of an object in the software language chosen. Thus, there will be at least the following modules/objects:

- compressor motor controller
- module to communicate with the outside unit using serial data communication

The process must adhere to the specifications described in Section 4.
Communication between modules

Communication between the remote control and the inside unit and between the inside unit and the outside unit must be simulated according to the protocols used.

- **IR communication**
  To simulate the IR communication a file will be used. When the remote control sends a bit that bit should be written into a file overriding the previous bit. When the inside unit wants to read a bit it just reads the value in the file.
  The IR protocol used should be chosen by the students and how it was simulated must be described in the final report.

- **Serial communication**
  To simulate the serial communication a file for each bit will be used. When the inside unit sends a bit that bit should be written into a file overriding the previous bit. When the outside unit wants to read a bit it just reads the value in the file.
  The serial communication protocol to be used is the RS-485. How this protocol was simulated must be described in the final report.

Debugging

In order to assert the correct functionality of the simulator some debugging functionality must be present on the previously described modules. Therefore, while the system is running, it must be possible to:

- see the value in the display
- see the light in the led
- set the value of the room temperature
- know the temperature that was set by the remote control
- set and read the value of the motion detector
- have a notion of the speed of the compressor motor
- “press” the different buttons of the remote control
- know the values that were sent by the remote control
- know the values that were sent by the inside unit to the outside unit

Testing

Taking into account the system specifications (See Section 4), prepare a procedure to test the simulator of the system. This procedure should be the one to be used in the project presentation (see Section 9).

In the presentation it should be possible to interact with the remote control and to see the intelligent mode functioning.
6 Hardware modules

The necessary hardware modules must be chosen according to the specifications of the system. Search the web and choose possible hardware modules to implement the main required functions. Whenever possible, relevant technical specification of the modules should be obtained (e.g. type of input/output, power consumption).

The software implementation of the simulators of the different modules must match the specifications of the hardware modules (e.g. type of input/output).

7 What should be delivered

The submission of the project will consist in uploading a zip file through the fenix system. The zip file must consist of a report and the source code of the simulator.

Report (no more than 10 pages)

The report should be in pdf format and must include at least the following items with the rationale for them when appropriate:

- General description of the system and enumeration of its features;
- System architecture (hardware and software architecture of the system and the hardware/software interaction);
- User manual of the simulator and the description of its different modules;
- Main problems found in implementing this system and solutions for those problems (when found);
- List of specifications not met and reasons for it.

Source code of simulator

All the source code of the simulator must be in a zip file. The zip file must also contain a text file describing all the steps to run a simulation.

8 Project milestones

March, 5 - Project presentation in Tagus (Lab class).
March, 7 - Project presentation in Alameda (Lab class).
Next weeks - Project development.
May, 4, at 17h00 - Deadline for submission of project in fenix.
Next two weeks - Project presentations and evaluations. Evaluations of “competitors” projects. Sending of evaluation reports.
9 Project assessment

After the submission of the project, groups will be associated in clusters. The evaluation will occur on the lab classes and will have the following phases:

1. Project presentation – Presentation of the project, including the demonstration of the simulator and the presentation of the main architecture, design, and features of the system. (Oral presentation, no slides required.)

2. Assessment of the projects of the other groups in the cluster – Students must participate in all the presentations of their cluster to clarify any issues related with the projects of the other groups of their cluster.

3. Evaluation report – Short report in pdf (one page max.) about all the projects of the cluster – including the own project of the group – stating what the group considers to be the main features and weaknesses of the projects of the cluster. These reports must be submitted by e-mail no more than 48 hours after the end of the presentations of the cluster.

4. Review meeting – Review evaluation meeting with each group and the teacher. Therefore each group must participate in two evaluation sessions – Project presentation (all the groups of the cluster), and review meeting (group by group).