AL AKHAWAYN UNIVERSITY IN IFRANE

SCHOOL OF SCIENCE & ENGINEERING

SMART HOME ENERGY MANAGEMENT SYSTEM

An Arduino Based Wireless System

Capstone Design

Spring 2015

By: Sofia Ait bellah

Supervisor: Dr. Ahmed Khallaayoun

Co-Supervisor: Mr. Rachid Lghoul
Acknowledgment:

I would like first to express my sincere gratitude to Dr. Khallaayoun for his unique and excellent supervision. He dedicated his time and invested his energy to guide us and give us his support. He has always given us strong motivation and helpful advice. Throughout the semester, he organized regular meetings to share information, get feedback and find together as a team a solution to any problems we could face in the project.

I am also grateful to Mr. Racheid Lghoul who was available for us in the research lab to help and bring new and interesting ideas. He has always encouraged us to give our best for the success of the project.

Moreover, I never would have made it without the help of my teammates Fatima Zahra Barnicha, Mouad Sahil and Bakr Sikal who didn’t hesitate to share their expertise and come to our help whenever needed. They gave me the opportunity to live a unique experience, learn how to do research, work as a team in a productive and a very enjoyable environment.

Last but not least, I want to take this opportunity to thank anyone who directly or indirectly has left his or her hand in this project including my parents who gave me their continuous support throughout my four years at Al Akhawayn University.
# Table of Content

List of Figures

List of tables

Acknowledgment: ................................................................. Erreur ! Signet non défini.

Abstract ......................................................................................... ix

Glossary .......................................................................................... x

1  Introduction .................................................................................. 1
   1.1  The Moroccan Context .......................................................... 1
   1.2  Previous Work ........................................................................ 1
   1.3  STEEPLE Analysis ................................................................. 3
   1.4  Problem Statement ............................................................... 5
   1.5  Report Summary ..................................................................... 6

2  Literature Review .......................................................................... 7
   2.1  An Overview of Smart Grids .................................................. 7
      2.1.1  What is a Smart Grid? ...................................................... 7
      2.1.2  Smart Grid Structure ...................................................... 8
      2.1.3  Smart grid systems: ....................................................... 9
   2.2  Integration of the Solar Systems into the Micro grid: ............ 9
      2.2.1  Photovoltaic (PV) panels ............................................... 10
      2.2.2  Inverters and batteries .................................................. 11
      2.2.3  Other components ......................................................... 11
      2.2.4  Long term cost benefits of the renewable energy ........... 12
   2.3  Home Energy Management System (HEM) .......................... 13
      2.3.1  Arduino microcontroller ............................................... 14
      2.3.2  Zigbee Wireless communication ................................. 14

3  Home Energy Management system hardware and software requirements and specifications ....... 15
   3.1  HEM requirements and specifications ................................... 15
   3.2  Hardware Requirements ....................................................... 16
      3.2.1  Arduino Uno Microcontroller ....................................... 16
      3.2.2  Wireless Communication Network ................................ 19
      3.2.3  Sensors ................................................................. 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.4 Relay Box</td>
<td>26</td>
</tr>
<tr>
<td>3.2.5 Open Source Arduino Software</td>
<td>26</td>
</tr>
<tr>
<td>4 Home Energy Management System Design Phase</td>
<td>28</td>
</tr>
<tr>
<td>4.1 System Architecture</td>
<td>28</td>
</tr>
<tr>
<td>4.1.1 Home Appliances</td>
<td>29</td>
</tr>
<tr>
<td>4.1.2 Current Sensors and Control Relay</td>
<td>29</td>
</tr>
<tr>
<td>4.1.3 Arduino Uno</td>
<td>30</td>
</tr>
<tr>
<td>4.1.4 Wireless network communication layers:</td>
<td>30</td>
</tr>
<tr>
<td>4.1.5 Web Server</td>
<td>31</td>
</tr>
<tr>
<td>4.1.6 Database</td>
<td>31</td>
</tr>
<tr>
<td>4.1.7 Web Application and User devices</td>
<td>31</td>
</tr>
<tr>
<td>4.2 Detailed Architecture</td>
<td>32</td>
</tr>
<tr>
<td>4.2.1 Server module and gateway</td>
<td>32</td>
</tr>
<tr>
<td>4.2.2 Server Controller</td>
<td>35</td>
</tr>
<tr>
<td>5 Home Energy Management System Implementation Phase</td>
<td>37</td>
</tr>
<tr>
<td>5.1 The previously designed wireless actuation network:</td>
<td>37</td>
</tr>
<tr>
<td>5.1.1 Components of the previous wireless actuation network:</td>
<td>37</td>
</tr>
<tr>
<td>5.1.2 Xbee series 2 configuration</td>
<td>38</td>
</tr>
<tr>
<td>5.1.3 Preparing the base station</td>
<td>38</td>
</tr>
<tr>
<td>5.1.4 Preparing the end nodes</td>
<td>41</td>
</tr>
<tr>
<td>5.2 The newly designed wireless actuation network:</td>
<td>43</td>
</tr>
<tr>
<td>5.2.1 Phase 1: design of the new base station</td>
<td>43</td>
</tr>
<tr>
<td>5.2.2 Phase 1: Xbee communication between the newly designed gateway and smartplugs</td>
<td>44</td>
</tr>
<tr>
<td>5.2.3 Phase 3: connect to database</td>
<td>47</td>
</tr>
<tr>
<td>5.2.4 Results</td>
<td>51</td>
</tr>
<tr>
<td>5.2.5 Constraints of the implementation phase:</td>
<td>53</td>
</tr>
<tr>
<td>6 Cost and Energy Demand Analysis</td>
<td>55</td>
</tr>
<tr>
<td>6.1 Cost Analysis and comparison of different HEMS</td>
<td>55</td>
</tr>
<tr>
<td>6.2 Energy demand and peak analysis:</td>
<td>60</td>
</tr>
<tr>
<td>6.2.1 Loading disaggregation</td>
<td>60</td>
</tr>
<tr>
<td>6.2.2 Batmax Loading Algorithms</td>
<td>61</td>
</tr>
<tr>
<td>7 Future Work</td>
<td>64</td>
</tr>
</tbody>
</table>
8  Conclusion.........................................................................................................................65
9  References ....................................................................................................................67
10  Appendices.....................................................................................................................70
List of Figures

Figure 1.4: Steeple analysis of HEMS

Figure 3.2.1.1: Atmega 328P microcontroller chip

Figure 3.2.1.2 Components of Arduino Uno

Figure 3.2.1.1: X-Bee Radio series2

Figure 3.2.1.1: X-Bee Shield

Figure 3.2.2.3: Arduino Ethernet shield

Figure 3.2.3.1: Temperature Sensor SEN 23292P

Figure 3.2.3.2: Graph showing the variation of resistance with respect to temperature

Figure 3.2.3.2: Light Sensor SEN 11302P

Figure 3.2.3.3: Split-Core Current transformer sensor YHDC STC 013-030 [18]

Figure 3.2.4: Control Relay Box

Figure 3.2.5 Arduino IDE structure

Figure 4.1: System architecture

Figure 4.2.1.2.1: Picture of the designed gateway

Figure 4.2.1.2.2: Detailed architecture

Figure 5.1.3.1: The implemented control function of the gateway of the HEMS

Figure 5.1.3.2: The implemented (sensor_data) function of the gateway

Figure 5.1.4.1: The implemented control function of the smartplug

Figure 5.2.2.1: Implemented Control and sensing functions of one to one wireless actuation network

Figure 5.2.3.1.2: Sketch for internet connection
Figure 5.2.3.1: Components of a web application

Figure 5.2.3.2: Implemented control function through the web application

Figure 5.2.3.3: Implemented sensing function through the web application

Figure 5.2.4.1: Results of the two ways communication of the smartplugs

Figure 5.2.4.2: Results of the two ways communication of the gateway

Figure 5.2.5.1: Displayed configuration failure message

Figure 6.1.1: Comparison of prices of different HEMS

Figure 6.2.2.1: BatMax scheduling algorithm

Figure 6.2.2.2: Demand-Side energy management
List of Tables

Table 2.1.1: Comparison between existent grids and smart grids

Table 2.2.4.1: Solar panel prices in Morocco

Table 2.2.4.2: Inverters prices in Morocco

Table 3.2.1: Characteristics of the Arduino Uno and its specifications

Table 3.2.1.1: characteristics and requirements of X-Bee radio series 2 module

Table 3.2.3.1: Temperature sensor requirements

Table 3.2.3.2: Light sensor requirements

Table 3.2.3.3: Characteristics of the current sensor YHDC STC 013-030

Table 4.2.1.1: Different Components of the gateway

Table 4.2.2 Different components of a Smartplug

Table 6.1.1: Approximation of prices of the Arduino Yun based HEMS components

Table 6.1.2: Approximation of prices of the Arduino Uno based HEMS components

Table 6.1.3: Approximation of the total price of the Arduino Uno based HEMS with different numbers of smartplugs

Table 6.1.4: Approximation of prices of the Raspberry Pie based HEMS components
Abstract

Electricity has become vital and important in our everyday life. Indeed, the demand and consumption of energy has been highly increasing these last years. This trend can be explained by many factors like the increasing population rate and the use of the technology and more electronic devices over the years. The electric energy is facing many issues mainly the increasing energy demand, the increasing costs and electricity bills, the dependence on the importation of the energetic needs especially the fossil fuels and the protection of the environment. To cope with these challenges, electricity should be efficiently consumed and produced. To reach this goal, a research team from Al Akhawayn University is working on the implementation of a home energy management system to be able to control and monitor the energy consumed and also to optimize the use of energy by integrating renewable energies and implementing the loading algorithms. The previously implemented home energy management system is not reliable and does not fulfill all the important requirements. The objective of this project is to implement a new HEMS to reach our energy optimization objective. This home energy management system is composed of multiple smart plugs and a gateway. The smart plugs are connected to home appliances to get data about the current, temperature and light. They are composed of Arduino microcontrollers, sensors and a relay box, and they constitute the end nodes of the wireless network. The gateway on the other hand, composed of an Arduino Uno, an Xbee shield and Ethernet shield, gathers data from all the smart plugs and stores it in the database, which can be accessed using any computer, Android, iOs via internet.
## Glossary

<table>
<thead>
<tr>
<th><strong>Gateway</strong></th>
<th>A base station that connects the smart plugs to the internet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HEMS</strong></td>
<td>Home Energy Management System</td>
</tr>
<tr>
<td><strong>Smartplugs</strong></td>
<td>Device equipped with sensors and a relay box connected to the home appliances</td>
</tr>
<tr>
<td><strong>X-bee</strong></td>
<td>Digi International chip used for Zigbee and other wireless communication protocols</td>
</tr>
<tr>
<td><strong>API Mode</strong></td>
<td>Application Programming Interface Mode</td>
</tr>
<tr>
<td><strong>PAN address</strong></td>
<td>Personnal Area Network address</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 The Moroccan Context

The increase of the energy consumption in Morocco raises the issue of the dependence on the fossil fuels, causes the lack of reliability of the electricity production systems and negatively affects the environment. Morocco is a country that doesn’t have oil resources and relies on importations. In 2009, the bill of electricity was estimated to reach approximately MAD62 billion, and the demand was expected to be multiplied by four in 2030. [11]

Therefore, the government has raised its investments on the energy distribution network. The development of the microgrids in Morocco will allow an accurate monitoring of the energy consumption and adapt the electricity production using renewable energies accordingly. This is achieved thanks to the possible two ways communication of the home energy management system composed of multiple smartplugs equipped with sensors that collect information and a gateway, the central element and communicator of information.

1.2 Previous Work

We will carry on a project that many capstone students have worked on semesters ago.

First, “Abdelkrim Adyel” and “Soukayna Mouatadid” studied the “Load Profiling in the Moroccan Residential Sector”. They have gathered data related to the consumption profile of a sample of households through surveys. A passive solar heating procedure has been simulated for the sample targeting a better and more efficient use of energy. In parallel to this work, two other capstone students “Imane L’hadi” and “Sarah Lahtani” have focused on implementing a “Home
Energy Management Android Application”. This latter consists on making Mathlab simulations to model the energy consumption of the home appliances and also to achieve an efficient generation of renewable electricity based on the continuous changing demand.

In this same context, another work was done to emphasize on the “Challenges of implementing the Smart Grid in developing countries” focusing on the consequences and hardships that may encounter Morocco while moving to the new technology of smartgrids.

The recent work done was an “Arduino Based Management System” realized by “Soukaina brangui” and “Ismail El Hamzaoui”. The goal was to implement an android application (HEM) that to implement a “home energy management system” to optimize the electricity consumption of each appliance at the residential level and maximize the use of renewable energies using smart plugs and a gateway. The smart plugs composed of an Arduino, a relay and current, power consumption, light and temperature sensors are directly connected to the home appliances. Once data is retrieved, the information is transmitted via the zigbee wireless standard to the gateway consisting of an Arduino Yun and a Xbee shield. As a result, the transmission of information is not reliable and is done in only one way from the smart plugs to the gateway.

Therefore, the purpose of this current work is to improve the already existing energy management system to meet some of the missing requirements which are the following:

- Minimize the cost of the system
- Improve the wireless communication between the smart plugs and the gateway
- Improve the security and privacy of the software application
1.3 STEEPLE Analysis

STEEPLE analysis is a tool that evaluates the factors that affect the environment of a project and improves a company’s decision-making. It was created in 1967 under the acronym of PEST (political economical, societal, technology), which evolved with time to SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) and finally to PEST analysis (Political, Economic, Social and Technological).

The figure 1.4 shows the STEEPLE model that stands for: societal, technology, environment, ethics, political, legal, and finally economical.
Society is facing the challenge of scarcity and the increasing energy demand. HEMS is a way to manage the electricity consumption and adapt to the changing behavior of the society mainly caused by the technological revolution.

Thanks to the creation of powerful microcontrollers and of the wireless communication platforms, electrical energy can be controlled and monitored via a home energy management system. The new technologies and innovations in the field of electronics are making the implementation of the HEMS simpler and easier.

The goal of the HEMS is to control the electricity at the residential level to decrease the consumption of energy and therefore, save the environment. Because electricity is produced from fossil fuels like coal oil, etc., an efficient use of energy is crucial to limit greenhouse effect and the emissions of pollutants.

One of the most important requirements of the HEMS is to ensure security and privacy of the user to avoid any misuse or modification of the information stored in the database.

The HEMS with renewable energies integrated in the system is one way to reduce the energetic interdependence of the country to oil producing countries and to become self-sufficient.

The Moroccan government has made many efforts to set the rules and regulations for the development of renewable energies. Indeed, the law 13-09 prohibits the injection of low voltage power on the grid. Besides, the laws 16-09 and 57-09 were launched to establish MASEN, the Moroccan Solar Agency and ADEREE that stands for Agency for the Development of Renewable Energies and Energy Efficiency. This latter is responsible for developing the policies for the renewable energies management.

The installment of a home energy management system, along with solar systems will have long term benefits. It will make the consumer save money and lower the electricity bills.

Figure 1.4: Steeple analysis of HEMS
1.4 Problem Statement

Today, the major part of the population lives in cities which consume 75% of the energy produced and cause 80% of the CO2 emissions. The challenge is then to transition toward smart grids to guarantee a sustainable energy to the future generations. Energy is running out, electricity prices are exponentially increasing, and the global warming issues are threatening the stability of our environment. All these issues are calling for urgent measures to be taken to reduce the energy consumption. To achieve this goal at the residential level, the consumer should be able to manage the power consumption of each individual appliance and make use of the renewable energies to optimize the energy usage and reduce the dependency on the grid.

The purpose is to come up with a way to reach the home energy efficiency. The usual power meters installed in houses measure only the amount of power currently used and the remaining available electricity. It does not provide details of consumption of each individual appliance and does not allow the monitoring of the devices. Moreover, in the previously implemented system, the interaction between the Arduino Yun and the XBee chip is sometimes unreliable which affects the steadiness of data and the reliability of the control via wifi. The objective of this project is to implement a new HEMS to solve all those issues and reach our energy optimization objective.
1.5 Report Summary

Section 2 of the report presents the literature review of the research. Going from the highest scale to the lowest, the first point discussed was the smart grid, its structure and different systems, and second point was the integration of solar systems into the microgrid. In this last part which is the renewable energies, information about the photovoltaic panels, inverters, batteries and the long term cost benefits were presented. Going deeper into the grid, the home energy management system HEMS was introduced.

Section 3 presents the home energy management requirements and specifications. The hardware requirements part includes the introduction of the Arduino Uno microcontroller, wireless communication network, the sensors, the relay box and the open source software.

Section 4 deals with the design of the HEMS. First, a general architecture of the whole system was presented, and then the details about both the server module and the server component were given.

How to implement the new HEMS was explained in section 5, which was followed by a cost and energy demand peak analysis in section 6.

Section 7 introduces the future work, the new ideas and functionalities to be added to the newly implemented HEMS to improve the efficiency of the system.

Finally, section 8 summarizes the main points and results of our current system and briefly mentions the achieved requirements.
2 Literature Review

2.1 An Overview of Smart Grids

2.1.1 What is a Smart Grid?

“A smart grid is a modern power grid that supports bidirectional communication between energy providers and consumers for fine-grained metering, control, and feedback”. [1]

The objective of a smart grid is to move from a one way interaction into a two way dialogue between the different parts of the electrical network to ensure an efficient, reliable and greener distribution of energy depending on the varying demand. The table 2.1.1 shows the difference between the existent grids and the smart grids. In smart grids, data can be exchanged between the electricity generation and the consumers thanks to the introduction of the new technology, computers and intelligence. [6]

Table 2.1.1: Comparison between existent grids and smart grids

<table>
<thead>
<tr>
<th>Existant Grid</th>
<th>Smart Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>One way communication</td>
<td>Two ways communication</td>
</tr>
<tr>
<td>No Interaction with the user</td>
<td>Interaction with the user</td>
</tr>
<tr>
<td>Electrical</td>
<td>Digital</td>
</tr>
<tr>
<td>Manual control</td>
<td>Self control</td>
</tr>
<tr>
<td>Power interruption and failures</td>
<td>Adaptive and reliable</td>
</tr>
<tr>
<td>Centralized generation</td>
<td>Distributed generation</td>
</tr>
<tr>
<td>Few sensors</td>
<td>Many sensors involved</td>
</tr>
</tbody>
</table>
2.1.2 Smart Grid Structure

The smart Grid network is subdivided into three distinct parts:

- The first one is the transmission system or power generation that has the supply demand control function. It consists on producing electricity using either nuclear or thermal or hydraulic power plants. Renewable energies like wind and solar power plants are also activated depending on the demand and the changing weather, and they are regulated accordingly. Electricity is very expensive at the peak times when extra energy is demanded and more power plants have to be activated to meet the demand. The smart grid is then a way to avoid the peaks and uniformly moderate the power generation over the day with the involvement of the consumers who communicate their daily demand.

- The second part is the power distribution system that regulates the electricity voltage based on the final electricity destination, which can either be residential, industrial or commercial. Low voltage is needed for home usage, more for offices, commercial building and even more for industries. The electricity is sent through transformers to increase the voltage, then goes through high-voltage transmission lines to get to substations and is finally regulated.

- The last one is the consumer. For homes, microgrids help managing the electricity consumption by controlling the appliances via android, IOs or a computer through the web. This would reduce the demand and therefore, generate less power, reduce the costs,
lower the electricity bills and save the environment. HEM that stands for home energy management systems are considered an advantage for every part of the network.

2.1.3 Smart Grid Systems:

Smart infrastructure system is about the energy transmission through the cables, the advanced wireless communication of information and the advanced information control and monitoring from the power plants to the consumers.

Smart management system is the system that takes care of well controlling and managing the energy consumption.

Smart protection system provides security, privacy and reliable electrical supply chain [2].

2.2 Integration of the Solar Systems into the Micro grid:

A microgrid consists on generating energy, monitoring and controlling the consumption and storage of electricity through an automated system called a home energy management system.

Integrating the renewable energies to the micro grid would not only allow the residents to be consumers but also local energy producers. They would then be more energetically independent, get rid of the burden of the monthly electricity bills and save the environment.
2.2.1 Photovoltaic (PV) Panels

Photovoltaic consists on converting the solar energy and radiations into electricity. Photons, small sun particles, are converted into electrons once they reach the solar panels. Cells are linked together to form photovoltaic panels which provide the electric power either for residential use or to supply a distribution network. The energy is produced using semiconducting materials that are causing the photovoltaic effect. These cells produce a direct current that then needs to be transformed into an AC current for home electricity usage.

There are three types of photovoltaic panels that could be used in microgrids, the monocrystalline solar cells, polycrystalline cells and the amorphous cells. The monocrystalline cells are made of one crystal of silicon and have the highest power to area ratio, but they are more expensive than the other panels. However, the polycrystalline panels are made of multiple crystals of silicon and have the highest performance/price ratio. They have a large lifetime that is estimated to be more than 35 years, but they are less efficient than the monocrystalline panels as they are made of less levels of silicon. They are the mostly used solar panels. The last type is the amorphous panel. They are made of different types of solar cells: Amorphous silicon (a-Si)Cadmium telluride (CdTe)Copper indium gallium selenide (CIS/CIGS) and Organic photovoltaic cells (OPC). They are lighter but have a lower performance and need a bigger surface area for an equal power generated by crystalline panels. They are used in farms rather than in houses on the top of the roof.

The panels are placed either in series or in parallel depending on the needed voltage to be supplied. The diodes are integrated in the circuit to control the flow of the current, which can make the solar panel more efficient. For instance, if a 6x10 cells panel is used, diodes can divide
it into a 2x10 cells panel and make the current flow in a way that would avoid the shaded panels to save energy. [8]

2.2.2 Inverters and Batteries

Electricity is generated by the solar panels as Direct current (DC). However, because home appliances use alternating current (AC), inverters are necessary to transfer the DC power into the home appliances. There are three types of inverters, standalone Inverters, Grid tie Inverters and Dual Inverters. The standalone inverters also called off-grid inverters transform the DC current generated from the solar panels and stored in batteries into an AC current when electricity is needed to power the home appliances. These inverters are independent from the smart grid. However, the grid tie inverters depend on the power grid. Indeed, the current generated by the solar panels is used locally for domestic usages and in case of any excess of energy is transferred to the smart grid without any battery installed for power storage. It is necessary to have an electricity meter to get credit for any electricity produced. The consumer is billed only when the energy needed exceeds the energy produced and is given to the utility company. The dual inverters combine both types and are the most efficient inverter but also the most expensive one. When energy is generated by the panels, it is stored in batteries and the excess is transmitted to the smart grid to optimize the energy consumption [9].

2.2.3 Other Components

In addition to the solar panels, a battery and an inverter, a charge controller is also needed to regulate the voltage of electricity produced to be compatible with the battery voltage.
A meter keeps track of the electricity produced. In case of any excess in energy not used, the consumer can get credit from the electricity company for the energy transferred into the grid.

Finally, other hardware elements are needed for the installment and configuration of the solar system like wires, cables, cable trays, junction box and a load contractor.

2.2.4 Long Term Cost Benefits of the Renewable Energy

The decrease of the solar panels price and the increase of the electricity price over the years make the renewable energy that is integrated into the micro grid a profitable option on the long term.

The cost of the solar systems varies according to the components chosen (The solar panels, the inverter, batteries, etc), the quality of the material, the number of kWh it produces, etc. Either all the energy is resold or only a part of the energy that has not been used is transferred to the electricity company. By getting credit from the electricity generated, the consumers will get refunded over the years, save and even earn money.

The table 2.2.4.1 and the table 2.2.4.2 show the ranges of prices of the solar system components. [7]
Table 2.2.4.1: Solar panel prices in Morocco [7]

<table>
<thead>
<tr>
<th>Solar Panel</th>
<th>Price (dhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luxor Solo 36/100 w</td>
<td>1200</td>
</tr>
<tr>
<td>AXIpower ac-250p/156-60s</td>
<td>2750</td>
</tr>
<tr>
<td>Victron polycristallin 130 Wc</td>
<td>5021.10</td>
</tr>
<tr>
<td>280 Wc victron</td>
<td>10727.10</td>
</tr>
</tbody>
</table>

Table 2.2.4.2: Inverters prices in Morocco [7]

<table>
<thead>
<tr>
<th>Inverters</th>
<th>Price (dhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sma Sunny Boy 1300tl</td>
<td>10356</td>
</tr>
<tr>
<td>Sma Sunny Boy 1600tl</td>
<td>11040</td>
</tr>
<tr>
<td>Sma Sunny Boy 2100tl</td>
<td>12660</td>
</tr>
</tbody>
</table>

2.3 Home Energy Management System (HEM)

The home energy management system is made of three major components, smart plugs, a gateway and a mobile application. These elements all together form a toolkit that makes the electricity controllable and save the energy consumption.
The smart plugs are made of a microcontroller Arduino Uno, sensors to collect data about the current, temperature and light and a relay box for remote control. Each smart plug is connected to a home appliance to get the information about every single appliance’s electricity consumption individually. The data is then sent to the gateway via Zigbee, an Xbee wireless communication protocol. Furthermore, the gateway is composed of Arduino Uno, an Xbee shield, and an Ethernet shield that collect information and send it via the wifi communication network to the application server. This latter identifies the data and stores it in the database [10].

2.3.1 Arduino Microcontroller

Arduino is an open hardware microcontroller that will be used for the smart plugs. It is compatible with the available current, temperature and light sensors and also with the ZigBee technology for the wireless communication between the smart plugs and the gateway.

2.3.2 Zigbee Wireless Communication

Compared with Wifi or Bluetooth, Zigbee is the best option for a reliable communication between the smart plugs and the gateway in the home management system. Indeed, it allows a communication that offers a larger range of communication approximately 100 feet than Bluetooth. Moreover, Zigbee is low cost, low power and easier to implement.
3 Home Energy Management System Hardware and Software
Requirements and Specifications

3.1 HEM requirements and Specifications

Many HEMS systems have been implemented since 2009, but none of them has all the necessary features. The home management system that we want to implement in this project should achieve all of the following requirements for a complete functional system:

- **Monitoring**: the client should be able to control the electricity consumption of every single appliance in the house by switching it on and off wirelessly depending on the demand to lower as much as possible the total energy cost.

- **Disaggregation**: the system should provide specific information about each appliance in the house separately to keep track of the electricity consumption on the long term and also the impact of switching into an energy saving appliance.

- **Availability and accessibility**: the system should be available to the client throughout the day and be easily accessed from a user friendly interface using either a computer or a web or mobile portal.

- **Information and integration**: Other than the electricity consumption, the system should provide the client with other information like temperature, motion, light and humidity which is retrieved using different sensors. The client should also be able to have access to the historical data of the different appliances.

- **Affordability**: The system should cost the lowest price possible and consume the lowest energy possible to be affordable to the clients. Besides, maintenance, installment and configuration should be simple and an easy task for the consumers.

- **Control**: The system should offer automatic control of the appliances without failures.
• **Security and privacy:** The challenging side of the system is to make sure that the application cannot be accessed by other people without security authentication. The consumption profile should be secure, well protected and private.

• **Intelligence:** the system should perform intelligent actions and implement load disaggregation algorithms to help the client, who may not have enough knowledge and backgrounds about electricity, to balance his energy needs and energy consumption. [1]

### 3.2 Hardware Requirements

#### 3.2.1 Arduino Uno Microcontroller

Arduino Uno is an open source microcontroller development platform that uses the Atmega 328P microcontroller chip shown in figure 3.2.1.

![Atmega 328P microcontroller chip](image)

**Figure 3.2.1.1:** Atmega 328P microcontroller chip [20]
The different characteristics are provided in details in the table 3.2.1 and the components are illustrated in figure 3.2.1.2.

Table 3.2.1: Characteristics of the Arduino Uno and its specifications

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Arduino Uno Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>ATmega328</td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limits)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>14 (of which 6 provide PWM output)</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>6</td>
</tr>
<tr>
<td>DC Current per I/O Pin</td>
<td>40 mA</td>
</tr>
<tr>
<td>DC Current for 3.3V Pin</td>
<td>50 mA</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>32 KB (ATmega328) of which 0.5 KB used by bootloader</td>
</tr>
<tr>
<td>SRAM</td>
<td>2 KB (ATmega328)</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1 KB (ATmega328)</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16 MHz</td>
</tr>
<tr>
<td>Length</td>
<td>68.6 mm</td>
</tr>
<tr>
<td>Width</td>
<td>53.4 mm</td>
</tr>
<tr>
<td>Weight</td>
<td>25 g</td>
</tr>
<tr>
<td>Average Price (dhs)</td>
<td><strong>450</strong></td>
</tr>
</tbody>
</table>
The advantages of Arduino Uno are:

- Less costly compared with the other microcontrollers
- The IDE is simple and free to use and many tutorials online are available for help
- Multiple platforms option like Windows, Linux and Macintosh
- Open source software
- It can be connected to shields to provide with different options like Gps, wifi, zigbee communication, etc.

Arduino Uno can be powered up either using a USB cable or by using an external power source like a battery of 6 to 20 Volts or AC-DC adapter through the input voltage Vin pins, the 5V pins or 3.3V pins that regulate the output voltage on the board, ground pins and IOREF that provides with the reference of the voltage necessary for the microcontroller.
3.2.2 Wireless Communication Network

3.2.1.1 X-bee radio series 2

Xbee radio shown in the figure 3.2.1.1 is a module used for wireless communication between microcontrollers, computers or any device with serial port. It enables many communication protocols like Zigbee that we will be using in our project for the transmission of information between the gateway and the smartplugs. There are two types of modules, the series 1 and series 2. In our system, we will use X-Bee series 2 manufactured by Digi International [12] that is used for mesh and multiple point networks instead of using series 1 for point to point networks. Xbee series 2 requires a complete set up and configuration by uploading the XCTU multiple platform application into the computer. This latter is used to change the firmware on the radios like switching a device from a router to coordinator or changing the AT mode to API mode. The table 3.2.1.1 shows the performance, features, power requirements, dimensions and networking security.
Table 3.2.1.1: characteristics and requirements of X-Bee radio series 2 module

<table>
<thead>
<tr>
<th>Performance</th>
<th>Features</th>
<th>Networking security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor/Urban range</td>
<td>Adjustable power</td>
<td>Network Topologies</td>
</tr>
<tr>
<td>up to 133 ft. (40m)</td>
<td>Yes</td>
<td>Point to point, Star, Mesh</td>
</tr>
<tr>
<td>Outdoor RF line-of-sight range</td>
<td>Configuration method</td>
<td>Number of Channels</td>
</tr>
<tr>
<td>up to 400 ft. (120m)</td>
<td>API commands</td>
<td>16 Direct Sequence Channels</td>
</tr>
<tr>
<td>Transmit Power</td>
<td>Frequency</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>ISM 2.4 GHz</td>
<td></td>
</tr>
<tr>
<td>2 mW (+3dbm)</td>
<td>Antenna Options</td>
<td></td>
</tr>
<tr>
<td>Receiver Sensitivity</td>
<td>PCB, Integrated Whip, U.FL,</td>
<td></td>
</tr>
<tr>
<td>-98dbm (1% PER)</td>
<td>RPSMA</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Transmit Current (typical)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 mA (@ 3.3 V)</td>
<td></td>
</tr>
<tr>
<td>Features</td>
<td>Idle/Receive Current (typical)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 mA (@ 3.3 V)</td>
<td></td>
</tr>
<tr>
<td>Features</td>
<td>Serial Data Rate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>250 Kbps</td>
<td></td>
</tr>
<tr>
<td>Features</td>
<td>Operating Temperature</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-40 to 85 C</td>
<td></td>
</tr>
<tr>
<td>Power requirements</td>
<td>Supply Voltage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8 - 3.6 V</td>
<td></td>
</tr>
<tr>
<td>Power requirements</td>
<td>Power-down Current</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 uA</td>
<td></td>
</tr>
<tr>
<td>Dimensions</td>
<td>Dimensions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0960&quot; x 1.087&quot;</td>
<td></td>
</tr>
</tbody>
</table>
3.2.2.2 Xbee Shield

Figure 3.2.1.1: X-Bee Shield [13]

For a wireless communication between Arduinos via Zigbee, it is necessary to plug an X-Bee shield on the top of an Arduino Uno. It can be used either as a serial/USB replacement or put into a command mode. The shield obstructs the analog pins and the digital pins from 2 to 7; however, the pins 8 to 13 do not interfere in the Zigbee communication. [14]

3.2.2.3 Ethernet Shield

Figure 3.2.2.3: Arduino Ethernet shield [15]
The Arduino Ethernet shield shown in figure 3.2.2.3 enables the Arduino to connect to the internet in minutes. By plugging the shield on the top of the Arduino board and connecting it to the network with the RJ45 cable, this latter can send and receive information from the cloud. It contains an Ethernet ship Wiznet W5100 that provides the IP address of the network. Thanks to the Ethernet library, we can connect to the internet by writing sketches and uploading them to the Arduino. It also contains an SD card useful for local storage of data and files over the network. A reset controller ensures that the Ethernet module is activated when the Arduino is powered-up. The communication between the Arduino and the shield is done through the ICSP header and digital pins 10,11,12,13. The pin 10 selects W5100 and pin 4 selects the SD card. [15]

3.2.3 Sensors
3.2.3.1 Temperature Sensor

Figure 3.2.3.1: Temperature Sensor SEN 23292P [16]
The temperature sensor that will be used in the project is shown in figure 3.2.3.1. It detects temperature and sends data in the form of resistance value using a thermistor. This latter is a resistor that is sensitive to temperature and can change its value with the variation of temperature as it is illustrated in figure 3.2.3.2. The Arduino converts the voltage measured by the input pin to the real value of temperature in °C that varies between -40 to 125°C, with an approximated error of 1.5°C. The characteristics of the temperature sensor are given in details in table 3.2.3.1. [16]

Table 3.2.3.1: Temperature sensor requirements

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>3.3 ~ 5V</td>
</tr>
<tr>
<td>Max power rating at 25°C</td>
<td>300mW</td>
</tr>
<tr>
<td>Zero power resistance</td>
<td>10 KΩ</td>
</tr>
<tr>
<td>Operating temperature</td>
<td>-40 ~ +125 °C</td>
</tr>
</tbody>
</table>

Figure 3.2.3.2: Graph showing the variation of resistance with respect to temperature [16]
3.2.3.2  Light Sensor

The light sensor presented in figure 3.2.3.2 is the one selected for our home energy management system. It uses a light-dependent photoresistor LDR called GL5528 that detects light and which resistance decreases when light intensity increases. For more accuracy of data, the chip LM358 is used to get the voltage value. The characteristics of the sensor are given in the table 3.2.3.2.

Table 3.2.3.2: Light sensor requirements

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>3-5V</td>
</tr>
<tr>
<td>Supply Current</td>
<td>0.5-3mA</td>
</tr>
<tr>
<td>Light resistance</td>
<td>20KΩ</td>
</tr>
<tr>
<td>Dark resistance</td>
<td>1MΩ</td>
</tr>
<tr>
<td>Response time</td>
<td>20-30 secs</td>
</tr>
<tr>
<td>Peak Wavelength</td>
<td>540 nm</td>
</tr>
<tr>
<td>Ambient temperature</td>
<td>30~70°C</td>
</tr>
</tbody>
</table>
3.2.3.3 Current Transformer Sensor

Figure 3.2.3.3: Split-Core Current transformer sensor YHDC STC 013-030 [18]

The figure 3.2.3.3 shows the type of current sensor we will be using in the project. There are two types of current sensors. Indeed, the invasive sensor has a direct connection with the device to sense, which is not the case with the non-invasive sensors like the Hall Effect sensors and the split core sensors. The table below shows the characteristics of the current sensor.

Table 3.2.3.3: Characteristics of the current sensor YHDC STC 013-030 [18]

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>input current</td>
<td>0-30A</td>
</tr>
<tr>
<td>turn ratio</td>
<td>1800:1</td>
</tr>
<tr>
<td>resistance grade</td>
<td>Grade B</td>
</tr>
<tr>
<td>output voltage</td>
<td>0-1V</td>
</tr>
<tr>
<td>build-in sampling resistance</td>
<td>RL 62Ω</td>
</tr>
<tr>
<td>work temperature</td>
<td>-25°C ~ +70°C</td>
</tr>
<tr>
<td>non-linearity</td>
<td>±1%</td>
</tr>
<tr>
<td>dielectric strength between shell and output</td>
<td>1500V AC/1min 5mA</td>
</tr>
</tbody>
</table>
3.2.4 Relay Box

Figure 3.2.4: Control Relay Box [19]

Arduino Uno is a low power microcontroller. The intensity of the current leaving Arduino is approximately 40 to 50 mA, which it is not enough to power up a high power electrical device. The control relay as shown in figure 3.2.4 operates then as a switch that can turn on or off any home appliance. It is made of a relay, resistors, a transistor and nails to transmit power.

3.2.5 Open Source Arduino Software

Arduino is programmed using an IDE (Integrated Development Environment) that provides a text editor in which you can write a piece of code called a sketch, run it and upload it into the Arduino Uno hardware with a bootloader. It is available online on the official Arduino website for different operating systems, Linux, Windows and Macintosh. It uses the C/C++ languages and libraries specific to Arduino. The Arduino IDE can also be split into five areas:
- The bar of the 5 menus: file, edition, sketch, tools and help
- Toolbar containing the following buttons: verify sketch, compile and upload the sketch, new sketch, open sketch and save sketch.
- A bar of tabs
- The editor where the piece of code is written
- The console that displays any programming error found in the code ran and that is used for debugging.

Figure 3.2.5 Arduino IDE structure [21]
4 Home Energy Management System Design Phase

4.1 System Architecture

Figure 4.1: System architecture
4.1.1 Home Appliances

Home appliances form the first layer of the architecture. Indeed, the purpose of the system is to be able to track the electricity consumption and to reach efficiency in energy usage. Therefore, data should be accurately gathered from each individual appliance in order to be informed about which appliance consumes the most and at what time during the day. This latter is necessary to manage the electricity consumption.

There are two types of appliances, the renewable energy devices and home appliances. Renewable energy devices include the solar panels, the batteries and the inverters. The other home appliances are like fridges, TVs, heater, etc.

4.1.2 Current Sensors and Control Relay

Each appliance is connected with an Arduino Uno based smartplug composed of sensors and a relay box. The different sensors available for this project are the current sensors, temperature sensors and light sensors. They are used to collect data about the power of current consumed by each appliance, temperature and the light of the environment and transmit it to the user. This information is important to control and optimize the energy consumed. All the data gathered from each appliance lead to the overall load profile of the house. The purpose of the smartplug is not only to send information to the user, but to also receive commands and execute them through a relay box. This latter plays the role of a low signal power switch and can either switch on or off any home appliance, which leads to an accurate control of energy of every single device using load algorithms.
4.1.3 Arduino Uno

Arduino Uno constitutes two layers of the system architecture as it is basis of both the smartplug and the gateway. Arduino Uno is the microcontroller chosen for the implementation of this project thanks to its different advantages (refer to section 3.2.1). It has these main functionalities in this system:

- Store data collected by the sensors
- Send information wirelessly via zigbee to the gateway
- Send data and make any update to the database of the web server via Ethernet or Wi-Fi.
- Receive commands from the user and send them to the smartplugs
- Execute the commands by sending a request in the form of signals to the relay box to switch on or off any home appliance

4.1.4 Wireless Network Communication Layers:

To communicate wirelessly between the smartplug and the gateway, zigbee is the best option in terms of cost and performance (refer to section 2.3.3). Indeed, as we don’t need either internet or a larger range of communication than 100 feet, zigbee is the cheap, easy to use and appropriate communication protocol to consider for this system.

However, to connect to the webserverserver and update information continuously, internet access is necessary. Wifi or Ethernet communication protocols are used to allow the transfer of data from the gateway to the database and more generally, from the home energy management system to the home energy management web application.
4.1.5 Web Server

The web server processes the requests of the client on a web page via Hypertext Transfer Protocol (HTTP). It hosts the home energy management application, the database that contains all the data about the energy consumption in the house, and the php files that send information execute requests and assure a communication between the system and the application.

4.1.6 Database

The database collects all the data about the current consumed by the appliances sensed by the smartplugs and stores it. The tables contained in the database store information about the current power consumed by every single appliance and its status. This latter can be changed from on to off depending on the request of the client and the loading profile algorithms which are executed continuously by default. Besides, graphs and statistics can also be generated from the data stored to keep the client updated about the evolution of the daily energy consumption.

4.1.7 Web Application and User Devices

Clients can use either a computer or user devices like smartphones or tablets equipped with ios or Android to have access to the home energy management application. The Web application can be downloaded for free and has the following options and functionalities:

- Track the present and historical current consumption of each appliance
- Control the devices by changing the status from on to Off
- Generate and view the graphs and statistics
- Schedule the activity of the appliances depending on the evolution of the energy cost
- Inform the user about the metering and electricity price over the time
4.2 Detailed Architecture

4.2.1 Server Module and Gateway

4.2.1.1 The previous System Architecture

In the previous capstone research done by Ismail El Hamzaoui and Soukaina Brangui, the purpose of the gateway is to be able to achieve a double wireless communication with both the smartplugs via zigbee and the web server via internet. Arduino Uno does not have the option of communicating wirelessly via wifi with a web server, two suggestions were then made. The first possibility was to connect the Arduino Uno with the Xbee shield and the wifi shield one at the top of the other, but both need to be connected with the ICSP pins, so this option had to be declined. The second way of achieving the double communication was to use an Arduino Yun instead. The gateway consisted then on an Arduino Yun and an Xbee shield along with the Xbee series 2. However, after the implementation phase of the system this base station could not create a serial communication channel to interface with the Xbee module because of the absence of a library other than the usual serial library of Arduino IDE that was required[]. The Zigbee wireless communication was not successful.
4.2.1.2  The New System Architecture

Figure 4.2.1.2: Picture of the designed gateway:
After a lot of research done, we came up with a way of forming the gateway by using the Arduino Uno, an Ethernet shield and an Xbee shield. We discovered that only pins MISO, SCK, Reset and MOSI are necessary for the Ethernet shield to connect to the microcontroller and only pins +VCC and Gnd for the Xbee shield. The idea is to stack first the Ethernet shield on the Arduino Uno by connecting the necessary ICSP pins to the shield using jumper wires, then connect the Xbee shield on the top of the Ethernet shield by connecting the remaining pins of the ICSP to it as shown in figure 4.2.1.2.1. This would solve the issue of double wireless communication.
### Table 4.2.1.1: Different Components of the gateway

<table>
<thead>
<tr>
<th>Gateway components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Uno</td>
</tr>
<tr>
<td>Arduino XBee Shield v1.1</td>
</tr>
<tr>
<td>XBee Series 2</td>
</tr>
<tr>
<td>Jumper wires</td>
</tr>
<tr>
<td>Ethernet shield</td>
</tr>
<tr>
<td>Electric Battery 9V 6LR61</td>
</tr>
<tr>
<td>Powering Cable for 9V Electric Battery</td>
</tr>
</tbody>
</table>

#### 4.2.2 Server Controller

The smartplugs constituting the end nodes are composed of an Arduino Uno equipped with an Xbee shield and Xbee radio series2, sensors and a relay box (figure 4.2.2.1). At first, the Arduino Uno is powered with a 9V battery of with any other source like a wall plug. Sensors are connected to the analog input pins of the microcontroller to send data about the room temperature and light and the current consumed by any device. The relay box however, is connected to the output pin to receive the command from the Arduino Uno and execute them by either switching on or off the appliances.
Table 4.2.2 Different components of a Smartplug

<table>
<thead>
<tr>
<th>Smart Plug Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Uno</td>
</tr>
<tr>
<td>Arduino XBee Shield v1.1</td>
</tr>
<tr>
<td>XBee Series 2</td>
</tr>
<tr>
<td>Electric Battery 9V 6LR61</td>
</tr>
<tr>
<td>Powering Cable for 9V Electric Battery</td>
</tr>
<tr>
<td>Light Sensor SEN23292P</td>
</tr>
<tr>
<td>Temperature Sensor SEN11302P</td>
</tr>
<tr>
<td>Non-invasive Current Transformer Sensor</td>
</tr>
<tr>
<td>Arduino Breadboard Bundler</td>
</tr>
<tr>
<td>Relay Control PCB</td>
</tr>
<tr>
<td>Relay SPST-NO Sealed – 30A</td>
</tr>
<tr>
<td>Common BJT Transistor – NPN 2N3904</td>
</tr>
<tr>
<td>Super Bright LED – RED – 10,000mcd</td>
</tr>
<tr>
<td>Diode Small Signal – 1N4148</td>
</tr>
<tr>
<td>2x Resistor 1k Ohm 1/6th Watt PTH</td>
</tr>
<tr>
<td>Screw Terminals 3.5mm Pitch (3-Pin)</td>
</tr>
<tr>
<td>Screw Terminals 5mm Pitch (2-Pin)</td>
</tr>
<tr>
<td>10uF capacitor</td>
</tr>
<tr>
<td>3x Resistor 10k Ohm 1/6th Watt PTH</td>
</tr>
</tbody>
</table>
5 Home Energy Management System Implementation Phase

5.1 The previously Designed Wireless Actuation Network:

5.1.1 Components of the Previous Wireless Actuation Network:

In the previous capstone project, the wireless actuation network was based only on Arduino Uno. It consists on connecting a base station also called coordinator to the computer to send ‘on’ and ‘off’ commands to one or many standalone stations called end nodes in order to remotely control an LED. These end nodes also send data about temperature, light and current of each appliance to the base station to be stored. The control and sensing were as a result successful using Arduinos only without connecting with the Web server. The base station was composed of:

- Arduino Uno
- Xbee shield
- Xbee series 2

The end nodes were composed of:

- An XBee Shield: communication platform between the XBee chip and the ARDUINO UNO.
- An XBee Series 2: reads the control command from the base station, and gives it to the Arduino Uno as serial input.
- An Arduino Uno: It contains a program that can read the input via Xbee and modify the status of the output connected to the relay. To switch on the relay, a HIGH message is received and to switch it off the LOW message is received.
- A relay box: it switches on or off any device depending on the message transmitted by the gateway
The steps to follow to realize the Xbee wireless communication network are the following:

- Configure the Xbee series 2
- Prepare the base station
- Prepare the end node

More details about the steps can be found in last semester’s capstone report. However, because the new wireless actuation network is the continuity of the previously implemented Arduino only based system, and the three steps will be needed and reused throughout the whole implementation, here is the summary of the process and what has already been implemented.

5.1.2 Xbee Series 2 Configuration

The Xbee configuration is done using a terminal program called X-CTU. The base station should be configured in the API mode to allow the communication of the coordinator with multiple end nodes. Using X-CTU, the PAN (personal area network address), the destination address high and the destination address low should be set. The PAN address (PAN) is the address specific to the network, and the two radios should have the same PAN. The destination address high (DH) and the destination address low (DL) are a 64-bit address high found at the back of the Xbee radio. First, we connect the Xbee radio to the computer, then using the X-CTU terminal, we upgrade the radio’s firmware, and finally after setting the Xbee attributes, we define a radio to be a coordinator or a router.

5.1.3 Preparing the Base Station

The base station should be able first to send commands of switching on and off to one or many smart plugs and second, to receive data sensed by the smartplugs and store it.
5.1.3.1 Control Function

The base station reads commands received from the computer and sends them to the Xbee that reads the data from the XCTU terminal program. We first upload an empty sketch in the Arduino, then plug the Xbee on the Xbee shield, and finally plug the whole in the Arduino. We upload the piece of code shown in figure 5.1.3.1 into the Arduino. To test whether a command is successfully sent, we open the terminal program and type either ‘H’ or ‘L’ depending on the demanded status of the appliance. The commands are sent as “Zigbee Transmit Request” packet to an end node equipped with an Xbee shield as well, and the information is displayed in the computer.

```c
void control_function(int choice, int led_status){
    Serial.write(0x7E);
    Serial.write(0x00);
    Serial.write(0x10);
    Serial.write(0x10);
    Serial.write(0x01);
    if((choice == 1 || choice == 2) && (led_status == 0 || led_status == 1)){
        // Sets destination address
        if(choice == 1){
            Serial.write(0x00);
            Serial.write(0x13);
            Serial.write(0xA2);
            Serial.write(0x30);
            Serial.write(0x40);
            Serial.write(0xAF);
            Serial.write(0xEB);
        }
    }
}
```
Figure 1.5.3.1: The implemented control function of the gateway of the HEMS
5.1.3.2 Sensing Function

Data sensed by the smartplugs is sent to the base station specifying the 16Bit address of the Arduino Uno of the gateway (figure 5.1.3.2).

```c
void sensor_data(ZBExResponse rx){
  uint8_t* data = rx.getData();
  int data_length = rx.getDataLength();

  Serial.print(rx.getRemoteAddress64().getMsb(),HEX);
  Serial.println("-");
  Serial.print(rx.getRemoteAddress64().getLsb(),HEX);
  Serial.println("-");
  Serial.print(rx.getRemoteAddress16(),HEX);
  Serial.println("-");
  Serial.print(data_length);
  Serial.println("-");

  for( int i=0; i < data_length; i++){
    char Byte = data[i];
    Serial.print(Byte);
  }

  Serial.println();
}
```

Figure 5.1.3.2: The implemented (sensor_data) function of the gateway

5.1.4 Preparing the End Nodes

The purpose of the code implemented for the end node is to be able to receive a command of switching on or off the appliance it is connected to and to send any data sensed to the gateway.

5.1.4.1 Control Function

To form the end nodes, we first upload an empty sketch to the Arduino Uno powered by the computer, then like for the base station, we plug on the top of it the Xbee and Xbee shield. The end nodes are connected to a relay circuit. Once the code in (figure 5.1.4.1) has been
uploaded into the Arduino Uno of the smart plug, the control commands sent as a serial data
from the Xbee radio by the gateway are read. If data received is ‘H’, the relay switch is closed
and the digital pin 2 is set to a high voltage; however, if ‘L’ is received instead, the digital pin 2
is set to a low voltage. The energy monitoring and control is achieved by executing the
commands received and switch on or off any home appliance connected to the smart plug.

```c
void read_control_command() {
    XBee xbee = XBee();
    XBeeResponse response = XBeeResponse();
    ZBRxResponse rx = ZBRxResponse();
    ModemStatusResponse mos = ModemStatusResponse();
    xbee.readPacket();

    if (xbee.getResponse().isAvailable()) {
        Serial.println("got_something");
        if (xbee.getResponse().getApiId() == ZB_RX_RESPONSE) {
            // got a zb rx packet
            // now fill our zb rx class
            xbee.getResponse().getZBRxResponse(rx);

            if (rx.getOption() == ZB_PACKET_ACKNOWLEDGED) {
                // Calculate Size of the Payload
                uint8_t* RF_data = rx.getData();
                int sizeOfData = sizeof(RF_data);
                // Length of the entire Data String
                int DataLenght = rx.getDataLength();

                for (int i=0; i<DataLenght; i++)
                    Serial.println(rx.getData(i));
                Serial.println();

                // Reusable Packet_Info material = 64-bit address associated with its payload
                uint8_t Packet_Info[sizeOfData];
                // 64-bit Address
                XBeeAddress64 addr64 = rx.getRemoteAddress64();
                
                // This retrieves the 64-bit address of the sender XBee chip
            }
        }
    }
}
```

The classes XBee, XBeeResponse, ZNRxResponse and ModemStatusResponse are part of the XBee API library.
The XBee object is used to read, send packets.
ZBRxResponse is a data type that stores packets of type "Zigbee Receive".
Checks if any incoming packet us available.
Checks that the incoming packet is of type "Zigbee Receive".
Stores the "Zigbee Receive" in rx of type ZBRxResponse.
Retrieves the data in the incoming packet in an array. Calculate the DataLenght using getDataLenght().
Prints data on serial monitor for debugging.
This retrieves the 64-bit address of the sender XBee chip.
Once the code has been uploaded into the Arduino, the smartplugs send information that it senses by connecting to the three different sensors using the analog pins. It displays the readings through serial communication. More details can be found in the capstone report done by my teammate Barnicha Fatima Zahra.

5.2 The newly Designed Wireless Actuation Network

5.2.1 Phase 1: Design of the New Base Station

From the previous capstone research, an Arduino only based system that can achieve sensing and control of appliances has been designed. However, when the Arduino Uno was replaced by the Arduino Yun, which has the ability to connect to a web server via wifi, the double communication is not reliable anymore. For the HEMS to become standalone, the whole system remains the same, but an Ethernet shield is added to the gateway to allow an Ethernet connection. This was impossible to test before because both Xbee and Ethernet shields use the ICSP pins. Therefore, many possible designs of the gateway have been tested like the use of an Arduino Yun, an Arduino Yun connected to an Arduino Uno, and two Arduino Unos connected to each others, but the communication failed each time. However, after a lot of research, the use of exclusively some of the pins by the Ethernet shield and the rest by the Xbee shield was
experimentally discovered in the physics laboratory. As explained in the hardware design of the gateway in section 4.2.1, jumper wires were used to connect pins Gnd and VCC to the Ethernet shield and the remaining pins to the Xbee shield. The implementation of a standalone and reliable home energy management system based on Arduino Uno only is from now on possible.

5.2.2 Phase 1: Xbee Communication between the Newly Designed Gateway and Smartplugs

5.2.2.1 Pair Topology: One to One Wireless Actuation Network

The purpose of this first phase is to test the Xbee communication of the new gateway equipped with an Ethernet shield with one end node. This requires following the same three steps previously explained. The only difference is that the Arduino Uno and Xbee shield is replaced with a package of Arduino Uno, Ethernet shield and Xbee shield. The code to be uploaded to the base station is shown in figure 5.2.2.1. It contains a part that enables the transmission of the commands to a specific end node by precising the 64Bit address of the host at the beginning of the code.
```c
#include <Xbee.h>  //include Header File
#include <SPI.h>
#include <Ethernet.h>

#define MSG_LEN 18  //Message Length
  // Enter a MAC address for your controller below.
  // Never Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = { 0x90, 0xAC, 0xDA, 0x0F, 0x98, 0x3F};
  // if you don't want to use DNS (and reduce your sketch size)
  // use the numeric IP instead of the name for the server:
  // IPAddress server(74,125,232,128);  // numeric IP for Google (no DNS)
char server[] = "www.capreality.ma";
char inString[32];  // string for incoming serial data
int stringPos = 0;  // string index counter
boolean startRead = false;  // is reading?
  // Set the static IP address to use if the DHCP fails to assign
IPAddress ip(192, 168, 0, 177);
  //String sensedData;
uint8_t payload[MSG_LEN];
  //XbeeAddress64 addr64 = XbeeAddress64(0x0013A200,0x409022C7);
  //define 64-Bit Xbee Adress of the Remote Host
XbeeAddress64 addr64 = XbeeAddress64(0x0013A200,0x40AF1B3);
  //XbeeAddress64 addr64 = XbeeAddress64(0x0013A200,0x408C72B);
ZBTxRequest zbtX = ZBTxRequest(addr64, payload, sizeof(payload)); //Make a Xbee Packet

  //define MSG_LEN 15 //Message Length
  Xbee xbee = Xbee();

ZBxResponse rx = ZBxResponse();
int led = 13;
```
Figure 5.2.2.1: Implemented Control and sensing functions of one to one wireless actuation network

5.2.2.2 Star topology: Multi node wireless actuation network

This second test consists on testing the communication between the base station with multiple end nodes. This test is similar to the previous one, but instead of preparing only one end node, multiple nodes are activated. The topology network chosen is a star topology network in which each host node is connected with a point to point connection with many end nodes. Data
of all the end nodes flows simultaneously from the gateway without specifying the 64bit address of each targeted node. The command packets transmitted consists on a set of addresses followed by the data, and then each concerned end node would read and search for its address and get the command that follows. In the code previously written, XBeeAddress64 addr64= XbeeAddress64(0X0013A200,0X40AF1BF3) is replaced by XBeeAddress64 addr64= XbeeAddress64(0X00000000,0X0000FFFF), and the rest of the code is similar to the previous one.

5.2.3 Phase 3: connect to database

5.2.3.1 Connecting to internet

To connect to the internet, we upload the sketch to the board and connect the Ethernet shield to the computer using an Ethernet cable. The shield should be assigned a unique MAC address present at the back of the shield and an IP address. The function Ethernet.begin() in the code figure5.2.3.1.1 is responsible for the network setting.

```cpp
EthernetClient client;
void setup() {
    pinMode(led, OUTPUT);
    Serial.begin(9600);
    connectEthernet();
}

void connectEthernet() {
    if (Ethernet.begin(mac) == 0) {
        Serial.println("Failed to configure Ethernet using DHCP");
        // no point in carrying on, so do nothing forevermore:
        // try to configure using IP address instead of DHCP:
        Serial.println("HEPE");
        Ethernet.begin(mac, ip);
    }
    // give the Ethernet shield a second to initialize:
    delay(1000);
}
```
String readPage() {
    // read the page, and capture & return everything between '<' and '>'

    stringPos = 0;
    memset(&inString, 0, 32); // clear inString memory

    while(true){

        if (client.available()) {
            char c = client.read();

            if (c == '<') { // '<' is our beginning character
                startRead = true; // Ready to start reading the part
            } else if (startRead){

                if (c != '>'){ // '>' is our ending character
                    inString[stringPos] = c;
                    stringPos ++;
                } else{
                    // got what we need here! We can disconnect now
                    startRead = false;
                    client.stop();
                    client.flush();
                    Serial.println("disconnecting.");
                    return inString;
                }

            }
        }
    }
}

5.2.3.1.1: Sketch for internet connection
The user and the system communicate through a web application (figure 5.2.3.1.2) hosted with a database and php files constituting the web server.

- The web application is the mean of interaction between the user and the system in which the user can change the status of any home appliance by selecting on ‘On’ or ‘Off’ and get any data or plot displayed on the interface.
- The database is a collection of data hosted on the Web Server.
- Php files are the communication bridge between the web server and the gateway.

5.2.3.2 Control

When the user switches an appliance ‘On’ or ‘Off’, the status is automatically modified in the database. Once the gateway is connected to the web server, it sends a ‘get’ http request to a (getter.php) file as indicated in figure 5.2.3.2 to extract the status of each appliance. This file sends the address and the command to the appropriate end node to switch on or off the appliance if any change of status occurred.
Figure 5.2.3.2: Implemented control function through the web application

5.2.3.3 Sensing

The web application enables a user to insert any data and get information about the daily energy consumption. Once the gateway receives sensed data from the end nodes, it sends a ‘write’ request to the (inserter.php) file to store it in the database. The piece of code to be added is in figure 5.2.3.3.

```java
String connectAndRead()
{
    //connect to the server
    Serial.println("connecting...");

    // if you get a connection, report back via serial:
    if (client.connect(server, 80)) {
        Serial.println("connected");
        // Make a HTTP request:
        client.println("GET /Tested/getter.php HTTP/1.1");
        client.println("Host: www.caprealty.ma");
        client.println("Connection: close");
        client.println();
        return readPage();
    } else {
        return "connection failed";
    }
}
```
String connectAndWrite(String data){
    //connect to the server

    Serial.println("connecting...");

    // if you get a connection, report back via serial:
    if (client.connect(server, 80)) {
        Serial.println("connected");
        // Make a HTTP request:
        client.println("POST /Tested/inserter.php?data="+data+" HTTP/1.1");
        client.println("Host: www.caprealty.ma");
        client.println("Connection: close");
        client.println();
        return readPage();
    } else{
        return "connection failed";
    }
}

Figure 5.2.3.3: Implemented sensing function through the web application

5.2.4 Results

At the level of the smartplugs, figure 5.2.4.1 shows that information about temperature, current, light and the status ‘on’ or ‘off’ of the appliance are successfully sent to the gateway. All these information are followed by the data length and the 16bit source address.
Figure 5.2.4.1: Results of the two ways communication of the smartplugs

Moreover, the sensed data is also well transmitted to the gateway, and this latter sends the command to the smartplug to switch on or off the appliance as displayed in figure 5.2.4.1.
5.2.5 Constraints of the Implementation Phase:

5.2.5.1 Configuration Issue

During the testing of the Xbee communication series 2 between the base station and the smartplugs, sometimes the message as in figure 5.2.5.1.1 in the X-CTU terminal program is displayed. We tried to change the whole configuration, including the PAN ID, the 16bits serial numbers stored in the two addresses SH (serial high) and SL (serial low). However, the issue was still persisting and consequently, the problem was not caused by the configuration data input. We found out that the configuration should only be reset each time keeping the same entered numbers until the respond to serial communication becomes successful.
5.2.5.2 Congestion Issue

During point to point Xbee communication between the gateway and only one smartplug, all the data is transmitted without any delay or problem. When more smartplugs are integrated into the system, a congestion problem occurs. Indeed, all the nodes send data simultaneously, which can lead to data packet losses, queuing delays or complete interruption of connection. To avoid these problems, the solution we came up with was to modify the delay time so that the data does not collide while being transmitted.
6 Cost and Energy Demand Analysis

6.1 Cost Analysis and Comparison of Different HEMS

In this section, we estimated the total cost of three different HEMS by considering all the costs of each individual component of the system. Figure 6.1.1 estimates the total cost of the previous system implemented by Ismail El Hamzaoui and Soukaina Brangui. This system used Arduino Yun as the base of the gateway, which is pretty expensive and the communication back and forth between the gateway and smart plugs is not reliable. However, the new Arduino Uno only based home energy management system that we designed is cheaper according to figure 6.1.2, and the monitoring and control of energy is successful. Figures 6.1.3 displays in details the total cost of the HEMS considering different numbers of smartplugs that would be used in a house. The next challenge is to substitute Arduino Uno with a cheaper and more powerful microcontroller which is Raspberry Pie as shown in figure 6.1.4.
Table 6.1.1: Approximation of prices of the Arduino Yun based HEMS components

<table>
<thead>
<tr>
<th>Components</th>
<th>Price (Dhs)</th>
<th>Components</th>
<th>Price (Dhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Uno</td>
<td>362.21</td>
<td>Arduino YUN</td>
<td>876.10</td>
</tr>
<tr>
<td>Arduino XBee Shield v1.1 &amp; XBee Series 2</td>
<td>912.60</td>
<td>Arduino XBee Shield v1.1</td>
<td>912.60</td>
</tr>
<tr>
<td>Electric Battery 9V</td>
<td>62.28</td>
<td>XBee Series 2</td>
<td>62.28</td>
</tr>
<tr>
<td>6LR61</td>
<td>912</td>
<td>Electric Battery 9V</td>
<td>62.28</td>
</tr>
<tr>
<td>Powering Cable for 9V Electric Battery</td>
<td>9.60</td>
<td>Powering Cable for 9V Electric Battery</td>
<td>9.60</td>
</tr>
<tr>
<td>Light Sensor SEN23292P</td>
<td>53.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature Sensor SEN11302P</td>
<td>53.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-invasive Current Transformer Sensor</td>
<td>88.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arduino Breadboard Bundler</td>
<td>192.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relay Control PCB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relay SPST-NO Sealed – 30A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common BJT Transistor – NPN 2N3904</td>
<td>71.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Super Bright LED – RED – 10,000mcd</td>
<td>53.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diode Small Signal – 1N4148</td>
<td>12.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2x Resistor 1k Ohm 1/6th Watt PTH</td>
<td>17.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Screw Terminals 3.5mm Pitch (3-Pin)</td>
<td>5.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Screw Terminals 5mm Pitch (2-Pin)</td>
<td>18.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10uF capacitor</td>
<td>19.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3x Resistor 10k Ohm 1/6th Watt PTH</td>
<td>21.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>27.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Price</strong></td>
<td><strong>1991.87</strong></td>
<td><strong>Total Price</strong></td>
<td><strong>1860.58</strong></td>
</tr>
<tr>
<td><strong>Smart Plug:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Gateway:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6.1.2: Approximation of prices of the Arduino Uno based HEMS components

<table>
<thead>
<tr>
<th>Price approximation of the Components of the Smart Plug</th>
<th>Price approximation of the Components of the Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td><strong>Components</strong></td>
</tr>
<tr>
<td>Arduino Uno</td>
<td>Arduino UNO</td>
</tr>
<tr>
<td>Arduino XBee Shield v1.1 &amp; XBee Series 2</td>
<td>Arduino XBee Shield v1.1</td>
</tr>
<tr>
<td>Electric Battery 9V 6LR61</td>
<td>XBee Series 2</td>
</tr>
<tr>
<td>Powering Cable for 9V Electric Battery</td>
<td>Shield Arduino</td>
</tr>
<tr>
<td>Light Sensor SEN23292P</td>
<td>Ethernet</td>
</tr>
<tr>
<td>Temperature Sensor SEN11302P</td>
<td>Electric Battery 9V 6LR61</td>
</tr>
<tr>
<td>Non-invasive Current Transformer Sensor Arduino Breadboard</td>
<td>Powering Cable for 9V Electric Battery</td>
</tr>
<tr>
<td>Bundler</td>
<td></td>
</tr>
<tr>
<td>Relay Control PCB</td>
<td></td>
</tr>
<tr>
<td>Relay SPST-NO Sealed 30A</td>
<td></td>
</tr>
<tr>
<td>Common BJT Transistor – NPN 2N3904</td>
<td></td>
</tr>
<tr>
<td>Super Bright LED – RED – 10,000mcd</td>
<td></td>
</tr>
<tr>
<td>Diode Small Signal – 1N4148</td>
<td></td>
</tr>
<tr>
<td>2x Resistor 1k Ohm 1/6w Watt PTH</td>
<td></td>
</tr>
<tr>
<td>Screw Terminals 3.5mm Pitch (3-Pin)</td>
<td></td>
</tr>
<tr>
<td>Screw Terminals 5mm Pitch (2-Pin)</td>
<td></td>
</tr>
<tr>
<td>10uF capacitor</td>
<td></td>
</tr>
<tr>
<td>3x Resistor 10k Ohm 1/6w Watt PTH</td>
<td></td>
</tr>
<tr>
<td><strong>Total Price Smart Plug:</strong></td>
<td><strong>Total Price Gateway:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6.1.3: Approximation of the total price of the Arduino Uno based HEMS with different numbers of smartplugs

<table>
<thead>
<tr>
<th>Number of smartplugs</th>
<th>Total price of the system (Dhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>11758.44</td>
</tr>
<tr>
<td>10</td>
<td>21717.79</td>
</tr>
<tr>
<td>15</td>
<td>31677.14</td>
</tr>
<tr>
<td>20</td>
<td>41636.49</td>
</tr>
</tbody>
</table>
Table 6.1.4: Approximation of prices of the Raspberry Pie based HEMS components

<table>
<thead>
<tr>
<th>Components</th>
<th>Price (Dhs)</th>
<th>Components</th>
<th>Price (Dhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Uno</td>
<td>362.21</td>
<td>Raspberry Pie</td>
<td>672</td>
</tr>
<tr>
<td>Arduino XBee Shield v1.1 &amp; XBee Series 2</td>
<td>912.60</td>
<td>Arduino XBee Shield v1.1</td>
<td>912.60</td>
</tr>
<tr>
<td>Electric Battery 9V 6LR61</td>
<td>62.28</td>
<td>XBee Series 2</td>
<td>62.28</td>
</tr>
<tr>
<td>Powering Cable for 9V Electric Battery</td>
<td>9.60</td>
<td>Electric Battery 9V 6LR61</td>
<td></td>
</tr>
<tr>
<td>Light Sensor SEN23292P</td>
<td>53.82</td>
<td>Powering Cable for 9V Electric Battery</td>
<td>9.60</td>
</tr>
<tr>
<td>Temperature Sensor SEN11302P</td>
<td>53.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-invasive Current Transformer Sensor</td>
<td>88.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arduino Breadboard Bundler</td>
<td>192.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relay Control PCB</td>
<td>71.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relay SPST-NO Sealed – 30A</td>
<td>53.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common BJT 2N3904</td>
<td>12.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transistor – NPN 2N3904</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Super Bright LED – RED – 10,000mcd</td>
<td>17.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diode Small Signal – 1N4148</td>
<td>5.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2x Resistor 1k Ohm 1/6 Watt PTH</td>
<td>18.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Screw Terminals 3.5mm Pitch (3-Pin)</td>
<td>19.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Screw Terminals 5mm Pitch (2-Pin)</td>
<td>21.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10uF capacitor</td>
<td>12.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3x Resistor 10k Ohm 1/6 Watt PTH</td>
<td>27.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total Price Smart Plug:** 1991.87   **Total Price Gateway:** 1656.48
The purpose is to reach a high efficiency system with a minimum total costs. The graph 6.1.5 compares the prices of different systems, and shows that the new implemented system is cheaper than the previous one while having a better performance and reliability.

![Cost Analysis and Comparison of Different HEMS](image)

Figure 6.1.1: Comparison of prices of different HEMS

### 6.2 Energy demand and peak analysis:

#### 6.2.1 Loading disaggregation

Load disaggregation [22] is a dispatching of the daily tasks throughout the day to save a maximum of energy and consequently reduce the costs. The scheduling of tasks is done according to the variation of electricity prices, which vary with the consumption rate. Indeed, at peak times when the use of electricity is at its highest rate, there are additional costs for operations, maintenance and fuel to produce more energy than usual. Therefore, the prices rise when the demand of electricity increases. Task scheduling is also done considering the deadlines
and need to accomplish the task at specific time intervals. Some appliances are kept on continuously like refrigerators, while others are turned on in a flexible way for a period of time like washing machines, hair dryers, etc.

The objective of load disaggregation is to avoid peak demands, use a maximum of electricity when the prices fall and fulfill the customer’s needs of energy.

Load monitoring would be easy to perform in a small simple network of two-state appliances like toasters; however, the disaggregation faces many challenges:

- The appliances can have many states from one run to another like for example a washing machine; the power consumption varies depending on the temperature of wash, the speed of wash, etc.
- The power consumption of devices can differ even if the measured appliance is of the same kind.

### 6.2.2 Batmax Loading Algorithms

![BatMax Scheduling](image)

**Algorithm 1 BatMax Scheduling**

Use output of minMax as input:
- \( n \): number of time slots, \( m \): number of demands
- For each time slot \( t \) where \( 1 \leq t \leq n \):
- Aggregate energy demand \( E_t = \sum_j r_j \) for all task \( j \) assigned to \( t \)
- \( b_t \) is the battery buffer at time \( t \) and \( B \) is the charging rate of battery per hour.

**Output:**
- \( b_t \) for all time slot \( t \)

1. Initialize \( b_t \) to zeros
2. Compute \( \text{avg} = \frac{\sum E_t}{n} \)
3. for \( i = n \) to 2 do
4. if \( E_i + b_t < \text{avg} \) then
5. \( \text{avg} = \text{avg} + \frac{\text{avg} - E_i - b_t}{i-1} \)
6. end if
7. if \( E_i > \text{avg} \) then
8. \( o = E_i - \text{avg} \)
9. for \( k = i - 1 \) to 1 do
10. Find time slot \( k \) where \( E_k + b_k < \text{avg} \)
11. Fill \( b_k \) with \( o \) such that \( E_k + b_k = \text{avg} \), or \( b_k = B \), or \( o = 0 \)
12. Break loop when \( o = 0 \)
13. end for
14. if \( o > 0 \) then
15. \( E_i = E_i + o \)
16. \( \text{avg} = \text{avg} - \frac{E_i + b_t - \text{avg}}{i-1} \)
17. end if
18. end if
19. end for

Figure 6.2.2.1: BatMax scheduling algorithm
The task demand $d$ is a function of $s$ (start time), $f$ (end time), $r$ (electricity requirement per hour), and $l$ (task length) and is expressed as $d_j = (s_j, f_j, r_j, l_j) \in D$. The purpose is to determine the optimal assignment of tasks $\{\max t \{X dj \rightarrow t r_j\}\}$ while respecting the deadlines $s_j \leq t$ and $f_j \geq t + l_j - 1$. Scheduling a task $(dj \rightarrow t)$ means assigning a task to a time interval but it falls in the interval $\{s,f\}$. To realize these objectives, a BatMax scheduling algorithm (figure 6.2.2.1) has been designed. For its implementation, a battery is required to store extra electricity that has not been used during off-peak times. Besides taking as input the demand of each time slot produced by the minMax algorithms, starting from the last time interval, any extra demand that goes beyond the average is shifted to the time slots where demand falls below the average. This process is repeated continuously until all the gaps of low demand are filled as shown in figure 6.2.2.2.

![Figure 6.2.2.2: Demand-Side energy management](image_url)

Figure 6.2.2.3 shows the difference between a scheduled and unscheduled energy usage. Taking 9pm, the peak demand time, the total electricity consumed without load disaggregation is
approximately 2,345.320 kW but reaches only 1,697.194 kW when the home energy management system is used, representing a 27.6% reduction in energy demand.

Figure 6.2.2.3: Distribution of a scheduled and an unscheduled electricity consumption over the day.
7 Future Work

The following ideas can be developed in the future projects for a better performance of the system and to integrate other functionalities that may improve the energy management:

- Instead of using an Ethernet shield, a wifi shield stacked at the top of the Arduino can be tested to avoid plugging an Ethernet cable.
- Instead of these components of the current gateway, Arduino Uno, Ethernet shield and Xbee shield, a more powerful and cheaper microcontroller can be used to minimize the costs. Raspberry Pi can be used for the following reasons:
  - There is enough documentation about Raspberry Pi that can help us in the implementation of the network
  - Raspberry Pi is more powerful and less costly
  - Raspberry Pi can be programmed using any programming language like c++, java, Python, etc.
  - Easy internet use for the Pi but for Arduino, a Wifi shield is needed
- Load algorithms can be developed to be implemented and executed in the system for an optimal energy management
- Other sensors can be plugged in the smartplugs like humidity sensors, smoke sensors, motion sensors, etc, to inform the consumer about a maximum of information about home appliances
- Renewable energies can be integrated into the home energy management system to manage and control the amount of energy produced by the solar panels and the energy to be stored in batteries to contribute to maintain a minimum consumption of electricity.
- Test the whole system in a real house to collect data and identify any issues or needed functionalities to be added to the system.
8 Conclusion

The biggest challenge related to renewable and non-renewable energy supply is to prevent imbalances in the electricity network distribution. When there is extra energy produced, the excess of electricity should be reused in an optimal way to save a maximum of energy. However, an efficient and fruitful management of energy wouldn’t be possible without making the consumers aware of the fluctuations of prices, the variation of energy demand and the local residential energy consumption. This is the reason why countries are willing to transition from the traditional grid to an intelligent electrical network called smart grid.

To realize these objectives at the lower scale and to fulfill the requirements, a reliable and standalone home energy management system has been implemented in this project. The previously designed system had many limitations and was not working successfully. Indeed, the implementation was done using Arduino Uno only since the Arduino Yun didn’t allow for a reliable wireless communication. As Arduino Uno is not an internet-connected microcontroller, the purpose of this project was to design a new reliable and standalone application.

The system consists on informing the consumer about the daily consumption parameters and status of all the appliances of the house and enabling him to control them wirelessly at anytime and anywhere. The implemented system has been designed using Arduino Uno only and consists of two main components, the gateway and multiple smartplugs. The functionality of the system is to monitor and collect data about light, current and temperature of each appliance using a smartplug. This latter communicates wirelessly via zigbee the information to the gateway implemented using an Xbee shield and an Ethernet shield stacked at the top of an Arduino Uno. The data is then sent to the database hosted in a web server. The access of data is possible using any android or iOS device. The second functionality of the HEMS is to control the devices by switching them on or off from the application interface. The selected command is communicated to the gateway via wifi and to the appropriate smartplug to control the relay and change the status of the appliance accordingly.

As a result, the new home energy management system fulfills the following requirements:
✓ Monitoring and control
✓ Accessibility
✓ Affordability
✓ Reliability
✓ Security and privacy

The system can also be developed as indicated in future work section to transmit information about residential intelligent devices like renewable energy sources mostly solar panels, smart meters, smart cars, etc.
9 References


# 10 Appendices

![DerbSellicon Logo](image)

DerbSellicon Store  
Rabat  
Telephone: +212 6 50 95 44 21  
info@derbsellicon.com  
http://store.derbsellicon.com

---

**DEVIS POUR**

Université Al Akhawayn  
Hassen II Avenue, Ifrane, Meknès-Tafilalet Region, 53000, Morocco

<table>
<thead>
<tr>
<th>PRODUIT</th>
<th>QTE</th>
<th>PRIX UNITAIRE</th>
<th>PRIX TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino YUN</td>
<td>3</td>
<td>876.10</td>
<td>2628.28</td>
</tr>
<tr>
<td>Arduino UNO</td>
<td>10</td>
<td>362.21</td>
<td>3622.08</td>
</tr>
<tr>
<td>Shield Arduino Ethernet</td>
<td>1</td>
<td>452.40</td>
<td>452.40</td>
</tr>
<tr>
<td>XBee Shield (Arduino Add-on)</td>
<td>10</td>
<td>912.60</td>
<td>9126</td>
</tr>
<tr>
<td>GSM Shield (Arduino Add-on)</td>
<td>1</td>
<td>1306.80</td>
<td>1306.80</td>
</tr>
<tr>
<td>Wii Shield (Arduino Add-on)</td>
<td>2</td>
<td>1162.37</td>
<td>2324.74</td>
</tr>
<tr>
<td>Light Sensor (Arduino Sensors)</td>
<td>6</td>
<td>53.82</td>
<td>322.92</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>2</td>
<td>672.00</td>
<td>1344</td>
</tr>
<tr>
<td>Temperature Sensor (Arduino Sensor)</td>
<td>6</td>
<td>53.82</td>
<td>322.92</td>
</tr>
<tr>
<td>Alimentation Cable for 9V electric battery</td>
<td>6</td>
<td>9.60</td>
<td>57.60</td>
</tr>
<tr>
<td>Electric Battery 9V 6LR61</td>
<td>20</td>
<td>62.28</td>
<td>1245.6</td>
</tr>
<tr>
<td>Accessories, wires ... Etc</td>
<td></td>
<td>Besoin de plus de détails sur l'article</td>
<td></td>
</tr>
<tr>
<td>Arduino Breadboard Bundle</td>
<td>5</td>
<td>192.60</td>
<td>963</td>
</tr>
<tr>
<td>Non-invasive AC Current Sensor</td>
<td>10</td>
<td>88.20</td>
<td>882</td>
</tr>
<tr>
<td>AC to AC power adapter</td>
<td>5</td>
<td>Besoin de plus de détails sur l'article</td>
<td></td>
</tr>
<tr>
<td>SmartLockPro GFCI (Ground Fault Circuit Interrupt) Outlet</td>
<td>2</td>
<td>728.00</td>
<td>1440</td>
</tr>
<tr>
<td>Nail Mount Housing</td>
<td>2</td>
<td>Besoin de plus de détails sur l'article</td>
<td></td>
</tr>
</tbody>
</table>

www.derbsellicon.com - contact@derbsellicon.com - www.facebook.com/derbsellicon
<table>
<thead>
<tr>
<th>PRODUIT</th>
<th>QTE</th>
<th>PRIX UNITAIRE</th>
<th>PRIX TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relay Control PCB</td>
<td>3</td>
<td>71.10</td>
<td>213.3</td>
</tr>
<tr>
<td>Relay SPST-NO Sealed - 30A</td>
<td>3</td>
<td>53.10</td>
<td>159.3</td>
</tr>
<tr>
<td>Common BJT Transistors - NPN 2N3904</td>
<td>3</td>
<td>12.00</td>
<td>36</td>
</tr>
<tr>
<td>Super Bright LED - Red - 10,000mcd</td>
<td>3</td>
<td>17.10</td>
<td>51.3</td>
</tr>
<tr>
<td>Resistor 10k Ohm 1/6th Watt PTH</td>
<td>3</td>
<td>9.00</td>
<td>27</td>
</tr>
<tr>
<td>Diode Small Signal - 1N4148</td>
<td>3</td>
<td>5.40</td>
<td>16.2</td>
</tr>
<tr>
<td>Resistor 1k Ohm 1/6th Watt PTH</td>
<td>6</td>
<td>9.00</td>
<td>54</td>
</tr>
<tr>
<td>Screw Terminals 3.5mm Pitch (3-Pin)</td>
<td>3</td>
<td>19.38</td>
<td>58.14</td>
</tr>
<tr>
<td>Screw Terminals 5mm Pitch (2-Pin)</td>
<td>3</td>
<td>21.66</td>
<td>64.98</td>
</tr>
<tr>
<td>SMART METER ITRON</td>
<td>1</td>
<td>Besoin de plus de détails sur l'article</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL H.T.**  
26 718.56 DH

**FRAIS DE LIVRAISON**  
100 DH

**TVA (20%)**  
5 343.71 DH

**TOTAL À PAYER TTC**  
32 182.28 DH

*Ce devis est valide pour la durée d’un mois, le délai de livraison 21 jours à partir de la date de commande.*