Controlling electrical home appliances, using Bluetooth Smart technology

Pedro José Vieira da Silva

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Maria Helena da Costa Matos Sarmento

Examination Committee

Chairperson: Prof. Gonçalo Nuno Gomes Tavares

Supervisor: Prof. Maria Helena da Costa Matos Sarmento

Members of the Committee: Prof. José António Beltran Gerald

November 2015

Abstract

Home Automation is a technological evolving subject, permitting to build smart houses. In a smart house, there is a home area network with all devices interconnected. These devices can be monitored and controlled by the homeowner inside or outside the house, and information can be exchanged between them. Home automation systems permit to improve comfort, security and energy efficiency at home. Energy efficiency management is a more recent add-on to Home Automation systems. Energy efficiency management intends to optimize the usage of electrical devices, connecting and disconnecting devices based on real-time price of electricity. Wireless technologies permit to implement the home area network, avoiding the use of additional wires. Wi-Fi, Bluetooth, ZigBee and Z-Wave are possible options, but Bluetooth Low Energy (BLE) is selected due to the reduced power consumption, low cost, and easy connection to tablets and smartphones.

This dissertation proposes the architecture of a Home Automation system, including Energy Management and implements a demo prototype. Bluetooth Low Energy is responsible to support communications. The prototype demonstrates the capability to control and monitor home appliances. It includes a main controller, local controllers, sensors, and actuator, being all connected to the BLE home area network. The main controller is connected to an Internet router. Home appliances can be accessed directly using BLE, or through the main controller using the Internet. Tests of power consumption and range and latency tests confirm BLE as possible technology for Home Automation.

Keywords: Home Automation, Smart houses, Energy Management, Bluetooth Low Energy.

Resumo

A Domótica é uma área em evolução tecnológica, permitindo a construção de casas inteligentes. Numa casa inteligente, existe uma rede de área local com todos os dispositivos interligados. Estes dispositivos podem ser monitorizados e controlados pelo proprietário dentro ou fora de casa, e podem trocar informação entre eles. Os sistemas de Domótica permitem melhorar o conforto, a segurança e a eficiência energética em casa. A gestão da eficiência energética é a mais recente funcionalidade presente nos sistemas de Domótica e pretende otimizar a utilização de dispositivos elétricos, ligando e desligando os dispositivos com base no preço em tempo real de eletricidade. Tecnologias sem fio permitem implementar a rede de área local, evitando o uso de fios adicionais. *Wi-Fi*, *Bluetooth*, *ZigBee* e *Z-Wave* são opções possíveis, mas *Bluetooth Low Energy* (BLE) foi selecionado devido ao reduzido consumo de energia, baixo custo e fácil conexão com *tablets* e *smartphones*.

Esta dissertação propõe a arquitetura de um sistema de Domótica, incluindo a gestão de energia e implementa um protótipo de demonstração. O protótipo demonstra a capacidade para controlar e monitorizar eletrodomésticos e inclui um controlador principal, controladores locais, sensores, e um atuador, estando todos ligados à rede de área local, suportada pelo BLE. O controlador principal está diretamente conectado a um *router* de Internet. Os aparelhos em casa podem ser acedidos diretamente usando o BLE, ou através do controlador principal utilizando a Internet. Testes de consumo de energia, de alcance e de latência confirmam a possibilidade de utilizar o BLE para a Domótica.

Palavras-chave: Domótica, Casas inteligentes, Eficiência Energética, *Bluetooth Low Energy*.

Contents

List of Figures

List of Tables

List of Acronyms

ACI Application Controller Interface
AES Advanced Encryption Standard
AMI Advanced metering infrastructure
ATT Attribute Protocol
BLE Bluetooth Low Energy
bps Bit per second
CRC Cyclic redundancy check
DK Development Kit
DR Demand Response
DSO Distribution System Operator
EK Evaluation kit
GAP Generic Access Profile
GATT Generic Attribute Profile
GUI Graphical User Interface
HA Home Automation
HAN Home Area Network
HCI Hardware Controller Interface
HEM Home Energy Management
HVAC Heat, Ventilation and Air Conditioning
HW Hardware
IC Integrated Circuit
IoT Internet of Things
ISM Industrial, Scientific, Medical
LAN Local Area Network
L2CAP Logical Link Control and Adaptation Layer
M2M Machine-to-Machine
PDU Protocol Data Unit
PLC Power Line Communication
PoE Power over Ethernet
RF Radio frequency
SDIO Secure Digital I/O
SIG Special Interest Group
SM Security Manager
SoC System-on-chip
SW Software
TI Texas Instrument
UART Universal Asynchronous Receiver/ Transmitter
USB Universal Serial Bus
Wi-Fi Wireless Fidelity
WiMAX Worldwide Interoperability for Microwave Access
WWW World Wide Web

1 Introduction

Home Automation (HA), a technological evolving subject, permits to improve, comfort, security and energy efficiency at home. Smart houses use Home Automation systems where the homeowner or pre-defined algorithms, controls home equipment (lighting, air-conditioning, home appliances, door-locks, etc.), based on monitored home variables parameters (temperature, light intensity, humidity, intrusion, etc.). The user can control the equipment connected to the HA system by using a smartphone, a tablet or a personal computer, locally or through the Internet.

Energy efficiency management is a more recent add-on to Home Automation systems. It permits to reduce end-users costs. Reduction of costs can be attained by optimize consumption and by using periods of lower electricity tariffs. For example, to optimize consumption, lights turn on if sensor detects the presence of a person in a room, and switches off when the person leaves the room, or light switch on/off is based on the sun light sensed by a sensor. Reduction of costs by using lower tariffs means for instance to turn on for instance a washing machine, only when electricity cost is low.

Lower tariffs is how DSO offer incentives to customers to reduce electricity consumption during peak periods. Peak consumption is reduced (peak shaving) because the consumption (demand) is more uniformly distributed along the day.

Home Energy Management (HEM) systems already exists on the market, however the information about the real-time price of electricity is used just to inform the homeowner. In general, the dynamic information about tariffs is provided through a smart meter installed by the Distribution System Operator (DSO). The energy management system, being aware of the electricity price and considering preferences and priorities specified by the homeowner, controls the electricity usage, by scheduling the use of home electrical appliances during the day.

In a HAN/HEM system, the main controller, the electrical appliances, the sensors and the smart meter must exchange information. Wired or wireless technologies can support the communications. Power Line Communication (PLC), that uses the electric wiring to communicate, and short wireless technologies, such as Wi-Fi, Bluetooth, ZigBee and Z-Wave are good candidates because they avoid new cables. Bluetooth Low Energy (BLE), an upgrade of Bluetooth classic, designed to reduce power consumption is the technology was initially proposed for this work. A comparison of the short-range wireless tecnhology permitted to evaluate if the BLE is a good alternative, due to its low power consumption, low cost and its general availability in tablets and smartphones, which can be easily used to control and monitor the all system.

The general objective of this thesis work is to specify a HAN system, with energy management capabilities (HEM) where appliances, sensors and other equipment is connected using BLE technology. The specific objectives are:

- To study the BLE technology comparing its advantages and disadvantages over other short-range wireless technologies for HAN systems.
- To specify the architecture of a HEM system based on analysis of requirements.
- To select the hardware platform to develop a prototype/demonstrator.
- To select some the minimum hardware permitting to implement the demonstrator.

- To develop the firmware, the application software and the user interface.
- To test the prototype.

After this introduction, the second chapter presents concepts related to Home Automation, Energy Management, Smart Grid and Internet of Things, and describes some commercially and under development HAN/HEM systems. Chapter 3 briefly analyses and compares existent short-range wireless technologies, such as Wi-Fi, ZigBee, Classic Bluetooth and Z-wave, presented in more detail BLE. Chapter 4 proposes an architecture for a HEM system, and describes the implemented prototype. Chapter 5 describes the tests and presents the obtained results. Finally, last chapter presents some conclusions and suggests future work.

2 Home Automation and Home Energy Management

In HA systems the homeowner can monitor and control his house from anywhere. Although, in HEM systems, the homeowner has control over its energy consumption to control costs and to contribute for environmental issues. HA systems improvements can benefit with research on Energy Management, Smart Grids, wireless communications such as IoT and even on smart appliances.

This chapter describes HA/HEM systems, presenting commercial and under developing systems, and introduces the concepts of Smart Grid, including DR, and IoT that can improve HA/HEM systems on the future.

2.1 Smart Grid/Demand Response

The balance between consumption and production is one of research topics addressed by Smart Grids. Smart Grids research also addresses energy storage, electrical vehicles, distribution automation, Advanced Metering Infrastructure (AMI), Demand Response (DR), etc.

The purpose of Smart Grid is to make electricity more reliable, economical and sustainable. Smart Grid research addresses topics such as energy storage, electrical vehicles, distribution automation, Advanced Metering Infrastructure, Demand Response, etc.

Demand Response mechanisms appear to balance energy consumption and production, in order to reduce the peak load. Peak load is a term used in energy demand management and refers to a period where electrical consumption is significantly higher than average supply level. To reduce peak load, the distribution of electricity consumption should be uniformly along the day. Monetary incentives to use electricity at night are already widespread. However, in Smart Grids, Demand Response goes further, being defined as "changes in electric use by demand-side resources from their normal consumption patterns in response to changes in the price of electricity, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized" [1].

Many DSOs are using the Advanced Metering Infrastructure to implement Demand Response. It is through the AMI infrastructure that the DSO deploys dynamic pricing programs to customers, dependent upon time, and uses the different prices to render the monthly bill correctly. Implementation of smart meters allows the two-way communication between the electricity meter, from the costumer, and the DSO.

HEM systems have the capability to contribute to DR at the customer side, scheduling automatically and in real time the operation of electrical appliances at home. The HEM system can establish the connection between the smart meter (AMI) and the electrical appliances.

2.2 Internet of Things

Autonomous systems are becoming increasingly intelligent and independent. Typically, IoT hopes to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine communications (M2M) and covers a variety of protocols, domains, and applications [2]. Therefore, it is need to uniquely identify and plug things into the main network, the Internet, in order to achieve that independence. Internet of things makes this possible.

Figure 2.1 shows the evolution from the communication between two computers, followed by the communication between computers through the Internet, the creation of the World Wide Web (WWW), the use of mobile internet, and the communication using smart devices like tablets, smartphones and personal computers, until the next step, the IoT connecting everything.

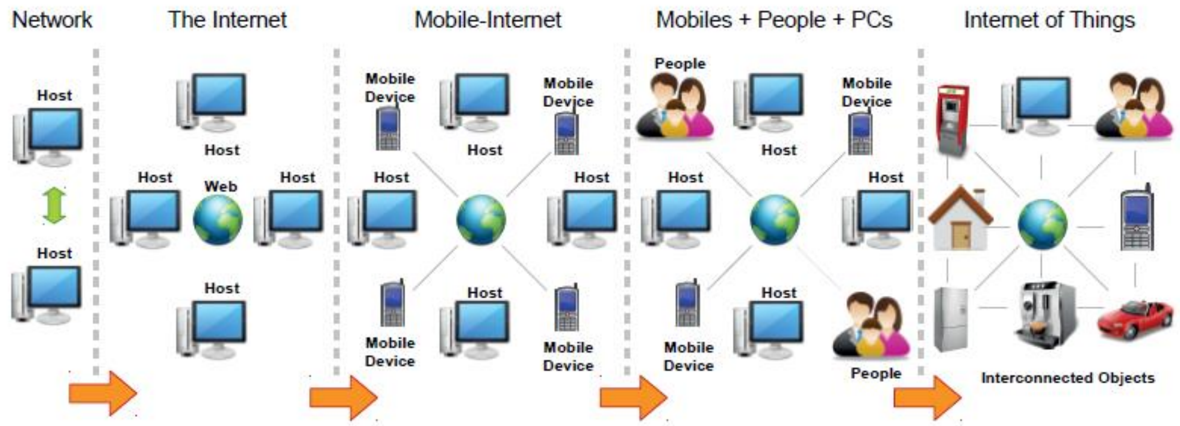

Figure 2.1 - From Internet of People to Internet of Things (extracted from [3]).

These “things” can be cars, domestic appliances, industrial machines, or any other natural or manufactured object where it is possible to assign an IP address. Thus, things have the ability to transfer data over network. IoT will be able to connect digital and physical entities and enable a completely new class of applications and services. This is the future of smart homes, buildings, hospitals, factories and cities.

Nowadays, smart houses works with sensors and electrical appliances linked to the home area network (HAN). However, in the future, IoT can make simpler the integration of smart appliances and sensors to build HA/HEM systems, because Internet will support communications. Then, monitor and control the house is possible from everywhere, because any “thing” will connect directly to internet.

2.3 HEM system overview

HEM system gives the homeowner full control and management of his home, from anywhere. HEM system consist in connect sensors and electrical home appliances inside the house. Then, an intuitive user interface provides all information about the house, like the data from sensors and state of appliances, and accept orders from user. In the user interface, the homeowner can turn on and off electrical appliances directly or he can set a period of time that the electrical home appliance must work. The user interface also accepts pre-set target values that HEM system achieve autonomously, like the temperature inside the house or lightness. The user interface is password protected, to prevent access to strangers.

The HEM system provides a daily schedule with electricity usage, respecting homeowner’s settings and orders, and tries to reduce electricity expenses based on real-time electricity price and electricity consumption. The HEM system is always checking data from sensors and actuate autonomously if necessary. The list below shows examples of direct and indirect controls.

- **Direct control.**
    - Lights.
    - Door locks.
    - Activate/deactivate alarm system.
    - Washing machine.
    - Dryer machine.
    - Oven.
    - Coffee machine.
- **Indirect control.**
    - Temperature.
    - Humidity.
    - Lightness.
    - Alarm system.

2.4 Home Automation / Energy Management architecture

HEM and HA systems have a very similar architecture. They both include one main controller, sensors, actuators, home appliances and a platform to support connections (wired or wireless). The HEM system also include an AMI, in order to manage energy consumption.

The main controller collects data from sensors and controls actuators and home appliances. It has a processing unit to actuate, autonomously, to environmental changes (temperature and humidity). The main controller also has a memory unit to store the homeowner preferences, such as environmental targets. This main controller must be compatible with communication platform implemented on HA/HEM system. Particularly on HEM systems, the main controller also communicates with an AMI to build an energy management schedule in order to turn on electrical appliances in a low-price periods, if it is possible.

To exchange information, all devices in HA/HEM system connect to the home area network (HAN). This network provides a path (physical or over-the-air) to connected home devices in order to exchange information between them. I.e., HAN makes possible to send and receive data packets, which contain device’s information, between all connected devices. As well as the homeowner can access to HAN through a smartphone, table or laptop, he can directly control electrical appliances and actuators and monitors environmental values.

Home automation systems must be easy to install and therefore a wireless connection should be the best solution since it is not necessary to drill holes through walls to run wires, unless these systems belongs to the original plan of the house. In the second case, the system will probably cease to be flexible and hard to change. Another alternative is use the wired technology called Power Line Communication (PLC), which use the existing electrical cables to exchange information between devices. However, PLC technology is a discarded option because it causes a lot of interference with appliances that use Radio Frequency (RF) like audio receivers, wireless mobiles, etc. [4].

Figure 2.2 shows an illustrative of the architecture of a Home Energy Management (including AMI). On the right side exist an AMI that communicate with the controller, located inside the house and this controller will communicate with electrical appliances that can be lights, washing machine, HVAC (Heat, Ventilation and Air Conditioning), and so many others. It is implicit that all devices are linked to a home are network that supports communications between them.

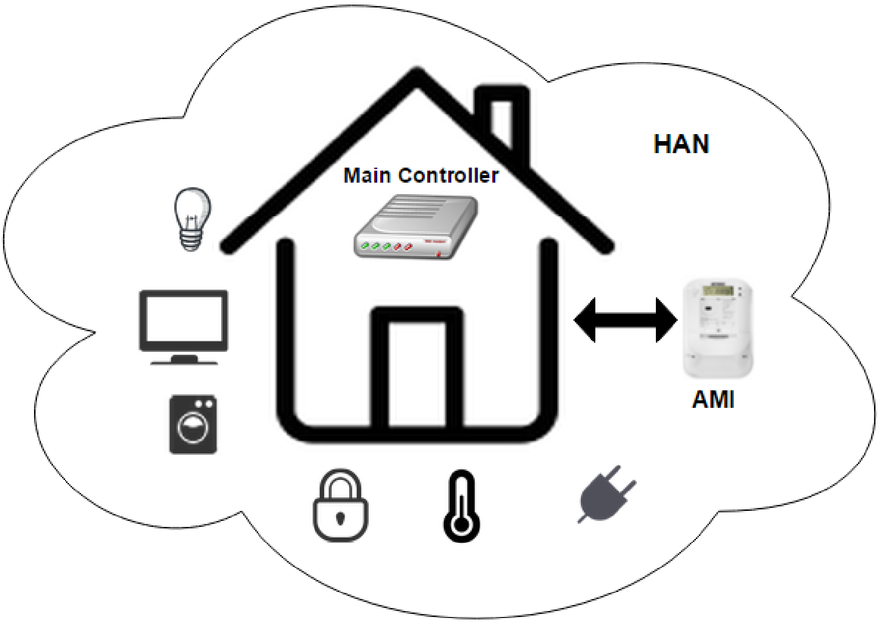

Figure 2.2 - Smart House with Energy Management.

2.5 Home automation/Energy Management systems

This section briefly describes two HA systems Control4 and HT-Zee and two HEM systems, AiMiR and one under developing [5].

Control 4

The Control4 [6] system can control automatically lighting devices, thermostats, the audio/video distribution system and the security system as presented in Figure 2.3. The master controller connects to a HAN, implemented with wired or wireless technologies, and to a ZigBee mesh network. The ZigBee network connects lighting devices and thermostats. The HAN connects the audio/video distribution system, the touch screens and the optional slave controllers, to use in different rooms. Either Power over Ethernet (PoE) or Wi-Fi performs the support communication on the HAN. Other audio/video components and the security system connect via a serial link (wired).

Finally, the homeowner actuate over a handheld remote control to control his appliances. Control4 also offers the possibility to run a user interface on a smart TV and control the home system through his TV.

The cost of starter kit including the HC-250 brain controller and a SR-250 wireless remote control to control devices of an entertainment system (TV, DVD player), music system, lighting, and more cost around 800 € [7].

HomeTroller Zee

HomeSeer develops systems to control lighting, temperature, security, mobile phones, irrigation, window shades and home entertainment equipment. HomeTroller Zee [8] (HT-Zee), presented in Figure 2.4, includes the cheapest HomeSeer controller.

The system includes a main controller wireless connects (Z-Wave) with the controlled devices, like lights, thermostats, doors, electric outlets, etc. It also connects with Internet via a Wi-Fi router, making possible to access and manage the Home Automation system from anywhere. Access to a local area network (LAN), with the specified software developed by HomeSeer, also allows the system management. This software can run as a program in a personal computer, or as an app in a tablet or smartphone.

HT-Zee controller cost around 160€ and the electrical devices (sensors and actuators) cost around 20€ to 50€.

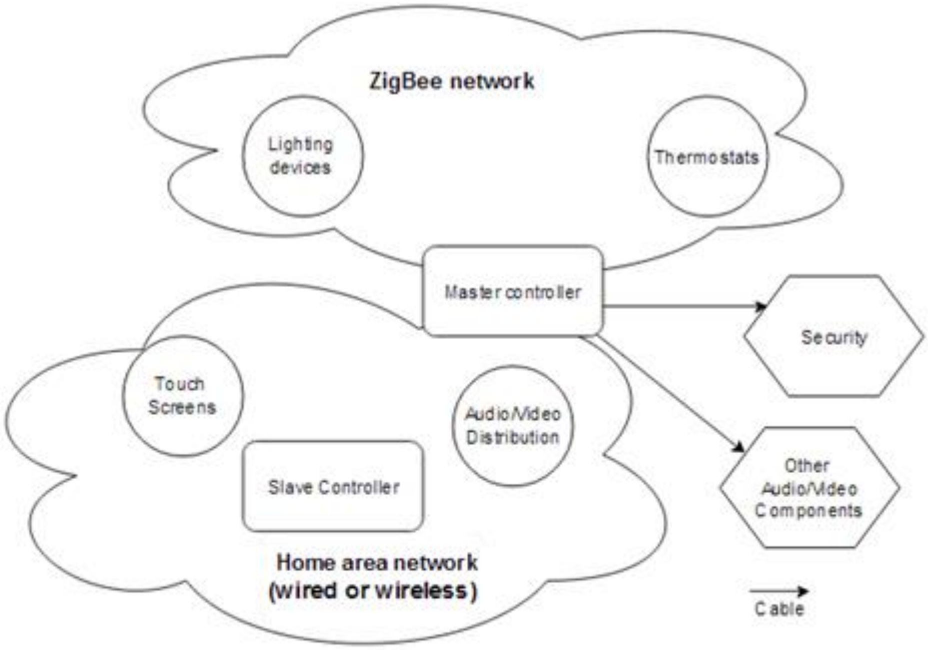

Figure 2.3 - Control4 architecture.

AiMiR

AiMiR [10] develops systems to manage and control home energy consumption. In Figure 2.5, it is possible to see that the architecture of AiMiR HEM system includes a main controller, smart appliances, electrical appliances, sensors, a router and some especial devices that help to measure and monitor power consumption. Smart appliances are prepared to change data over the air, which is a great advance, since there is not necessary to add extra hardware to control and monitor the appliance.

Monitor the price of electricity usage gives the homeowner an opportunity to make informed decisions about when use energy and allows control the costs. Therefore, AiMiR develop the AiMiR Home Metering Unit and AiMiR Smart Plug.

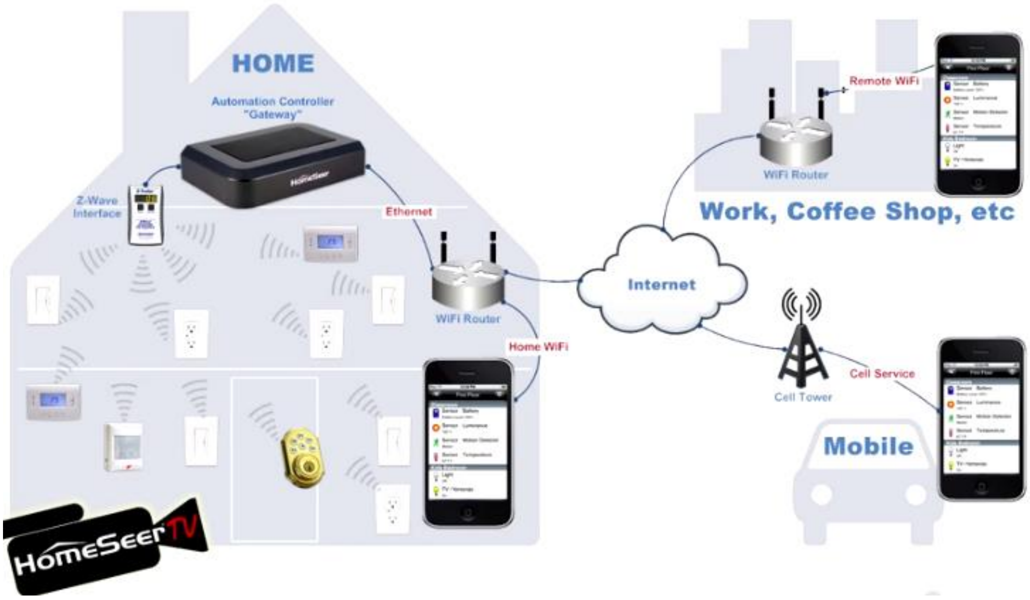

Figure 2.4 - HomeTroller Zee architecture (extracted from [9]).

The system uses a developed smart meter from DSO to receive in real time the price of electricity and to send to DSO the consumption of the house. AiMiR Home Metering Unit metering whole house power consumption, including voltage, current, power factor, etc., and provide that information to AiMiR in-home device (IHD), wirelessly. AiMiR Smart Plug is a device, which is between smart appliances and the electric outlet, and is responsible to measure the consumption of respective device. AiMiR Smart Plug also connects to AiMiR IHD. Therefore, AiMiR IHD receives data from these three devices and is able to provide that information to homeowner.

AiMiR IHD also connects to a router, creating a ZigBee HAN, which provides to homeowner the ability to monitor and control his system from home, linked to HAN. As IHD also connects with a router, the homeowner can interact with his system or from anywhere, trhough Internet. Homeowner also can manage the system in the AiMiR IHD display. Unfortunately, the website does not present the commercial price.

These systems provide different advantages to the costumer. Control4 and HomeTroller Zee systems provide just home automation, while AiMiR system provide home automation with energy management. Both Control4 and HomeTroller Zee systems are difficult to install in a built house, because they require a lot of Hardware and some wires that difficult the installation process within a built house. Although, AiMiR system requires less wires and less hardware. However, it controls smart appliances that must be part of the system. Concluding, for a costumer with a built house without smart appliances, none of these systems is a good solution. Considering a costumer that will build a house, AiMiR system is the most powerful and energy efficient system of these three.

Figure 2.5 – AiMiR HEM architecture (extracted from [10]).

The next paragraph motivate the smart/autonomous energy management, presenting an algorithm that can predict the price of electricity during a day. There is not on-market systems that perform an autonomous management of electrical appliances, considering energy saving.

Research

The authors of [5] develop an algorithm for a Home Energy Management scheduler to reduce the cost of energy consumption, using real-time pricing of electricity, provided by an advanced metering infrastructure (AMI). The algorithm receives the real-time electricity monitoring, appliance's data and stochastic scheduling, and provide decisions about the state of appliances. The stochastic scheduling is important due to uncertainties in electricity price variation, appliance operation, user behaviour and preferences. The stochastic scheduling use a Markov decision process to minimize cost of energy consumption by predicting the appropriate curtailment of appliances based on the stochastic behaviour of cost of consumption. Then, Home Energy Management scheduler computes an optimal policy using stochastic dynamic programming to select a set of appliances to control. Finally, the algorithm produces outputs, based on HEM scheduler, that control selected home appliances.

This autonomous HEM scheduler is only available when the price of electricity is dynamic and changes with variations of energy consumption. Besides, simulation results validate the proposed method for HEM scheduler. Figure 2.6 represents a descriptive diagram of HEM scheduler and unit interference.

These results proof that is possible to perform an autonomous energy management to reduce consumption during the peak loads.

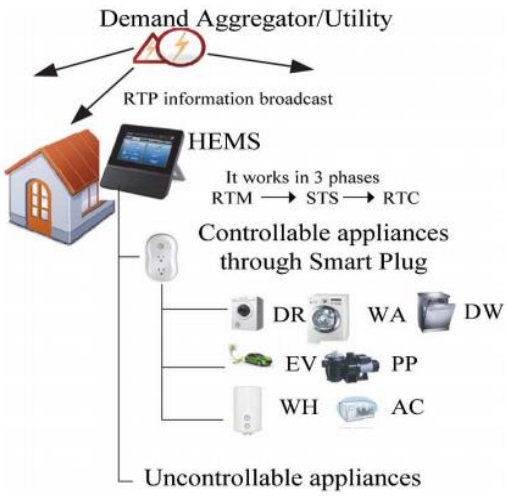

Figure 2.6 - Descriptive diagram of HEM scheduler and utility interference (extracted from [5]).

HEM – HEM scheduler; RTP – Real-time pricing; RTM – Real-time monitoring; STS – Stochastic scheduling; RTC – Real-control

3 Bluetooth Low energy

This work aims to study Bluetooth Low Energy. However, other wireless technologies can support communications. Firstly, this chapter presents an analysis of Wi-Fi, ZigBee, Z-Wave and Bluetooth classic. Afterwards, a full description of how Bluetooth technology works is present in order to understand it and to compare it with a new version of his, Bluetooth Low Energy. Finally, this chapter presents available commercial BLE development platforms to use on the prototype.

3.1 Wireless Technologies

Several Wireless technologies exist that can be used to connect devices that want to control. Table 3.1 presents these technologies and their main characteristics.

ZigBee standard [11], built on top of IEEE 802.15.4, is a relatively simple technology, which uses a protocol of data packets with specific characteristics, being designed to provide flexibility in the types of devices that can control. ZigBee Alliance already is in the home automation market. The most differential characteristic of ZigBee over WiFi and Bluetooth is the capability to support more than 64000 nodes in the same network, beside 8 from Bluetooth and 30 from WiFi as Table 3.1 shows.

Wi-Fi standard [12] refers to any type of IEEE 802.11 Wireless Local Area Network. Wi-Fi makes it possible to deploy networks that connect computers and other compatible devices (smartphones, tablets, game consoles, printers, etc.) that are geographically close, around 100 meters. Wi-Fi Data rate is much higher than Bluetooth’s and ZigBee’s Data Rate, as Table 3.1 shows. It means that Wi-Fi is the fastest wireless technology.

Classic Bluetooth (BT) [13], standardized as IEEE 802.15.1, enables the wireless communication between devices with short-range (about 10 meters). Devices that have to be close, for example MP3 wireless phones can use this technology. One big advantage of Bluetooth is the possibility of use a Smartphone, a tablet or a personal computer to communicate, as happens in Wi-Fi communications.

Zensys develop Z-Wave [14] to home automation systems. Z-Wave uses a small bandwidth to send control and data commands, but do not have support to transmit audio and video. This technology was specifically designed to home automation, being very simple to pair one electrical device with the controller without complicated programming and wireless. Z-Wave is a mesh network, because all nodes can communicate among them, passing the signal along to another until reached the target node. A PC, tablet or smartphone that have an internet connection available also can control this Z-Wave system.

Choose a wireless technology to develop a project depends on the purpose of that project. Depend on the number of devices that will be connecting to a network, where ZigBee is the one that support more devices. It also depends of the transmission, Data Rate, which the project should need, i.e. if it needs to transmit faster and with what frequency. Finally it depends on the Range and battery lifetime that project will need. Battery lifetime means the time that device works without any charge.

The wireless technologies operating in the ISM band does not depend on licenses to operate, and operate at 2.4 GHz.

Table 3.1 - Comparison of wireless technologies

	ZigBee	Bluetooth Classic	WiFi	Z-Wave	Bluetooth 4.0
Promoter	ZigBee Alliance	Bluetooth SIG	WiFi Alliance	Z-Wave Alliance	Bluetooth SIG
Standard	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.11	-	IEEE 802.15.1
Network topology	Mesh, Tree, Star	Star	Star	Mesh	Star
Nodes	>64000	8*	32	232	8*
Frequency band	2.4 GHz	2.4 GHz	2.4 GHz	900 MHz	2.4 GHz
Data Rate	250 kbps	1-3 Mbps	11 Mbps	10 kbps - 40 kbps	1 Mbps
Data Protection	16-bit CRC	16-bit CRC	32-bit CRC	8-bit CRC	16-bit CRC
Range	100 m	10 m	10-100 m	100 m	30 m
Power	Very Low	Low	High	Low	Very low

*Bluetooth support eight nodes per Piconet. However, connecting multiple Piconets, creating Scatternets, the number of nodes is unlimited.

3.2 Bluetooth Technology

Bluetooth technology is a wireless communication technology that is simple, secure and low power. A combination of software and hardware allows transferring data between devices, since they are close. If two devices are close, they mutually detect each other and this proximity increasing the fidelity of data transmission. Classic Bluetooth (BT) uses 79 of 1 MHz channels on the 2.4 GHz ISM band with a pseudo-random frequency hopping sequence. Bluetooth technology is able to paring one Master device with no more than seven Slave devices, performing a Piconet network. However, Slave devices can connect with more than one Master device, creating overlapping Piconets that performs Scatternets. Scatternets allows an unlimited number of interconnect devices because devices from different Piconets exchange data, since there is a path between them, created by intermediate devices. Figure 3.1 helps to understand the difference between Piconet and Scatternet. In Figure 3.1, any device from one Piconet can communicate with all devices from the other Piconet, because there is path created by Bluetooth devices that allows the message arrive to destination. Assuming a slave in a Piconet that wants to communicate with a slave of another Piconet, there are three intermediate devices, which are two masters and one slave that connects with these two masters, which make it possible.

This work focuses on Bluetooth Low Energy, so it is made a technically description of BLE and when it is possible make a comparison with Classic Bluetooth.

Bluetooth Low Energy [15], also known as Bluetooth Smart, appeared in 2011. BLE is not just a revision of classic Bluetooth, is a completely new technology. The purpose of this technology is to decrease power consumption maintaining the same purpose, transfer data between devices. It also enables devices to connect to the internet. The purpose of BLE design is to run, for a very long time, on a coin-cell battery and to be easy to develop, at a cheap price. In Figure 3.1, the slaves communicate with the Masters that are within the same black circle.

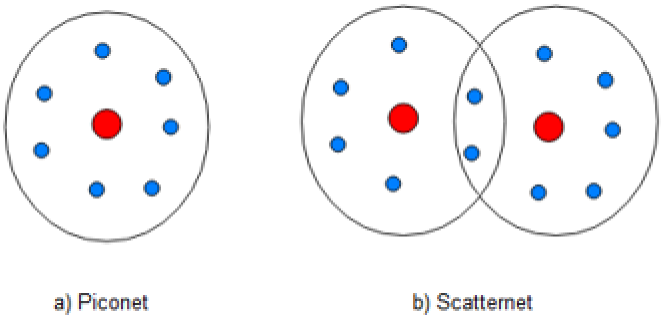

Figure 3.1 - Bluetooth networks topologies.

Radio Frequency

Unlike classic BT, that uses 79 channels on 1 MHz spacing in the 2.4 GHz ISM band, BLE has 40 channels with 2 MHz spacing as depicted in Figure 3.2. The function of the three of 40 channels, represented in green, is to advertise and discover/find other devices. The other 37, represented in blue, permit to transmit data between two paired devices. Both technologies use frequency hopping, despite of BLE use frequency hopping at slower rate, and both use GFSK modulation. Data rate on BLE could be 1 Mbps.

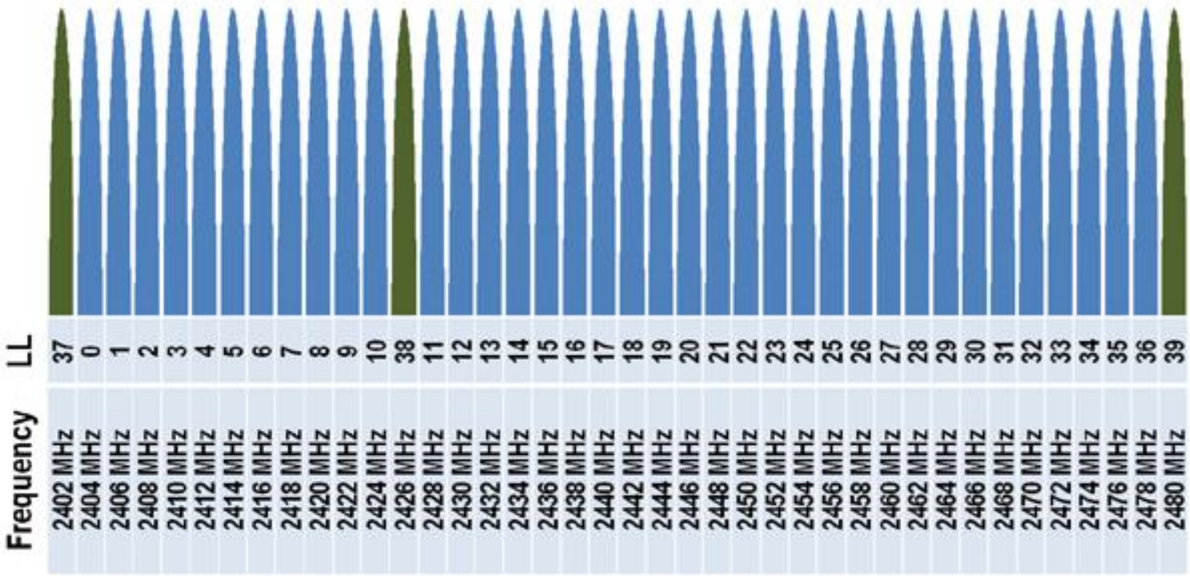

Figure 3.2 - Channels in BLE technology (extracted from [16]).

Modes

BLE devices can operate in a Master or Slave device mode, an advertising mode and a scanning mode. Master and Slave modes allow the device to read, to write and to query each other, since they are connected. Scanning mode is used to capture advertise packets. Advertising mode function is periodically advertising information of possible connections and responding to queries from other devices. When establish connection between two devices, the device that is on advertising mode will assume the Slave mode and the device that is on scanning mode will assume the Master mode.

Packet Format

There are two types of data packets, one for the data and another for the advertising.

Data packets needs to include a preamble (1 byte), access codes (4 bytes), Protocol Data Unit (2-39 bytes) and CRC (3 bytes). A preamble is a signal used in network communications to synchronize transmission timing between two or more systems. By identity issues, RF channel randomly generates access code. Protocol Data Unit (PDU) is the data. Cyclic redundancy check **(**CRC) is an error-detecting code to detect accidental changes to raw data. The data packets can have a 10 Bytes length until 47 Bytes length.

Advertise packets on the other hand, have PDU containing a 2 bytes header and up to 31 bytes of data.

Pairing

Pairing is the action that defines a connection between two devices. In order to pairing, in Bluetooth technology, a device has to ask another if it wants connects with him. Therefore, on one hand, the advertising device transmit packets on the advertise channels with PDU containing the device address and some additional information. On the other hand, the scan device is able to see the advertiser address and some additional information. When the connection is established, scanning device supply the advertiser connection interval and slave latency. Slave latency is the number of connections intervals that a slave can ignore without losing the connections. Connection interval determines the start of connection events, which is the action of exchange data packets.

Note that before establishing connection, it is possible to obtain many of information about de Slave device. It is useful, for example, to know the distance between two devices.

Data transfer

After pairing, the communication is support by remain 37 channels. Master device starts the communication event and alternates between Master and Slave devices until one of them stop transmitting. The PDU's have to up to 37 bytes of payload, together with packet header, and a message integrity check of 4 bytes.

Data Protocol Stack

Data Protocol Stack has two independent pieces: Controller and Host. The controller (Link Layer controller) runs in lower levels of the stack and it is responsible to handle physical layer packets and all associated time. Controller can also have the same function of a firewall, filtering data packets. On the other hand, the Host handles the upper levels of the stack and it are included the application, attribute protocol and L2CAP. Usually, the Controller and the Host used Hardware Controller Interface (HCI) to communicate. That communication can be over the Universal Synchronous Receiver/Transmitter (UART), Universal Serial Bus (USB) or Secure Digital I/O (SDIO) physical layer interface. However, the Controller and the Host can be in the same microcontroller, and in that case, HCI is not need. Figure 3.3 presents BLE Stack architecture of a Peripheral device.

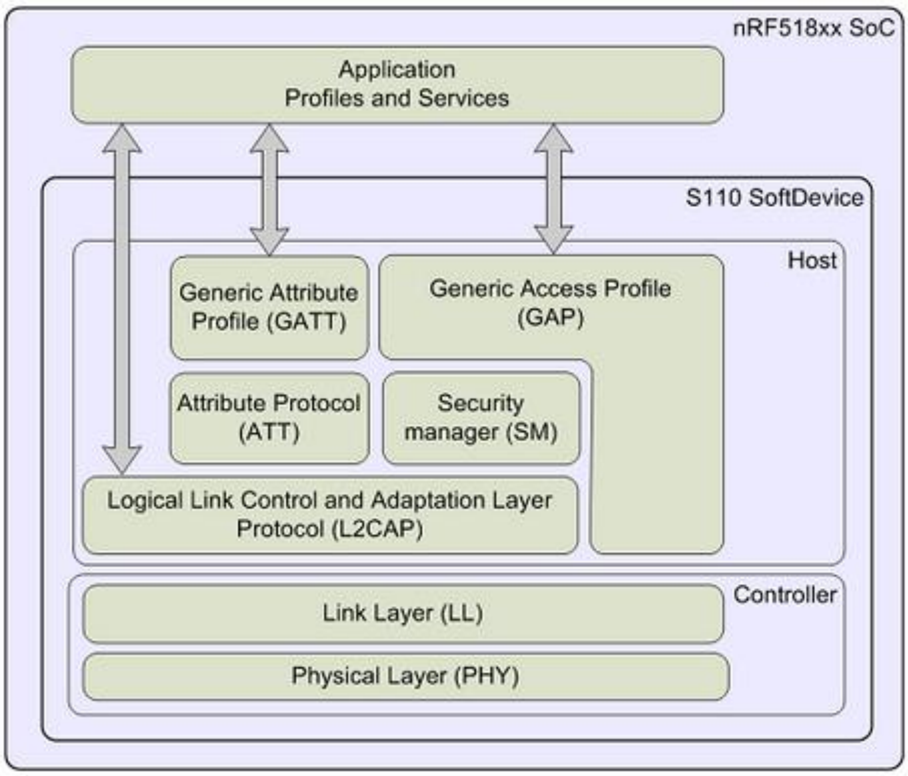

Figure 3.3 – BLE Peripheral Stack architecture (extracted from [17]).

Logical Link Control and Adaptation Layer (**L2CAP**) is the protocol that allows the communication between the Host and with the Controller via HCI, or directly if there is no HCI. L2CAP function is to provide data packets in the form that Controller and Host needs to receive because Controller need to receive fragmented packets and the Host receive entire packets. Therefore, the main function of L2CAP is group and ungroup data packets.

The Generic Access Profile (**GAP**) is responsible for defining generic procedures used in the pairing and linking of the device. It is the interface for the application layer to implement the different Bluetooth modes (Advertising, Scanning, etc.).

The Security Manager (**SM**) is responsible for authentication and encryption and it uses an AES-128 bit encryption. It is also influence the pairing.

The Attribute protocol (**ATT**) is a communication method designed to optimize transmission of small packets. ATT are pairs of attributes and values used to read, write, or discover by other devices.

The Generic Attribute Profile (**GATT**) is an extension to ATT and its main function is describes the different service frameworks and communicate with application layer through application profiles. Each application profile defines data formatting and how application should interprets it. GATT uses the ATT as its transport protocol to exchange data between devices. Sections called **services**, which group conceptually related pieces of user data called characteristics, organizes hierarchically this data. It is responsible of GATT to define device role, which is Server or Client. GATT client requests to a server and receive responses, since it does not know anything in advance about server's attributes. To discover the server's attributes the GATT client perform a service discovery. GATT Server is responsible to send responses to GATT Client requests and store organized data in attributes to make this data available for GATT client.

Range

Range of Bluetooth classic is a power-class-dependent divided in three classes:

- Class 1: Range of 100 meters with a maximum power of 100 mW.
- Class 2: Range of 10 meters with a maximum power of 2.5 mW.
- Class 3: Range of 1 meter with a maximum power of 1 mW.

Devices of different classes can connect to devices of other classes, since the range of those classes is satisfied. Range of BLE is about 30 m, in perfect environment conditions, but there is no limit imposed by the Specification.

Security

Frequency hopping helps to force security, because changing the channel that communicates difficult to capture data by an attacker.

BLE use an encryption of 128-bit AES, which has shown to be unfeasible to crack using a brute force attack. Finally, a device can require a PIN or a Paraphrase at the time of establishing a connection to establish the pairing. The same happens in Classic Bluetooth.

Low Energy

BLE has three main characteristics, which classic Bluetooth do not have, that influence the power consumption. Firstly, BLE uses a lower duty cycle, which allows a device to pass more time sleeping and wake up with less frequency to send and receive packets. Secondly, BLE only connects when there is data to send or receive, otherwise, BLE it goes sleep. Unlike classic Bluetooth which is connected, even if no data to transfer. Lastly, but no least important, using GATT profiles it is able to send small packets in each connection. It means that BLE send a portion of packet, goes to sleep and send another portion of the same packet, and repeat that cycle until all the packet is sent. It is save power because, according to Table 3.2, the transmit time on BLE is lower than on classic version and despite have to connect several times, the peak current consumption is slower too.

BT and BLE are fundamentally different, so there are two options for implementation, a single-mode and a dual-mode. In a single-mode, devices have only BLE. The design of these devices has prepared to operate with low power consumption, so it is not possible to run classic BT on them. An example of a single-mode device is a heart rate sensor. In other way, dual-mode devices included both BT and BLE. These devices are not so good in terms of power consumption because of the implementation of these two technologies. Nevertheless, these devices provide the advantages of BT and BLE. An example of dual-mode device is a mobile phone.

Table 3.2 - Technical Specifications of Classic Bluetooth and BLE.

Technical Specifications	Classic Bluetooth	Bluetooth Low Energy
Modulation Technique	Frequency Hopping	Frequency Hopping
Frequency Hopping	2400 to 2483.5 MHz	2400 to 2483.5 MHz
Modulation Scheme	GFSK	GFSK
# Channels	79	40
Channels Bandwidth	1 MHz	2 MHz
Nominal Data Rate	1-3 Mbps	1 Mbps
# Nodes	8*	8*
Security	56/128 bit	128 bit AES
Range	100 m	30 m
Total Time to send data	10 ms	3 ms
Network topology	Scatternet	Scatternet
Relative power consumption	100%	10 - 50 %
Peak Current	< 30mA	< 20 mA

*Bluetooth support eight nodes per Piconet. However, connecting multiple Piconets, creating Scatternets, the number of nodes is unlimited.

BLE has some limitations because of it low power consumption. Data Transfer Rates on BT can be 3 Mbps but on BLE is lower than 1 Mbps. Means that BLE will lose its advantage of power consumption when has to perform continuous transmission. BLE is not design to stream large amounts of data. Therefore, BLE is not better than BT and otherwise. Choose of Bluetooth technology depends on the application.

3.3 Available commercial BLE development platforms

This subsection introduces some specific development platforms for designing and prototyping BLE devices. Since Smartphone integrates BLE firmware, BLE chipsets opens a new world of controlling devices and power the IoT. Larger companies in electronics area, like Nordic Semiconductor and Texas Instrument, believes on the future of BLE technology, which enables a new and extended range of applications to benefit from BLE wireless technology as watches, proximity tags, sports and fitness sensors, healthcare sensors and remote controls. The goal is choose an inexpensive and reasonable development platform to support BLE communications on desired workable prototype.

This subsection describe some BLE chipsets developed by Nordic Semiconductor and Texas Instrument. It presents a comparison between these BLE chipsets, considering features, Hardware and Software required to program this chipsets and the cost of purchase everything to start design and prototyping BLE devices.

nRF51 DK (Nordic Semiconductors)

Nordic’s nRF5182 [18] is a highly integrated system-on-chip (SoC), which include BLE-compatible radio and an ARM Cortex-M0 core running at 16MHz with 128/256 kB of flash memory and 16kB of RAM.

Being an entirely flash-based device is the strongest factor to use nRF51822 in new BLE designs. It means that BLE stack integrates the flash memory, allowing updates, without necessarily requiring a new chip revision/device. In order to implement BLE support on its chip, Nordic uses SoftDevice, which occupies the bottom of memory and implements features such as the BLE stack and peripheral/central roles. To evaluate Nordic nRF51822 SoC the best platform to begin is Nordic’s nRF51 DK [19] (development kit). The nRF51 DK is a single-board development kit, presented on Figure 3.4 a), for Bluetooth Smart, ANT and 2.4GHz proprietary applications using the nRF51822 SoC. The kit includes five nrf51822 SoC samples, one board and one 2032 battery. The board offers all pins available on nRF51822, enables an I2C or SPI connection with sensors or peripherals, talks to other devices over UART, etc. This board also includes a J-Link on board to program and debug the nRF51822, as well as additional circuitry to measure power consumption and make prototyping and debugging easier. The board includes a USB power receiver and include a CR2032 battery holder to allow powering board with a small battery. To debugging, the nRF51 DK can work with a Bluetooth Smart Sniffer [20], presented on Figure 3.4 b), using Nordic’s nRF Sniffer Software, which allows capturing and analysis data packets over the air. Using the Master Emulator firmware and the Master Control Panel Software enables simulation central device role and test connection and functionality of developed application.

Nordic provides the Nordic SDK (Software development kit), which a broad selection of drivers, libraries, SoftDevices and radio protocols, .To develop code and debug application it uses Keil MDK-ARM and J-link Software, respectively. Registering board’s uniquely identifier number on Nordic’s company allows free access to necessary software. Finally, Nordic’s provide nRFgo Studio Software to flash SoftDevice and application code to the chip.

Total cost is approx. **105€** (include nRF DK and Bluetooth Smart Sniffer)**.**

a) b)

Figure 3.4 - Nordic Semiconductor's nRF51 DK (adapted from [19] and [20]).

CC2541 Mini DK (Texas Instrument)

Texas Instrument (TI) has design a number of BLE SoC to achieve BLE solutions, as the CC2541 [21]. CC2541 is a SoC that combines an 8051 core with a 2.4GHz radio. About memory, this chip includes a 128/256 kB of flash memory and 8kB of RAM. TI has some advantage over rivals because its BLE stack is feature complete, essentially covering the entire 4.0 version of the Bluetooth Core Specification, whereas some rivals choose to not implement certain infrequently used options features. The main disadvantage of the CC2541 is to require the IAR Embedded Workbench that is too expensive. To program the CC2541, the best platform is the low-cost CC2541 Mini DK [22], presented in Figure 3.5, (DK means Development Kit). This kit includes all the Hardware to start working with CC2541, including a Hardware debugger, a USB dongle that can run with central role, and a key fob development board that can run custom BLE code. Texas instrument provide a free 30-day trial of IAR, because licence to download IAR Embedded Workbench cost around 3500€.

Total cost is approx. **3600€** (include mini DK plus a license to download IAR embedded Workbench).

Figure 3.5 - Texas Instruments' CC2541 Mini DK [22].

Waveshare nRF51822 EK

This is an evaluation/development kit design for nRF51822, which include a motherboard BLE400 [23] and the wireless module Core51822, presented on Figure 3.6. This core includes the nRF51822 SoC and onboard 2.4GHz antenna. The motherboard allows programming the nRF51822 trhough J-link debugger. This kit is very similar to Nordic's nRF51 DK, however this kit include the Core51822 module that can work alone without the board connection. This decreases a lot the space required to incorporate BLE communication on sensors or appliances, comparison with Nordic's nRF51 DK. This board provides UART, SPI and I2C interfaces, powering a whole range of end-use applications. As the chipset is nRF51822, it is compatible with Nordic SDK and nRFgo Studio. To develop code and debug application it uses Keil MDK-ARM and J-link Software, respectively. Keil MDK-ARM provides a free limited version that allows code size until 32kB. To develop prototype, 32kB is enough. After purchase J-link hardware, the J-link Software is free.

One big disadvantage of this evaluation kit is the debugging, because the kit does not provide a USB dongle to simulating a central device neither a device to push data out to Wireshark.

Total price is approx. **80€** (BLE400 + 2 Core51822 + J-link EDU)

Figure 3.6 - Waveshare's nRF51822 Eval Kit (extracted from [23]).

nRF8001 Arduino Shield

Nordic's nRF8001 [24] is a highly integrated single-chip BLE connectivity integrated circuit (IC). It is very similar to nRF51822 about BLE features, except this chip only support peripheral role. RedBearLab design a low-cost BLE Arduino Shield V2.1 [25], presented in Figure 3.7 that include both 2.4GHZ antenna and nRF8001 chipset, which support the features of BLE technology. This Shield connects to an ordinary Arduino board, such as Uno, Due, Mega2560 and Leonardo, creating a BLE-compatible radio and powerful microcontroller. The free Software to program this Shield is the same that programs Arduino boards, which is free open source Arduino IDE provided by Arduino. Then, it becomes a good solution, to works as peripheral device, because the shield combines a powerful BLE chipset with a powerful microcontroller, at low cost. Total cost is approximate **40€** (include Arduino Uno board and BLE Shield v2.1).

Both nRF51822 and CC2541 chipsets satisfy the requirements to perform functions of central and peripheral devices. However, the software to program CC2541 is too expensive, therefore, the option to use the CC2541 chipset is discard. If the size is not a problem, BLE Arduino Shield V2.1 is the best solution to support peripheral role, such as sensors. Furthermore, Arduino provide analogue and digital I/O, UART and SPI interface, etc., which increase the whole range of end-use applications.

The suggested prototype also requires a central device to control peripheral devices inside the house. Therefore, both nRF51 DK and Waveshare nRF51822 Eval kit are possible solutions to support central role. The nRF51 DK provide an easy way to debug application and way to analyse exchanged packets between BLE devices. However, for each peripheral or central device it needs a new nRF51 DK, which is a more expensive solution that Waveshare nRF51822 Eval kit that only needs a new Core51822 to add another BLE device, either central or peripheral, to home network. A new Core51822 costs less than 10€. Concluding, nRF51 DK provide better tools for designing and prototyping BLE devices, while Waveshare nRF51822 Eval kit is cheaper.

Figure 3.7 - RedBearLab Arduino BLE Shield V2.1 (extracted from [25]).

Considering end-use application, which is a demonstration of a Home Automation system, Waveshare nRF51822 Eval kit with several core51822 seems to be the better solution. Table 3.3 shows the main characteristics of the present BLE chipsets in the above presented platforms.

Table 3.3 - BLE chipsets characteristics.

	CC2541	nRF51822	nRF8001
Mode	Single Mode v4.0	Single Mode v4.1 / ANT	Single Mode v4.0
Processor	8051	Cortex-M0	-
Flash	256kB	128kB/256kB	-
RAM	8kB	16kB/32kB	-
Current Consumption (RX/TX)	17.9mA / 18.2mA	9.7mA / 8mA	14.6mA / 12.7mA
Average Current 1s / 4s Connection interval	24µA / 6.8µA	15.5µA / 5.6µA	-
Price*	4.79 €	4.65 €	4.28 €

* The price is for a single purchase unit using the prices present in the distributor's website *Mouser Electronics.*

Considering local controllers, which have to process data, a chipset with built-in processor has more advantage. The current consumption is an important parameter to consider, because local controllers probably are powered with batteries. Therefore, the nRF51822 chipset presents a processor and it presents the lowest current consumption. Although, nRF51822 has more RAM to store the application.

4 Proposed HEM System

This chapter presents the proposed HEM system, describing system's architecture and requirements, and the implemented prototype, describing Hardware and Software. The proposed system is a solution for Home Automation with Energy Management, where BLE supports the communications inside the house.

4.1 Architecture

The architecture of proposed HEM system, presented in Figure 4.1, has a main controller, sensors, router, Smart Meter and two types of electrical appliances classified as critical and variable appliances. This proposal assumes that electrical appliances and sensors includes an internal load controller, with BLE support, to communicate with main controller.

Critical appliances are appliances that working when the homeowner decides, like lights, microwave, oven and coffee maker, and appliances always working, like the fridge. The variable appliances are appliances that do not have a fixed period of the day when they have to work, such as HVAC, water heater, dishwasher, clothes washer, etc. These appliances allows managing energy usage, in order to reduce energy consumption during the peak-loads. Although, there are variable appliances that cannot turn off since when they are on, such as dishwashers. Figure 4.2 shows some examples of critical and variable loads.

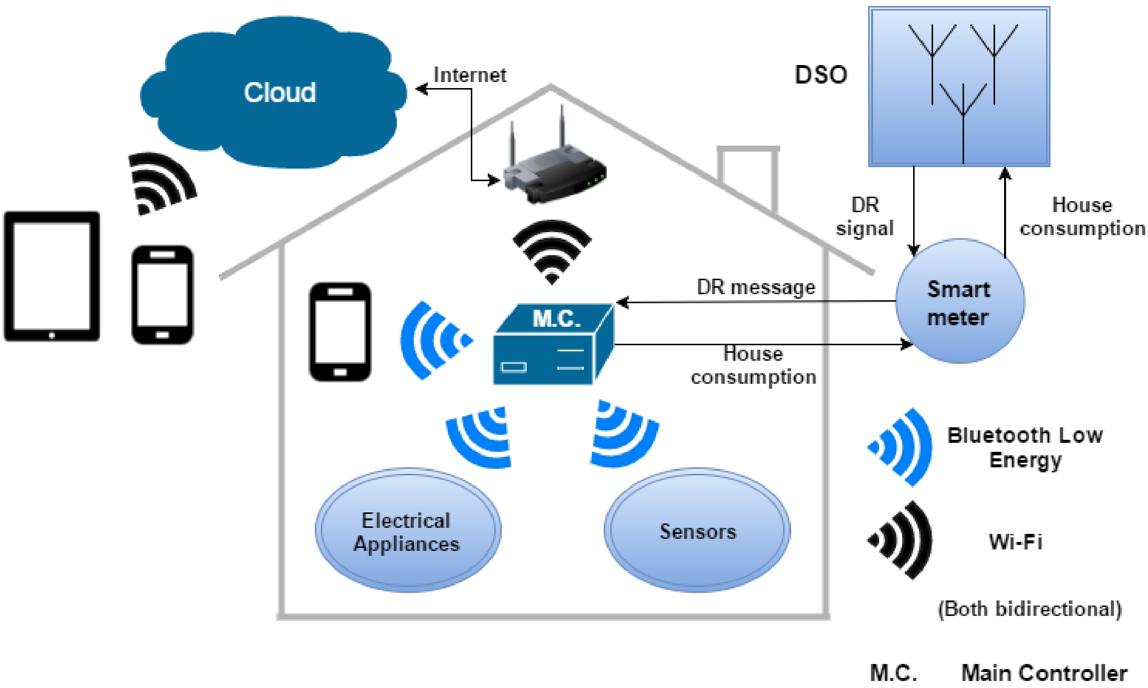

Figure 4.1 - Proposed HEM system architecture.

The main controller, which is the "brain" of the house, collect data from sensors, collect current consumption of entire house, which it is not present on architecture, collect current consumption of variable appliances and it actuates over both critical and variable loads. However, it just manages the time of work of variable loads. The main controller supports BLE.

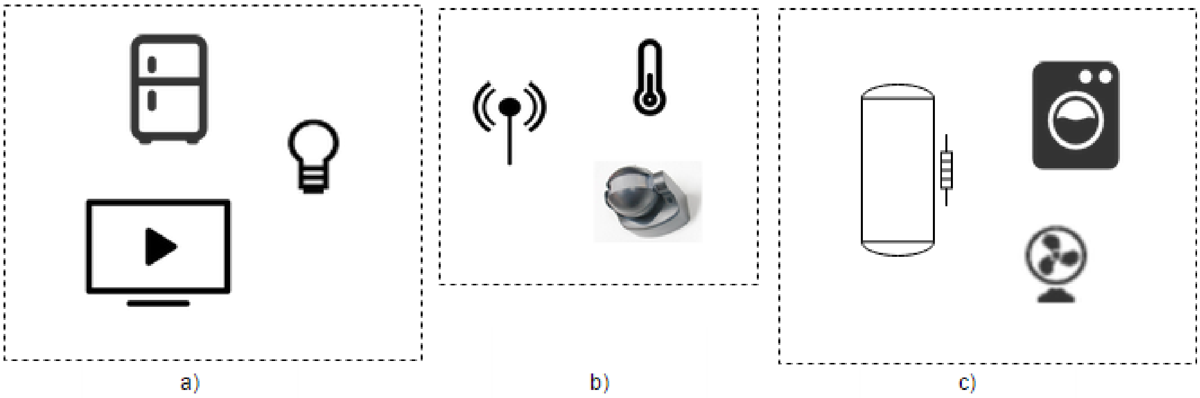

Figure 4.2 – Examples of loads. a)Critical loads; b)Sensors; c) variable loads.

Load controller is responsible to collect data from sensors, including current consumption of variable appliances, and/or to control the respective electrical appliance, based on control signals sent by main controller. Load controller sends to main controller the data collected and responses to control signals.

The smart meter is part of the HEM system. The smart meter receives a message from DSO, containing the real-time price of electricity or requests for decreasing the consumption. Smart meter sends to main controller that adjusts the schedule of variable loads, based on priorities defined by the homeowner. Smart meter also send data to the utility about the house energy consumption.

Architecture described in Figure 4.1 extends the HEM system control to the outside world. It includes a Wi-Fi connection to the home router, which makes the HEM unit addressable, through an IP address. Wi-Fi is obviously more convenient than wired Ethernet cables, mainly because the mobility. The proposed system also allows to homeowner interact directly with main controller, through a BLE App, running on smartphone. This range of possibilities increase significantly the comfort of homeowner, since he can access the system outside of the house and inside, even if internet connection fails.

As described in Table 3.2, the range of BLE is about 30m in open space. Obstacles, like walls, inside the house, decrease the range to less than 15m. Therefore, each room includes a local controller, creating Scatternets.

4.2 Requirements

This subsection presents the functional requirements for the architecture presented in Figure 4.1:

1. The DSO:
   - Provides DR information to smart meter.
   - Provides electricity.
   - Monitor houses power consumption.
2. The Smart meter:
   - Process and provide the real-time price of electricity to main controller.

3. The main controller controls smart appliances, monitors sensors and controls the distribution of Energy usage. The main controller:
   - Distributes the usage of electricity to appliances that it controls, communicating with load controllers.
   - Establish a Wi-Fi connection with router.
   - Establish a connection with a smartphone using BLE (allows user intervention inside the house).
   - Provides all necessary information to be presented in the user interfaces.
   - Collects data from sensors.
   - Collects data from appliances, such as current consumption.
   - Provides real-time consumption to smart meter.
   - Collects DR message from smart meter.
   - Accepts homeowner's settings.
   - Creates schedule to switch on/off electrical appliances.
   - Collects from local controllers the entire house real-time electrical consumption.
4. The load controller control electrical appliances and monitors sensors. It:
   - Turn ON and OFF their respective appliance is ON and OFF, based on commands given for main controller.
   - Collects real-time electrical consumption.
   - Communicates via BLE with main controller.
5. The router provide an IP address to the HEM system. It:
   - Transfers data from main controller to user interface.
   - Transfers data from user interface to main controller.
6. The smartphone allows access to the HEM system through BLE Application.
   - Accessing to home system from anywhere, with internet connection.

4.3 Prototype implementation

This section describes the Hardware and Software developed, as well as the communications between: the main controller and the load controller, the main controller and the webserver, the load controller and an electrical appliance, the load controllers and the sensors and, finally, the user and the user interface. The prototype demonstrates a direct control over the Smart Plug, an automatic control over environmental factors and a security system.

4.3.1 Adapted architecture to prototype implementation

The developed prototype does not include the smart meter and the connection to the DSO proposed for a complete HEM system, as presented in Figure 4.3. It implements a BLE communication between the main controller and home appliances, in order to control the electrical appliance and monitor sensors. It also implement an Ethernet connection between main controller and webserver, in order to send and receive data from it.

The prototype includes a main controller, three load controllers, one smart plug capable to switch ON/OFF an electrical appliance and two sensors. It provides a webserver in the Cloud and a web user interface to collect data from sensors and to control an electrical appliance. Each main and load controllers link to BLE module, to support BLE communications.

Figure 4.3 presents an Ethernet connection between main controller and router, instead of Wi-Fi connection. This prototype allows the main controller to be near by the home router. Then, for a cost reasons, the Ethernet connection replaces the Wi-Fi connection.

Prototype implements a continuous connection between main controllers and load controllers to increase the transmission performance, decreasing transmission latency. Considering this option, the access to Home Automation is always through the main controller. The available smartphone acts as a Master device in a BLE connection, once it runs over Android 4.4 version. Android announces that versions higher or equal than 5.0 allow acting as Slave in a BLE connection [26]. Considering BLE protocol, which it does not allow a communication between two Master devices, it is impossible to smartphone communicate over BLE with master controller.

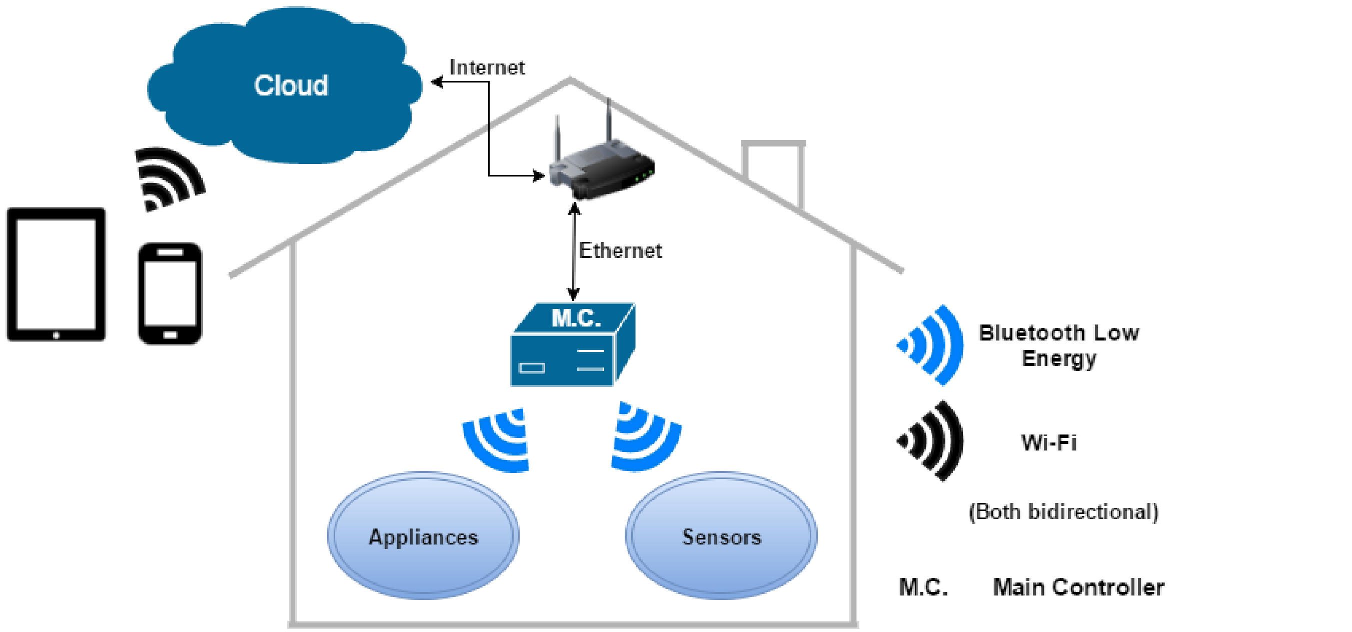

Figure 4.3 – Prototype top-level architecture.

4.3.2 BLE protocol

BLE supports communications inside the house. There is the concept of Central/Peripheral, which refers to establish a connection, also known as GAP role. A Peripheral advertises to inform the Centrals that it is available to start a connection, although it is only a Central that can actually send a connection request to establish a connection (pairing). When pairing occurs, the Central is Master and the Peripheral is Slave. There also is a concept of Client/Server, also known as GATT role. The GATT Client is the device that accesses data on the remote GATT Server via read, write, notify, or indicate operations, while GATT Server is a device that stores data locally and provides data access methods to a remote GATT Client. As explained in 3.2, one Master can connects with seven Slaves at a time. Therefore, the main controller links to a Central and load controllers links to Peripherals.

BLE peripheral device

Peripheral starts advertising and wait for a connection request. As soon as it receives a connection request, the connection starts. When establishing a connection, the Peripheral will suggest a connection interval to the Central, and the Central device will try to reconnect (connection event) every connection interval to see if any new data is available. Between connection events, the radio of the Peripheral is inactive and the Peripheral enters in a low power mode to save power. The only thing normally running in between connection events is the RTC0 timer and the 32 kHz low frequency clock that is required to operate the RTC0.

Peripheral organizes data hierarchically, through services and characteristics, as explained in 3.2. It makes use of Attribute Protocol (ATT), which is used to store Services, Characteristics and related data in a simple lookup table using 16-bit IDs, called handles, for each entry in the table. Therefore, GATT Client may use these handles to reference, access, and modify it. Every characteristic has one main attribute that allows access to the actual value stored in the database for that characteristic.

This prototype implements two GATT servers with one service and two characteristics, one for keep the current state of appliance or the actual measures of sensors and another to receive orders from GATT client, such as update the sensor measures or switch on and off the electrical appliance.

The UUID is a 128-bit number that identify the Peripherals and allows to Central recognize them.

The UUID base used for that application is 0000xxxx-1212-efde-1523-785feabcd123, where the set of “x” are handles that represents different attributes. Set of “x” is 1523 to identify service, is 1524 to identify read characteristic and is 1525 to identify write characteristic. It is equal to both BLE peripheral devices.

BLE central device

Central starts scanning and send a connection request to an advertising Peripheral to starts a connection. After connecting, the Central discovers Peripheral services and, if services’ UUID matches with services’ UUID that it expects, it creates a client for each Peripheral, storing the discovered services. Then, the Central search for characteristics’ UUID and, if matches with characteristics’ UUID that it expects, it enables the client characteristic configuration descriptor (CCCD), which enables notifications about that characteristic.

4.3.3 Hardware

The sections bellow describes the hardware to develop the prototype. The prototype includes a main controller, three load controllers, two for the sensors and the other for the electrical appliance, and BLE modules to support communications between controllers. Figure 4.4 presents the complete prototype’s architecture. It includes specific Hardware used to build a workable prototype and the types and directions of connections between devices.

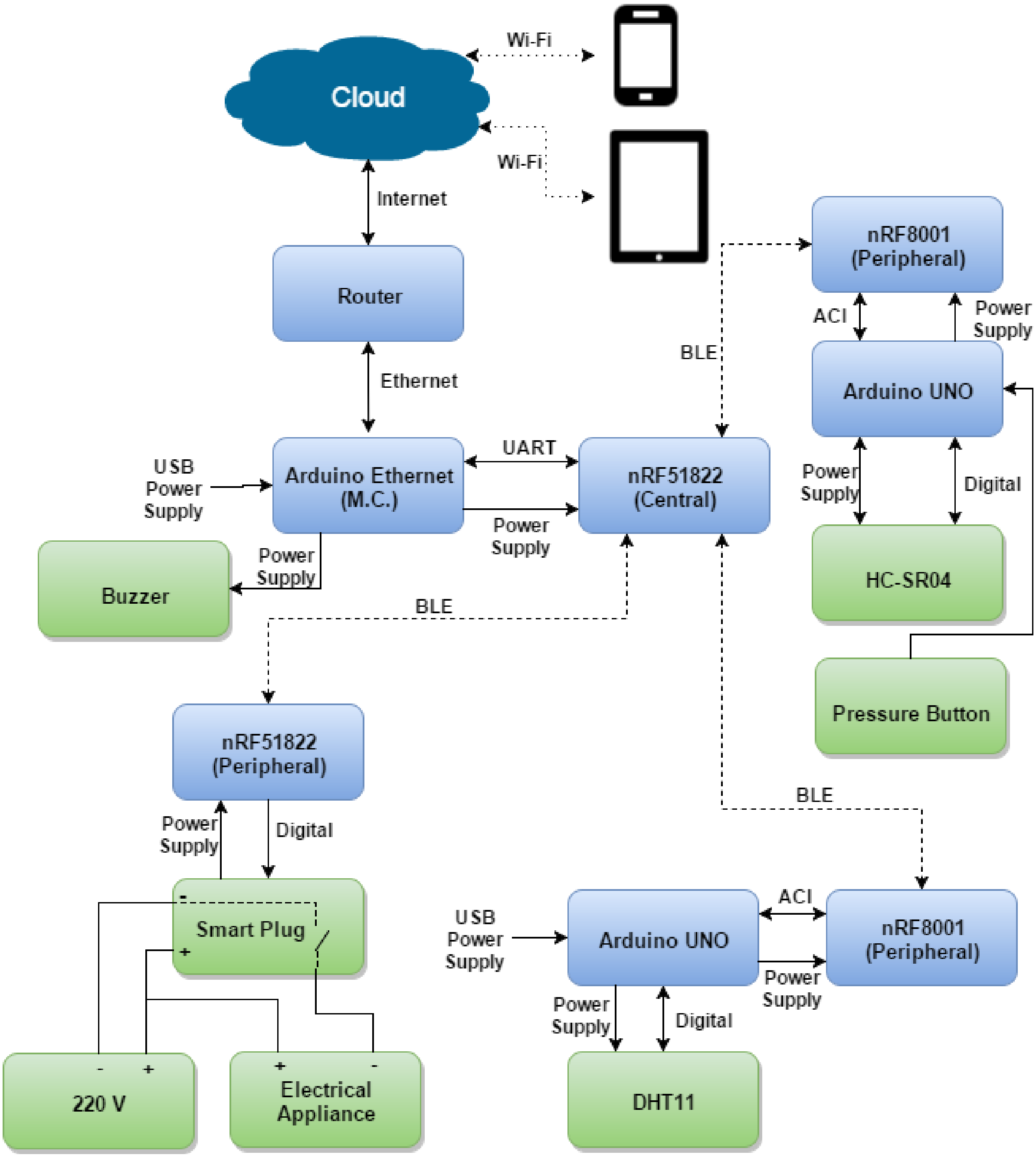

Figure 4.4 - HA System Prototype.

BLE modules

This demonstration uses two types of BLE modules: **Core51822** [23] and **BLE Shield V2.1** [25], as suggested on 3.3. The demonstration uses two different BLE modules to achieve better results of BLE tecnhology. Core51822, presented on Figure 4.5, which supports Peripheral and Central roles, links to main controller and to the load controller that controls electrical appliance, because of its small size. The BLE Shield, presented on Figure 4.6, which only support Peripheral role, links to load controller that collect data from sensors. As described in 3.3, to program and debug Core51822 it is need the motherboard BLE400 and J-link EDU, and to program BLE Shield it is need an Arduino board.

As described in 3.3, nrf51822 is built around a 32-bit ARM Cortex M0 CPU with 256kB flash plus 32kB RAM, which can replace one load controller, maintaining BLE functionality.

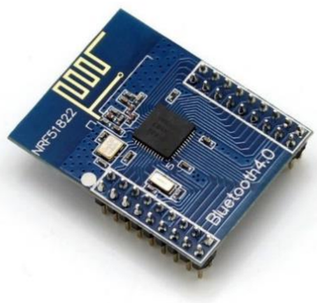

Figure 4.5 - Core51822 Bluetooth 2.4GHz RF module (extracted from [23]).

The physical ACI interface on nRF8001 consists of five pins, MISO, MOSI, SCK, REQN, and RDYN. MISO (Slave->Master) and MOSI (Master->Slave) pins exchange data between Arduino Uno (Master) and nRF8001 (Slave). SCK pin provides the clock generated by Arduino board (Master). The pins REQN and RDYN, controlled by Master and Slave, respectively, determine the permissions of write and read.

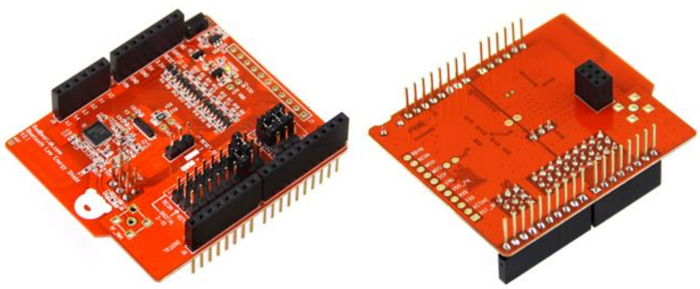

Figure 4.6 - Arduino BLE Shield V2.1 (extracted from [25]).

Load Controllers

There are three load controllers, one to control an electrical appliance and the others to collect data from sensors. One is the ARM Cortex-M0 CPU included on Core51822 and the others are Arduinos Uno [27].The Arduino Uno is microcontroller board based on the ATmega328, which has with 32kB of flash memory. The Arduino provide 14 digital I/O pins and 6 analogue input pins. The ATMega328 also support UART, SPI and I2C interfaces.

Main Controller

The main controller is an Arduino Ethernet [28]. Arduino Ethernet, like Arduino Uno, is microcontroller board based on ATmega328. Arduino Ethernet also support micro SD card, which can hold an invariant HTML page, and it has a RJ45 connector to provide Ethernet connection.

Sensors

DHT11 [29] sensor measure Temperature and Humidity and produce values as a 40-bit digital output.

HC-SR04 [30] sensor is an ultrasonic sensor that provides 2-400 cm non-contact measurement function with 3mm of ranging accuracy. This sensor sends out an 8-cycle burst of ultrasound at 40 kHz and raise its echo. The sensor measures the time between sending trigger signal and receiving echo signal and produces a digital output. This time multiplied by the speed of sound and divided by 2 is the distance between the sensor and the object.

Smart Plug

Smart Plug turns on and off any electrical appliance, receiving wireless control signals. This is possible because Smart Plug includes a relay, which is an electrically operated switch that open and close circuits electromechanically or electronically, and a RF receiver, which decode wireless commands and actuate over relay. Relays control one electrical circuit by opening and closing contacts in another circuit.

Studying a commercial remote electrical socket [31], which works with 433MHz RF technology, permits to build a new one with BLE tecnhology. This remote electrical socket receive messages from a 433MHz transceiver that switches on and off the electric socket. Figure 4.7 presents the block diagram of electrical circuit part responsible for switch relay, and consequently, switch the electrical appliance.

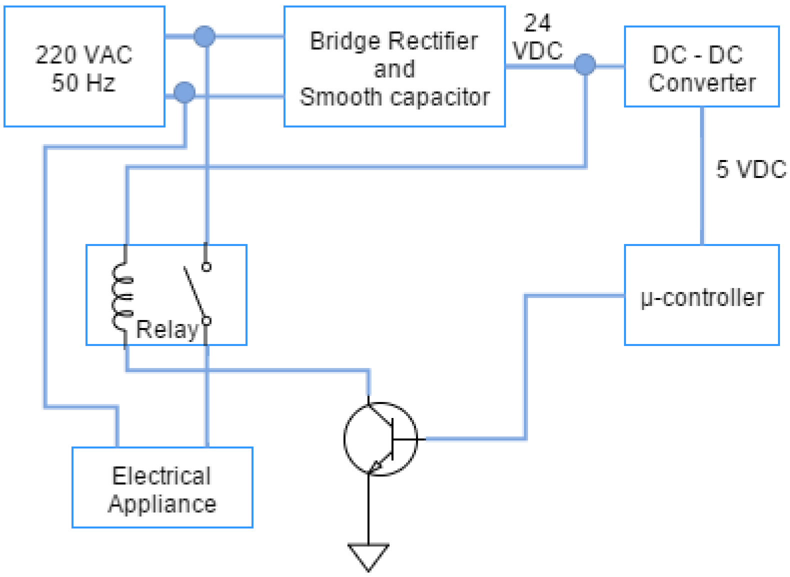

Figure 4.7 - Diagram block of Smart Plug

The 220V AC source powers the circuit. The circuit divides in two parts, one for powering the electrical appliance and another for powering the relay and the load controller. To power the electrical appliance, one terminal of supplier links directly to one terminal of load and another terminal of supplier links to relay switch. Then, when relay close the switch and, consequently, close the circuit, the electrical appliance receives current to switch on. Relay works with 24V DC and the load controller works with 5V DC. Therefore, the circuit needs to convert 220V AC voltage to 24V DC voltage and convert 24V DC to

5V DC. To achieve 24V DC, the circuit has a full-wave bridge rectifier (diode bridge), which impose current to flow in the same direction. Afterwards, the smoothing capacitor converts the full-wave rippled output of the rectifier into a smooth DC output voltage [32]. Finally, the circuit has a DC-DC converter to convert 24V DC to 5V DC.

After powering relay, when an electric current passes through the coil it generates a magnetic field that activates the contact and, consequently, closes the circuit that powers the load. This electric current only flows if the transistor conducts. The transistor conducts if the voltage between the base and the emitter is higher than 0.7V (threshold voltage). Load controller is responsible to controls the transistor. Thus, the load controller is able to switch the appliance.

Finally, it was removed the part of circuit responsible for communication and it is add the Core51822. The circuit powers the Core51822 with 5V and one output pin of Core51822 links to transistor’s base. The Core51822 is inside the Smart Plug box, as Figure 4.8 shows.

Figure 4.8 – Adapted Smart Plug with Core51822.

The prototype Hardware cost around 130€. However, this cost significantly reduced by replacing load controllers to microcontrollers built in printed circuits. This alternative solution reduces about 60% of total cost. For massive production, the cost reduces about 80% of initial cost. Considering massive production, and increase the prototype features, this system is cheaper than presented system on 2.5.

4.3.4 Software

This section describe the three parts of Software to design the prototype, which are the used software tools, the developed software and the firmware.

4.3.4.1 Software tools

The software tools to develop the prototype are Arduino IDE, Keil MDK-ARM, Nordic SDK, nRFgo Studio, Evothings Studio and Apache Cordova.

Arduino IDE, the open-source Software provided by Arduino, was used to write an upload code to the Arduino's boards and Shields. To design a HTLM page, which accomplish the targets of this prototype, the Arduino Ethernets includes the Ethernet library, provided by Arduino.

The Ethernet library is possible to design a simple HTLM page that accomplish the targets of this prototype.

Keil MDK-ARM was used to program and debug ARM Cortex-M0 controllers in Core51822 module. Keil MDK includes the µVision IDE/Debugger, ARM C/C++ Compiler, and essential middleware components. It was used the S110 SoftDevice [33] and the SoftDevice S120 [34], a precompiled, linked binary software implementing BLE technology, which provides an Application Programming Interface (API) to create Peripheral and Central applications, respectively. To flash SoftDevice is necessary another Software, provided by Nordic Semiconductor, nRFgo Studio. However, the chose board, BLE400, was not a Nordic Semiconductor board and, for that reason, some functionalities of nRFgo Studio are not available, but it is able to erase and flash SoftDevice as expected.

Nordic Software Development Kit (SDK) provides drivers, libraries, SoftDevices and examples, which are helpful to start designing. Drivers are programs responsible for communication between the computer operating system and the development boards. Libraries are files containing functions to implement on main code.

Concluding, nRFgo Studio flash S110/S120 SoftDevice, according if the application to program is peripheral or central, respectively, and Keil MDK flash the application code and allows debugging it. Core51822 uses SDK version 6.1.0 and S110 or S120 SoftDevices version 7.0.0 and 1.0.0, respectively.

Software Evothings Studio was used to develop a BLE App. Evothings Studio is a set of tools to develop mobile Apps using web technologies. Evothings Workbench is an integrated development environment, which runs on windows, where it is possible to upload the JavaScript code, developed in any text editor. Evothings Client is a mobile App that allows connecting, wire or wirelessly, with Evothings Workbench. After connecting Evothings Client to Evothings Workbench and uploading App code to Evothings Workbench, it is possible to run the App on Smartphone. Generally, the Smartphone runs an App, which is BLE App, over another App, which is Evothings Client. Afterwards, every time that JavaScript code changes, the App instantly reloads and it is possible to directly see the result. The Evothings Studio also provides examples that are starting points to develop an App. One of these examples was the support to configure the BLE android App used in this prototype.

Apache Cordova is an open-source mobile development framework. Apache Cordova uses JavaScript content developed in Evothings Studio and convert it in a native Application.

4.3.4.2 Developed software

The software includes Arduino's boards programming, nRF51822 programming, nRF8001 programming, HTML page development and BLE Application configuration.

Arduino Ethernet programming

The Arduino Ethernet is also responsible to create the webserver. Figure 4.9 presents a flowchart that represents the code programmed on Arduino Ethernet.

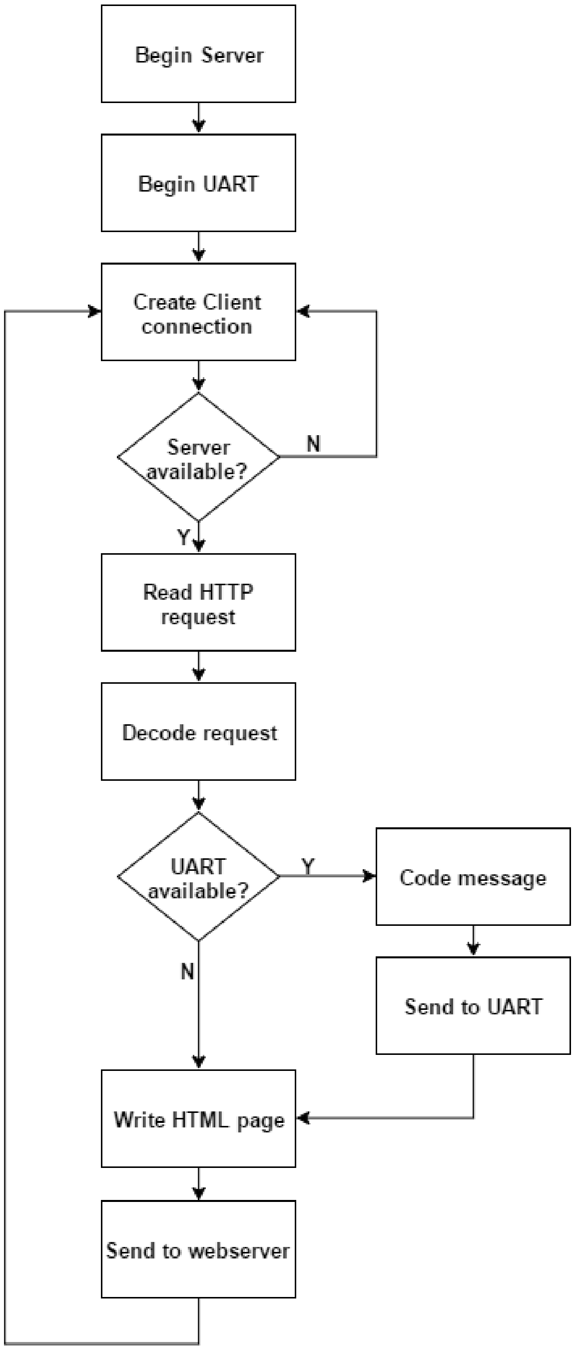

Figure 4.9 - Representation of program running on Arduino Ethernet.

Arduino Ethernet create a layout of user interface, as Figure 4.10 presents, which appears on webserver. The user interface shows the name and status of BLE Peripheral devices available, which are a Smart Plug and two sensors. It provides seven buttons to:

- Switch on and off the electrical appliance.
- Request data from DHT11 sensor.

- Achieve a target temperature or humidity, defined by the homeowner, automatically.
- Enable and disable the alarm system.
- Reset the main controller.

To achieve the target temperature or humidity, the homeowner must connect the electrical appliance that controls the selected environmental parameter to the Smart Plug and define the target value. Then, on automatic mode, the main controller verifies the measures from DHT11 sensor and turn on the Smart Plug if the measures do not accomplish the homeowner's settings.

The BLE Peripheral have names to easy identify them. “SHome_Outlet” corresponds to BLE module linked to Smart Plug, “SHome_T&H” corresponds to BLE module linked to DHT11 sensor and “SHome_SEC” corresponds to BLE module linked to HC-SR04 sensor.

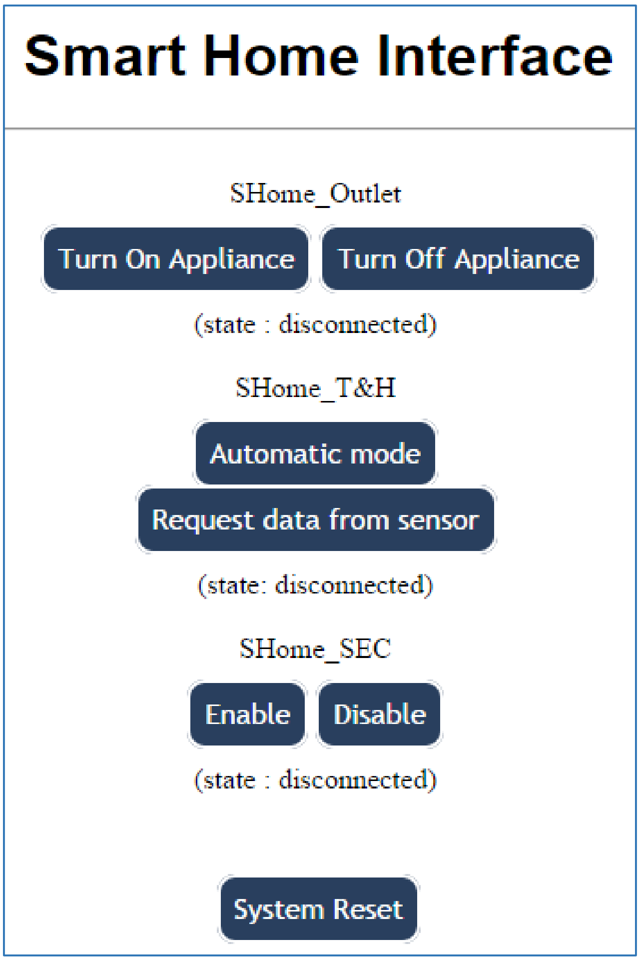

Figure 4.10 – Web user interface when BLE peripheral devices are disconnected.

The homeowner can select one action to execute. When the user selects a button of user interface, the webserver sends to Arduino Ethernet an HTML request containing information about the user action. Table 4.1 shows the messages sent to Arduino Ethernet, when the user press a button on user interface. In consequence of this process, the Arduino Ethernet also send to BLE Central device the homeowner's action. These messages, presented on Table 4.2, has a standard data format that both Arduino Ethernet and BLE Central device expects.

Table 4.1 - Messages sent from HTML to main controller.

Event	HTML message
Turn on appliance	GET /?button1on HTTP/1.1
Turn off appliance	GET /?button1off HTTP/1.1
Request data from sensor	GET /?button2on HTTP/1.1
Reset Main Controller	GET /?reset HTTP/1.1
Automatic Mode	GET /?automatic HTTP/1.1
Manual Mode	GET /?manual HTTP/1.1
Enable the Alarm system	GET /?act HTTP/1.1
Disable the Alarm system	GET /?disact HTTP/1.1

Table 4.2 - Messages sent from main controller to BLE central device.

Event	UART message
Turn on appliance	"Device_id, 1."
Turn off appliance	"Device_id, 0."
Request data from sensor	"Device_id, 1."
Reset Main Controller	"4,-."
Enable the Alarm system	"Device_id, 1."
Disable the Alarm system	"Device_id, 0."

Table 4.3 presents the standard data format that both Arduino Ethernet and BLE Central device expects.

Table 4.3 - Messages sent from BLE Central device to Arduino Ethernet.

Event	**UART message**
Connect	"**N**,**device_name**,**device_id**\n"
Disconnect	"**D**,**device_id**\n"
Data received	"**R**,**device_id**,**byte_0**,**byte_1**, …,**byte_n**\n"

After receiving messages from nRF51822 central device, through UART, Arduino Ethernet creates a new HTML page to send to webserver. The function in the Ethernet library to create webserver connection is a loop function that only breaks when it receive HTML requests. In order to break the loop and detect available data in the UART connection, the HTML page refreshes every one second. Figure 4.11 shows the scheme of main controller connected to nRF51822, with an Ethernet I/O.

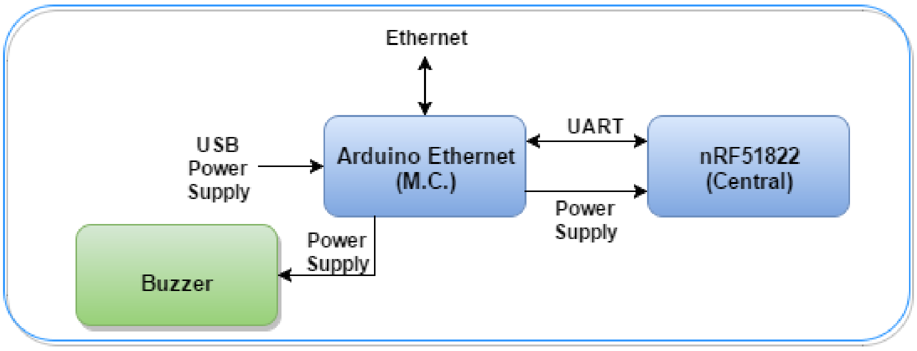

Figure 4.11 - Scheme of main controller connected to BLE module, with Ethernet interface.

Arduino Uno with BLE Shield programming

This load controller (Arduino board plus BLE Shield) links to DHT11 sensor, as Figure 4.12 shows, respond to BLE central device requests, measure temperature and humidity every 5 seconds and send the measures to Main Controller. The response is a two bytes message, one for the temperature value and the other for the humidity value. These values are 255 if a measuring error occurs, in case of sensor damaging.

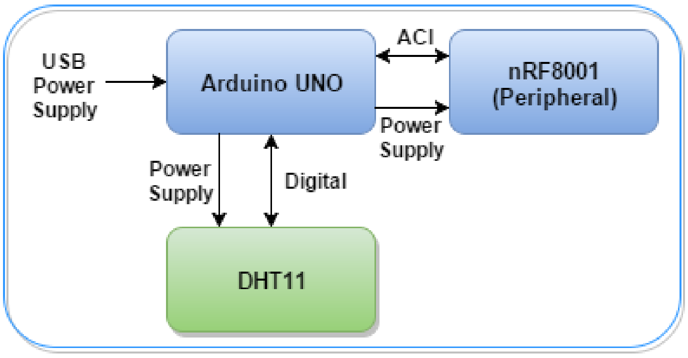

Figure 4.12 - Scheme of Load controller connected to DHT11 sensor and BLE module.

The other load controller (Arduino board plus BLE Shield) links to HC-SR04 sensor, as Figure 4.13 shows, respond to BLE central device orders and it is always measuring distance to an obstacle. When the sensor detects movement, the load controller sends an Alarm message to main controller to fire the alarm system. The Alarm system can be enable and disable through the web interface or through a pressure button to improve convenience. For example, when homeowner gets out he press the button to enable/activate the alarm and when he arrive he press the button to disable/deactivate the alarm system. The response is one-byte message, representing the alarm status.

About GATT protocol, these Peripherals include one service and two characteristics. For load controller linked to DHT11 sensor, one characteristic is to keep the actual measures of sensors and another is to receive orders from GATT client, such as update the sensor measures. For load controller linked to HC-SR04 sensor, one characteristic is to keep the actual status of alarm and another is to receive orders from GATT client, such as activate/deactivate the alarm system. Developed code respects the UUID and handles defined on 4.3.2.

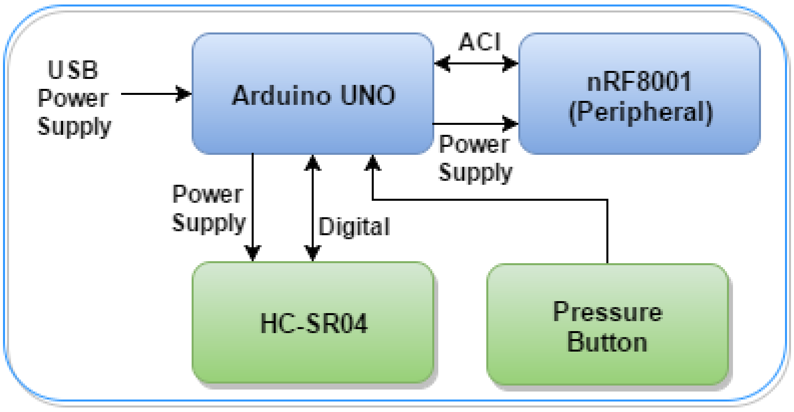

Figure 4.13 - Scheme of Load controller connected to HC-SR04 sensor and BLE module.

nRF51822 programming (Central device)

The developed code to program the Core51822 as central device needs to:

- Start Scanning if connected devices are less than 7.
- Enable UART.
- Discover Peripheral devices that are advertising.
- Connect to peripherals if service UUID matches.
- Discover services.
- Enable CCCD if characteristics UUID matches.
- Send read and write requests to a characteristic value of a peripheral's service.
- Execute commands from UART.
- Send notifications to main controller, through UART.

nRF51822 programming (Peripheral device)

The developed code to program the Core51822 as peripheral device needs to:

- Create one service with two characteristics.
- Start advertising and wait for connection requests.
- Connects when receive a connection request.
- Respond to read and write requests from a connected central.
- Enter in low power mode.

BLE Android App configuration

The Application provided by Evothings Studio already has a BLE stack implemented. It was configured to establish a connection with one BLE device and send one byte when the user press a button on App interface. To include the App, running on Smartphone, on this prototype, it is need to change the BLE profile (service and characteristics UUIDs). Changing the BLE profile permits to exchange information with Peripheral devices on the prototype. It also needs to change the interface to allow the user selects the Peripheral that he wants to connect and select the order to control the electrical appliance or monitor the sensor. The App searches devices by name and when the user choose an order, i.e. press a button, the App writes on characteristic created for writing and waits for receiving data packets from characteristic created for reading. The data packets exchanged are the same with BLE central device. Nowadays, BLE android App acts like a BLE Central device. The App not includes the Alarm system, because the Alarm system is included after App configuration. Figure 4.14 resents the App interface.

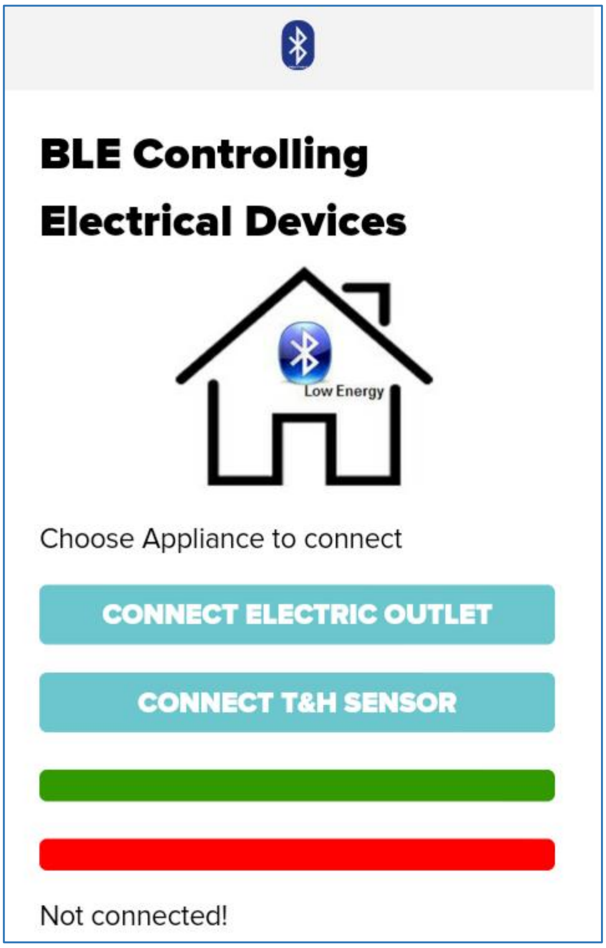

Figure 4.14 - BLE App interface, without any connection.

4.3.4.3 Firmware

This section describes how to upload code to hardware.

Program nRF51822 (central device)

Firstly, it is necessary to erase the chip memory, to clean up previous application programmed, through nRFgo Studio. After erasing, it is necessary to flash SoftDevice, also through nRFgo Studio software. When the board in powered and J-link linked, the nRFgo Studio recognize immediately the board and the chip, as Figure 4.15 shows. When flashing SoftDevice, nRFgo Studio create two regions, one for SoftDevice and another for application. It indicates the memory space occupied and available in each region. As series compatibility matrix shows [35], the S120 SoftDevice occupies 96kB of memory. As the nRF51822 has 256kB of memory, remains 160kB to application. The nRFgo Studio presents the first memory address available for program application, which is, in this case, 0x18000. Those values are necessary to set Keil settings.

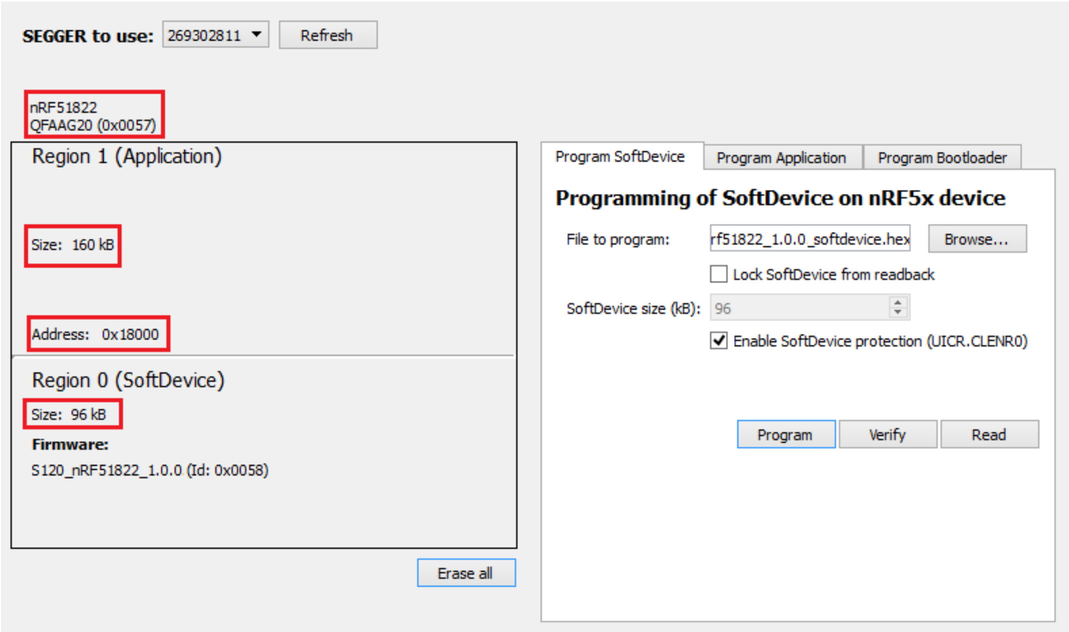

Figure 4.15 - Extracted from nRFgo Studio environment.

To flash developed code to nRF51822, the Keil settings must be in agreement with target device. The definitions of ROM and RAM have to match with ROM and RAM that nRF51822 expect after flashing SoftDevice to nRF51822. As nRFgo Studio indicates, the ROM start on address 0x18000 and the size is 160kB. 160kB correspond to 28000 (160*1024 = 163840 = 28000_{hex}) in hexadecimal. S120 SoftDevice Specification [34] presents the RAM start address, which is 0x20002800, and RAM size, which is 10kB. Figure 4.16 shows the Keil settings for nRF51822. Finally, with correct settings and project compiled, the Keil environment allows directly flash developed code to nRF51822.

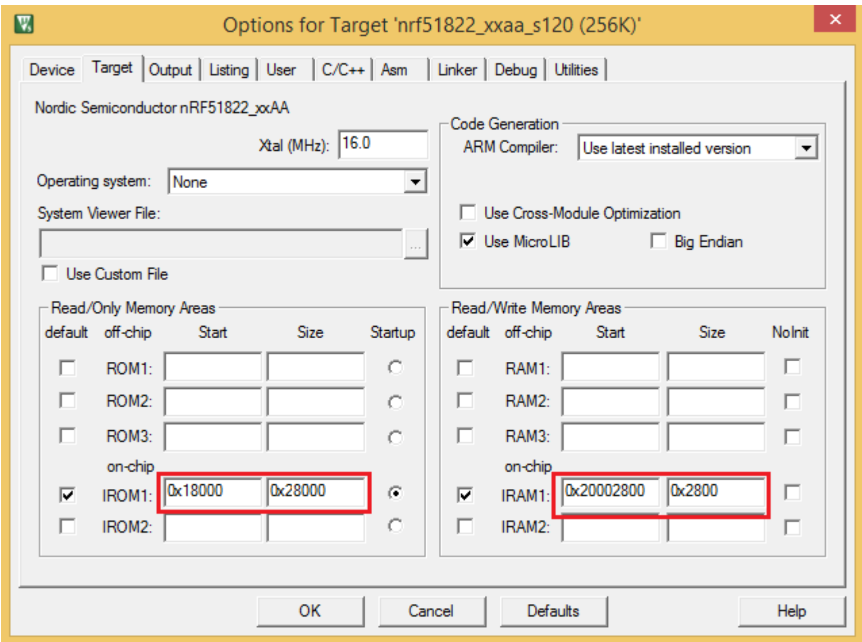

Figure 4.16 - Keil settings - Options for target nrf51822_xxaa_S120 (256kB).

Another option is use nRFgo Studio to flash a *.hex* file, created by Keil compiler, in the section "Program Application", present in Figure 4.15. Keil Software also provide debug mode.

Programming nRF51822 – Peripheral role

Method of programming BLE peripheral device is very similar to programming BLE central device. The only change is the SoftDevice and, consequently, the memory organization. According to S110 SoftDevice Specification [33], S110 SoftDevice requires 88kB and, therefore, remains 168kB (256kB – 88kB = 168kB) to program application. Thus, the ROM start on address 0x16000 and the size is 168kB. 168kB correspond to 2A000 (168*1024 = 172032 = $2A000_{hex}$) in hexadecimal. Still reference on S110 SoftDevice Specification is the RAM start address and size. Figure 4.17 shows the settings for this target, included in Keil settings.

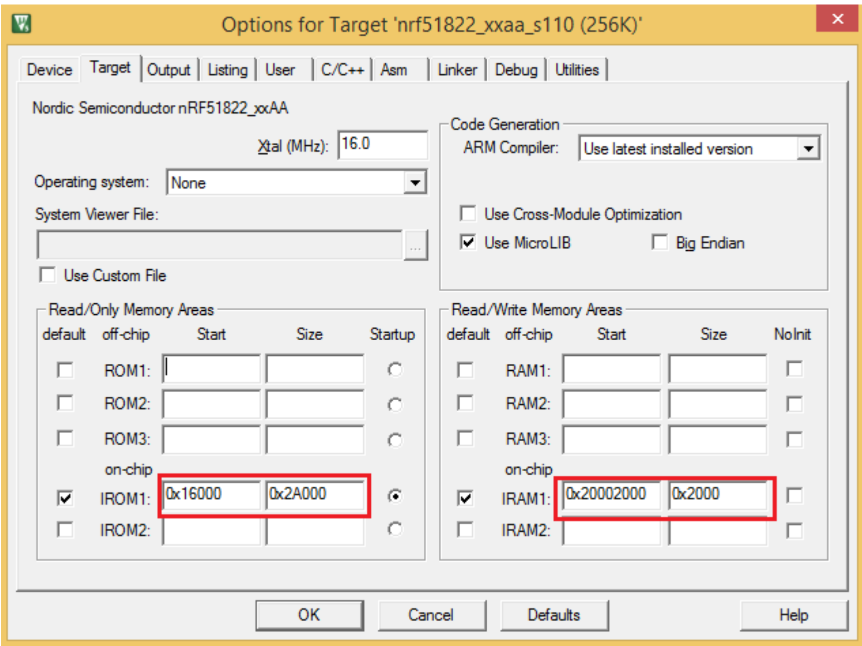

Figure 4.17 - Keil settings - Options for target nrf51822_xxaa_S110 (256kB).

The figures below shows the some parts of prototype development. Figure 4.18 presents the Main Controller (without buzzer) and Figure 4.19 presents the system that monitors temperature and humidity and that it is responsible to send these measures to Main Controller over BLE. Finally, Figure 4.20 represents the alarm system that is responsible to inform the Main Controller when it detects movement. It is possible to see the pressure button that enable and disable the alarm manually.

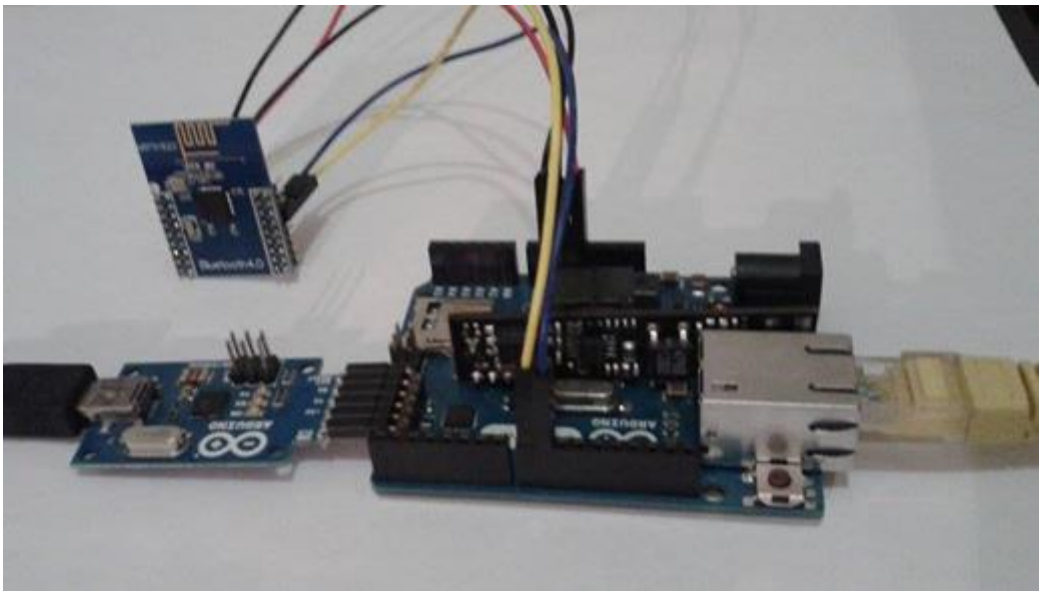

Figure 4.18 - Arduino Ethernet linked to Core51822 and to home router.

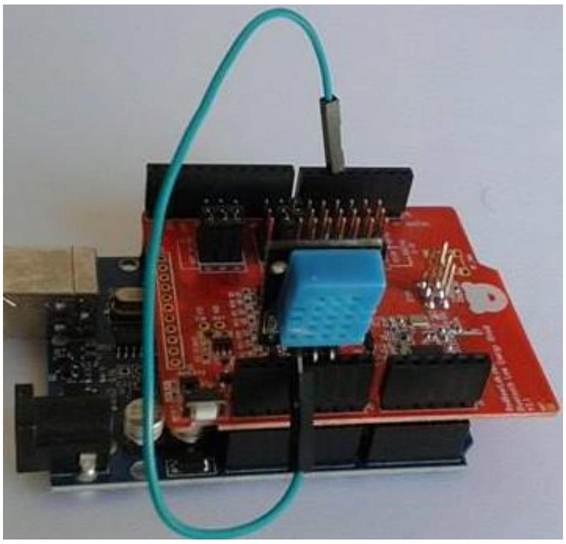

Figure 4.19 – Temperature and Humidity monitor system.

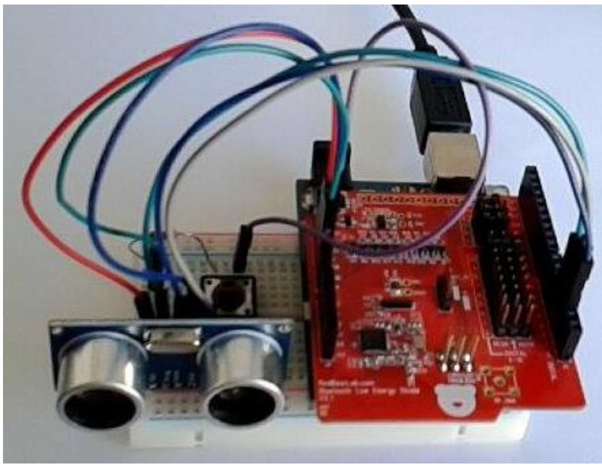

Figure 4.20 - Alarm system.

5 Tests and experimental results

This chapter presents functional results and experimental results about the current consumption in different BLE modes (scanning, advertising, idle and standby), BLE range and transmission latency in the developed prototype.

5.1 Functionality

Functional tests intends to study the Home Automation system behaviour. They test the BLE, UART and Ethernet connections, the performance of controllers and the performance of Smart plug.

To perform these tests, it was execute a sequence of actions to evaluate the system response.

1. Power everything and monitor the user interface on webserver.
2. Request data from DHT11.
3. Turn off Appliance.
4. Disconnect Ethernet cable.
5. Connect to “SHome_Outlet” and turn on the electrical appliance using BLE App.
6. Disconnect from to “SHome_Outlet”, connect to “SHome_T&H” and request data from DHT11 sensor using BLE App.
7. Connect Ethernet cable and monitor user interface.
8. Enable alarm system.
9. Put an object in front of Alarm system.
10. Disable alarm system.

Next figures, from Figure 5.1 to Figure 5.8, shows the results of tests 1 and 2, 3 and 5 to 10, respectively.

The automatic mode is also tested. The user define that the temperature should be higher than 22ºC and the system always turn on the heater, through the Smart Plug, when the temperature is lower than 22ºC. It also works with electrical appliances to cool the temperature and to reduce the humidity. The user always can change between automatic and manual mode. On manual mode, the Main Controller ignore the target values.

The status on the web user interface refers to BLE connection or to the status of respective system. The status referent to systems, like “appliance turn on”, is a response of load controller to an order from Main Controller. It means that the status only change if the load controller confirm that the order was successfully executed.

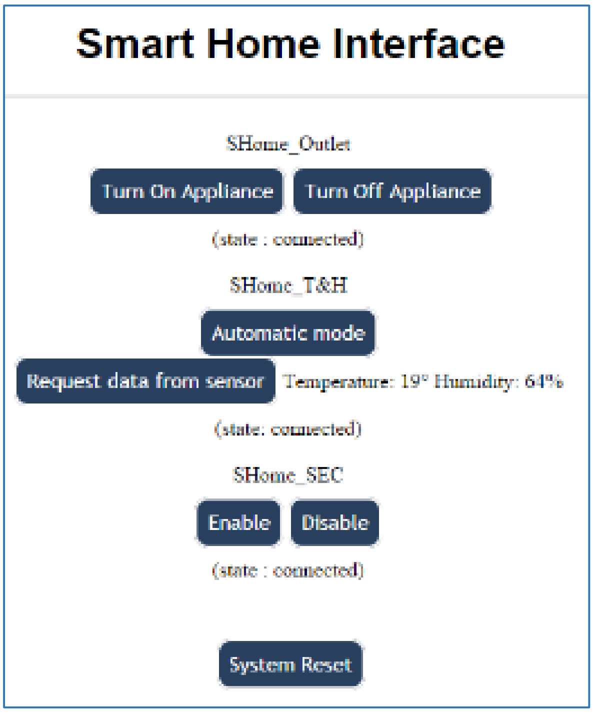

Figure 5.1 - Web user interface, after execute the test 1 and test 2.

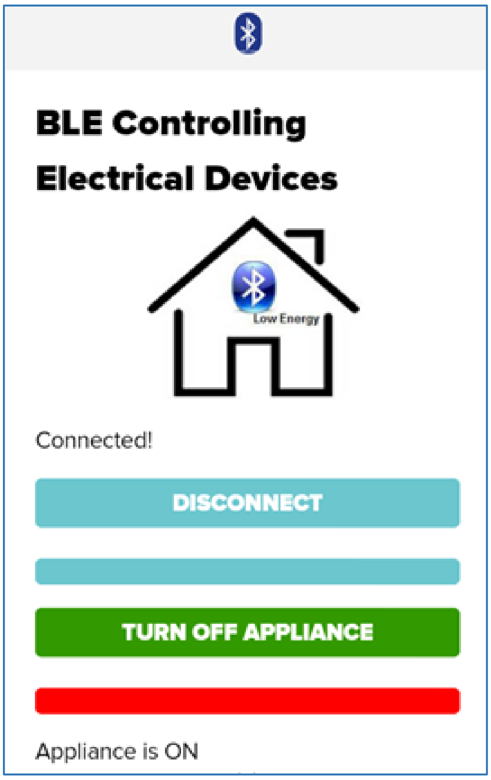

Figure 5.3 - BLE App interface, after execute test 5.

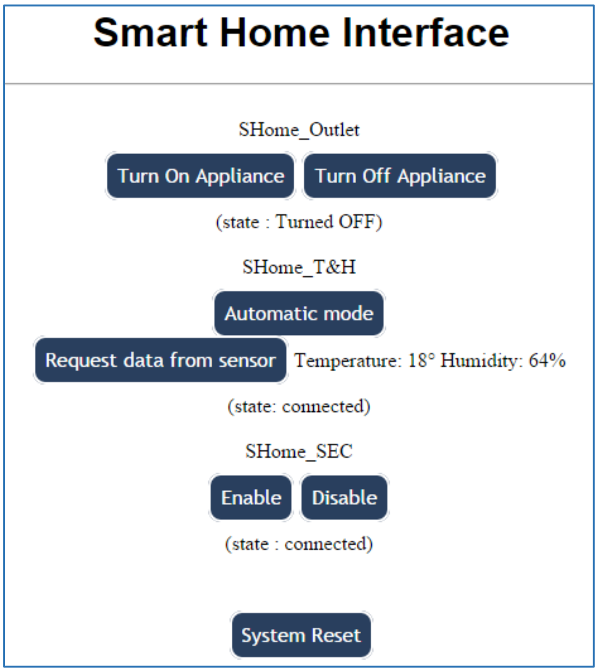

Figure 5.2 - Web user interface, after execute the test 3.

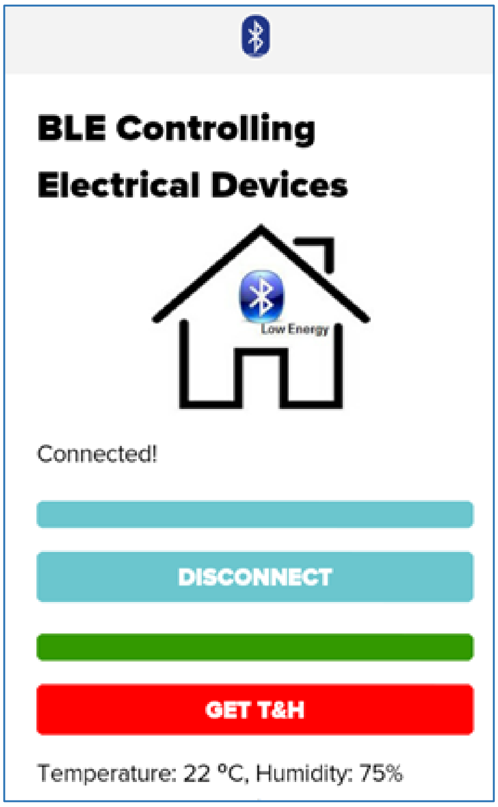

Figure 5.4 - BLE App interface, after execute test 6.

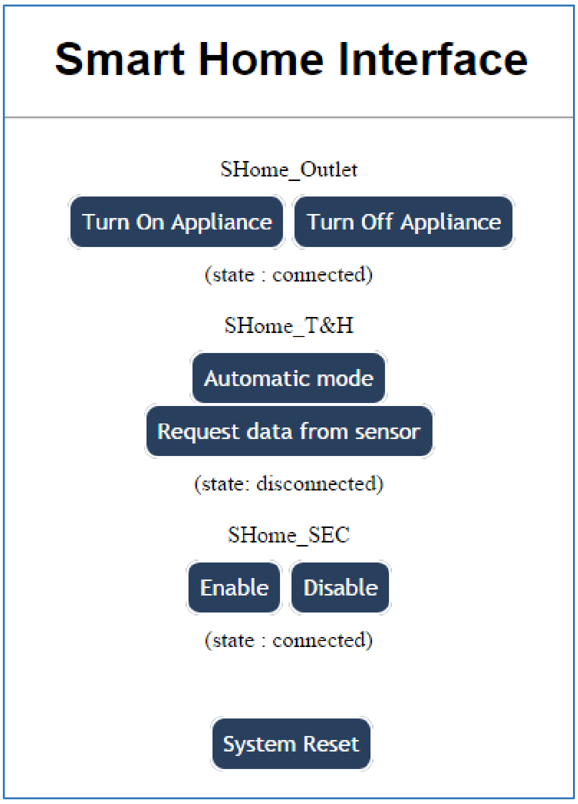

Figure 5.5 - Web user interface, after execute test 7.

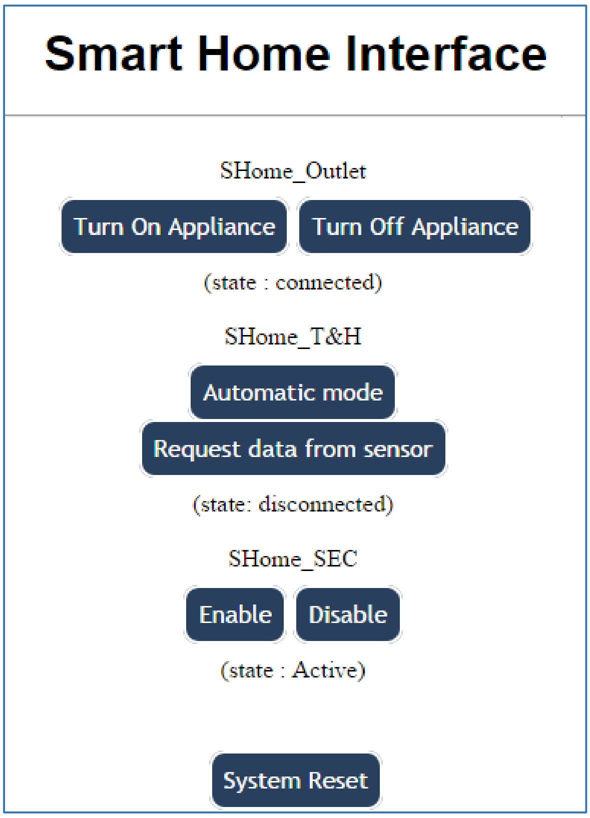

Figure 5.6 - Web user interface, after execute test 8.

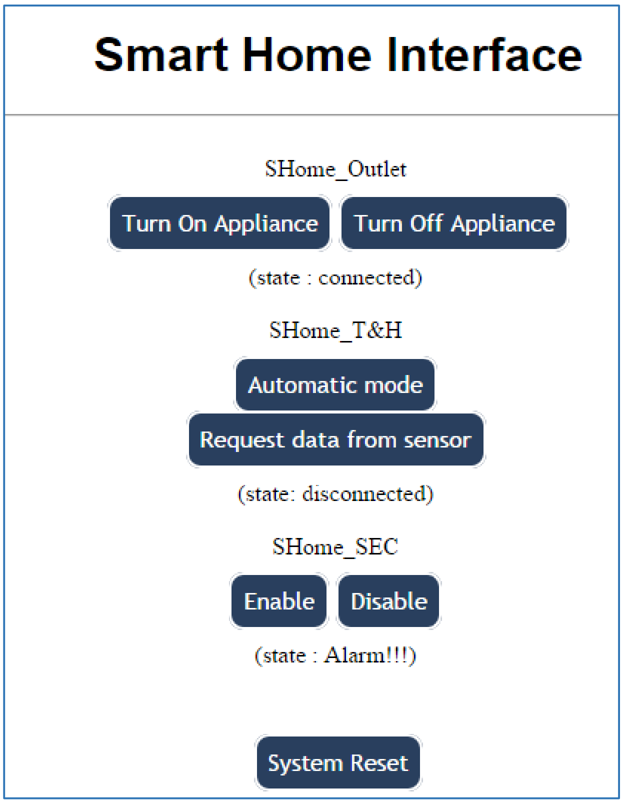

Figure 5.7 - Web user interface, after execute test 9.

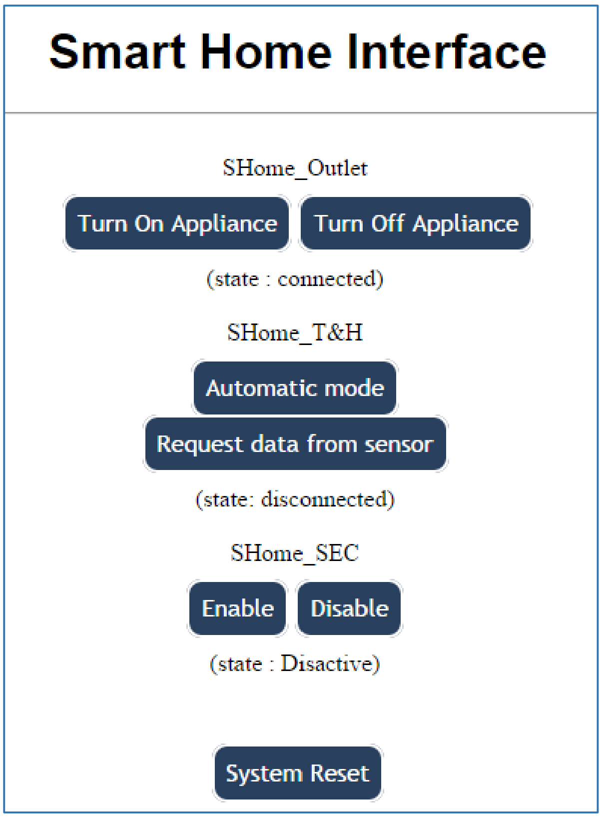

Figure 5.8 - Web user interface, after execute test 10.

As figures show, the Home Automation system works exactly how it should. List below present some notes about results.

- The test 1 is equal to the test 2 because the DHT11 sensor sends data every 5 seconds.
- The test 4 does not produce results because the HTML page shows a white page.
- On test 5, the Smart Plug provides power to electrical appliance.
- From Figure 5.5 to Figure 5.8, the “SHome_T&H” appears as disconnect on Web user interface because it is connected to BLE App.
- After test 9, the buzzer starts beep until realize test 10.

5.2 Latency

Latency tests intends to measure how much time one action takes to achieve the action target.

The list below shows the actions undergoing to latency tests, when both the Arduino Ethernet and the BLE App control and monitor the Home Automation system and when BLE App

1. From connecting BLE peripheral device to change the status on user interface.
2. From user order to turn on appliance to appliance turning on.
3. From user request data from sensor to the measures appears on user interface.
4. From put an object in front of HC-SR04 to buzzer start beep and Alarm message appears on Web user interface, when alarm system is enable.

An external digital timer measures the time since the action occurs until the action target occurs.

Table 5.1 and Table 5.2 show the latency results, in seconds, from actions described in the list above. As the App is not configure to be compatible with Alarm system, BLE App does no submit to the experiment 4.

Table 5.1 – Latency when the Arduino Ethernet controls and monitors the system.

Action	Latency
1	1s - 2s
2	<1s
3	<2s
4	<1s

Table 5.2 – Latency when the BLE App controls and monitors the system.

Action	Latency
1	<1s
2	<0.5s
3	<1s

Despite of the measurements are not highly accurate, with a major error of 0.5 seconds, the results satisfied the requirements, when considering Home Automation. The range of acceptable values for system response, in this area, is around 1 to 3 seconds.

5.3 Range

These experiments intend to test the real range of BLE modules in two different environments, open space and inside house, where obstacles exist. These experiments include tests on Core51822 and BLE Shield, using the maximum transmitting power (+4 dBm).

To measure the range inside the house, both Core51822 and BLE Shield have been moved away until they cannot connect to the central, which is fixed. To measure the range in open space, both Core51822 and BLE Shield are fixed, and the Smartphone running BLE App has been moved away until it cannot connect with both BLE chips. The experiment inside the house includes three walls between two devices.

Table 5.3 - Range measured of BLE modules.

Chip	Environment	Range [m]
nRF51822	Open Space	**58**
	Inside house	13
nRF8001	Open Space	**55**
	Inside house	11

Compared to BLE range described on Table 3.2, the open space range is higher than expected. The range inside the house decreases a lot with obstacles. Considering Home Automation, this range is short to the purpose of Home Automation. However, the range fit in small houses. Large houses requires additional hardware to repeat the signal, increasing the range of technology.

5.4 Current consumption

These experiments intend to discover the power consumption of Core51822 mode, actuating as BLE Peripheral, in advertising, pairing and sleep mode. The current consumption depends on several parameters, such as TX power and power supply. Power supply used is 5V. BLE peripheral device uses a maximum transmitting power (+4dBm), an advertising interval of 40ms, an advertising payload data of 10 bytes and always advertising.

A digital multimeter measures de DC current of Core51822 in different stages. When Peripheral is advertising, its consumption is constant, so it is possible to measure with a multimeter. When pairing, the consumption has current peak and decreases to low power mode. The multimeter measure the maximum value of current during this peak, which take about 10 ms when it is five meters away from central. This time was measured with an internal timer that Core51822 provide. The current consumption on low power is constant, which makes possible to multimeter measure it. The standby mode allows the device to enter in a low power mode, staying connected, but capable to listen BLE messages coming.

The nRF8001 presented on Arduino BLE Shield is not submitted to this experiment because is very difficult to analyse only the current consumption spent in a BLE connection because the Arduino consumption is influence this measures.

Table 5.4 presents current consumption in different operation modes of nRF51822, working as BLE Peripheral.

Table 5.4 - Current consumption of Core51822 module, working as Peripheral device.

Mode	Average Current consumption [mA]
Advertising	1.2
Pairing	2.82
Standby	0.22
Sleep	0.01

The results shows that Core51822 is a low-power micro-controller and that the standby mode is more power efficient than advertising. Considering Home Automation, where the system requires a faster answer between Central and Peripheral, it is more efficient if they are always connected, instead of Central request a connection every time it needs. Sleep mode is very power efficient. However, there are only two ways of the chip wake-up. One is by a physical contact, which does not satisfy the requirements because it removes the autonomous function of the system. Another way is to define an interval for the chip wake-up. If this interval is small, the consumption is higher than in a standby mode because the device has to advertise and pair. And if the interval is large, the system response will be less efficient.

6 Conclusions

This work achieves the proposed objectives, such as the implementation of a BLE communication between a smartphone, a tablet or a personal computer with home electrical appliances, as well as design an architecture of a Home Automation system with energy management (HEM). These two main objectives permit to create a demonstrative system of the proposed architecture. This prototype permit to perform a full analysis of BLE technology. However, to simulate a larger Home Automation system it needs some improvements.

The requirements influence the architecture. The main requirement is to control and monitor electrical home appliances and sensors, from everywhere. To achieve it, the main controller communicates with these devices through BLE, inside the house, and it has an internet connection to provide the homeowner the full access to his home system from everywhere. To achieve HEM, the main controller communicates with smart meter to perform a smart energy management.

The developed prototype was created to demonstrate a Home Automation system, without energy management, where BLE support the communications inside house, and the homeowner controls and monitors his home system through internet. A prototype with three Arduino boards, two BLE shield, two BLE modules, two sensors and one smart plug allows testing a demonstration of using BLE in Home Automation, which matches with Home Automation requirements proposed. Software was develop to program the Arduinos, Shields and BLE modules to accomplish the requirements. Prototype was successfully build. However, some details can improve system performance, price and mobility. Ethernet communication decrease the Main Controller mobility. Arduinos as load controllers are too large, too expensive and consumes a lot of current considering their purpose. On user interface on Web, the status confuses the user. This parameter must refer only to the BLE connection status and the appliance or system status should be present on another place. About BLE App, it is outdated, because it is not allow the automatic mode and does not communicate with Alarm system.

About future work, there are some changes to increase the system value to homeowners and to decrease the costs. The first, which is the most important, is replace the main controller to another with more available memory to programming. Raspberry pi could be a solution because it is a microprocessor, which is much faster than Arduino and provide 512 MB of memory. Raspberry pi also include an Ethernet interface. Although, the HTLM page occupies about 34% of dynamic memory available of Arduino Ethernet. If the cloud allocates code to program HTML page (normally paid) instead of Arduino Ethernet creates HTML page, Arduino Ethernet offers more memory for programming. The security is also an improvement to perform. Security on alarm system, on web user interface and on BLE communication. To achieve an HEM system, the HA system needs to include a smart meter to implement the communication with DSO to receive information about power consumption and real-time electricity price. It is also necessary to implement scheduling algorithms, based on homeowner's settings and preferences, which reduce the power consumption during the periods of peak loads

Experimental tests permitted a deeper analysis of BLE technology, namely testing current consumption with different values of TX power, with different values for advertising interval, connection interval, and so one.

Concluding, the results of this work shows that BLE can support communications inside house, due to its limits, like the small range. However, the compatibility with smartphones and tables is a great advantage in this area. The low-cost and low power consumption increase a lot the potential of this tecnhology.

References

[1] F. E. R. Commission, “Assessment of demand response and advanced metering,” 2012.

[2] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand e D. Boyle, From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Elsevier: Academic Press, 2014.

[3] C. Perera, A. Zaslavsky, P. Christen e D. Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey,” IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 16, NO. 1, 2014.

[4] X. Carcelle, Power Line Communications in Practice, Artech House, 2009.

[5] C. Vivekananthan, Y. Mishra e F. Li, “Real-Time Price Based Home Energy Management Scheduler,” IEEE TRANSACTIONS ON POWER SYSTEMS, 2014.

[6] C. Corporation, “Control4,” Control4® Corporation, 2014. [Online]. Available: http://www.control4.com/basics. [Acedido em 13 11 14].

[7] Corporation, Control4, “Control4 Offers A Starter Kit for Home Automation,” Control4 Corporation, 27 07 2012. [Online]. Available: http://www.control4.com/press_releases/2012/06/27/Control4-Offers-A-Starter-Kit-for-Home-Automation. [Acedido em 25 11 14].

[8] HomeSeer, “CONTROL YOUR HOME FROM ANYWHERE WITH THE #1 RATED HOME CONTROL SYSTEM,” HomeSeer, 2013. [Online]. Available: http://www.homeseer.com/index.html. [Acedido em 13 11 14].

[9] HomeSeerTV, “Control Home with a iPhone, iPad or Android Smart Phone or Tablet Computer,” 10 02 2012. [Online]. Available: https://www.youtube.com/watch?v=pFTyV7qb1qQ. [Acedido em 29 11 14].

[10] N. Telecom, “AiMiR Home Energy Management System (HEMS),” Nuri Telecom, 2010. [Online]. [Acedido em 26 11 14].

[11] Z. Alliance, “Understanding ZigBee Gateway,” 2010.

[12] W.-F. Alliance, “Wi-Fi Programs,” Wi-Fi Alliance, 2014. [Online]. Available: http://www.wi-fi.org/certification/programs. [Acedido em 14 11 14].

[13] B. SIG, “Bluetooth Basics,” Bluetooth SIG, 2014. [Online]. Available: http://www.bluetooth.com/Pages/Basics.aspx. [Acedido em 14 11 14].

[14] S. Designs, “Z-Wave,” Sigma Designs, 2014. [Online]. Available: http://www.z-wave.com/. [Acedido em 11 27 14].

[15] B. SIG, “The Low Energy Technology Behind Bluetooth Smart,” Bluetooth SIG, 2014. [Online]. Available: http://www.bluetooth.com/Pages/low-energy-tech-info.aspx. [Acedido em 17 11 2014].

[16] R. Nilsson, “Bluetooth Low Energy Technology Makes New Applications Possible,” Digi-Key corporation, 2014. [Online]. [Acedido em 27 11 2014].

[17] Nordic Semiconductor, “Nordic nRF51 SDK,” 2015.

[18] Nordic Semiconductor, “nRF51822 - Product Specification v3.1,” 2014.

[19] Nordic Semiconductor, “nRF51 DK - Development Kit for Bluetooth Smart, ANT and 2.4GHz applications.,” 2014.

[20] Nordic Semiconductor, “nRF51 Dongle - Development and test dongle for Bluetooth Smart, ANT and 2.4GHz applications.,” 2014.

[21] Texas Instrument, “2.4-GHz Bluetooth low energy and Proprietary System-on-Chip (Rev. D),” 2013.

[22] Texas Instrument, “Bluetooth Low Energy - CC2541 Mini DEvelopment Kit USer's Guide,” 2013.

[23] Waveshare International Lda, “nRF51822 Eval kit,” Waveshare Electronics, 2014. [Online]. Available: http://www.wvshare.com/product/NRF51822-Eval-Kit.htm. [Acedido em 04 2015].

[24] Nordic Semiconductor, “nRF8001 - Product Specification V1.3,” 2015.

[25] Red Bear Company, “BLE Shield V2.1,” 2015. [Online]. Available: http://redbearlab.com/bleshield/. [Acedido em 06 2015].

[26] Android, “Android 5.0 APIs,” Android developers, 2015. [Online]. Available: https://developer.android.com/about/versions/android-5.0.html. [Acedido em 07 2015].

[27] Arduino, “Arduino UNO,” 2015. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Acedido em 08 2015].

[28] Arduino, “Arduino Ethernet,” 2015. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardEthernet. [Acedido em 09 2015].

[29] D.-R. UK, “DHT11 Humidity & Temperature Sensor,” 2010.

[30] Elec Freaks, “Ultrasonic ranging module : HC-SR04”.

[31] Silvercrest, “Electrical sockets,” [Online]. Available: http://www.lidl-service.com/cps/rde/xchg/SID-BC755AC4-7610F0F9/lsp/hs.xsl/product.html?id=11133081&title=Conjunto.

[32] D. S. M. Paul Scherz, Practical Electronics for Inventors, Third Edition, McGraw-Hill Education TAB, 2013.

[33] Nordic Semiconductor, “S110 nRF51822 - SoftDevice Specification v1.3,” 2014.

[34] Nordic Semiconductor, “S120 nRF51822 - SoftDevice Specification V1.1,” 2013.

[35] Nordic Semiconductor, “nRF51 Series Compatibility Matrix V1.2,” 2015.