TEHNIČKO VELEUČILIŠTE U ZAGREBU
PROFESSIONAL STUDIES OF ELECTRICAL ENGINEERING

Eneko Malo

WIRELESS TEMPERATURE CONTROL SYSTEM AND LIGHTING

FINAL WORK no. E2035

Zagreb, July, 2016.

TEHNIČKO VELEUČILIŠTE U ZAGREBU

PROFESSIONAL STUDIES OF ELECTRICAL ENGINEERING

Eneko Malo

JMBAG: 02460663050

WIRELESS TEMPERATURE CONTROL SYSTEM AND LIGHTING

FINAL WORK no. E2035

Mentor: dr.sc. Ljubivoj Cvitaš dipl.ing.

Zagreb, July, 2016.

TEHNIČKO VELEUČILIŠTE U ZAGREBU
POVJERENSTVO ZA ZAVRŠNI RAD

Zagreb, 7. srpnja 2016.

ZAVRŠNI ZADATAK br. E2035

Pristupnik: **Eneko Malo Mirada (0246066305)**
Studij: Stručni studij elektrotehnike
Smjer: Energetska elektrotehnika

Zadatak: **WIRELESS TEMPERATURE CONTROL SYSTEM AND LIGHTING**

Opis zadatka:

The main objective of this diploma thesis should be the design, development and production of functional model system for measuring the temperature and lighting in the building. Depending on the measurement via dedicated sensors the temperature and illumination should be able to control remotely.
To make this possible, the system will be programmed in C language. The system consists of two "slaves" units that are powered by batteries 3V, realized with MSP430G microprocessor and a "master" unit realized with a microcontroller - based mbed module. Master communicates with slave devices via Bluetooth module HC05. The light intensity is measured by a diode BPW41N and temperature sensor KTY10. Master controls the lights on the LED light module and includes ventilation depending on the measured temperature. There is a secondary master used an Android device with the application "Blueterm", which reads data from subordinate units, and if necessary, control the LED on them.

Zadatak uručen pristupniku: 7. srpnja 2016.
Rok za predaju rada: 11. srpnja 2016.

Mentor:

dr. sc. Ljubivoj Cvitaš, pred.

Predsjednik povjerenstva za završni rad:

Abstract

The main objective of this thesis is to design, develop and make a functional model of system for measuring the temperature and lighting in a building. Depending on the measure obtained it could me change this temperature an light by a remote way. For making this possible, the system is going to program in C language. The system consist of a two “slaves“ units which are powered from a 3V battery, realized with MSP430G microprocessor and“master“ unit realized with a microcontroller – based MBED module. Master communicates with slave units via Bluetooth module HC05. Light intensity is measured using diode BPW41N and the temperature with sensor KTY10. Master controls the lights on the LED light module and includes a ventilation depending on the measured temperature. There is a secondary master used Android unit with the application “Blueterm“ which is reading data from the slave units, and if it is necessary, control the LEDs on them.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In recent decades, the world of technology and electronics, in particular, is growing exponentially. It started with simple analog devices with a single use or functionality that they were a revolution for those times, and over the years, the electronic components were improving and with them also the designed devices and their functionalities.

As a result of the comments, the number of devices that today include an electronic circuit design is unimaginable. They can be found from everyday objects such as electrical appliances, mobile phones or computers to more complex and sophisticated professional systems.

Thanks to the evolution of engineering is taking place a change in society in the sense that people are changing their lifestyle to a one more connected to the Internet. A clear example is that a few years ago no one thought we would have a Smartphone with so many capacities, while now anyone can imagine a life without this device.

Because of this change in the human mind, right now and for the future, society is in a process of changing the way of connecting to Internet. At present it is people who connect to internet this being known as the Internet of people, i.e. it is people who connect to the Internet forming a network. However, many important engineering companies are showing their interest in starting to connect all kinds of devices to the Internet, this is known as the concept of Internet of things, and i.e. the network will be made up of objects.

There is a growing need for speed in production, a need of greater savings in terms of daily processes, etc. Therefore, to achieve these objectives it is important to automate most processes, whether industrial or not. This requires having a network of sensors, actuators etc., which connecting all of them can give us the desired information at any time. Even without a human help, i.e. talking to each other they will be able to perform complex tasks.

This thesis is oriented in this direction. It desired to design a system capable of controlling temperature and lighting of a building remotely. For this purpose a series of modules and sensors are used to monitor these values and if necessary to act changing them. Remote or wireless connection is made using the Bluetooth protocol because it offers good features at close range. This connection is made through a free and easy to use android application called Blueterm which will be detailed in later chapters.

The monitoring and temperature control system consists of two devices, on the one hand, a sensor able to get the temperature value to be placed at the point of the desired room, KTY10. On the other hand, a ventilator that can vary the temperature if required. There are three different states. First, the state in which the temperature is low and it is not necessary to use the ventilator. Second, the state in which the temperature is high enough to use the fan, and depending on this value cause the fan to rotate faster or slower. And last and third, the state in which the temperature is so high that it can be because there is fire in the room so it is necessary to turn off the ventilator so that the flames cannot spread over the building.

The monitoring and lighting control system also consists of two devices, on the one hand, a sensor capable of receiving the light information of the desired room which in this case will be a BPW41N diode. On the other hand, the LED which can be change its brightness depending on the time making it glow more or less or even turning this off.

On one hand, there is a part of the system which the temperature and lighting sensor is managed by MSP430G microprocessor. This part of the system will be the slave part. In the final system two equal slave modules to measure at two different points the desired parameters are used.

On the other hand, as master module is the control part of the system. In this part the ventilator and the LED are managed by the microprocessor MBED. Communication between the two microprocessors as already mentioned is via Bluetooth. To this end, HC-05 modules are used.

Finally, there is also another communication between MBED microcontroller and the PC in the master module.

The block diagram is shown in the Figure 1.

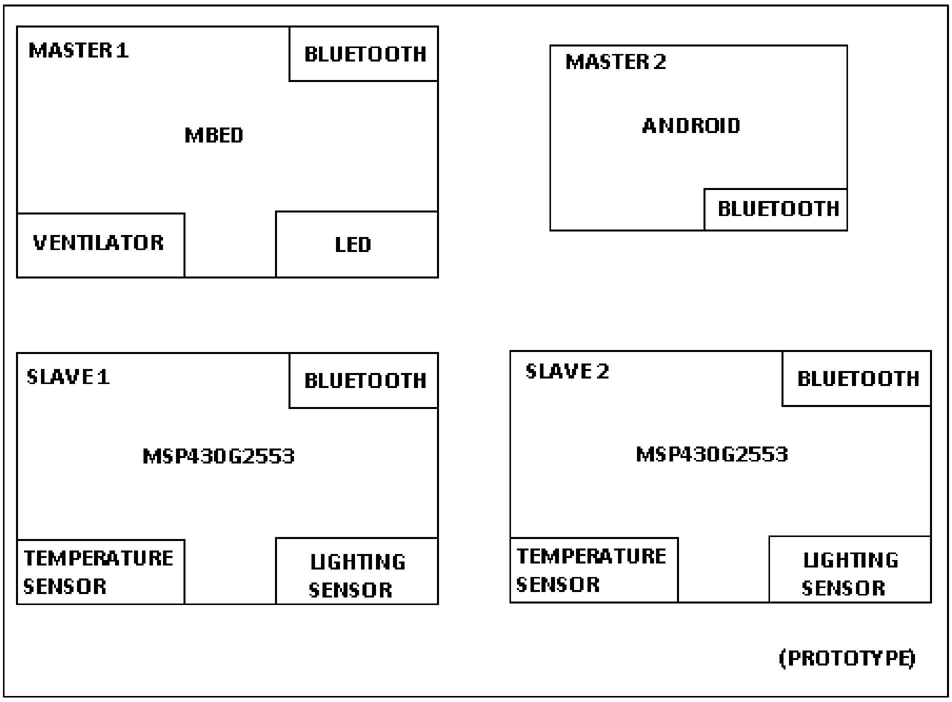

Figure 1: Block Diagram

2. GOALS AND PURPOSE

As already mentioned in previous chapter, the main objective of this thesis is to design, develop and implement a functional model of system for measuring the temperature and lighting in a building. This design will be programmed in the high level programming language of C / C ++.

The complete system consists of two master blocks, two slave blocks and a wireless communication between them. Master blocks have the main feature of being able to control part of the circuit where as slaves provide the desired values. The first master block is able to modify the ventilator and the LED brightness depending on certain parameters via UART connection to the PC. This connection requires a computer connected to MBED microcontroller with a serial cable that adapts USB-A and micro USB-B.

The second master block is another module capable of controlling devices wirelessly by the application "Blueterm". On the other hand, the slave’s blocks are two equal blocks that provide the measurements of temperature and lighting desired at the indicated points.

Related to the main objective of the thesis a series of secondary objectives that this system can provide are shown in the following paragraphs.

1) Improved quality of life.
2) Saving energy in efficient way.
3) Greater security/safety.

2.1 Secondary objectives

2.1.1 Improved quality of life

This thesis has the peculiarity of being oriented in some way to the internet of things, since they are being connected home devices to the Internet. And these are being controlled from another device also connected to the Internet. Internet of things allows us a more comfortable home life which explains the improvement of living standards. Because it allows time-saving conveniences of home. This thesis is also aimed at the near future because more and more the devices will be connected to Internet, either heating, washing machine, microwave etc.

2.1.2 Saving energy in efficient way

This thesis as it is connected to a computer allows us to analyze when it is more cost-effective use of energy, thanks to what is known as smart meter. In this thesis this will be a minority target because the values measured have a clear dependence on the time of day. In this case it is rather a first step to adapt to future devices such as washing machines because they can be programmed when the energy is cheaper. That's why a more efficient use of energy provides economic savings.

2.1.3 Greater security/safety

This thesis as already explained briefly above can be monitored temperature values, which in extreme situations can mean the existence of fire in the room. Looking ahead could also connect smoke detectors and depending on certain measured trigger other devices to extinguish this fire. This system provides a certain level of security.

3. BENEFITS

In this part of the document presents the main benefits offered by the design system of the thesis. In this paragraph explains the general benefits and not a lot of details of the technical ones, because there is another section in the document for this purpose. These mainly benefits can divide in three fields.

1. Technical benefits
2. Economical benefits
3. Future implementation benefits
4. Other benefits

3.1 Technical benefits

The development of this thesis has summarized as a control a number of devices remotely. This form of control devices without the need of cables provides certain benefits in terms of mobility. Also allows much more constant maintenance of these and more active accordingly. That is, it allows to know at any certain time the desired data. In the case of this thesis the wireless communication technology used is Bluetooth so that the distance between the devices cannot be very large, as will be studied in subsequent sections of the document. However, in futures improvements can be connected in some way to internet and control will be more widespread.

3.2 Economical benefits

The main objectives of the design of this system as already mentioned above tend to be oriented with the economic issue. That is, get the most efficient use of energy for saving costs in the short and long term. In addition, the issue of being continuously monitoring of some things, in this case, the temperature and light, you can get to avoid certain loss of property. As already explain before, for example prevent an outbreak of fire quickly.

3.3 Future implementation benefits

As explained previously this thesis is easily expandable and upgradeable, for getting this it only would have to orient a broader way on the Internet of things, that is, moving from control only the light and temperature of a house to control all kinds of devices as washing machines, vacuum cleaners autonomous, television, microwave, in short, all possible appliances. This allows the standard of living increases since certain everyday activities may be replaced by some simplest. And that it somehow will be smart houses which offers a high number of profits.

3.4 Other benefits

In addition there are other benefits commented not so important but are also interesting to comment. The mere fact of being able to control certain things remotely allows us to save time because for example it is possible to control the temperature of our house while we are on the way to home from work, which will enter a house in which the temperature it will be more pleasant and also we do not have to connect the heating or air conditioning on arrival.

In short, this thesis offers a number of benefits on a small scale since its implementation is bounded to two different measures. However, if you want to bring the large-scale thesis benefits would be much greater and of all kinds. Some of these possible applications and benefits are explained in following chapters of this document.

4. FUTURE APPLICATIONS AND IMPROVEMENTS

4.1 General description

As briefly mentioned in previous chapters of the document, it is interesting to show future applications and improvements to the thesis. In other words we can say that this thesis is the base of a close improvement or an idea for the future.

The main application this offers is the aforementioned concept of Internet of things, that is, change the current paradigm known as the Internet of people, ie, it is people who connect to the Internet to connect things to the Internet. The following figure shows studies by CISCO of the number of Internet-connected devices. It is estimated that by 2020 there will be around 50 billion connected devices. This exponential growth in some way shows the importance of this concept (Figure 2).

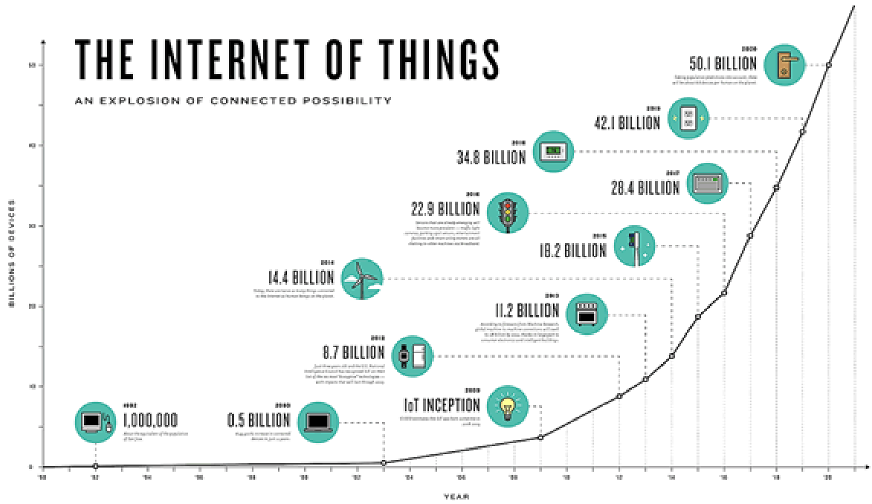

Figure 2: Growth of Internet of things [1][1]

The main application of this technology is known as sensor networks. In short, form a network with many sensors. For example, from dozens of sensors for monitoring a home to thousands of them to monitor forests and be able to ensure to no forest fires occurred.

[1] http://www.i-scoop.eu/internet-of-things/

Then a series of applications of this technology will be shown but without going into much detail as it is not the main objective of the thesis but yes something important.

4.2 Some applications

4.2.1 Smart buildings

One of the first applications and more precisely linked to this thesis is smart buildings. Especially the part of digital home, because it is possible divided in two distinct types, the aforementioned and monitoring structures, the latter more used in the Americas.

On the one hand, the benefits of home automation have already mentioned before summing so here are energy savings, improved comfort and safety of both people and buildings.

On the other hand, structures monitoring by using a sensor network deployed in certain structures, whether bridges, buildings or any type of construction allows detect some faults. For example, vibration in bridges etc.

4.2.2 Automotion sector

As the automotive sector concerns there are two difference areas: the first is aimed at the automotive industry while the second is geared more towards road safety.

In the first area the improvements and the greatest benefits can find in the production section, accelerating assembly processes, improve logistics, increase quality control and improve customer service.

However, the second sector improves communication between vehicles that are circulating in the same area which means many improvements in road safety as it is possible to make contact to warn of an area of ice sheets thanks to some special sensor, warn that another vehicle is in a dead angle, distance control between vehicles, etc. (Figure 3).

Figure 3: Communications between vehicles [2][2]

4.2.3 Medicine

Medicine is another very promising field. With the reduction in size of the sensor nodes, the quality of life of patients who have to have their vital signs checked, can improve substantially. This is because, thanks to these sensor networks may monitor keystrokes, pressure, blood glucose level, acting in all these cases very quickly.

Using these intelligent sensor networks in the field of health monitoring significantly reduces the number of hospitalizations and medical visits, and thus the annual costs of health care worldwide.

4.2.4 Agriculture

In the field of agriculture can highlight the following improvements. Better control of the amount of water, fertilizer or pesticide required per plant, because to measure the soil moisture. Optimize production and quality of a crop because of having some type of alarm for animal intrusion or damage incurred by meteorological phenomena.

---

[2] http://www.ideaseinventos.es/2014/02/05/eeuu-implementaria-sistema-de-comunicacion-entre-vehiculos/

5. ANALISIS OF ALTERNATIVES

To develop the thesis, there is a need to make some decisions when making the design that will be relevant for the final success or failure of the thesis. This is why it is important to analyze the most important alternatives that can be taken to achieve the best possible solution.

In this part of the document the most important alternatives that have been studied to solve the various issues are discussed. Besides, it is also show the selected solution, and the views or selection criteria that have been taken into account to make the final decision.

In the following paragraphs will discuss what the problem to be solved in this thesis is. What are the different options available to resolve and, finally, what is the best solution to achieve this.

5.1 General description

The main task of this thesis is to make a design, development and implementation of a system capable of controlling the temperature and light intensity of a house. This control must be able to perform remotely.

To carry out this thesis has taken into account the following:

First, it may decide what will be the measures that are to be controlled or monitored, for making this it is important to study a little more extensively which are the possibilities and monitoring devices allow the desired measurements.

Second, it is interesting to study which wireless communication protocol will be used for remote control. Other enhancements offered on the same style, cost involved etc.

Third, microprocessors are to be used for thesis design, i.e. its compatibility with other devices, ease of use, cost etc.

And fourth and finally, although it seems something more obvious by the number of thesis of this type in which it is wrapped, it is important to choose which programming language to use.

5.2 Study of alternatives

In this section is presented the study of the alternatives that have been analyzed for the development of this thesis. As mentioned above, which measures control, which wireless protocol use, which microprocessor select and also in which programming language program.

5.2.1 Measures to control

This section could analyze infinite measures but obviously not all have much sense for the behavior in a home. That is why focuses more on important control points ahead of a number of benefits discussed above.

- **Temperature**

  First, one of the most important control measures in a home is the temperature. In this case the two most important measures would be to measure are the temperature outside and inside the house but can also be interesting in certain situations measuring the temperature of a swimming pool or boiler.

  This allows control over the temperature to control a home HVAC system as well as a system of opening and closing windows.

  The temperature sensor used in such cases is called thermostat. Obviously in this thesis the prototype that has been done is on a very small scale so the sensor is simpler as it will be discussed in subsequent chapters. Some examples of temperature sensor are shown in Figure 4. This are the final prototypes in which in each one it is able to find a thermopar, thermoresistor and show on.

Figure 4: Temperature sensor

- **Lighting**

Another of the most important steps in a home can be the lighting system. Thanks to this it is possible to control the amount of sunlight entering the home or the lighting system itself.

This allows you to control the intensity of light from each bulb or power LED and so regulate the lighting or also control the blinds depending on the amount of sunlight entering through the window. Some examples are shown in the Figure 5.

Figure 5: Lighting sensor

- **Smoke**

This sensor can warn of the existence of uncontrolled fire somewhere in the house, especially in the kitchen. This not only sensor used in home automation since it is widely used in any security system manufactured. An example is shown in the Figure 6.

Figure 6: Smoke sensor

- **Presence and movement**

This type of sensor is known worldwide as it has many applications. It is used in so common in passageways, public toilets, garages, etc. It allows controlling a particular point of light and thanks to this to know if it is occupied the area of some point to actuate the valve, marking the availability of a parking garage or simply as security measures. An example is shown in the Figure 7.

Figure 7: Presence and movement sensor

5.2.2 Remote communication protocols

In this section the possible forms of communication studied in terms of the realization of the thesis are analyzed. To this is taken into account that the desired communication is remote protocols so will be wireless.

This thesis is aimed at short-distance remote control, it is important to talk about it because the distance is very important on the issue of remote topic.

The following figure shows some different areas depending on the distance, also called networks. These are HAN (Home Area Network), LAN (Local Area Network) and WAN (Wide Area Network). For this thesis, especially since study the HAN looking for a home environment (Figure 8).

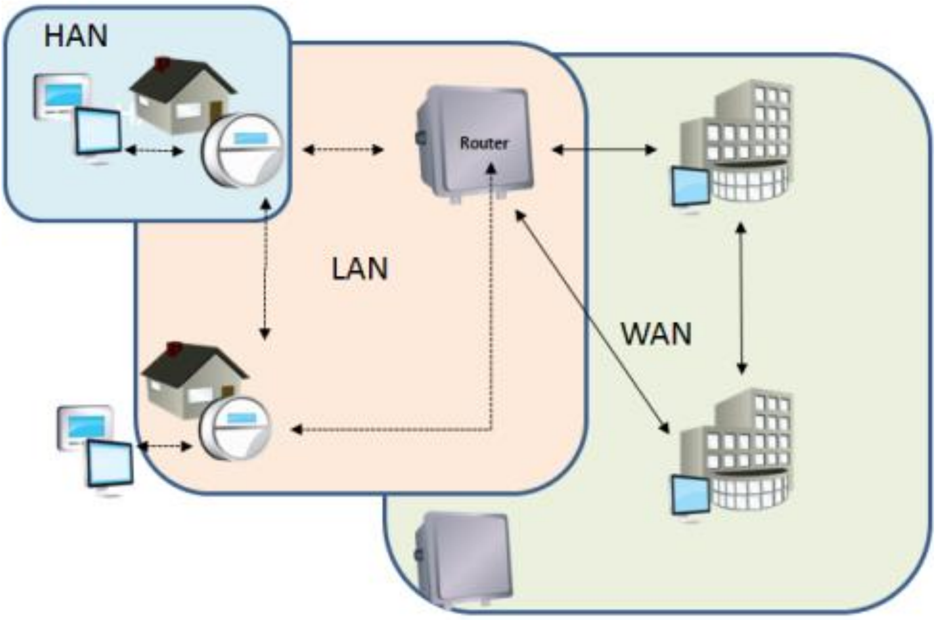

Figure 8: Classification of Area Network

In this case the protocols are the following studied. Bluetooth, ZigBee and 6LoWPAN.

- **Bluetooth**

  Bluetooth is a common wireless communications system used to exchange a short amount of data over short distances. Bluetooth is born to meet such requirements offering safe, relatively fast and economic links. Enables voice and data transmission between different devices using a radio frequency link in the Industrial Scientific and Medical (ISM) band of 2.4 GHz. Concretely the IEEE standard for Bluetooth is IEEE 802.15.1.

  The main objectives of this standard are relation with the list shown on the following lines:

  - Facilitate communications between mobile and fixed equipment.

- Provide high speed in short communication.
- Remove cables and connectors between devices.
- Be easily adaptable for all users.
- Offer the possibility to create small wireless network.
- Provide data synchronization between personal computers.
- Be a small scale and low cost technology.

From its beginnings, the specification has been growing up continuously, causing a number of versions. The following table shows the evolution of versions and its bandwidth.

Table 1: Versions of Bluetooth

Version	Bandwidth
Version 1.2	1 Mbit/s
Version 2.0 + EDR	3 Mbit/s
Version 3.0 +HS	24 Mbit/s
Version 4.0	32 Mbit/s

According to the transmission power there are different classes that provide different range and output power. The following table shows the mentioned classes:

Table 2: Classes of Bluetooth

Class	Maximum output power (mW)	Maximum output power (dBm)	Range (m)
Class 1	100	20	~ 100
Class 2	2.5	4	~ 5 - 10
Class 3	1	0	~ 1

It has the ability to transmit in full duplex with a maximum of 1600 hops/s. Frequency hopping are given a total of 79 frequencies with intervals of 1 MHz.

There are two topologies used in the Bluetooth i.e. Piconet and Scatternet. A Piconet is formed by a Wireless Personal Area Network (WPAN) in which a mobile device is acting as a master and other mobile devices will be serving as slaves. A Scatternet consists of two or more Piconets. Each Piconet as maximum can have 8 nodes, 1 master and 7 slaves. It can be point to point or point to multi-point(Figure 9).

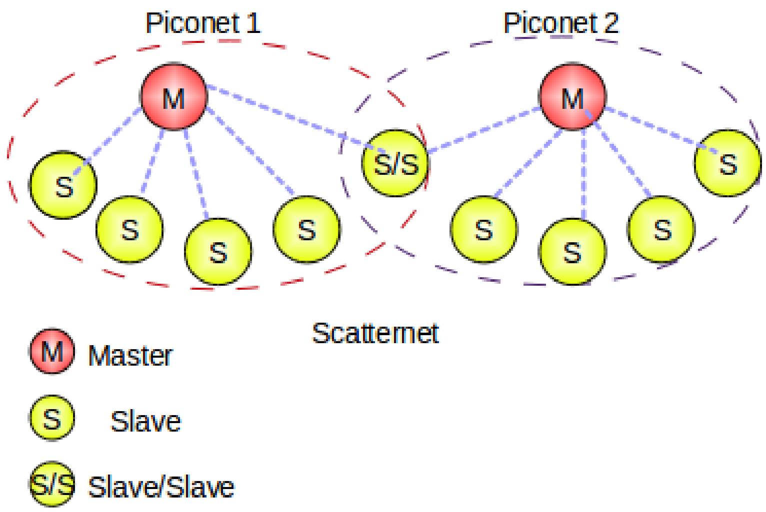

Figure 9: Bluetooth topologies

- **6LoWPAN**

6LoWPAN is the IETF working group formed in 2005. It enables IEEE 802.15.4 and IPv6 to work together in order to achieve IP enabled low power networks of small devices including sensors, controllers etc.

The main focus of the IETF working group, was to optimize the transmission of IPv6 packets over low-power and lossy networks, specifying:

- Header compression, which compresses the 40-byte IPv6 and 8-byte UDP headers by assuming the usage of common fields.
- Fragmentation and reassembly.
- Stateless auto configuration, which is the process where devices inside the 6LoWPAN network automatically generate their own IPv6 address.
- Enables a standard socket API.
- Minimal use of code and memory.
- Point to point integration with internet.

There are some benefits that this technologies provide. They are shown in the following list.

- Open and reliable standard.
- Easy learning curve.
- Transparent integrity with Internet.
- Network maintenance.
- Global scalability.
- End to end data streams.
- It use existing infrastructure.

But also it has too many challenges because is being developed by the moment. The architecture in 6LoWPAN are stub networks, it means that they have no knowledge of other networks, no carry extraneous traffic through them and to communicate with other networks have certain points of departure defined (Figure 10).

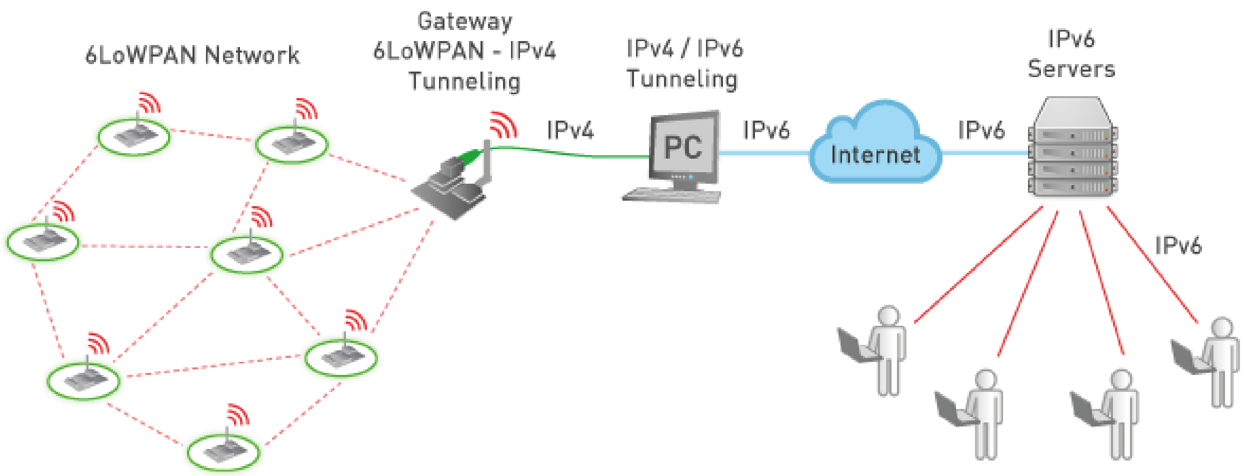

Figure 10: 6LoWPAN network

- **ZigBee**

ZigBee is the name of the specification of a set of high-level protocols of wireless communication for use with low-power digital broadcasting, based on the IEEE 802.15.4 wireless personal area networks. They target applications that require secure communications with low rate shipping data and maximizing the life of batteries. This technology can be used broadly in device control, reliable messaging, home and building automation, consumer electronics, remote monitoring, health care, and many other areas. It is a low power network provided all devices are interconnected by IEEE 802.15.4 links with Direct Sequence Spread Spectrum (DSSS), it is a channel coding method, before modulation, in spread spectrum digital signal for transmission over radio waves.

ZigBee uses the ISM band, concretely, 868 MHz in Europe, 915 in the United States and 2.4 GHz worldwide. However, when designing devices, the companies almost always opt in the 2.4 GHz band, simply because it is free worldwide. The technology development is focused on simplicity and low cost than other WPAN wireless networks, such as Bluetooth. The Zig-Bee fuller node requires in theory near 10% of the hardware of a Bluetooth or Wi-Fi typical node, this number falls to 2% for the simplest nodes, and the code size is about 50% the size the Bluetooth.

Estimated data rates are 250 kbps per channel in the unlicensed 2.4 GHz band, 40 kbps per channel in the 915 MHz band and 20 kbps per channel in the 868 MHz band. ZigBee supports 10–75 m point to point, typically 30 m indoor and unlimited distance with mesh networking. In a mesh network each node can be reached by multiple links and connections are dynamically updated and optimized. Mesh networks are de-centralized and each node can manage itself in the changing conditions and is able to dynamically self-route and connect with new nodes as needed. These features offer scalability, greater stability and tolerance against node/link failures. This along with low power utilization and low deployment cost makes ZigBee very attractive for the smart grid HAN applications.

There are three different types of ZigBee devices according to their role in the network:

- ZigBee Coordinator (ZC). The type of more complete device. There must be at least one in the network. Its functions are in charge of controlling the network and the paths to be followed by the devices to connect with each other. Full Function Device (FDD).

- ZigBee Router (ZR). Interconnect separate devices in the topology of the network, in addition to offering a level of application for the execution of user code devices.
- ZigBee End Device (ZED). It has the functionality needed to communicate with its parent node (coordinator or router node), but it cannot transmit information to other devices. Thus, this type of node can be asleep most of the time, increasing the average life of its batteries. A ZED has minimum memory requirements and is therefore significantly cheaper. Reduced Function Device (RFD) (Figure 11).

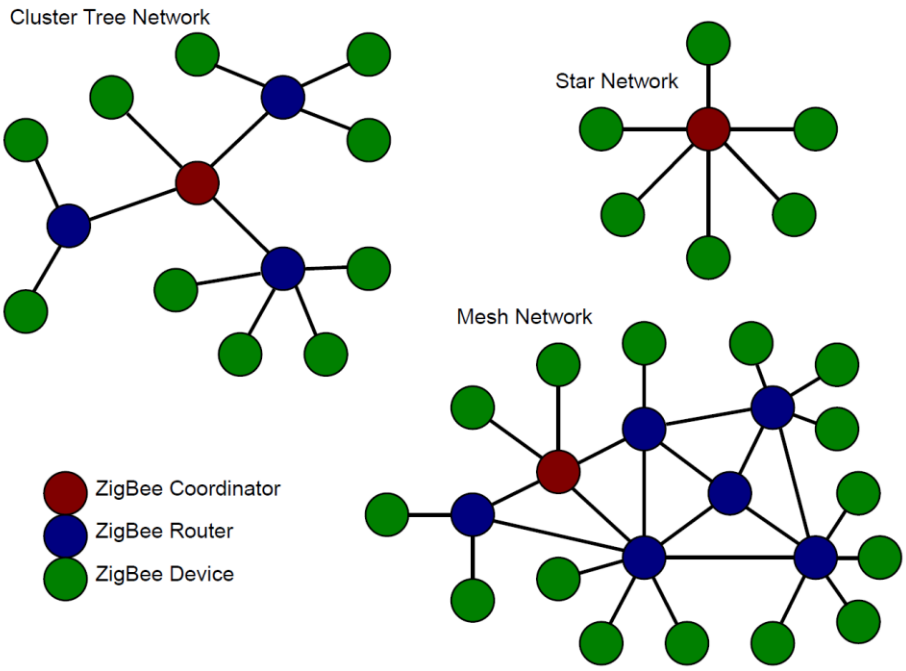

Figure 11: ZigBee Network

5.2.3 Implementation boards

In this section, the possible implementation boards are analyzed. For the election it is important to note compatibilities between each one, the price of each, etc. For the implementation of this thesis it has taken into account the use of two microprocessors so that one of them is used in master mode and the other in slave mode. Boards that were studied are as follows.

- **MBED**

MBED is a platform for rapid prototyping and experimentation with microcontrollers ARM Cortex-M3 and Cortex-M0 ARM 32-bit. The ARM Cortex-M3 microcontroller has the following basic characteristics:

- NXP LPC1768 Microcontroller
- ARM® Cortex™-M3 high performance
- 96MHz, 32KB RAM, 512KB FLASH
- Ethernet, USB Host/Device, 2xSPI, 2xI2C, 3xUART, CAN, 6xPWM, 6xADC, E/S general purpose (GPIO)

While the characteristics of ARM Cortex -M0 are:

- NXP LPC11U24 Microcontroller
- ARM® Cortex™-M0 low power consumption
- 48MHz, 8KB RAM, 32KB FLASH USB Device, 2xSPI, I2C , UART, 6xADC, E/S general purpose (GPIO)

In both cases the compiler is online with the new paradigm of cloud applications known as EDI. It also offers libraries for fast and flexible prototyping platform C / C ++, moreover libraries for peripherals that simplify immediate use and drivers of analogs and digital I/O, and serial UART, I2C, SPI, etc.

In the next picture, Figure 12, pins that have this type of board is shown.

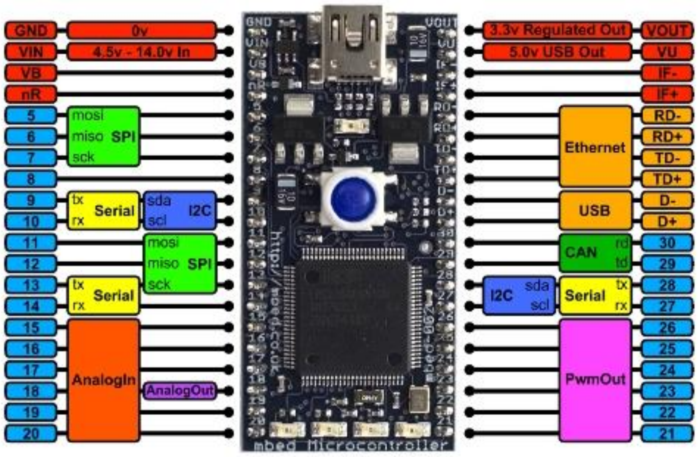

Figure 12: MBED pins

- **MSP430G**

The main features of the microcontroller MSP430G are as follows:

- Easy to use, it includes all hardware and software needed to get started. Two pre microcontrollers programmed and code examples to help users get started quickly.
- Accessible, LaunchPad includes a development board, 2 microcontrollers MSP430 programmable mini USB cable, PCB connectors for expandability, external crystal for accuracy clock and software integrated development environment (IDEs) unrestricted, everything needed for a fast start.
- Scalable, The LaunchPad provides a simple introduction to microcontrollers MSP430 family. If an application requires changes, the programs developed in the LaunchPad they can be migrated to more sophisticated MSP430 devices.

In the next picture, Figure 13, pins that have this type of board is shown.

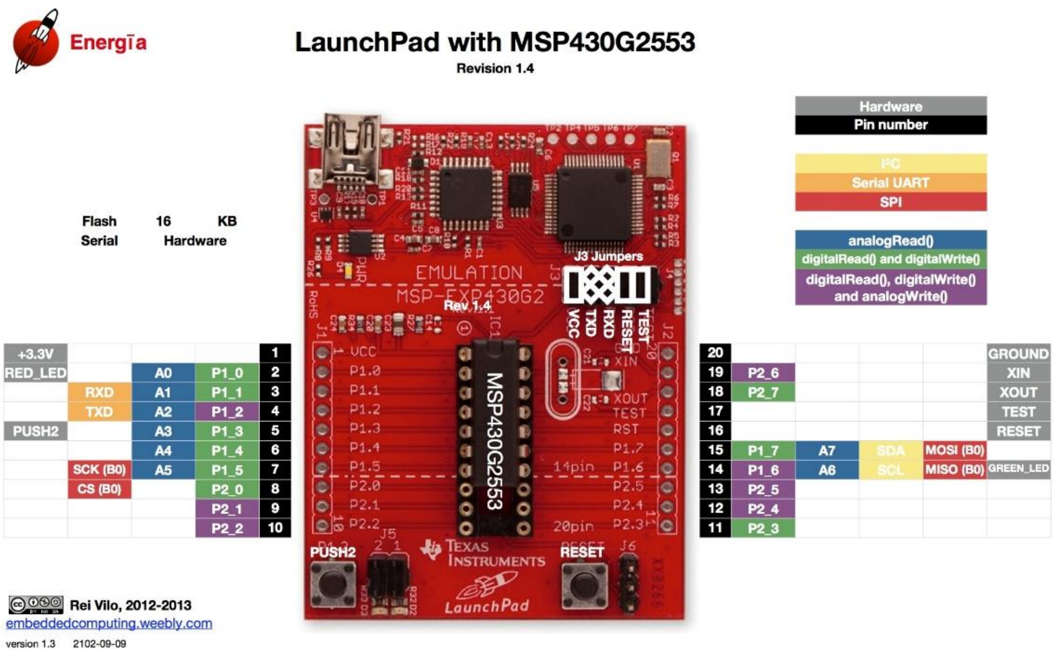

Figure 13: MSP430G pins

- **Arduino UNO**

Arduino Uno is a microcontroller board based on the ATmega328 microcontroller. It has 14 digital inputs / outputs, of which 6 can be used as PWM (pulse-width modulation) outputs and other 6 are analog inputs. It also includes a 16 MHz ceramic resonator, a USB connector, a supply connector, one ICSP header and a reset button. The board includes everything needed for the microcontroller to do its job, just plug it in to a computer with a USB cable or electrical current through a transformer.

The main features of the Arduino UNO are as follows:

- Flash Memory: 32 KB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

In the next picture, Figure 14, pins that have this type of board is shown.

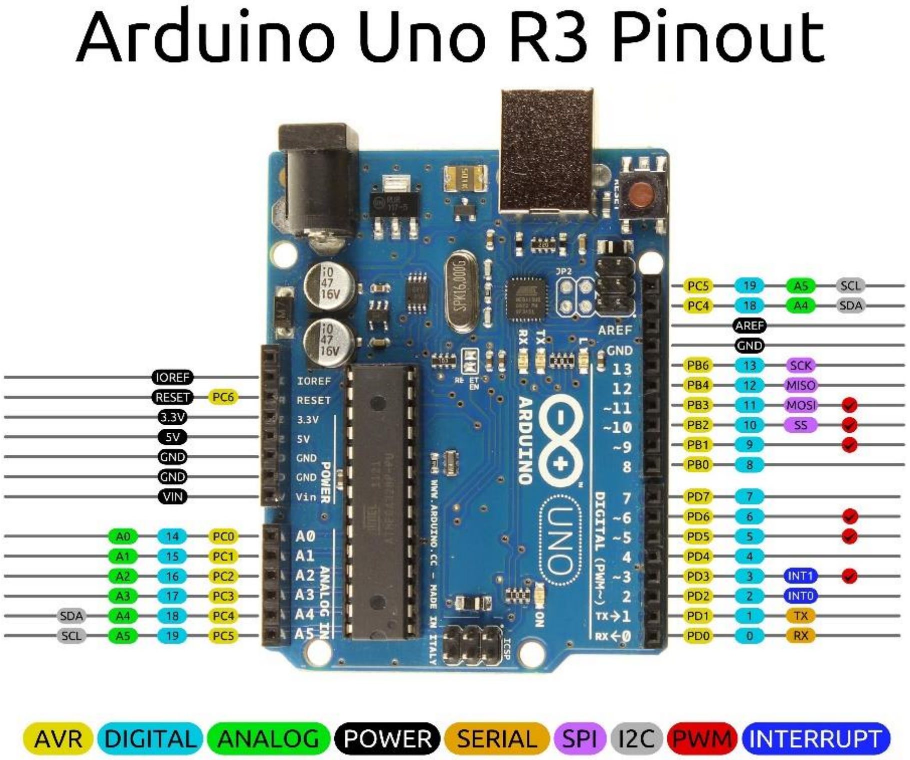

Figure 14: Arduino UNO pins

- **Freescale Freedom KL257**

The Freedom board is based on a Kinetis microcontroller core ARM Cortex-M0 + of the Freescale company that runs 48 MHz, has 128KB of program memory and 16KB of RAM.

The Freedom plate Freescale is a low-cost development platform for Kinetis L microcontrollers, Series KL1x and KL2x MC-core ARM® Cortex ™ -M0 +. The board includes easy access to I/O MCU, operation of low power consumption, a standard factor with expansion options of the board and debug interface included for programming Flash and execution control. Freedom is supported by software development Freescale and other companies.

The main features of the Freedom KL25Z are as follows:

- 128 KB Flash
- 16 KB SRAM
- 64 B Cache
- Max frequency of CPU: 48 MHz

- Real Time Clock (RTC)
- Watchdog
- 4 channels DMA
- 12 bits ADC
- Analog Comparer (6 inputs)
- USB full-speed Controller
- Low power consumption UART
- 2 x UART

In the next picture, Figure 15, pins that have this type of board is shown.

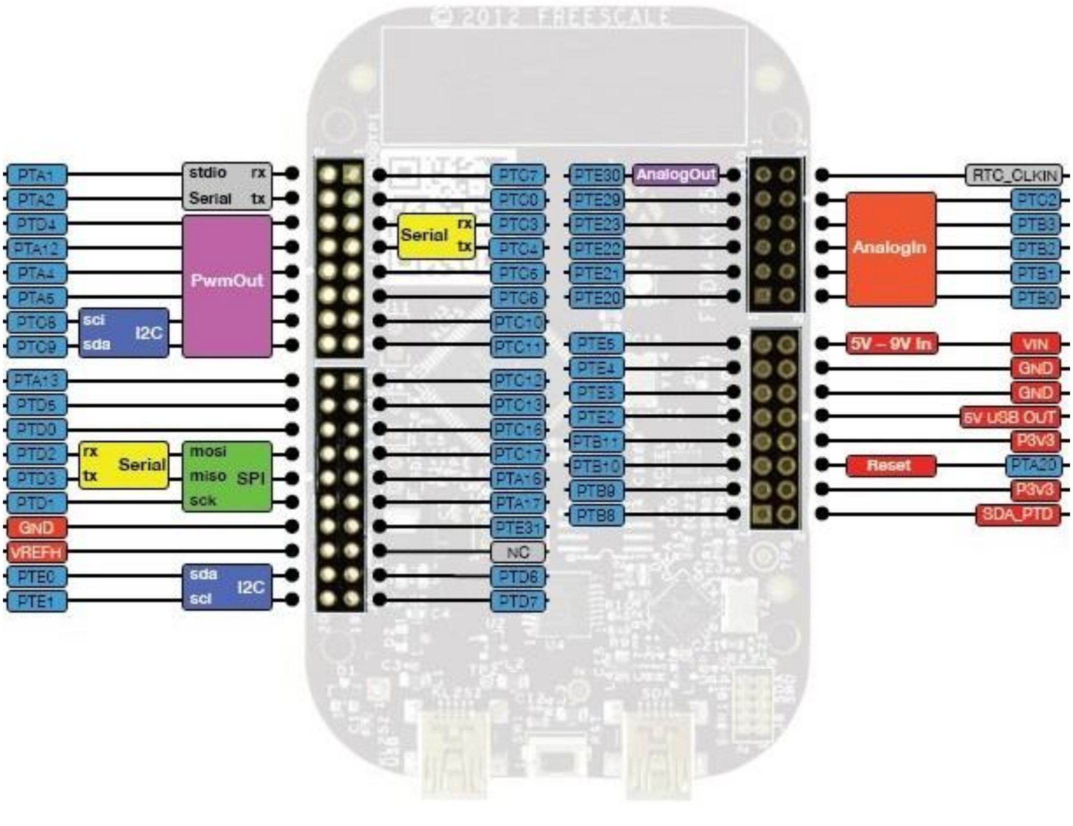

Figure 15: Freedom KL25Z pins

5.2.4 Programming language

This section will analyze the different possibilities that exist talking about the programming language used. In this case it seems obvious that the language to be used will be the C but it is interesting to note some variable to make the thesis more complete.

In this case the languages studied rely on the plates used since the plates are not compatible with all programming languages. In this case because of the studied plates can be highlighted three different languages, two medium level while the latter is a low level. The languages studied are C, C ++ and assembly language.

- **C**

C is a language oriented to the implementation of operating systems. It has a simple core language, with important functionalities added, as mathematical functions and management of files, provided by libraries. It is very flexible, allows you to program with multiple styles. In addition, allows access to low-level memory by using pointers. It is one of the languages most used programming today and it is very structured because they often use different functions for each task.

- **C++**

C ++ is a really similar that its predecessor C language in which certain qualities and characteristics are added that it lacked. It is said that C ++ is a multi-paradigm language because it has facilities generic programming, structured programming and object-oriented. But everybody knows as a imperative object-oriented language.

Currently, the C ++ is a versatile, powerful and general language. Its success among professional programmers has led him to take the lead as application development tool. The C ++ maintains the advantages of C in wealth of operators and expressions, flexibility, conciseness and efficiency. It has also removed some of the difficulties and limitations of the original C.

- **Assembler**

Assembly language is a programming language for low-level computers, microprocessors, microcontrollers and other programmable integrated circuits. Implements a symbolic representation of binary machine codes and other constants needed to program a given CPU and is the most direct representation of specific machine code for each architecture readable by a programmer architecture. It is based on mnemonics that symbolize processing steps (instructions), processor registers, memory locations and other language features.

5.3 Selection of alternatives

Finally, this section the criteria taken into account in the selection of these alternatives will be discussed. First, in this thesis has been chosen to control the temperature and lighting for several reasons. One of them is that sensors used for prototyping i.e. small scale are very cheap. Besides these two measures can have a very significant control at home and with the help of some actuators can be deployed improved systems as those discussed before of controlling blinds, windows, etc.

In the case of the selection of the protocol used there are more doubts because each module used can be any restrictions or unique feature so you will have to study in detail. This is why that the reason of why has been chosen Bluetooth as a protocol for remote communication is the availability of modules in the laboratory as well as the cost of these programs and facilities offered by the manufacturer because it is much easier to find modules Bluetooth than ZigBee .

In the case of the boards used in this thesis has decided to use MBED and MSP430G, the main reason has been the good compatibility that they have. Also is important to mark the simplicity of getting a compiler, in the case of MBED it is free and online and in the case of MSP430G it is called “Energia” which is also free.

Finally, we have chosen the programming language C / C ++ is a combination of two of the studied because it is simple and offers many possibilities.

6. METHODOLOGY

This section of the document will explain the overall design that will take place throughout the thesis. First, the system architecture will be explained, after the structure of the different modules that make up the prototype and the used products, finally, the operating modes and their implementation will be explained will be explained.

6.1 Architecture

As explained above the complete prototype is divided into 2 master and 2 slave modules. These modules can be seen in the following figure. This figure shows that there are two different types of connections, one is wireless Bluetooth connection whereas the other is the serial UART connection. The Bluetooth connection is among the master modules and slaves, whereas the UART connection is among one of the master modules with the PC. The block Diagram is also shown in Figure 16.

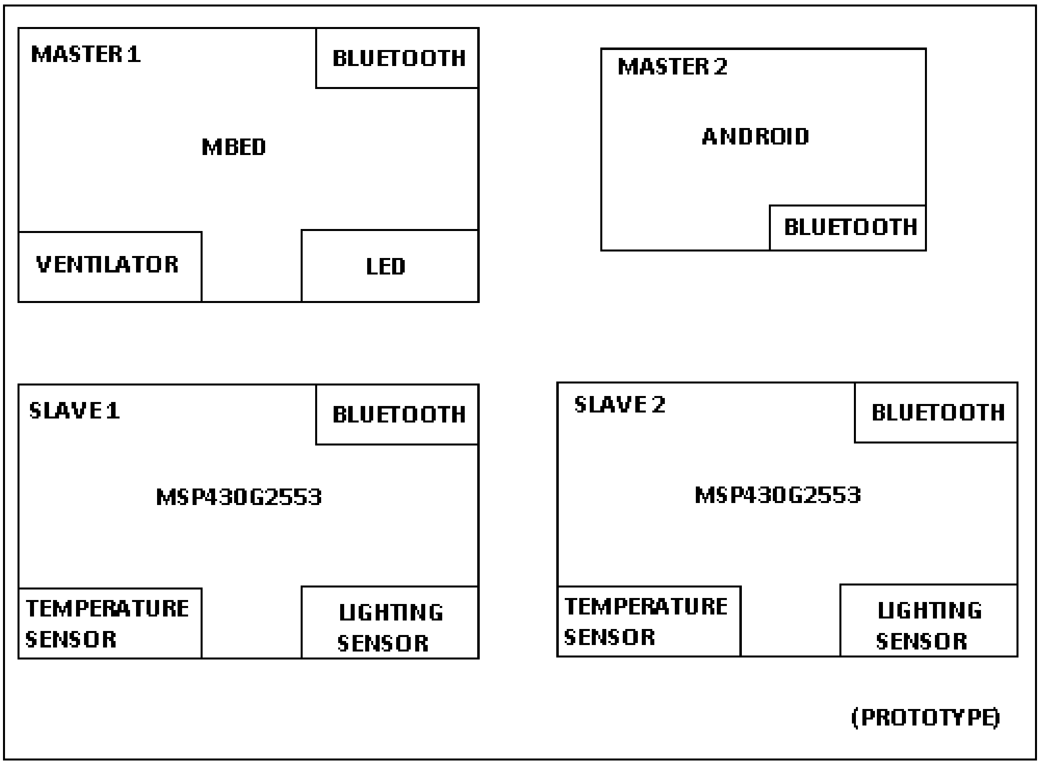

Figure 16: Block Diagram

6.1.1 MBED Master Module

The master module MBED is responsible for controlling the ventilator and lighting depending on what is the desired. As explained below, the designed system is able to work in automatic mode and manual mode. These two options can only be used in the case that the master module is used is the MBED master one. On the contrary, the one controlled by the master Blueterm android application can only use in manual mode.

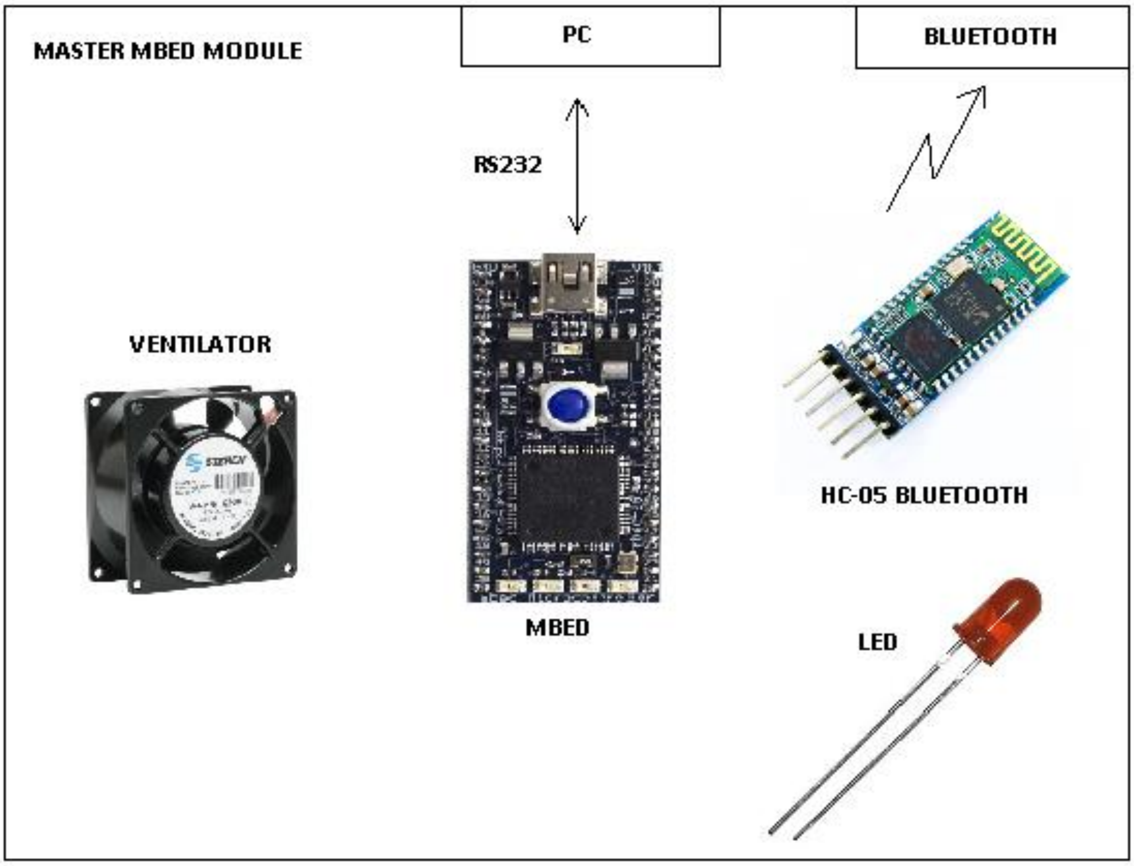

Figure 17: MBED Master Module Block Diagram

As shown in the figure 17 above, this module consists of 4 main parts, the first is the microcontroller that manages everything and attach it to control vary the intensity by the LED or fan in the desired time. The second is the HC-05 Bluetooth module that can connect with other MSP430 microcontroller explained below as well as to the android terminal to control the fan and LED. And finally, it is possible to see the two devices that can be control, the ventilator to lower the temperature of the room, and the LED to control the lighting of the room.

6.1.2 Blueterm Master Module

This master module simply consists on an android application that allows connecting via Bluetooth with other modules and therefore allow some remote control. In this thesis it was used to control the ventilator and the LED module previously explained. Previously you can connect with any slave modules that are explained in the following paragraphs to read the desired sensors and can thus take appropriate decisions (Figure 18).

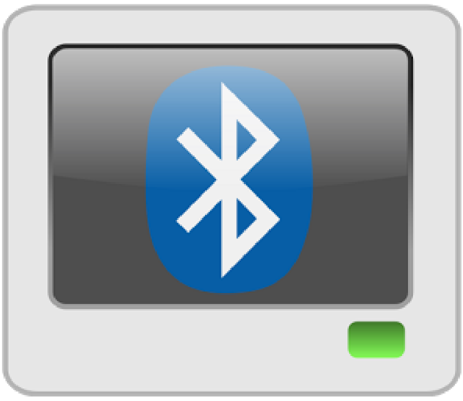

Figure 18: Blueterm

6.1.3 Slave Module

In this case the two slave modules are identical so only one is explained. The following figure shows the different parts that make these slave modules. This is shown in the Figure 19.

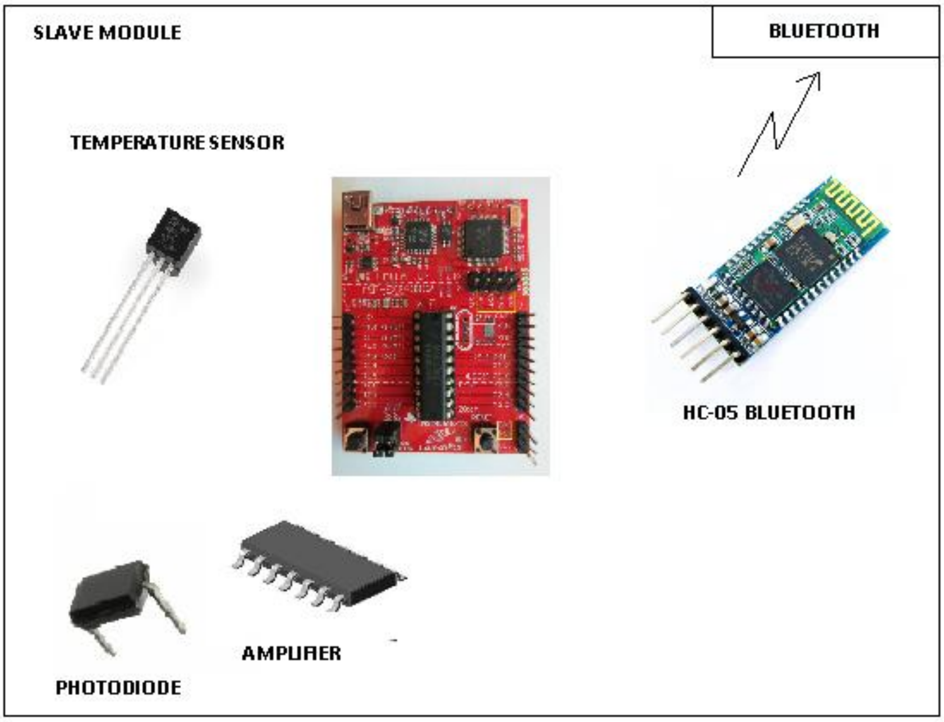

Figure 19: Slave module

The main function of this module is to collect the values of temperature and lighting through the needed sensors. This data is sent to either master modules for them can control both the fan and the lighting. As it is shown in the figure there is an amplifier near the photodiode because the voltage of this is too low.

6.2 Implementation

In this section is explained in a more detailed way the devices, cables and elements that have been used for the development of the thesis. In some cases the technical aspects have already been explained in the section analysis of alternatives so it would not be explained again. In addition, the code programmed into each of the both microcontroller is provided.

6.2.1 MBED Microcontroler

MBED is one of the two microcontrollers of the thesis. This is working in the master module. Through this microcontroller the prototype connect with PC via serie with a USB cable. In addition, you can connect via Bluetooth with the slave modules. Finally, it is responsible for controlling both the temperature (controlling ventilator) and the lighting (controlling de LED) in the room. The following table shows simply inputs and outputs of this microcontroller.

Table 3: I/O MBED

Name	I/O	Pin
Serial pc RX	I	USBRX
Serial pc TX	O	USBTX
Serial Bluetooth RX	I	P9
Serial Bluetooth TX	O	P10
Led		
Led iluminacion		P21
ventilador		

6.2.2 MSP430G2553 Microcontroler

This microcontroller is in both slave modules and it can read the two data that are of interest in this thesis, temperature and lighting through sensors. This data will be sent via Bluetooth to both master module when the masters ask for the data. The following table shows simply inputs and outputs.

Table 4: I/O MSP430G2553

Name	I/O	Pin
Serial Bluetooth RX	I	P1.1
Serial Bluetooth TX	O	P1.2
Analog read photodiode	I	P1.5
Led 1	O	P1.0

Led 2	O	P1.6

6.2.3 Bluetooth module HC-06

This Bluetooth module allows the design to communicate remotely both microcontrollers. For this thesis, there is on module configured as master on the MBED module and two configured as a slave in each MSP430 module. Its use is simple when you only need to match it with one of the modules, but when you want to pair with more than one the use become more complicated because it is not possible for a master module has several matched to the slaves simultaneously. To solve this problem a class is created. This class depending on which sensor is wanted to read allow the master module pairing with the desired slave (Figure 20).

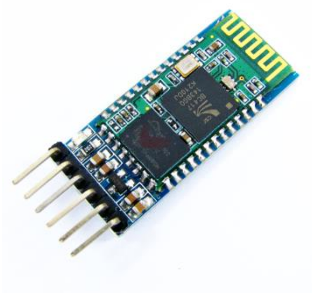

Figure 20: HC-05 module

The way to configure it in each case is explained below. It is required to send a series of commands.

Is important to advice that there are some differences between configuring a slave module and a master one. First, it is explain how to configure a slave module. The following table shows how to connect the HC-05 slave module with USB TLL device.

Table 5: Connection between Slave HC-05 and USB TTL

SLAVE HC-05	USB TTL
RX	RX

TX	TX
+5v	VCC
GND	GND
+3.3 v	KEY

Once connected properly, a serial communication program is used such as Tera Term, the baud rate is set to 38400 and in this moment is possible to start sending commands. The following table shows the commands required to configure the slave as well as its explanation.

Table 6: Commands for Slave module

Command	Explanation
AT+NAME?	To get the name of the module.
AT+NAME=NAME	To change the name of the module.
AT+PSW?	To get the password of the module.
AT+PSW=PSW	To change the password of the module.
AT+ROLE?	To get the role of the module.
AT+ROLE=0	To chance the role of the module. “0“ for SLAVE, “1“ for MASTER.
AT+ADDR?	To get the address of the module.

After explaining how to configure the slave module, how to configure master module is explain. The following table shows how to connect the HC-05 master module with USB TLL device. Note the cross connection here.

Table 7: Connection between Master HC-05 and USB TTL

SLAVE HC-05	USB TTL
RX	TX
TX	RX
+5v	VCC
GND	GND
+3.3 v	KEY

Table 8: Commands for Master module

Command	Explanation
AT+ORGL	Restores the module to original state.
AT+NAME?	To get the name of the module.
AT+NAME=NAME	To change the name of the module.
AT+PSW?	To get the password of the module.
AT+PSW=PSW	To change the password of the module.
AT+ROLE?	To get the role of the module.
AT+ROLE=1	To chance the role of the module. “0“ for SLAVE, “1“ for MASTER.
AT+ADDR?	To get the address of the module.
AT+RMAAD	Remove any previous PAIR condition.
AT+CMODE =1	“1“Allows connecting to any address, “0“ just to bound address
AT+INIT	To initialize the SPP profile library.
AT+INQ	To start searching bluetooth devices.
AT+LINK=<address>	To connect to specify module.

6.2.4 Ventilator

The ventilator used to simulate the control room temperature is a simple one. Depending some conditions it will be turn on or off. This will use to warm the temperature of the room (Figure 21). This ventilator is going to work between 600-2000 RPM (revolutions per minute) because is just for the prototype. For the one that will install in the house it is going to be totally different.

Figure 21: Ventilator

6.2.5 LED

The LED used to simulate the control room lighting is a simple and inexpensive LED. This LED is connected between the analog output value and a resistance to ground. Depending on the value that is set as analog output that shine more or less (Figure 22).

Figure 22: LED

6.2.6 Temperature Sensor ADC10

The temperature sensor used in this thesis is incorporated to the microcontroller MSP430G2553, it is called ADC10. The value offered is not very accurate but it serves completely for the scale on which this thesis works.

6.2.7 Lighting Sensor

The sensor used to measure the lighting is a simple photodiode. The use of an operational amplifier is necessary because the voltage in the photodiode is too low. With this amplifier the voltage is grow to readable values (Figure 23).

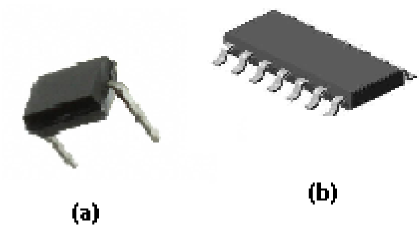

Figure 23: Lighting sensor (a) photodiode (b) amplifier

6.2.8 USB TTL

As it has mentioned in a previous section this device will serve to configure the Bluetooth modules in master or slave mode (Figure 24).

Figure 24: USB TTL

6.3 Use modes

In the next section the operating modes of design are explained. As can be seen in the figure below, it is possible to differentiate some states, thus creating a kind of machine states depending on what the user entered. The diagram shown in the following Figure 25:

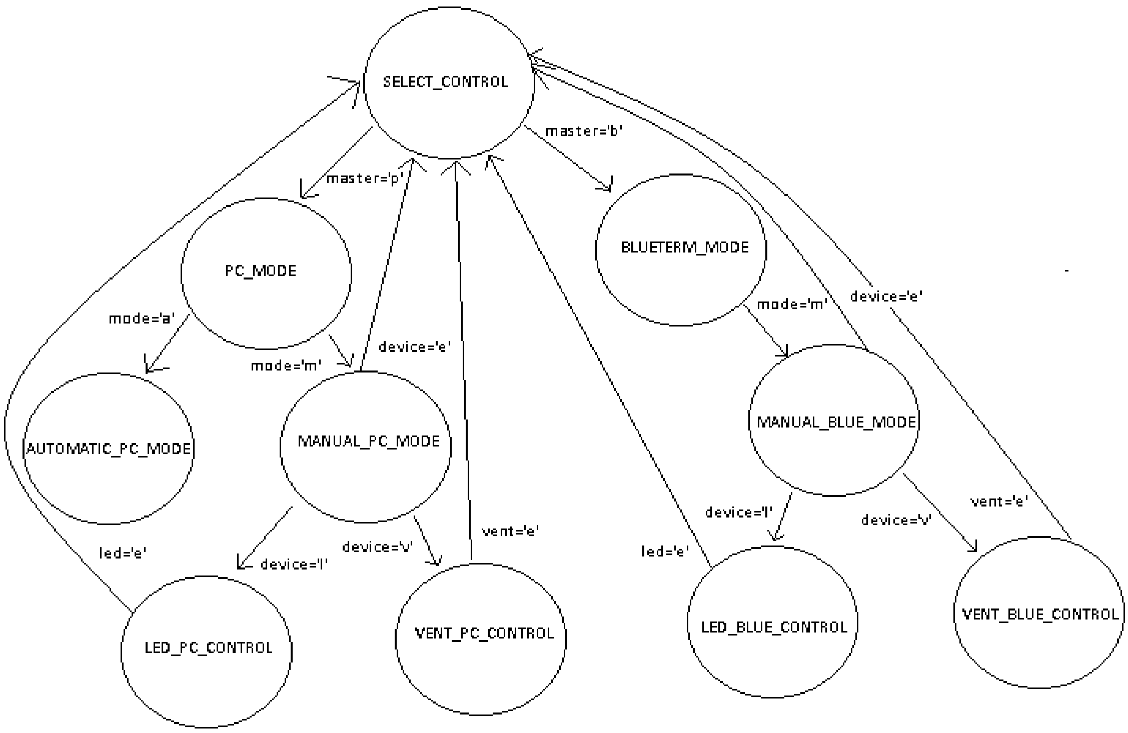

Figure 25: Operating modes Diagram

When the program starts, it is in the initial state "Select_Control" where you have to choose the master module with which you want to control the system. That is, you have to choose if it is controlled from the PC or the mobile application Blueterm. For making this possible a menu indicates the options and the need to access one or another command. In this case the variable that controls this is called "master". If this variable is given the value "p" is passed to control the system from the PC, however, if the given value is "b" the system is controlled by the mobile terminal.

Next, the possibilities that exist in the event that is controlled from the PC are explained because there are some differences from the mobile application.

Once in PC mode must be selected if desired controlled manually or automatically. In manual form two "states" are distinguished, the first one controls the LED. This one allow to control the Brightness. The second one controls the ventilator to do the same with temperature. These two states operate in a similar way.

The case of the state that controls manually the LED offered the following options showed in the following list:

- Led='1': Turn on the LED to100% of brightness (ON).
- Led='0': Turn off the LED (OFF).
- Led='u': Upload the brightness one tone (brigthter).
- Led='d': Download the brightness one tone (dimmer).
- Led='x': Ask for brightness data.
- Led='e': Come back to "Select Control" state.

The case of the state that controls manually the ventilator offered the following options showed in the following list:

- Vent='1': Turn on the ventilator (ON).
- Vent ='0': Turn off the ventilator (OFF).
- Vent ='z': Ask for temperature data.
- Vent ='e':Come back to "Select Control" state.

Now the automatic control is explained. In this case, the user will not do anything and the system itself will they realize everything. This will read the respective data both temperature and brightness and acts depending on the values of one form or another. If the temperature is read is greater than 30 °C the fan is automatically activated. As for the brightness, the LED lights up depending on the percentage to be read at the receiver, this will be inversely proportional. For example, if you read that there is an 80% brightness the LED will be 20%.

It the following paragraph it will explain the operation when control is from the mobile application. In this case there is only manual option that is very similar to the control way from the PC. The difference is that you have to change the Bluetooth pairing manually. This is because it must be connected to modify the values of both the LED and the ventilator in the master module and to request the data from the slave module.

6.4 Codes

6.4.1 MBED Code

```
/*
This Program makes the menu for moving around the desired states to read both measures, or
to chance some parameters in outputs.
*/
#include "mbed.h"


Serial pc(USBTX, USBRX);
Serial blue(p9, p10);


AnalogIn Ain(p20);


DigitalOut myled(LED1);
DigitalOut myled2(LED2);
DigitalOut myled3(LED3);
DigitalOut myled4(LED4);
//OUTPUT SIGNAL
PwmOut PWM1(p21); //LED
```

```
PwmOut PWM2(p22);//VENT
```

```
//FUNCTIONS FOR SELECT THE MASTER CONTROLLER AND THE STATE MACHINE
void Select_control();
void BlueTerm_mode();
void PC_mode();
void Automatic_pc_mode();
void Manual_blue_mode();
void Manual_pc_mode();
void LED_blue_control();
void LED_pc_control();
void Ventilator_blue_control();
void Ventilator_pc_control();

//FUNCTIONS FOR READ THE PHOTODIODE DATA
float read_brightness_pc();
float read_brightness_blue();

//FUNCTION FOR READ THE TEMPERATURE DATA
float read_temperature_pc();

//FUNCTION FOR CLEAN THE BUFFERS
void flushSerialBuffer();

//VARIABLES FOR THE STATE MACHINE
char master;        // char for choose the master --> PC or BlueTern
char mode;          // char for choose the operation mode --> automatic or manual
char manual_mode;   // char for choose the device to control --> Ventilator or LED
char ex;            // char for exit in automatic mode
char device;        // char for choose the desirable device
char led;           // char to control the LED
```

```
char vent;              // char to control the ventilator


//VARIABLES FOR READ THE PHOTODIODE DATA
float light;
float light1;
float light_per;
float light_aux;


//VARIABLES FOR READ THE TEMPERATURE DATA
int temp_sensor[4];
float temp;
float temp1;
float temp_dec;
char d_sensor;
int i;


// VARIABLES FOR CONTROLLING THE OUTPUTS
float brightness=0.0;   // float to control the brightness of the LED


//MAIN
int main()
{
   blue.baud(38400);
   pc.baud(9600);


   while (1)
   {
     Select_control();
   }
}
```

```
void Select_control(){

   pc.printf("\r\nBluetooth Start: PC mode=p, Blueterm mode=b \r\n");
   blue.printf("\r\nBluetooth Start: PC mode=p, Blueterm mode=b \r\n");

         //master = blue.getc();
         wait(0.001);
         master = pc.getc();
         wait(0.001);

         if (master=='b'){             // BlueTerm mode
             BlueTerm_mode();
          }
          if (master=='p'){             // PC mode
             PC_mode();
          }
   }

void BlueTerm_mode()
{
   blue.printf(" \r\nBlueTerm Mode: Manual mode=m\r\n");
   //myled=1;
   mode = blue.getc();
   wait(0.001);

   if (mode=='m'){             // Manual mode
      Manual_blue_mode();
      }
   }
```

```
void PC_mode()
{
   pc.printf(" \r\nPC Mode: Manual mode=m, Automatic mode=a  \r\n");
   //myled2=1;
   mode = pc.getc();
   wait(0.001);
   if (mode=='a'){              // Automatic mode
      Automatic_pc_mode();
      }
   if (mode=='m'){              // Manual mode
      Manual_pc_mode();
      }
   }

void Automatic_pc_mode (){
   while(device!='e'){
   pc.printf("\r\n You have selected the Automatic mode:\r\n");
   temp1=read_temperature_pc();
   light1=read_brightness_pc();
   light_aux=light1/100;
   PWM1=1.0-light_aux;
   if (temp1>27.00){
      PWM2=1.0;
      }
   else {
      PWM2=0.0;
   }
   device=pc.getc();
   wait(0.001);
   }
   Select_control();
```

```
  }
void Manual_blue_mode ()
{
  device='y';
  while(device!='e'){
  blue.printf("\r\nManual mode: LED control=l, Ventilator control=v  \r\n");
  device= blue.getc();
  wait(0.001);
  if(device=='l'){
   LED_blue_control();
   }
  if (device=='v'){
   Ventilator_blue_control();
   }
   }
   Select_control();
  }
void Manual_pc_mode ()
{
  device='y';
  while(device!='e'){
  pc.printf("\r\nManual mode: LED control=l, Ventilator control=v  \r\n");
  device= pc.getc();
  wait(0.01);
  if(device=='l'){
   LED_pc_control();
   }
  if (device=='v'){
   Ventilator_pc_control();
   }
   }
```

```
    Select_control();
  }
void LED_blue_control (){
  led='y';
  blue.printf("\r\nLED control: ON=1, OFF=0, brighter=u, dimmer=d, read value=x \r\n");
    PWM1.period(0.001);
    PWM1=1.0;
  while(led!='e'){
    myled3=1;
    led=blue.getc();
    wait(0.001);
    if((led=='u')&&(brightness<1.0)){
      brightness+=0.1;
      PWM1=brightness;
      }
     if((led=='d')&&(brightness>0.0)){
      brightness-=0.1;
      PWM1=brightness;
      }
    if (led=='1'){
      PWM1=1.0;
      }
    if (led=='0'){
      PWM1=0.0;
      }
    if (led=='x'){
      light1=read_brightness_blue();
      }
  }
  Select_control();
  }
```

```
void LED_pc_control (){
   led='y';
   pc.printf("\r\nLED control: ON=1, OFF=0, brighter=u, dimmer=d, read value=x \r\n");
   PWM1.period(0.001);
   PWM1=1.0;
   while(led!='e'){
      led=pc.getc();
      wait(0.001);
      if((led=='u')&&(brightness<1.0)){
         brightness+=0.1;
         PWM1=brightness;
         }
       if((led=='d')&&(brightness>0.0)){
         brightness-=0.1;
         PWM1=brightness;
         }
      if (led=='1'){
         PWM1=1.0;
         }
      if (led=='0'){
         PWM1=0.0;
         }
       if (led=='x'){
         light1=read_brightness_pc();
         }
   }
   Select_control();
   }
void Ventilator_blue_control (){
   while(vent!='e'){
```

```
    blue.printf("\r\nVentilator control: ON=1, OFF=0, read data=z \r\n");
    vent=blue.getc();
    if (vent=='1'){
        PWM2=1.0;
        }
    if (vent=='0'){
        PWM2=0.0;
        }
    wait(0.001);
    }
    Select_control();
    }
void Ventilator_pc_control (){
    while(vent!='e'){
    pc.printf("\r\nVentilator control: ON=1, OFF=0, read data=z \r\n");
    vent=pc.getc();
    if (vent=='1'){
        PWM2=1.0;
        }
    if (vent=='0'){
        PWM2=0.0;
        }
    if (vent=='z'){
          d_sensor='5';
    blue.putc(d_sensor);
        temp1=read_temperature_pc();
        }
    wait(0.001);
    }
    Select_control();
    }
```

```
  float read_brightness_pc(){
  pc.printf("Photodiode value... \n\r");
  light=100*Ain;
    pc.printf("%f \n\r",light);
    if(light>100.0){
      pc.printf("Photodiode percentage is: \n\r");
      light_per=100.0;
      pc.printf("%0.2f \n\r",light_per);
      }
    else {
      pc.printf("Photodiode percentage is: \n\r");
      light_per=100-light;
      pc.printf("%0.2f \n\r",light_per);
      }
    wait (2);
    return light_per;
   }
float read_brightness_blue(){
  blue.printf("Photodiode value... \n\r");
  light=100*Ain;
    blue.printf("%f \n\r",light);
    if(light>100.0){
      blue.printf("Photodiode percentage is: \n\r");
      light_per=100.0;
      blue.printf("%0.2f \n\r",light_per);
      }
    else {
      blue.printf("Photodiode percentage is: \n\r");
      light_per=100-light;
      blue.printf("%0.2f \n\r",light_per);
      }
```

```
      wait (2);
      return light_per;
    }
float read_temperature_pc(){
myled3='1';
   int i=0;
   for(i=0;i<3;i++){
      temp_sensor[i]=blue.getc();
      }
   wait(2);
   temp_dec=(temp_sensor[2]-48)*10+(temp_sensor[3]-48);
   temp=(temp_sensor[0]-48)*10+(temp_sensor[1]-48)+temp_dec/100;
   pc.printf("The temperature is: ");
   pc.printf("%0.2f Celsius \n\r",temp);
   wait(0.5);
   flushSerialBuffer();
   return temp;
   }
void flushSerialBuffer() {
   char char1 = 0;
   while (blue.readable()) {
      char1 = blue.getc();
       }
    ; }
```

6.4.5 MSP430G2553 Code

```
/*
Notes:

This program is used to take the temperature data from the internal sensor of
```

```
the MSP430G2553 and send it via bluetooth.

*/

#include "io430.h"
#include "variables.h"
#include "funciones.c"
#include "interrupciones.c"

long ADCDATA;//Variable that receives the information of ADC10MEM
long TEMPERATURAC;//Variable of conversion to centigrades
int TABLA_TEMP[4]; // Array to send the temperature data to the MBED Master module
int TABLA_BLUE[]={'T','E','M','P','E','R','A','T','U','R','A','=','*','*','.','*','*','C',0x0D};//0X0A
//TABLA_BLUE[] Variable to send the temperature data to Blueterm Master module
long TEMP;//Temporary variable
int i;
int j;

void main( void )
{
  conf_WDT    ();               // Configure WDT of the system
  conf_CLK    ();               // Configure CLK of the system
  conf_IO     ();               // Configure Inputs/Outputs
  conf_UART   ();               // Configure UART
  conf_TimerA ();               // Configure Timer A

  __enable_interrupt();         // Interrupciones ON.

    for (;;) //it is like a while(1)
          {
```

```
if (datoUART == '5') {            // DATA TO THE PC

ADC10CTL1 = INCH_10 + ADC10DIV_3;   // Temp Sensor y f_ADC10 ~ 5MHz/3

  ADC10CTL0 = ADC10SHT_3 + SREF_1  + REFON + ADC10ON;    // 64/(5MHz/3)

 for(char degC = 30; degC > 0; degC-- );

ADC10CTL0 |= ENC + ADC10SC;          // Sampling and conversion start
while (ADC10CTL1 & ADC10BUSY);        // check for ADC conversion is completed

 ADCDATA = ADC10MEM;
// Read ADC value
// Temperature in degrees conversion by using formula ADCDATA =
(1024*(Vtemp/Vref))

  //ADCDATA= conf_ADC10(datoUART);

TEMPERATURAC = ((ADCDATA - 673) * 42300) / 1024; //CENTIGRADES

//Impression of the values
//ASSIGNMENT OF VALUES TO THE ARRANGEMENT
//ASSIGNATION OF TEMPERATURE IN CELSIUS
TEMP=TEMPERATURAC;//TEMPERATURAC STORED ON A TEMPORARY VARIABLE
TABLA_TEMP[3]= (TEMP%10)+0X30;
TEMP=TEMP/10;
TABLA_TEMP[2]= (TEMP%10)+0X30;
TEMP=TEMP/10;
TABLA_TEMP[1]= (TEMP%10)+0X30;
TEMP=TEMP/10;
TABLA_TEMP[0]= (TEMP%10)+0X30;
```

```
    TEMP=TEMP/10;


  for(i=0;i<3;i++)
        {
            while (!(IFG2&UCA0TXIFG));            // USCI_A0 TX buffer ready?
            UCA0TXBUF = TABLA_TEMP[i];                 // TX -> RXed character
        }
   __delay_cycles (100000);



  ADC10CTL1=0;
    ADC10CTL0=0;
    }




   if (datoUART == '6') {



     ADC10CTL1 = INCH_10 + ADC10DIV_3;   // Temp Sensor y f_ADC10 ~ 5MHz/3
     ADC10CTL0 = ADC10SHT_3 + SREF_1  + REFON + ADC10ON;     // 64/(5MHz/3)


     for(char degC = 30; degC > 0; degC-- );


     ADC10CTL0  |= ENC + ADC10SC;            // Sampling and conversion start
     while (ADC10CTL1 & ADC10BUSY);          // check for ADC conversion is completed


      ADCDATA = ADC10MEM;


      // Read ADC value
```

```
            // Temperature in degrees conversion by using formula ADCDATA =
(1024*(Vtemp/Vref))



          TEMPERATURAC = ((ADCDATA - 673) * 42300) / 1024; //CENTIGRADES


          //Impression of the values
          //ASSIGNMENT OF VALUES TO THE ARRANGEMENT
          //ASSIGNATION OF TEMPERATURE IN CELSIUS
          TEMP=TEMPERATURAC;//TEMPERATURAC STORED ON A TEMPORARY VARIABLE
          TABLA_BLUE[16]= (TEMP%10)+0X30;
          TEMP=TEMP/10;
          TABLA_BLUE[15]= (TEMP%10)+0X30;
          TEMP=TEMP/10;
          TABLA_BLUE[13]= (TEMP%10)+0X30;
          TEMP=TEMP/10;
          TABLA_BLUE[12]= (TEMP%10)+0X30;
          TEMP=TEMP/10;
          for(i=0;i<17;i++)
                {
                      while (!(IFG2&UCA0TXIFG));            // USCI_A0 TX buffer ready?
                      UCA0TXBUF = TABLA_BLUE[i];               // TX -> RXed character
                }
         __delay_cycles (100000);


           ADC10CTL1=0;
          ADC10CTL0=0;
          }
          }
 }
```

7. PROTOTYPE AND SOME PROBLEMS

In this section explains how is physically the prototype of the design as well as some problems that have been at the time of the realization of the thesis.

7.1 Prototype

In this section the prototype made during this thesis is shown (Figure 26). It can be seen several modules. The prototype made consists only of one slave module. The other slave module simply would have to implement it as before because they are identical.

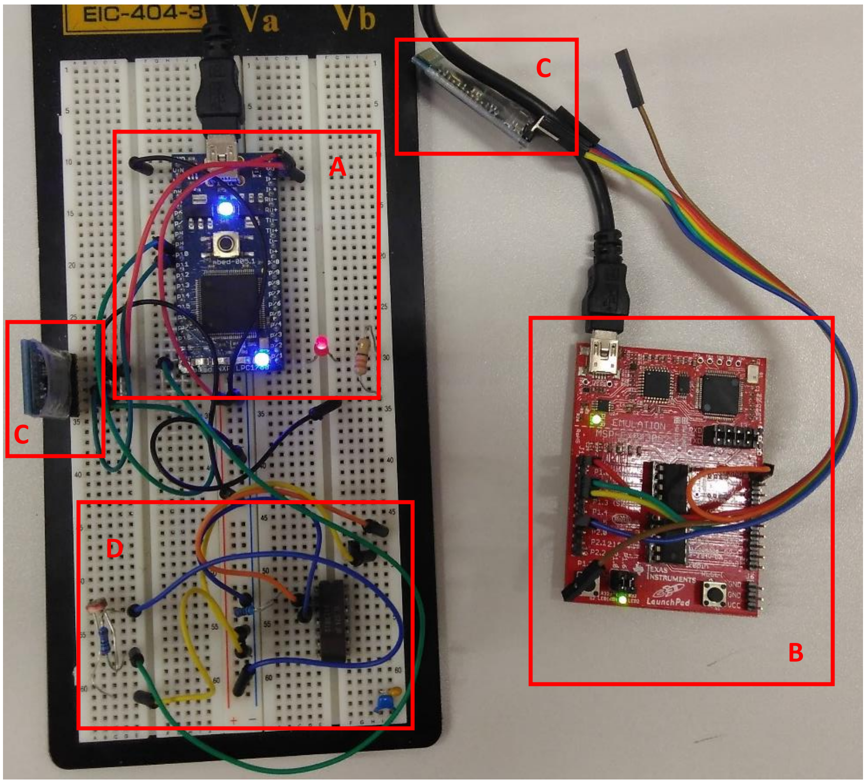

Figure 26: Prototype of the design (A) MBED (B) MSP430G (C) Bluetooth modules (D) Lighting sensors

In the following figures (27, 28, 29), it is shown how is the procedure for taking the temperature and lighting data. This is just some examples of the possiblilities.

In the Figure 27 it is shown how to read the lighting data expressing this one in percentaje of light. In that state as has explained before the user can modify the brightenss of the LED.

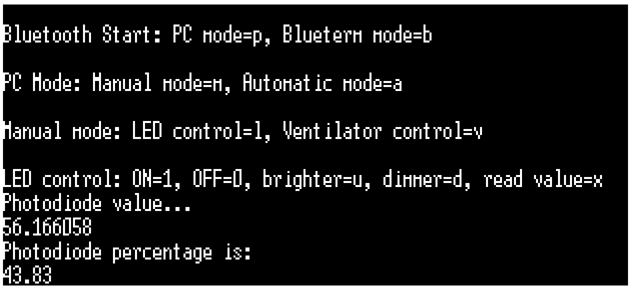

Figure 27: Reading lighting data

In the Figure 28 it is shown the procedure of reading the temperature data. In this state as has explained before the user can swith on or swith off the ventilator.

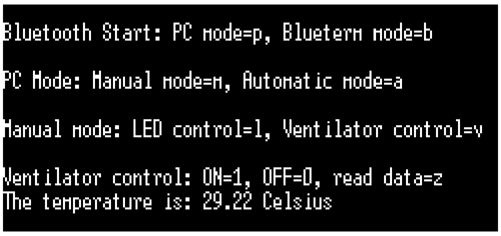

Figure 28: Reading temperature data

In the Figure 29 it is shown how is working the system automatically. First, it reads both data of interest, temperature and lighting and then depending on this datas modifying both LED and ventilator.

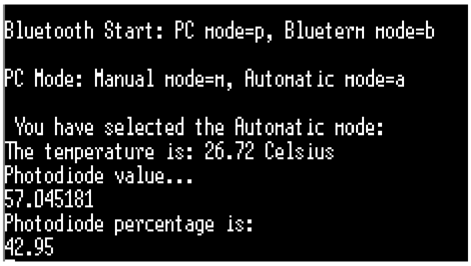

Figure 29: Reading lighting and temperature data

7.2 Problems during the design of the thesis

In this section the most significant problems that have been taken during the development of the thesis are discussed. There are two main problems.

The first is when the desired task is reading the data of the light from the board MSP430G2553. The problem is when the measure was done at the same time with the temperature. Due to not having an extensive knowledge of the IAR program the problems appear with the clocks that attend to certain interruptions. This did not happen if is reading independently. If the reading is performed separately in different programs there is no problem.

The solution that has been considered for this problem has been to control the reading light from the MBED module itself.

The second problem is in the Bluetooth communication. This problem is because the HC-05 modules can only be paired with another HC-05 module either slaves or masters. So when trying to control the outputs from the mobile application, this terminal (mobile) could not decouple other Bluetooth modules to pair with them.

The solution that has been found to this problem is simply to use another extra module for connection to the mobile terminal.

8. CONCLUSIONS

As explained during the documentation, the thesis has made consists on design, develop and implement a system that monitors temperature and brightness of a room of a building. Both temperature and luminosity can be controlled by means of a LED and a ventilator remotely.

During the project, the proper functioning of the system has been tested. Once designed each module, they have been simulating individually to check for proper operation. After the individual test will test the entire system as a whole. In addition to this simulation, it should test the implementation to ensure the proper operation in real time.

As explained above monitoring certain parameters is very important. Obviously depending on which parameter being monitored can become more useful or not. It is not the same monitoring some measures than others. Mainly because the objective are completely different. For example, the objective of monitoring temperature and luminosity is more to save money, taking care of the environment or improved quality of life. In other cases it can be monitoring vital signs using some chips in the human body that allow act simultaneously in emergency situations. That it is the reason of why Internet of Things is so important.

One of the pluses of this thesis is that it is a development that may add other modules to monitor and thus make it a much more complete.

9. INFORMATION SOURCES

[1] Internet: http://www.tecnoimprese.it/user/File/Eventi/RF08-14feb_melchioni.pdf

[2] Internet: https://es.wikipedia.org/wiki/Bluetooth

[3] Internet: A review of wireless communications for smart grid

[4] Internet: http://www.i-scoop.eu/internet-of-things/

[5] Internet: http://serconint.com/blog/sensores-para-domotica/

[6] Internet: http://www.domoprac.com/hardware-y-productos-domoticos/sensores-tipos-y-funcionalidades.html

[7] Internet: http://tecbolivia.com/index.php/venta-de-componentes-electronicos-11/tableros-de-experimentacion/kit-msp-exp430g2-launchpad-detail

[8] Internet: http://tecbolivia.com/index.php/articulos-y-tutoriales-microcontroladores/32-plataforma-mbed-de-prototipado-rapido-con-microcontroladores-de-32-bits

[9] Internet: http://saber.patagoniatec.com/hc-05-bluetooth-conectar-esclavo-hc05-maestro-master-save-wireless-tutorial-iot-celular-smartphone-arduino-argentina-ptec/

[10] Internet: http://www.prometec.net/bt-hc06/

[11] Internet: http://zygzax.com/enviar-y-recibir-datos-con-modulo-bluetooth-para-arduino/

[12] Internet: http://forum.arduino.cc/index.php?topic=283740.0

[13] Internet: http://energia.nu/Guide_MSP430LaunchPad.html

[14] Internet: https://giltesa.com/2012/12/21/conectar-dos-arduino-mediante-bluetooth

[15] Internet: http://hetpro-store.com/TUTORIALES/bluetooth_hc-06_app_arduino/

[16] Internet: https://alselectro.wordpress.com/2014/10/21/bluetooth-hc05-how-to-pair-two-modules/

[17] Internet: http://unbarquero.blogspot.hr/2010/09/msp430.html

[18] Internet: http://unbarquero.blogspot.hr/2013/01/msp430-ldr-led.html

[19] Internet: https://e2e.ti.com/support/microcontrollers/msp430/f/166/t/412698

[20] Internet: https://prezi.com/j4ovmpc_fahs/microcontrolador-msp430g2553/

[21] Internet: http://microcontroladormsp430.blogspot.hr/2013/06/puertos-de-entradasalida.html

[22] Internet: https://es.wikipedia.org/wiki/ZigBee

[23] Internet: http://www.zigbee.org/

[24] Internet: https://en.wikipedia.org/wiki/6LoWPAN

[25] Internet: http://www.ti.com/lit/ds/slas735j/slas735j.pdf

[26] Internet: http://pdf1.alldatasheet.es/datasheet-pdf/view/93636/ETC/IL324.html

[27] Internet: http://www.i-scoop.eu/internet-of-things/