American University of Beirut
Faculty of Engineering & Architecture
Electrical and Computer Engineering Department

EECE 501 Final Year Project (FYP)

Group Final Report

Fall 2011-2012

AAHA: Arduino/Android Home Automation

Group & Disciplines

Kim Baraka	*Circuits and Electronics*
Marc Ghobril	*Computer Architecture*
Sami Malek	*Software*

Advisor: *Prof. Rouwaida Kanj* (ECE Department)

Co-advisor: *Prof. Ayman Kayssi* (ECE Department)

Sponsor: *Intel Corp.*

1 Executive Summary

With the advent of capable and relatively cheap Smartphones, microcontrollers and communication techniques driven by technological advances, building comfortable, secure and energy-efficient homes and offices has become a trend. The objective of our project is to make use of Home Automation techniques to design and implement a remotely controlled (through mobile device) and energy-efficient Smart Home that safeguards the residents' comfort and security, if not maximize them as well.

Many companies have security, energy and comfort (heating/cooling lighting) solutions packages. These packages are expensive and are not very flexible. We investigate the techniques and standards used by these companies and come up with a practical design that would optimally satisfy the objectives of our project. However the available systems can suffer from lack of flexibility, scalability or practicality (for those who are not “plug and play”) and most of the technologies used (in terms of networking protocols especially) are proper to each company, and are often easily accessible. Most importantly, the major issue remains the high cost of such systems. This has led many Do-It-Yourself (DIY) enthusiasts to consider building alternative options based on easy to use microcontrollers, like Arduino, and other cheap devices. Our challenge will be to bring the DIY solutions to the next level and make it competitive with the systems available on the market.

We will use an Android tablet and an Arduino microcontroller interfaced with home appliance controllers to be able to control lighting, cooling/heating, door locking and other home-related actions. Other options include timers, suggesting a rescheduling of tasks to efficiently manage energy resources and creating customized scenarios. Our current prototype is an Arduino with Internet connectivity that receives HTTP requests from an Android app with user-friendly GUI. The Arduino turns LEDs on and off on a breadboard based on the user's request. A temperature sensor is also interfaced with the Arduino to simulate A/C (Air Conditioning) control.

2 Acknowledgments

We thank our Advisor Prof. Kanj and co-Advisor Prof. Kayssi for helping us and giving us precious advice for the good functioning of our work.

We would also like to thank Intel corporation for providing us with the Android tablet that constitutes an important part of our prototype.

Table of Contents

List of Figures

List of Tables

3 Introduction

3.1 Motivation

In a world which is becoming increasingly automated, Home Automation is acquiring more attention from the households around the world. Automating our daily activities in the household saves us time, effort and worry. We could be notified of important daily tasks that have not been done, or get warnings when something goes wrong (no need to worry about lights turned on for a whole week when on vacation, or leaving the house inundated when an unpredictable storm hits...). If many industrial facilities (and even offices) have moved to almost fully automated systems, there are only a few houses (usually the most fancy and ridiculously expensive ones) that are automated. Home Automation should be more accessible since it does not require an incredibly advanced technology and can usually be implemented with off-the-shelf devices.

As devices get smarter, a lot of effort is aimed at automating many of our day-to-day activities in a relatively easy way. A new Home Automation field is emerging with the incorporation of the mobile communications technologies into the automation systems, to control appliances either via the existing electrical wiring of the house, or using a wireless network. Indeed, users are using mobile applications on their phones to control their houses from distance. Trivial examples are turning the A/C on 30 minutes before your arrival, or opening the door lock using your mobile phone. Furthermore, remotely accessing the house network (consisting of sensors, controllers and actuators) is achieved through connecting this house network to the Internet, a feature which can be fairly placed under the framework of current endeavors for an Internet of things (IoT). You could for instance make the system send a daily email to your inbox with all sensor information in your house (temperature, pressure, humidity, energy usage.

luminosity ...), make a tweet every time the bell rings, or post on your Facebook wall every time you forget to feed the cat.

The Figure below illustrates one possible Wireless Home Automation network.

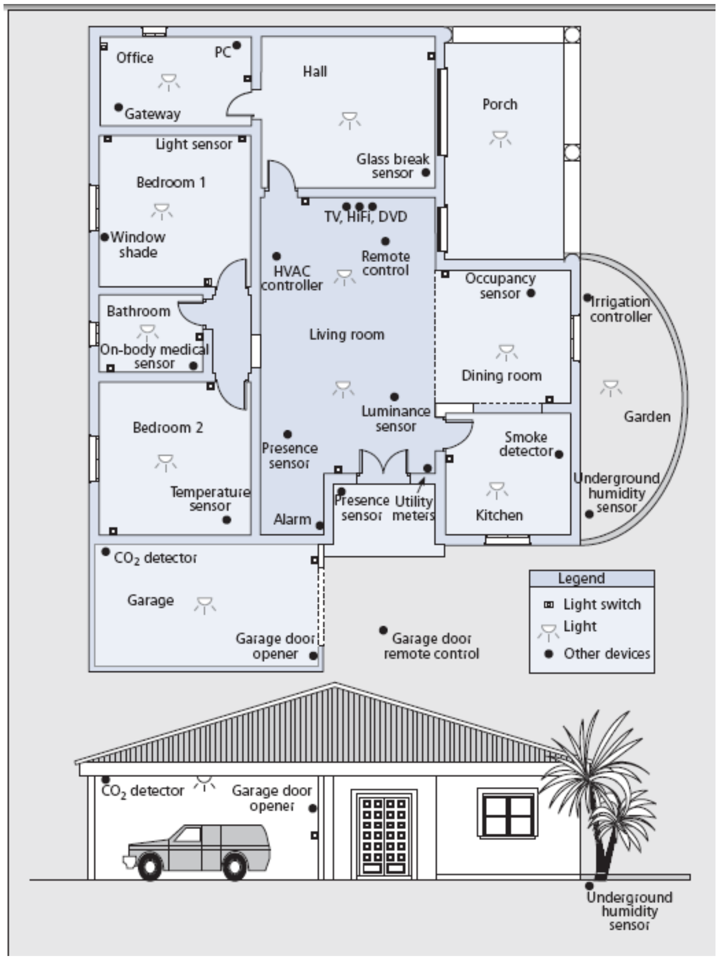

Figure 1: An example of a Wireless Home Automation network, source [1]

The integration of the mobile technology (and in particular Smartphones) is essential for our system. Based on the statistics stated in [2]:

- Today, 80% of the world population has a mobile phone
- 1.08 Billion of the world mobile phones are Smartphones

- 89% of Smartphone users use their phones throughout the day

To demonstrate the necessity of mobile Home Automation system, we differentiate 2 cases [3]:

a) **“Enforced undesired mobility”**: this is when an individual has to get up and move in order to satisfy a particular need, for example moving to turn on/off the lights.

b) **“Inhibited desired mobility”**: this is when an individual has to remain in his place, and not moving, because a certain activity requires his presence, for example watering the plants.

Using our system, these two issues can be resolved. But as mentioned above, in addition to comfort, the automation of the home has many other advantages ranging from security to maintenance, but also easy scheduling, time saving and most of all contribute to a more harmonious and coherent environment to live in.

In addition, exploiting such technologies may contribute to a greener planet; and if by a "Smart" Home we mean one that is energy-efficient and that will be able to appropriately manage its energy resources, then this feature is essential for a safer and healthier future. Here are some facts concerning the lighting control systems [4]:

- When we dim the lighting by 25% we are saving 20% of energy. Likewise, when we dim the lighting by 50% we are saving 40% of energy.
- The lifespan of luminescent bulbs is 20 times longer for dimmed bulbs at 50%.

According to the US energy Information Administration [29], the residential energy consumption constitutes about 14% of the total energy consumption in the world, as shown in Figure 2. This consumption is distributed on heating, lighting, and appliances usage.

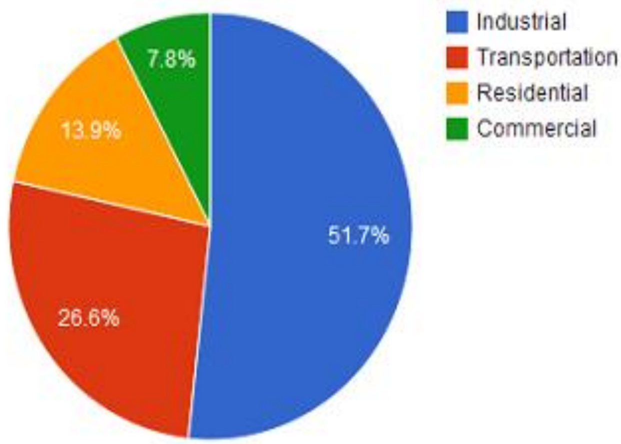

Figure 2: World Energy Consumption by sector 2012 (EIA data), source [29]

In the United States the residential consumption is growing steadily and approximately doubled in 50 years (from 10 quadrillion Btu in 1960 to 20 quadrillion Btu in 2010) [29] as revealed in Figure 3.

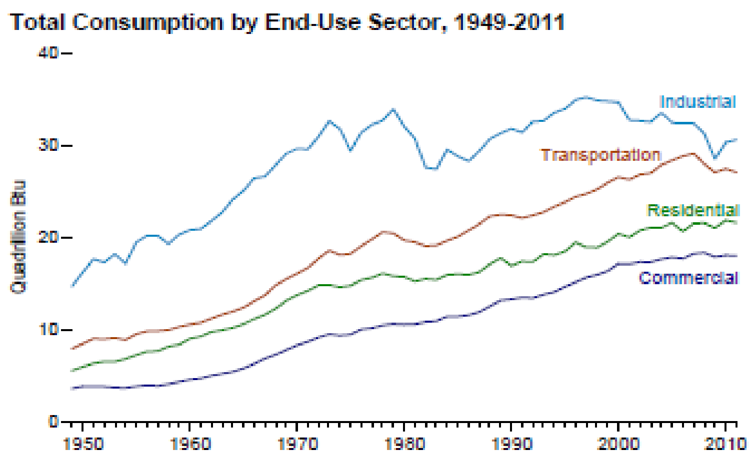

Figure 3: Consumption by End-User Sector in US, 1949-2011, source [29]

The total residential consumption is distributed as shown in the Figure 4 below. Space heating (32%) and cooling (13%) as well as water heating (13%) and lighting (12%) are the main residential end uses. Together they account for 70% of the home consumption [30].

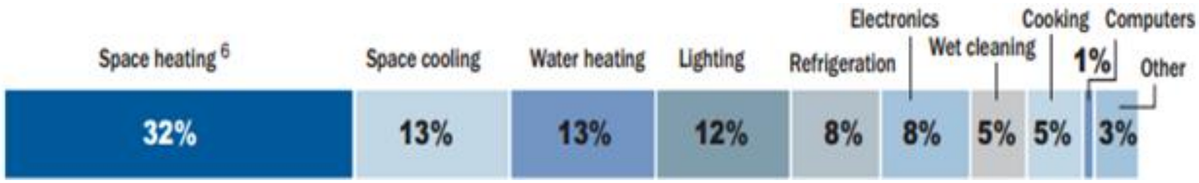

Figure 4: Residential energy end usage in the US, 2007 (EIA data), source [30]

That increasing consumption in the residential sector can have a significant impact on the environment and ultimately contribute to the global warming problem.

The solution to problems like global warming and other related ecological problems is that every citizen be aware of the environmental threats that are becoming alarming. Saving the planet starts nowhere but in our cozy and modest homes.

3.2 Desired Needs

There are many problems that are tackled when using automation in homes. The need for power management for efficiency purposes was the main motive behind considering Home Automation. We believe that the concept can contribute in minimizing wasted energy, whether by controlling the heating/cooling and lighting systems, or other appliances that are greedy in power. As we moved forward in our objective, we discovered that other needs also as important can be satisfied as a byproduct of equipping a house/office with a smart system backbone: these are Security and Comfort. What more can one ask in terms of security when one's home is monitored 24/7 by a robot (no room for treason) and is notified in cases of emergency through his Smartphone, from which he can remotely take a peak of what's going on in his house or checking on his children... ? Moreover, the Smart Home concept can allow the resident to customize his home according to his preferences, whether by scheduling his Stereo system to play his favorite song as his wake up call or creating ambiances, or automatically closing the shutters at night.

As mentioned above, in the old days, the concept was relevant to either luxury houses or to industrial manufacturing. Nowadays, every home/office should benefit from at least its basic features. Going back to the energy management element of the system, we could say that it would help meet power or energy targets. For instance, in houses worldwide where renewable energies are being exploited and in which local batteries are used to power up the house (or part of it), there is naturally an energy target per day (think about personal solar PV cells for example), but also a power target (the load power should not exceed the capacity of the batteries). According to [21], energy generation and consumption in such homes exhibits predictability, a characteristic which can be made use of to meet such constraints. Similarly, referring to the Lebanese electricity situation, most houses are equipped with an auxiliary generator to be used when the mains electricity is down, and these generators have power limits that have to be dealt with. Moreover, on a more global level, the smart scheduling and monitoring of energy usage in homes could benefit the power grid as a whole and avoid breakdowns like the one that happened in huge areas of the Midwest and Northeast United States and Ontario, Canada on August 14, 2003 [34].

The objective of our project is to make use of Home Automation techniques to design and implement a remotely controlled and energy-efficient Smart Home with basic features that safeguard the residents' comfort and security. We will be using in our system a low-cost Android tablet as the controller, and an Arduino microcontroller to offer connectivity with the electronic devices of the house network. The mobile device is the component on which we will develop a user-friendly application, the portal to the house network (either remotely or locally through direct wireless connection to local base).

In many available systems, the house network is either completely wireless or completely wired. There are exceptions (like Insteon that use a hybrid system), but these are often more expensive than other comparable systems, and use closed standards. To the best of our knowledge, our system will be the first complete solution to make use of both wireless Zigbee and wired X10 technologies. Zigbee ensures the reliability of the system and X10 allows the integration of low cost controllers to optimize the overall cost of the system (X10 actuators are the cheapest, although not the most reliable).

In addition, granting the user the ability to choose between two sources to power up each device or each set of devices is a novelty in Home Automation. For example, an option to choose between a renewable energy source from local solar or wind energy and government electricity may be integrated in our system.

Ultimately, the controller software will be able to do smart decisions based on sensors readings, user preferences and external inputs, with source and load-side power management.

4 Requirements and Deliverables

4.1 Requirements, Specifications and Deliverables

The basic requirements and specifications assigned to us at the beginning of the semester are the following:

- Building a Home Automation system using a low-cost Android Smartphone that can send commands to an Arduino microcontroller which controls appliances.
- Possible enabling of applications such as Home Automation power control system and home digital timer controller.
- Creativity in the nature of the system features and/or in their comprehensiveness and/or in the fact that it competes with systems with much higher cost.

The following is a list of deliverables that have to be ready upon completion of this two-semester-long project:

- List of hardware needed when our final design is ready to be implemented.
- Android App with user-friendly GUI that can run on any Android device..
- Program to be downloaded on Arduino board.
- Communication interface between Arduino and Android device that is reliable and secure.
- A power control system that tracks energy usage in the home and smartly manages it.

- Detailed block diagram of the scalable system as a whole.
- Actual prototype of the system with a limited number of appliance controllers and sensors.
- Any additional features that can be thought of and realistically implemented.

5 Technical and non-Technical Constraints

5.1 Technical Constraints

- GUI should be unsophisticated and accessible to ordinary users
- Network should be able to route measurements (current, temperature…) with a decent bit rate
- High reliability: Reliable connectivity to all network components, especially for security applications
- Robustness: The Network should work even with electric power outages (convenience, security and power management purposes) and hence be power efficient to run on standby batteries.
- System should decrease overall household/office power consumption.
- System should increase the user's control and convenience and maintain, if not also improve security.
- Design should be easily scalable
- Use of Arudino as Network microcontroller and Android as software platform

5.2 Non-Technical Constraints

- **Cost** being a crucial component in our project, the estimated cost should not exceed 30% of the existing marketed solutions price

- **Time**: Implementation should be done by the end of the Spring Semester.
- The project should be **well documented** to avoid design errors and unnecessary detours.
- **Resource constraints**: Even if the on-paper design makes use of components that were found to best fit our project, our actual implementation would have to include devices that are currently available on the market and easy to get within our time margins.

6 Literature Review

6.1 General Home Automation Products

Numerous Home Automation products exist on the market. Most of them are functional only for products within the brand that sells the system as a whole, but some are compatible with other existing "standard" technologies like X10, KNX, Insteon, Z-wave, etc.

We will not present specific products in their technical details, because the aim of our project is to create a low-cost alternative to such expensive systems. We will instead extract, in a first step, the main features of a Home Automation system. These are usually common to all products available on the market, with some slight variations. These features will be presented next, but before doing so, we would like to emphasize on the fact that the main difference between these products is the interface between user and central controller. Some systems require a software running on PC, other are Web-compatible, making it easy to control your home from any mobile or non-mobile device with Internet connectivity. Most of the systems available on the market have a central controller, which is exactly the role that our Arduino board will play, and that communicates with all appliance controllers within the house. To cite a few of the systems on the market, Belkin's Wemo allows the control of almost anything through an App. It also can make use of sensors to activate custom events in the house, or use the Web to notify of events happening in the house using the IFTTT (If This Then That) service (for instance, sending an email every time the cat visits the litter box) [5]. The magDomus system is another solution for Home Automation; this one is compatible with most existing technologies, which is very practical. The most popular computer software for general Home Automation applications (an open-source software) is MisterHouse, which is customizable, very flexible and compatible with most technologies. [6]. However, it runs only on Windows and Linux and is not compatible with Arduino. Another full system is the Home Easy system which controls Lighting, Appliances, TV/DVD/Satellite, Heating and Door Access from a base station, a remote control or through SMS from a mobile phone [7].

We will investigate paper projects or DIY projects starting next section. But before looking at specific projects, let us extract the general features of any Home Automation system available on the market.

- Lighting and modules: Dimming the lights or switching ON/OFF any appliance with a controllable socket attached to it.
- Climate: Temperature control of central heating/cooling system.
- Security and sensors: Door locks and blinds, security cameras, motion sensors management.
- Irrigation: Scheduled irrigation of plants (possibly based on humidity levels).

- Energy measurement: Tracking of power consumption on different devices.

This categorization can be found in most of the Apps available on the Android market, like SmartHome iDom, Universal Home, Nexia etc.

6.2 Low-cost Projects for Home Automation applications

Almost every project that we found used a form of central intelligence that would get commands from the user (or devise its own commands) and send them through the house for execution (Figure 5). This is expected, since it is hard to conceive of a mesh network of appliance controllers, when each controller often has a very simple task like a simple switching. It is true that the Internet of things is an emerging trend, but in the case of Home Automation, the need for a central controller has apparently remained till now the best option

.

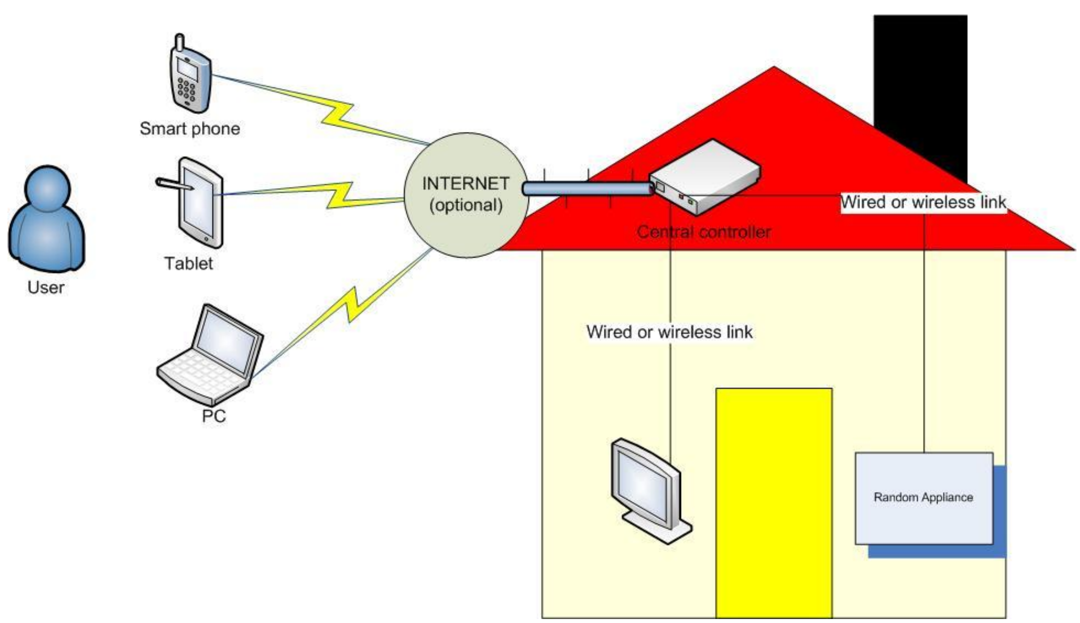

Figure 5: Basic Home Automation components in most projects found

6.2.1 Different forms of the central controller (or brain of the system)

Some Home Automation projects have been designed for sending commands within the home only. For this purpose, commands may be received from a short range wireless technology like Bluetooth. This is the case in [8], where the commands are sent from a Smartphone through Bluetooth to an Arduino Mega board as a central controller. Another option is to connect the microcontroller to the Internet to be able to communicate with it from any device that has Internet-connectivity. This can be achieved by the presence of a Web server. The Web Server can run directly on the microcontroller, or on a separate PC with connectivity to the microcontroller, as depicted in [9] or on a router running OpenWrt, like in [10]. Another option is to use cloud services like Cosm (previously called Pachube), whose services (storage of sensor data on cloud servers) can relieve the microcontroller's memory and computational power. This is done in a project from University of Tennessee Martin, from which we only had access to the poster [11]. The Cosm server converts messages sent from a Smartphone to messages that the Arduino board understands. In our project, we may use Cosm to use the real-time information collected by the house sensors for instance, but Cosm services in themselves only give access to raw data and do not offer computing power for processing or running decision algorithms. This is why other more general cloud services may be used, if not for extra computing power, at least for data backup. The idea is to take advantage of the Internet connectivity of the microcontroller to use the cloud.

6.2.2 Communication protocols between Home Automation devices (the brawn of the system)

When designing a Smart Home Automation environment, one of the essential concerns is to decide on the communication protocol (or standard) that will link the “brain” to the different devices of the system. These protocols are available in an extensive variety (X10, Zigbee, Insteon, KNX...) with different features and characteristics [12].

- Direct connection

The most basic approach is to use a direct connection of the devices to the microcontroller, like in [9]. Such a procedure is very undesirable in our case, since the Arduino board represents our central unit and not a gateway between central unit and devices like in this project. This would offer very low scalability (limited to the pinout of the Arduino) and would need extra infrastructure or rewiring.

Another project with direct connectivity is [15]. It is true that it presents a very low cost alternative as long as a limited number of devices is being controlled, but it doesn't fit our project in terms of flexibility, scalability and easiness of use.

- Wired protocols

X10 is one of the most popular protocols in the Home Automation business, with millions of homes using it around the world, mainly due to the cheap accessibility of its components. Furthermore, most of its units are really easy to set up (it is “plug and play”) [13]. In [11], X10 devices are controlled from an Arduino board, making use of the built-in X10 Arduino libraries to send X10 messages through output pins.

The X10 protocol may be included in our project due to its easiness of use and compatibility with Arduino. However, in [14] the restrictions of X10 are presented. First, collisions may appear if one is using several controllers or repeaters, and X10 doesn’t possess the capacity to detect such collisions. This may sometimes lead to altering the status of several devices for no reason. Second, the protocol is considered to be relatively slow (sending a command takes about ¾ of a second). Third, the X10 signal may experience attenuation when there is electrical noise, which occurs very frequently in a country like Lebanon. As a result, one of the devices can be accidentally switched on. As a conclusion, if X10 technology is used in our project, it would have to be for applications that are error- tolerant like lighting for instance, but not for applications like security.

Note that there are other wired protocols that have been used in general Home Automation projects like KNX, Insteon or UPB (Universal Powerline Bus). In Lebanon, the large majority of Home Automation systems available use KNX as a wired protocol to communicate between central unit and appliance controllers, as we have seen by touring the main Home Automation companies around Beirut. These wired technologies are not as compatible with Arduino as X10 is, so if we choose a wired protocol for part of the appliance controllers, our choice would go for X10, especially that [16] provides a way to overcome X10's problems by adding extra units like repeaters.

- Wireless protocols

Since most RF devices that work on a remote control operate at a frequency of 433 MHz (European Standard), some DIYers have considered the option to reverse-engineer the ASK modulation scheme used to create Arduino libraries that emulate the functioning of the remote control. A system with this capability can mostly control sockets, plus it is not very flexible and its scalability is questionable. Some have even considered using the actual remote control connected to the Arduino board to send RF signals in an easy way. Although very practical, there is a serious scalability issue since most remote controls support up to 8 devices.

Other wireless options that are "cleaner" (no hacks involved) include the wireless standard that has succeeded to the old X10 wired technology, which is called Z-Wave. Since it is relatively new, no DIY projects have been found using this technology and it is nowadays mostly integrated in market solutions or products development. Z-Wave is not a completely license-free protocol and may be quite expensive compared to other wireless options.

Another option for a wireless communication within the house network is the well-known Zigbee protocol, like in [14]. Their system uses a mesh-network topology for greater robustness, by making use of the capability of the nodes to self-organize themselves into a mesh network. As a wireless protocol, we will see that Zigbee provides one of the most efficient ways to integrate a large number of wireless nodes in a Home Automation setting. This is why it may have an important part in our project.

A final note on wireless communication between devices and central controller: some Home Automation projects incorporate control of the TV/DVD etc. or any other devices that work through an Infrared (IR) remote control. This is why [8] has used an IR transmitter for these purposes, but the system only works for short distance communication. Including infrared modules in our system is a desirable feature since it would not require any rewiring and would increase the compatibility of our system with existing appliances.

6.2.3 Sensors

Sensors are an important part of our design. They constitute a way to gather information on the environment to possibly trigger automated events at the Arduino board, or provide information to the end user. These sensors are mostly for temperature, humidity, luminosity, pressure, motion etc. and can be either digital or analog. In most DIY projects we found, sensors were wired to the Arduino board. This is for instance the case in [17] and [18]. However, a low-cost wireless alternative was implemented successfully in [19], using sensors that communicate through the Zigbee protocol. Sensors are mounted on an Xbee chip running the Zigbee protocol stack and which can communicate with the Arduino that has an Xbee shield on it. The Xbee chip is compatible with both analog and digital sensors. Such an approach is very advantageous for our project and could be implemented for sensors, since we would like to avoid as much as possible any wired connection to the Arduino board, both for reliability (no risk of overcurrents), practicality (sensors and microcontroller board are not physically tied together), and scalability (pinout of the Arduino would cease to be limiting).

6.2.4 Energy management

People nowadays are aware of the necessity to manage their energy consumption in a more reasonable way but efforts to succeed in doing so remain not as easy as we think they are. Sometimes, we may have the desire to live "ecologically" but don't really know how. Home Automation can solve this issue and make a significant impact concerning an improved energy organization [20] [21]. There exist several aspects of our houses that we can improve using Home Automation technologies [21]. If we consider the house's lighting, for example, dimming the lights or making their intensity adjustable according to the activity carried out, can yield a considerable reduction in the amount of electricity payment. The same applies for the heating and cooling systems. Home Automation can also help monitoring the water consumption, especially when it comes to the watering of the plants.

The system in [20] is constituted of two essential kinds of devices: sensors and actuators. The sensor measures a physical quantity (temperature), converting it into a readable signal, while the actuator changes or monitors the measured quantity (changing the temperature). When the status of a device in the network is changing, the information is sent to a rule engine which will in turn generates an action to be executed. We will use a similar approach in our project in collecting data from the individual appliances. We may, in a fashion similar to [20], implement a learning algorithm that will get knowledge about the frequency of use of each appliance, to provide the user with suggestions to improve his daily task schedules to optimize average and/or instantaneous power usage.

In [21], the system makes use of an optimization framework that allows the creation of an equilibrium between the amount of energy collected and the energy spent, in a way that would satisfy the user's needs and meet his constraints within some limit. The system can also relate the weather predictions with the measures read on the devices, and use an algorithm which can put forward some recommendations for the user on a Smartphone application.

It is to be noted that for energy monitoring application, the authors of [22] believe that Zigbee is the most suitable technology compared to other wireless technologies, in terms of power usage, robustness, reliability and cost.

7 Applicable Standards

The standards and protocols mentioned in the report are:

- X10 protocol
- IEEE 802.15.4 protocol and Zigbee standard
- IEEE 802.11 protocol and Wifi standard
- Insteon
- Z-wave
- UPB
- Bluetooth

All these protocols and standard will be described in details and compared in section 9.

8 Proposed Solution Methodology

This section presents the process we followed to get to the adopted alternative and to our prototype. For a detailed schedule and plan for the next semester please refer to section 12.

I. <u>Literature Review</u>

- Research the general Home Automation products available:

  →Extract the high level features of our Android GUI

- Study low cost Projects for Home Automation applications

  → Check what other people have already done and build on their results

- Compare the available communication protocols between Home Automation devices

  → First, we researched the wired protocols: X10 seemed an appropriate candidate.

  →However, X10 turned out to have some issues and couldn't be used on its own.

  →We explored wireless communication protocols like Z-wave and Zigbee.

- Research the energy management field in Home Automation

  →Examine the available products in the market

  →However, these products only give access to raw data so we looked in research papers for scheduling algorithms

II. <u>Design</u>

- Settle on the communication protocol

  →Zigbee is chosen for its practicality, reliability and scalability

  →System will be hybrid: X10 will be integrated for specific appliances

- Decide on the hardware based on data rates, range, power and other relevant specifications

  →Arduino Mega as a central controller with Xbee and Ethernet shields has been chosen

- Choose the Xbee's network topology

  →Tree topology is better than star topology for range issues and better than mesh topology for traffic reduction

  →Some Xbees will be configured as routing nodes to provide scalability

- Data management plan

  →Data from sensors and user customization may be pushed on a cloud server

- Software conception

  → Android GUI will be user friendly and will easily communicate with the Arduino board through HTTP messages

  →Arduino software will be coded in a simple fashion and then will be expanded to become dynamic and self-sufficient

III. Implementation & Testing

- Build the actual prototype of the system with a limited number of appliance controllers and sensors

- Test the prototype, determine the changes needed and improve on it

  → Test the assembled network by sending dummy requests

  → Test successful control over house network using local IP address

  → Test the capacity to convert HTTP request to Xbee/X10 commands

- Implementation of additional features if time constraints permit

9 Design

9.1 Design Alternatives and Decisions

9.1.1 Alternatives and Decision for Overall Design

A central controller, which would receive user commands to execute, is mandatory. The device has to also have the capability of taking its own decisions based on learning or at least recommend actions to the user based on previous history. This central unit may or may not have Internet connectivity. On the user side, a mobile device provides an interface with the system as a whole through a user-friendly application. The mobile device can be either wired to the central controller (through USB cable for instance), or communicates with it wirelessly. Within the scope of the home, wireless connectivity can be achieved using a Wifi antenna on the central controller. By installing such an antenna, there is no extra cost in connecting to the Internet as well. This way, we would be able to access the controller either locally or remotely through the Internet. This is why Internet accessibility is desirable in our system.

Any Internet connected device acts either as a peer, as a client or as a server. In our case, the client-server architecture is the one to opt for, since the central controller is a fixed entity that responds to clients' (mobile devices) requests (and eventually sends them notifications as well). Hence the need for a server (at the application level, i.e. a piece of code that is able to respond to client requests) closely tied to the central controller.

The end-nodes within the home can be categorized into two separate types: they are either sensors that send environmental information (temperature, luminosity, motion, humidity, ...) or actuators that perform specific tasks (switching, dimming, ...). These nodes constitute a network that can be wired, wireless, or hybrid. Topologies can vary and have to be wisely chosen to optimize traffic load and reliability.

Figure 6 shows all the necessary components in our system. Each project component or group of components can be designed and implemented in many different ways. Based on our research and the above literature review on different technologies and protocols used for Smart Home Automation, we present in next section our design alternatives.

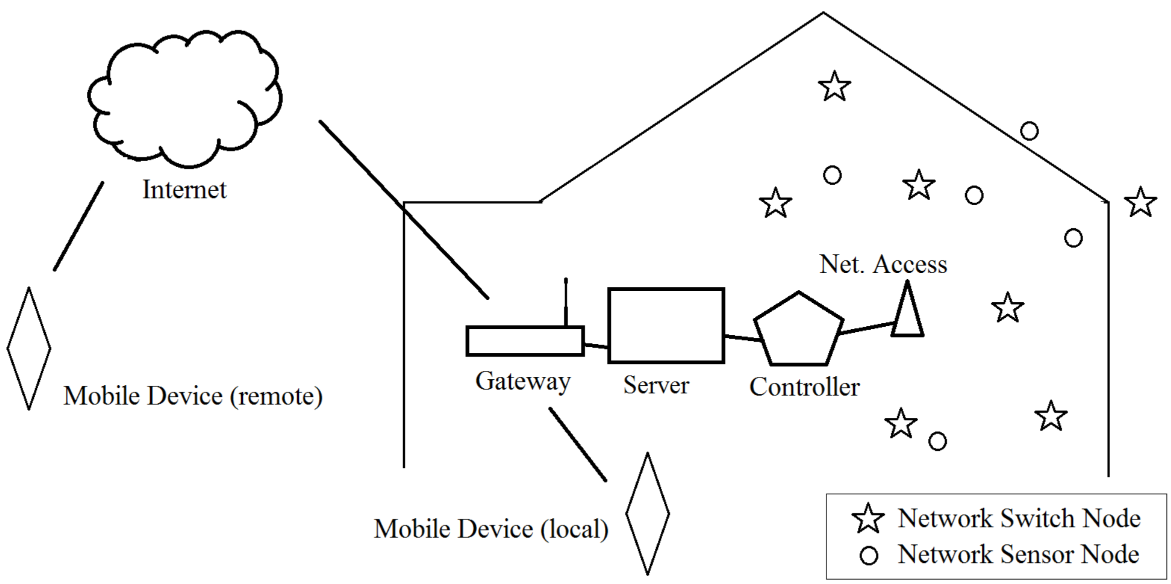

Figure 6: High level schematic

Figure 7 below shows a functional block diagram of our system. It embodies the main operations that will be performed by the distributed software on the mobile device/central controller/house network nodes.

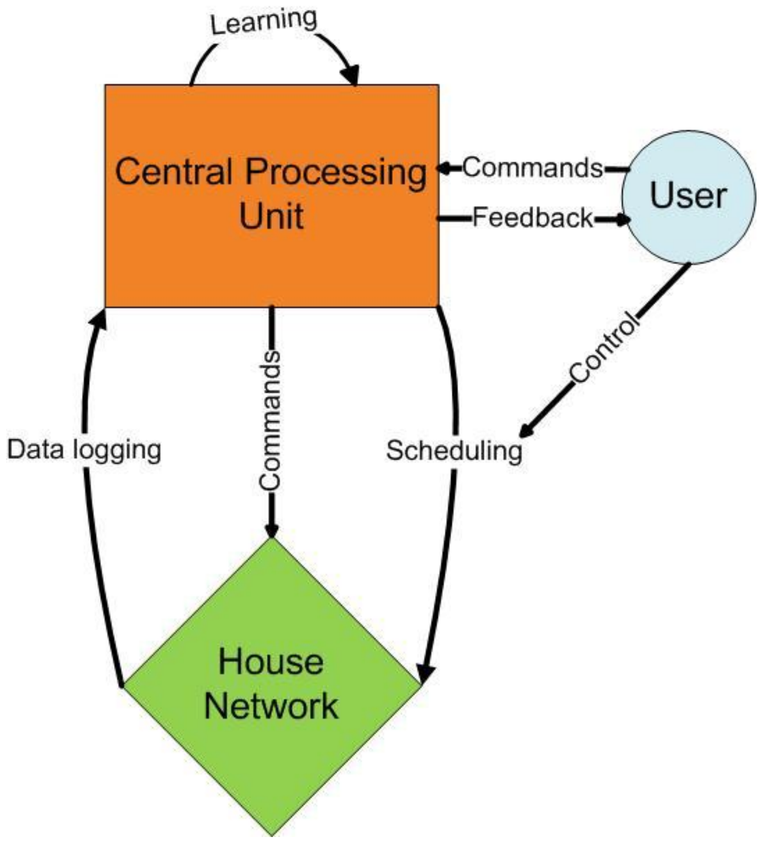

Figure 7: Functional diagram of the system

Next, we will discuss different alternatives for each mandatory component of our system, before summarizing our final design in a comprehensive diagram.

9.1.2 Alternatives and Decision for Mobile Device

The mobile device is a project constraint. It should be an Android Smartphone or Tablet. We will mainly use the Medfield-based Red Ridge Intel tablet running Android OS as a reference. This is an advantage since the Android OS is an open-source software and programming our GUI on it using free developer tools will be a relatively easy task.

9.1.3 *Alternatives and Decision for Internet Gateway*

The Gateway consists of any access point to the Internet. We can use a Wifi router or an Ethernet cable connecting the central controller to the router access point. We opted for Ethernet connection, since the central unit is a static device (no mobility) and Ethernet connectivity is simpler and lower in cost (Ethernet shield for Arduino is cheaper than Wireless shield).

9.1.4 *Alternatives and Decision for Server*

The Server is the software component responsible for receiving user commands (passing through the gateway) for execution. It has also the task of getting sensor information from the controller (originating from sensor nodes) and either storing them or forwarding them through the gateway to the Mobile Device. When we think of server, our mind directly envisions big bulky power hungry machines. This is not our case. Our server can actually be supported on the Arduino chip using the Ethernet Shield (costs around $45) or even on an OpenWRT small router. In case we needed more space, we can always upgrade the devices' memory or use a cloud server (especially if we're going to use security cameras). Our current decision goes in favor of a simple Web server application running on the Arduino, and that communicates through the HTTP protocol with our Web-based Android application.

9.1.5 *Alternatives and Decision for Central Controller*

The controller is a project constraint. It should consist of an Arduino microcontroller (around $60). This is the main controller that can directly talk to the Server/Gateway. Still, we can use other small limited functionality microcontrollers (PIC microchip) on some network nodes in case processing was required. Since we chose the Server to be embedded in Arduino, we can use the Arduino Ethernet module (50 $) pictured below (Figure 8) to be mounted on the Arduino board.

Figure 8: Arduino ethernet module

Arduino boards come in different flavors, with differences in processing speed, memory capabilities and pinout. The most basic and general-purpose Arduino is the UNO, but our code will most probably not fit in it. Number of pins and mobility or size considerations not being an issue for our project, we opted for the Arduino Due, the most powerful Arduino in terms of processing power and memory capabilities (could fit a complex code). Unfortunately, the board is out of stock on all online stores, due to limited production, and may not be available in time for our project. Therefore, we selected the Arduino Mega (Figure 9), which is closest to the Due in performance.

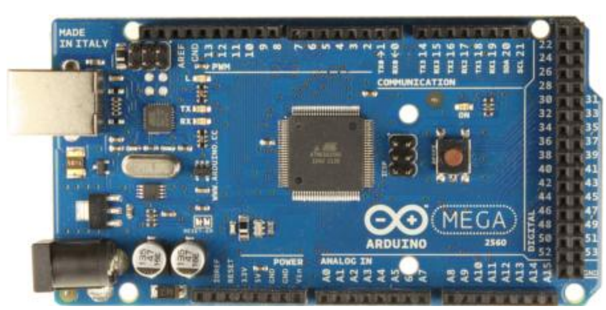

Figure 9: Arduino Mega

9.1.6 Alternatives and Decision for House network technology

- <u>Alternatives:</u>

The network to monitor the house environment (actuators that control lights, appliances etc. and sensors, both with local network connectivity) can be implemented in many different ways. Each technology has its advantages and inconveniences which are largely application-dependent. We will expose the majority of the technologies and standards currently used for Home Automation and choose the one that optimally achieves our objectives.

i. X10

X10 is the precursor standard for all Home Automation techniques. It started in the 1970s. The method consists in sending information (bit pulses) on the powerline AC wave at each zero-crossing [31]. Figure 10 below illustrates the concept. It has a low bit rate of around 60 bps or 1 command/s and is not suitable for a large number of sensor measurements. X10 is intrinsically not reliable, subject to external and self-interference noise (filters can solve the problem). It also lacks security. The 220V modules require tweaking the 110V modules which might be a dangerous task, or buying the more expensive 220V ones. Also, devices respond with a high latency of 0.75s. On the other hand, X10 is largely used with Arduino (an X10 library is available). With all its downsides, X10 is the most widely used protocol for Home Automation due to its relatively cheap cost and extremely simple home installation.

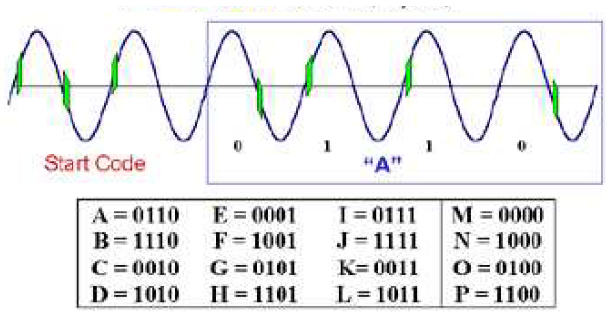

Figure 10: X10 protocol, source [32]

ii. Universal Powerline Bus (UPB)

The Universal Powerline Bus™ (UPB) communication method was developed by Powerline Control Systems (PCS). This proprietary, open-source protocol consists of reliably (with packet ACKs) sending control commands as well as requesting device statuses through AC powerline wiring. Like X10, the concept is to modulate the AC carrier wave by inserting voltage pulses, as shown in Figure 11 [23]. The standard UPB packet can be seen in Figure 12 [23]. UPB packets can also be repeated (up to 4 times) by network nodes/devices, which adds to its reliability. It is claimed that UPB reaches a 99% overall reliability [24], of course in a setting where power shortages are extremely infrequent.

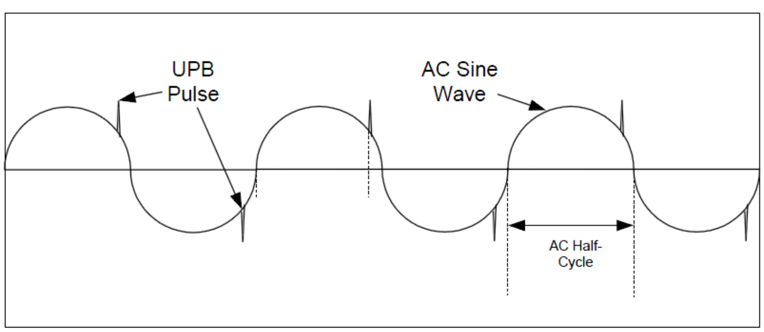

Figure 11: UPB method

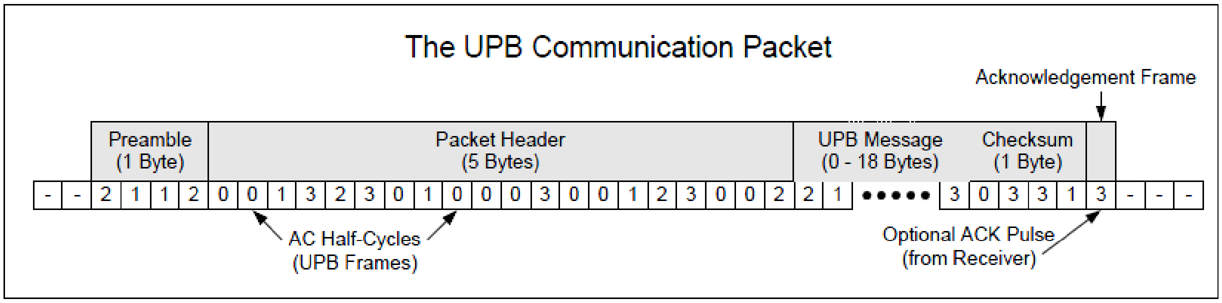

Figure 12: Standard UPB communication packet

UPB has a data rate that can reach 480 bits/sec and supports up to 250 nodes per Network [24]. UPB protocol has built-in commands for dimming and blinking at a programmable rate. The UPB components consist of a PIM-R (P/N: HDPIMRS232) as the Network access device that can connect serially (RS232) with a PC/microcontroller from one side and a wall socket from the other side. The company that provides this model also provides UPB wall switches and dimmers [25]. Unfortunately no Arduino-UPB interface was found in the literature.

The main downside is an expensive software tool.

iii. Insteon

Insteon (2005 SmartLabs company proprietary) is a propriety open hybrid method that works both on Powerline wiring and Radio Frequency. It has a typical data rate of 2,400 b/s, an RF (904 MHz) range of 46m and can support up to 16,777,216 devices per network. The device response time is around 0.4 seconds. It supports 2-way communication and is very reliable due to the dual communication medium. It is affordable (a developper's kit is 250 $ according to the SAM website). Its installation is easy, requiring only a SmartLink device that connects to the wall socket and to the local router with internet access. The router should be tweaked to allow remote control. It is suitable for sensing as well as control. Still, there is no Arduino-Insteon interface support.

iv. Z-wave

Z-wave is a protocol using the wireless medium, designed for Home Automation. Devices respond in a short time (0.05 s). It operates at 900MHz. This band might interfere with cordless telephones and other consumer electronics, but not with Wifi. It is supported by a large number of manufacturers worldwide. This proprietary standard seems perfect for Home Automation (by design), however the standard is not open and hence there are currently no protocol interfaces with Arduino. Old Z-wave products have a bit rate of 9.6 kb/s and the newer ones 40 kb/s. Z-wave has a range of 30m indoor and 100m outdoor. It can support up to 232 devices per network. Each Z-wave node can be a packet repeater. The modules consume very little power [33]. There is a small community trying to work for an open firmware for Zwave. Z-wave is also used for smoke alarms, security, consumer electronics… and is suitable for sensing and control applications.

v. Zigbee

Zigbee is a wireless standard based on the MAC/physical layer IEEE 802.15.4 protocol and was developed by Z-Alliance (proprietary and open source and released in 2007) for short range, low complexity, low power, low data rate and low cost applications. IEEE 802.15.4 can be used for mesh networking applications, thus largely increasing its range and robustness. This way, each mesh node can be used as a hop to transmit information to the final destination. Zigbee and IEEE 802.15.4 are often used interchangeably in star or tree topologies but when a mesh network is involved, Zigbee implements network layers functionalities like route discovery and multi-hopping. For instance, in a Zigbee network, if one node fails to forward information, the network uses another path. The universal frequency is around 2.4 GHz which might interfere with a 2.4 GHz Wifi (with frequency hopping/blacklisting). A Zigbee network can support up to 65,000 devices per node. It is mostly used in Wireless Sensor Networks (WSNs), wireless automatic meter reading system of water, gas, heat, electricity meters, street lights control, fire safety alarms, building monitors, laser guns… At 2.4 GHz, the data rate is 250 kb/s.

The IEEE 802.15.4 standard defines the physical layer and the medium access control layer (link layer). Using IEEE 802.15.4, we can either implement a tree-like topology or a mesh network using the Zigbee network layer options. Moreover ZigBee has a proprietary protocol suite for Home Automation and is compatible with Arduino. The literature review indicates that the Xbee module (a chip with analog and digital input/output and an antenna, servicing either the IEEE 802.15.14 standard only or ZigBee) is extensively used with the Arduino, and hence we have good documentation if we choose this technology along with Arduino.

Next, we look at other wireless options for completeness.

vi. Wifi

The IEEE 802.11 (known as Wifi 1999) is designed for high data rate / Internet access usage (WLANs). It is a reliable (ACKs are implemented), wireless protocol operating at 2.4 and 5 GHz with data rates reaching 54 Mb/s. Its typical range is 32m indoors to 95m outdoors. It supports up to 2^{32} nodes per network. It supports WEP, WPA and WPA2 encryption. Some of its drawbacks are the high power consumption and high cost. Very few if not none of Home Automation implementations using Wifi were found in the literature. Arduino is Wifi compatible (Wifi shield costs around $85). Recently, new Wifi-based protocols called “Low Power Wifi” have been released with different characteristics (lower power, range and data rate) and may be more suitable for our project description (40 $/module). Still, we did not find Low power Wifi - Arduino interfaces in the literature.

vii. Bluetooth

Bluetooth v2 is a proprietary and open-source protocol. It has been used in some small-scale projects found in the literature. However, it will be directly abandoned for our project due to its short communication range, high power and complicated networking setup.

viii. Custom RF protocol

One possibility is to make our own ISM wireless protocol, or improve on an existing one (following a certain IEEE standard). Although it will require more effort, we would design it in a way so as to meet our project objectives perfectly. Home automation being an application where current networking technologies are more than sufficient, we will abandon this direction.

- <u>Comparison</u>

Figure 13 and tables 1 & 2 below compare further the wireless techniques for Home Automation. The comparison table 3 is a comprehensive table of all listed protocols and their specifications regarding our requirements.

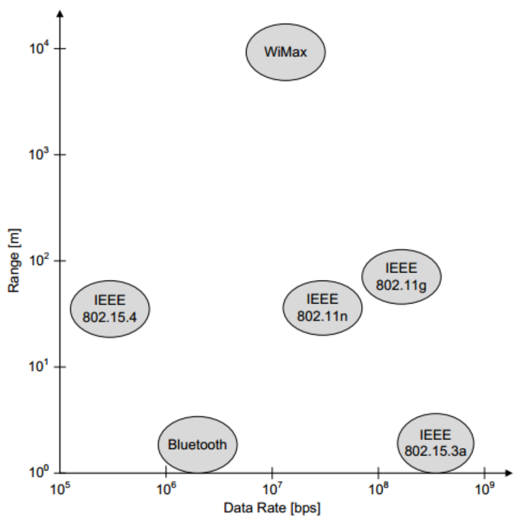

Figure 13: Range vs Data Rate for different protocols

Protocol	max. number of devices	range indoor	range outdoor
Dash7	2^{32}	10 m	10 km
ONE-NET	4096	100 m	500 m
EnOcean	>4000	30 m	300 m
Z-Wave	232	30 m	300 m
WiFi n	2^{32}	70 m	250 m
Zigbee	1024	30 m	200 m
6LoWPAN	100	30 m	200 m
WiFi g	2^{32}	40 m	140 m
WiMi	1024	125 m	550 m
Bluetooth	2^{16}	10 m	100 m
Insteon	2^{24}	n.a.	45 m

Table 1: Maximum number of devices and range for different protocols

Protokoll	power consumption sleep	transmit	receive
IEEE 802.15.4	0.06 μW	36.9 mW	34.8 mW
ANT	1.8 μW	39 mW	33.9 mW
ONE-NET	0.3 μW	63 mW	57.9 mW
EnOcean	0.6 μW	99 mW	72 mW
Z-Wave	8.25 μW	75.9 mW	120 mW
WiMax	33.6 μW	224 mW	358 mW
Bluetooth	330 μW	215 mW	215 mW
WiFi	6600 μW	835 mW	1550 mW

Table 2: Power consumption for different protocols

	UPB	X10	ZigBee	Zwave	Insteon	Custum RF	WiFi (WLAN)	Bluetooth 4 Low Energy	Low Powe WiFi	Bluetooth 2.0
Communication Medium	Powerline	Powerline	Wireless (2.4G|868M|900sM Hz)	Wireless (900 MHz)	Hybrid (RF/Powerline)	Wireless	Wireless (2.4 & 5 GHz)	Wireless (2.4 GHz)	Wireless (2.4 GHz)	Wireless (2.4 GHz)
Typical Data Rate (bits/sec)	480	60 (1 cmd/s)	< 250k	40k or 9.6k	2400	10k	< 54M	< 1 M	1M and 2M	1 to 3 M
Max Range (m)	N/A	N/A	50i - 90o	30i - 100o	46	?	32i-95o	50	50i - 250o	5 or 10 or 100
Max nodes / Network	250	256	65,000	232	16,777,216	p	2^32	8	x	8
Avg Response Time (sec)	0.3	2	0.001	0.05	0.4	x	x	0.006	x	0.1
Feedback (2 way comm) ?	yes	no	yes	yes	yes	p	yes	yes	yes	yes
Intrinsic Security ?	no	no	yes (AES 128)	no	yes	p	yes WPA2	yes 128-bit AES	yes	yes 56/128-bit + app layer
Overall Reliabililty (%)	99%*	70-80%*	High (mesh - repeater)	High (mesh - repeater)	High (Hybrid/mesh)	p	High	High	High	High
Power Consumption	n/a	n/a	very Low 3VDC 40mA peak	Low	Low	Low	High 4.2 V 140mA peak	very Low peak <20 mA	Medium 7.5uA sleep <200mA active 10 mW RF	Medium 3.6V 40mA max
Battery Life	n/a	n/a	Years	Years		Years	Hours-Days	Years	Years (on lithium AA)	Days
Price	160 $ UPB controller	low (!220V)	23 $ / Xbee transceiver		dev kit 250 $	cheapest	Xbee module 80 $	x	40 $ / module	15 $ BT/ Beetransceiver
Prefab Components price	73 $ module	cheap	expensive (ZigBee certified)		inexpensive	cheapest	expensive	cheap	x	cheap
Arduino Compatible ?	no	yes PSC05	yes Xbee shield (23 $)	no	no	p	yes Wifi Shield (85$)	no	no compatibility	yes BT sield 20 $
Proprietery ?	yes	yes	yes		yes	no	no	yes	no	yes
Open Source ?	yes		yes	no	yes	n/a	yes	yes	yes	yes
Release Date	1999	1970s	2007		2005	n/a	1999	2007		
Additional options	dim/blink@ rate	dim				p			Timer and Event Triggered	
Other uses	none	none	industrial energy management personal health care	suitable for control and sensing apps	none		Wireless LAN connectivity broadband Internet acces	connectivity between devices such as phones, PDA, laptops,	Auto-Reporting Capability	
website			zigbee.org	z-wavealliance.org						
Scalability	Powerline Control Systems Inc	lots of X10 products no sensing	might not work between different companies	works between different companies over 160 manufacturers	SmartLabs company only, compatible with X10					
interference	low	yes	wifi	some cordless phones consumer	x	depends on Freq		wifi		wifi
set up	easy, but expensive software tools	very easy	medium	medium	easy SmartLink router	medium	medium	medium	medium	medium

Table 3: Comparative table summarizing all protocols

The Arduino incompatibilities of several technologies leave us with the following options:

- X10
- ZigBee/802.15.4
- Wifi
- Bluetooth v2
- Design an interface for desired non-compatible technique (except Z-wave which is not open source)

Based on Table 2 above, the Bluetooth v2 and Wifi consume a lot of power in exchange for a high data rate. We do need a decent data rate for our project (measurements transmission mainly), however a 250 kbps is good enough, at least for data measurement and simple commands transmissions. Moreover, Wifi is expensive relatively to the others. If we drop these two, we end up with X10, ZigBee/802.15.4 and the custom ISM RF. One suggestion is to use a combination of X10 (for simple light/appliance switching) and ZigBee/802.15.4 (monitoring/sensing/security), since X10 is cheap and simple to implement and its weaknesses can be negated by a wireless protocol's intelligence.

- <u>Decision</u>

[1] concludes that ZigBee, unlike Z-wave, INSTEON, and other IP-based solutions were designed for general purposes. This reinforces our decision to use ZigBee as a combination with the simple X10 and gives our system more flexibility while increasing its scalability.

The time constraint will most probably not allow us to build a new protocol that serves the project objectives or develop protocol interfaces; nevertheless we will keep in mind the possibility of improving on the methods we are using.

The Xbee chip will constitute the main nodes in our wireless network. The specifications of Xbee (Figure 14) are shown in Table 4.

Specification	XBee
Supply Voltage	2.8 VDC – 3.4 VDC
RF Power	0 dBm, 1 mW
Outdoor Distance (LOS)	300 ft (90 m)
Indoor Distance	100 ft (30 m)
Current Draw, Receive	45 mA
Current Draw, Transmit	50 mA
Current Draw, Sleep	< 10 μA
RF Data Throughput	250 kbps
Operating Frequency, Channels	2.4 GHz, 16 Channels
Receiver Sensitivity	-92 dBm

Table 4: Xbee's specifications, source [26]

Figure 14: XBee 1mW Wire Antenna, source [27]

- **<u>Xbee 802.15.4 or Xbee ZB ?</u>**

Xbee ZB modules use the ZigBee protocol. The latter uses three types of nodes in a network:

- Coordinator: Master node in the Network
- Router: Routes packets
- End device (mote): typically sensor nodes; they send to coordinator directly or via router(s)

Note that Coordinator and Routers are always active (no sleep option means more power) and cannot be powered by batteries for a long time, as opposed to the End devices (most of the time sleeping).

9.1.7 *Alternatives and Decision for the House network*

In the ZigBee Star topology (or more generally tree topology if more routing nodes are added), any communication between two nodes has to pass through the central Coordinator (routed by the Routing nodes). The topology is illustrated in Figure 15.

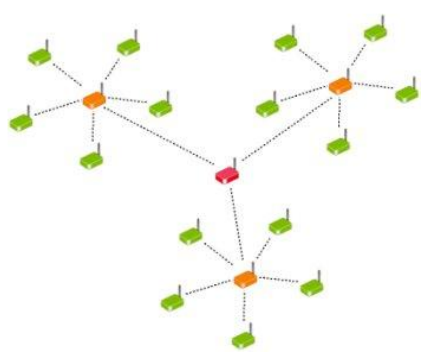

Figure 15: Zigbee star topology, source [28]

Xbee 802.15.4 modules can talk to each other using the P2P or Ad-Hoc fashion. Each module can talk to 1 or multiple “brother” module(s) in range. However, to create a mesh-network (with packet forwarding), each module should be an End Device and Router at the same time. This added functionality can be achieved using the DigiMesh protocol over the IEEE standard. With DigiMesh, all nodes become equal, can forward packets and sleep when inactive (higher range, higher reliability and lower power). The topology we get is illustrated in Figure 16 below:

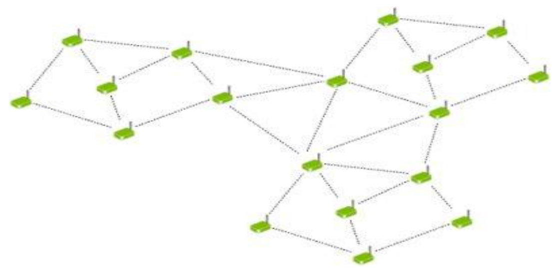

Figure 16: Zigbee mesh topology, source [28]

In our project, communication between two end-nodes is not crucial. In fact, by nature, the Home Automation framework is highly centralized. We may have conditional actions between two end-nodes for instance (for example switch the light on when someone is approaching), but since the central controller has to have knowledge about the status of each end-node, hopping to the Coordinator node is inevitable. This is why we chose to implement a tree topology, where routing nodes have the role of hopping information to faraway end nodes, and possibly, aggregating sensor data coming from different sensors to send them in a single packet (and hence reduce traffic load in the network).

9.1.8 *Alternatives and Decision for Data Management*

The system will be collecting sensor data to be used for fusion, learning and decision making. This data along with other data (user customizations, list of nodes in the house network...) has to be stored somewhere, and if important information is involved, a backup may be needed. The Arduino Ethernet shield has an SD card slot that can accommodate an extra storage with enough memory for information that is frequently updated. However, for semi-permanent data (like user configuration and network nodes), these have to be backed up on an external server like a cloud server (free services are available online). Storing and graphing of sensor data can easily be done using Cosm cloud services that are primarily designed for easy Arduino interface.

Using the cloud as a storage/backup is a desirable feature for reliability (in case the central controller fails) and synchronization between different users on different mobile devices (no need to make the already overwhelmed Arduino send each time the whole network information to the Android App instantiation.

9.2 Design Iterations

This section provides a chronological summary of the reasoning explained in the above sections.

The first step in our design was to choose an appropriate communication protocol. At first X10 seemed like a suitable option. However, it turned out to have some problems (slowness, unreliability, no collision detection…). That’s why we decided to combine it with another wireless communication protocols.

We did a comparative study to evaluate the available protocols based on several criteria such as cost, typical data rate, average response time, compatibility with Arduino... Following that study, we settled on Zigbee as our wireless communication protocol for its practicality, reliability and most importantly for its scalability. Indeed, the tree topology in Zigbee is appropriate for range issues and some can be configured as routing nodes to provide the scalability.

Ultimately our system is a hybrid one and X10 will be integrated for specific appliances and connected to the Arduino and home power lines.

Next, we had to decide on the central controller that will be used in our project. The Arduino microcontroller is a project constraint, but there exists a multitude of Arduino on the market to choose from, with different characteristics (operating voltage, SRAM, Flash memory, clock speed...). After some research, we opted for the Arduino Due for its powerful capabilities. Unfortunately, it was out of the stock on all online stores. Consequently, we decided to work with the Arduino Mega. We also chose to mount an Ethernet shield on the Arduino board to provide the connectivity.

Finally we discussed our data management plan. Our first thought was to store data on the SD card available on the Ethernet shield. However, we found that it would be useful to have an external server like a cloud server to store a particular type of data like user configuration and network nodes database. So we decided to use Cosm cloud services that could easily be interfaced with our Arduino.

9.3 Final Design Picture

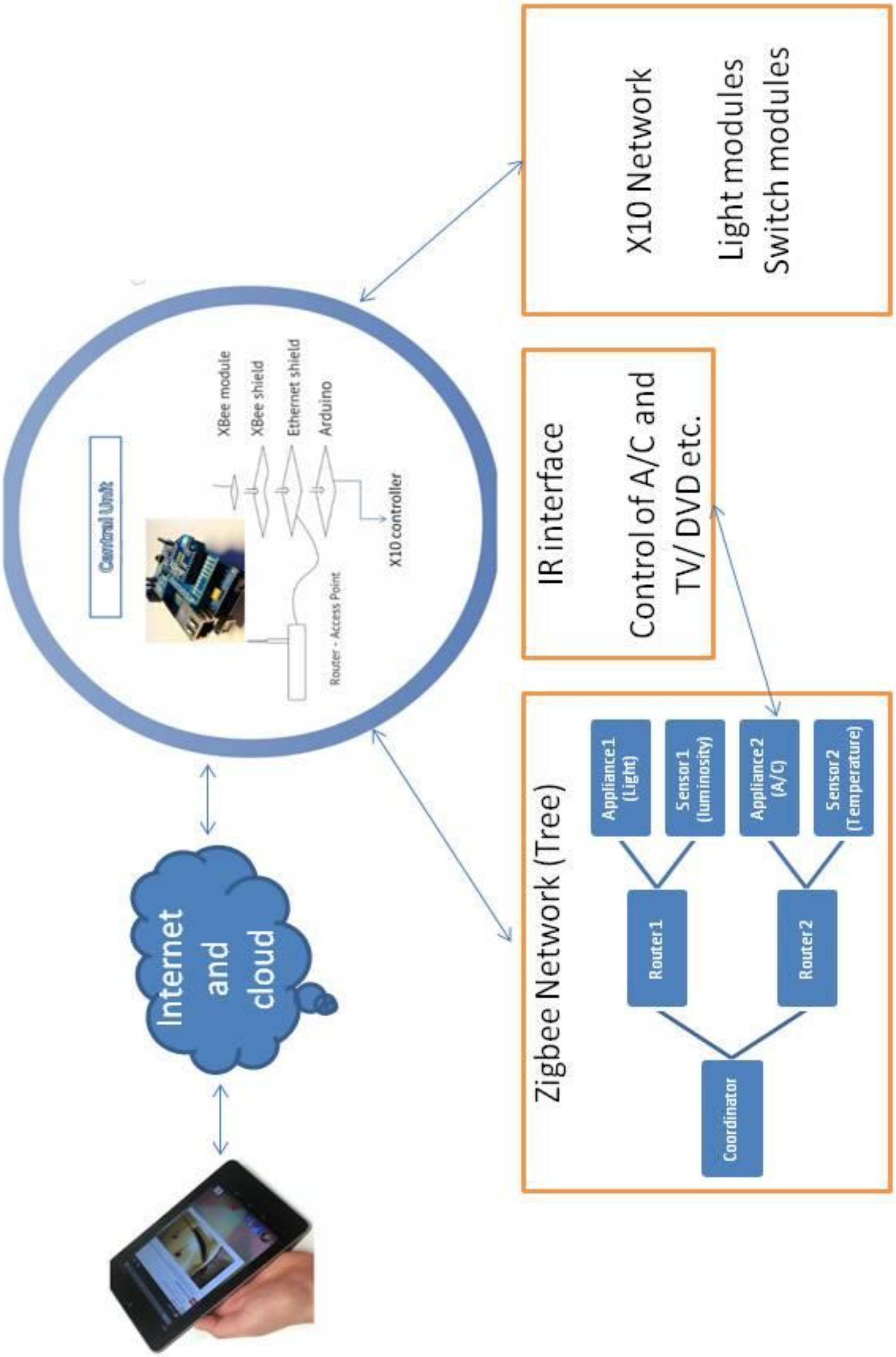

Figure 17: Picture of the final Design (not all end nodes have been mentioned for clarity)

10 Preliminary Implementation and Testing

We were offered an Android device around mid-semester so we started implementing the Android software and GUI for the user to manage and access his house network. We also bought an Arduino and an Ethernet shield from a local provider to start our implementation early, but unfortunately, no XBee or X10 devices were found.

10.1 Android Software

We programmed an Android application that allows the user to set up new devices added to his network. The application supports both X10 and ZigBee protocols, in accordance with our design decision. It still needs some rigorous work for completing it and improving the user interface. In addition, we did not limit the application functionality for Home Automation, but rather for any kind of Network, thus increasing its scalability.

<u>Nodes Page</u>

When the user runs the application, a "Nodes" page appears where all previously created nodes are displayed in a Gridview along with their corresponding names. The following Figure illustrates the basic design.

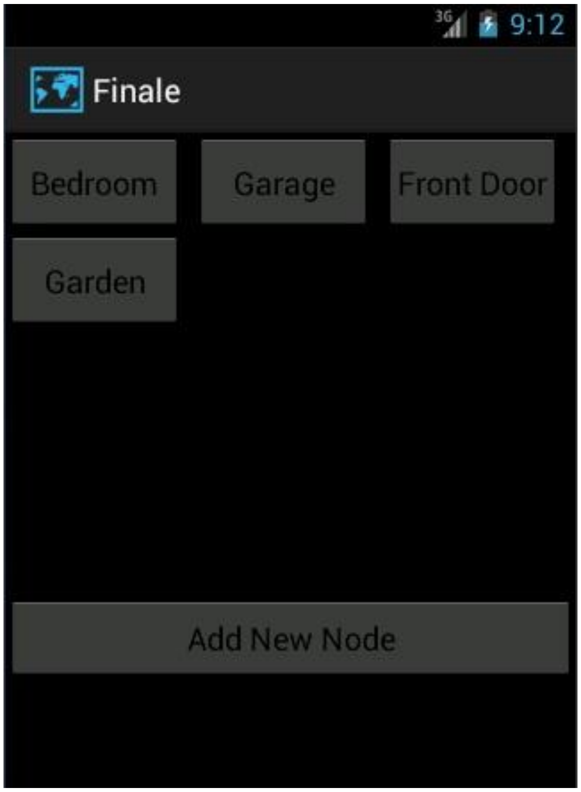

Figure 18: Basic design of "Nodes" page

Here, the user has the option of either clicking on a specific Node, or adding a new Node.

- Adding a new Node

When the user clicks the "Add New Node" button, a new page appears with labeled fields:

Name	Textbox for entering a user chosen name for the Node
Type	A drop down list with options of: XBee, X10
Address	Address of the Node

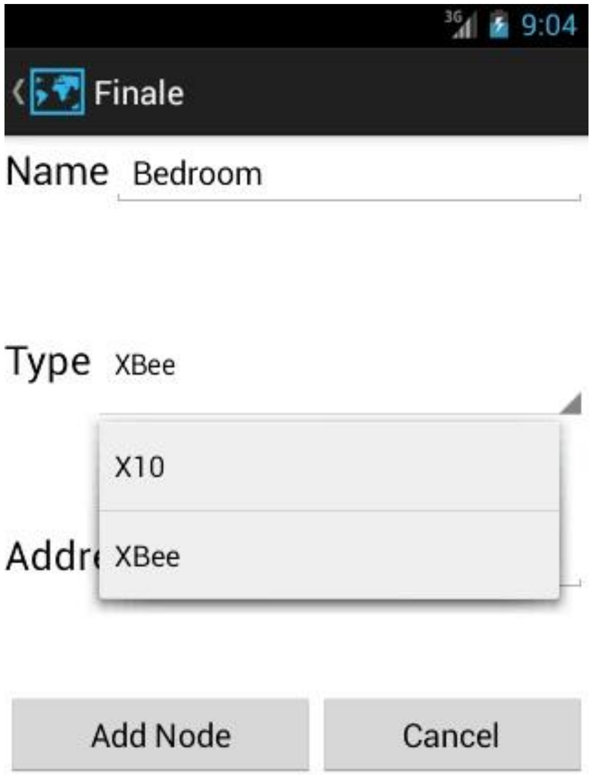

Figure 19: "Add New Node" page

The user can either Cancel and return to the main Nodes page, or click Add Node, and the Node he just added will appear on the GridView of the Nodes page to which he will be redirected.

- Clicking on a specific node

Once he selects a node from the Gridview of the Nodes page. He gets directed to a new page, where all information about the node is displayed. For an X10 node, he would have reached an end device with a Unique address. Here he can toggle the device ON/OFF or DIM...

For an XBee node (XBee is just a microcontroller-ZigBee complex), he has the option to add new Devices to the node or Control/Monitor already added devices. If he had previously added a sensor device, he can monitor its status or value on this page. All devices he adds to the node are displayed in a ListView as shown in the Figure below. Till now, only the Switch device is implemented.

Figure 20: “Clicking on a specific node” page

If the user selects "Add New Device", he is directed to a device configuration page like the one shown below.

Figure 21: "Add New Device" page

Name -- Name of device

Type -- Drop Down List Choices:

- Switch
- Boolean Sensor
- Value Sensor
- Dimmer
- Camera

Pin #-- The XBee pin number on which the device is connected.

Template -- For Better User Interface

Drop Down List Choices

- Temperature Sensor

- Water Level Meter
- Motion Sensor
- Camera

After the user selects "Add Device", his/her device will now be visible in the ListView of the corresponding Node. If he/she chooses a Switch, a ON/OFF switch will appear as well in the ListView next to the Device Name. For a Value Sensor, a value display with the corresponding value will displayed next to the sensor's name. He/she can also view more device details by clicking the device's name in the ListView. He/she can refresh any device by clicking the local refresh button, or refresh all the nodes by clicking on the global refresh button.(also can refresh all Network) This will poll the network and get the most recent values of the devices (sensors and switches).

As we have said earlier, the Switch device is now fully functional. Once the user clicks on a Switch, (for example turning on the Bedroom Lamp), an http Post command is sent to a web server hosted on the Arduino. The Post message contents are as follows:

NodeTy type_of_node (X10 or XBee)

NodeAd address_of_node (address of node either X10 node address or XBee module address)

Pin pin_number (only valid for XBee nodes)

Type type_of_device_on_PIN

possibilities of Type:

SW (switch)

AS	(sensor with numeric value)
DS	(sensor with boolean value)
DIM	(dimmer)
CAM	(webCam)

NewVal   New_Value_of_device   (only exists if the device can be controlled --> not sensors)

NewVal can be 1 or 0 for a SW and a numeric in a range for a DIM.

The Arduino is responsible for parsing the Http message, retrieving the information, forwarding them to the network and replying to the Android with the appropriate response.

We tested the code above on a computer instead of an Arduino with Ethernet shield. On the computer, we ran an ISS script with a PHP script that parses the Post message and writes its values to a text file. Our testing was successful, and the following was displayed in the Text file when toggling a Switch on and off.

```
NodeType: XBee

NodeAddress: 777

Pin#: 123

DeviceType: SW

NewVal: 1


NodeType: XBee

NodeAddress: 777

Pin#: 123

DeviceType: SW

NewVal: 0
```

NodeType: XBee

NodeAddress: 777

Pin#: 123

DeviceType: SW

NewVal: 1

10.2 Arduino programming and electronics

For our prototype, we used an Arduino UNO board with Ethernet shield to provide local network connectivity (Android tablet connected on the same network wirelessly). We will send commands from the Android tablet to the Arduino board, and get sensor readings from the Arduino to appear on the Android application GUI.

We coded a simple Web Server on the Arduino board that is able to connect with clients and send sensor information as an HTML file (see appendix for full code). In the few days remaining, we will synchronize the Arduino to the Android app to be able to control LEDs on a breadboard and automate the lighting intensity of one of the LEDs based on the readings of a temperature sensor.

11 List of Resources and Engineering Tools Used/Needed

Below is the list of Hardware and Software we will need to complete our project

Hardware Item Name	Item Price ($)	Count	Total Price
Arduino Mega 5620 (Arduino Due out of stock)	50	1	50
Arduino Ethernet Shield	55	1	55
Arduino Xbee Shield	17.09	1	17.09
XBee 2mW PCB Antenna - Series 2	25.95	6	155.7
Xbee Explorer USB	24.95	1	24.95
Xbee Explorer Regulated	9.9	4	39.6
OpenWRT Router	25	1	25
Ethernet Wire	3	2	6
X10 Two way PLC TTL CMOS Interface XM10	42.46	1	42.46
Screw in X10 dim lamp module LM15E	35	1	35
x10 switch AM12	35	1	35
IR transmitter	2	3	6
IR receiver	10	1	10
Power Tail Switch II	26	2	52
Non-Invasive Current Sensor - 30A	9.95	4	39.8
LP2950-33 voltage regulator	0.57	5	2.85
Switch - Floor Mat - 14 Inches x 30 Inches	14.95	1	14.95
Ultrasonic Module HC-SR04 Dist Meas. Trans. Sensor	5.83	1	5.83
TMP36	2	4	8
Mini Photo Cell	1.5	4	6
Directed 8600 Magnetic Micro Switch	9.99	1	9.99
PIR Motion Sensor	9.95	1	9.95
WebCam	6.48	1	6.48
USB Cable. Standard A-B	4	1	4
USB A to Mini B cable	1.41	1	1.41
Adapter for Arduino Mega 220AC in/ 7-12 V out	6.65	1	6.65
9V Battery Ni-MH	2.91	3	8.73
Adapter for Xbee regul/volt Regul 220AC in/ 5-16V out	6.65	1	6.65
Pack Way Plug Wireless RemCont Out Switch 220V 3-Socket	26	1	26
Transistors/Capacitors/Resistors/Wires/Soldering…	x	x	X
RF Link Transmitter - 434MHz	3.95	1	3.95
RF Link Receiver - 4800bps (434MHz)	4.95	1	4.95
			719.99

Software	Usage
Android Developer Platform	Develop the GUI
Android OS	Demonstration
Arduino IDE developer tools	Develop the Arduino Program
Digi-mesh	Runs on Xbee
X-CTU / FTDI drivers	Control PC ports Xbee testing
NewSoftSerial library	Collect signals from Xbee
X10 library	Interface Arduino – X10

12 Detailed Project Plan and Schedule

Our plan for the spring semester can be divided into 3 main phases. Each phase is subdivided into a hardware part and a software part (Arduino and Android programming).

Phase 1

1. Hardware

- Assemble the house network and connect the hardware
    - Build the stack (Arduino, Xbee/Ethernet shield)
    - Connect the X10 controller to the Arduino and home powerlines
    - Connect the end nodes to Xbee pins
    - Install X10 actuators (switches, sensors..) to appliances
    - ConFigure X10 devices and Xbee chips
    - Set up a network of Xbees
- Test the assembled network by sending dummy requests

2. a) Android Software

- Conceive the GUI of the Android application
- Ability to add nodes and categorize them based on their type (Xbee/X10)
- Create predefined templates for typical sensors
- Poll for sensors readings
- Test successful control over house network using local IP address
- Verify that state of appliances matches on Android and Arduino

b) Arduino Software

- Implement a database that stores the current state of each controlled appliance
- Synchronize the Arduino with the Android application
    - Test the capacity to convert HTTP request to Xbee/X10 commands

Phase 2

1. Hardware

- Add more nodes to the house network
- Improve the scalability of the network
    - Possibly test a more sophisticated topology

2. a) Android Software

- Multi-user handling
- Add notifications/warnings when special events occur
- Graphing capability for sensor readings
- Test successful remote access and control of house network

b) Arduino Software

- Implement a learning algorithm on Arduino (or cloud server) capable of suggesting a schedule to the user
    - Test the scheduling algorithm
- Create a data backup on the cloud

Phase 3

1. Hardware

- Improve the reliability of the X10
- Minimize the number of Xbees by grouping similar end nodes together (an Xbee weather station for example)
    - Test the new configuration
- Add interface to music/stereo control

2. a) Android Software

- Conflict management
    - Add notifications/warnings when a conflict arises
- Voice recognition

b) Arduino Software

- Conflict handling when two (or more) users are sending a request

The Gantt chart below shows our work plan for the spring semester. All tasks related to hardware are colored in green while the software activities are either highlighted in blue (Android programming) or red (Arduino programming)

ID		Task Name	Duration	Start
1		**Phase 1**	**22 days**	**Fri 01/02/13**
2		Assemble the house network and connect the hardware	12 days	Fri 01/02/13
3		Test the assembled network by sending dummy requests	6 days	Mon 18/02/13
4		Finish the GUI of the Android application	15 days	Fri 01/02/13
5		Create predefined templates for typical sensors	10 days	Mon 18/02/13
6		Poll for sensors readings	6 days	Sun 24/02/13
7		Implement database storing current state of controlled appliance	14 days	Fri 01/02/13
8		Synchronize the Arduino with the Android application	5 days	Mon 25/02/13
9		**Phase 2**	**24 days**	**Mon 04/03/13**
10		Add more nodes to the house network	10 days	Mon 04/03/13
11		Test a more sophisticated topology	5 days	Fri 15/03/13
12		Multi-user handling	14 days	Mon 04/03/13
13		Graphing capability for sensor readings	10 days	Mon 04/03/13
14		Implement and test a scheduling algorithm	14 days	Mon 18/03/13
15		Create a data backup on the cloud	7 days	Fri 22/03/13
16		**Phase 3**	**14 days**	**Fri 05/04/13**
17		Improve the reliability of the X10	7 days	Fri 05/04/13
18		Add interface to music/stereo control	10 days	Fri 05/04/13
19		Conflict management	14 days	Fri 05/04/13
20		Voice recognition	14 days	Fri 05/04/13
21		Conflict handling when two users are sending a request	14 days	Fri 05/04/13
22		Write the final report and prepare for the presentation	7 days	Thu 25/04/13
23		Final presentation	0 days	Mon 13/05/13

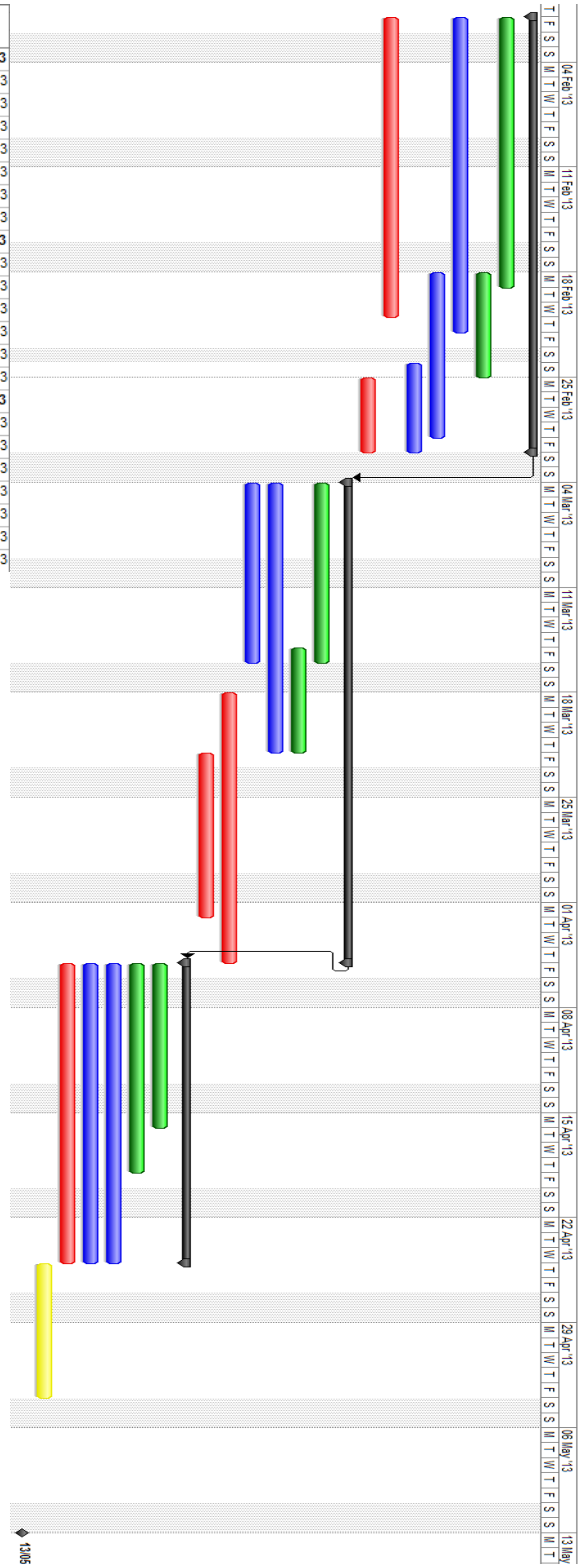

Figure 22: Gantt chart

13 References

[1] C. Gomez and J. Paradells, "Wireless Home Automation Networks:A Survey of Architectures and Technologies," IEEE Communications Magazine, 2010.

[2] "SMARTPHONE USERS AROUND THE WORLD - STATISTICS AND FACTS," 1 2 2012. [Online]. Available: http://www.go-gulf.com/blog/smartphone.

[3] A. Rosendahl, J. F. Hampe and G. Botterweck, "Mobile Home Automation - Merging Mobile Value Added Services and Home Automation Technologies," in Sixth International Conference on the Management of Mobile Business (ICMB 2007), 2007.

[4] "how can i be efficient in my home?," Harper Technology Group, [Online]. Available: http://www.harpertechnologygroup.com/division/facts.html.

[5] "WHAT CAN WEMO DO?," [Online]. Available: http://www.belkin.com/us/wemo.

[6] "MisterHouse -- It Knows Kung-Fu," 3 12 2008. [Online]. Available: http://misterhouse.sourceforge.net/.

[7] "What is Home Easy?," 2010. [Online]. Available: http://www.homeeasy.eu/WhatIsHomeEasy/.

[8] I. Petrov and S. Petrov, "uControl Solutions".

[9] A. ElShafee and K. Alaa Hamed, "Design and Implementation of a WiFi Based Home Automation System," World Academy of Science, Engineering and Technology, 2012.

[10] "Home Automation server with router," [Online]. Available: http://www.instructables.com/id/Home-automation-server-with-router/.

[11] D. Banks, "Home Automation Iphone Application," 2010. [Online]. Available: http://www.utm.edu/departments/engineering/documents/Home Automation using x10 network.pdf.

[12] A. San-Salvador, "Contacting the Devices: A Review of Communication Protocols," [Online]. Available: http://gicap.ubu.es/publications/2012/PDF/2012_c05_Contacting.pdf.

[13] "X10," [Online]. Available: http://en.wikipedia.org/wiki/X10_(industry_standard).

[14] J. A. Infantes Diaz, "WIRELESS SENSOR NETWORKS CONTROLLED WITH PIC MICROCONTROLLERS AND ZIGBEE PROTOCOL," Mikelli University, May 2012.

[15] J. Gottfried, "DIY Home Automation Using Twilio, PowerSwitch, Arduino, and Pusher," 21 August 2012. [Online]. Available: http://www.twilio.com/blog/2012/08/diy-home-automation-using-twilio-powerswitch-arduino-and-pusher.html.

[16] "Smart Home Hacks," [Online]. Available: http://shop.oreilly.com/product/9780596007225.do?sortby=publicationDate.

[17] "Home Automation with arduino and android," [Online]. Available: http://arduino.cc/forum/index.php/topic,8826.0.html.

[18] jfurcean, "Arduino X10 Home Automation," [Online]. Available: http://thenewtech.tv/tech-life/arduinox10-home-automation-over-wifi.

[19] [Online]. Available: https://circle.ubc.ca/bitstream/handle/2429/43231/Cramb_M_et_al_ENPH_459_2012.pdf?sequence=1.

[20] R. J. C. Nunes, "Home Automation - A Step Towards Better Energy Management," [Online]. Available: http://www.icrepq.com/pdfs/CALEIRA416.PDF.

[21] N. Banerjee, "Automating Energy Management in Green Homes," [Online]. Available: http://conferences.sigcomm.org/sigcomm/2011/papers/homenets/p19.pdf.

[22] "Intelligent Home Automation System," [Online]. Available: http://www.netvox.com.tw/download/Netvox%20Catalog(English%20Version)%2020120628.pdf.

[23] "UPB Technology Description," [Online]. Available: http://www.pulseworx.com/downloads/upb/UPBDescriptionv1.4.pdf.

[24] "UPB Home Automation," [Online]. Available: http://www.hometheaternetwork.com/automation4.php.

[25] "Universal Power Bus," [Online]. Available: http://www.homeauto.com/Downloads/ConnectivityPartners/AlloysUPBTechSheet.pdf.

[26] M. Hebel, "Getting Started with XBee RF Modules," [Online]. Available: http://www.makershed.com/v/vspfiles/assets/images/122-32450-xbeetutorial-v1.0.1.pdf.

[27] "XBee 1mW Wire Antenna (802.15.4) - Series 1," [Online]. Available: http://www.parallax.com/Store/Accessories/CommunicationRF/tabid/161/List/0/ProductID/639/Default.aspx?SortField=ProductName%2CProductName.

[28] D. Gascón, "802.15.4 vs ZigBee," 17 November 2008. [Online]. Available: http://www.sensor-networks.org/index.php?page=0823123150.

[29] U. E. I. Administration, "Annual Energy Review2011," September 2012. [Online]. Available: http://www.eia.gov/totalenergy/data/annual/pdf/aer.pdf.

[30] A. P. Society, "Buildings," [Online]. Available: http://www.aps.org/energyefficiencyreport/report/energy-bldgs.pdf.

[31] E. B.Driscoll, "The history of X10," [Online]. Available: http://home.planet.nl/~lhendrix/x10_history.htm.

[32] A. P. Society, "Buildings," [Online]. Available: http://www.aps.org/energyefficiencyreport/report/energy-bldgs.pdf. [Accessed January 2013].

[33] Z.-w. alliance, "About Z-Wave Technology," [Online]. Available: http://www.z-wavealliance.org/about-z-wave.

[34] M. Weeks, "The Great Northeast Blackout of 2003," 2 November 2011. [Online]. Available: http://mattweeks.hubpages.com/hub/The-Great-Northeast-Blackout-of-2003.

14 Appendix

14.1 Updated Project Description and Agreement Form

Faculty Supervisor	**Prof. Rouwaida Kanj**
Co-Supervisor [optional]	Prof. Ayman Kayssi
Sponsor [optional] *Is there industry support or funding the project?*	Intel
Project Title *Descriptive title* *not necessarily the final title that will be adopted by the team*	Arduino/Android Home Automation System
Project Description and Design Aspects *What is the main motivation for the project?* *Specify the desired needs that the final product is expected to meet.*	Motivation: As devices get smarter, a lot of effort is aimed at automating many day-to-day activities. Imagine turning on the AC 30 minutes before your arrival, or opening the door lock using your mobile. Exploiting technology is key and critical for safer and better future. Desired Needs: The goal of this project is to build a Home Automation system using a low-cost Android smartphone or tablet as the controller, and the Arduino device to offer the electronics connectivity. Together, applications such as Home Automation power control system and home digital timer controller can be enabled.

Expected Deliverables *Required deliverable(s) from the team at the conclusion of the design project*	Build a Home Automation system using a low-cost Android smartphone that can send commands to an Arduino device that controls appliances. Applications such as Home Automation power control system and home digital timer controller can be enabled.
Technical Constraints *A preliminary list of multiple realistic technical constraints, e.g. power, accuracy, real-time operation , … The technical constraints included should be detailed and specific to the design project not generic.*	• GUI should be unsophisticated and accessible to ordinary users • Network should be able to route measurements (current, temperature…): decent bit rate • High reliability: Reliable connectivity to all network components • Robustness: The Network should work even with electric power outages (convenience, security and power management purposes) and hence be power efficient to run on standby batteries. • System should decrease overall household/office power consumption. • System should increase the user’s control and convenience and maintain, if not also improve security. • Design should be easily scalable • Use of Arudino as Network microcontroller and Android as software platform

Non-Technical Constraints *A preliminary list of multiple realistic non-technical constraints, e.g. cost, environmental friendliness, social acceptance, political, ethical, health and safety, etc… The non-technical constraints included should be detailed and specific to the project not generic.*	• Cost being a crucial component in our project, the estimated cost should not exceed 30% of the existent marketed solutions' price • Time: Design should be finished by the end of the Fall Semester and Implementation should be finished by the end of the Spring Semester
Resources and Engineering Tools *Identify resources and engineering tools needed and whether they are available or need to be acquired (if known), e.g. software licenses, instruments, facilities, components, …*	1- Arduino IDE developer tools 2- Android Developer Platform 3- Android OS 4- NewSoftSerial library
Possible Applicable Standards *List potential standards directly or indirectly used or involved in the project*	• X10 protocol • IEEE 802.15.4 • IEEE 802.11 • Insteon • Zigbee • Z-wave • UPB • Bluetooth

List of Disciplines *Identify at least THREE engineering disciplines (within or outside ECE)*	○ Circuits and Electronics ○ Communications and Networking ○ Hardware, Computer Architecture, and Digital Systems ○ Intelligent Systems ○ Software Engineering
Number of Students *Please consider the number of disciplines checked above*	○ 3 students

14.2 Minutes of all Meetings up-to-date

Meeting #: 1

Date: 12.12.2012	Time: from to 2 to 2:45 pm	Location: RGB 404

Meeting called by	Advisor
Attendees	Professor Rouwaida Kanj-Professor Ayman Kayssi-Kim Baraka-Marc Ghoril-Sami Malek
Minutes taker	Marc Ghobril

Agenda Item: General information about FYP

Discussion

- We discussed the possible features that we can implement in our project. We considered designing our automation system to have three main options: comfort, power management and security. However, our advisors suggested that for now we should focus on the first two options.
- We also discussed if it's better to have a small but scalable project or a large project that covers multiple aspects of the Home Automation area.

Conclusions

- Our project will be small and scalable
- In the system, there will be a central processing unit plus a user interface
- Need to come up with a plan that includes the list of tasks covered in the project with an approximate duration for each task

Agenda Item: Literature review phase

Discussion

- Need to read as much as possible about the Home Automation systems available in the market and getting familiarize with them
- Professor Kanj will provide us with useful links and research papers

Conclusions

- Each student will be responsible of researching a specific area in the project

Action Items	Person Responsible	Deadline
• Research the hardware available in the market • Figure out how it could be integrated in our project	Kim Baraka	Within two weeks
• Analyze the Home Automations systems advertised in the market • Think about ways to make our product unique	Marc Ghobril	Within two weeks
• Research about the language used in Arduino programming • Explore the Arduino codes accessible online	Sami Malek	Within two weeks

Agenda Item: Defining technical constraints

Discussion

We discussed how we are going to integrate the microcontrollers to cover a full house:

- Use of several Arduinos
- Use of a single Arduino with a hierarchical system
- Use of more complex microcontrollers with more I/O channels

Conclusions

- More research needed to discover the functionality of Arduino and possibly other microcontrollers

Action Items	Person Responsible	Deadline
• Analyzing Arduino's hardware • Looking for other microcontrollers as alternative	Kim Baraka and Marc Ghobril	By next meeting

Meeting #: 2

Date:17.10.2012	Time: from 2:00 to 3:30	Location: RGB

Meeting called by	Dr. Ruweida Kanj
Attendees	Kim Baraka, Marc Ghobril, Sami Malek, Dr. Kanj
Minutes taker	Kim Baraka

Agenda Item:

Discussion	**General Architecture based on projects found online**

The general architecture can be divided into 3 main schemes:
1 - Arduino interfaced with controllers+ Server running on PC
2 - Arduino hardwired to controllers and sensors, with Ethernet shield (communicate through HTTP)
3 – Arduino with Ethernet capabilities
4 – Arduino with Wifi antenna, installed on Wifi shield (for Internet connectivity of the board only)
5 – Ethernet-connected Arduino, connected through LEDs + optocouplers to a remote control that can control up to 8 x10 devices (we would like more than that)
6 – Ethernet-connected Arduino connected through a single pin to a 433 MHz AM transmitter that can accommodate several wireless devices (not necessarily x10).

Conclusions	**Comparison of these architectures**

Wireless connectivity to the Internet of the Arduino is not crucial -> more cost efficient to use Ethernet.
However, a wireless control of controller is very much desired -> use transmitter/receiver architecture with (theoretically) unbounded number of nodes to control. Nodes being, among other things wireless sockets, sensors, intensity controllers …

Action Items	Person Responsible	Deadline
Research more about x10	Marc Ghobril	By next week
Get more information about the project whose architecture we found the best, and get all the relevant details (if possible by contacting the author himself)	Kim Baraka	By next week

Agenda Item: Power Management

Discussion	**Brief discussion on this topic**

Power management may be hard to do, specially if each device has to have different input coming from each source.

Conclusions	We may have to group appliances together for good flexief ibility/practicality tradeoff

Action Items	Person Responsible	Deadline
Investigate about power management issues	Sami Malek	By next week

Meeting #: 3

Date: 12.12.2012	Time: from to 2:00 to 2:30 pm	Location: RGB 404

Meeting called by	Dr. Rouwaida Kanj
Attendees	Dr. Rouwaida Kanj, Dr. Ayman Kayssi, Kim Baraka, Marc Ghoril, Sami Malek
Minutes taker	Marc Ghobril

Agenda Item: Integrate Arduino with Home Automation X10 system

Discussion	

We discussed how X10 can be integrated in our system:

- The way we can send and receive X10 commands from an Arduino module
- Using the wired or wireless X10 protocol

Conclusions

- We decided that X10 can be definitely integrated in the project (unless there are simple, cheap alternatives to x10-to check)
- Connect X10 to the power lines

Action Items	Person Responsible	Deadline
• Research the open source capabilities of X10	Kim Baraka	By next week
• Verify if it is possible to connect the Arduino to the Internet, using both wired and wireless interfaces • Check if there are simple, cheap alternatives to x10	Sami Malek	By next week
• Explore all the different kinds of Arduino shields	Marc Ghobril	By next week

Agenda Item: Home Automation market

Discussion

We discussed from where we can get the parts needed for our project (Arduino, X10, sensors…). Professor Kayssi suggested that explore the Lebanese electrical companies (Debbas, Ketteneh…) that deal with automation.

Conclusions

- A study of the Home Automation market is required
- If some parts are not available, we will order them

Action Items	Person Responsible	Deadline
Look for the required parts in the local Home Automation market	Kim Baraka, Marc Ghobril, Sami Malek	Within two weeks

Agenda Item: Progress Report

Discussion

The advisors recommended that we start writing our progress that is due on November 12.

Conclusions

- Start of the writing phase for the progress report

Action Items	Person Responsible	Deadline
• We divided the report among the team and will start writing it starting this week	Kim Baraka, Marc Ghobril, Sami Malek	November 12

Meeting #: 4

Date: 12.12.2012	Time: from to 2:00 to 3:30 pm	Location: Bechtel 406C

Meeting called by	FYP progress weekly meeting
Attendees	Prof. Rouwaida Kanj, Prof. Ayman Kayssi, Kim Baraka, Sami Malek , Marc Ghoril
Minutes taker	Sami Malek

Agenda Item: Discuss alternatives to choose optimal implementation

Discussion

We discussed primarily the pros and cons of X10, Zwave and XBee techniques in Home Automation systems.

- X10 although the cheapest is not much reliable and does not work without electricity (non robust)
- XBee is extensively used with Arduino in the literature for WSNs but rarely for Home Automation
- Zwave seems good but has not yet an interface with arduino.

We also discussed the software through which the user will be able to control his home/office.

- Cloud based (HTML) => poor graphical design => but access from all devices with internet access
- Android phone target => access exclusively from android phones => but better graphical design

Conclusions

- Make a comparative study integrating all aspects of all technologies currently used in Home Automation (range, reliability, scalability, price, ...) and summarize them in a table to make an accurate and appropriate judgment.
- XBee seems promising, and why not use 2 technologies simultaneously.
- Do both cloud based and Android target

Action Items	Person Responsible	Deadline
• Detailed comprehensive and comparative search: X10, Insteon, and other (summarize in table)	Kim Baraka	By next week
• Detailed comprehensive and comparative search: Xbee, Zwave, and other (summarize in table)	Sami Malek	By next week
• Collect projects and papers that uses any of the above technologies for Home Automation	Marc Ghobril	By next week

Agenda Item: Intel sponsorship

Discussion

We discussed an email previously sent by Prof. Kanj that describes an opportunity that would allow us benefit from Intel (free hardware/ training) since our project seemed to meet their offer criteria: Energy Saving Awareness / Smart Home Automation ...

Conclusions

- Write a paragraph for Intel that describes the objective of our project and the way we're aiming to complete it.

Action Items	Person Responsible	Deadline
Write the paragraph for Intel	Sami Malek ,Kim Baraka, Marc Ghobril	Next week

Agenda Item: Progress Report due soon

Discussion

We should complete a draft of the progress report that is due soon.

Conclusions

- The Action Items of Agenda 1 and 2 constitute a major part in the progress report.

Action Items	Person Responsible	Deadline
• Each team member writes about what he researched most before integrating everything into a report.	Kim Baraka, Marc Ghobril, Sami Malek	November 5

Meeting #: 5

Date: 12.12.2012	Time: from to 2:00 to 3:30 pm	Location: RGB 404

Meeting called by	FYP progress weekly meeting
Attendees	Prof. Rouwaida Kanj, Prof. Ayman Kayssi, Kim Baraka, Sami Malek , Marc Ghoril

Minutes taker	Sami Malek

Agenda Item: Discuss alternatives to choose optimal implementation

Discussion

We discussed INSTEON and ZigBee technologies.

- INSTEON makes use of both wired and wireless techniques for its network
- Zigbee uses only wireless

We also discussed the way we should write our report.

- Detailed scientific comparison between all alternatives we find available before choosing the one that optimally serves our project objectives.

Conclusions

- Most probably we will use both wireless and wired techniques
- Finish the report before the deadline

Action Items	Person Responsible	Deadline
• Write his assigned parts of the progress report	Kim Baraka	November 10
• Write his assigned parts of the progress report	Sami Malek	November 10
• Write his assigned parts of the progress report	Marc Ghobril	November 10

Agenda Item: Intel sponsorship

Discussion

We discussed the Project Description paragraph I wrote destined to intel.

Conclusions

- Good paragraph – add some more details.

Action Items	Person Responsible	Deadline
Forward the paragraph to all and make some minor changes. (add the option of scheduling…)	Sami Malek	asap

Meeting #: 6

Date: 12.12.2012	Time: from to 2:00 to 2:30 pm	Location: RGB

Meeting called by	Prof. Kanj
Attendees	Prof. Rouwaida Kanj, Prof. Ayman Kayssi, Kim Baraka, Sami Malek , Marc Ghoril
Minutes taker	Kim Baraka

Agenda Item: Zigbee technology and its integration in our project

Discussion

- We started by updating the professors about our new findings concerning Zigbee technlogy compared to other technoloogies. We discussed whether our system has to work with any Zigbee device available in the market or only for devices that we will create.
- Adhoco devices that can form a self-organizing network. Maybe we can achieve something similar in an ultimate stage...

Conclusions

- Our primary goal is to make the system work with our own devices. Flexibility with other Home Automation devices from the market would be a plus.
- Building our own wireless energy meters (current sensors) would be a good idea.

Agenda Item: Software and Use of Cloud Computing

Discussion

- To relieve computation and storage at the microcontroller level, we discussed the possibility of using the cloud through Cosm servers.
- Till now we have focused on the computer architecture aspect of our system. We should start also thinking about the algorithms running at the microcontroller, especially concerning energy management.

Conclusions

- To use Cosm (or any other cloud-oriented approach, for example Google Drive) is practical for reliability and ease of access when opening the App from a mobile device.

Action Items	Person Responsible	Deadline
• Investigate software-related details to see how to divide our code between Xbee chips/microcontroller/the Cloud	All group members	By end of month
• Finalize our design and have a more precise list of hardware to order	All group members	By end of month
• Search local shops to check for available hardware that wouldn't have to be shipped	All group members	By next week

Meeting #: 7

Date: 12.12.2012	Time: from to 2:15 to 3:00 pm	Location: Bechtel 406C

Meeting called by	Prof. Kanj
Attendees	Prof. Kanj, Kim Baraka, Marc Ghobril, Sami Malek
Minutes taker	Marc Ghobril

Agenda Item: Progress Report

Discussion

In the meeting we mainly discussed our progress report that was submitted last week. Prof. kanj pointed to the strong parts in our report but also said that some parts need improvement for the final report. She also sought clarifications regarding some incomplete sections.

Conclusions

We decided to improve our report and correct the incomplete/missing sections taking into consideration Prof. Kanj's remarks:

- The Methodology should be more detailed and include more technicality
- Some parts of the literature review should be developed further. For example, we mentioned a scheduling algorithm without explaining it in details. This has to be rectified for the final report
- Show clear distinction between the standards IEEE 802.15.4/ IEEE 802.11 and Zigbee
- Subdivide the "Network" section in the Progress part in a better way (wired/wireless protocol…)
- Subdivide the "Comparison and Decision" section into one section concerning the comparison and another for the decision
- Complete the "List of Resources and Engineering Tools Needed" section with a list of price for all the hardware
- Correct the typo mistakes

Action Items	Person Responsible	Deadline
• Improve our report and correct the incomplete/missing sections	All group members	By next week

Agenda Item: Getting familiar with Arduino programming

Discussion

We also discussed briefly how we can start getting more familiar with Arduino and its programming. Prof. Kanj gave us an Arduino (not the one we will use in the project) so we can start "exploring" it and get familiar with its functions. We also talked about the need to research the local market to look for the hardware we will need in our project and to make a list of the devices prices.

Conclusions

- We have to look in the local market for the hardware needed
- Make a list of the devices price
- Get familiar with Arduino programming

Action Items	Person Responsible	Deadline
• Get familiar with Arduino programming and "explore" the Arduino handed to us by professor Kanj	Kim Baraka and Sami Malek	By next week
• look in the local market for the hardware needed	Marc Ghobril and Sami Malek	By next week
• Make a list of the devices price	Marc Ghobril and Sami Malek	By next week

Meeting #: 8

Date: 12.12.2012	Time: from to 2:00 to 2:30 pm	Location: RGB

Meeting called by	Prof. Kanj
Attendees	Prof. Kanj, Prof. Kayssi Kim Baraka, Marc Ghobril, Sami Malek
Minutes taker	Kim Baraka

Agenda Item: Progress/Final Report and Finalizing our Design

Discussion

Wrap up of progress report discussion with Prof. Kayssi who didn't attend last meeting.

Conclusions

- Need to think about our final report. It has to be persuasive; we should add more quantitative elements to it. Ex: How much memory is needed; how much data rate?

Action Items	Person Responsible	Deadline
• Improve on the progress report by carrying out refined research (final stage) and quantitative analysis.	All group members	By next week

• Final design and list of hardware should be ready by next week. Local options for hardware are preferred, especially for the prototype.	All group members	By next week
Agenda Item: An extra Idea		
Discussion		
Why not use a GPS tracker in Android app and send a notification to Arduino when the user approaches?		
Conclusions		
May be added in a later stage.		
Action Items	**Person Responsible**	**Deadline**
Meeting #: 9		
Date: 12.12.2012	Time: from to 2 to 2:30 pm	Location: RGB 404
Meeting called by	Advisor	
Attendees	Professor Rouwaida Kanj-Professor Ayman Kayssi-Kim Baraka-Marc Ghoril-Sami Malek	
Minutes taker	Marc Ghobril	
Agenda Item: Prototype		
Discussion		
• We discussed the list of hardware that we will order. Professor Kayssi told us that we have a budget of 1000$ approximately so we should take this into consideration while finalizing the list • Professor Kanj informed us that there are tablets (sponsored by Intel) that we can borrow from Professor Hazem Hajj and use them in our project. The tablets will be available as of Friday 7^{th} December • Professor Kayssi indicated that we should send the finalized list of hardware to Mr. Khaled Joujou		
Conclusions		
• A list of the items to order should be ready by the end of the week • The list should be emailed to the advisors and to Mr. Khaled Joujou		
Action Items	**Person Responsible**	**Deadline**
• Finalize the list of hardware and send it to the advisors and Mr Khaled Joujou	All	By 7^{th} December
• Take an appointment with Mr Khaled Joujou to discuss with him what hardware can be provided by AUB (so that we don't order it)	Kim Baraka and Sami Malek	By next week
Agenda Item: Finalized design & Prototype		
Discussion		
• The advisor asked us to put our design on paper ie to draw a comprehensive Figure that summarize our design in a detailed way • Professor Kanj suggested that we can apply to the Stars of Sciences program if we are still confident in our design after the prototyping phase • We also discussed which features to include in the prototype		
Conclusions		
• We need to draw a detailed Figure that summarize our design • We have to start thinking about the features of our prototype		
Action Items	**Person Responsible**	**Deadline**

• Decide on the features to include in the prototype and make a preliminary design	Marc Ghobril	By next meeting
• Research the Android programming for our application	Sami Malek	By next meeting
• Research the GPS tracker that we can use in our Android application	Kim Baraka	By next meeting

Meeting #: 10

Date: 12.12.2012	Time: from to 2 to 2:30 pm	Location: Wing D

Meeting called by	Weekly meeting
Attendees	Professor Rouwaida Kanj, Sami Malek, Kim Baraka, Marc Ghoril
Minutes taker	Sami Malek

Agenda Item: Prototype build

Discussion

- We discussed the prototype that should be built by the end of this semester.
- We decided on the minimum requirements for the prototype:

  A functioning Android – Arduino – Xbee ~~ Xbee – sensor/control node system, where the user can remotely read data from a sample sensor and control an LED.

Conclusions

- We should test the newly received intel tablet device and run on it a sample code as a first step.
- We should design and run the Android-Arduino communication code on the Android tablet.
- We should design and run the Arduino-Network communication code on the Arduino chip.

Action Items	Person Responsible	Deadline
• test the newly received intel tablet device and run on it a sample code • write the A-A comm code on the A-tablet	Sami Malek	By Next week / week after
• write a sample code and the final code draft to be run on the Arduino chip.	Kim Baraka	By next week / week after

Agenda Item: Presentation and Final Report

Discussion

- We discussed the approaching deadlines such as the Presentation and Final Report submission.
- Since we were squeezed during the Progress Report, we decided to start working on the Final report as early as possible.

Conclusions

- We need to start a Presentation draft (for the Fall semester)
- Whe should also to start drafting the final Report

Action Items	Person Responsible	Deadline
• Draw the detailed schematic of the Final project design on the Computer	Sami Malek	By next meeting
• Update the Progress Report, and add to it the Testing Part	Marc and Kim	By next meeting
• Start drafting the presentation and Final Report	Marc Ghobril	By next meeting

14.3 Arduino Code **(adapted from David A. Mellis and Tom Igoe)**

```
#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(169, 254, 106, 10);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80);

void setup() {
 // Open serial communications and wait for port to open:
  Serial.begin(9600);
   while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }


  // start the Ethernet connection and the server:
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}


void loop() {
  // listen for incoming clients
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        // if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
        if (c == '\n' && currentLineIsBlank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connnection: close");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
```

```
          // add a meta refresh tag, so the browser pulls again every 5 seconds:
          client.println("<meta http-equiv=\"refresh\" content=\"5\">");
          // output the value of each analog input pin
          for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
            int sensorReading = analogRead(analogChannel);
            client.print("analog input ");
            client.print(analogChannel);
            client.print(" is ");
            client.print(sensorReading);
            client.println("<br />");
          }
          client.println("</html>");
          break;
        }
        if (c == '\n') {
          // you're starting a new line
          currentLineIsBlank = true;
        }
        else if (c != '\r') {
          // you've gotten a character on the current line
          currentLineIsBlank = false;
        }
      }
    }
    // give the web browser time to receive the data
    delay(1);
    // close the connection:
    client.stop();
    Serial.println("client disonnected");
  }
}
```

14.4 Android code sample (31 KB of code not shown)

```
package com.example.finale;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.Spinner;
import android.support.v4.app.NavUtils;

public class AddMemberActivity extends DisplayNodeActivity {

        int Node_id;
        String cstype;
        String template;
        EditText Dev_Name;
        EditText Pin_Num;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_add_member);
                Intent intent = getIntent();
                Node_id = intent.getIntExtra("Node_id", -1);
                // Show the Up button in the action bar.
                getActionBar().setDisplayHomeAsUpEnabled(true);

                Spinner spinner1=(Spinner) findViewById(R.id.cstype);
                spinner1.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
                        public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {
                     // Here you check the spinner values sum
                                cstype=parent.getItemAtPosition(pos).toString();
                  }
```

```
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });
    //create an arrayadapter usin the string array and a default spinner layout
    ArrayAdapter<CharSequence>
adapter=ArrayAdapter.createFromResource(this,R.array.cstypes,android.R.layout.simple_spinner_dropdown_item);
    //specify the layout to use
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    // Apply the adapter to the spinner
    spinner1.setAdapter(adapter);

    Spinner spinner2=(Spinner) findViewById(R.id.spinner1);
    spinner2.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {
          // Here you check the spinner values sum
                    template=parent.getItemAtPosition(pos).toString();
        }
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });
    //create an arrayadapter usin the string array and a default spinner layout

adapter=ArrayAdapter.createFromResource(this,R.array.Templates,android.R.layout.simple_spinner_dropd
own_item);
    //specify the layout to use
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    // Apply the adapter to the spinner
    spinner2.setAdapter(adapter);
    Dev_Name=(EditText)findViewById(R.id.DevName);
    Pin_Num=(EditText)findViewById(R.id.Pin);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.activity_add_member, menu);
    return true;
}

public void Add_Device(View view){
```

```
		if(Dev_Name.getText().length()>0 && Pin_Num.getText().length()>0){
			member Devi=new member(Dev_Name.getText().toString(),cstype,Pin_Num.getText().toString(),template);
			((Node)Nodes.get(Node_id)).members.add(Devi);
			Intent intent = new Intent(this, com.example.finale.DisplayNodeActivity.class);
			intent.putExtra("Node_id",Node_id);
			startActivity(intent);
			finish();
			return;
		}
	}
	public void Cancel(View view)
	{

		Intent intent = new Intent(this, com.example.finale.DisplayNodeActivity.class);
		intent.putExtra("Node_id",Node_id);
		startActivity(intent);
		finish();
		return;
	}
	@Override
	public boolean onOptionsItemSelected(MenuItem item) {
		switch (item.getItemId()) {
		case android.R.id.home:
			// This ID represents the Home or Up button. In the case of this
			// activity, the Up button is shown. Use NavUtils to allow users
			// to navigate up one level in the application structure. For
			// more details, see the Navigation pattern on Android Design:
			//
			// http://developer.android.com/design/patterns/navigation.html#up-vs-back
			//
			NavUtils.navigateUpFromSameTask(this);
			return true;
		}
		return super.onOptionsItemSelected(item);
	}
}
```