UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Tanel Lips

# Analysis and Comparison of Smart Homes

Bachelor's Thesis (6 ECTS)

Supervisor: Satish Narayana Srirama, PhD
Co-supervisor: Carlos David Paniagua Gomez, BSc

Author: ........................................................................... "....." .......... 2012

Supervisor: ...................................................................... "....." .......... 2012

Supervisor: ...................................................................... "....." .......... 2012

Allowed to defence

Professor: ........................................................................ "....." .......... 2012

Tartu 2012

# Acknowledgements

First of all, I would like to acknowledge Carlos David Paniagua Gomez for helping and guiding me throughout the thesis. Secondly, I thank Satish Narayana Srirama who helped me understand the main concept of a scientific thesis. Furthermore, I'm also grateful to Jürmo Mehine who provided comments and proofreading concerning my thesis. Last but not least, I thank Huber Flores for helping me with rephrasing.

# Acronyms

**HVAC** - Heating, ventilation and air conditioning
**RF** - Radio frequency
**Wi-Fi** - Wireless fidelity
**PDA** - Personal digital assistant
**VM** - Virtual machine
**IDE** - Integrated development environment
**PCB** - Printed circuit board
**DIY** - Do it yourself
**GUI** - Graphical user interface
**MAC** - Media access control
**BSS** - Basic service set
**WPAN** - Wireless personal area network
**HBS** - Home bus system
**EIA** - Electronic Industries Association
**CEBus** - Consumer Electronic Bus
**APDU** - Application Layer Protocol Data Unit
**LLC** - Logical Link Control sublayer
**NPDU** - Network Layer Protocol Data Unit
**HES** - Home Electronic Systems
**IEC** - International Electrotechnical Commission
**ISO** - International Organization of Standardization
**WWW** - World-Wide Web
**HAP** - The Home Automation Protocol
**PLC** - Power line communications
**JSON** - JavaScript Object Notation

# Contents

# 1

# Introduction

A "smart home" typically is a domestic environment that has been partially automated. Home automation includes centralized control for lighting, HVAC (heating, ventilation and air conditioning), appliance management, and others. Home automation aims to enhance the comfort, energy consumption efficiency and security (1) in domestic scenarios. Generally, houses are equipped with independent control panels to control all of the systems and appliances present in the house. Moreover, those control panels are often not related each other. The main purpose of a smart home is to centralize the control of all the devices into a single control unit which can be programmed to do specific tasks suitable for the owner and the home in question. The goal of a smart home is not only convenience but also to reduce the consumption of resources such as power, gas, etc. Due to the current pricing on energy, resource conservation has become a part of a persons day-to-day life. If a person has the possibility to control his home automation remotely he can reduce the consumption of energy and thus cutting down on expenses. Furthermore, environmental sustainability has gained relevance in the latest years. If a person is away from home there is no need for the air conditioner or ventilation to operate. The same principle applies to illuminations, heating and other appliances. Some smart homes systems pause the operation of applicances until they are needed again. Furthermore, there are several different technologies for implementing such smart homes. Some standards utilize complex communication protocols and control wiring, others rely on embedded signals in the existing power circuit of the house. A portion rely on radio frequency (RF) signals, and others become hybrids by combining several methods. All of the controlling tasks are done through a microprocessor, for example Arduino (2), which enables the communication and upon receiving some commands controls the different systems in the house. Finally, the commands to control the appliances in the house are sent by a central control unit such as a computer, remote control or smartphone (iOS, Android).

## 1.1 Motivation

Due to the huge number of smart home technologies available on the market it is quite difficult to select the most appropriate system that satisfies the requirements for a specific scenario. To the best of the author's knowledge there are not adequate comparisons between the different smart home approaches that provide a decision criteria in order to easily decide which system should be adopted or implemented. Nevertheless there are few magazine articles (3) and supplier reviews that often happen to be subjective. This work tries to spotlight weaknesses and strenghts for the most popular home automation systems on the market. Further, the study presents the implementation of a smart home using Arduino and Android open platforms and Wireless communications (Bluetooth and Wi-Fi).

## 1.2 Contributions

To address these problems this work analyzes and compares six smart home technologies. These technologies are: Insteon, Zigbee, X10, Android@Home, Z-Wave and DomoticHome. All these solutions are comparable with each other in terms of reliability, installation, communication mechanism, scalability and cost. In addition the work includes the implementation of a wireless-based home automation system. Initially, it was intended to implement Android@Home. However, this solutions is still not available on the market. Therefore, we decided to adopt a similar solution, DomoticHome. Domotichome is similar to Android@Home relying on the Arduino and Android technologies. The implementation consists of an Arduino (2) board complemented with an ethernet shield or a Bluetooth shield.

Arduino is an open-source electronics prototyping platform. For communicating with the Arduino the study considers a smartphone with Android operating system (4).

## 1.3 Outline

**Chapter 2**:
The second chapter surveys the home automation systems and a reviews the current state of the art.
**Chapter 3**:
Chapter 3 addresses the analysis and comparison of the six home automation technologies available for consumers at the moment.
**Chapter 4**:
Chapter 4 discusses the practical implementation of wireless-based smart homes.
**Chapter 5**:
Chapter 5 concludes the study and summarizes the findings.

# 1. INTRODUCTION

# 2

# State of the Art

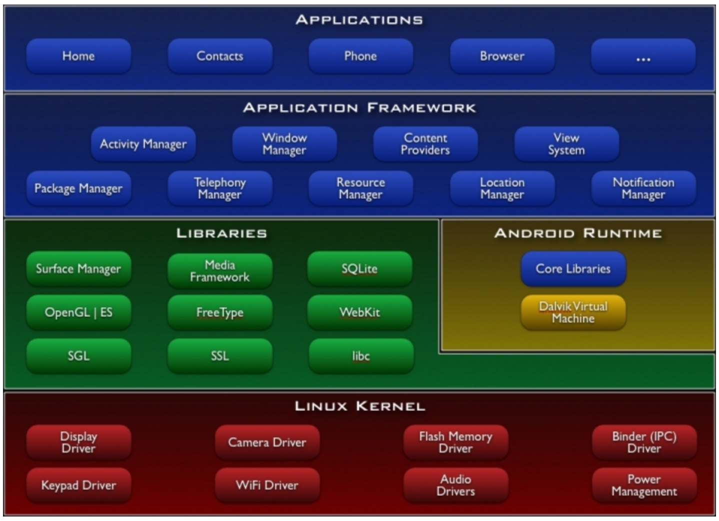
The chapter goes through Android, a mobile operating system, and Arduino, the open hardware powered by Arduino. Furthermore, the chapter reviews the wireless communication protocols such as Bluetooth and Wi-Fi. All the technologies discussed in this chapter are further used to support the implementation of a smart automation system. Finally, a survey of home automation technologies from their beginning until the latest developments in recent years.

## 2.1 Android

Android (5) is an operating system, powered by Google, for mobile devices such as tablets, smart phones and PDAs. It was initially developed by Android Incorporated but was later acquired by Google. Android is an open development platform, which means that handset makers can use the platform and customize it to fit their own requirements. By the time this thesis was developed it is estimated that there are more than 300 million activated Android devices (6) in the world and Android is installed on $\approx 56\%$ of devices worldwide. Concerning the architecture, Android comprises of five layers being: Linux kernel, Android runtime environment, Libraries, Application framework, and Application layer. The figure 2.1 illustrates the Android's architecture.

The most important part of the Android OS is the Linux kernel. The Linux kernel is used for memory management, process scheduling, networking and other services. Every manufacturer can modify the Linux kernel to suit the needs for

**Figure 2.1:** Android architecture (7).




their mobile device. The Linux layer is followed by the Android layer that contains the native libraries that are installed for a specific phone and vendor. Android runtime comprises the Dalvik virtual machine and the core Java libraries sits on the kernel. Dalvik VM is Google's interpretation of Java that is specifically designed for mobile devices. The Application Framework layer lies on top of these layers and is the layer used by developers to create applications. Finally, the applications and widgets are built on top of the previous layers. Each interface screen is exhibited by Activity classes. A single application may consist of one or more activities and a Linux process that contains the activities. An application can run in the background, even if its process has finished. It means that the activity's life cycle is not tied to the process life cycle. In short, Linux processes are just trivial containers for activities that can be disposed when they are not longer needed (5).

## 2.2 Arduino

Arduino is an open-source electronics prototyping platform. It is claimed that Arduino is based on flexible, easy-to-use hardware and software. Furthermore, it is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving data from a variety of sensors and can also affect its surroundings by controlling lights, motors, and other actuators (2).

The hardware comprises of a processor on board with input/output support. The Arduino board can be extended with a plethora of add-on modules known as "shields." Shields are boards that can be plugged on top of the Arduino PCB extending its capabilities (2). To program the microcontroller one needs to make a sketch. Sketches are software programs that are created on a computer using the Arduino IDE using C and C++ languages. The IDE is also used for transferring the code to the Arduino board. Arduino programs are composed by a structure, values (variables and constants) and functions (8).

Arduino has several advantages when compared to other microcontroller toolkits. For example, the software development is platform independent and the developer can use any platform (e.g Windows, Linux, Macintosh) for programming purposes. The open nature of Arduino has enabled the emergence of a massive community which provides a plethora of tutorials online; finally, it is relatively cheap compared to other prototyping platforms. Therefore, it has been pleasently welcomed in the DIY community. In addition, since it was initially developed as an educational prototyping platform it is relatively easy to learn (9).

## 2.3 Amarino

Amarino is a software toolkit developed by Bonifaz Kaufmann (9). It includes two main parts: an Android application and an Arduino library. Amarino enables the communication channels for sending and receiving data via Bluetooth between an Arduino-based device and an Android phone. It is done by an event-based communication standard. The Arduino provides a callback mechanism for incoming events, so a developer can register specific callbacks for each event. After that the microcontroller allows extracting data concerning to the event.

Amarinos Android application has a GUI to easily connect to several boards, create specific events and monitor the data stream. The application is also capable of using the phones sensors and sending info to the Arduino regarding the received information (9).

## 2.4 Wireless Communication Technologies

Wireless telecommunications enable the transfer of information between two or more points that are not physically connected. Distances can vary depending on the environment and technology used for the data transmission. This work focuses mainly on short-range technologies such as Bluetooth and Wi-Fi.

### 2.4.1 Wi-Fi

Wireless fidelity or Wi-Fi is based on IEEE 802.11a/b/g/n (10) standards for wireless local area networks. It allows wireless connection to the internet at broadband speeds. The architecture is comprised of several components that interact with each other to provide a wireless local area network (10) such as the physical layer and data link layer. Moreover, MAC procedures are defined for accessing the physical layer. In a Wi-Fi based network each cell is a basic service set(BSS). BSS comprises of a set of Wi-Fi stations and clients connect to such stations. Wi-Fi therefore defines a distributed coordination function (11) among the peers.

### 2.4.2 Bluetooth

Bluetooth is based on a wireless radio system that is designed to remove the need for cables for short-range devices, such as mice, printers etc. A network that includes such devices is called a wireless personal area network. Topologies that are defined in Bluetooth are named Piconet and Scatternet. A Piconet is a WPAN that consists of two or more devices. One of them serves as a master and the other are slaves. All of the devices in a Piconet are synchronized with each other using the clock of the master. Slaves communicate only to the master. Master can communicate with any device. Further, Scatternet comprises of different Piconets that overlap time and space. Two or more Piconets can be connected with each other to form a Scatternet. A Bluetooth device can be a part of different Piconets at the same time. This allows the data to flow beyond the range of a single Piconet. A device can only be a master in one Piconet, but a slave in many Piconets (10).

## 2.5 The History of Home Automation

Even though home automation has been available to the average consumer for nearly 35 years, the concept of home automation developed has been extensively explored in the 20. and 21. century. The table 2.1 summarizes the developments.

In 1975 a company called Pico Electronics developed and patented the X10 Power line carrier technology. The company had already tried nine different approaches with no success, but while developing the system for tenth time they finally manage to succeed, so they decided to call the technology X10. The idea behind X10 was to transmit a 120khz signal on the electrical power line. Every signal was specifically coded with a House and Unit code. Although such technologies had been developed for the best part of 50 years (30) none of them was implemented in any similar fashion. After they patented their work it took just a few years for introducing their first product into the market. So, in 1978 they released the X10 protocol to the market (12). Due to the fact, that data transmission was done reusing existing power lines, it was relatively cheap because no additional wiring was needed. The technology is analyzed in detail in the chapter 3.

The 80s were the springboard of home automation. In 1983 Murata, Namekawa and Hamabe proposed a standardization plan for home automation systems because there was no compatibility among the different manufacturers that were proponents of home automation systems in Japan. The research group was called HBS study group. In 1984 after two years of work seven manufacturers reached an agreement on standardization. They proposed a home bus system(HBS) that consisted of three bands: Base-band, for controlling signals; Sub-band, for high-speed data signals; and FM/TV-band, for visual information.

In comparison to the X10 protocol the HBS used a coaxial cable that, according to HBS study group, had a lower installation cost than other protocols. In such case, if a house already has a coaxial wired installation, it is possible to incorporate HBS to the existing system with few modifications. Their plan was to install the interface LSIs for the HBS into electronic devices to be shipped in the following decade (13).

**Table 2.1:** Contributions

| Year | Author | Contribution |
|---|---|---|
| 1975 | Pico Electronics (12) | Invention and patenting X10 power line carrier technology |
| 1978 | Pico Electronics (12) | First x10 products introduced to the public. |
| 1983 | Masashi Murata et. al. (13) | Japanese proposal of a standardized protocol called HBS |
| 1983 | Christos Douligeris et.al. (14) | American proposal of a standardized protocol called CEBUS |
| 1984 | David MacFadyen (15) | Introduction of the Smart house concept |
| 1985 | Masahiro Inoue et al. (16) | First system developed based on HBS |
| 1986 | Ryuji Hamabe et al. (17) | Japanese renewed the proposal for HBS |
| 1988 | Christos Douligeris et al. (14) | Publication of CEBus working draft standard |
| 1989 | IEC and ISO (18) | Organization of a joint committee called HES |
| 1992 | Christos Douligeris et al. (18) | Completion and release of CEBus |
| 1993 | Smart House Inc. (19) | Marketing of the Smart house |
| 1996 | Peter M. Corcoran et al. (20) | Demonstrated CEBus access via WWW |
| 2000 | Renato J. C. Nunes et al. | Proposed an internet application for smart home controlling |
| 2002 | N. Sriskanthan et al. (21) | Proposal for a Bluetooth based home automation system |
| 2002 | Europeans (22) | Decision to make one standard protocol |
| 2004 | A. Alheraish (23) | Proposal of a system based on M2M through GSM |
| 2005 | A. R. Al-Ali et al. (24) | Proposal of a Java-Based HAS |
| 2005 | SmartLabs (25) | Release of Insteon |
| 2005 | ZigBee Alliance (26) | Release of ZigBee |
| 2005 | ZenSys (27) | Release of Z-Wave |
| 2006 | Arduino | Development of Arduino |
| 2009 | Mattia Lipreri (28) | Introduction of DomoticHome |
| 2011 | Google (29) | Introduction of Android@home |

As a result of the work in 1984 by Murata et.al a home automation system was developed based on the HBS in 1985 by Inoue, Uemura, Minagawa, Esaki and Honda. It consists of four subsystems.

1. Room Monitor Control subsystem
    - System for controlling home functions and security
2. Telephone subsystem
    - Telephone and security alarm
3. Telecontrol subsystem
    - Controlling of devices and security sensors via telephones that are not located in the house
4. In-house Video Control subsystem
    - For receiving video signals from picture phone sentry (16).

After 3 years of successful experiments and two built model houses, Murata revised and updated the proposition made in 1983. The new proposed HBS had two bands for transmitting information within the house instead of 3 bands that were proposed 3 years earlier. The base band remained the same but in lieu using sub-band and FM/TV-band, so they merge the two bands into a broad-band (17).

Parallel to the Japanese, manufacturers in the United States also recognized the problem of standardization. A lot of new domestic electrical equipment was being offered to the consumers, but none of these had a way of integrating every device into a central network. The only way for a consumer to integrate every single device was to acquire products from one manufacturer only. In 1983 the Electronic Industries Association organized a committee to develop a standard that addressed the problem of standardization (14).

David MacFadyen was the founder and chief executive officer of Upper Marlboro, Md., consortium. In 1984 he introduced the Smart House concept. While other companies on the market offered home automation systems that people could install into their homes, David MacFadyen proposes a new approach. The consortium started developing houses with home automation systems already preinstalled and by the year 1987 they planned to have 5000 houses ready to be sold. The houses would be programmable and every device would interact with each another via computer networks (15).

Meanwhile, also in early 80's, the Electronic Industries Association(EIA) (31) had been working on Consumer Electronic Bus (CEBus). After five years of extensive research and development they published the working draft standard. According to the EIA, CEBus was designed to allow manufacturers to produce electronic devices that can interact with other manufacturers' products over different media channels. A CEBus' features was the allowing naming devices. For example, a user could name his living room television named as "Living room TV". This removed the necessity of knowing a numeric code for each device (31).

CEBus had a four layer network model.

1. Application layer
    - Device requests the generation of an Application Layer Protocol Data Unit(APDU) and this is handed down to the network layer.
2. Network layer
    - Receives the APDU and makes it into a Network Layer Protocol Data Unit(NPDU). Then it sends it to Logical Link Control sublayer(LLC) which is a part of the Data Link layer.
3. Data Link layer
    - Logical Link Control sublayer appends a header with additional service information to the NPDU to make a LPDU and invokes the services of Medium Access Control(MAC)
    - Medium Access Control Here the LPDU is made into a MDPU and sent to the physical layer
4. Physical layer
    - For receiving video signals from picture phone sentry (14).

After the introduction of the CEBus standard research continued for the EIA.

Although Smart House planned 5000 houses to be ready by 1987, their plan failed due to a number of obstacles. For instance, when Smart House was introduced the company wanted to make one large cable containing telephone, TV, power and communications wiring. But the wire came out so thick that it was not feasible to run it between the walls. And by the end of 1987 founder and CEO David MacFadyen left the company because of persistent reoccurring problems with wiring. Further, Gerald Engel expressed "*The company underestimated the difficulty of getting the group of disparate industries involved with Smart House to agree on anything*". Lastly, the project engineers focused on adding new features to the Smart House instead of release its first products to the market (15).

In late 80's there were four key players in the home automation standard industry: HBS, Smart House, Esprit Home Systems and CEBus. The Japanese and Europeans groups wanted to make their standard a world-wide standard (18). So in 1988, a committee named Home Electronic Systems(HES) was created. It comprised of International Electrotechnical Commission(IEC) and International Organization of Standardization(ISO). Their job was to specify home automation standards. By the end of 1989 the U.S. team required to HES to consider their home automation system. Later, in 1992 there were already seven home control systems included by HES:

1. BatiBUS (France)
2. CEBus (USA)
3. D2B (Holland)
4. EIB (Germany)
5. ESPRIT Home Systems (Common market)
6. HBS (Japan)
7. CEBus (USA) ...

Also in 1992 was the completion and release of CEBus Standard IS-60 (18). 1993 was a critical year for SMART HOUSE automation. They released the first product to the market. Further, first SMART HOUSE model opened in Naperville, USA (19).

Meanwhile, the World-Wide Web(WWW) was developed by scientists in the CERN laboratory in 1990. It became the technology of choice for internet applications and in 1996 Peter M. Corcoran, Joe Desbonnet and Karl Lusted proposed a way to remotely access CEBus networks via the HTTP protocol. They implemented a fault diagnosis system. The hardware, part of the implementation, consisted of a CEBus network, an active CEBus node and a PC-based server. The software included a modified CEBus protocol, WWW server modification, server-end CGI and client-end Java interfaces. Accordingly to Peter M. Corcoran et. al. the system can be thought as "*The "active" CEBus-node contains a dedicated microcontroller with modified protocol software to facilitate test and monitoring activities. The Medium Access Control (MAC) sublayer of the Datalink layer has been reprogrammed to allow the system to receive and record all network packets. These captured packets are passed to the Application layer along with channel timings in "unit system times" (USTs). The HTTP server software runs on a standard PC which is also interfaced with the CEBus network by means of an "active" CEBus node, as described above. A "daemon" program monitors CEBus activity. A range of services are available from this "daemon". Basic services including the logging of network traffic to a server-end database and transmitting selected network traffic to a remote WWW client. The software can also accept sequences of network packets from a remote WWW client for transmission on the CEBus network via the "active" node.*" (20). It is considered to be the pioneer of online remote controlling of not only CEBus protocol but also for all home automation systems.

By 1997 there were eight national home automation standards and one international home automation systems. These included:

1. X-10(U.S)

2. CEBus(U.S)

3. LonTalk(U.S)
4. SMART HOUSE(U.S)
5. BatiBUS(Europe)
6. EIB(Europe)
7. European Home systems(Europe)
8. HBS(Japan)
9. HES(international) ...

Manufacturers had a huge number of different standards to choose. Such big number of standards delayed the development of home automation products. By the year of 2002 the U.S had eight standards, 12 open protocols and 10 proprietary protocols. This diversity only complicated to the manufacturers to develop commercial products. Fortunately the Europeans efforts realized that the vast number of protocols slows down development of home automation technology and decided to combine the three rivalling protocols into one called Konnex. In contrast, the U.S approach did did the opposite pushing on the development of multiple protocols. That was a negative sign for a lot of the technologies. The problem was foreseen by Kenneth Wacks: "*The market for home systems is not large enough support so many protocols. Therefore, many will not last.*" (22).

Although people knew that there were too many home automation protocols and technologies on the market, it did not stop them from inventing and proposing new systems in the 2000s. One of such new solutions was proposed by Sriskanthan, Tan and Karande. It was a home automation system that relied on Bluetooth. They chose Bluetooth since it covered, in their opinion, all the basics needed for automating a home. For example, it operated over the unlicensed and available frequency of 2.4 GHz and it could link devices within a range of 10m. The system developed consisted of a Host Controller that was implemented on a Personal Computer and a microcontroller that was able to communicate with the host through Bluetooth. They named it The Home Automation Protocol (HAP) and realised great future for their proposed solution. Sriskanthan et. al says " *With our* *home system, which consists of the HC that usually takes a form of PC, Internet connectivity can easily be established and control be made available. Efforts in such direction will help realize a truly wireless, fully automated home automation system* (21)".

All types of different home automation solutions were presented in the coming years. For example:

1. 2004 A. Alheraish proposed an M2M (machine-to-machine, man-to-machine or mobile-to-machine) solution based on GSM cellular communication network (23),

2. 2005 A. R. Al-Ali and M. AL-Rousan presented a home automation system based on Java (24).

2005 there was a huge expansion in home automation systems. Most of the systems that are widely popular today were first introduced that year. First, a home automation system named Insteon was developed by a company called SmartLabs. The main characteristic of Insteon was that it had a mesh topology which composed of RF and PLC. In short, one can use only RF, only PLC or both at the same time.

Secondly, a wireless networking technology called ZigBee was introduced by the ZigBee Alliance. It was developed for small datarate and short-range applications. One of the distinctive features is that a ZigBee based network can easily scale without the requirement of powerful transmitters. And lastly there came the Z-Wave. It is a wireless protocol by ZenSys. The manufacturer states that "*The Z-Wave protocol is a low bandwidth half duplex protocol designed for reliable wireless communication in a low cost control network. The protocols main purpose is to communicate short control messages in a reliable manner from a control unit to one or more nodes in the network* (27). These three last approaches are considered for the analysis in chapter 3.

Different technologies are becoming more and more cheaper and accessible to everyone. In 2006 an open-source single-board microcontroller was introduced by a group of Italian students who named it Arduino. The motiviation for creating Arduino was to develop a device that was less expensive than other prototyping systems and easily extendable. It was not meant exclusively for home automation but a lot of people saw a way of integrating the board into their home automation systems. A work that relies in the utilization of the Arduino boards for home automation is DomoticHome introduced in 2009. The author Mattia Lipreri wanted to develop a simple and cost efficient way to automate the lights and the garage door in his house (28). The specifics of the project are discussed in the following chapter. Similarly, in 2011 Google announced of developing their own home automation system that lets Android applications discover, connect, and communicate with electrical appliances and devices in the home. Android@Home will use a mesh networking protocol that functions in the 900MHz frequency bands just like Z-Wave (29). The wireless protocol used for the Android@Home demo at Googles' I/O Developers Conference was based on SNAP from Synapse Wireless (32).

Home automation has become more and more popular the recent years. As a result of the extensive research carried on in the home automation domain, there are approximately 15000 articles and/or patents published since 2001 and approximately 5000 have been published in the last 2 years.

# 3

# Analysis and Review of Home Automation Systems

This chapter describes the current smart home solutions on the market and discusses the differences between different systems in terms of scalability, cost, reliability, communications and ease of installation.

## 3.1 Insteon

Insteon is a home automation technology developed by SmartLabs. A distinctive features of Insteon is that it uses both radio frequency(RF) and already existing power lines(PLC). Insteon is one of the few home automation systems that works in a dual-mesh network. It is possible to use only RF or PLC but it is also possible to use them both at the same time. The systems RF band operates at the 904 MHz frequency. The data rate is for instantaneous 13,165 bits/sec and for sustained 2,880 bits/sec (25) (34).

All Insteon devices act like repeaters. This means that they can play the role of sender, relayer or receiver. If the devices that are trying to communicate are not within the coverage area transmit messages using a multi-hop strategy. All messages are received by the Insteon devices in the network, and if the message is not intended for the device that receives the message it forwards the message to other Insteon devices. The maximum number of hops for a message is three so avoiding to flood the network. Other devices act the same way until it is received

**Figure 3.1:** Insteon (33).


by the designated device. Every message consist of two 2-bit fields. One bit contains the maximum number of hops and the other number of hops left. If a device receives a message with zero hops left, the message is not sent again. The range of a Insteon signal is approximately 45 meters reduced significantly by the presence of walls and other physical blockers. Due to the fact that all Insteon devices use two-way communication, when a message arrives to the intended recipient, it sends out a message of a successful transmission. If the original sender of the message does not receive such message it automatically resends the message up to five times and increases the Max Hops until the maximum limit (three hops).

Insteon devices are relatively easy to set up. They automatically connect to an Insteon network as soon as they are powered up. One can connect up to 1024 devices to a single network but is also possible to link Insteon networks so the number of devices multiply by the number of networks. It was designed in such a way that it is capable of transmitting X10 messages over the power line. So if a person already has X10 set up at his home he does not have to discard the previously acquired X10 devices since they are compatible and can be incorporated into the Insteon network (25).

The price for a starter kit starts from 72 euros (depending on the distributor) and the modules from 14 euros (35).

## 3.2 X10

**Figure 3.2:** X10 (36).


The X10 Power Line Carrier was designed and proposed in the late 1970s by a company named PICO electronics. It was designed to use the existing power lines of a home and in the 1990s it was extended to use RF as well. The X10 network consists of several types of devices:

1. Transmitters - Control devices that transmit the original message- plug-in, phone, PC controllers etc.

2. Receivers - Devices that receive messages and act accordingly - such as light switches, doors etc.

3. Transceivers - Devices that receive RF messages and re-transmit it on the power line (12).

The messages consist of four bits. A bit value of zero is communicated if there is no 120 kHz pulse and a value of one is represented by a millisecond pulse of 120 kHz. X10 does not have error detection and not all X10 devices have two-way communication capabilities. The original X10 devices were only one-way communication. So if a receiver does not receive a message the original transmitter did not have any way to know if the message was successfully delivered to the recipient. In 1997 more advanced bu expensive devices were developed with two-way communication capabilities. The technology incorporates a slight error prevention technique which instead of sending a message once, sends out the message twice to double the chances of successfully transmitting the message. The RF band frequency on which X10 operates is 310MHz or 433MHz and the range is around 30 meters depending on the environment. Due to the fact that data rate is approximately 20 bit/sec, the transmission of X10 data is relatively slow. This confines the technology to switching the devices on and off or other simple actions (37). Finally, the maximum number of devices supported is 256.

Price for a starter kit is from approximately 44 euros and the cost of an individual module is 4 euros (38).

## 3.3 Z-Wave

The Z-Wave technology was developed by a company named Zensys. Z-wave consists of four layers and and RF media that is controlled by the MAC layer.

1. Application layer - Controls the decoding and execution of commands within a Z-Wave network
2. Routing Layer - Controls the routing of packets within a Z-Wave network
3. Transfer Layer - Controls the transfer of data between devices - this includes retransmission, acknowledgements and checksum check
4. Mac Layer - Controls the usage of the radio frequency medium.

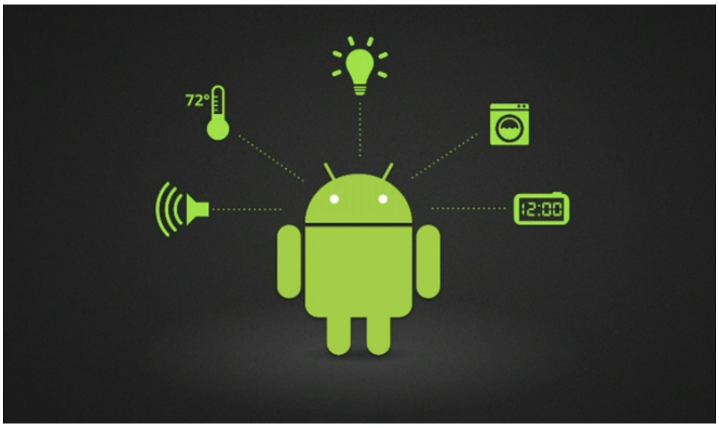
**Figure 3.3:** Z-Wave (39).


Z-wave uses the RF communication type. It works on the 868 MHZ, 908 MHZ or 2400 MHZ frequency band. The range of RF signals is 30-100 meters and the data rate is 20 kbit/s. Z-wave has two types of devices - they are Controlling devices and Slave nodes. Controlling devices initiate the communication by sending commands to other nodes; further slave nodes forward messages to other nodes or if they are the intended recipients reply on and execute the commands received. Controlling devices have the full routing table of the Z-Wave network and is able to communicate with all the devices in the network. Slave nodes cannot independently send direct messages to other nodes unless they are ordered by the controlling devices. If a slave node receives a command it executes it and after it sends a reply to the controlling device notifying about the successful command execution. If the controlling device does not receive an acknowledge message, the frame is retransmitted with a random delay to avoid a potential collision. Maximum number of devices supported is 232 (27).

The prices of starter kits start from 56 euros and the cost of a single module from 16 euros (40).

## 3.4 Android@Home

**Figure 3.4:** Android@home (41)




Android@Home was announced by Google in May 2011. The system is announced to work with a mesh network in the 900MHz frequency bands. Google chose 900MHz because it is least likely to be crowded than the wifi 2400 spectrum. It is assumed that their protocol, announced in the Google's I/O Developers Conference, was based on SNAP from Synapse Wireless (32).

It is still a closed protocol. Initially, Google announced that they will create an Android bulb with Lightning Science. According to Ted Russ, chief business development officer for Lightning Science, the bulbs will be using the 6LowPAN standard. Consequently, Android@Home protocol is likely to be based on 6LowPAN technology. Geoff Mulligan describes "*6LoWPAN is a protocol definition to enable IPv6 packets to be carried on top of low power wireless networks, specifically IEEE 802.15.4. The concept was born from the idea that the Internet Protocol could and should be applied to even the smallest of devices* (42)." 6LoWPAN works in the 915mHZ frequency band and has a range of 10-100 meters. A distinct feature of 6LowPAN is the number of hops which is up to 255, so it is practically certain that a message will reach the intended node (34).

Furthermore, with the release of the Arduino boards Google lets people try and build peripheral devices and accessories which are compatible with their Android@home system. Consequently, both open platforms, Android and Arduino, join together to support and extend Google's Android@home environment automation approach. Moreover, the open nature of Android and Arduino resolve some issues such as licenses and fees. In addition, since there are already more than 230 million Android devices activated, it is very promising to develop devices with pre-existing Android connectivity (6).

## 3.5 ZigBee

**Figure 3.5:** ZigBee (43).

ZigBee is a wireless technology developed by the ZigBee Alliance. Its architecture is composed by four main layers:

1. Physical layer - responsible for sending and receiving commands and data.

2. Medium access control layer - responsible for networking.

3. Network layer - Controls the correct usage of the medium access control layer.

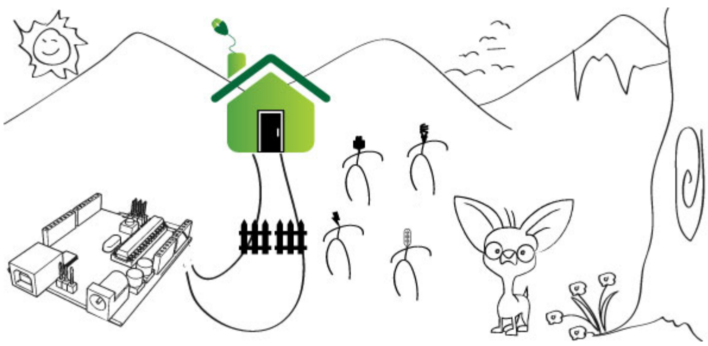
4. Application Layer - consists of APS sub-layer and ZDO:
   - APS sub-layer - provides services such as discovery and binding.
   - ZDO - defines the roles of devices, initiates and responds to binding requests, and handles security aspects.

Zigbee uses RF communication type. The frequency bands in which ZigBee works are 868MHz, 915MHz and 2400MHz. The range varies from 10 to 100 meters and the transmission data rate is 250kbit/s. There are two types of devices in a ZigBee network being Full and Reduce function. A Full function device usually acts as networks coordinator. A ZigBee network requires at least one network coordinator. A network coordinator keeps a network tree of the other devices that can be contacted. In addition, the network coordinator is the center node. The system uses hand-shaked protocol. In other words if an end-device gets a command it responds to the coordinator that it has received and executed (26). ZigBee is capable of connecting more than 64000 devices. It is possible due to the fact that ZigBee networks are extendable with each other so in theory the number of devices can be infinite.

Starter kits start from 150 euros and modules from 16 euros.

## 3.6 DomoticHome

**Figure 3.6:** DomoticHome (44).



DomoticHome is an open-source project originaly developed by Mattia Lipreri. Lipreri's main goal was to improve the comfort of a domestic environment. The home automation systems that are currently available in Italy happened to be expensive or hard to install and run. So Lipreri decided to develop his own system with two goals, that the system should be low cost and accessible to everybody. The system relies on Arduino boards extended with an Ethernet Module to connect it to the home local area network. DomoticHome.net is a website where one can generate Arduino code for communicating with different devices and the Android phone application. The DomoticHome system works under the Wi-Fi frequency band, 2401MHz to 2495MHz. The range depends on the capabilities of the router and the Android device but it is approximately 20 meters.

There is no error detection on this system. If Arduino gets the command from the phone it replies that the command has been executed. However, the Arduino simply acts as a intermediary between the appliances and the mobile devices. The system supports up to 14 devices since the Arduino board has only 14 pins that you can connect your devices to, it does not explore any multiplex strategy for increasing the number of devices that can be manage by the Arduino board.

The Arduino board costs around 50 euros and the necessary ethernet module costs 60 euros.

## 3.7 Analysis

### 3.7.1 Reliability

Reliability is one of the key factors when one is considering of buying a new device, vehicle, household item etc. The same applies for home automation systems. If there is no error detection or correction in a system, there is no way to find out if a command got through to the device it was intended unless you can physically see or touch the machine. Specially in home automation systems the reliability in handling some devices is crucial. For instance, a smoke detector managed by the home automation system required high levels of reliability from the device and from the system controller.

**Table 3.1:** Zigbee, Z-Wave and Insteon

| | ZigBee | Z-Wave | Insteon |
|---|---|---|---|
| Communication type | RF | RF | PLC/RF |
| RF band (MHz) | 868/915/2400 | 868/908/2400 | 904 |
| RF range (m) | 10-100 | 30-100 | 45 |
| Data rate | 250 (kbit/s) | 20 (kbit/s) | 3/13 (kbit/s) |
| Kit | 199-299 | 74.99 - 829.99 | 94.99-159.99 |
| Modules | 21.50-37.50 | 20.99-89.95 | 17.99-49.99 |
| Error detection | Yes | Yes | Yes |
| Retransmission | Yes | Yes | Yes |
| One/two-way communication | Two | Two | Two |
| Message routing | Yes | Yes | Yes |
| No. Devices possible | 64000+ | 232 | 1024 |
| Same networks | Yes | Yes | Yes |
| Other networks | No | No | Yes |

**Table 3.2:** X10, Android@home and DomoticHome

| | X10 | Android@home | DomoticHome |
|---|---|---|---|
| Communication type | PLC/RF | RF | RF |
| RF band (MHz) | 310 | 868/915/2400 | 2400 |
| RF range (m) | 30 | 10-100 | 20 |
| Data rate | 20 (bit/s) | 20/40/250 (kbit/s) | 9,4 (kbit/s) |
| Kit | 57.99-130.99 | NaN | 57.6 |
| Modules | 4.99- 30.99. | NaN | 60 |
| Error detection | No | NaN | No |
| Retransmission | No | NaN | No |
| One/two-way communication | One | Two | One |
| Message routing | No | Yes | No |
| No. Devices possible | 256 | NaN | 14 |
| Same networks | No | NaN | No |
| Other networks | No | NaN | No |

In terms of error detection and redundancy all of the discussed systems have strategies and mechanism to guarantee that the messages are delivered and executed correctly. Insteon, Zigbee, Z-Wave and Android@home (assumed to be based on 6LowPAN) employ checksums. The difference is that Insteon and Z-Wave use 8-bit checksums, but Android@home and Zigbee use the IEEE 802.15.4 defined 16-bit checksum. If a message sent by a controller does not reach the intended device or the command is not initiated, then the end-device does not send a successful message back. If the controller does not get the acknowledge in a specified time window it deploys a retransmission thus the network can be considered as reliable. In contrast, DomoticHome and X10 have only one-way communication capabilities so they do not have error detection or retransmission strategies which makes them unreliable.

### 3.7.2 Ease of Installation

Another important aspect when selecting a home automation system is ease of installation. The question is if can a home owner install the system by himself without requiring extensive prior knowledge or hiring technicians to support the installation.

In this matter, Insteon claims to provide easy installation. When an Insteon device is powered up for the first time it automatically connect to the Insteon network. Similarly, Android@Home provides a plug-and-play installation approach, but it requires initial pairing of the Android control device with the appliance. Other systems are not that easy to install. In contrast, since Z-Ware and Zigbee are protocols rather than products the market offers a wide number of solutions based on Z-Ware and Zigbee. Therefore, the ease on installation of such systems is provider dependent. In addition there is no automatic connection to the networks with the device. Every device has to be manually added to the network. Further, the X10 requires more effort than its competitors in terms of ease of installation. It needs additional know-how of electrical wiring since it has to be connected to the existing home wiring systems (if any otherwise the house required to be wired). All of the previously mentioned solutions are commercial. In contrast, DomoticHome is not a commercial product, which means that it has not been developed for end-customers. It is mainly used by developers or people with technical skills. Consequently, installing such a system is more difficult than others.

### 3.7.3 Communication

In a modern home there are a lot of devices that use RF communication technologies. The more there are devices the more interference in the environment. For example a lot of homes have Wi-Fi that works on 2.4GHz frequency. A residential building might have 10 or more different Wi-Fi networks colliding and interfering with each other. Some home automation systems employ multi-channel approach which are capable of using different frequencies to reduce the probability of interference (e.g. ZigBee, Z-Wave and Android@Home). ZigBees main advantage is that the coordinator in the ZigBee network is capable of automatically changing the working frequency if any node in the network detects major interference. The frequency range of the other technologies can be changed manually depending on the users needs. Insteon, DomoticHome and X10 use single-channel approach which means that they have only one frequency. Insteon works in the 900Mhz frequency which is less crowded than the DomoticHome Wi-Fi frequency that can be a problem. But a lot of manufacturers are designing devices that use the sub-gigahertz frequency which in turn can also start overcrowding the frequency range.

Physical range is equally important. But the signal strength is very dependant on the environment. If there are walls made of concrete all around the receiver/transmitter of a system, then the range is reduced significantly. Most of the systems compared can use nodes in the network to extend the range of the message. X10 and DomoticHome do not have such capabilities.

### 3.7.4 Scalability

Scalability is the ability to extend the network by adding extra networks or nodes. This may include using networks that are based on the same technology or networks that are using some other technology.

The only technology that can utilize device from another technology is Insteon. It is capable of using original X10 devices because it shares the message protocol of the system. Insteon is capable of having 1024 nodes in its network which is a bigger number when comparing to X10 or Z-Wave that can address up to 256 and 232 devices in a network respectively. The weaker of them happen to be DomoticHome. It can utilize 14 of its pins and this is also the maximum number of devices it can incorporate. Although, this is enough in the case of a medium-sized apartment. However, as mentioned before no multiplexing strategies are considered. If the 14 bits are use by the board in multiplex mode it would be possible to manage 196 devices depending of the multiplex protocol used. The most extendable is ZigBee which allows 64000 devices and it is extendable to theoretically limitless number of devices since one can add infinite amount of ZigBee networks to an existing network.

### 3.7.5 Cost

The cost of a home automation system is also important to customers. Every starter kit includes the basic setup hardware and software one needs to automate lighting devices.

As we can see from the table the cheaper the system the harder it is to install. DomoticHome does not have a starter kit. For the installation you need an Arduino board complemented with an ethernet module which is approximately 120 euros. Some other resources one may need (including wiring, relays etc.) are rather cheap. Further, the prices for Android@Home are not available yet and Insteon starter kits are sold from 75 euros and the modules from 17.99 euros, but at the time the technology is not compatible with european voltages and sockets. Manufacturer of Insteon have announced that they will start distributing EU-compatible devices in the summer of 2012. On the other hand, X10 is the cheapest of them all with prices for the starter kit from 57.99 euros and modules from 4.99 euros. This is due to the fact that the technology has been around for nearly 35 years and it is outdated. However, the installation happen be difficult to install so the hiring of technicians is required - this turns in extra costs for support and consulting. The same applies for ZigBee and Z-Wave. A starter kit is nearly 200 euros and the modules are from 21.50 euros. This is a major drawback since it already brings some difficulties to install and the cost is higher when compared to its competitors.

# 4

# The Implementation of Wireless-based Home Automation Systems

This chapter addresses the implementation of a home automation systems which rely on two different wireless communication technologies. The home automation systems presented rely on Arduino and Android open platforms.

As mentioned earlier the fist idea was to implement Android@Home but since it is not available on the market yet we considered DomoticHome. Arduino was considered because it is open hardware and compared to other prototyping platforms it is the cheapest. Furthermore, it has a strong and active community and it is estimated that there are more than 300000 Arduino devices sold since 2006. Arduino also has a cross-platform open IDE. Other home automation systems are relatively expensive, so the systems tries to be low cost by utilizing Arduino as hardware platform. Furthermore, Android was considered because of its popularity among smartphones nowadays. Google estimated that there are over 300 million activated Android devices and 850 thousand are activated daily. In addition, Arduino is highly Android compatible.

## 4.1 Scenario

The scenario describes a person who has installed a home automation system at his/her home. Every light in the apartment is connected to Arduino-based hardware. If he leaves a room he can easily turn on/off the lights in the room he is leaving via his smart phone. He starts an Android application that is connected to the Arduino board using either Bluetooth or Wi-fi. Later, he presses on the button named 'Bedroom' and presses 'ON' button on the screen sending the "Light On" command to the lights installed in the room labelled as "Bedroom". Later, the command is received by the Arduino board which execute the command turning on the lights. Further, he chooses "Living room" from the phones screen and presses 'OFF' to turn off the lights and the command is executed in similar fashion. When heading to the room he might want to regulate the temperature of the home by pressing 'Temperature' on his phone. The telephone displays the current temperature which it gets from a thermistor module connected to the Arduino. In similar fashion several devices can be controlled from the handset.

## 4.2 Architecture

### 4.2.1 Bluetooth

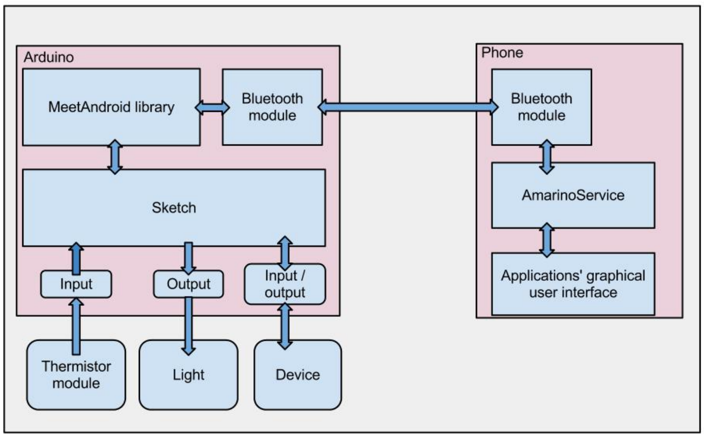
For the implementation the following items were needed

- Arduino BT - Arduino microcontroller with Bluetooth-connectivity
- Android phone with Bluetooth-connectivity - Used for communicating with Arduino via DomoticHome
- LED module
- Thermistor module

Amarino toolkit was chosen for communication.

On the phone side, Amarino library has been considered. Amarino is an Android application for monitoring Bluetooth connections. It runs in the background

**Figure 4.1:** Arduino and Android communication architecture




as a service named AmarinoService. The main responsibilities of the AmarinoService is to maintain and handle communication between the phone and Arduino. At the same time it processes command coming from the GUI, plug-ins or other external third-party applications that use the service. The graphical user interface communicates with the service by sending Android intents.

Arduino has a Bluetooth module attached to it that is used to receive and send data to the phone via the Bluetooth protocol. A sketch that runs on the Arduino board uses the MeetAndroid library to manage incoming and outgoing messages. The library uses a serial port of the Arduino to send and receive data. The messages contain identifiers and content. Identifiers are symbols from the alphabet. The content can be different things, such as strings, integers, booleans etc.

#### 4.2.1.1 Implementation

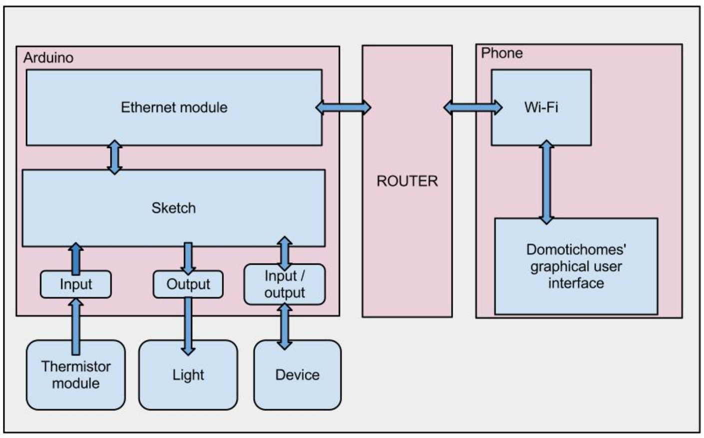
First, the thermistor module is connected to the analog pin 5 and the LED module to digital pin 8. Later, the Arduino sketch has to be created to programm the microprocessor. The communication protocol sets the identifier "T" as the command to retrieve the temperature readings and under identifier "L" the control of the LED. After powering the Arduino up and uploading the code with Arduino IDE, the Arduino was ready for receiving commands and connections. From the smart phone once the application has been installed it first needs to peer with the Arduino BT. Later, the Arduino Board and the phone can exchange messages each other. After the connection happened the user is able able to check temperature readings and turn the LED on and off.

### 4.2.2 Wi-Fi

For the implementation we needed the following items:

- Arduino MEGA ADK - Microcontroller for implementing Wi-Fi-based HAS
- Ethernet module - For connecting with the router in Wi-Fi-based HAS
- Cat 5 cable - for connecting Arduino with router
- Android phone with Wi-Fi connectivity - Used for communicating with Arduino via DomoticHome
- Wi-Fi-capable router
- LED module
- Thermistor module

**Figure 4.2:** Arduino and Android communication architecture



On the phone side, the Android application DomoticHome must be installed. For configuring the connection the application must know the IP of the Arduino board. For communicating with the Arduino it uses HTTP protocol. It sends the request along with the identifier for the device and status that the user needs. Further, the sketch installed in the Arduino board contains the logic for handling the user's requests. After an action is executed the board replies with a JSON response to the phone.

#### 4.2.2.1 Implementation

The implementation of DomoticHome using Wi-Fi connection requires the ethernet module to be installed in the Arduino board. Later, a LED is connected to the digital pin number 8 and a thermistor module to the analog pin number 5. A Arduino Sketch is needed to program the microcontroller. DomoticHome has an online Arduino code generator that can be accessed after registering an account on their website. The generator requires information about the modules that are used and on which pins they are connected. In addition it needs the IP for the Arduino, that it will use in the network, and it's MAC address. Due to the fact that the Arduino considered in the study is a newer model, some corrections were needed in the code due to different syntaxes among the versions. After uploading the code to Arduino via Arduino IDE and USB-cable and synchronizing the connection between the board and phone, we we were able to control the LED and get temperature readings through the DomoticHome application.

## 4.3 Summary

After evaluating both wireless mechanisms it is observed that it is difficult to set up if one does not have the technical knowledge concerning microprocessors and wireless communications. However, with little research, a system like DomoticHome is rather feasible in a home environment. In comparison the other systems needs programming skills and overall computer knowledge but as a result, one can customize the solution to fit specific requirements. In addition, DomoticHome is connected to the router or a power outlet - which means, that it is powered by the router therefore no battery consumption issues arise. In contrast, the Bluetooth-based communications brings the power consumption issues to the table since it needs to be powered by 3 AA batteries. In a parallel work to this study it has been shown that under a scenario when a message is sent to the phone every minute the Arduino BT solution shows reasonable levels of performance. It has been observed that the battery stands for a period of 39 hours 20 minutes (45).

# 5

# Conclusions

Home automation is becoming more popular due to the latest developments in hardware which have significantly reduced the cost and improved the capabilities. It is due to the fact that technology around us evolves and the access to needed information is easier than ever. Consequently, demand for these systems is increasing and different manufacturers have realized that. This has been the motivation for several smart home approaches such as Mattia Lipreris' DomoticHome and sophisticated Insteon. Furthermore, toolkits like Amarino were not intended for home automation but they can be easily adapted to fit the smart environment requirements.

Automating your home is feasible these days. Although, it has been around for a while, it has not been a potential option for a lot of people due to its immense cost. However, thanks to the development of Android and Arduino technologies, practically anyone can implement some kind of automation at their home. After the analysis of latest developments, such as Zigbee, Z-Wave, Android@Home, Domotichome, X10, Insteon, we have highlighted different decision criterions and brought out the advantages and disadvantages of every system.

Although microcontrollers have been in home automation solutions for a long time, none of them have been open-hardware and open-source. The emerging of microcontrollers like Arduino fosters the development of smart homes solutions. With the add-on modules Arduino gives us endless opportunities to link and configure different devices in our home. By implementing the wireless-based systems, we showed how Arduino can be adapted in a smart home environment.

Furthermore, there are several solutions in the market. The thesis contribution includes the analysis of several solutions to highlight weakness and strengths so one can choose an appropriate solution in terms of installation difficulty, reliability, communication type, scalability and cost.

Google announced the release of Android@Home by summer 2012 and it is expected that this solution will gain popularity in the market. This may revolutionize the home automation domain since a big number of people already have an Android device at their disposal for controlling devices. Furthermore, Google is a billion-dollar corporation that has the knowledge for developing a system that is intuitive and feasible for practically everyone. Lastly, a lot of devices already have Android capabilities so integrating them in a home environment will be easier. However, the problem of standardization remains unresolved. The possibility of a global standard is remote.

Finally, we consider that the selection of the automation system depends on the scenario and requirements. In the long term home automation solutions can help in reducing costs, foster the centralization of devices to a single control unit and help in reducing the carbon footprint with intelligent resource utilization.

# Tarkade majade analüüs ning võrdlus

## Bakalaureusetöö (6 EAP)
## Tanel Lips
## Resümee

Tark maja on kodu, mis on osaliselt automatiseeritud. Targa maja peamine ülesanne on kõik seadmed koondada tsentraalse juhtimise alla, mis võimaldab programmeerida erinevate sündmuste korral tegevusi, mis kõige paremini maja ja omaniku soovidega kokku sobivad.

Käesolev bakalaureusetöö analüüsib ning võrdleb erinevaid tehnoloogiad, mille abil on võimalik enda kodu automatiseerida. Võrreldakse kuut lahendust milleks on: Insteon, X10, Z-Wave, Android@Home, Zigbee ning DomoticHome. Lisaks on teises peatükis välja toodud kodu automatiseerimise lühiajalugu, mida siiamaani pole veel keegi kirja pannud. Samuti rakendame kaks juhtmeta ühendusega lahendust. Mõlemad süsteemid põhinevad Arduino mikrokontrolleril ning Android operatsioonisüsteemiga mobiiltelefonil. Erinevus seisneb suhtlusprotokollis. Esimene süsteem, DomoticHome, kasutab suhtlemiseks Wi-Fi tehnoloogiat. Teine põhineb Amarino toolkit-il ning Bluetooth suhtlusprotokollil. Mõlema süsteemiga saab ühendada valgusteid, temperatuuri andureid ning muud vajalikku, mis võib ühes kodus vaja minna. Seejärel saab telefoni abil neid seadmeid sisse- ja välja lülitada ning saada erinevaid mõõtmisi Arduino sensoritelt.

Analüüsis sai välja toodud kõikide võrreldud tehnoloogiate plussid ning miinused vastavalt kriteeriumitele, mis lõputöö algul said ära otsustatud. Rakendamisel valmisid kaks lahendust. Mõlemaga sai sisse- ja välja lülitada Ar-

duino külge ühendatud LED-valgustit ning mõõta toatemperatuuri thermistor-mooduli abil. Kuna DomoticHome tehnoloogia genereerib automaatselt Arduino mikrokontrollerile juhtimiskoodi, siis selle seadistamine oli võrdlemisi lihtsam kui Bluetooth ühendusega süsteemi ülespanemine. Amarino lahenduse jaoks oli vaja algul õppida Arduino mikrokontrollerit programmeerida. Tavainimese jaoks võib see osutuda keeruliseks.

DomoticHome lahenduse Arduino mikrokontroller saab voolu seinapistikust või ruuterist, mille külge ta on ühendatud. Erinevalt DomoticHome-st saab Amarino Arduino mikrokontroller voolu 3-lt AA-tüüpi patareilt. Paralleelselt valminud Steve Mägi bakalaureusetöö eksperimendid näitasid, et süsteem võib mõõdukal kasutamisel järjest töötada 39 tundi ning 20 minutit. Kui inimesel pole võimalik sellise aja tagant vahetada patareisid, siis see süsteem ei sobi antud isikule ja peab valima DomoticHome lahenduse.

Arvestades, et praeguseni on kodu automatsioon olnud suhteliselt kallis, siis võib loota, et kui Google enda süsteemi turule toob, siis tarkade majade arv kasvad hüppeliselt. Põhjus on selles, et süsteem toimib Androidi operatsioonisüsteemi põhjal ning tänaseks on maailmas juba 300 miljonit aktiveeritud Android seadet. Üleüldiselt on tegu kiiresti areneva ning huvitava valdkonnaga. Kindlasti vajab see edasist uurimist ning arendamist.

# Bibliography


[1] Wikipedia, Home automation.
URL http://en.wikipedia.org/wiki/Home_automation 1

[2] Arduino, Arduino.
URL http://www.arduino.cc 2, 7

[3] T. Ricker, Mi casa verde vera review: Home automation, simplified.
URL http://www.engadget.com/2010/12/17/mi-casa-verde-vera-review-home-automation-simplified/ 2

[4] V. Matos, R. Grasser, Building applications for the android os mobile platform: a primer and course materials, J. Comput. Sci. Coll. 26 (1) (2010) 23–29.
URL http://dl.acm.org/citation.cfm?id=1858449.1858455 3

[5] E. Burnette, Hello, Android, The Pragmatic Bookshelf. 5, 6

[6] C. Arthur, Google data suggests about 42m android devices activated in december.
URL http://www.guardian.co.uk/technology/2012/jan/05/google-42m-android-activated-december 5, 25

[7] Android architecture.
URL http://developer.android.com/images/system-architecture.jpg 6

[8] Arduino, Arduino language reference.
URL http://arduino.cc/en/Reference/HomePage 7

[9] B. Kaufmann, Design and implementation of a toolkit for the rapid prototyping of mobile ubiquitous computing, Master's thesis, Alpen-Adria-Universitt Klagenfurt (2010).
URL http://www.amarino-toolkit.net/tl_files/thesis/amarino_thesis_kaufmann_2010.pdf 7

[10] J.-S. Lee, Y.-W. Su, C.-C. Shen, A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi, in: Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE, 2007, pp. 46 –51. doi:10.1109/IECON.2007.4460126. 8

[11] E. Ferro, F. Potorti, Bluetooth and wi-fi wireless protocols: a survey and a comparison, Wireless Communications, IEEE 12 (1) (2005) 12 – 26. doi:10.1109/MWC.2005.1404569. 8

[12] SmartHomeUSA, X10 story.
URL http://www.smarthomeusa.com/info/x10story 9, 10, 22

[13] M. Murata, T. Namekawa, R. Hamabe, A proposal for standardization of home bus system for home automation, Consumer Electronics, IEEE Transactions on CE-29 (4) (1983) 524 –530. doi:10.1109/TCE.1983.356359. 9, 10

[14] C. Douligeris, J. Khawand, C. Khawand, Communications and control for a home automation system, in: Southeastcon '91., IEEE Proceedings of, 1991, pp. 171 –175 vol.1. doi:10.1109/SECON.1991.147729. 10, 11, 12

[15] Popular science (1993). 10, 12, 13

[16] M. Inoue, K. Uemura, Y. Minagawa, M. Esaki, Y. Honda, A home automation system, Consumer Electronics, IEEE Transactions on CE-31 (3) (1985) 516 – 527. doi:10.1109/TCE.1985.289966. 10, 11

[17] R. Hamabe, M. Murata, T. Namekawa, A revised new proposal for standardization of home bus system for home automation, Consumer Electronics, IEEE Transactions on CE-32 (1) (1986) xi –8. doi:10.1109/TCE.1986.290110. 10, 11

[18] K. Wacks, The impact of home automation on power electronics, in: Applied Power Electronics Conference and Exposition, 1993. APEC '93. Conference Proceedings 1993., Eighth Annual, 1993, pp. 3 –9. doi:10.1109/APEC.1993.290658. 10, 13, 14

[19] C. Tribune, Smart house model opens in naperville.
URL http://articles.chicagotribune.com/1993-11-20/news/9311200042_1_smart-house-ceiling-ashbury 10, 14

[20] P. Corcoran, J. Desbonnet, K. Lusted, Cebus network access via the world-wide-web, in: Consumer Electronics, 1996. Digest of Technical Papers., International Conference on, 1996, p. 236. doi:10.1109/ICCE.1996.517285. 10, 14

[21] N. Sriskanthan, F. Tan, A. Karande, Bluetooth based home automation system, Microprocessors and Microsystems 26 (6) (2002) 281 – 289. doi:10.1016/S0141-9331(02)00039-X.
URL http://www.sciencedirect.com/science/article/pii/S014193310200039X 10, 16

[22] K. Wacks, Home systems standards: achievements and challenges, Communications Magazine, IEEE 40 (4) (2002) 152 –159. doi:10.1109/35.995865. 10, 15

[23] A. Alheraish, Design and implementation of home automation system, Consumer Electronics, IEEE Transactions on 50 (4) (2004) 1087 – 1092. doi:10.1109/TCE.2004.1362503. 10, 16

[24] A. Al-Ali, M. Al-Rousan, Java-based home automation system, Consumer Electronics, IEEE Transactions on 50 (2) (2004) 498 – 504. doi:10.1109/TCE.2004.1309414. 10, 16

[25] P. Darbee, Insteon (8 2005).
URL http://www.insteon.net/pdf/insteondetails.pdf 10, 19, 20

[26] P. Kinney, Zigbee technology: Wireless control that simply works (10 2003).
URL http://search.mouser.com/pdfdocs/ZigBeeTechnology.pdf 10, 26

[27] JFR, Z-wave protocol overview.
URL http://www.eilhk.com/en/product/Datasheet/Zensys/ 10, 16, 23

[28] DomoticHome, Domotichome.
URL http://www.domotichome.net 10, 17

[29] T. Ricker, Editorial: Android@home is the best worst thing that could happen to home automation (5 2011).
URL http://www.engadget.com/2011/05/11/editorial-android-home-is-the-best-worst-thing-that-could-happe/ 10, 17

[30] R. K. Honaman, Carrie transmission over power circuits (1924).
URL http://www.google.com/patents 9

[31] B. Markwalter, C. Russell, Consumer electronics bus, a robust communications system, in: Consumer Electronics, 1988. Digest of Technical Papers. ICCE., IEEE 1988 International Conference on, 1988, pp. 42 –43. doi:10.1109/ICCE.1988.10695. 12

[32] C. Maxfield, Googles android@home the plot thickens (7 2011).
URL http://www.eetimes.com/electronics-blogs/other/4217521/Google-s-Android-Home---The-plot-thickens- 17, 24

[33] Insteon starter kit.
URL http://smarthomegadgets.com/wp-content/uploads/2008/01/insteon.jpg 20

[34] C. Gomez, J. Paradells, Wireless home automation networks: A survey of architectures and technologies, Communications Magazine, IEEE 48 (6) (2010) 92 –101. doi:10.1109/MCOM.2010.5473869. 19, 24

[35] Insteon prices.
URL http://www.thehomeautomationstore.com/insteonwire-in-outlet-modules-.html 21

[36] X10 starter kit.
URL http://cf.mp-cdn.net/02/fe/53344e0288966abd1dd14d3d8127.jpg 21

[37] SmartHomeUSA, X10 theory.
URL http://www.smarthomeusa.com/info/x10theory 22

[38] X10 prices.
URL http://www.thehomeautomationstore.com/x10-appliance-modules.html 22

[39] Z-wave starter kit.
URL www.smarthomeusa.com/Products/WD-HA-KIT/images/largeWD-HA-KIT.jpg 23

[40] Z-wave prices.
URL http://www.thehomeautomationstore.com/zw-garage.html 23

[41] Android@home.
URL http://www.intomobile.com/wp-content/uploads/2011/05/Android-Home.jpg 24

[42] G. Mulligan, The 6lowpan architecture, in: Proceedings of the 4th workshop on Embedded networked sensors, EmNets '07, ACM, New York, NY, USA, 2007, pp. 78–82. doi:10.1145/1278972.1278992.
URL http://doi.acm.org/10.1145/1278972.1278992 24

[43] Zigbee starter kit.
URL http://media.digikey.com/photos/DigiInt'lPhotos/XK-B11-SK-W.JPG 25

[44] Introductory picture of domotichome.
URL http://www.domotichome.net/images/p2/p_2_wiring.JPG 26

[45] S. Mgi, Arduino based tracking framework for object localization (2012). 38