Modular Design and Implementation of a Low Cost Home Automation System using Web-Services

A thesis presented in partial fulfilement of the
requirements for the degree of

Master of Engineering
in
Computer Systems

at
Massey University, Albany,
New Zealand

Sunny Peter Gomasa

December 2011

Abstract

The idea of home automation has existed and worked on by several researchers. The idea of controlling devices, around a home, in way to provide improved comfort and security has been in the way since early 19th century.

Current advancements in computing and communication technology allow for designing smart home automation systems that can manage several devices from one central location. Several researchers have been working in designing one that can manage, maintain and process data with very little user interaction. Therefore, the aim of this thesis is to design and implement a low-cost home automation system that is independent of networking protocol, is scalable and easy to deploy & maintain.

The system works as several modules operating independently while coordinating with a central gateway. Zigbee protocol was used provided wireless communication to devices that require low-power and not a whole lot of computing. Lastly a discussion was made to use Open-source software to keep cost to minimum.

Acknowledgements

A number of people have supported me throughout my research. The following acknowledgments go a small way to expressing my thanks.

Firstly, I would like to thank my academic supervisor Dr. Fakhrul Alam for his continued guidance, and support. I would also like to thank my family for their kind understanding and for supporting me though some intense times.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Home Auomation Problems

In the recent years there has been increase in researchers developing various systems that provide improved comfort, convenience, security and efficiently manage energy in a standard home[1][2][3]. Even with such growth and contribution, this technology is still absent in an average home today[4] as such systems are often custom designed at the time of construction. This tends to put a hefty price tag to attract an average home owner. Several standards, such as X10, OSGi, Zigbee, that support in building an automated home already exists, however, both the end-user and the designer needs to discipline themselves to one of these standards. This not only puts limitations on scalability but also can be very limiting in-terms of future proofing. It is clear that establishing a high level of interoperability among these several standards is very important[3][5]. Users need to be able to purchases several devices and be able to easily deploy and install them with limited technical knowledge.

Among several researchers, only few suggest the idea of retro-fitting smarts into an existing home[6]. Retro-fitting becomes very important as several users who wish to add smarts to their existing home do not need to go through an expensive refurbishing or upgrading.

It would be an understatement to say that internet has become the back-bone to our modern day-to-day society. Several researches such as [1][4] have tried to integrate their smart home system, using a centralised control that is accessible over the internet. This provides way for the users to gain control of their devices from any were on the planet. However, not many have addressed the possibility of internet disconnections, security. To attract the general populace, these central systems need manage un-expected events gracefully without impeding the overall function.

1.2 Centralised Control

In order to achieve a centralised control several research such as [7][8][4] proposed the idea of a central gateway. Through this, a simple manageable point can be established in order to control devices in a home that act as a translator between several different networking protocols. This also can be use to gain access to the internet by simply connecting to a modem or a router.

A gateway uses some common sets of instructions and commands to manage and control devices. There has been several impelementations of a gateway that extends its capable to the internet that use a Ethernet connection or a Wi-Fi connection. Few researchers have used a standard computer with custom software, while others have designed low-powered embedded hardware with just enough resources.

1.3 Wireless networking

All the devices in a home automation environment are networked using some type of networking protocol. While several of these protocols already exits, some offer better performance while others offer better economical value [9].

Wireless technologies are preserved as being more flexible in terms of deployment and management. In recent years, there has been a huge growth across both the industry and academia giving rise to several of wireless standards that are both low-power and low-cost while offering features, such as mesh-networking, which are usually seen in wired networks[10]. [11] surveys the main current and emerging architecture and technologies tailored to or suitable for Wireless home automation networks.

1.4 Smart devices

Devices that would essentially provide improved comfort to day-to-day life need to be smart or at-least are cable of performing basic tasks efficiently[3]. A switch for example, should maintain a state and not be tied down to light as it is in a regular home. Giving users the ability to change which other device or devices the switch controls is important, as this provides extended functionality as well as let the switch do what it does best, turn another device on or off.

1.5 Web services

With the ever increasing demand for web based services, several home automation researchers in [5] have integrated their systems to the internet using SOAP based web-

services. A web services has been gaining tremendous movement in the last decade due to its multi-platform and multi-language support. Currently, all the modern programming languages provide support for web-services while older languages are getting retro-fitted with this ability. Some of the most popular services such as Twitter and Facebook are good examples of well implemented web services.

1.6 Objectivies of the Thesis

This thesis proposes a design and implementation of a modular low-cost home automation system that is independent of networking protocol, is scalable, easy to deploy and maintain. The design aims to provide end users with three levels of interfacing.

1. **Regular:** Devices that look and work like their *non-smart* counterparts, for example, turning on a light switch at the wall.

2. **Local:** A central device that can be used to control other devices on the network.

3. **Internet:** A system that can be used to control devices over the internet.

This will be achieved by modularising various section of overall architecture. This ensures that at-least one interface is operational at all times. To reduce over-all cost of system the decision to use robust open-source platforms for software development has been made.

1.7 Scope of the Study

The final objective of this project is to design and implement a modular home automation system based on Zigbee devices with at least three levels of interfacing.

Chapter 2

Literature Review

2.1 Overview

In the recent years there has been an exponential growth and advancements in computing technology[3][12]. There has also been an increase in use of these technologies, even, among non-technical users as they are no longer limited to a personal computers that occupy fixed desk space. Instead, there has been increasing trend towards ubiquitous computing that integrates seamlessly into peripheral environment to assist with ones day-to-day life. In many cases, end-users are not aware of this network of devices as they seamlessly serve required information.

Several standards are been proposed each day promising to solve other standards issues. Home automation is no different. At a very basic level, home automation has been around as early as 19th century with the introduction of water supply and energy distribution systems[13]. Since then, several solutions were presented by both the industry and the academia but the progress has been relativity slow.

Few systems [9][14] aim to solve issues such as ease of access [15] and scalability, while others [5][16] look into making several different standards communicate with each other seamlessly. The use of wired standards or technologies, such as X-10, has been slowing losing the momentum to wireless standards [17]. Wireless standards also offer solutions that simplify retrofitting existing homes with new features for a reasonable cost. There have been several advances in both hardware technologies and wireless communications which enable development of robust wireless sensor networks (WSN) that can be used in home automation [16].

2.2 Wireless Sensor Netwoks

A wireless network, when compared to wired networks, provides several advantages such as flexibility, cost, and security[18]. This is even more so when dealing with applications that do not require high bandwidth or low-latency[19]. A wireless network composed of numerous sensors is known as Wireless Sensor Network (WSN). These networks are used for monitoring physical conditions such as weather, temperature and in some cases control of other devices[20]. The use of WSN has been gaining interest among several home automation researchers[9][16][18][19].

A WSN consists of small or large nodes known as sensor nodes. These nodes transmit and route necessary data to another node that will result in efficient use of power and resources[19]. The strength of WSN lies in the ability to deploy large number of these nodes that assemble and configure themselves[21]. Though WSN support several networking topologies, the most favoured is mesh. In a mesh network, as long as there is sufficient density, a single network of nodes can grow to cover

limitless area with the ability to route data across different paths[22].

Due to several technological advances many WSN protocols have emerged offering low power radio transceivers, small form factor and extreme scalability[23]. In WSN, the physical radio layer defines the operating frequency, modulation scheme, and hardware interface of the radio to the system. There protocols that offer properties make WSN ideal[22][23].

2.2.1 IEEE 802.11

IEEE 802.11, also known as Wi-Fi, is a standard that is designed for wireless local area networking. Since this is used to replace wired LAN it aims to achieve relatively high bandwidth and data transfer[24]. IEEE 802.11 has a typical transmission range of 30 meters indoors and 90 meters in line-of-sight. The data rate can vary between 1 Mbps to 150 Mbps depending on the protocol version[25]. IEEE 802.11 standard offers very high data rates with high power consumption. This high power requirement usually prevents researches from using standard[26][27]. However, it has been used in home automation applications in which audio/video equipment is used[28].

2.2.2 IEEE 802.15.1

IEEE 802.15.1, popularly known as Bluetooth, is a standard that has much lower power requirement than that of IEEE 802.11. It is a personal area network (PAN) standard originally aimed to serve to transfer data from a computer to peripheral devices such as keyboard, mouse or cell phones[29]. It uses star network topology that supports up to seven remote nodes communicating with a single base-station.

IEEE 802.15.1 standard is considered to have a relatively high power for a short transmission range. Numbers nodes are limited to seven and take long time to synchronise to network when returning from sleep. Due to this, the use of IEEE 802.15.1 has not been a popular choice among several researches. However, it is not completely ignored as this standard is extremely common among new devices such as mobiles and cameras[29][30].

2.2.3 IEEE 802.15.4

The IEEE 802.15.4 standard has been specifically designed to meet the requirements of wireless sensing applications. The standard is very flexible as it supports multiple data rates, transmission frequencies and network topologies. The power requirements are moderately low; however, the hardware is designed to allow for the radio to be put sleeping, which reduces the power to a minimal amount. When compared to IEEE 802.15.1 rapid synchronisation is achieved when a node is waking up from sleep. This capability allows for very low average power supply current when the radio can be periodically turned off[31].

IEEE 802.15.4 standard can operate across several frequencies such as, 868 MHz, 902–928 MHz and 2.48–2.5 GHz while offering data rates of 20 Kbps (868 MHz Band) 40 Kbps (902 MHz band) and 250 Kbps (2.4 GHz band). Standard also offers optional use of AES-128 security for encryption of transmitted data[4][31]. The 2.4-GHz band has be widely used as it is a worldwide license-free band and it also offers high data rates resulting in lower system power due to the lower amount of radio transmission time to transfer data as compared to the lower frequency bands[32].

IEEE 802.15.4 has been widely accepted for wireless sensing applications among several researchers and it forms the basis of several WSN specifications such as Zigbee, WirelessHART, 6LoWPAN and MiWi. At the time of this writing Zigbee seems to be the most favoured specification[4][17].

2.2.3.1 Zigbee

Using the IEEE 802.15.4 standard, Zigbee Alliance is seeking to standardise a higher level protocol that helps in developing applications such as lighting control and HVAC monitoring systems. Zigbee protocol has been greatly welcomed by several WSN and home automation researchers[17][33][34]. Several companies such as Atmel, Digi International, Freescale, Ember and many others are already developing low-powered radios based on this protocol.

2.3 Internet and Web Services

It cannot be denied that internet connectivity is been treated as a necessity in our modern day-to-day life[3]. Several services, such as checking latest weather report or finding closest restaurant, are been offered to user that enable them to access information faster[12]. It is suggested that users, including non-technical, do not take a business seriously that do not have a web-site. Such popularity has sparked several home automation researchers to design their system that have internet capability.

In all its glory, internet has also been subjected to issues like privacy. Devices constantly transmit and receive information that users prefer not to publish. When it comes to Home automation privacy becomes the most important concern that users

and researcher alike face[35]. Storing which door in a house is unlocked could open up a serious threat to the home owner, while ability for him/her to check for any locked doors while not at home could provide an extra sense of security.

There have been several suggestions on integrating WSN to the internet in order to provide a flexible system[18][36][37]. WSNs are usually connected to the internet with help of a central gateway that extends control over the internet[18]. While some take simple approach of hosting software on users computer which then enables internet control, while others try to design systems that operate on own or use a web services to relay information[36][37].

2.3.1 Internet integration

While using WSN protocols, such as Zigbee, one must consider on how to it can be interfaced with Internet. Several researchers such as [18][38] have proposed of a gateway that provides a simple manageable setup through which different protocols can be translated to IP packets. Zigbee based gateways, such as [34][8][39][4][40], were implemented with few using a regular computer as a there gateway while others use much more streamlined approach by using microcontrollers. Some researchers have also implemented Bluetooth based sensor network [38].

The gateway approach for integrating WSN to the internet has been preferred [4] as applications for sensing will result in small amount of data per packet, while using IP based protocols such as IPv6 has large amount of header information per packet [41]. In case of low-powered sensing devices, smaller header information will result in little data to be transmitted and thus using less power. Few researchers have pushed

the idea of using 6lowpan, a protocol based on IEEE 802.15.4, for home automation [42].

Software architecture becomes important when designing a system with a web front end. Protocol translating methods play a vital role in successfully integrating the system to the internet. The architecture proposed in [43] allows multiple family members to simultaneously monitor devices using an external internet resources. A personal computer was used as a hardware server to host the software and to provide internet connectivity. JavaBeans and JSP were used to publish both html based web pages as well as RSS feeds [1]. Zigbee protocol was used in [44] as it provided the researchers a very secure (uses Advance Encryption Standard) and self-configuring networks.

A modular based system was implemented in [45] with accessible way to control by means of TV's remote control. Few systems, such as [46][47][48] have based their systems on OSGi service framework (Java based framework that enables modular services). Other researchers such as [12][18] have used XML based profiles in order to modularise their systems.

In [4] a home automation system, that integrated with internet, was designed and developed using both Zigbee and Wi-Fi. Here central gateway not only provided internet connectivity but also helped low data rate devices communicate and control devices that require high data rate, such as multimedia.

2.3.2 Web Services

Few researchers have implemented their systems that offer home automation as a service using standards such as OSGi [48] or custom implementations [49]. In WSN, similar service based monitoring systems were proposed by [18][50] that use web services.

The concept of web services is that a set of application programming interfaces (APIs) are located at a remote location and any application, with given appropriate access, can consume it. Its architecture provides a means of integrating with systems through internet in a relatively simple manner that provide elegant publication and discovery options. Consequently web service based frameworks can be employed to conveniently access different devices through the internet [14].

Web services provide a means of integration application by using open standards, protocols and languages that are widely adopted on the internet. The core of its architecture is composed by HTTP, through which XML message are exchanged using the SOAP (Simple Object Access Protocol). SOAP is a set of, strictly defined, XML based rules that are passed between the web service provider and program consuming it. This provides the flexibility to employ different programming languages such as PHP, Java, C++ and.NET, to develop various components of the system [14].

Similar approach could be used while designing a home automation [5]. However, due to the use of XML, large amount data may be passed back and forth from the service provider and the consumer when a system needs to be connected to a remote server to function correctly. Systems designed without the concept of service, in [47][48], have hosted HTTP services on users personal computers. This reduces the

amount of data handled by the server at one time but also does not provider same flexibility as using web service.

Few researches used gSoap, a C++ libary for developing web services, to expose their WSN sensor data over the internet. gSoap aims to provide low foot-print C or C++ stubs that help in building a web services server or a web service client. Though using gSoap at a sensor level would be very resource heavy, using such libary on a gateway would be very advantageous [51].

2.4 Common Platform

Several implementations of a home automation that have been proposed by researchers use messaging formats specially cratered to the application [4]. If a systems aims to operate in a protocol neutral manner it needs to communicate in language that is understood by all the other protocols. SOAP based Web Services tend to achieve this by using XML. Here, objects in a programming language are strictly translated into SOAP which can be translated back to language specific objects, even in a different programming environment. In order for sensor nodes to co-exist with other WSN standards, the data transmitted needs to be structured. For example, data objects represented in XML may be read by any application while a custom representation requires a detailed documentation. Structured data objects, using XML, are also easily understood programmatically [14].

In [52][53] the researchers use a driver based approach were an extra layer of software is placed that acts as a translation unit. With this, a Zigbee network can be allow to communicate with other Zigbee nodes without any overhead.

In [53] researchers propose the idea of sensor node profiles. The basic idea is that a system would know and understand what a sensor node does when its joins the network for the first time. For example, the coordinator does not have any prior knowledge of temperature sensor node before one joins the network. A temperature sensor will carry a profile with all the required information that will make the network coordinator understand what it does.

Few researchers have suggested the use of already existing standards such as XMPP [54] while some proposed newer ideas [55][56][57]. [58] aims to represent Zigbee profiles using XML in order to reduce the development time and increase the interoperability between vendors that develop Zigbee applications.

2.5 Summary

The literature reviewed in this section outlines potential methods, techniques and technologies that can be used to build a home automation system is low-cost, easy to deploy and maintain. The use of Zigbee protocols seems to be popular among WSN. Researchers have also attempted to build Zigbee based home automation systems that integrate with internet. Gateway based centralised architecture appears to be most common and most favourable approach.

Internet integration appears to be a popular theme among researchers; however, events such as temporary internet disconnections need to be addressed. Allowing other systems to integrate using Web Services offer potential scalability and easy way of handling data formats generated by various WSN standards.

Chapter 3

System Architecture

The focus of this thesis is to design and implement a low-cost home automation system that is independent of networking protocol, is scalable, easy to deploy and maintain. The system will also provide options to interface with internet using web services, but will not depend on it for its operation. Users will be given at least three levels of control while maintaining security and privacy.

3.1 Architecture

The presented modular architecture has been layered with several abstractions. This section provides an over-view on how system would function. As suggested by [34][8][39][4] in previous chapter, a central gateway approach was selected. This provides a central access to manage and access to internet without using high-powered radio devices.

The gateway handles and processes all communications while the system is opera-

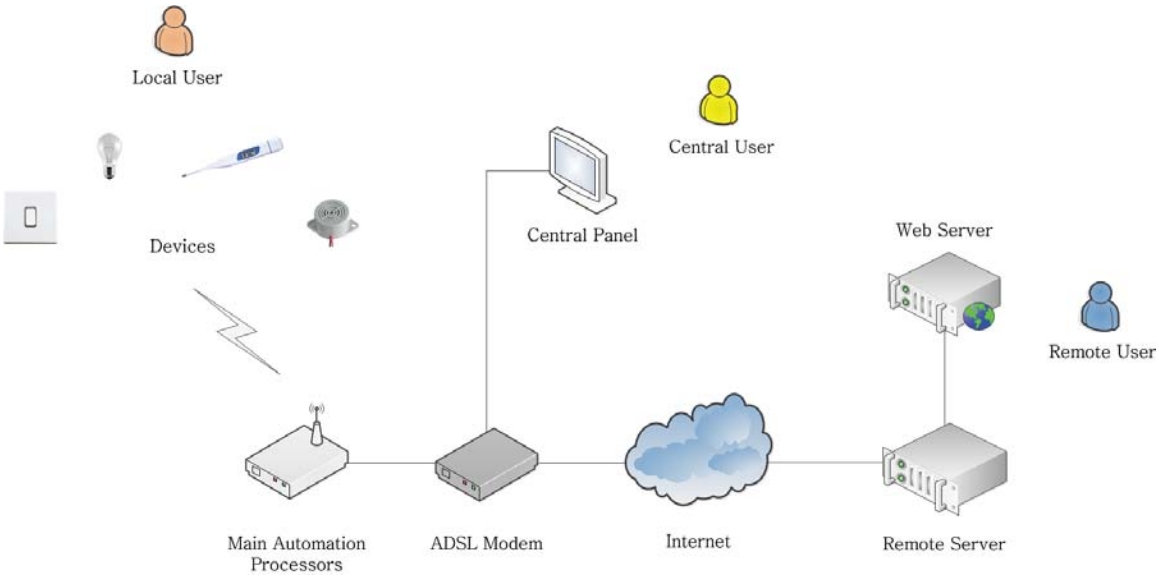

Figure 3.1: System architecture

tional. It is also responsible for handling events, hosting web service and other trivial tasks. However, the gateway presented in this design is much more than a device that passes information around and therefore it is called Main Automation Processor (MAP). Fig 3.1 shows the system architecture at the most abstract level.

In [39][4][40], a standard computer was used for this purpose. However users may not be willing to leave their computer turned on all the time, or simply do not have a computer to spare. The alternative was to build the MAP using embedded hardware. This however would become quite a challenge as this embedded device needs to perform several high-level operations, such as hosting web-service and web-site.

A compromise was made by implementing MAP on a Single board computer, like researche[59]. PC Engine ALIX 3C2 was used as it provided enough computer power and resources while maintaining the low-cost, low-power and small form factor. It

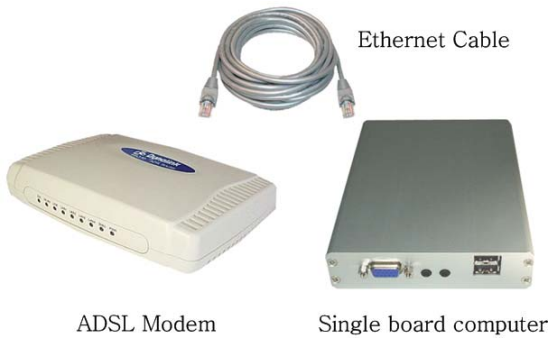

Figure 3.2: Single board computer, ALIX 3C2 and ADSL Modem

also provided network connectivity using Ethernet and options to run Voyage Linux, a light weight embedded operating system based on Debian Linux. Figure 3.2 shows ALIX 3D2 next to a standard ADSL modem. With this approach the MAP just operates like any other computer but has very specific tasks to perform. In order to ensure that MAP operates without any user interaction, the programs running on the MAP are demonised and configured to run at system start up.

The architecture presented allows for three levels of user interaction with the system. Users will be able to interact with devices as they would with any normal device. For example, user will be able to simply walk up to a switch and turn it on. Other level of interaction is where a central interface is provided, for example a touch screen system. The final level of interaction that users could take advantage of is the access over internet. Careful separation and interaction of these interfacing will ensure that at least one is operational at all times.

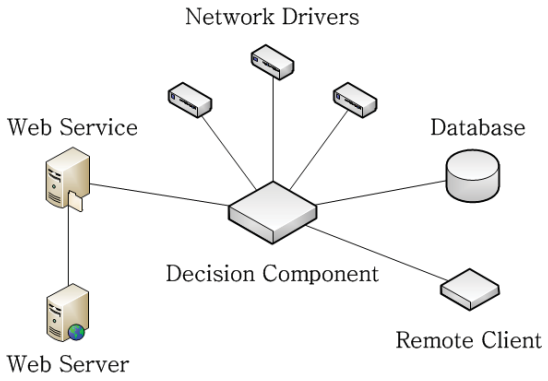

Figure 3.3: Main Automation Processor (MAP)

3.2 Modular Design

At an abstract level, the system is divided into several modules. These modules operate concurrently in order to maintain all devices, device events, local web server, web service and remote access client/server. This section provides an overview of all these modules and were they fit in the overall system.

3.2.1 Device Network

Device network consists of devices that are essential for any home-automation system. These may include devices like, but not limited to, lights, switches, sensors and alarms. These devices are designed perform all the tasks normally expected of them plus provide a way to interface with MAP.

In this thesis, device network is based on Zigbee protocol by using Digi's XBee modules for Zigbee stack. Microchip's PIC18F1220 are used for limited amount of processing and memory required for holding its profile, interfacing with XBee modules and responding to inputs from both external sources and MAP.

3.2.2 Network Driver

Network Driver, a program running on MAP, acts as the coordinator for a specific network. This module ensures that any protocol specific data packets are translated to/from XML packets which are understood by the Decision Component module. A single network driver is specifically written to interface with one networking protocol and several network drivers can exist and operate concurrently. Each driver translates incoming data to a common understood XML structure and also ensures that data is translated from XML standard to protocol specific packets.

3.2.3 Decision Component

Decision Component is the most important module that runs on MAP. It sits at the centre of all other modules such as database, web service and network drivers. It is directly or in directly, responsible for communicating with all other modules in overall architecture and very similar to central gateway implemented by researches in [7][8][4]. To ensure commonality between modules all the information that is coming in and going out off Decision Component is encoded as XML. Decision Component, as name suggests, is also be responsible for making decisions and carrying out tasks such as deciding when to change the room temperature to an "acceptable" level.

3.2.4 Database

Researchers in [40] have setup a simple database to allow for scalability of devices in a home automation network. Similarly database module, which is configured on MAP, will be not only be used to log historic information but also to maintain

information on device networks, devices on networks and user defined rules. The advantage of using a database is that it provides an easy way to pair up devices that are of different networks. Decision Component heavily depends on the database and is the only module with direct access to it.

3.2.5 Web Service

Web Service, as proposed by [18][50], will be used to expose home-automation network over the internet or just LAN. As web services are platform and language independent the information can be projected on to virtually any modern device, such as smart phones, websites, PDAs. All calls made to this module are processed and passed on to Decision Component. Decision component then performs the specified task and passes it back to Web Service which them is returned back to the caller. This approach ensures that all the calls made by any module are only handled by Decision Component.

3.2.6 Local Web Interface

Local Web Interface module, hosted on MAP, will consume the Web Service and present system configuration settings, network and device information to the user. Users could use this interface for to control device from any other computer or a device with a web browser. The interface will be designed using XHTML, CSS, and JavaScript while PHP will be used to interact with the web service module.

3.2.7 Remote Client

Remote Client module is only used if user wishes to access their home-automation network over the internet. The single purpose of this module is to provide Remote Server module with the current global IP address and port on with the Web Service is hosted. If user has a static IP address there will be no use for this module, however many users do not have a static IP assigned to their home internet connection.

3.2.8 Remote Server

Remote Server module is hosted outside of user's network and MAP. This module could be run as a service to which helps users to their home network over the internet. Remote service gets an update from Remote Client with information such as latest IP address and port on which web service is hosted on. For privacy reasons, only this information is stored in a database of this module.

3.2.9 Remote Web Interface

This module will be used when a user wants to control their network over the Internet. This module uses information stored by Remote Server and consumes web service hosted by user's network. The interface provided to users in this module also designed using PHP.

3.3 Security

Addressing privacy and security becomes extremely important when a network of devices that monitor, store and process information based on sensors in a home. Information moving across difference systems needs to be encrypted in case of any middle-man attacks.

3.3.1 Device to Device security

In a complex system as this, devices on the network store and transmit sensitive information that may potentially compromise home security. RF packet sniffers could be used to find out the status of alarm or in extreme cases shut it off.

Zigbee protocol, as discussed by [17][33][34],used within the scope of this thesis, standard supports various levels of security that can be configured as needed by the application. Security provisions include 1) 128-bit AES encryption. 2) Two security keys that can be pre-configured or obtained during joining. 3) Support for a trust centre. 4) Provisions to ensure message integrity, confidentiality and authentication.

With security enabled, packets transmitted are encrypted using 128-bit AES (Advanced Encryption Standard). Use of a network key and optional link key can be used to further extend security of the network. Only devices with the same keys are able to communicate together in a network. Routers and end devices that will communicate on a secure network must obtain the correct security keys.

Allowing devices to join the network in a secure manner is also important. For example, when a user, without necessary technical knowledge, buys a new device he/she needs to be able to use it with very little effort without scarifying the integrity and

security of their existing network. Implementation of device association is discussed in chapter 5.

3.3.2 HTTP Security

The use of web services and web interface simplifies the security overhead and planning as both operate on HTTP protocol. Secure socket layer (SSL) is the most common protocol used to secure communication across internet. It is designed to prevent eavesdropping and tampering information from middle-man attacks.

gSoap, C++ library is used to provide web services, offers the use of HTTPS and SSL for WS-Security by using OpenSSL. Likewise the webserver lighttpd, used to present local web interface, also uses OpenSSL to provide a secure connection when accessing using a web browser.

3.4 Extensible Mark-up Language

The extensible mark-up language (XML) is a simple, flexible and plain text based data format for describing information. The specification was design mainly to provide structure to data that is easily understood and transported across various protocols and programming languages.

The use of XML has been applied on all communications that occur between devices. This approach offers the flexibility of operating in a neutral communication protocol. For example, both Zigbee and Bluetooth based devices offer to turn on a function to XML string <switch status="1" />. This approach adds an over-head to devices as they now have to parse XML. However, this is a very small price to pay in order to achieve compatibility.

XML could be used to publish device profiles, events, and issue commands to functions. However, a device as simple as a switch will not have a microcontroller capable of parsing XML due to limited processing power and memory. Researchers [58] propose several techniques that can be used to reduce the amount of XML required and processed. Although this approach has many limitations to regular XML based structures, it helps reduce memory usage and offers an elegant way to parse XML on microcontrollers.

3.5 Profiles

As discussed in section 2.3.1, researchers such as [3][17] have put forward the idea of using XML based profiles in order to modularise their systems. This approach

advantages of using a common mark-up structure that remains the same across several protocols.

In order to achieve inter protocol communication, network profiles and device profiles will be used. These profiles will be hard coded into a network module or a device and describe on how that network or device operates.

3.6 Network Profiles

A network profile describes on how network operates and how messaging should be handled. This profile will be "hard coded" and be an integral part of the Network Driver module. The main purpose is to inform the Decision Component module and how device on this network operate. Information such as type of network, little description, profile type and device addressing will be described using XML.

The profile described in Figure 3.4 details information necessary for Decision Component module to understand Zigbee network and how to interact with devices on it.

```
<Profile Name="Zigbee" Type="Network">
	<Description>Zigbee Network</Description>
	<NetworkType>Mesh</NetworkType>
	<Device>
		<NetworkFunction Uses="1">
			<Coordinator MaxMin="1"/>
			<Router MaxMin="*"/>
			<EndDevice MaxMin="*"/>
		</NetworkFunction>
		<Addressing Uses="All">
			<Address64 fixed="true" type="hex" length="16"/>
			<Address16 fixed="false" type="hex" length="2"/>
		</Addressing>
	</Device>
</Profile>
```

Figure 3.4: Zigbee Profile

The attribute Type on main Profile element can contain two values, 1) Network and 2) Service. In case of Zigbee, Profile type is a network while in case of Web Service the Type is service. This becomes essential to Decision Component when deciding which network devices can be paired. For example, a program consuming the web server should not be able to pair with a light switch as the program call only exists for a very small period. Child elements <Description> and <NetworkType> are purely for information purpose but required when the profile type is Network. When profile type is Service, the <NetworkType> is not required.

The <Device> element in profile describes how devices are recognised. In case of Zigbee network, each device can perform as one of three functions, 1) Coordinator, 2) Router and 3) End Device. Using the <NetworkFunction Uses="1"> element the Decision component module can be informed that each device can perform one of the functions. Likewise, the Max attribute on each function element informs on how many of these functions can exist on one network. This case of this example, Maximum of one coordinator is allowed while there is no limit to routers and end devices. Using this information Decision may treat devices differently based on its function, for example it may allow End Devices to sleep.

3.7 Device Profiles

Each device in a network has a profile with information describing it. The formatted XML for device is based on application specific format techniques, as suggested by [58], to reduce the amount of information stored and transmitted.

Each device has a set of functions to perform and profiles describe these functions

Table 3.1: Device functions

Function Code	Input/Output	Function
0x01	Input	On/Off Input
0x02	Output	On/Off Output
0x03	Input	Level Control Input
0x04	Output	Level Control Output
0x05	Input	Level Control with Switch Input
0x06	Output	Level Control with Switch Output
0x07	Input	Sensor Input
0x08	Output	Sensor Output

to the Decision component module using profiles. Profiles may also include a little description on its overall function and its manufacturer. Table 3.1 shows common functions that devices common perform.

Function codes are used to state what each function is meant to do on one device. For example, a lamp has two functions 1) On/Off that is input (switch) and 2) On/Off that is output (lamp). A light source however has only function, On/Off output. Using these basic function types a complex device can be built.

Fig 3.5 shows a profile for Lamp that has two functions. The <DE> element states the device description while <MF> holds manufacture name. Each function element contains a required TY attribute which says the type of function it is, for example the first function item has a type of 01 which indicates that it is an input On/Off while the second item has a type of 02 indicting that it is an output On/Off. The optional attribute DE just states the description.

A profile may provide its functions with identification numbers, which is always an integer, by specifying an ID attribute to each function item. However, this is not required.

```
<P>
        <DE>Lamp</DE>
        <MF>Massey</MF>
        <F ID="00" TY="01" DE="Light"/>
        <F ID="01" TY="02" PY="00" DE="Push Switch"/>
</P>
```

Figure 3.5: Lamp Profile

An automatic ID, starting with the value 1, is assigned by decision component module based on its location in the profile. In Fig 3.6 a. and b. are the profiles that considered valid. However, if one function is assigned an ID in profile all the other functions must be assigned too. Profile in Fig 3.6 c. is considered invalid. If function IDs are assigned in a profile, as seen in Fig 3.6 d, they need not follow the pattern of incrementing integer.

```
<D>
        <DE>Lamp</DE>
        <MF>Massey</MF>
        <F ID=1 DE='Light Switch'/>
        <F ID=2 DE='Light Bulb'/>
        <F ID=3 DE='Brightness'/>
</D>
```

a.

```
<D>
        <DE>Lamp</DE>
        <MF>Massey</MF>
        <F DE='Light Switch'/>
        <F DE='Light Bulb'/>
        <F DE='Brightness'/>
</D>
```

b.

```
<D>
        <DE>Lamp</DE>
        <MF>Massey</MF>
        <F ID=1 DE='Light Switch'/>
        <F DE='Light Bulb'/>
        <F ID=3 DE='Brightness'/>
</D>
```

c.

```
<D>
        <DE>Lamp</DE>
        <MF>Massey</MF>
        <F ID=34 DE='Light Switch'/>
        <F ID=89 DE='Light Bulb'/>
        <F ID=12 DE='Brightness'/>
</D>
```

d.

Figure 3.6: Device Profiles

Using this technique several bytes can be saved when storing profiles on a device

with limited memory.

3.8 Device Functions

Device operations are ways to interact with functions using MAP. For example, a light will have an On/Off output function (0x02) and this function can have one of two values, 0x00 for light begin OFF and the 0x01 for light being ON. When a MAP interactions with this one function on a device it need invoke an appropriate operation. Table 3.2 list all the functions that are available in this is framework and operations available on each of those functions.

All operation invokes are made using XML. For example, when MAP invokes "set" operation for a light it needs to tell the device the function it is trying to invoke this call using "ID" attribute and value using "Val" attribute. For a light source, this could be as simple as <sT id="01" val="01"/>. Table 3.3 lists all the operations that are available in this framework and its syntax.

Table 3.2: Function Operations

Code	Possible values	Operations	Description
0x01	0x00 - off 0x01 - on	Set Get Toggle	A device will use this as one of its functions when it has a button that is capable of two stage input, like an on/off switch.
0x02	0x00 - off 0x01 - on	Set Get Toggle	A device will use this as one of its functions when it has two stages of output. Usually this is On or Off.
0x03	0x01 - lowest setting 0x******** - up to 32bits	Set Get LevelUp LevelDown	A device will use this as one of its functions when an input with more than one level is needed. The lowest value is 0x01 and it should NOT turn the device off. Max value is set in device profile and can NOT be change at run time.
0x04	0x01 - lowest setting 0x******** - up to 32bits	Set Get LevelUp LevelDown	A device will use this as one of its functions when an output with more than one level is needed. The lowest value is 0x01 and it should NOT turn the device off. Max value is set in device profile and can NOT be change at run time.
0x05	0x00 - off 0x01 - lowest setting 0x******** - up to 32bits	Set Get LevelUp LevelDown	A device will use this as one of its functions when an input with more than one level is needed. The lowest value is 0x01. Setting it to level 0x00 will turn off the device.
0x06	0x00 - off 0x01 - lowest setting 0x******** - up to 32bits	Set Get LevelUp LevelDown	A device will use this as one of its functions when an output with more than one level is needed. The lowest value is 0x01. Setting it to level 0x00 will turn off the device.
0x07	Reading value, up to 20 bytes. Unit of Measurement, up to 20 bytes	SetReading	A device will use this as one of its functions when it requires a value from a sensor
0x08	Reading value, up to 20 bytes. Unit of Measurement, up to 20 bytes	GetReading	A device will use this as one of its functions when it provides a reading from a sensor

Table 3.3: Function Operation Calls

Name	Operation	Parameter(s)	Example
Get	GT	Function ID {int} Value {int}	<GT ID="01" />
Set	ST	Function ID {int} Value {int}	<ST ID="01" VAL="00" />
Toggle	TG	Function ID {int}	<TG ID="01" />
Level Up	LU	Function ID {int}	<LU ID="01" />
Level Down	LD	Function ID {int}	<LD ID="01" />
Get Reading	GR	Function ID {int}	<GR ID="01" />
Set Reading	SR	Function ID {int} Value {int}	<SR ID="01" VAL="35"/>

3.9 Device Events

Events are generated by the device when a value on any function is change using manually source, such as someone walking up to a switch and turning it on. When such an event occurs, device records the function, and its value, on which it was generate and transmits that value to MAP using simple XML string <evt id=”01” val=”01”/>. This information is used to invoke functions on another device by MAP.

Chapter 4

System Implementation

This chapter further details the implementation of modular architecture proposed in this thesis. As seen in Fig 3.3 there are several modules that operate with each other to form the system. Few of these modules, such as Decision Component and database, are crucial for system to operate, while modules such as web server and remote client provide nice to have functionality.

4.1 Main Automation Processor

Main automation processor (MAP) acts as the local gateway to the whole system with several important modules running under it. The hardware is based on ALIX 3C2 single board computer that features a 500 MHz x86 compatible chip, 256 MB ram, one Ethernet port for internet connectivity, a DB9 serial port which can be used as terminal port and two USB ports. A 2GB compact flash card was selected as a storage device. Small form factor of 100 x 160 mm makes it ideal and 3-5 watts power

consumption makes it efficient.

Voyage Linux, Debian Linux derived distribution for embedded platform, was selected as the main operating system. Several modules (binary executable), were configured to start up as a dameon to allow for virtually no direct interaction by the users. Flow chat presented in Fig 4.1 shows the start-up process of the operation system.

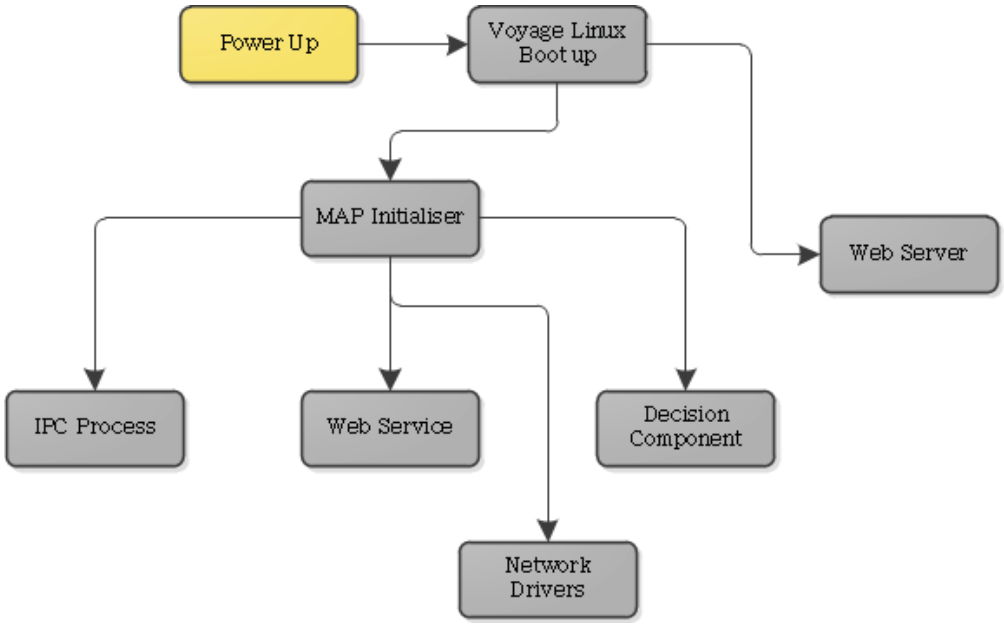

Figure 4.1: MAP Start-up Process

4.2 Zigbee Coordinator

A coordinator module is need for a Zigbee network and module directly interacts with Zigbee Network driver running on MAP. The hardware was custom built that can be plugged directly into one of ALIX 3C2's (MAP) USB ports. The coordinator

module is powered via USB and designed to operate at 3.3V.

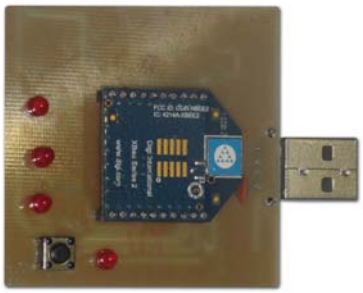

Figure 4.2: Zigbee Coordinator

The XBee module acting as the coordinator is configured to work in API mode to allow the flexibility of interacting with Network Driver module. The commission switch was also made available on the coordinator for making device join easy.

4.3 Zigbee Network Drive

The Zigbee network drive module is mainly responsible for translating information between MAP and Zigbee network. It also deals with forming the network and issuing appropriate commands to the coordinator module. It communicates with Decision Component module using Inter-process communication.

As mentioned in last section, XBee module was configured to work in API mode. This network driver is responsible for forming appropriate API packets. The driver also decodes any API packets issues by the XBee module.

Figure 4.3: Profile request XML string encoded as XBee packet

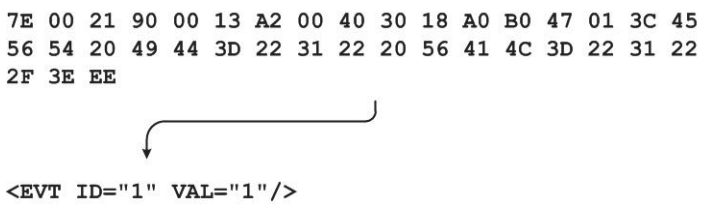

Figure 4.4: XBee packet, with device event information, to XML String

4.4 Inter-process communication (IPC)

Simple socket based IPC was written in C++ with sole purpose of allowing different modules, which run on MAP, communicate with each other. This provides the stability of not relying on different modules for a smoother operation. For example, if XBee coordinator module has been unplugged from MAP the Decision Component still operates as usual. Flowchart in Fig 4.5 shows how this module operates.

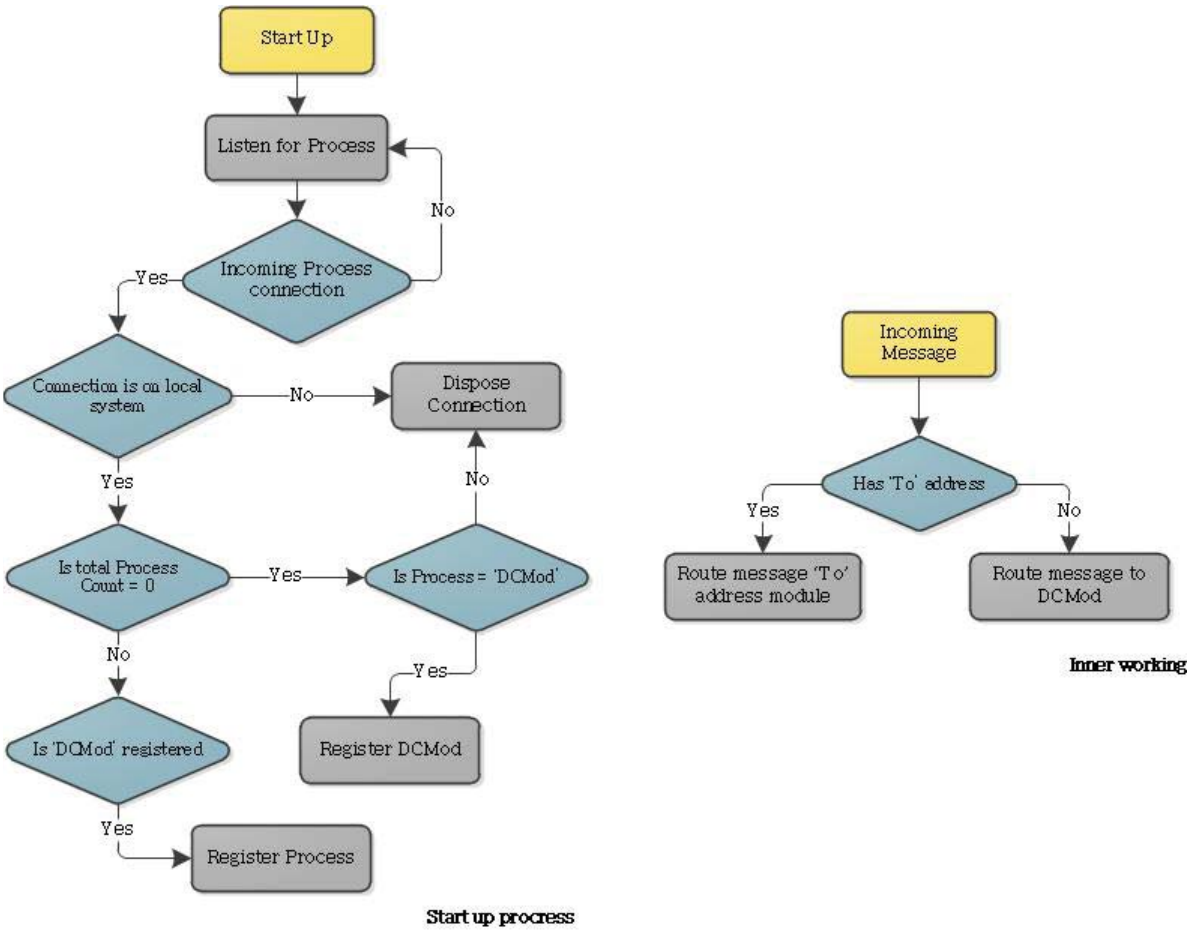

Figure 4.5: Inter-process communication

4.5 Decision Component

This module is considered to be the brain of the system, all the other modules communicate with decision component directly or indirectly. This module only communicates with other modules using XML suggested in Chapter 4. This allows it to make decisions at a very abstract level.

Decision component module is the only module that has direct access to database. This allows for a central management, reduces concurrent reads and behaves similarly

to any inputs.

4.5.1 Thread Operations

Decision Component has been written in C++ and is multi-threaded. Fig 4.6 show the start-up process of this module and threads it spawns for its smooth operation. Table 4.1 describes the operations of each thread.

Table 4.1: Decision Componet - Threads

Name	Description
Main thread	Main thread only operates at module start up. Its purpose is to allocate memory and spawn all the other thread. However, this thread wakes up every second to check if other threads are still operational. If one of the child threads, for unknown reason, fails to operate it re-spawns.
Database thread	This is "created" right after the main thread and all communications to the database are handled here.
Client thread	In order to communicate with other modules, using ICP module, client thread was created. All communication to and from Decision Component are handled by this thread. This includes all the calls made by web service and devices.
Timer thread	Decision component has a built in timer that "wakes-up" every 10 seconds to check for any time based events.

4.5.2 Events

Events in this system can be generated by any module and is handling them efficiently makes it very powerful.

Events, as a basic level, perform a task based on something that has already happened. For example, when a user hits a switch on the wall a device event is generated and its new status (On/Off) is transmitted to Decision Component. This

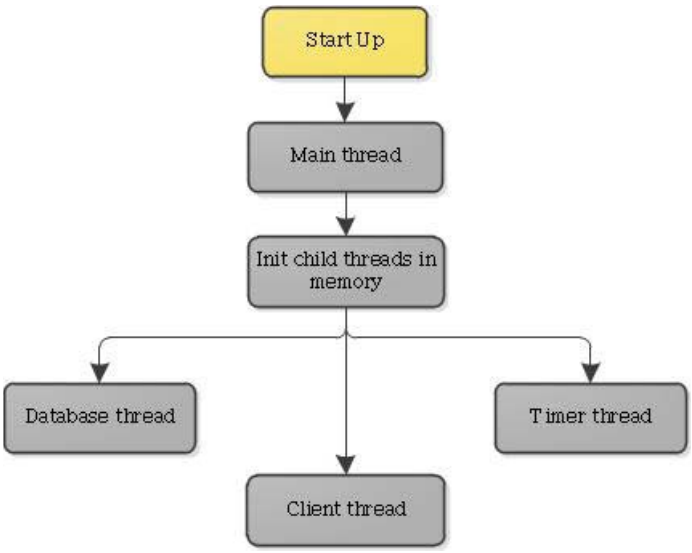

Figure 4.6: Decision Componet Start up

change is recognised and a database look up is performed to checks if anything else needs to be done, if a match is found a function operation is invoked and a light, located at the other end of a house, is switched on.

The other type of events is based on time and run by the timer thread. For example, if a user creates an event that turns a light on at 8.00 AM the information is stored in the database. The timer thread, once resumes from sleep, queries the database for events based on current time. If any events are found, the timer thread invokes them and goes back to sleep.

Even though the system does not allow users to assigned events at a second interval, timer thread still checks for events every 10 seconds. This makes the system respond to events within 10 seconds of allocated time.

Several events could be strung together to perform complex operations making this very powerful. Fig 4.7 shows one such possibilities.

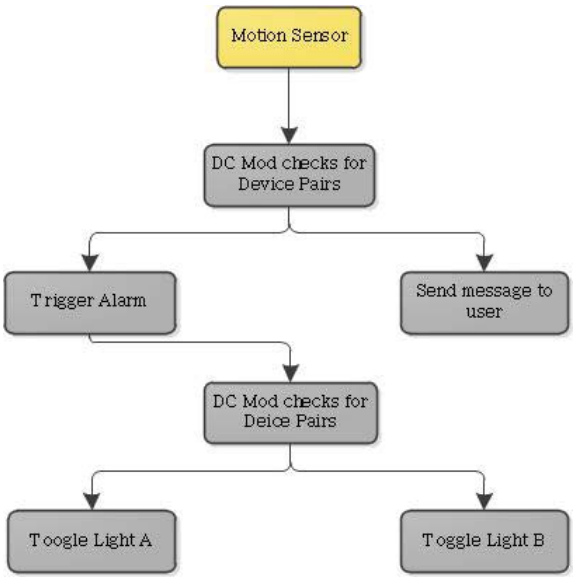

Figure 4.7: Motion sensor events

4.6 Database

As MAP has been designed on embedded system a fully operational database management system was considered to be an over-kill. Due to limited hardware resources SQLite was selected as it offered relatively small (~300 kB) library with excellent documentation. It uses a dynamically and weakly type SQL syntax and offers multitasking. The C/C++ API offered in this library is heavily used by Decision Component.

The database was carefully designed to accommodate several users, devices networks, devices, device functions and even custom events. System rules that defined the database structure are show in Table 4.2.

Fig 4.8 shows the entity relationship diagram based on system rules. Implementation of primary keys and foreign key are in place to reduce data redundancy and allow

for entity relationships. The Log entity on the diagram is solely used for storing historic information. Table 4.3 breaks on each entity into database table and describes it purpose with relation to system rules.

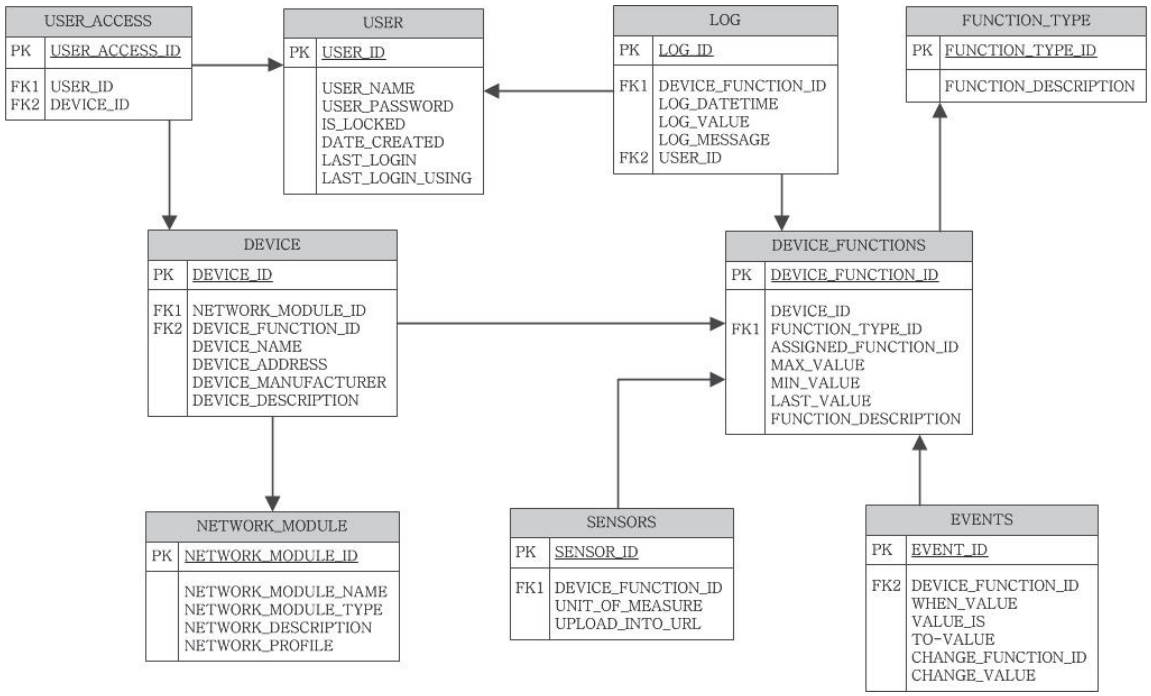

Figure 4.8: Entity relationship diagram

Table 4.3: Database tables and relationship to system rules

Entity	**Based on rule**	**Comments**
User	1, 2	Having dedicated table for users allows for multi users.
User_Access	1, 2, 3	This table has 1 to many relationships with User table. This allows customise device access to each user.
Device	1, 2, 3, 4	All the information regarding a device will be in here.
Network_Module	5, 6	Network information will be stored in here. This table has a 1 to many relationship Device table.
Device_Functions	7, 8	List of device functions will be stored in here. This table has 1 to many relationship with the device table
Function_Type	8, 9, 10	All the allowed functions types are stored in here. Using this table allows the flexibility of adding new functions at a later date.
Sensors	9, 10, 11, 12	In case of sensor type function, additional information will be stored in here. This allows to separate sensor related information into a different table and save memory when function is not of sensor type.
Events	7, 8, 13, 14	Information regarding function events and condition are stored here.
Log	15	Historic information is logged here.

Table 4.2: Database System Rules

Rule	Name	Description
1	Multi User	The system needs to be able to cater for more than one user.
2	User Access	The system needs to be able to restrict and allow access to devices based on users
3	User Device	Each user should be able to control one or more devices
4	Device User	Each device can be controlled by multiple users
5	Network Devices	Each network can have multiple devices
6	Device Network	Each device can only be in one network
7	Device Functions	Each Device can have multiple function
8	Function Device	Function is part of one device
9	Function Type	Each function is a can be only of one function type
10	Type Function	Each function type can be used by several functions
11	Function Sensor	If a function type is of sensor, one set of in additional data such as unit of measure is needed
12	Sensor Function	One set of sensor data is linked to one device function
13	Function Events	Function have multiple events
14	Event Function	Events can be functions from on same device or different. Event should NOT be triggered and operated on the same function
15	Logs	All information needs to be logged for historic reasons.

4.7 Web Service

SOAP based Web service was designed using open-source library gSoap [60]. The gSoap library provides an automated SOAP and XML data binding for C and C++ applications that simplifies the development of SOAP based Web Services. It supports WSDL 1.1, SOAP 1.1, SOAP 1.2 and SOAP RPC encoding style and document/literal style. gSoap also provides WS-Security, WS-Addressing with support for SSL (HTTPS) using OpenSSL.

A standalone executable was created to act as the web service server. Even though gSoap allows to CGI type mechanism, a standalone method offered multi-threaded capability as it ran just as any other process.

The service was configured to run on port 9999 with document style and literal encoding for maximum compatible. It also has the ability to respond to WSDL and clientaccesspolicy.xml request for web based technologies such as Silverlight and Flash.

4.7.1 Security

gSoap offers several security options for Web Service implementations. OpenSSL was used to implement secure communication between the server and the program consuming it. This offers a certain level of protection to middle-man attacks [60].

Another application level security, using session keys, was added to further secure the service. User trying to access the web server needs to be authenticated using a user name and password. This information is passed on to web service using the web method login. On successful authentication, the Login web method returns a unique

256 bit hex string known as session key. This session key is used for every other web method call made. The session key expires after 5 minutes of inactivity for security reasons. Fig 4.9 shows the flowchart of this process.

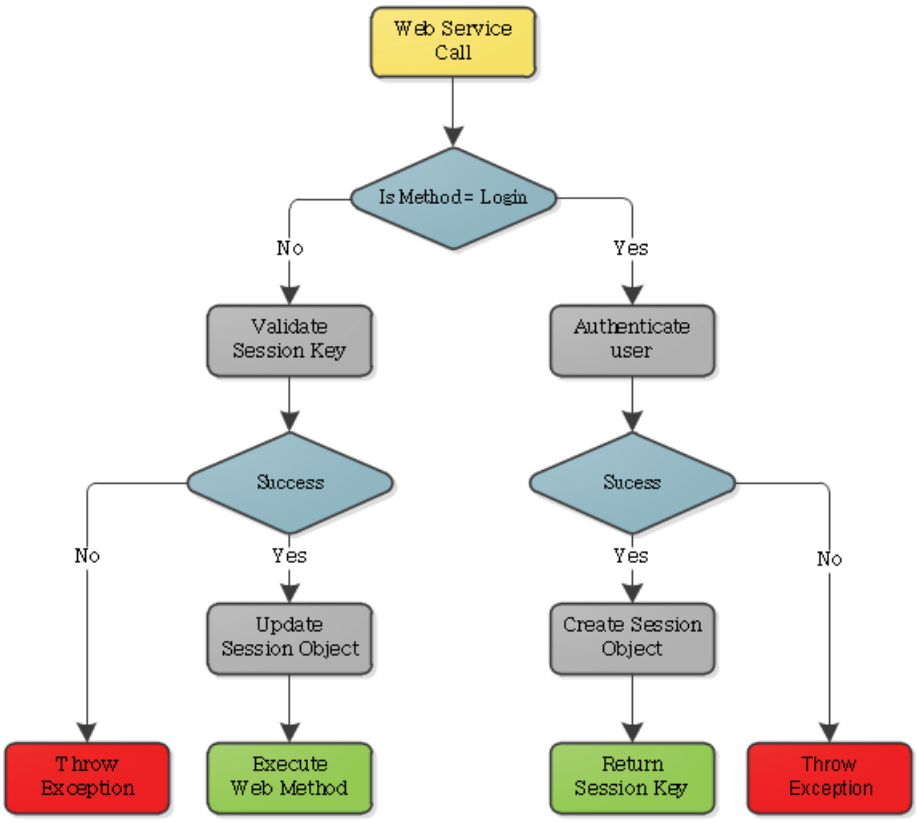

Figure 4.9: Web Service session management

4.7.2 Web Methods

Programs consuming the web service are presented with several web methods as listed in Table 4.4.

Table 4.4: Web Service Methods

Method	Parameter(s)	Return Type
Login	username {string} password {string}	authReponse {AuthResponse}
Logout	sessionKey {string}	result {boolean}
GetNetwork	networkId {int} sessionKey {string}	networkResponse {Network}
GetNetworks	sessionKey {string}	networks[] {Network}
GetDevice	deviceId {int} sessionKey {int}	device {Device}
GetDevices	networkId {int} sessionKey {string}	devices[] {Device}
GetFunction	functionId {int} sessionKey {string}	function {Function}
GetFunctions	deviceId {int} sessionKey {string}	functions[] {Function}
GetEvent	eventId {int} sessionKey {string}	event {Event}
GetDeviceEvents	deviceId {int} sessionKey {string}	events[] {Event}
GetFunctionEvents	functionId {int} sessionKey {string}	events[] {Event}
AddEvent	event {Event} sessionKey {string}	result {boolean}
RemoveEvent	eventId {int} sessionKey {string}	result {boolean}
UpdateEvent	event {Event} sessionKey {string}	result {boolean}
GetUser	userId {int} sessionKey {string}	user {User}
GetUsers	sessionKey {string}	users[] {User}
AddUser	user {User} sessionKey {string}	result {boolean}
UpdateUser	user {User} sessionKey {string}	result {boolean}
DeleteUser	userId {int} sessionKey {int}	result {boolean}
InvokeOperation	operation {Operation} sessionKey {string}	result {boolean}

4.8 Web Server

The web server is used to host a simple collection of web pages that provide an interface to control the system. Due to limited resource available on the system a light weight web server named Lighttpd (pronounced lighty) was used [61].

Lighttpd is an open-source web server that is optimised for speed-critical environments without compromising security and flexibility. Lighttpd offers a very small memory footprint and small CPU load when compared to other web servers and makes a good choice for embedded applications. It also supports SSL and TLS using OpenSSL library.

4.8.1 PHP

As any good open-source web server, Lighttpd offers support for PHP using FastCGI. PHP is an open-source scripting language designed for web development in order to produce dynamic webpages. Newer versions of PHP support SOAP based web services. PHP was used to interact with the gSoap web service and produce HTML documents that users can interact with.

4.9 Remote Server

This module has a two components 1) simple user interface that will allow them to login 2) web service that is consumed by Remote client.

The user interface component was developed using ASP.Net and hosted on IIS. The main purpose of this component is to provide an interface to users wanting to

access their home automation network over the internet. Fig 4.10 shows the login page.

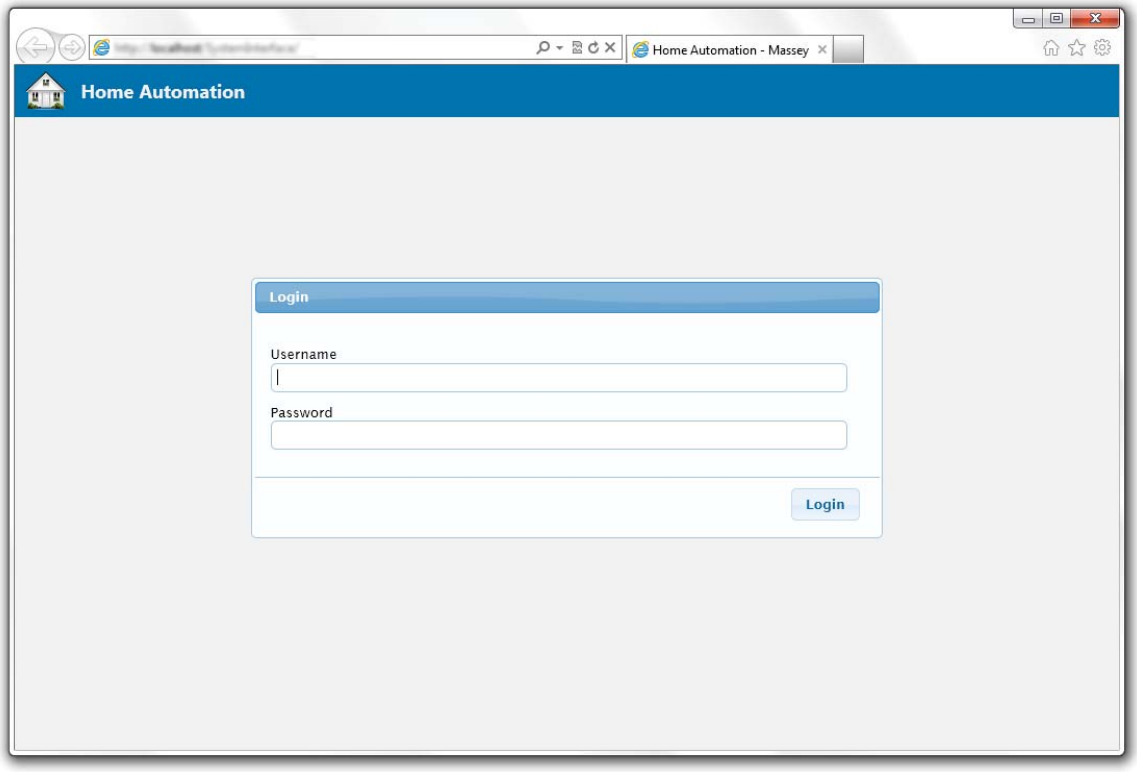

Figure 4.10: Login Page

Web service on this module, for simplicity sake, was developed using Microsoft's Windows Communication Foundation and Microsoft SQL Server database. The service has only one method, RegisterMap which is consumed by Remote Client module. Database has one table MapMatches which holds the information on different MAPs and its last known IP address. Fig 4.11 shows the process on how a user would take advantage of this module.

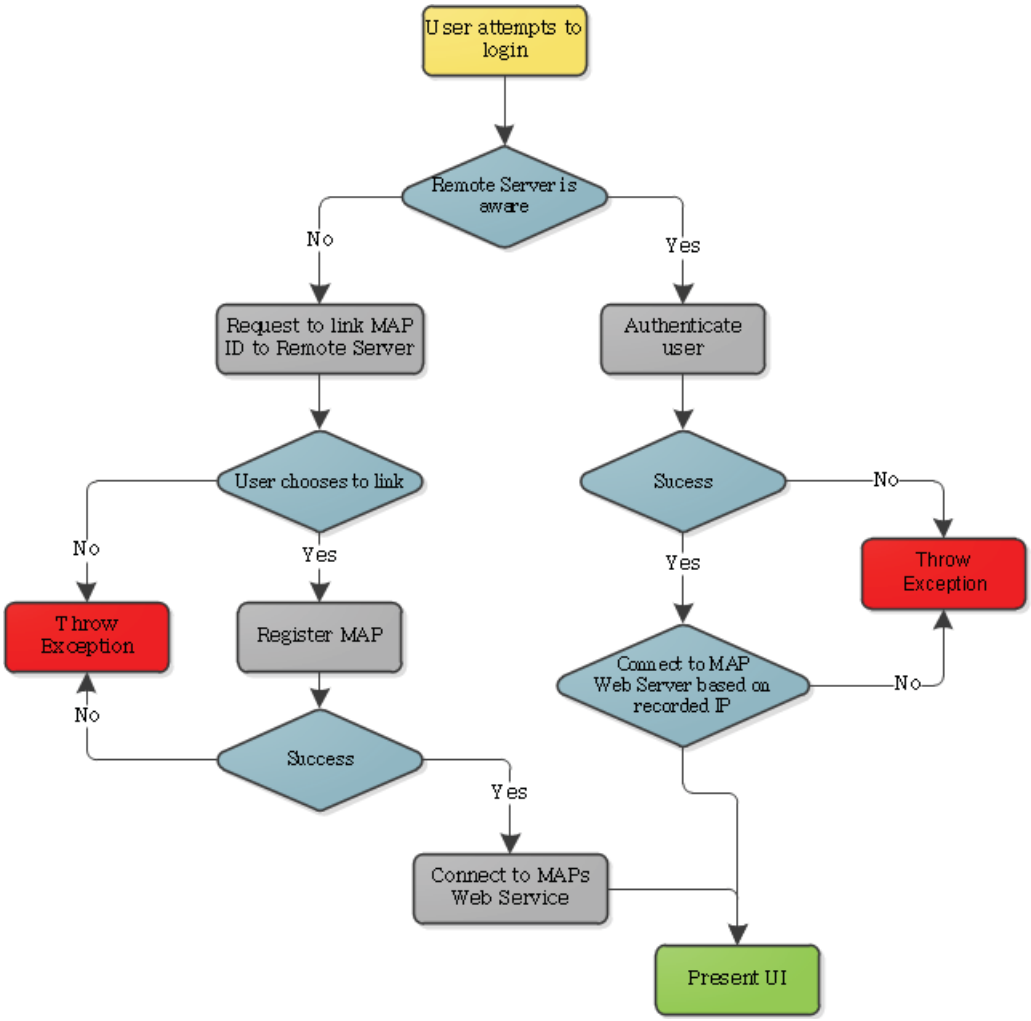

Figure 4.11: Remote Server process

4.10 Remote Client

Remote client is a simple Web Service client that connects to Remote Server module. Its only purpose is to inform Remote server about its current ip address and the port on which web service module is operating on.

This module is written in C++ using gSoap library as it also offers functionality to build web service clients. It calls the RegisterMap web method hosted by the Remote Server module. Fig 4.12 shows how this module functions.

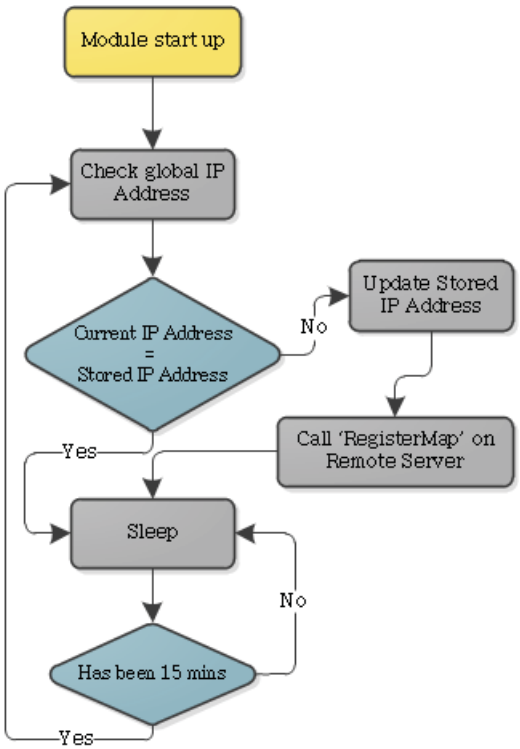

Figure 4.12: Remote Client process

4.11 On-board Interface

The on-board interface is a simple web site that is hosted on Web Server. This module has been written using PHP for server side scripting that consumes the Web Service hosted on MAP.

The interface itself was designed using basic XHTML, CSS and JavaScript. jQuery, a JavaScript library, was also used to proved some modern web interface components such as tool-tips, modal dialog boxes and styles. Figures 4.13 & 4.14 describe the interface in detail.

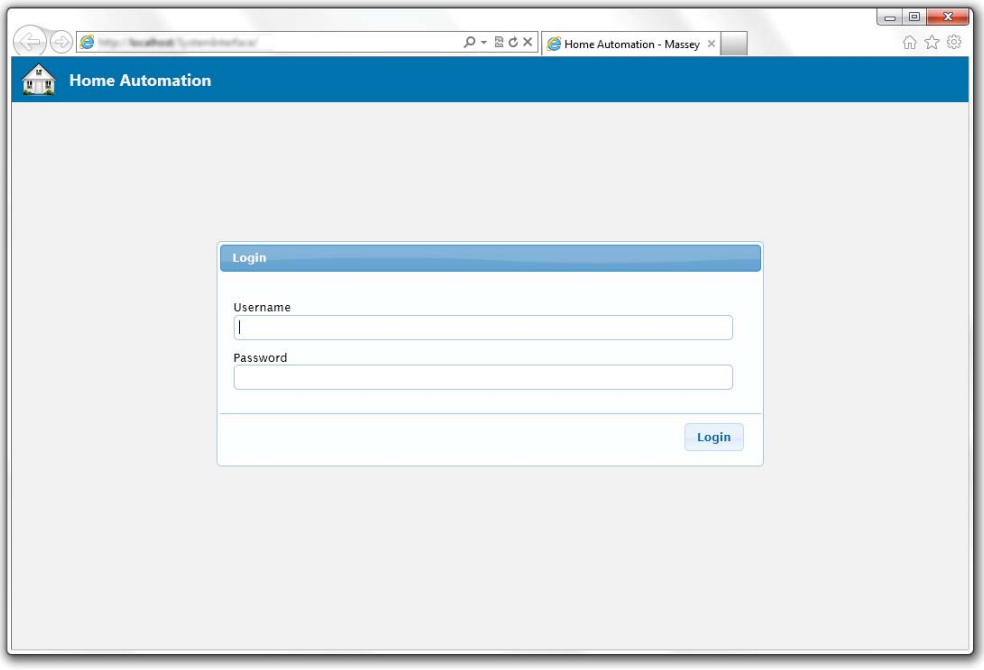

Figure 4.13: On board Login Page

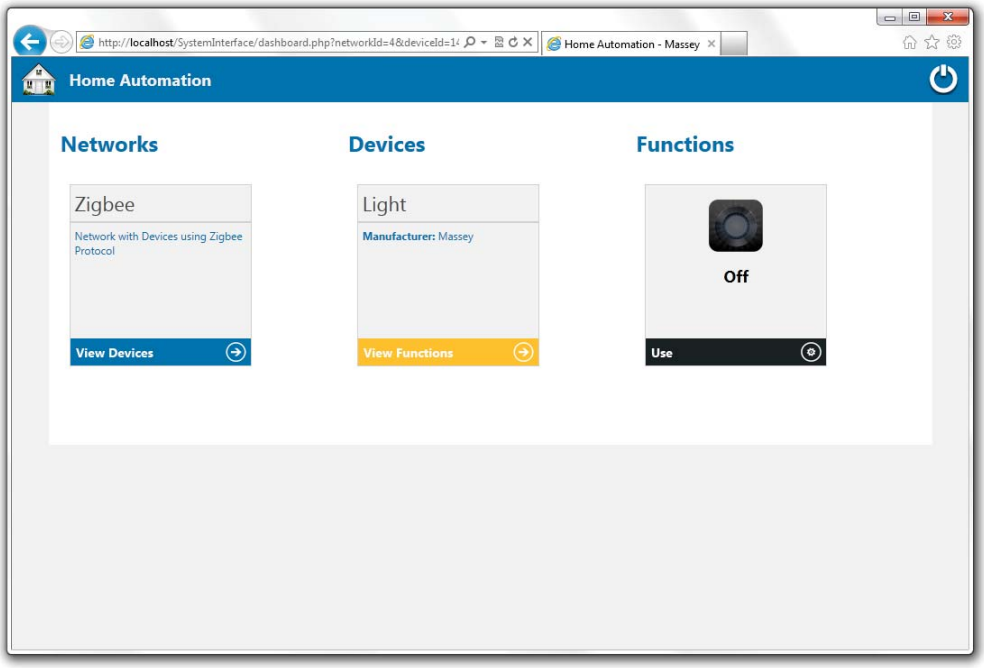

Figure 4.14: On board Network, Device & Functions

4.12 Central Panel

Central Panel is combination of hardware and software that can be placed at a 'central' location and used to control devices in home. For simplicity, a laptop was used run the required software and interfaced to a touch screen monitor.

The user interface was developed using Windows Presentation Foundation (WPF). This module interacts with MAP using Web Service.

Figure 4.15: Central Panel

4.13 Devices

Few devices were designed and built in order to demonstrate the concept presented in this thesis. All the devices are based on Zigbee protocol and working on XBee modules interfaced with PIC18F1220 microcontroller. Fig 4.16 shows the test board used to place all the devices.

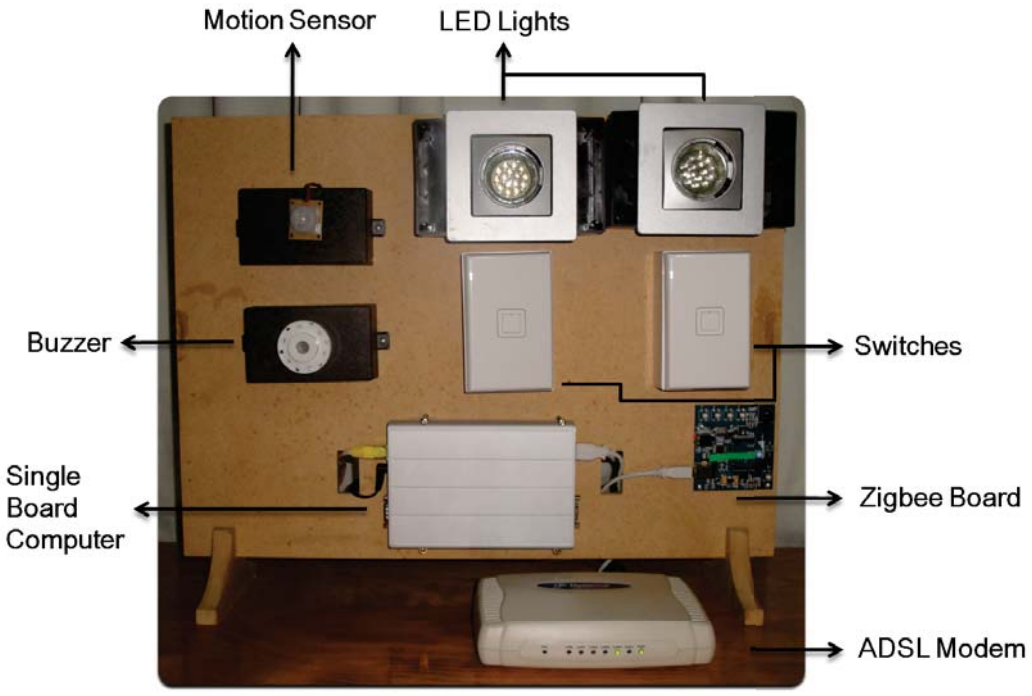

Figure 4.16: Test board

Using same PIC18F1220 and XBee modules for all devices made it easier to maintain the same code based with minor changes. For example, all devices are required to have a commission button, implement commission process and ability to parse application specific XML strings.

Software was written in C language using Microchip's C18 complier. Both interrupts and polling methods were used to ensure devices always respond to events and communications. Fig 4.17 shows the basic flow-chart on how devices operate.

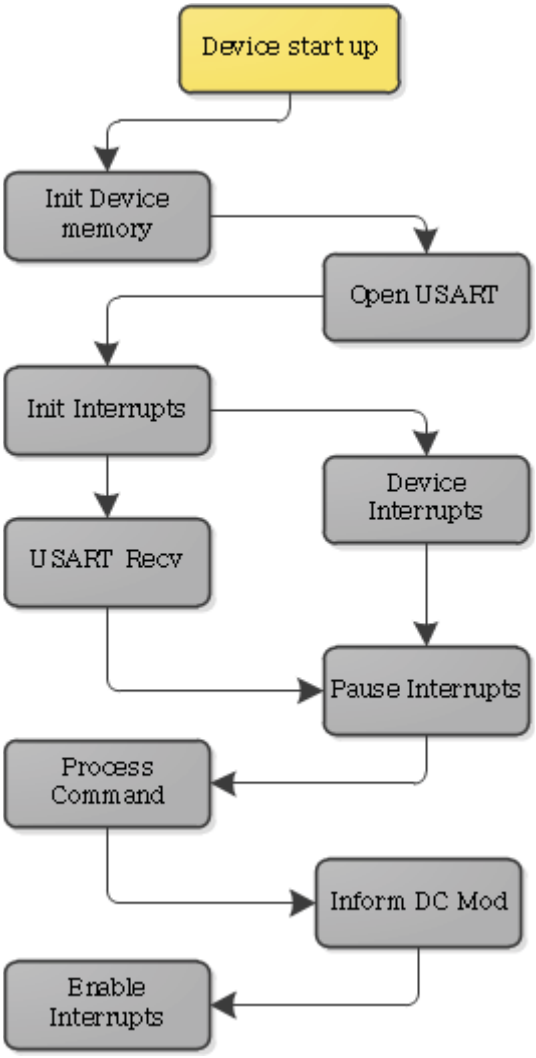

Figure 4.17: Device Operation

4.13.1 Switch

A simple switch was designed with intent to look and feel like regular switch. A mounting box Fig 4.18, with 73mm x 58mm dimensions, was selected as it had same form-factor as regular switch. It also had 36mm space between the wall and switch place which was used accommodate the circuitry required for PIC18F1220 and XBee

module. Fig 4.18 shows the profile used for this switch.

```
<P>
        <DE>Switch</DE>
        <MF>Massey</MF>
        <F TY="01" DE="Switch"/>
</P>
```

Figure 4.18: Switch Profile

4.13.2 Light

With the intent to keep devices manageable a 12V DC LED light was used to design a light that can interface with the system. Fig 4.19 shows the profile used for this device and Fig 4.20 shows the device PCB Layout.

As mentioned early, the LED light selected to build this device runs at 12V DC while PIC18F1220 functions at 5V DC. In order to safely interface and control the light LM317 voltage regulator was used for PIC18F1220 and simple transistor was set to operate as a switch.

```
<P>
        <DE>Light</DE>
        <MF>Massey</MF>
        <F TY="02" DE="Light"/>
</P>
```

Figure 4.19: Light Profile

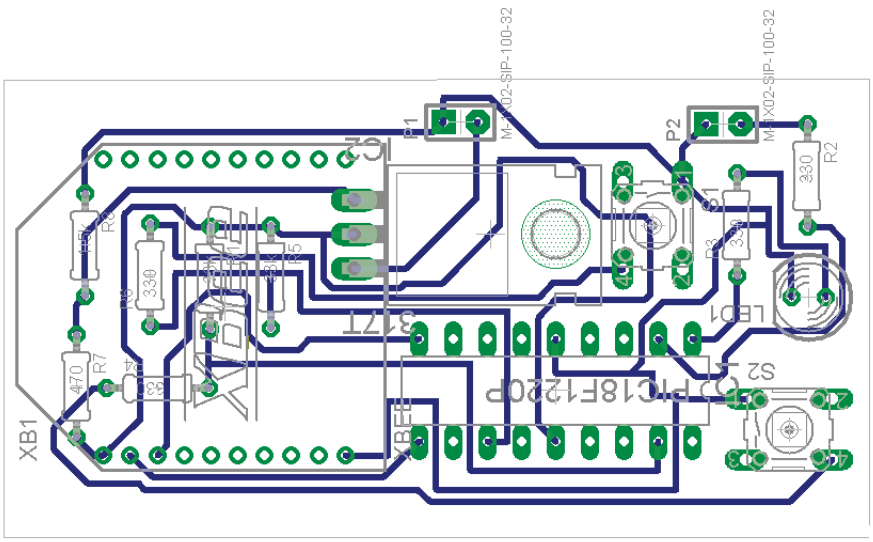

Figure 4.20: Light PCB

4.13.3 Alarm

Alarm and Light share exactly the same code base with expectation to device profile as shown in Fig 4.21. The hardware was also nearly identical as a Buzzer was used instead for a light source.

```
<P>
        <DE>Alarm</DE>
        <MF>Massey</MF>
        <F TY="02" DE="Alarm"/>
</P>
```

Figure 4.21: Alarm Profile

4.13.4 Motion Sensor

A simple PIR motion sensor was integrated with PIC18F1220 and XBee module. It is a device that generates an event when it detects motion. Fig 4.22 shows the

profile used.

```
<P>
        <DE>Motion Sensor</DE>
        <MF>Massey</MF>
        <F TY="02" DE="Motion Sensor"/>
</P>
```

Figure 4.22: Motion Sensor Profile

4.14 Device Pairing

Pairing is the process of tying functions from one device to functions of other devices. As this system aims to support multiple networking protocols, a common pairing process was established using database and device profiles. Flow-chart show in Fig 4.23 details this process.

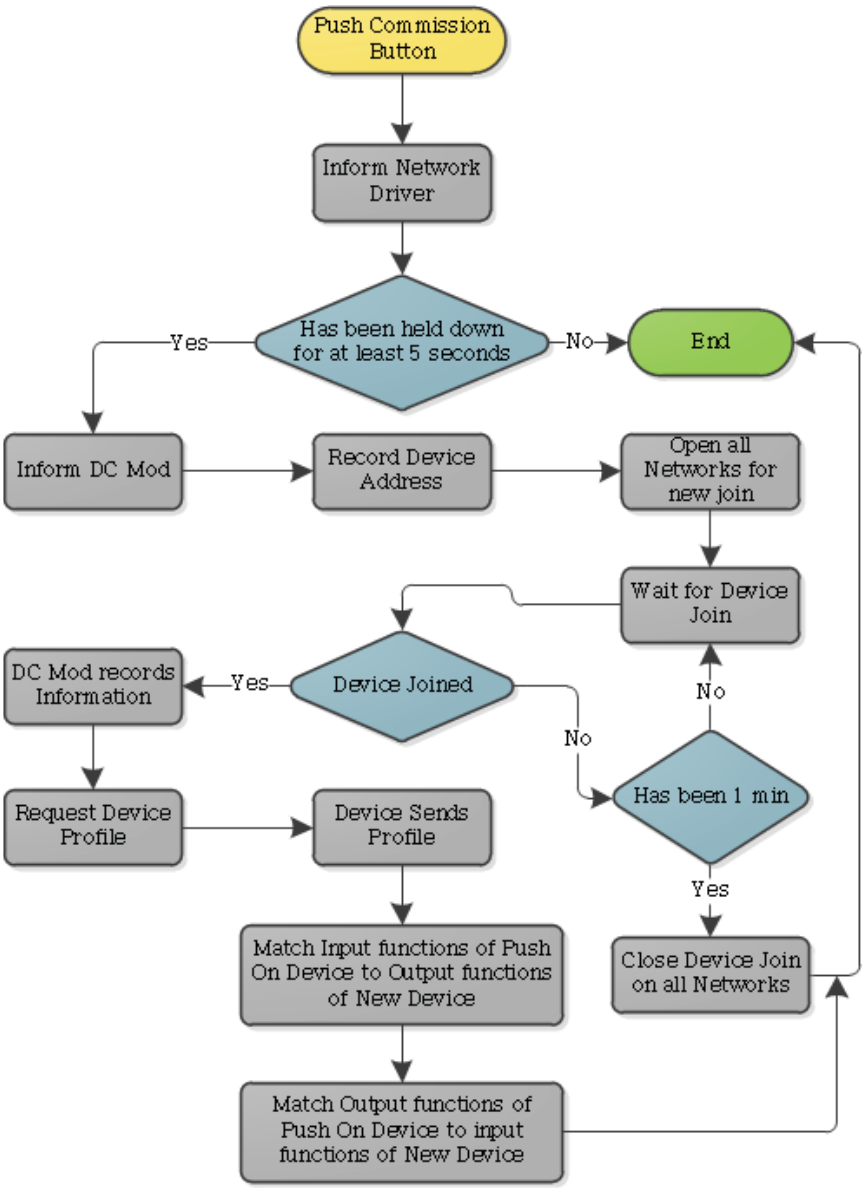

Figure 4.23: Device Pairing

Chapter 5

System Evaluation

The system was tested against several scenarios which looked at stability, usability and performance. Due to the difficulty of measuring very short period between issuing a command and confirming if it was done, a simple Class name Performance was written.

Tables 5.1 and 5.2 shows the properties and methods that make up this class. Objects Performance class were created at beginning of MAP execution to ensure that time taken to for object creation is not included. The excetuded time was displayed out on console window.

Table 5.1: Performance Class - Properties

Property Name	**Type**	**Description**
StartTime	clock_t	Records the time before command is issued.
EndTime	clock_t	Records the time after response has been received
Measuring	string	This meta-data variable is used for reference propose. Used when out putting the elapsed time.

Table 5.2: Performance Class - Methods

Method Name	Arguments	Return	Description
Performance	Measuring {string}	void	This is Overriding Constructors that accepts a string argument. The string argument is stored to private property "Measuring" which is used for logging.
Start	N/A	boolean	This method use before starting a job that needs to be timed. Current time it recorded and stored in private variable "StartTime". Returns true on a successful execution and false for unsuccessful execution.
End	N/A	boolean	This method records the current time and stores it into local private variable "EndTime". It then calculates the elapsed time by subtracting "StartTime" from "EndTime". Finally it outputs a message to console window based on "Measuring". *E.g. Elapsed time for local switching: 300ms.* Method returns true on a successful execution and false for unsuccessful execution.
Clear	N/A	boolean	This method sets "StartTime" and "EndTime" to 0. Sets "Measuring" to "Unknown". Method returns true on a successful execution and false for unsuccessful execution.

5.1 Using Physical Control

As discussed in previous chapter switches were designed through with users can interact directly. In this test case, the system was setup and each switch was paired up with one other light. As seen in Fig 5.1 switch 1 has been paired with light A and switch 2 has been paired with light B.

Users who never used this system were expected to turn the lights on/off five

times using the appropriate switch. Time between each push varied and delay was recorded between button push and light turning on was also recorded as shown in the table 5.3.

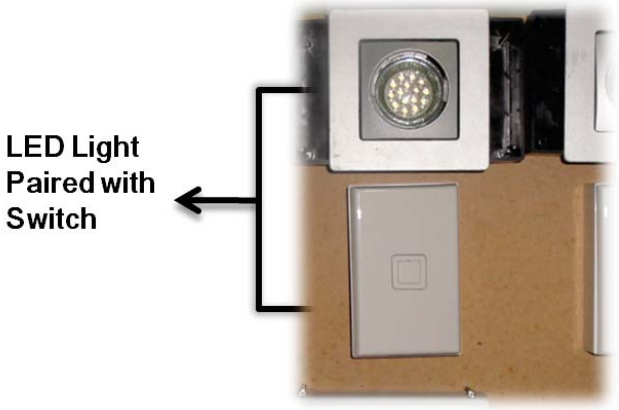

Figure 5.1: Light paired with switch

Table 5.3: Physical Contorl - Results

	Light A	Light B
Lights Switched On/Off 5 times (average)	560ms	520ms

5.2 Using Local Control

System was also testing by using the local web interface where user was asked to use the web interface, hosted on single board computer, and central touch control, that is communicating with web service hosted on single board computer. The purpose of this test was as ensure that users are able to access and use the system within their local home network without requiring internet access.

Similar test of turning lights on and off were performed several times. Successfully executions were recorded, in table 5.4, along with the delay between each execution.

Table 5.4: Local Control Results (Average of 5 trials)

	Light A	Light B
Local Web Interface - Hosted on MAP	600 ms	580 ms
Central touch control.	780 ms	720 ms

5.3 Using Remote Control

The system was also tested over the internet by asking the user to access their system from both a different location (using a different global IP address) and from the same location (using the same IP address). Interface used the buy the user "appeared" to be identical to that of 'local control' however, the data is accessed over the internet and routed thought addition layers.

Similar test involved turning lights on and off were performed several times and successful executions and time delays were recorded as shown in table 5.6.

Table 5.5: Remote Web Interface (Average of 20 trials)

	Light A	Light B
Local IP Address - Sharing same global IP Address with MAP	740 ms	770 ms
Remote IP Address - Accessing MAP from differnt location	980 ms	1200 ms

5.4 Events

As mentioned in chapter 3, the system was designed to handle any user assigned events such as time based and event based. Events are pre-assigned task that system triggers when some other tasks in completed. The system checks for any time events every 10 seconds, this may impose a maximum of 10 second delay. However, time based events are assigned by the minute and should have very low impact on this 10 second delay.

Functions events were tested by performing some time based actions and events based on device functions.

5.4.1 Time events

Lights, A and B, were set to turn toggle at a certain times and tested to see how much of a delay occurred between the time set and the actual time the devices toggled. The performance class was modified so that the start time was the time set and end time was when the event actually finishes. The results are shown in table 5.6.

Table 5.6: Time events - Each test is average of 5 trials

	Light A	Light B
Test 1	8 sec	1sec
Test 2	3 sec	9 sec
Test 3	6 sec	8 sec
Test 4	2 sec	5 sec
Test 5	8 sec	3 sec

5.4.2 Device function events

Device function events are events that take outcome of an event and trigger a new event. The system handles these events almost the same way it handles paired devices. The only difference here is that several device functions or even devices can get triggered.

The test-bed presented in previous chapter contains a motion sensor, which was used to generate the event, and buzzer which was used to output the triggered event. These two devices were paired up in a way that when the motion sensor detects a change the buzzer will sounds an alarm and delay is shown in the table 5.7.

Table 5.7: Motion sensor - Each test is average of 5 trials

	Buzzer Delay
Test 1	1200 ms
Test 2	980 ms
Test 3	600 ms
Test 4	780 ms
Test 5	930 ms

5.5 Overall

The overall responsiveness of the system appears to be expected and fall within acceptable level. In table 5.4 delay between calls made by web server to the web service are both physically running on MAP providing the fasting response times. The delay was slightly high when interacting using the touch interface. This is expected, and acceptable, overhead as the interface itself runs on a different machine and use .net framework.

The remote control has even further overheard especially when accessed from different location, as shown in table 5.5. This accounts for network latency and internet speeds.

Device function events occur when functions between devices are paired. The delay was hard to measure as it was difficult to figure out if the PIR motion sensor actually detected any motion. However, ~1 second delay was considered to acceptable. The highest delay was experienced when events were set to trigger based on time. This is also expected as the timer thread 'wakes-up' every 10 seconds and issues any events.

Chapter 6

Conclusion and Future Work

6.1 Summary

The idea of home automation is to improve every-day quality of life. However, it is not widely adopted in our modern time. Implementation of such system needs to offer home owners with maximum improvement in quality of life without sacrificing their personal privacy or ways to control their home environment as they desire.

This thesis presented a low-cost modular home automation system that uses XML instead of depending on a specific networking protocol. It supported complex event handling and offered three levels of interfacing, including internet access. It proposes a neutral way of communication that can controlled and managed at a central location which does not require home owners to buy expensive hardware.

The use of web services offers a way for other developers to design hardware and software that further improvers the usability of the system across many platforms.

6.2 Future Work

System presented in this these can profit from several improvements that make it more secure, reliable, stable and user friendly. This section discusses such potential improvements.

6.2.1 Device Networks

The Zigbee protocol, using XBee modules, was used in order to demonstrate the system. However, the system is capable of interfacing with other networking protocols. With given time and resources, a network of 6lowPan or Bluetooth could be developed. Using high-power standards, such as 802.11, could provide a way to interact with multi-media equipment.

6.2.2 Web Service

The gSoap library offered several feature that made publishing web service great. However, due to custom multi-threaded implementation little time was allocated for testing. While web methods hold up extremely well against concurrent calls the module could have used more testing time.

6.2.3 Central Panel

This module was indented to be designed and developed using Windows 7 touch hardware. However, due to several reasons it was just a WPF application on an old POS monitor. Several improvements, such as using embedded windows 7 and

Silverlight interface could have been developed that provided users with a friendly interface.

6.2.4 Database

Currently all the information in the database is stored in clear-text format. This information needs to be encrypted in order protect sensitive information in case of theft, break in or hacking.

6.2.5 Power

Currently, all the devices are powered regulated 5v or 3.3v using a 12v DC source. This decision was made due to safety concerns. However, for devices to operation in ever day home they need to be power direct from mains in an efficient manner and also pass local electric standards.

The overall system holds up reasonably well despite its hardware limitations and resources, however further testing needs to be done to ensure safety and security of information.

Bibliography

[1] A. Al-Ali and M. AL-Rousan, "Java-based home automation system," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 498–504, May 2004.

[2] U. Bischoff, V. Sundramoorthy, and G. Kortuem, "Programming the smart home," *3rd IET International Conference on Intelligent Environments (IE 07)*, pp. 544–551, 2007.

[3] Y. Dahl, "Redefining smartness: the smart home as an interactional problem," *4th International Conference on Intelligent Environments (IE 08)*, pp. 4C3–4C3, 2008.

[4] K. Gill, S.-H. Yang, F. Yao, and X. Lu, "A zigbee-based home automation system," *IEEE Transactions on Consumer Electronics*, vol. 55, pp. 422–430, May 2009.

[5] T. Perumal, A. R. Ramli, C. Y. Leong, and S. Mansor, "Interoperability for Smart Home Environment Using Web Services," *International Journal*, vol. 2, no. 4, pp. 1–16, 2008.

[6] D. Tudor, A. Stancovici, B. Popescu, and V. Cretu, "Zombee: a home automation prototype for retrofitted environments," *Environments*, 2009.

[7] S. Ok and H. Park, "Implementation of initial provisioning function for home gateway based," *Communication Technology, 2006. ICACT 2006. The*, pp. 20–23, 2006.

[8] V. Trifa, S. Wieland, D. Guinard, and T. Bohnert, "Design and Implementation of a Gateway for Web-based Interaction and Management of Embedded Devices," pp. 1–14.

[9] X. Chen and W. Huang, "The Design of Intelligent Home Alarming System Based on Wireless Sensor Network Technology," *Computer and Information Science*, vol. 2, no. 1, p. P211, 2009.

[10] O. A. Taiwo, "Wireless Short Range Communication Technologies for Home Automation Wireless Short Range Communication Technologies for Home," *Electrical Engineering*, no. June, 2008.

[11] C. Gomez and J. Paradells, "Wireless Home Automation Networks : A Survey of Architectures and Technologies," *IEEE Communications Magazine*, no. June, pp. 92–101, 2010.

[12] B. Benatallah and F. Casati, "An Overview of Standards and Related Technology," *Discover*, pp. 135–162, 2002.

[13] R. Nunes and J. Delgado, "An architecture for a home automation system," *1998 IEEE International Conference on Electronics, Circuits and Systems. Surfing the Waves of Science and Technology (Cat. No.98EX196)*, pp. 259–262.

[14] Y.-m. Hsieh and Y.-c. Hung, "A scalable IT infrastructure for automated monitoring systems based on the distributed computing technique using simple object access protocol Web-services," *Automation in Construction*, vol. 18, no. 4, pp. 424–433, 2009.

[15] A. Ziya, A. Member, J. Roach, and D. Baysal, "IP Based Home Automation System," *Electronics*, pp. 2201–2207, 2010.

[16] M. Eisenhauer, P. Rosengren, and P. Antolin, "A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems," *2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, vol. 00, pp. 1–3, June 2009.

[17] M. Osipov, "Home automation with ZigBee," *Next Generation Teletraffic and Wired/Wireless*, pp. 263–270, 2008.

[18] S. Gomasa and F. Alam, "Wireless sensor networks over internet protocol and web services," *International Journal of Intelligent Systems Technologies and Applications*, vol. 9, no. 2, pp. 108–120, 2010.

[19] L. Cheng, Y. Zhang, T. Lin, and Q. Ye, "Integration of wireless sensor networks, wireless local area networks and the Internet," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, vol. 1, pp. 462–467, IEEE, 2004.

[20] L. Srivastava, "Wireless innovation for smart independent living," *Science & Medicine*, vol. 2009, no. June, 2009.

[21] F. Ciancetta, B. D'Apice, D. Gallo, and C. Landi, "Architecture for distributed monitoring based on smart sensor and web service," in *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE*, no. April, pp. 2054–2059, IEEE, 2006.

[22] F. Huang, Y. Yang, and L. He, "A flow-based network monitoring framework for wireless mesh networks," *Wireless Communications, IEEE*, vol. 14, no. 5, pp. 48–55, 2007.

[23] S. Bouckaert, E. De Poorter, P. De Mil, I. Moerman, and P. Demeester, "Interconnecting Wireless Sensor and Wireless Mesh Networks: Challenges and Strategies," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pp. 1–7, Ieee, Nov. 2009.

[24] S. Xu, "Advances in WLAN QoS for 802.11: an overview," in *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, vol. 3, pp. 2297–2301, IEEE, 2003.

[25] C. Christodoulou, R. Jordan, and C. T. Abdallah, "Wireless Communications and Networking : An Overview,"

[26] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. Costa, and B. Walke, "The IEEE 802.11 universe," *Communications Magazine, IEEE*, vol. 48, no. 1, pp. 62–70, 2010.

[27] S. Saliga, "An introduction to IEEE 802.11 wireless LANs," in *Radio Frequency Integrated Circuits (RFIC) Symposium, 2000. Digest of Papers. 2000 IEEE*, pp. 11–14, IEEE, 2000.

[28] M. Umberger, I. Humar, A. Kos, J. Guna, A. Žemva, and J. Bešter, "The integration of home-automation and IPTV system and services," *Computer Standards & Interfaces*, vol. 31, pp. 675–684, June 2009.

[29] S. Beak, S. Choi, S. Rhee, S. Han, and T. Jeong, "Energy Charging Circuits in Bluetooth Environment for Smart Phone," *Engineering*, pp. 109–110, 2011.

[30] C. Bisdikian and Others, "An overview of the Bluetooth wireless technology," *IEEE COMMUN MAG*, vol. 39, no. 12, pp. 86–94, 2001.

[31] N. Salman and I. Rasool, "Overview of the IEEE 802.15. 4 standards family for Low Rate Wireless Personal Area Networks," *Wireless Communication Systems (*, pp. 701–705, 2010.

[32] J. Zheng and M. Lee, "Will IEEE 802.15. 4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard," *Communications Magazine, IEEE*, vol. 42, no. 6, pp. 140–146, 2004.

[33] P. Duan and H. Li, "Zigbee wireless sensor network based multi-agent architecture in intelligent inhabited environments," in *Intelligent Environments, 2008 IET 4th International Conference on*, pp. 1–6, 2008.

[34] Y. Ha, "Dynamic Integration of Zigbee Devices into Residential Gateways for Ubiquitous Home Services," *Ubiquitous Intelligence and Computing*, pp. 221–235, 2009.

[35] K. Hua, R. Peng, and G. Hamza-Lup, "Dissemination of sensor data over the internet," *Embedded and Ubiquitous Computing*, pp. 209–216, 2004.

[36] H. Dai and R. Han, "Unifying micro sensor networks with the Internet via overlay networking [wireless networks]," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pp. 571–572, IEEE, 2004.

[37] M. Gaynor, S. Moulton, M. Welsh, E. LaCombe, A. Rowan, and J. Wynne, "Integrating wireless sensor networks with the grid," *IEEE Internet Computing*, vol. 8, pp. 32–39, July 2004.

[38] P. Ferrari, a. Flammini, D. Marioli, E. Sisinni, and a. Taroni, "A Bluetooth-Based Sensor Network With Web Interface," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, pp. 2359–2363, Dec. 2005.

[39] K.-i. Hwang, J. In, N. Park, and D.-s. Eom, "A design and implementation of wireless sensor gateway for efficient querying and managing through world wide web," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 1090–1097, Nov. 2003.

[40] A. Ziya, A. Member, and U. Buhur, "An Internet Based Wireless Home Automation System for Multifunctional Devices," *System*, pp. 1169–1174, 2005.

[41] R. Silva, J. Silva, and F. Boavida, "Evaluating 6lowPAN implementations in WSNs," *Proceedings of 9th Conferncia sobre Redes de Computadores Oeiras, Portugal*, pp. 1–5, 2009.

[42] K. Mayer and W. Fritsche, "IP-enabled wireless sensor networks and their integration into the internet," in *Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*, no. chapter VI, p. 5, ACM, 2006.

[43] I. Zualkernan, A. Al-Ali, M. Jabbar, I. Zabalawi, and A. Wasfy, "InfoPods: Zigbee-based remote information monitoring devices for smart-homes," *Consumer Electronics, IEEE Transactions on*, vol. 55, pp. 1221–1226, Aug. 2009.

[44] D. Miao, K. Xin, Y. Wu, W. Xu, and J. Chen, "Design and implementation of a wireless automatic meter reading system," in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, vol. I, (New York, New York, USA), pp. 1345–1349, ACM, 2009.

[45] B. Garc\'\ia, I. Ruiz, J. Vicente, and A. Méndez, "BIOHOME: A House Designed for Assisted Living," *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, pp. 671–674, 2009.

[46] A. Rezgui and M. Eltoweissy, "Service-oriented sensor-actuator networks: Promises, challenges, and the road ahead," *Computer Communications*, vol. 30, no. 13, pp. 2627–2648, 2007.

[47] P. Bergstrom, K. Driscoll, and J. Kimball, "Making home automation communications secure," *Computer*, vol. 34, no. 10, pp. 50–56, 2002.

[48] J. García-Reinoso, I. Vidal, F. Valera, and A. Azcorra, "Zero config residential gateway experiences for next generation smart homes," *Computer Networks*, vol. 53, pp. 2967–2984, Dec. 2009.

[49] W. Teng, Y. Pao, and S. Chung, "Design of MyServer: A Residential Server in Smart Home Systems," *IEEE Asia-Pacific Services Computing Conference,*, pp. 580–586, 2008.

[50] I. Samaras, J. Gialelis, and G. Hassapis, "Integrating Wireless Sensor Networks into Enterprise Information Systems by Using Web Services," *Conference on Sensor*, no. Xml, pp. 580–587, 2009.

[51] R. V. Engelen, "Developing Web Services for C and C ++," *Ieee Internet Computing*, no. April, pp. 53–61, 2003.

[52] J. Blesa, P. Malagón, A. Araujo, and J. Moya, "Modular Framework for Smart Home Applications," *, Soft Computing, and*, pp. 695–701, 2009.

[53] N. Borisov, D. Brumley, H. Wang, J. Dunagan, P. Joshi, C. Guo, and I. Nanjing, "A generic application-level protocol analyzer and its language," in *14h Symposium on Network and Distributed System Security (NDSS)*, Citeseer, 2007.

[54] A. Hornsby, P. Belimpasakis, and I. Defee, "XMPP based wireless sensor network and its integration into the extended home environment," *Consumer Electronics*, pp. 794–797, 2009.

[55] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an agent communication language," in *Proceedings of the third international conference on Information and knowledge management*, p. 463, ACM, 1994.

[56] M. Barbuceanu and M. Fox, "COOL: A language for describing coordination in multi agent systems," in *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 17–24, Citeseer, 1995.

[57] A. Paschke, H. Boley, A. Kozlenkov, and B. Craig, "Rule responder: RuleML-based agents for distributed collaboration on the pragmatic web," in *true*, 2010.

[58] Savarimuthu, M. Bruce, and M. Purvis, "A software framework for application development using ZigBee protocol," *Discussion Paper Series*, 2009.

[59] J.-h. Su, C.-s. Lee, and W.-c. Wu, "The Design and Implementation of a Low-cost and Programmable Home Automation Module," *October*, pp. 0–5, 2006.

[60] G. Machado, F. Siqueira, R. Mittmann, and C. e Vieira, "Embedded systems integration using web services," *Learning*, 2006.

[61] S. M. Infrastructures, "Pervasive Monitoring-An Intelligent Sensor Pod Approach for Standardised Measurement Infrastructures," *Data Processing*, pp. 11440–11467, 2010.