# DEVELOPMENT OF ANDROID BASED HOTEL AUTOMATION SYSTEM


**By**

**Muhammad Dayan Tariq**

**Maaz Aziz**

**Saad Ahmad**

**Hamza Jwan**

**PROJECT SUPERVISOR: Dr Adnan Ahmad Khan**

**Project Report Submitted to Faculty of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Rawalpindi in Partial Fulfilment for the Award of B.E Degree in Telecommunication Engineering**

**JUNE 2014**

# DEVELOPMENT OF ANDROID BASED HOTEL AUTOMATION SYSTEM

**By**

**Muhammad Dayan Tariq**

**Maaz Aziz**

**Saad Ahmad**

**Hamza Jwan**

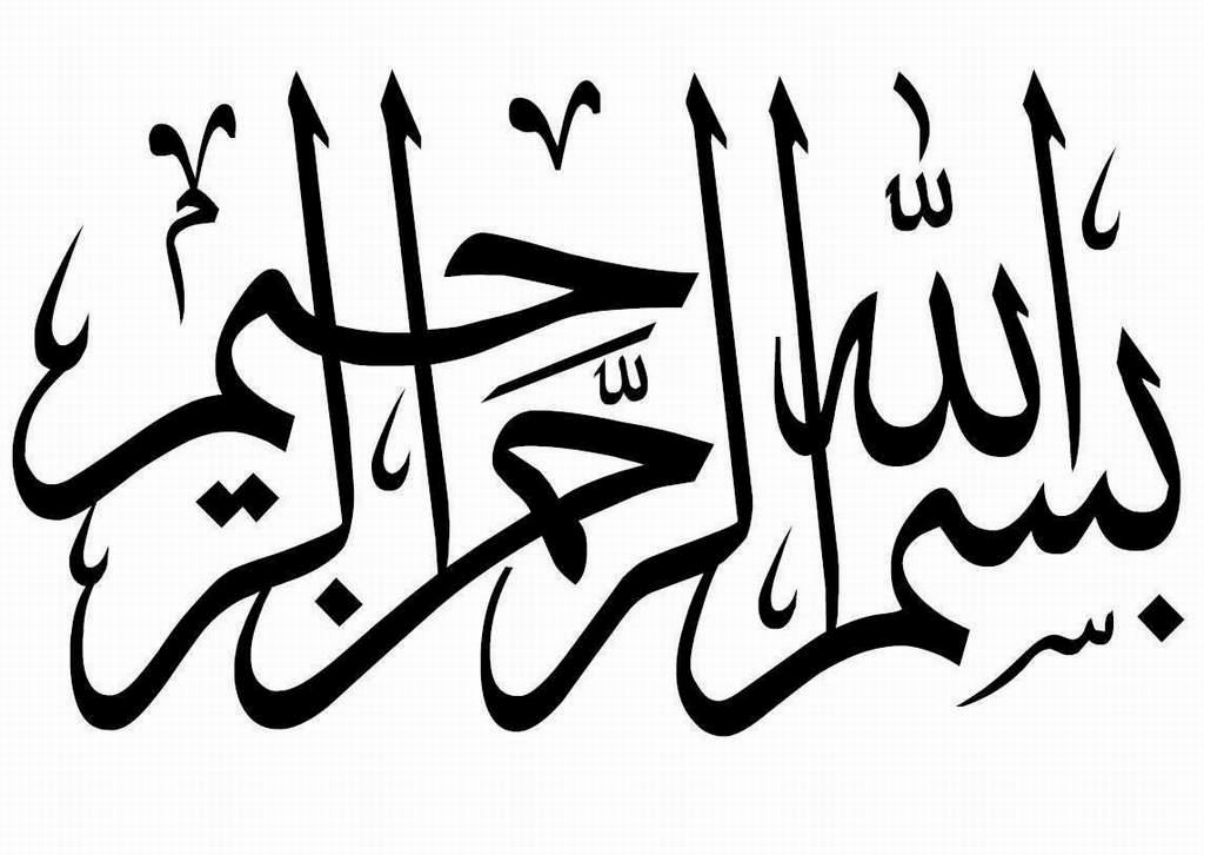
**Bachelor of Telecommunication Engineering**

**National University of Science and Technology**

**Project supervisor: _____**

**Dr Adnan Ahmad Khan**

**Program authorized to degree** **Bachelor of telecommunication engineering offer**




*“In the Name of Allah,*

*The Most Beneficent,*

*The Most Merciful”*

# A B S T R A C T

Today we are living in 21st century where automation is playing important role in human life. Home automation allows us to control household appliances like light, door, fan, AC etc. It also provides home security and emergency system to be activated. Home automation not only refers to reduce human efforts but also energy efficiency and time saving. The main objective of home automation and security is to help handicapped and old aged people who will enable them to control home appliances and alert them in critical situations. This paper put forwards the design of home automation and security system using Android ADK. The design is based on a standalone embedded system board Android ADK(Accessory Development Kit) at home. Home appliances are connected to the ADK and communication is established between the ADK and Android mobile device or tablet. The home appliances are connected to the input/output ports of the embedded system board and their status is passed to the ADK. We would develop an authentication to the system for authorized person to access home appliances. The device with low cost and scalable to less modification to the core is much important. It presents the design and implementation of automation system that can monitor and control home appliances via android phone or tablet.

This project titled "Development of Android based Hotel Automation System" enable us to develop hardware and software integrated system using android application for hotel automation, to wirelessly control electrical appliances using smart phone or tablet. It will provide power On/Off and status monitoring of the appliance. It will also provide control of an AC/TV like remotes on a single android device. We are able to achieve its core objectives using sensitive products like raspberry pi and Zigbee modules. The basic motivation behind the project is to provide user the control of the appliances and video surveillance wirelessly on the move. We will provide the user an attractive and user friendly android application which will be integrated with the hardware through Wi-Fi hotspot. We have used the latest available equipment. The research work we did helped us a lot in enhancing our knowledge and working skills. If we define the sequential layout of the project than it will go as our android device is connected to raspberry pie through Wi-Fi router and raspberry in turn is connected to Zigbee modules in a mesh network.

## D E C L A R A T I O N

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

# D E D I C A T I O N

Dedicated to our parents and mentors who have been a constant source of encouragement for us and to our teachers who have given us inspiration throughout our degree.

# A C K N O W L E D G E M E N T

All praises for ALLAH who gave us the strength and determination and enlightened us with the requisite knowledge on portion of this subject to complete this project. We gratefully acknowledge the continuous guidance and motivation provided to us by our project advisor Dr Adnan Ahmad Khan, without his personal supervision advice and help timely help completion would not have been possible

We deeply treasure the unparalleled support and forbearance that we received from our friends for their critical reviews and useful suggestions that helped us in completion of our degree project. We are deeply indebted to our family for their never ending patience and support for us and to our parents for the strength they gave us through their prayers

# T A B L E O F C O N T E N T S

# T A B L E O F F I G U R E S

# LIST OF ABBREVIATIONS

MCB – Main Control Board

Raspberry – Raspberry Pi

Xbee – Zigbee modules

TCB – Terminal Control Board

App– Android Application

TV- Television

Tx- Transmitter

Rx- Receiver

HA- Home Automation

# CHAPTER 1

# INTRODUCTION

## 1.1 Purpose

This project titled "**Development of Android based Hotel automation system"** is an innovative product capable of controlling electrical appliances, AC, TV remotely and on the move. It will give situational awareness solution by receiving live video streaming from cameras installed at the periphery of a setup. Features like receiving live videos as soon as door bell or any alarm is triggered with alerts will be available in this system. The main purpose is to wirelessly automate the full control over a setup externally and internally while acquiring video from security cameras installed. The project will be commercialized to office, institutes, hotels and industries. Our approach would be to learn Android application development, Raspberry Pi Linux programming, Zigbee module coding and then controlling the required hardware through the specially designed application running on a smart platform.

The project involves research and indigenous development of product that will be able to provide control of all electrical appliances and live video streaming from security cameras on Smart phone remotely and on the go. It will be a complete solution for security monitoring, theft control, remote alarm generation over a smart phone based platforms at a distant location.

A client-server network will be deployed where android device and microcontrollers act as clients and the server will be made on a computer. The server will control the communication between microcontrollers and android device.

The idea of home automation using android is just that. We are committed to develop an efficient, productive and dynamic system of home automation using an android device (2.2 and above).

## 1.2 Background/Motivation

Most advanced home automation systems in existence today require a big and expensive change of infrastructure. This means that it often is not feasible to install a home automation system in an existing building. The Home Automation is a wireless home automation system that is supposed to be implemented in existing home environments, without any changes in the existing infrastructure. Home Automation lets the user to control his home from his or her Smart Phone. In the program the user can create actions what should happen with electrical devices in the network depending on the sensors sensing surrounding environment. Every Home Automation box is a stand-alone device. It is connected to the mains and controls the power outlet of the electrical device that is plugged into it. There will be a receiver and transmitter in each of the box, so they can exchange information with the master (a computer). People can control power supply of electrical devices in order to create an interactive home environment to facilitate the control without changing any home appliance. People can enjoy the high technology and simplicity modern life style. Each device will be with standard setup and while adding it into network; it can be given an address and tasks to do. All the setting will be easily resettable to default value, so people can move the devices between different electrical devices and networks. Home Automation boxes will be put into different rooms at home, depending on the needed functionality. Various different sensors could be attached to the boxes. The sensors are used as triggers for actions, that user can set up in the computer program.

Android does not simply represent an operating system for mobiles and tablets. It represents a whole new market with an ever increasing growth. There is great room for engineers, developers and employers to utilize this emerging market by bringing new ideas in this industry and developing such systems that take android usability to new heights.

The idea of hotel automation using android is just that. We are committed to develop an efficient, productive and dynamic system of hotel automation using an android device.

We have developed fast and reliable automated hotel architecture using android platform over Wi-Fi environment, rather than using slow and cumbersome techniques. After proposing the project's design and getting started with it, many hitches urged us to make

amendments in the proposed design and work in a different scheme, which then turned to fruitful results. The main purpose is to wirelessly automate the full control over home externally and internally while acquiring video from security cameras installed. The idea may be commercialized to office, institutes and industries as well. Our approach would be to learn Android application development and then controlling the required hardware through this application.

### 1.3 Project Description

In this project we have developed a home automation system, which is mainly integration of hardware devices control module and Android application. The hardware control part would be done using microcontrollers, Relays, Zigbee and raspberry pi where the front end application would be based on Android. The Android based application would be used for performing several applications from android smart phone. The Android application includes image/video acquisition from camera, capable to ON and OFF certain switches, control fan speeds, brightness of bulbs, AC/TV remote and even manual control over all appliances too. Figure 1 represents the flow diagram of the project.

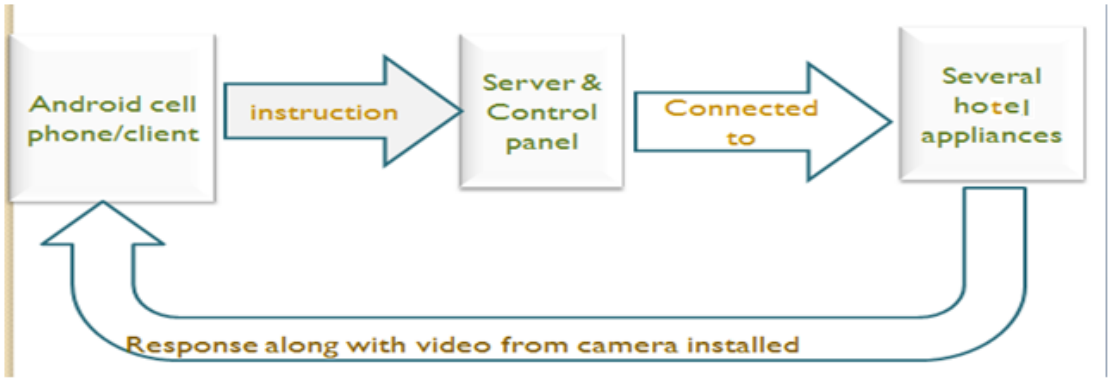



Figure 1: Flow Diagram

Project overview diagram is shown in below figure:

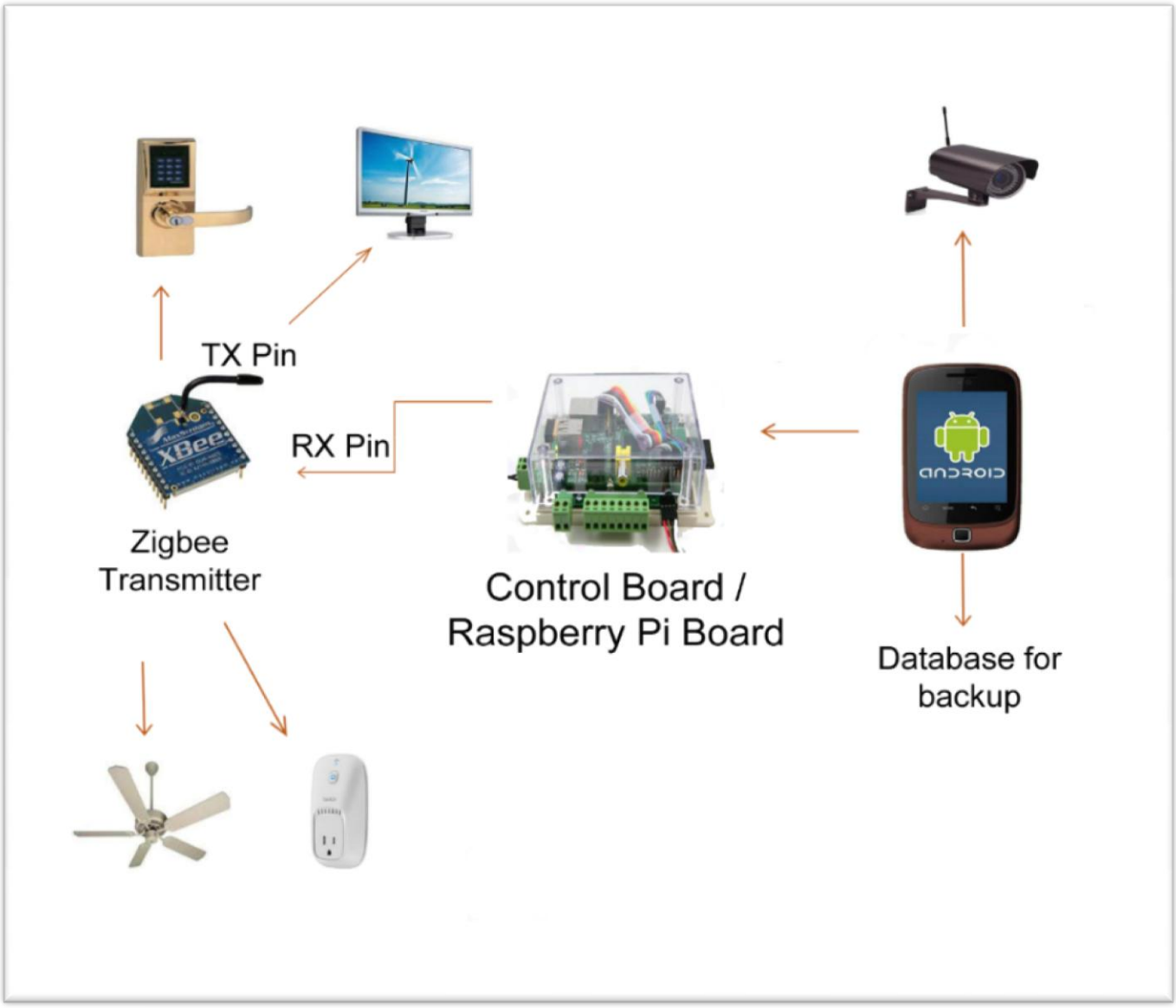



Figure 2: Project Overview

# CHAPTER 2

# SCOPE, OBJECTIVES, SPECIFICATIONS AND DELIVERABLES

## 2.1 Scope

The development of Android application is very innovative. The main purpose is to wirelessly automate the full control over home externally and internally along with the video streaming. The idea may be proposed to our public and private sector as well.

## 2.2 Objectives

Objectives of our project are following:

- Developing the android application for home automation system.
- To wirelessly control home appliances using this android application.
- To add TV/AC remote control liability.
- To get real time video surveillance on our android smart phone.
- Developing a website through which we can automate our house form anywhere.

## 2.3 Design Specifications

Most important and sensitive components used in this project are

- Raspberry Pi
- Zigbee Modules
- Micro Controllers

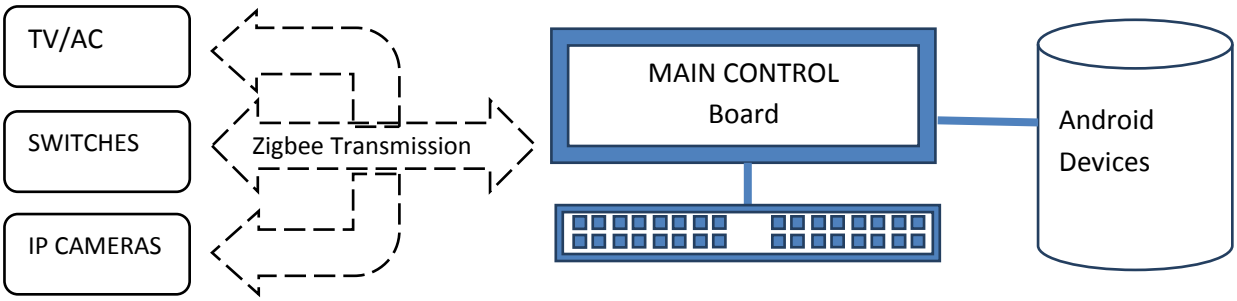



Figure 3: Project Diagram

### 2.3.1 Raspberry Pi

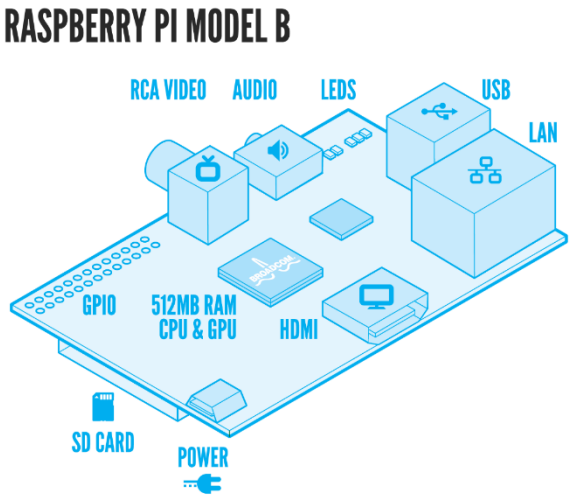



Figure 4: Raspberry Pi model

The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video. We want to see it being used by kids all over the world to learn programming.

*Dimensions*

The Raspberry Pi measures 85.60mm x 56mm x 21mm (or roughly 3.37″ x 2.21″ x 0.83″), with a little overlap for the SD card and connectors which project over the edges. It weighs 45g.

*Technical Features*

Raspberry Pi is a credit card-sized computer powered by the Broadcom BCM2835 system-on-a-chip (SoC). This SoC includes a 32-bit ARM1176JZFS processor, clocked at 700MHz, and a Videocore IV GPU. It also has 256MB of RAM in a POP package above the SoC. The Raspberry Pi is powered by a 5V micro USB AC charger. Raspberry Pi LogoWhile the ARM CPU delivers real-world performance similar to that of a 300MHz Pentium 2, the Broadcom GPU is a very capable graphics core capable of hardware decoding several high definition video formats.

The Model B — features HDMI and composite video outputs, two USB 2.0 ports, a 10/100 Ethernet port, SD card slot, GPIO (General Purpose I/O Expansion Board) connector, and analog audio output (3.5mm headphone jack). The less expensive Model A strips out the Ethernet port and one of the USB ports but otherwise has the same hardware. It is this model that is the "$25 PC" that originally made so many headlines.

### 2.3.2 Zigbee Modules

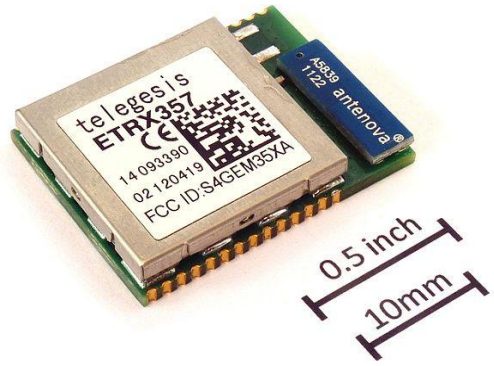



Figure 5: A Zigbee Module

Zigbee offers Green and global wireless standards connecting the widest range of devices to work together intelligently and help you control your world.

ZigBee is a specification for a suite of high level communication protocols used to create personal area networks built from small, low-power digital radios. ZigBee is based on an IEEE 802.15 standard. Though low-powered, ZigBee devices can transmit data over long distances by passing data through intermediate devices to reach more distant ones, creating a mesh network. The decentralized nature of such wireless ad hoc networks make them suitable for applications where a central node can't be relied upon.

*Technical Features*

ZigBee is used in applications that require only a low data rate, long battery life, and secure networking. ZigBee has a defined rate of 250 kbit/s, best suited for periodic or intermittent data or a single signal transmission from a sensor or input device. ZigBee protocols are intended for embedded applications requiring low data rates and low power consumption. The resulting network will use very small amounts of power — individual devices must have a battery life of at least two years to pass ZigBee certification.

Typical application areas include:

- Home Entertainment and Control — Home automation, smart lighting, advanced temperature control, safety and security, movies and music
- Wireless sensor networks
- Industrial control
- Embedded sensing
- Medical data collection
- Smoke and intruder warning

### 2.3.3 AT89C51 Microcontroller

After the signal is received the microcontroller plays an important role as it is used for interfacing the modules for their specific functions. Here we are using two microcontrollers (AT89C51) i.e.; one is used for interfacing Zigbee module and Raspberry Pie module while the second is used for interfacing of AC/TV remote

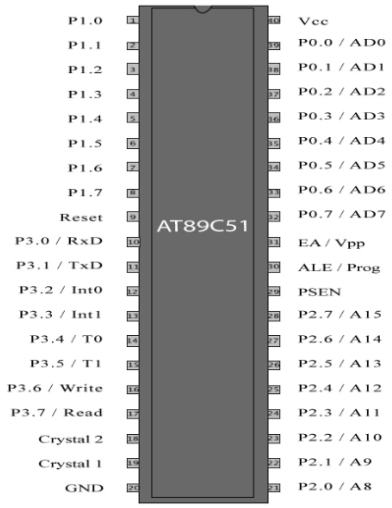



Figure 6: AT89C51 block diagram

Figure 7: AT89C51 microcontroller

### 2.4 Deliverables

The technical details of final deliverable are:

- Android app with user-friendly GUI enabling user to control Home appliances within Wi-Fi zone
- Receiving live video streaming on android device from security camera.
- Door access through android device. The user can monitor when anyone arrives at the door and open the door by just giving command from his smart phone using the app.
- Controlling all the room appliances by android device.
- A password scheme for client-server connection for security

# CHAPTER 3

# LITERATURE REVIEW

## 3.1 Background

Home Automation is a concept that has been developing reasonably slowly when you compare it to how other such as televisions has progressed. Whereas other technologies such as high-definition televisions have developed and become much cheaper (around 20% per annum), home automation is still generally quite an expensive and exclusive concept for most people. This chapter will look into what Home Automation is and what current technologies exist. I will look at how technologies are currently implemented and the groups of people that will use the technologies. I will look at the use of automation as a disability aid as well as it just being a luxury within a home.

## 3.2 What Is Home Automation

Home Automation (also referred to as Domotics) is "the use of one or more computers to control basic home functions and features automatically and sometimes remotely, an automated home is sometimes called a smart home". Home Automation can be used for a wide variety of purposes; from turning lights on and off to programming appliances within a home and the programming of timers for these various devices. Home Automation is often used as a luxury convenience system within a home and often it is expensive to have installed due to their relative exclusivity in the current market. As Home Media devices become cheaper, Home Automation is a technology that more people will be looking into to install in their house.

## 3.3 Why Is It Important

Home Automation (HA) is quite a broad area and therefore has a variety of uses. Some areas are very important and can greatly improve the quality of life for individuals, whilst other aspects of

HA are used for convenience rather than an essential item. Starting off with the more essential aspect of HA are security aspects. Cameras and sensors can be connected to a HA system. These can be used to monitor and record activity around a building/house and can make remote monitoring much easier. This can then make the technology of burglar alarms much more complex as they not only recognise movement with sensors but they can also store and relay video images for the owner to then show the police if necessary.

Another use for HA is with the elderly and people with impaired physical mobility. Tasks that are simple for some people are much harder if you are less mobile can be made much easier using an automated system.

Automated systems can be linked to motors and switches to perform tasks controlled on a simple control panel. For example the opening and closing of curtains in a room could be controlled by a remote control. The most dominant uses of HA are with home lighting, multimedia and smart home appliance control. This tends to be the more exclusive market and often quite expensive.

Home Automation will only be adopted if it is at least as easy to use as the original task in which it is replicating For example if switching on a light via a HA system is more complicated than pressing a button on a wall then there is arguably no advantage to having the device automated and it might just promote user aggravation.

HA software in Australia is being used to shut down lighting and devices in homes from their computers and mobile phones. A pilot study from the company who produced the software showed that an office building was able to cut its energy consumption by 25 percent. With the constant strive to create a greener planet, HA could certainly help us in doing so.

### 3.4 Current Home Automation Systems

Several products are available around the world which focuses on one aspect or the other. However, no complete indigenously developed product is at the moment available in market that can provide features competitive with international products at a reasonable cost and reliable services. A review of few products and their features is as under:-

(1) Control4

A full fledge product has been developed named Control4 which offers the ultimate home automation solution by making the products and systems you already have and use everyday, work together. By integrating everything from lighting control, music, home theater, climate control, security—even iPads, iPhones and Android smartphones and tablets—a smart house by Control4 creates personalized experiences that enhance your life and provide added comfort, savings, convenience, and peace of mind.

Start small by automating just your family room or home theater. Or go big with whole-home automation. Control4 is built to grow as you go. No matter what the application, the experience is unforgettable and you'll wonder how you ever lived without home automation.

Main purpose of this project is to develop a product which should provide industry as well as domestic level automation and security surveillance involving monitoring of machines and appliances as well. The details are at www.control4.com.

(2) Loxone

Loxone's Miniserver based Home Automation solution allows you to control any task in and around your Smart House. From simple lighting and blind control and intelligent, cost-efficient zoned heating to burglar alarms. And this appilicatio was basically developed for old-age and handy capped people. Details can be viewed at www.loxone.com

(3) Wcharger

Another product that has been developed named WLight, which also is just a bulb, whose colors can be changed from smart phones. This application has very high graphics and user friendly, it is available at www.wcharger.com

This project will be similar to the above mentioned products but will have very powerful processer i.e Raspberry Pi that can manage a large bandwidth with smaller size. Main features of the existing products with low cost and indigenous production will be hallmark of this project.

# CHAPTER 4

# DESIGN AND DEVELOPMENT

## 4.1 Detailed Design

The plan to carry out this project consists of 4 main tasks these are:

1. Configuration of Main Control Board (MCB)

2. Creation of Terminal Control Board (TCB)

3. Establishing communication between MCB and TCB

4. Establishing connection between android device and MCB

5. Development of Android application

### 4.1.1 Configuration of Main Control Board (MCB)

MCB consists of Raspberry Pi, a charger, and a zigbee module.

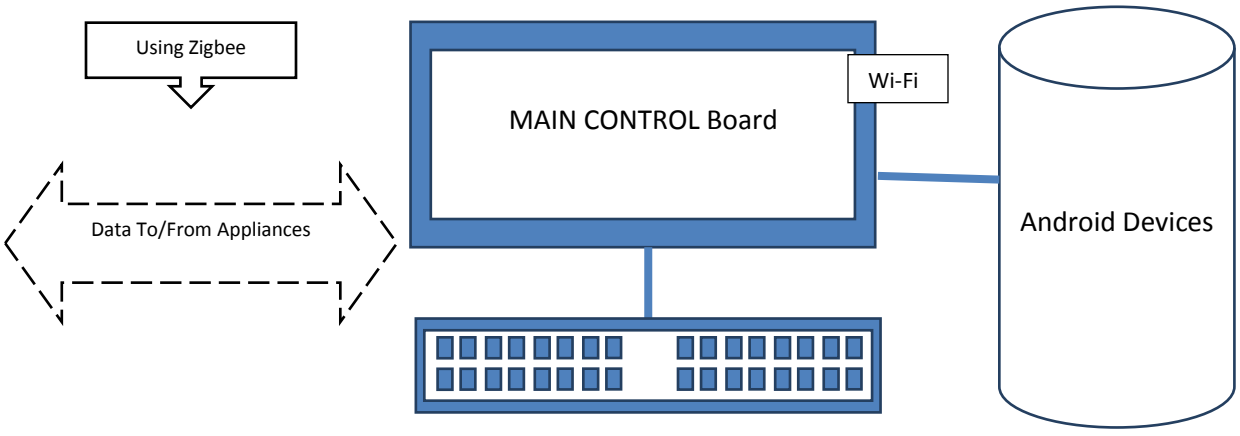



Figure 8: Main Control Board Overview

*Performing initial configuration:*

For raspberry, we flashed an SD card with Raspberry Pi image (Raspbian i-e raspberry pi operating software available in internet), and plug the SD card into Raspberry Pi.

We did initial configurations on Raspbian-powered Raspberry Pi by using raspi-config tool. The first time Raspberry Pi setup includes expanding local file system into the entire SD card space, changing host name, activating SSH service, enabling booting to desktop, etc.

After the Raspbian system is installed on an SD card, put the SD card into the Raspberry card reader. We need to make sure that everything (e.g., external HDMI monitor/TV, keyboard and mouse) is connected properly. Upon the first time booting, we see the following setup options appear in the screen. This is a Raspberry Pi configuration tool called raspi-config.

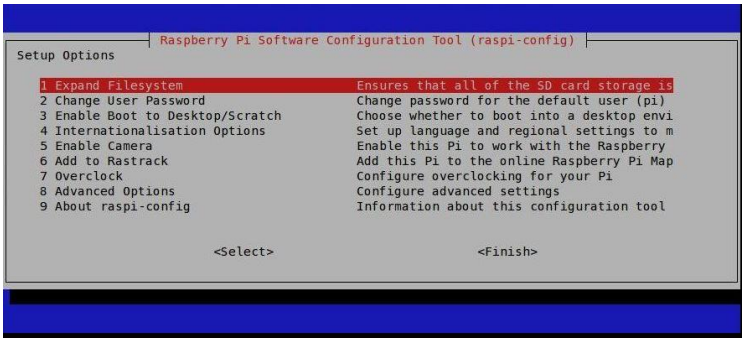



Figure 9: Raspberry Pi configuration tool called raspi-config

*Expanding the File System:*

When we boot the raspberry, option for file expanding appears which screen as below

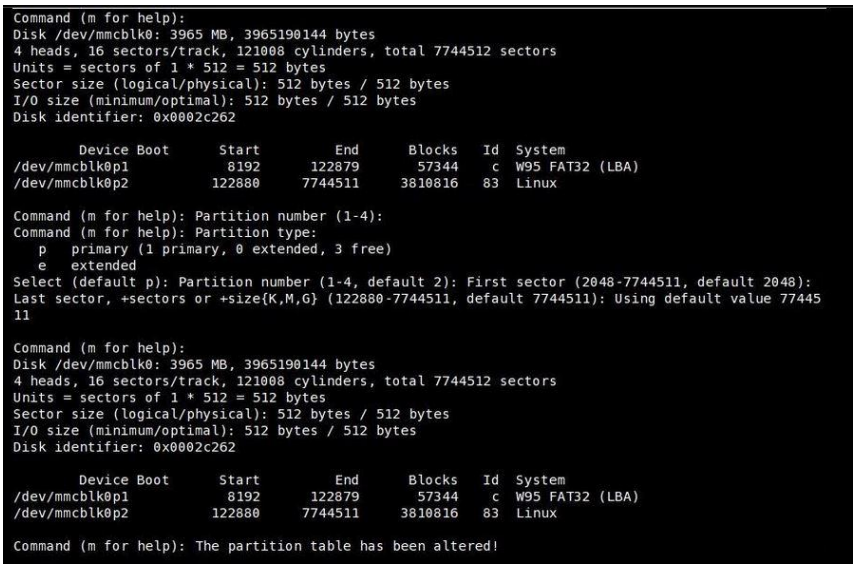



Figure 10: Expanding the File System

The resulting partition layout change will take effect after rebooting Raspberry Pi.

*Enable Boot to Desktop / Scratch:*

The menu option called “Enable Boot to Desktop” in raspi-config allows us to automatically launch and log in to the GUI desktop of Raspbian upon booting. We can also enable the Scratch programming environment. By default, Raspbian is set to boot into a text-based console log in. Depending on our case, enable booting to desktop.

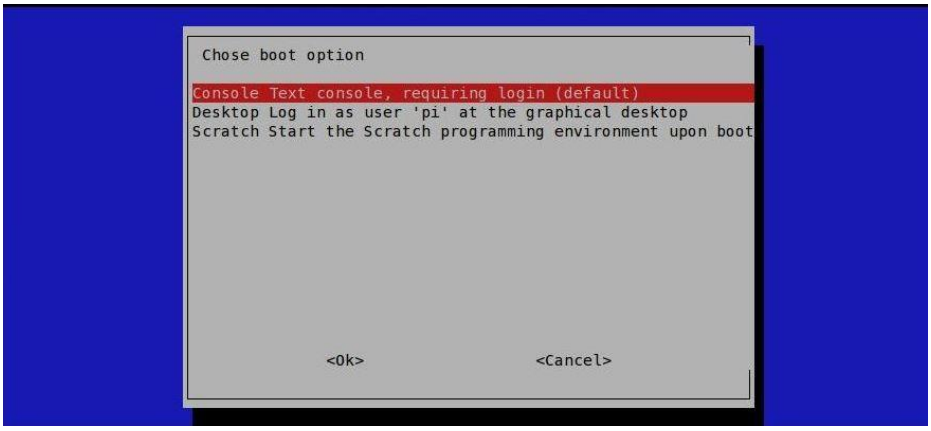



Figure 11: Enable boot to Desktop

*Advanced Options:*

In the "Advanced Options" menu, we can manage several important options for Raspberry Pi. Under this menu, we see several advanced options like below. The "Overscan" option is useful when we set up an external HDMI monitor or TV properly. The default value is enabled, but to make sure, we choose "Enable" button again inside the Overscan menu.

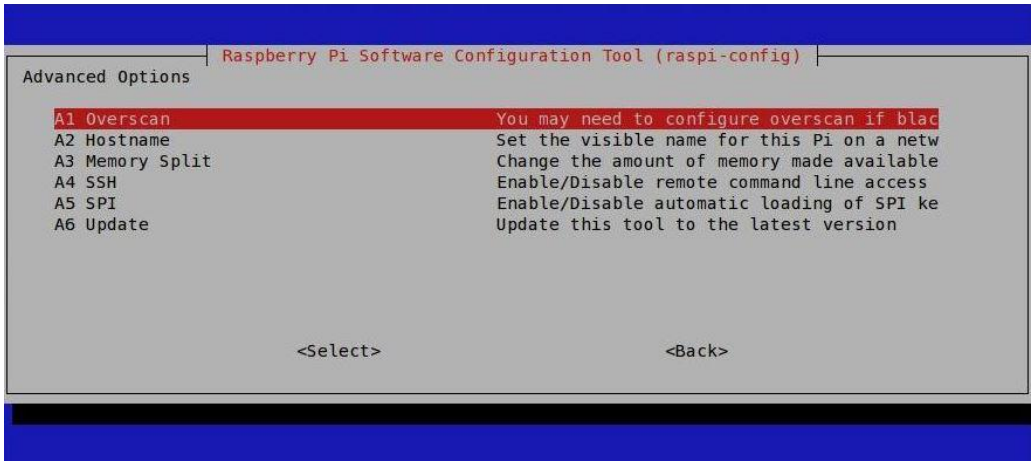



Figure 12: Over scan Option

Finally, we activated SSH service in Raspberry Pi by choosing the SSH menu and entering "Enable" button. The SSH service is needed when we want to run Raspberry Pi headless. Once we are done with configuration, finished raspi-config, and reboot Raspberry Pi. The following screenshot shows the Raspbian GUI desktop with default login (user name: "pi", and password: "raspberry").

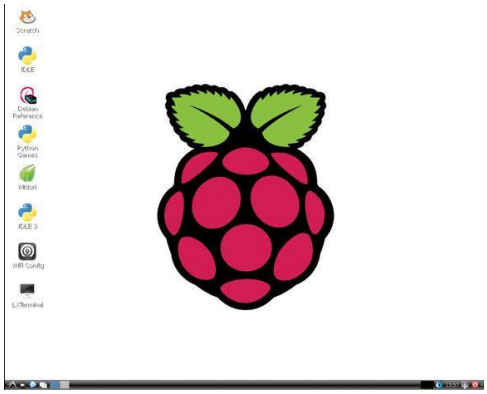


Figure 13: Final desktop screen of raspberry pi

### 4.1.2 Creation of Terminal Control Board (TCB)

This phase consists of control board with which all the appliances will be attached. It can be called as “Switching Board” like which are installed at homes and offices but this board will be attached to zigbee module which will be receiving commands from raspberry and perform accordingly.

Circuit diagram for this is shown as :-

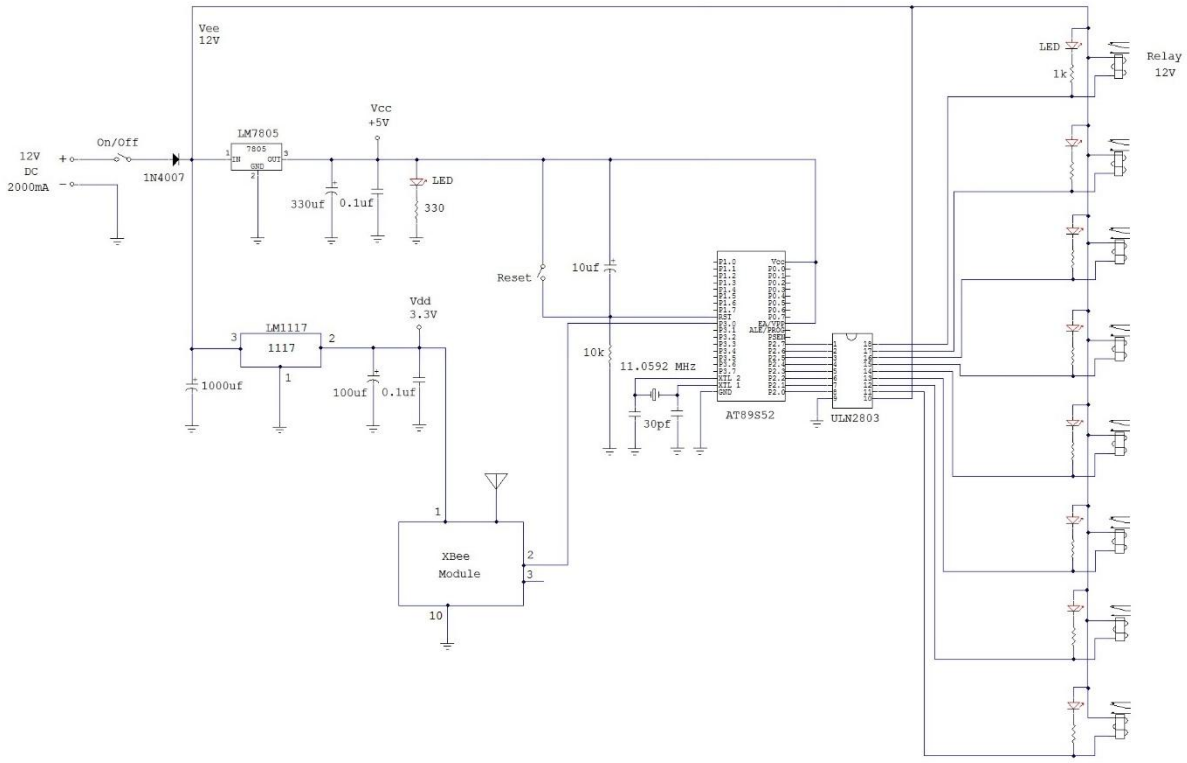



Figure 14: Circuit Diagram, Terminal Control Board

Here we are controlling 8 relays, each of them are connected to auto couplers. We are using Atb9s52 microcontrollers which have 10 output ports, out of which eight are connected to auto couplers. Zigbee receiver is connected to to microcontroller input port 3 and getting voltage of 3.3v. We are using battery of 12v, LM1117 and LM7805 IC for the power input and to control voltage regulations.

### 4.1.3 Establishing communication between MCB and TCB

As we know that what is MCB and TCB, we need to establish communication between them meaning how data will flow from raspberry to control board wirelessly through Xbee modules.

One zigbee module is attached to raspberry which acts as a transmitter and the other one is attached with microcontroller in the Terminal Circuit Board.
We followed the following steps in providing communication between the two.

Software Setup
Step 1: Download and Install Telegesis Software (Built in driver provided by the manufacturer)

Step 2: Download and Install USB Drivers (Hardware USB Bus & Virtual COM Port Drivers): The USB interface board is a "plug-and-play" device that should be detected by the PC automatically. To interface between the modem and a PC/Raspbian OS, two drivers must be installed: a USB driver and a virtual COM port driver that makes the USB port look and perform like a physical COM port. After the modem is detected, a wizard for installing the USB driver is launched.

*Configure RF Modules*

*Setting Parameters:*

1. Launch the Telegesis Software: TelegesisTerminal4_0Setup.

2. In Telegesis, under the PC Settings tab, select the PC serial COM port from the displayed list. The USB port will be labeled as a Digi PKG-U Serial Port Adapter

3. We need to to be sure that the baud rate and data settings match the internal settings of the radios. The default settings for the radios are Baud Rate: 9600, Flow Control: HARDWARE, Data Bits: 8, Parity: NONE, and Stop Bits: 1.

4. In the Telegesis Modem Configuration tab, click Read to display a list of the RF module configuration parameters. Parameters shown in green are still set to default. Parameters in blue have been changed.

5. We select any of the module parameters we wish to change (e.g. Destination Address, Encryption, etc.) and type in or select the desired value.

6. Now we can either broadcast or unicast the message to a particular xbee device.

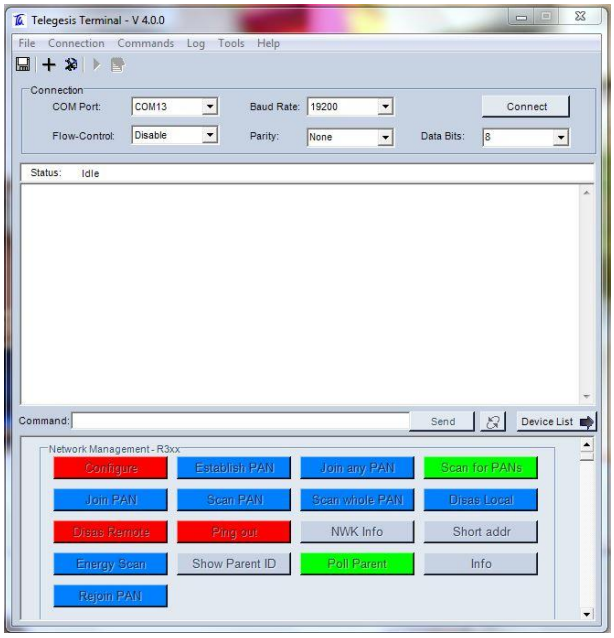



Figure 15: Xbee Telegasis GUI

### 4.1.4 Establishing connection between Android device and MCB

Previously we can pass commands through raspberry and could switch our electrical appliances. But we need to send commands from our android application through MCB where raspberry process the command and forward them to TCB using xbee where by android application will be communicating to MCB through Wi-Fi. The simple way to link the raspberry Pi with an Android phone is via a web interface i.e. the Pi runs Apache + CGI (Common Gateway Interface) scripts,

and the Android phone simply connects to the same Wi-Fi network and a local web page access panel.

*1st - Connect raspberry to Wi-Fi network and acquire IP address.*

After plugging in USB WiFi dongle, power up Raspberry Pi. Double-click on "WiFi Config" icon on the desktop.

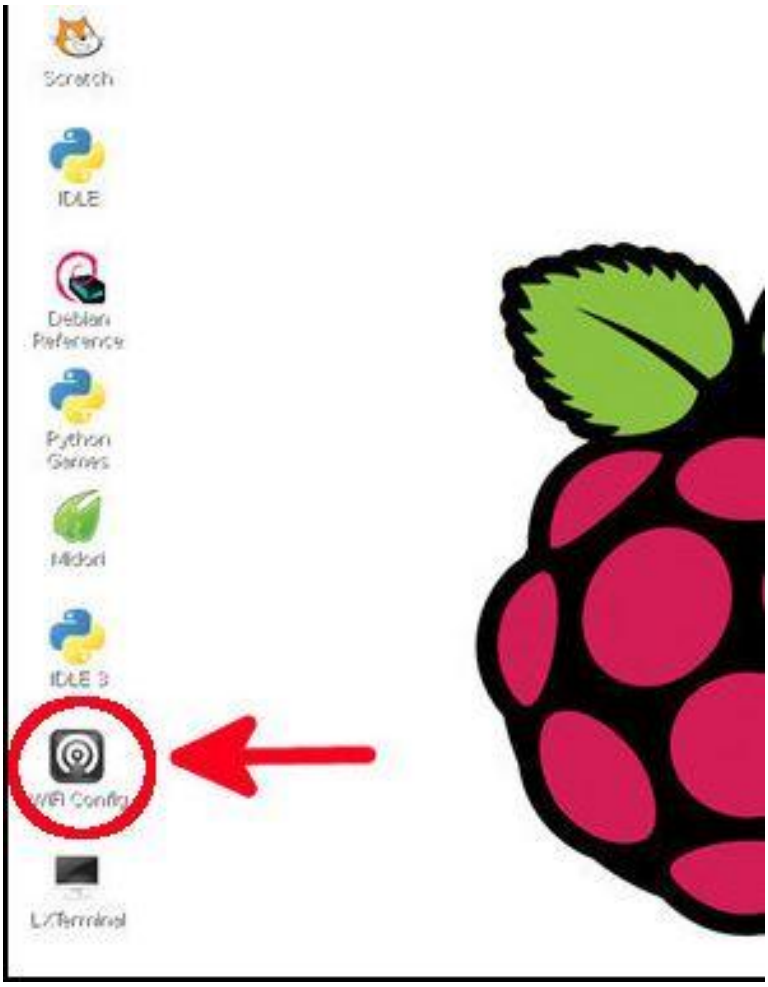



Figure 16: Wi-Fi Config icon

This will give you a "wpa_gui" window (see picture below).


Figure 17: Wifi Config Window

Click on "Scan" button to open the following scan result window.

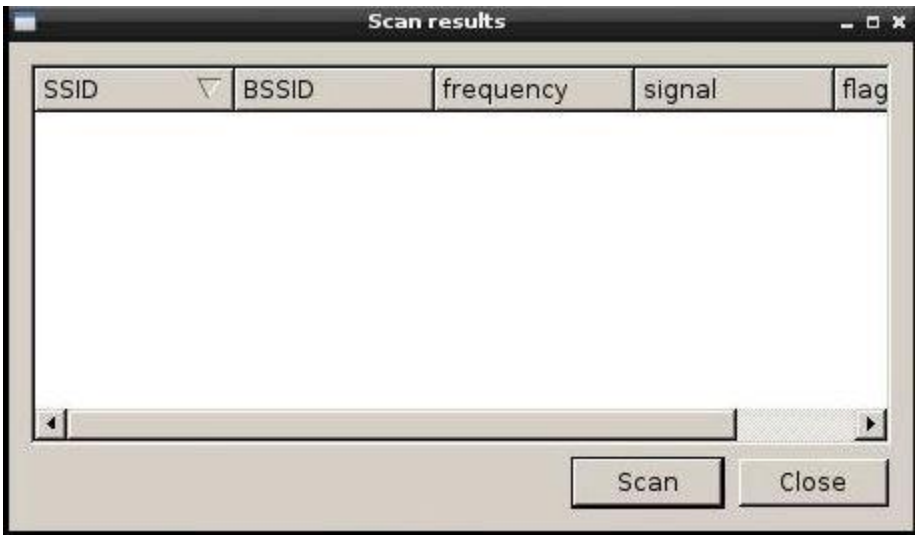



Figure 18: Scaning Wi-Fi

Click on "Scan" button on this window to get a list of available WiFi access points.

Double-click on the SSID of the access point to connect to.

Click on "Current Status" tab, and then click on "Connect" button to connect the access point. It will take a couple of seconds until you get an IP address from DHCP negotiation.


Figure 19: Acquiring IP Address

2nd – Enter that IP address in android app in order to handshake raspberry with android app module.

3rd – When connection is established, green light will start flashing at raspberry board and we can control our appliances via android phone.

### 4.1.5 Development of Android Application

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit (SDK), but other development tools are available.

The Android software development kit (SDK) includes a comprehensive set of development tools. These include adebugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later; for the moment one can develop Android software on Android itself by using [AIDE - Android IDE - Java, C++] app and [Android java editor] app. The officially supported integrated development environment (IDE) isEclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit andApache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

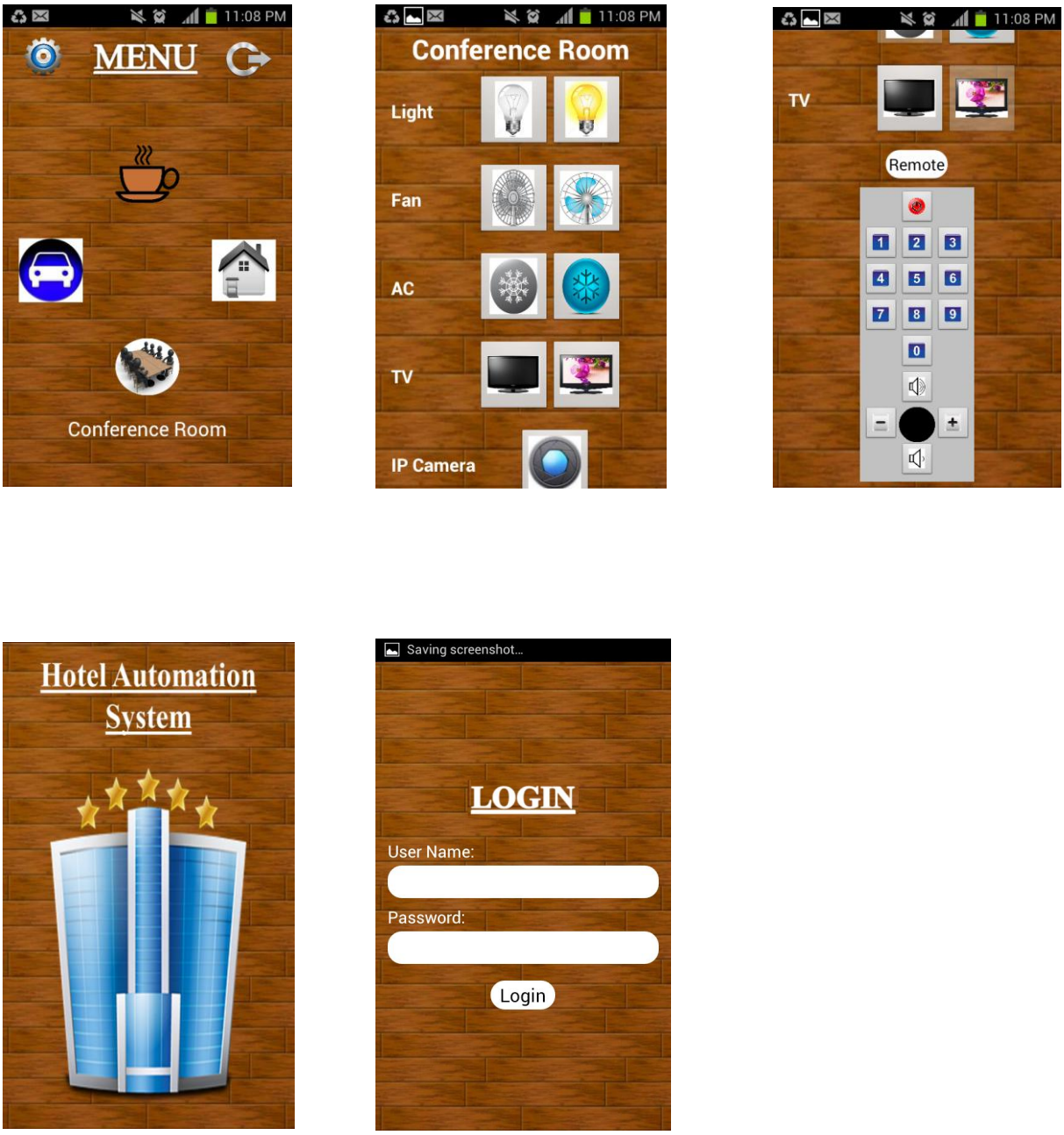



Figure 20: Android App Layout

# CHAPTER 5

# RESULT AND ANALYSIS

## 5.1 Switching:

As it is mentioned in the project deliverables that by using our application that is android based hotel automation application we will be able to control the electrical appliances that are usually the essentials of every room, these include lights, bulbs, fans and electrical switches.

## 5.2 AC/TV Control:

The second deliverable of this project is the AC/TV control. We have designed the AC/TV remote and we have integrated it with our main control board and terminal control board. And this additional feature will enable us to switch on and off the AC and to set the thermo-state of AC and TV remote will give us the control to switch on and off the TV and to switch channels and volume control.

## 5.3 Real time video surveillance:

The induction of IP camera has given us the security feature of having real time video surveillance of the vicinity in which the camera will be deployed. We have integrated the IP Camera with our android application and main control board. So you can have the real time video surveillance on your cell phone or any device which in running on Android.

The final result obtains at the completion of the hardware having a complete layout of house appliances i.e. switches, fans, blub, and remotes. Our Hardware have in total 16 relays, 8 for the appliances and 8 for the remote functions and 1 security camera attached to it for the real time video surveillance. The prototype that is being made is shown below:

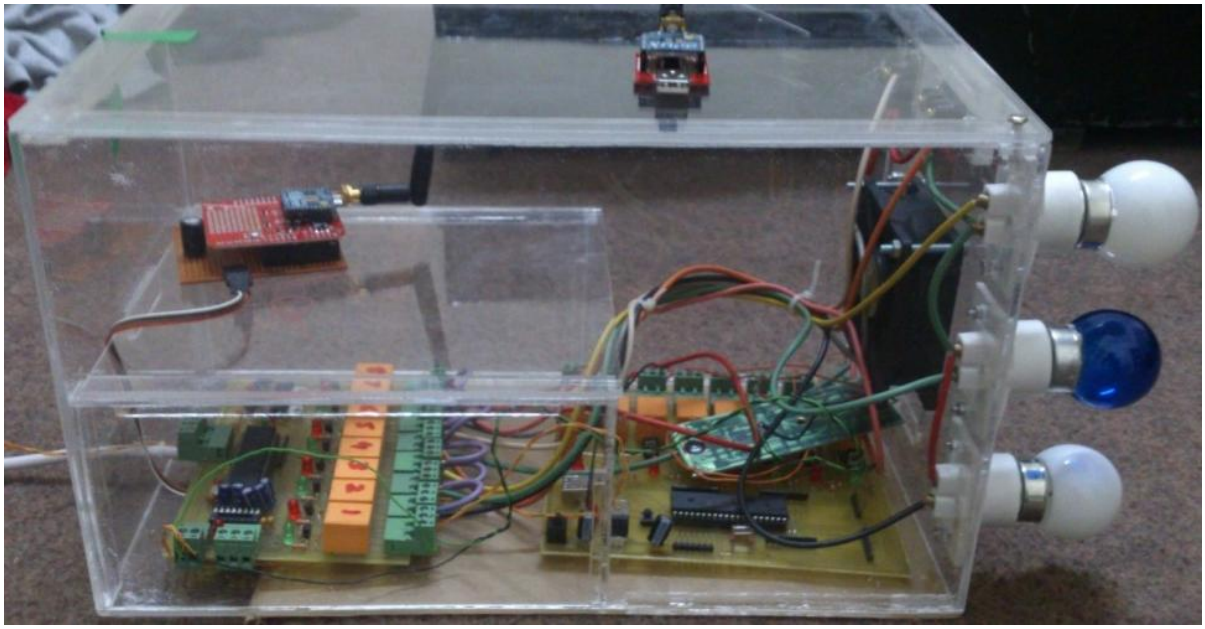


Figure 21: Hardware Box

### 5.4 Limitations

- This is prototype which means there is a lot of room for improvement. High resolution camera, more user friendly graphical interface for android application can be implemented as per the requirement.
- The range of Wi-fi and zigbee modules has their limitations. Although this issue is very much resolved by using the zigbee modules in mesh network.
- The implementation of this project on a larger or industrial scale will be costly but that will be only the first time installation cost. And moreover the maintenance cost will be accounted but that would be minimal.

# CHAPTER 6

# RECOMMENDATIONS FOR FUTURE WORK

## 6.1 Present Work

Currently the home system is used for two main purposes. They are

1) To configure equipments to the system

2) To act as at the mediator between the server and the household equipments.

There are several improvable areas in both of these sections.

Currently the server decides whether the home system has failed or running and alerts the mobile user. This might result in a small delay between the failure in home system and alerting the user. Currently the home system is a standalone application and it doesn't have a web interface.

## 6.2 Future Work

With the arrival of 3G and 4G technology now in Pakistan, this project can be extended to be operated on cellular technologies. In future it is possible to make the home system send alerts directly to the mobile user if there is a failure in the ADSL connection. Though this may result in an increase in the cost of the system this is useful especially if security is a critical requirement. In future a web interface can be added to the home system.

# CHAPTER 7

# CONCLUSION

As a final review, the project aims to find variety of applications in industrial as well in public sector. This project is a wireless based home automation system which can be controlled through Smart phone working on Android platform. In the proposed architecture, wireless component is added by GPRS and for home networking ZigBee technology is used. For command processing an application is developed and installed in the mobile phone.

## 7.1 APPLICATIONS:

### 7.1.1 Industry

Every aspect of an industry requires monitoring. The factories and manufacturing units also need a constant check during manufacturing and other processes. The managing staff has always faced a difficulty to keep up with all the processes, continuously checking their status after certain periods and making decisions accordingly. This makes the job of a manager very hectic.

The automation and security system can be installed in the factory to ease the manager's job. He would be able to view status of machinery and processes on his Android phone/tablet and control the devices connected to machinery. Through live streaming of video, he would also be able to keep an eye on the subordinates by viewing their activities on his phone/tablet. It will increase the efficiency of manager by making his job easier and improve the overall productivity of industry.

### 7.1.2 Public Sector

The public sector may install the system for security purposes, such that every user of an organization is able to view live video stream from the desired area. Schools may implement this system as well, automating the classrooms, audio-visual aids and monitoring system during examinations for plagiarism.

Similarly hospitals may implant the automation system for Emergency departments, Operation Theatres, automatic doors and patient wards as well. Doctors would be able to keep a check on

their patients without having to physically attend them and hence treat a greater number of patients without getting fatigued.

### 7.1.3 Commercial Sector

The commercial sector is always looking for new marketing tactics and technological enhancements that they can boast in order to boost their sales. In an a highly competitive market like this, such enhancements can make all the difference that they need. Such a system if installed in a motel for example, can give it an advantage over other motels by promising increased comfort and security. This system would add comfort for the user who stays in the room as the user would be able control the appliances (fan, Air Conditioner, Lights, TV etc.) of his room and also be able to order food, receive and pay his bills and do other things through this system.

In addition, the management and security staff will work much more efficiently as the cameras installed in corridors, galleries, entries and exits of motel will provide live streaming to the security manager taking security of motel to a new level.

# Appendix A

The design code for Xbee is under:

```
package com.rapplogic.xbee.examples.zigbee;


    import java.util.List;


    import org.apache.log4j.Logger;
    import org.apache.log4j.PropertyConfigurator;
    import com.rapplogic.xbee.api.ApiId;
    import com.rapplogic.xbee.api.AtCommand;
    import com.rapplogic.xbee.api.AtCommandResponse;
    import com.rapplogic.xbee.api.PacketListener;
    import com.rapplogic.xbee.api.XBee;
    import com.rapplogic.xbee.api.XBeeException;
    import com.rapplogic.xbee.api.XBeeResponse;
    import com.rapplogic.xbee.api.zigbee.ZBNodeDiscover;
    import com.rapplogic.xbee.util.ByteUtils;
    public class ZBNodeDiscoverExample {
    private final static Logger log = Logger.getLogger(ZBNodeDiscoverExample.class);
    private XBee xbee = new XBee();
```

```
public ZBNodeDiscoverExample() throws XBeeException, InterruptedException {

    try {
        // replace with your serial port
        xbee.open("/dev/tty.usbserial-A6005v5M", 9600);
        // get the Node discovery timeout
        xbee.sendAsynchronous(new AtCommand("NT"));
        AtCommandResponse nodeTimeout = (AtCommandResponse)
        xbee.getResponse();
        // default is 6 seconds
        int nodeDiscoveryTimeout =
        ByteUtils.convertMultiByteToInt(nodeTimeout.getValue()) * 100;
        log.info("Node discovery timeout is " + nodeDiscoveryTimeout + "
        milliseconds");
        log.info("Sending Node Discover command");
        xbee.sendAsynchronous(new AtCommand("ND"));
        // NOTE: increase NT if you are not seeing all your nodes reported
        List<? extends XBeeResponse> responses =
        xbee.collectResponses(nodeDiscoveryTimeout);
        log.info(" (NT)");
        for (XBeeResponse response : responses) {
            if (response instanceof AtCommandResponse) {
```

```
                    AtCommandResponse atResponse = (AtCommandResponse)
                    response;

                    if (atResponse.getCommand().equals("ND") &&
                    atResponse.getValue() != null && atResponse.getValue().length > 0)
                    {

                        ZBNodeDiscover nd =
                        ZBNodeDiscover.parse((AtCommandResponse)response);

                        log.info("Node Discover is " + nd);

                    }

                    }

                }

            } finally {

                xbee.close();

            }

    }


    public static void main(String[] args) throws XBeeException, InterruptedException {

        PropertyConfigurator.configure("log4j.properties");

        new ZBNodeDiscoverExample();

    }

    }
```

To configure the relays:

```
#include<at89x52.h>
        #include <string.h>
    //Pin Definitions
        sbit Relay_1        = P0^7;
        sbit Relay_2        = P0^6;
        sbit Relay_3        = P0^5;
        sbit Relay_4        = P0^4;
        sbit Relay_5        = P0^3;
        sbit Relay_6        = P0^2;
        sbit Relay_7        = P0^1;
        sbit Relay_8        = P0^0;
    //Variables
        unsigned char       x;
        unsigned char       y;
        unsigned char       input;
        unsigned char       counter;
        unsigned char       buffer[10];
        unsigned char       device;
        bit                 status;
        #include "subroutines.c"
    //MAIN----------------------------------------------------
```

```
void main(){
        ser_init(9600);
        counter=1;
        P0=0;
        while(1){
                input=ser_in();
                if((input==13)|(input==10)){
                        counter=0;
                        device=buffer[2];
                        status=buffer[4]&0x01;
                        switch (device){
                                case '1': Relay_1=status;break;
                                case '2': Relay_2=status;break;
                                case '3': Relay_3=status;break;
                                case '4': Relay_4=status;break;
                                case '5': Relay_5=status;break;
                                case '6': Relay_6=status;break;
                                case '7': Relay_7=status;break;
                                case '8': Relay_8=status;break;
                        }    }
                buffer[counter++]=input;
```

```
                ser_chr(input);

                if (input==13)ser_chr(10);

            }

} //end main
```

# Appendix B

Android app is developed in Eclipse SDK. The layout code for the application is as under:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hotelautomationsystem"
    android:versionCode="1"
    android:versionName="1.0" >
     <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
        <uses-permission
        android:name="android.permission.INTERNET"/>
            <application
        android:allowBackup="true"
        android:icon="@drawable/icon"
        <activity
            android:name="com.example.hotelautomationsystem.Start"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.example.hotelautomationsystem.Login"
            android:label="@string/title_activity_login"
```

```
    android:theme="@android:style/Theme.NoTitleBar" >
</activity>
<activity
    android:name="com.example.hotelautomationsystem.MainMenu"
    android:label="@string/title_activity_menu"
    android:theme="@android:style/Theme.NoTitleBar" >
</activity>
    android:name="com.example.hotelautomationsystem.MainOffice"
    android:label="@string/title_activity_main_office"
    android:theme="@android:style/Theme.NoTitleBar" >
</activity>
    android:name="com.example.hotelautomationsystem.ConferenceRoom"
    android:label="@string/title_activity_conference_room"
    android:theme="@android:style/Theme.NoTitleBar" >
</activity>
<activity
    android:name="com.example.hotelautomationsystem.Parking"
    android:label="@string/title_activity_parking"
    android:theme="@android:style/Theme.NoTitleBar" >
</activity>
<activity
    android:name="com.example.hotelautomationsystem.Settings"
    android:label="@string/title_activity_settings"
    android:theme="@android:style/Theme.NoTitleBar" >
</activity>
<activity
    android:name="com.example.hotelautomationsystem.TeaBar"
```

```
            android:label="@string/title_activity_tea_bar"
android:theme="@android:style/Theme.NoTitleBar" >
        </activity>
        <activity
            android:name="com.example.hotelautomationsystem.Animation"
            android:label="@string/title_activity_animation"
            android:theme="@android:style/Theme.NoTitleBar" >
        </activity>
        <activity
            android:name="com.example.hotelautomationsystem.CommonClass"
            android:label="@string/title_activity_common_class"
             android:theme="@android:style/Theme.NoTitleBar" >
        </activity>
        <activity
            android:name="com.example.hotelautomationsystem.MainActivity"
            android:label="@string/title_activity_common_class"
             android:theme="@android:style/Theme.NoTitleBar" >
        </activity>
    </application>
```

# Appendix C: Datasheets

## AT 89c51, M8929, TX-2B and RX-2B Datasheets

**Flash Programming Modes**

| Mode | | RST | $\overline{PSEN}$ | ALE/$\overline{PROG}$ | $\overline{EA}$/$V_{PP}$ | P2.6 | P2.7 | P3.6 | P3.7 |
|---|---|---|---|---|---|---|---|---|---|
| Write Code Data | | H | L | | H/12V | L | H | H | H |
| Read Code Data | | H | L | H | H | L | L | H | H |
| Write Lock | Bit - 1 | H | L | | H/12V | H | H | H | H |
| | Bit - 2 | H | L | | H/12V | H | H | L | L |
| | Bit - 3 | H | L | | H/12V | H | L | H | L |
| Chip Erase | | H | L | (1) | H/12V | H | L | L | L |
| Read Signature Byte | | H | L | H | H | L | L | L | L |

Note: 1. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

**Figure 3.** Programming the Flash

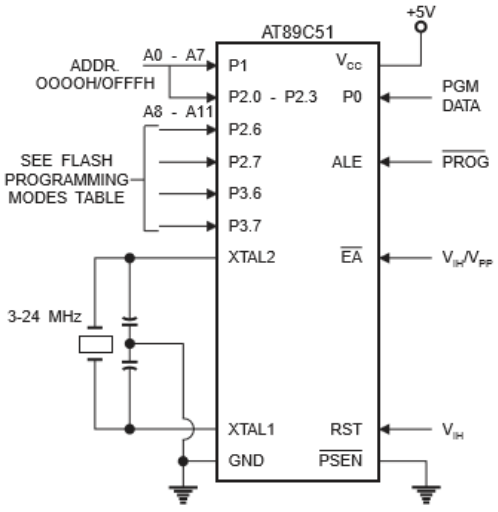
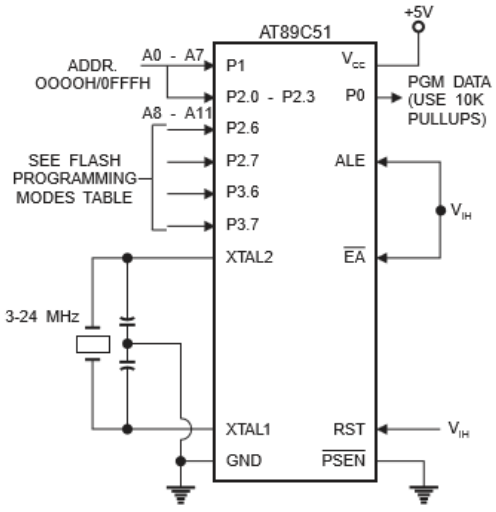



**Figure 4.** Verifying the Flash



## Flash Programming and Verification Characteristics

$T_A$ = 0°C to 70°C, $V_{CC}$ = 5.0 ±10%

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $V_{PP}$[1] | Programming Enable Voltage | 11.5 | 12.5 | V |
| $I_{PP}$[1] | Programming Enable Current | | 1.0 | mA |
| $1/t_{CLCL}$ | Oscillator Frequency | 3 | 24 | MHz |
| $t_{AVGL}$ | Address Setup to $\overline{PROG}$ Low | $48t_{CLCL}$ | | |
| $t_{GHAX}$ | Address Hold after $\overline{PROG}$ | $48t_{CLCL}$ | | |
| $t_{DVGL}$ | Data Setup to $\overline{PROG}$ Low | $48t_{CLCL}$ | | |
| $t_{GHDX}$ | Data Hold after $\overline{PROG}$ | $48t_{CLCL}$ | | |
| $t_{EHSH}$ | P2.7 ($\overline{ENABLE}$) High to $V_{PP}$ | $48t_{CLCL}$ | | |
| $t_{SHGL}$ | $V_{PP}$ Setup to $\overline{PROG}$ Low | 10 | | µs |
| $t_{GHSL}$[1] | $V_{PP}$ Hold after $\overline{PROG}$ | 10 | | µs |
| $t_{GLGH}$ | $\overline{PROG}$ Width | 1 | 110 | µs |
| $t_{AVQV}$ | Address to Data Valid | | $48t_{CLCL}$ | |
| $t_{ELQV}$ | $\overline{ENABLE}$ Low to Data Valid | | $48t_{CLCL}$ | |
| $t_{EHQZ}$ | Data Float after $\overline{ENABLE}$ | 0 | $48t_{CLCL}$ | |
| $t_{GHBL}$ | $\overline{PROG}$ High to $\overline{BUSY}$ Low | | 1.0 | µs |
| $t_{WC}$ | Byte Write Cycle Time | | 2.0 | ms |

Note: 1. Only used in 12-volt programming mode.

# Appendix D

Code for configuring our server:

```
#include <homelab/usart.h>

#include <homelab/module/lcd_gfx.h>

#include <homelab/delay.h>

#include <homelab/pin.h>

//Adding ZigBee module support

#include <homelab/module/zigbee.h>

// Determination of the USART interface

usart port = USART(1);

// LED and button pin setup

pin leds[3] = { PIN(C, 5), PIN(C, 4), PIN(C, 3) };

pin buttons[3] = { PIN(C, 0), PIN(C, 1), PIN(C, 2) };

// Max numbers of ZigBee modules in network + broadcast address +1

#define ZIGBEE_MAX_NODES 4

int8_t wait_button(uint16_t ms);

// Address array of other Zigbee modules found in network

zigbee_node_t nodelist[ZIGBEE_MAX_NODES];
```

```
int main(void)
{
   uint8_t adr = 0;       //Acquired module order list

   // LED and buttons I/O ports configuration
   for (int i=0; i<3; i++)
   {
      pin_setup_output(leds[i]);
      pin_setup_input(buttons[i]);
   }

   // Configuration of USART interface for communication whit ZigBee module
   usart_init_async(port,
                    USART_DATABITS_8,
                    USART_STOPBITS_ONE,
                    USART_PARITY_NONE,
                    USART_BAUDRATE_ASYNC(9600));

   // LCD display initalization
   lcd_gfx_init();

   // Clear LCD
   lcd_gfx_clear();
```

```
// Turning back light ON
lcd_gfx_backlight(true);

//Wait until other ZigBee modules are searched from area near by
lcd_gfx_clear();
lcd_gfx_goto_char_xy(0, 0);
lcd_gfx_write_string("  ZigBee demo");

lcd_gfx_goto_char_xy(0, 1);
lcd_gfx_write_string("Searching...");

// Searching other ZigBee modules
// fills nodelist array with found modules info
zigbee_find_nodes(port, nodelist, ZIGBEE_MAX_NODES);

//lcd_gfx_clear();
lcd_gfx_goto_char_xy(0, 1);
lcd_gfx_write_string("Found:  ");

// Displays the list of found modules on LCD
// (on what line to write, where takes addr., how many max)
zigbee_lcd_show_nodes(2, nodelist, ZIGBEE_MAX_NODES);
hw_delay_ms(3000);
lcd_gfx_clear();
```

```
// Displaying connecting on LCD
lcd_gfx_goto_char_xy(0, 0);
lcd_gfx_write_string("Connecting...");
lcd_gfx_goto_char_xy(0, 1);
//Displays only 8 last digit form the address
lcd_gfx_write_string((nodelist + adr)->address64l);

// Confederate ZigBee to send info for chosen ZigBee module's node,
// in this case to first [0]
// (What port is using, where takes the address)
zigbee_set_destination(port, &nodelist[adr]);

// Displays info on LCD
lcd_gfx_goto_char_xy(0, 0);
lcd_gfx_write_string("  ZigBee demo");

lcd_gfx_goto_char_xy(0, 1);
lcd_gfx_write_string("Button1: LED1");

lcd_gfx_goto_char_xy(0, 2);
lcd_gfx_write_string("Button2: LED2");

// Save the state of previous button to avoid multiple button pushes at once.
// At first is -1 ie none of the buttons is pushed.
```

```
int8_t previousButton = -1;

// Endles-loop for communicating between modules
while (true)
{
   int8_t button;      //variable for saving button pushes
   // wait 1 millisecond for button
   button = wait_button(1);

   // if in last cycle button wasn't pressed but now is
   if (previousButton == -1 && button != -1)
   {
      // Convert button's index by adding A
         // and sent it to other modules
      // A is for first button, B for second and so on
      usart_send_char(port, 'A' + button);
   }

   // read from USART
   if (usart_has_data(port))
   {
      // Read bait, convert to leds array index
         //and change the output.
      pin_toggle(leds[usart_read_char(port) - 'A']);
   }
```

```
        // remember what button was pressed
        previousButton = button;
    }
}

// Wait for button to be pressed for ms milliseconds.
//If button is pressed returns the queue number of the button
int8_t wait_button(uint16_t ms)
{
    // By default -1 means, that no button is pressed.
    int8_t button_nr = -1;
    uint16_t counter = 0;
    do
    {
        // check if one of the buttons is pressed
        for (uint8_t i=0; i<3; i++)
        {
            if (!pin_get_value(buttons[i]))
            {
                button_nr = i;
                break;
            }
        }
```

```
        // wait for 1 millisecond
        hw_delay_ms(1);

        // increase millisecond counter
        counter++;
    } while (button_nr == -1 && (counter < ms));

    return button_nr;
}
```

# References

[1]: http://www.flickr.com/

[2] http://www.telegesis.com/

[3] http://www.glyn.co.nz/HF_Opto_Telegesis.htm

[4] http://www.homeseer.com/automation_basics.htm

[5]Iscold, P., Pereira, G. A. S., and Torres, L. A. B. Development of a automation system. IEEE Transactions on Electronic Systems, Vol. 46, 1 (Jan. 2010), 335–348.

[6] Cortes, Francisco Javier (2002) "Convergencia TCP/IP en Home Networking". Tesis Doctoral en Electrónica Comunicaciones. Instituto Tecnológico de Aragón (ITA) Advanced services department. December 2002.

[7] Myers, Brad A. et al (2004) "Taking handheld devices to the next level". IEEE Computer Society, December 2013.

[8] Agent-oriented, multimedia, interactive services in home automation Luigi Ceccaroni and Xavier Verdaguer).

[9] HOME AUTOMATION: RESEARCH ISSUES Dr Leslie

[10] (www.books.google.lk/Wireless IP and building the mobile internet by Sudhir Dixit, Ramjee Prasad)

[11] Bengtsson Department of Computer Science and Electrical Engineering Luleå University of

Technology, SE-971 87 Luleå, Sweden

[12] European Standard for Home and Building Electronic Systems, http://www.konnex.org/ (Accessed 22 November 2013).

[13] A. Roychowdhury, A. Moyer. "Instant Messaging and Presence for SIP Enabled Networked Appliances",2001 http://www.voip.nce.ufrj.br/courses/graduacao/lc4/Instant Messaging and Presence for SIP Enabled Networked Appliances.pdf (Accessed 10 February 2014)

[14] Universal Pug and Play, http://www.plug-n-play-technologies.com/ [15]IEEE AESS UAV Team. AESS UAV Team, Missouri University of Science and Technology (2012), http://www.aessuav.org/, last access Jan. 23, 2013.

[15] C. Chewar and D. S. McCrickard. Adapting uems for notification systems. Accepted to Design and evalua tion of notification interfaces for ubiquitous computing, Ubicomp 2002 workshop 9, September 2002.

[16] E. H. M. van Dantzich, D. Robbins and M. Czerwinski. Scope: Providing awareness of multiple notifications at a glance. In Proceedings of the 6th Intl Working Conf. on Advanced Visual Interfaces (AVI ’02), page To Appear. ACM Press, May 2002.