

Low Cost Embedded System/Android Based Smart Home Automation System Using Wireless Networking

¹Subhamay Sarker, ²Mithun Chakraborty and ²Anindita Banerjee

¹*Asst. Professor, E.C.E. Deptt., Dream Institute of Technology, Kolkata, India, subhamay.gemini@gmail.com*

²*Asst. Professor, E.C.E. Deptt., Surendra Institute of Engineering & Management, Siliguri, India, mithunchakraborty03@gmail.com*

²*Asst. Professor, E.C.E. Deptt., Dream Institute of Technology, Kolkata, India, ani.eie09@gmail.com*

Abstract

The past decade has seen significant advancement in the field of consumer electronics. Various 'intelligent' appliances such as cellular phones, air-conditioners, home security devices, home theatres, etc. are set to realize the concept of a smart home. They have given rise to a Personal Area Network (PAN) in home environment, where all these appliances can be interconnected and monitored using a single controller. In this paper, we make use of Home Automation techniques to design and implement a remotely controlled, energy-efficient and highly scalable Smart Home with basic features that safeguard the residents' comfort and security. Busy families and individuals with physical limitation represent an attractive market for home automation and networking. This system will also assist and provide support in order to fulfill the needs of elderly and disabled in home. A wireless home network that does not incur additional costs of wiring would be desirable. Bluetooth technology, which has emerged in late 1990s, is an ideal solution for this purpose. This paper describes an application of Bluetooth technology in home automation and networking environment. It proposes a network, which contains a host controller (smart phone), wireless network and several client modules (home appliances). The client modules communicate with the host controller through Bluetooth devices. As a central controller; we are using a standalone embedded module that communicates with an Android application, our user interface. The prototype implementation of the proposed system is evaluated based on the criteria considered after the requirement analysis for an adequate home automation system.

Keywords— Home automation; PAN; Bluetooth technology; Wireless network; Embedded Systems; Android

I. INTRODUCTION

The “Home Automation” concept is a brand new approach of wirelessly controlling the home appliances. The terms “Smart Home”, “Intelligent Home” is associated with this type of system and has been used to introduce the concept of networking appliances and devices in the house. Home automation Systems (HASs) represents a great research opportunity in creating new fields in engineering, architecture and computing. There are certain issues involved in the design of a home automation system. The system should be scalable, so that new device can easily be integrated into it. It should provide a user-friendly interface on the host side, so that the devices can be setup, monitored and controlled. A cost effective system would qualify it for mass adoption.

Neng-Shiang Ling has presented architecture for home automation [1] where the system is based on a dedicated network. This system depicted how to solve home automation problems at the software level and no hardware aspects were included. Yavuz and Hasan [2] presented a telephone and PIC based remote control system. Other studies such as those presented in [3] [4] give examples of web based home automation system. Another PC based home automation system for appliances control was proposed by Sriskanthan [5]. However, the system cannot be controlled by a mobile/cell phone. R. Piyare proposed a Bluetooth based home automation system using cell phone [6]. Due to the advancement of wireless technology, there are several different connections are introduced such as GSM, WIFI, ZIGBEE, and Bluetooth. Each of the connection has their own unique specifications and applications. Among the four popular wireless connections that often implemented in HAS project, Bluetooth is being chosen with its suitable capability. The Bluetooth wireless technology is set to revolutionize the way people perceive digital devices in our homes and office environment. Now they are no longer just the individual devices; instead, with the embedded Bluetooth technology, they form a network in which appliances can communicate with each other [7,8]. This wireless technology is especially useful in home environment, where there exist hardly any infrastructure to interconnect intelligent appliances. It could be suitably used for home automation in a cost effective manner. Operating over unlicensed, universally available frequency of 2.4 GHz, it can link digital devices within a range of 10 m (expandable to 100 m, by increasing the transmitted power) at the speed of 1 Mbps. In addition, a Bluetooth master device is able to connect up to 7 devices in a “Piconet” [9,10], but in this paper we present the design of point to point Piconet topology in a modified approach which increases the scalability of the system to a great extent. On the other hand most of the current laptop/notebook or cell phones come with built-in Bluetooth adapter. It will indirectly reduce the cost of this system. Building upon this theme; we propose a home automation system based on Bluetooth technology.

The paper has been structured as follows. Section II of this paper explains the hardware architecture and implementation of the overall system. It also describes the

salient features of the user interface and Bluetooth interface to setup communication between Smart Phone and the standalone Embedded module which act as the host controller for the appliances. Section III explains the Software Development part of the project. This includes development of a user friendly Android application and a firmware for the microcontroller. Section IV discusses about the final setup. Finally some recommendations regarding further research in this area are discussed.

II. SYSTEM ARCHITECTURE AND IMPLEMENTATION

As shown in figure 1, the proposed system is comprised of multiple stand-alone embedded modules dedicated to each of the rooms. On the other hand the user interface is present in a Smart Phone, which is an Android application specifically designed and built for this project. The Android application communicates with the embedded modules wirelessly using Bluetooth technology. The system is highly scalable and cost effective as the main complexity of the design lies within the Android application. The details about the android application and the microcontroller programming will be discussed in section II, B.

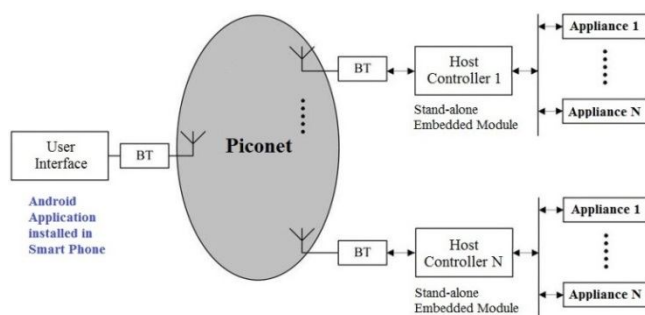


Fig. 1. Overall system architecture

Key hardware components that make up the home automation system are the Smart Phone, the Microcontroller Board, the Bluetooth module connected to the Microcontroller board and relay switches that drive the electrical appliances. The other components that are also present include a 16x2 char LCD display, a Real Time Clock (DS1307), a Temperature sensor (LM35) and a Light Dependent Resistor (LDR).

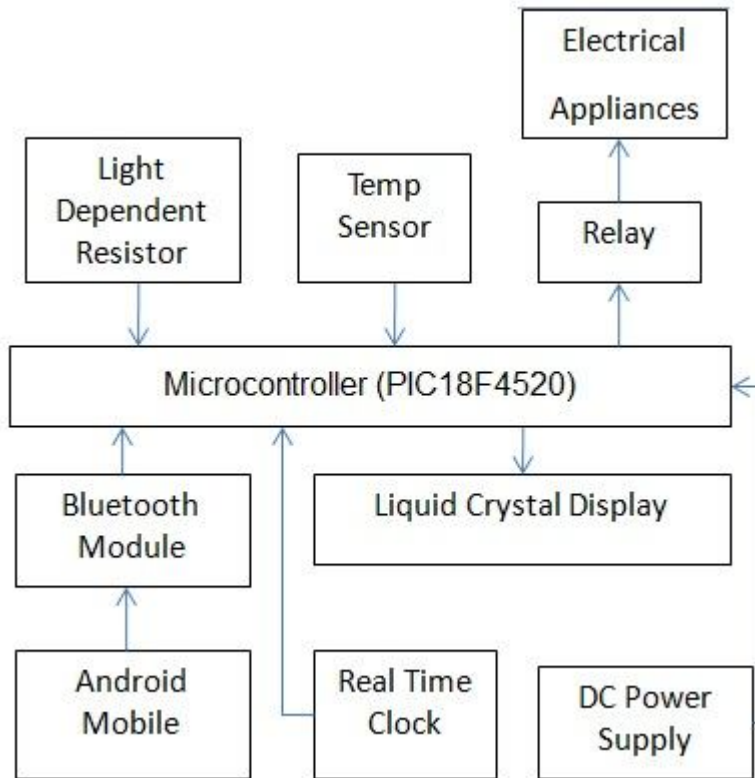


Fig. 2. **Block diagram of the hardware architecture.**

A. *Embedded Module*

The heart of the embedded module is a Microchip PIC18F4520, which is a 40 pin microcontroller [11]. This particular microcontroller is chosen because of its features which are essential for this project. The features are as follows: C compiler optimized architecture, Linear program memory addressing to 32 Kbytes, Linear data memory addressing to 1.5 Kbytes, Four Crystal modes, up to 40 MHz, Master Synchronous Serial Port (MSSP) module (supporting 3-wire SPI™ and I²C Master and Slave Modes), Enhanced Addressable USART module, 10-bit and up to 13-channel Analog-to-Digital, Three external interrupt pins and 4-Timer modules. The embedded module controls/monitors the status of all the other peripherals connected to it and communicates with the user through Bluetooth. It also contains an LCD which displays the real-time information like time, temperature in a particular room, light intensity, and the status of the relay switches which corresponds to each of the appliances. Apart from these components, each of the embedded modules contains a dedicated DC power supply unit with fixed 5V and 12V supplies. Figure 3 shows the circuit design of the stand-alone embedded module. Designed using ExpressSCH software. Figure 4 shows the equivalent PCB design of the embedded module developed using ExpressPCB.

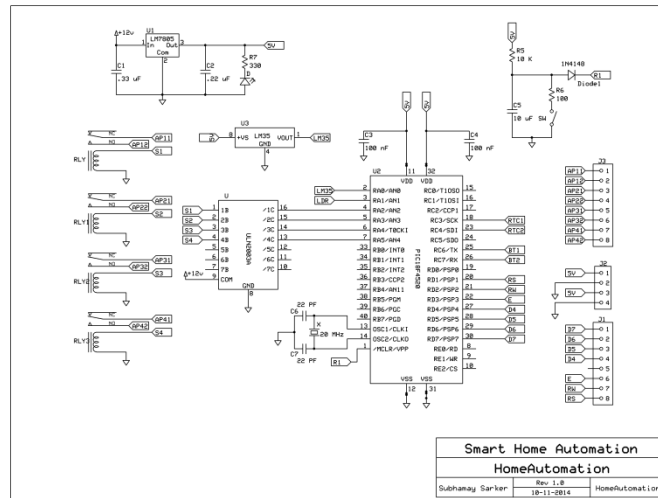


Fig. 3. Circuit Design of the embedded module

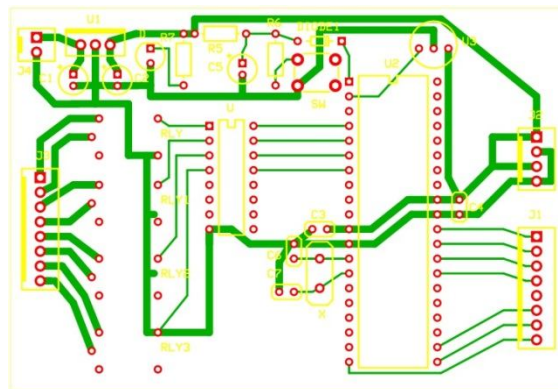


Fig. 4. PCB Design of the embedded module

B. Bluetooth Module & Wireless Network

The Bluetooth module allows us to wirelessly transmit and receive data. The module that we're using is based on the Bluetooth V2.0 protocol and is having a range of 10 meters operating at frequency of 2.4GHz with a maximum data exchange rate of 2.1Mbps. For this project we have used HC-06 Bluetooth module. This module permits any microcontroller with a standard RS232 serial port to communicate with a PC or a Smartphone equipped with a Bluetooth Master module. Its main specifications are: Bluetooth number: JY-MCU-HC-06, surface-mount with integrated antenna. Operating Voltage: 5 volt, reduced to 3.3 volts, @ 8 mA. Default baud rate: 9600 bps. Default PIN: 1234, Default name: BlueBolt. Class: 2, with up to 10 meter coverage. Figure 5 shows the HC-06 BT module used.

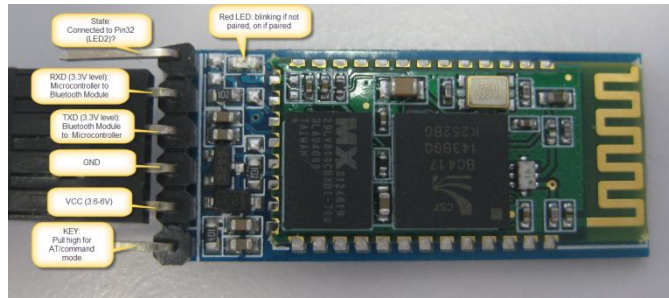


Fig. 5. **JY-MCU-HC-06 Bluetooth module.**

To make Bluetooth as robust as possible the operating band is divided into 1MHz-spaced channels, each signalling data at 1Mb/s. This is achieved by using GFSK (Gaussian Frequency Shift Keying) modulation scheme. After each packet, both devices re-tune their radios to a different frequency, effectively hopping from radio channel to radio channel; this is known as FHSS (Frequency Hopping Spread Spectrum). Therefore if a transmission is compromised by interference on one channel, the retransmission will always be on a different (hopefully clear) channel. Each Bluetooth time slot lasts $625\mu\text{s}$ (625 microseconds) and generally devices hop once per packet, giving a hop rate of 1600 hops/second.

If devices are to hop to new frequencies after each packet, there has to be some sort of agreement between the devices. This is where the concept of Master and Slave appears; the Master is the Bluetooth device that sets the frequency hopping sequence. The Slave synchronises to the Masters in time and frequency by following the Master's frequency hopping sequence. Every Bluetooth device has a unique Bluetooth device address and a 28-bit Bluetooth clock. The baseband part of the Bluetooth System uses a special algorithm, which calculates the frequency hop sequence from the master's clock and device address. In addition to controlling the frequency hop sequence, the Master controls when Slaves are to transmit using Time Division Multiplexing (TDM).

When there is just one Master and one Slave the system is called a Point to Point connection. When many Slaves are connected to one Master, the system is called a Point to Multipoint. Both these types are referred to as a Piconet (shown in figure 6) and all follow the frequency hopping sequence of the Master. The Slaves in the Piconet only have links to the Master and no direct links between Slaves.

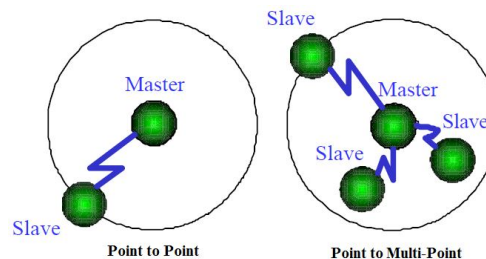


Fig. 6. **Point to point and point to multipoint piconets.**

The specification limits the number of Slaves in a Piconet to seven, however larger networks can be set up using many Piconets to communicate with one another, and this is known as a Scatternet[9]. In a Scatternet a device can be a Master in one Piconet and a Slave in another, or a Slave in both, but never a Master of more than one Piconet. But this in turn creates a complex network and also affects the scalability of the system. In this paper we present the use of a modified point to point Piconet connection, which not only brings simplicity in design but also makes the system highly scalable and cost effective. Further details on the operation of Bluetooth network is explained in Section III, B.

C. *Relay, RTC, LCD and LDR*

The relay section that we're using comprises of 4 relays that can each handle a load of up to 6A. The relays are driven using a ULN2003 high voltage high current Darlington array

IC. The 16x2 char LCD display allows us to see the time (derived from the RTC, DS1307), the temperature (given by the LM35) and also the light intensity (extracted from the LDR). The light intensity monitoring is used to switch on an emergency light if the intensity of light goes lower than the basic visible level. We've also implemented a mechanism where the entire system can be halted in case of an emergency by manually pressing a push button. Different home appliances are connected to the digital ports of the PIC18F4520 microcontroller. We've used four lamps each of 10W running at 230V AC.

D. *Smart Phone*

For this project any Smart Phone preloaded with Android 2.2 and above can be used to operate the system. The Bluetooth adapter present in the phone can be programmatically configured to send data commands to the Bluetooth module on the stand-alone embedded systems that would in turn control the electrical appliances. We have built an android application (Smart HAS v1.0.apk) which needs to be installed first. When the application is opened it will automatically connect to the particular HC-06 BT device and the application starts to communicate with the embedded module.

III. SOFTWARE DEVELOPMENT

One of the main features of this system is its modularity. The hardware complexity of the system has been decreased by the use of Android Application as the user interface. The use of stand-alone embedded modules also makes the design simple and scalable. The main complexity of the design lies within the software development part.

A. *Microcontroller Program*

In this project we have used CCS C compiler, v4.068 in order to write the program for PIC18F4520 microcontroller [12]. The program was written and developed in Microchip MPLAB IDE v8.73a [13]. After building the project in the IDE, the HEX file was burn in to the PIC18F4520 Chip using Microchip PICKit 2 programmer [14].

The microcontroller program is written to perform following specific tasks: simultaneously scans the analog input channels ‘AN2’ & ‘AN3’ connected to the LDR and the temperature sensor respectively; reads the RTC using I²C protocol for current time; it receives the user command through RS232 serial port and operates specific relay switches to turn ON/OFF home appliances; and then finally the program displays all the current information on the LCD using parallel ports. All the above tasks are arranged and executed at the occurrence of a Timer overflow Interrupt, which keeps on repeating after a predefined interval of time. In this project we have used TIMER1 of PIC18F4520 for the purpose. The ‘main()’ function of the program contains the initializing codes for LCD, RTC, Timer, ADC and an infinite loop for receiving the user command from smart phone and turning ON/OFF the home appliances. A display code is also included in the infinite loop for updating the real-time information on LCD. The detailed flowchart of the program is explained below in figure 7.

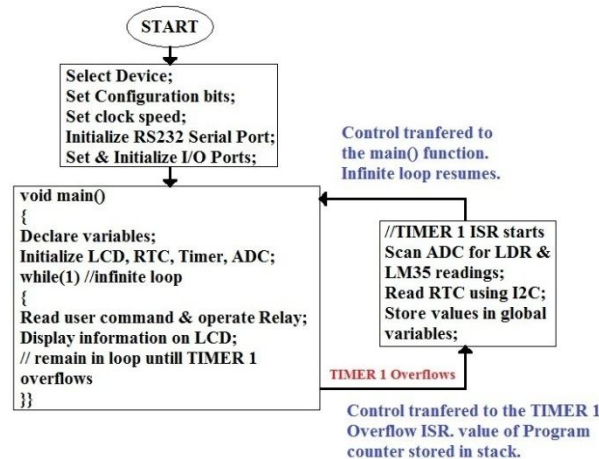


Fig. 7. Flowchart of Microcontroller program.

B. Android Application

Android is a software stack for mobile devices that include an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. By providing an open development framework, Android offers developers the ability to build extremely rich and innovative applications. Developers have full access to the same framework APIs used by the core applications. Android includes a set of C/C++ libraries used by various components of the Android system. They include System C library, Media library, Surface Manager, LibWebCore, SGL, SQLite, FreeType and 3D libraries. Android applications are written in Java programming language. The Android SDK compiles the code along with any data and resource files into an Android package, an archive file with an .apk file extension. All the code in a single .apk file is considered to be one application and

is the file that Android powered devices use to install the application.

Once installed on a device, each Android application lives in its own security sandbox. Some important application fundamentals are:

- The Android operating system is a multi-user Linux system where each application is a different user.
- By default, the system assigns each application a unique user ID. The system sets permission for all the files in an application so that only the user ID assigned to that application can access them.
- Each process has its own virtual machine, so an application's code runs in isolation from other applications.
- Every application runs its own Linux process.

The Android platform [15] includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs.

These APIs let applications to wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features. Using the Bluetooth APIs, an Android application can perform the following:

- Scan for other Bluetooth devices.
- Query the local Bluetooth adapter for paired Bluetooth devices.
- Establish RFCOMM channels.
- Connect to other devices through service discovery.
- Transfer data to and from devices.
- Manage multiple connections.

The program flow chart that scans; establishes a connection and then sends data to our Bluetooth enabled embedded module is shown in figure 8. As the application starts it opens up the main menu which comprises of 4 buttons each representing a particular room in the house, shown in figure 9.(b). The user then clicks the desired button which selects a particular room. As shown in the flowchart, as soon as any one of the buttons is pressed we first get hold of the local Bluetooth adapter present in the smartphone and establish a connection with our external Bluetooth module that is connected to the microcontroller unit. The Bluetooth modules used in this project (HC-06) have a unique MAC address which is used by the application to identify and establish connection with that particular device.

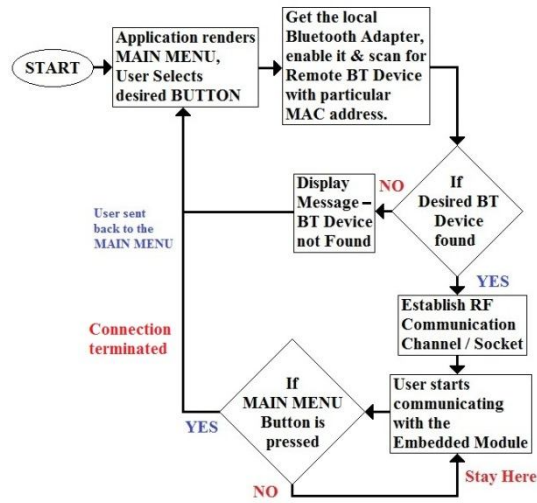


Fig. 8. **Android Application program flowchart.**

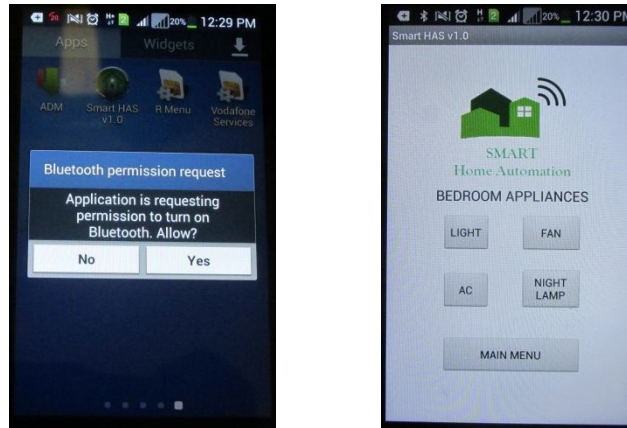
Once the connection is successfully established, the application opens up the ‘CONTROL MENU’ which displays 5 buttons, 4 of which correspond to home appliances and the 5th one is the ‘MAIN MENU’ button, as shown in figure 9.(d). The four buttons act as toggle switches which are used to turn ON/OFF the appliances. The use of toggle switch reduces the number of buttons. Pressing a particular toggle button the user sends command from the smartphone to the embedded module which is then used by the microcontroller to decide which appliances to switch ON/OFF. On pressing the ‘MENU’ button, user is sent back to the main menu and the existing Bluetooth connection is terminated. The same algorithm is repeated if a different button is pressed, but the connection is established with a Bluetooth device having different MAC address. One of the important features of our system is to simplify user interaction with the system by hiding all the process in the background while giving the user only the bare essentials. Figure 9 indicates some examples of our graphical user interface.



(a) Application Icon.



(b) Main Menu.



(c) Connecting to BT Device. (d) Control Menu.

Fig. 9. User Application Interface Screens.

IV. FINAL SETUP

Figure 10 shows the prototype setup that was made by the authors:



Fig. 10. Prototype working design

V. CONCLUSION

In this paper, we've introduced the design and implementation of a low cost Android based Smart home automation system. This system can be easily manufactured on a large scale for mass adoption owing to its simplicity and ease of design. Another advantage of the system is that the application software is based on Android, which today has the largest smartphone base. With improvements in technology and the fact that Android is free and open source, cheap Smartphones can be used as the controller in our project, making the overall system cost affordable for mass adoption.

Further enhancements can be implemented on the system such as the integration of an intelligent controller that controls the various home devices based on

various factors like humidity, temperature etc.. Also, we can easily integrate Google's latest offering, Google Cloud Messaging to control our home systems from the Internet, thus making it possible to monitor our home appliances from anywhere in the world [16].

REFERENCES

- [1] E. Yavuz, B. Hasan, I. Serkan and K. Duygu. "Safe and Secure PIC Based Remote Control Application for Intelligent Home". International Journal of Computer Science and Network Security, Vol. 7, No. 5, May 2007.
- [2] K.Tan, T.Lee and C.Yee Soh. "Internet-Based Monitoring of Distributed Control Systems-An Undergraduate Experiment". IEEE Transaction on Education, Vol. 45, No. 2, May 2002.
- [3] N. Sriskanthan and Tan Karand. "Bluetooth Based Home Automation System". Journal of Microprocessors and Microsystems, Vol. 26, pp.281-289, 2002.
- [4] R.Piyare, M.Tazil. "Bluetooth based home automation system using cell phone". 2011 IEEE 15th International Symposium on Consumer Electronics.
- [5] Official page for scmRTOS development: (<http://scmrtos.sourceforge.net/ScmRTOS>)
- [6] Official Android developer website: <http://www.developer.android.com>
- [7] The official Bluetooth website from Bluetooth SIG www.Bluetooth.com, Date viewed: March 21, 2001, Bluetooth Specification Version 1.1.
- [8] Bluetooth Committee, Specifications of the Bluetooth System (Core), December 1999, V1.0B.
- [9] J. Mander and D. Picopoulos, "Bluetooth Piconet Applications". pp. 1-25.
- [10] K. Elissa, "RTOS based Home Automation System using Android". International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 480 – 484 (2013)
- [11] Microchip Technology Inc, PIC18F2420/2520/4420/4520 Data Sheet, 28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology. www.microchip.com.
- [12] CCS C Compiler Manual PCB / PCM / PCH, Custom Computer Services, May 2014. www.ccsinfo.com.
- [13] Microchip Technology Inc, MPLAB® IDE User's Guidewith MPLAB Editorand MPLAB SIM Simulator. www.microchip.com.
- [14] Microchip Technology Inc, PICkit. 2 Programmer/Debugger User's Guide. www.microchip.com.
- [15] Reto Meier, "ProfessionalAndroid™ ApplicationDevelopment". Published by Wiley Publishing, Inc. www.wrox.co.
- [16] Google Cloud Messaging for Android. <https://developer.android.com/google/gcm/index.html>.