*International Journal of Electronics and Computer Science Engineering*

*Available Online at www.ijecse.org* ISSN- 2277-1956

# Home Automation System (HAS) using Android for Mobile Phone

Sharon Panth [1], Mahesh Jivani [2]

[1] *Shri M & N Virani Science College, Rajkot-360005 (Gujarat) India*
[2] *Department of Electronics, Saurashtra University, Rajkot-360005 (Gujarat) India*
[1] *Email- sharon.panth20@gmail.com*
[2] *Email- mnjivani@gmail.com*

**Abstract – Automation of the surrounding environment of a modern human being allows increasing his work efficiency and comfort. There has been a significant development in the area of an individual's routine tasks and those can be automated. In the present times, we can find most of the people clinging to their mobile phones and smart devices throughout the day. Hence with the help of his companion – a mobile phone, some daily household tasks can be accomplished by personifying the use of the mobile phone. Analyzing the current smart phone market, novice mobile users are opting for Android based phones. It has become a second name for a mobile phone in layman terms. Home Automation System (HAS) has been designed for mobile phones having Android platform to automate** an **8 bit Bluetooth interfaced microcontroller which controls a number of home appliances like lights, fans, bulbs and many more using on/off relay. This paper presents the automated approach of controlling the devices in a household that could ease the tasks of using the traditional method of the switch. The most famous and efficient technology for short range wireless communication- Bluetooth is used here to automate the system. The HAS system for Android users is a step towards the ease of the tasks by controlling one to twenty four different appliances in any home environment.**

Keywords – **Home Automation System, Android, Microcontroller, Bluetooth, Mobile Phone, Appliance**

## I. INTRODUCTION

Today's homes require sophistication control in its different gadgets which are basically electronic appliances [1]. This has revolutionized the area of home automation with respect to an increased level of affordability and simplicity through the integration of home appliances with smart phone and tablet connectivity. Smart phones are already feature-perfect and can be made to communicate to any other devices in an ad hoc network with a connectivity options like Bluetooth [2]. With the advent of mobile phones, Mobile applications development has seen a major outbreak. Utilizing the opportunity of automating tasks for a smart home, mobile phone commonly found in normal household can be joined in a temporary network inside a home with the electronic equipments. Android, by Google Inc. provides the platform for the development of the mobile applications for the Android devices [3]. Home automation system is a mobile application developed using Android targeting its vast market which will be beneficial for the masses. According to the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, Android maintained its leadership position in global market share [4]. Bluetooth is a short-range wireless communication technology that comes in handy as the solution while communicating over an ad hoc network environment like the home environment for connecting the home appliances with the mobile phones [5]. Bluetooth works over 2.4 GHz frequency range up to the range of 100 m with 1 Mbps speed, providing a safe and efficient solution for controlling home automation [2].

## II. SYSTEM ARCHITECTURE

The Home Automation System (HAS) was developed using Java Me [6] and MoSync [7] during the course of research and now a User Interfaced (UI) Android Application program implemented on an Android based Bluetooth enabled mobile phone, and an 8 bit microcontroller based relay driver circuit with Serial Bluetooth Module, which is able to communicate with the Home-

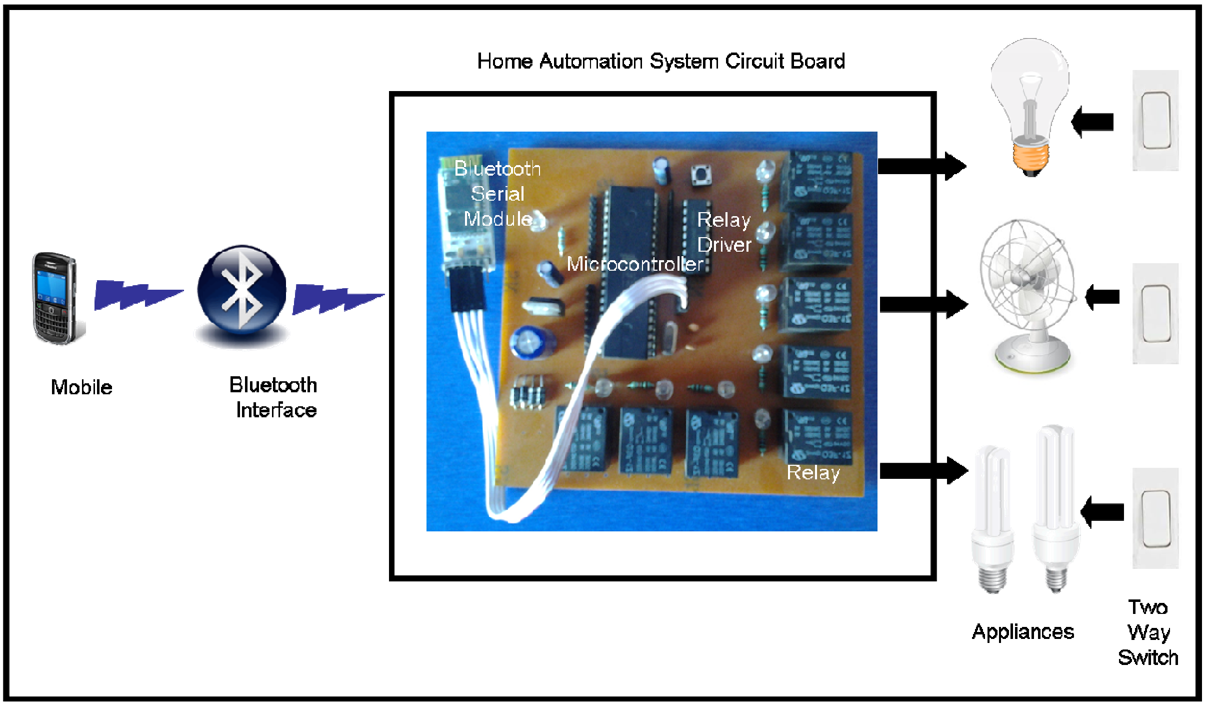


Figure 1. System Architecture

-Appliances over Bluetooth link. The system is based on serial data transmission using Bluetooth wireless communication in order to facilitate the appliances control in a HAS. This system ensures a secured exchange of data on wireless communication. It also supports conventional ON/OFF system of appliances. A user interface (UI) on the Android enabled mobile phone offers system connection and control utilities. ULN 2803 relay driver [8] and Serial Bluetooth Module from TINY OS [9] as well as Keil µvision IDE [10] for compiling C Language Code and uCflash+ programmer [11] for burning HEX file into microcontroller were used for the development. An ATMEL 89C51, 8 bit microcontroller [12] was used as an embedded relay controller.

## III. DEVELOPMENT PLATFORM

This section describes the technologies used for developing the mobile phone application of the Home Automation System. The mobile phone application development presented in this paper is based on the following technologies: Android [3], Bluetooth [2]. Android is a platform developing and deploying android based applications on mobile devices supporting it. Bluetooth has its own benchmark as a wireless communication technology for permanent and itinerant devices. Combining the power of Bluetooth, the best known and used wireless technology for short range communication provide a facility to create Android based mobile applications using the Bluetooth Wireless Technology. The Android Bluetooth specification [13], added in API level 5 of Android APIs.

### *A. Android*

Android is everywhere. Present days Phones, Tablets, TVs and set-top boxes powered by Google TV. Soon, Android will be in cars, in in-flight entertainment systems on planes, and even in robots [13]!

Android was originally created by Andy Rubin as an operating system for mobile phones, around the dawn of this twenty-first century. In 2005, Google acquired Android Inc., and made Andy Rubin the Director of Mobile Platforms for Google. Many think the acquisition was largely in response to the emergence of the Apple iPhone around that time; however, there were enough other large players, such as RIM Blackberry, Nokia Symbian, and Microsoft Windows Mobile, that it seemed a salient business decision for Google to purchase the talent and intellectual property necessary to assert the company into this emerging space, which has become known as Internet 2.0 [13].

Android's releases prior to 2.0 (1.0, 1.5, 16) were used exclusively on mobile phones. Most Android phones and some Android tablets now use a 2.x release and Android 3.0 was a tablet-oriented release but does not officially run on mobile phones. The current Android version is 4.3. Android's releases are nicknamed after sweets or dessert items like Cupcake (1.5), Frozen Yogurt ("Froyo") (2.2), Ginger Bread (2.3), Honeycomb (3.0), Ice Cream Sandwich (4.0), Jelly Bean (4.1) and KitKat (4.4) being the recent one.[14].

Android applications are written in the Java programming language. The Android SDK provides tools for code compilation and packaging data and resource files into an archive file with '.apk' extension called as an *Android package*. Android devices used the '.apk' file to install the application. Android's application framework allows for the creation of extremely feature rich and novel applications by using a set of reusable components.

The amalgamation of the Android development environment with the Bluetooth wireless technology is known by Android's support for the Bluetooth network stack, which permits a device to wirelessly exchange data with another Bluetooth device. The application framework enables access to the Bluetooth functionality using the Android Bluetooth APIs. These APIs allow wireless applications to connect to other Bluetooth devices for point-to-point and multipoint wireless features.

Using the Bluetooth APIs, an Android application can carry out the following functions:

- Scrutinize for other Bluetooth devices
- Enquire about the local Bluetooth adapter for paired Bluetooth devices
- Establish the RFCOMM channels
- Connect to other devices through service discovery
- Exchange data to and from other devices
- Administer multiple connections

Android had unique support for Bluetooth in Android-powered devices including: Classic Bluetooth for more battery-intensive operations such as streaming and communicating and with low power requirements, Android 4.3 (API Level 18) introduces API support for Bluetooth Low Energy [15].

### *B. Bluetooth*

Wireless networks for short range communications have a wide spread usage of Bluetooth radio transmissions between 2400–2480 MHz by Telecom vendor Ericsson since 1994 [2]. Bluetooth technology forms small ad hoc networks termed as Personal Area Networks (PANs) also provides a mechanism to emulate the RS-232 data cables, supervised by the Bluetooth Special Interest Group, since 1998. Modern mobile devices embed small, low-powered and cheap integrated chips functioning as short-range radio transceivers for Bluetooth radio communications. Device pairing, authentication, encryption and authorization techniques have given recognition to Bluetooth technology due to its vital security mechanisms.

Different types of Bluetooth applications can be developed using Android platform architecture using the Bluetooth profiles . The device manufacturers provide the services using the support of these profiles in their devices to maintain compatibility for the Bluetooth technology.

The Bluetooth profile used in Home Automation System (HAS) Android mobile phone application is the Bluetooth Serial Port Profile (btspp) [15]. RFCOMM is a connection-oriented protocol. It provides streaming communication between the devices. The btspp profile and RFCOMM protocol are used in the application to access the serial port and communicate using streaming data. All of the Bluetooth APIs is available in the android.bluetooth package.

### *C. Keil μVision IDE*

Keil development tools for the 8051 Microcontroller Architecture support every level of embedded software development. The industry-standard Keil C Compilers, Macro Assemblers, Debuggers, Real-time Kernels, Single-board Computers, and Emulators support all 8051 derivatives. 'C Language Program code' for AT89c51 microcontroller is developed, compiled and debugged using Keil μVision IDE [10].

### *D. ucFlash+ Programmer*

The ucFlash+ Programmer [11] is an affordable, reliable, and fast programmer for MCS51/AVR Microcontrollers and 24Cxx I2C EEPROMs. The programmer is designed to operate with the Intel Pentium-based IBM-compatible desktop computers and notebook computers. No interface card is necessary to plug the module into a PC (this feature is especially handy for notebook computer users). The menu-driven software interface makes it easy to operate. ucFlash+ Programmer is used here for programming AT89C51 microcontroller for HAS.

## IV. Home Automation Hardware

Home Automation Hardware is work as client part in Home Automation System and it is formulated in two ways like: (A) Automated and (B) Conventional

### A. *Automated*

Automated way is an actual system and known as circuit for Home Automation System which is shown in Figure 4. Home Automation Circuit comprises microcontroller AT89C51, Serial Bluetooth Module, octal peripheral driver array ULN2803, regulator IC 7812, IC7805 and a few discrete components. Here in this circuit, microcontroller AT89C51 is worked as main programmable switching unit which receives data from Bluetooth serial module and transferred appropriate program data to ULN2803 for operating relay ON and OFF. The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, five vector two-level interrupt architecture, a full duplex serial port, and on-chip oscillator and clock circuitry. [12]

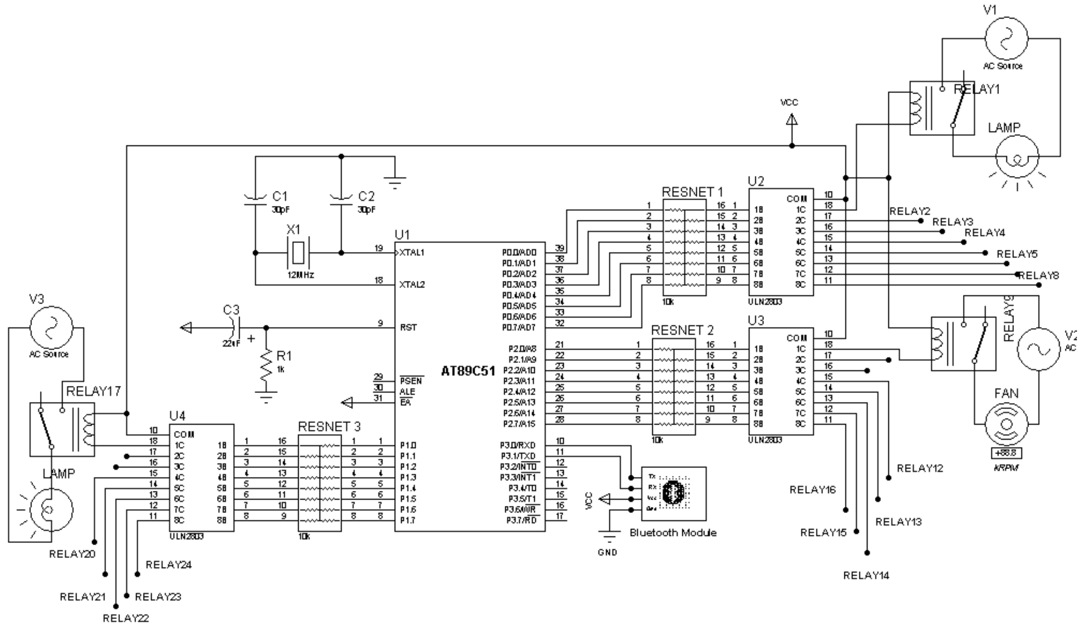


Figure 2. Home Automation Circuit

### B. *Conventional*

Conventional way means traditionally all the devices are controlled using “Switches” through switch board. Any user wants to make device ‘ON/OFF’ without using HAS then he/she can use particular switch attached to device. To incorporate this, relays are connected through ‘Two Way Switch’ as shown in figure 3.

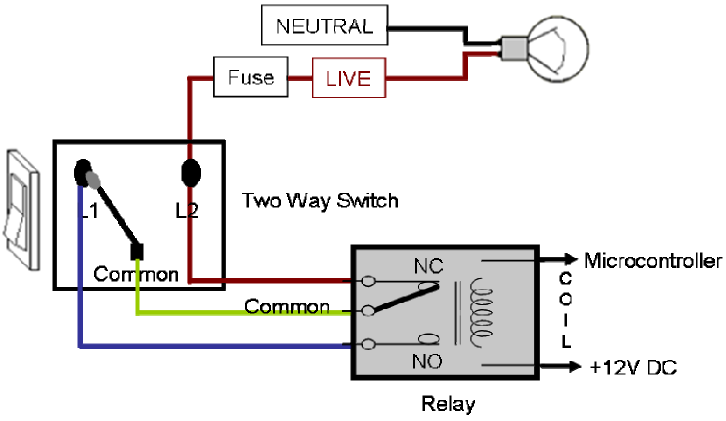


Figure 3. Two Way Switch Connection with Relay

## V. HOME AUTOMATION SYSTEM APPLICATION PROGRAM

Home Automation System Application is developed using Android platform for mobile phones and tablets with Android support having the Bluetooth connectivity option. Android devices have almost covered a larger pie in the market, hence keeping in mind its popularity; this application can prove to be a boon for those users longing to make their house – an automated home. Home appliances can be controlled using this interactive application within their home environment easily. Application is based on expandable list view and it controls the ON/OFF using Bluetooth connectivity. Expandable list view is used to group list data by categories like Living room, Bed room, Kitchen, Guest room, Store and Porch. It has the capability of expanding and collapsing the groups when user touches header as shown in Figure 4 (a).

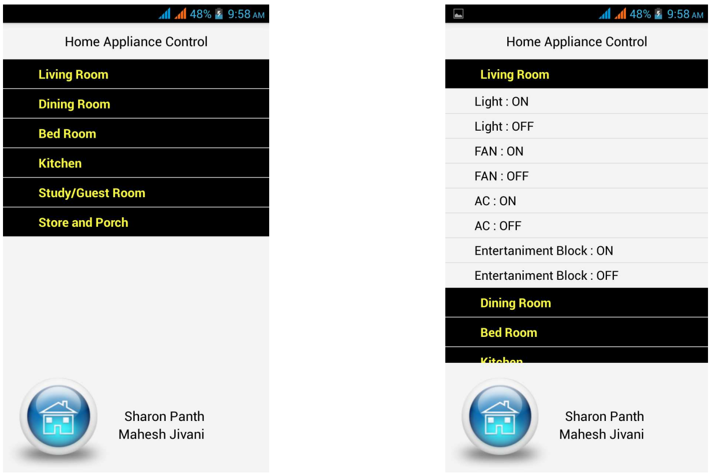


Figure 4. Screenshot of Home Automation System Application: (a) List View (b) Expanded List view

## VI. BLUETOOTH CONNECTIVITY

Home Automation System Application has the capability of expandable list view is used to exchange the ASCII data with HAS circuit through Bluetooth facility of mobile phone when user touches header as shown in Figure 4 (b).The Android platform comprises of the Bluetooth network stack. This allows a device to wirelessly exchange data with other Bluetooth devices. The application framework then provides access to the Bluetooth functionality with the help of the Android Bluetooth APIs. These APIs make the applications to connect wirelessly to other Bluetooth devices, for point-to-point and multipoint wireless features. Flowchart of Application program is shown in Figure 5(a).

### *A. The Bluetooth APIs*

All of the Bluetooth APIs' are available in the Android Bluetooth package. The following is the overview of the classes needed during the application's development.

- BluetoothAdapter: Represents the local Bluetooth adapter (Bluetooth radio)
- BluetoothDevice: Represents a remote Bluetooth device, to query information such as its name, address, class, and bonding state.
- BluetoothSocket: Represents the interface for a Bluetooth socket (similar to a TCP Socket).
- BluetoothClass: Describes the general characteristics and capabilities of a Bluetooth device.

### *B. Bluetooth Permissions*

In order to use Bluetooth features in an Android application, at least one of two Bluetooth permissions: BLUETOOTH and BLUETOOTH_ADMIN are needed to be declared.

We declared the Bluetooth permission(s) in our application's AndroidManifest.xml as below:

```
<manifest ... >
  <uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    ….
</manifest>
```

### *C. Methods for Bluetooth connectivity*

Normally, before commencing communication, devices can use two methods for initiating communication with each other which can be done normally either by discovering other nearby devices to detect the address and services that are provided by other devices or by knowing the device address beforehand and directly using that address for further communication process. In Home Appliance Control, the later method is used.

#### *1) The Discovery method:*

The devices participating in the communication process must be set to the discovery mode.

#### *2) The Known Address method:*

The communication with a known remote device is helpful in faster communication as the discovery time is avoided. In this automation system, the appliances would be already known to the Bluetooth module as and when required. It is established in the following manner:

##### *a) SPP*

SPP (Serial Port Profile) in the Bluetooth profiles is implemented as the Bluetooth Serial Port Profile (btspp). Bluetooth profiles are the implementation of the Bluetooth protocols in full or partial manner as defined and adopted by the Bluetooth SIG. They reside over the Bluetooth protocol stack for their full or partial support. The implementation hence uses the support of Bluetooth Serial Port Profile (btspp) and RFCOMM protocol which is a connection-oriented protocol for Radio Frequency Communication, the replacement for the RS-232 cable to provide serial emulation.

##### *b) MAC Address*

Bluetooth devices have a 12 hexadecimal digit MAC address which is to be known beforehand.

A complete specification for the connectivity in Home Appliance Control is done using the **Known Method** as follows:

The entire setup described here includes the completion of these important steps using all classes and interfaces of the Android Bluetooth APIs available in the android.bluetooth package.

**Step one:** Bluetooth verification and enabling process

- Check for Bluetooth support
- This can be accomplished by using the BluetoothAdapter in the application which serves as an entry point to all Bluetooth interactions. There is only one adapter for entire system and it represents the devices' Bluetooth radio (adapter). If it is null the device does not have Bluetooth support.
- Enable Bluetooth
- Check to make sure it is turned on in the application itself. Otherwise, request the user to turn on Bluetooth without leaving the application.

Then there are two cases: first, if Bluetooth is not turned ON,

**Step two:** Set up a pointer to the remote node using its MAC address.

- The BluetoothAdapter from step one, is able to instantiate a BluetoothDevice using its pre-known MAC address.

  Two things are needed to make a connection:
  - A MAC address.

    We get it from the Bluetooth module's MAC address. For example, a 12 digit hexadecimal MAC address can be represented as 00:12:08:17:21:55.
  - A Service ID or UUID.

    In this case we are using the UUID for SPP. Services can be identified by a UUID. A Universally Unique Identifier (UUID) identifies each service and service attribute in bluetooth uniquely. Each such identifier is guaranteed to be unique across all time and space. The `UUID` class in util package of java can be represented by short (16- or 32-bit) and long (128-bit) UUIDs. Constructors create a UUID from a `String` or from a 16- or 32-bit value, a method to compare two UUIDs (if both are 128-bit), and a method to covert a `UUID` into a `String`. The `UUID` instances are immutable, and only services identified by UUIDs are discoverable.

**Step three:** Establish the connection.

- After obtaining the BluetoothDevice object that represents the remote device, it is used to get the BluetoothSocket and initiate the connection by creating the insecure '*rfcomm socket*' to service record by passing the SPP UUID to it that is hard coded before.

**Step four:** Create a data stream.

- The data stream helps to send message to the remote device, here, this helps to talk to the appliances finally in the Home Appliance Control application.

The Android emulator does not emulate Bluetooth so real devices were used for basic program testing and final implementation.

## VII. MICROCONTROLLER AT89C51 PROGRAM

The program (HAS.c) for the microcontroller is written in C and compiled using Keil µVision IDE to generate hex code. The generated hex code is burnt into the microcontroller using a µC Flash+ programmer. The program coding starts with '#include <reg51.h>' header files. The microcontroller port pins are defined using '*sbit*' function for interfacing with the surrounding peripherals.

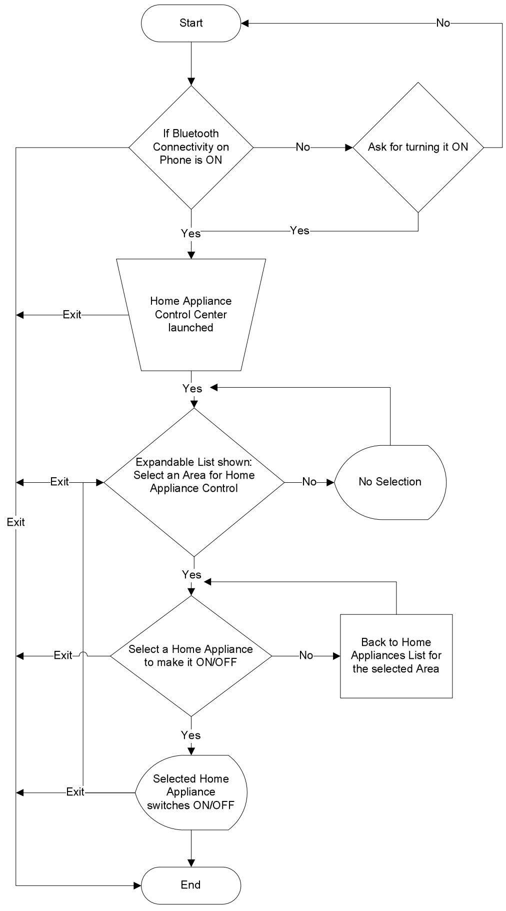
By selecting device from list menu of Home Automation Application program which is developed using Android installed on mobile device, will transmit ASCII characters serially through Bluetooth and on the receiving side microcontroller receives data through serial Bluetooth module on serial port of microcontroller AT89C51. 'SBUF register' will read this ASCII characters as equivalent HEX value which will be ported to Port 0, 1, and 2's pins of AT89C51. Flowchart of microcontroller program is shown in Figure 5(b).

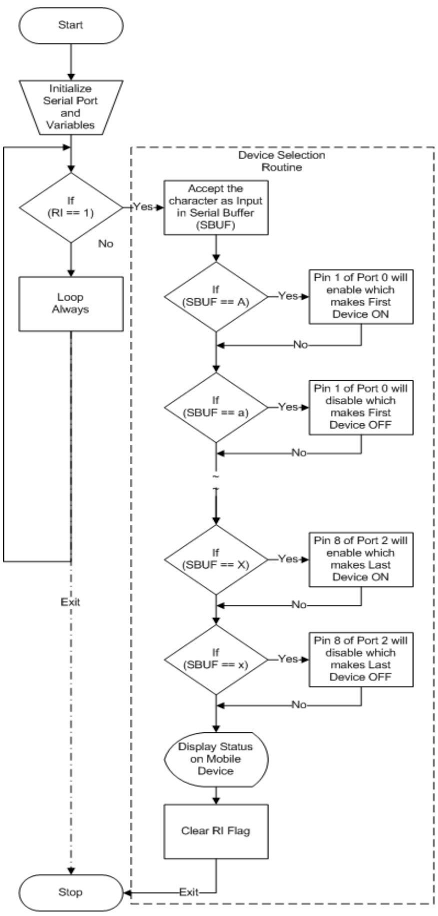
For operating Relay1 in ON condition, one has to select item ‘Light 1 ON’ of list menu ‘Living Room’ means ASCII character “A” will be transmit and microcontroller will receive equivalent value in HEX through SBUF register which is “0x41” through Bluetooth serial module which makes ‘pin 1 of Port 0’ high and for making Relay1 in OFF condition simply select ‘Light 1 OFF’ item of ‘Living Room’ list menu means ASCII character ‘a’ will be transmit, microcontroller will receive equivalent value in HEX through SBUF register which is “0x61” which makes ‘pin 1 of Port 0’ low. Same way you can ON and OFF relay connected to Port 0, 1 and 2. Home Automation System Android Application program on mobile transmits following combination given in Table 1.

TABLE I. HOME AUTOMATION SYSTEM APPLICATION PROGRAM TRANSMISSION COMBINATION

| Character | AT89C51 Port | Relay Number | Status |
|---|---|---|---|
| Capital letters A to H | Port 0 | 1 to 8 (Individual) | ON |
| Small letters a to h | Port 0 | 1 to 8 (Individual) | OFF |
| Capital letters I to P | Port 1 | 9 to 16 (Individual) | ON |
| Small letters i to p | Port 1 | 9 to 16 (Individual) | OFF |
| Capital letters Q to X | Port 2 | 17 to 24 (Individual) | ON |
| Small letters q to x | Port 2 | 17 to 24 (Individual) | OFF |

## VIII. PROGRAM FLOW CHART


Start
No
If Bluetooth Connectivity on Phone is ON
No
Ask for turning it ON
Yes
Yes
Exit
Home Appliance Control Center launched
Exit
Yes
Expandable List shown: Select an Area for Home Appliance Control
No
No Selection
Exit
Yes
Exit
Select a Home Appliance to make it ON/OFF
No
Back to Home Appliances List for the selected Area
Yes
Exit
Selected Home Appliance switches ON/OFF
End


Start
Initialize Serial Port and Variables
Device Selection Routine
If (RI == 1)
Yes
No
Accept the character as Input in Serial Buffer (SBUF)
Loop Always
If (SBUF == A)
Yes
Pin 1 of Port 0 will enable which makes First Device ON
No
If (SBUF == a)
Yes
Pin 1 of Port 0 will disable which makes First Device OFF
No
Exit
If (SBUF == X)
Yes
Pin 8 of Port 2 will enable which makes Last Device ON
No
If (SBUF == x)
Yes
Pin 8 of Port 2 will disable which makes Last Device OFF
No
Display Status on Mobile Device
Clear RI Flag
Exit
Stop

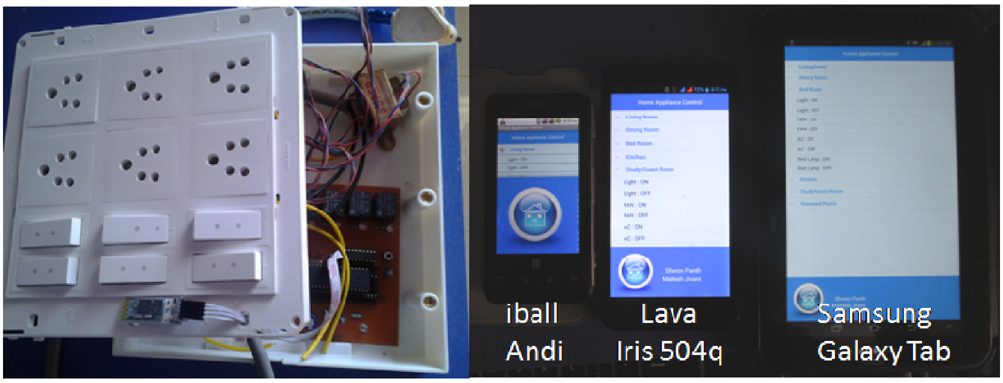
(a) (b)

Figure 5. Flow Chart : (a) Android Mobile Application Program (b) Microcontroller Program

## IV.IMPLEMENTATION

Power up the Circuit and scan the Bluetooth devices on your Mobile Device. If everything's done correctly you will be able to find a Bluetooth device named 'Tiny OS' or name of Bluetooth Module. You will be asked for a pairing code in case of the above model (Tiny OS) it is **1234** but it might be different if you are using a Bluetooth module from another vendor. Figure 6 shows the real implementation of HAS. As shown in figure 6, follow the steps for operating devices listed in program.

- Start Bluetooth connection of your mobile phone.
- Open the 'HAS' application on your Android mobile phone.
- Select connect device from option menu using scan for new devices.
- 'Bluetooth Serial Module will be available under 'select a device to connect' list.
- Pair 'Bluetooth Serial Module' by providing pair code '1234'.
- Start selecting particular device for making 'ON/OFF' from List Menu as shown in figure 6, you will see relay ON and OFF according to combination shown in Table 1.



(a) (b)

Figure 6. Real implementation (a) HAS Prototype Board (b) HAS Android Application Screen Shot

## IX. CONCLUSION

Design and implementation of a Home Automation System using Android for mobile phone has been discussed. The purpose of the system is to use mobile phone's inbuilt Bluetooth facility for automation without using Air Time. Different hardware and software unit of the system are described. The complete application software has been designed using Android, Bluetooth API and C Language. The HAS application program is tested on various Android mobile phones and the results are presented in table 2, which are quite satisfactory and response received from the community in general is encouraging. The HAS furnishes a good paradigm for any Automation System based on Android Mobile Phone and Bluetooth.

TABLE II. LIST OF ANDROID MOBILES ON WHICH HOME AUTOMATION SYSTEM APPLICATION TESTED

| N0. | Mobile Phone | Android OS Version and Name | Screen Size in inch |
|---|---|---|---|
| 1 | iBall Andi | 2.2, "Froyo" | 3.5 |
| 2 | Samsung Galaxy Duos | 2.3, "Gingerbread" | 4.3 |

| N0. | Mobile Phone | Android OS Version and Name | Screen Size in inch |
|---|---|---|---|
| 3 | Micromax Bolt A35 | 2.3, "Gingerbread” | 4 |
| 4 | LAVA iris 504q | 4.2, "Jelly Bean” | 5 |
| 5 | Tablet Samsung GTP3100 | 4.1, "Ice-Cream Sandwitch” | 7 |

## V. REFERENCE

[1] M. Chan, D. Esteve, C. Escriba, E. Campo, “A review of smart homes—Present state and future challenges”, Computer methods and programs in biomedicine, Elsevier, 9I, pp. 55-81, 2008.

[2] J. Haartsen, "BLUETOOTH—The universal radio interface for ad hoc, wireless connectivity", Ericsson Review No. 3, pp. 110-117, 1998.

[3] Why Android ?, http://www.android.com/about/, last seen on October 2013.

[4] R. Llamas, R Reith, M. Shiere, “Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains, According to IDC” 7th August 2013, IDC Press Release, http://www.idc.com/getdoc.jsp?containerId=prUS24257413.

[5] N. Sriskanthan, F. Tan, A. Karande, "Bluetooth based home automation system", Microprocessors and Microsystems, Elsevier, 26, pp. 281-289, 2002.

[6] S. Panth, M. Jivani, “Designing Home Automation system (HAS) using Java ME for Mobile Phone”, International Journal of Electronics and Computer Science Engineering, Vol. 2 No. 02, pp. 798-807, July 2013

[7] S. Panth, M. Jivani, “Device Control in an Ad-hoc Network Environment by using MoSync for Multiple Platform Mobile Application Development”, International Journal of Computer Science & Engineering Technology, Vol. 4 No. 08, pp. 1145-1152, August 2013

[8] Darlington Transistor Array, Texas Instruments, http://www.ti.com/lit/ds/symlink/uln2803a.pdf, last seen on April, 2013.

[9] Serial Bluetooth Module, Tiny OS Electronics, http://www.tinyosshop.com/index.php?route=product/product&product_id=330, 2013.

[10] Keil µvision IDE, http://www.keil.com/uvision/, last seen April 2013.

[11] uCFlash+ Programmer, UC Micro Systems, http://www.ucmicrosys.com/products/ic-programmers/**ucflash**.html, last seen on April 2013.

[12] AT89c51 8 bit Microcontroller, ATMEL Corporations, http://www.**atmel**.com/images/doc0265.pdf, last seen on October 2013.

[13] A.Göransson, D.C.Ruiz, "Android Open Accessory Programming with Arduino", John Wiley & Sons, Inc., ISBN: 978-1-118-45476-3, 2013

[14] Operating Systems, Planet Gadget., http://www.planetgadgetph.com/operating-system.html, last seen october 2013.

[15] Bluetooth API Guides, https://developer.android.com/guide/topics/connectivity/bluetooth.html, last seen on october 2013.