Home Automation System (HAS) using Android for Mobile Phone

Shikhar Sharma¹, Shripad S. Desai²

¹Student, department of electrical Engineering, Bharati vidyapeeth Deemed University, College of Engineering, Pune, India ²Assistant professor, department of Electrical engineering Bharati vidyapeeth Deemed University, College of Engineering, Pune, India

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.

Index Terms - Home Automation System, Android, Microcontroller, Bluetooth, Mobile Phone, Appliance

1.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 equipment. 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. 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.



Figure 1. System Architecture

The system is based on serial data transmission using Bluetooth wireless communication in order to facilitate on wireless communication. It also supports conventional Android enabled mobile phone offers system connection and control utilities. ULN 2803 relay driver Bluetooth Module from TINY OS programmer for burning HEX file into microcontroller were used for the development. An ATMEL 89C51, 8-bit microcontroller was used as an embedded relay controller.

3. 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 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.





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



4. HOME AUTOMATION SYSTEM APPLICATION PROGRAM

Home Automation System Application is developed using android platform by mobile phones or tablets with Android support having the Bluetooth connectivity 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 connectivity. Expandable list view is used to group list data by categories like Living room, Be room, Store and Porch. It has the capability of expanding and collapsing the groups when user touches the header as shown in Figure 4 (a).



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

5. BLUETOOTH CONNECTIVITY

Home Automation System Application has the capability of expandable data with HAS circuit through Bluetooth facility of mobile phone when user touches header as shown in Figure 4(b). The Android platform comprises with other Bluetooth devices. The application framework helps 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.

- Bluetooth Adapter: Represents the local Bluetooth adapter (Bluetooth radio)
- Bluetooth Device: Represents a remote Bluetooth device, to query information such as its name, address, class, and bonding state.
- Bluetooth Socket: Represents the interface for a Bluetooth socket (similar to a TCP Socket).
- Bluetooth Class: 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 Bluetooth Adapter 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 Bluetooth Adapter from step one, is able to instantiate a Bluetooth Device using its pre-known MAC address.

Two things are needed to make a connection: 1 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.

2 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 Bluetooth Device object that represents the remote device, it is used to get the Bluetooth Socket 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.

6.IMPLEMENTATION

Power up the Circuit and scan the Bluetooth devices on your mobile device. If everything is 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 above model (Tiny OS) it is 1234 but it might be different if you are using Bluetooth from another vendor. Figure 6 shows the implementation of HAS.

Follow the following steps:

- Start Bluetooth connection of your mobile phone.
- Open the 'HAS' application on your
- Select connect device from option m
- '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
- ON and OFF according to combination shown in Table 1.



7. 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 Airtime. Different hardware and software unit of 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.

Tested								
NO	Mobile Phone	Android	Screen					
		Version	Size in					
			Inch					
1	iBall Andi	2.2	3.5					
2	Samsung Galaxy Duos	2.3	4.3					
3	Micromox Bolt A35	2.3	4					
4	LAVA iris 504q	4.2	5					
5	Tablet Samsung GTP3100	4.1	7					

List of	android	devices	on	which	HAS	Application
Tested						

8.ACKNOWLEDGEMENT

We would like to express our special thanks of gratefulness to Dr. D.S banker (Head of department of Electrical Engineering) for their able guidance and support for completing my research paper. I would Also like to thank the faculty members of the Department of Electrical Engineering who helped us with Extended support.

REFERENCE

- 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", Microprocessor and Microsystem, 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] Darlington Transistor Array, Texas Instruments, http://www.ti.com/lit/ds/symlink/uln2803a.pdf, last seen in April 2013.
- [8] Serial Bluetooth Module, Tiny OS Electronics, http://www.tinyosshop.com/index.php?route=pro duct/product&product_id=330, 2013.
- [9] Keil Dision IDE, http://www.keil.com/uvision/, last seen April 2013.
- [10] uCFlash+ Programmer, UC Micro Systems, http://www.ucmicrosys.com/products/icprogrammers/ucflash.html, last seen in April 2013.
- [11] A.Göransson, D.C.Ruiz, "Android Open Accessory Programming with Arduino", John Wiley & Sons, Inc., ISBN: 978-1-118-45476-3, 2013
- [12] Operating Systems, Planet Gadget., http://www.planetgadgetph.com/operatingsystem.html, last seen October 2013.
- [13] Bluetooth API Guides, https://developer.Android .com/guide/topics/connectivity/bluetooth.html, last seen in October 2013.