

Android Based Bluetooth Appliance Control Mechanism

¹Aritra Sasmal, ²Abhradeep Palit, ³Abhishek Majumder, ⁴Deepjyoti Bhowmik,

⁵Sougata Chanda, ⁶Aritra Bandyopadhyay

^{1,2,3,4,5,6}Supreme Knowledge Foundation Group of Institutions, Mankundu, West Bengal, India

Abstract

The proposed algorithm basically is an android application which possesses the capability to control any sort of electrical appliances remotely over the Bluetooth. The control mechanism mainly refers to the switching (on / off) of the appliances. It emphasizes on creating a virtual switch board, giving the user an exact experience of what he does regularly when switching off his bedroom lamp or his air – conditioner and for a lot of other household appliances.

Keywords

Android, Virtual Switch Board, Bluetooth

I. Introduction

Home automation has been a feature of science fiction writing for many years, but has only become practical since the early 20th century following the widespread introduction of electricity into the home, and the rapid advancement of information technology. Early remote control devices began to emerge in the late 1800s. For example, Nikola Tesla [4] patented an idea for the remote control of vessels and vehicles in 1898. The emergence of electrical home appliances began between 1915 and 1920; the decline in domestic servants meant that households needed cheap, mechanical replacements. Domestic electricity supply, however, was still in its infancy — meaning this luxury was afforded only the more affluent households. In our project we have dealt with a critical section of Home Automation, i.e. the controlling of household appliances remotely through Bluetooth.

II. Literature Review

There are a number of home appliance control systems are proposed. Each one has featured some new parameters and usability.

N. Sriskanthan [5] has depicted the use of Bluetooth in home automation system. It has been shown that various devices can be controlled without the change of the core of the system but it cannot be implemented in the mobile technology.

E. Yavuz [6] has introduced a telephone and PIC remote controlled device for controlling the devices pin check algorithm. It was supported for cable not for wireless connection.

Pradeep G [8] has depicted a home automation system by using Bluetooth. In the proposed method the concept of saving a pre-loaded list has been introduced. It sufficiently reduced the time and the cost.

Piyare [7] has depicted a low cost and scalable wireless device which is can be controlled by minimum change of its core.

The proposed appliance control mechanism that has been designed is mainly based upon the Bluetooth service of the Android device in which the application would be installed. The Android platform [1] includes support for a Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices, just as we need to send particular data values to the hardware module (from the Android device), in order to control the appliances remotely. The application framework provides access to the Bluetooth functionality through the Android Bluetooth

APIs [2]. These APIs let applications wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features. Using the Bluetooth APIs [3], 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 other devices
- Manage multiple connections

A. Flow Diagram

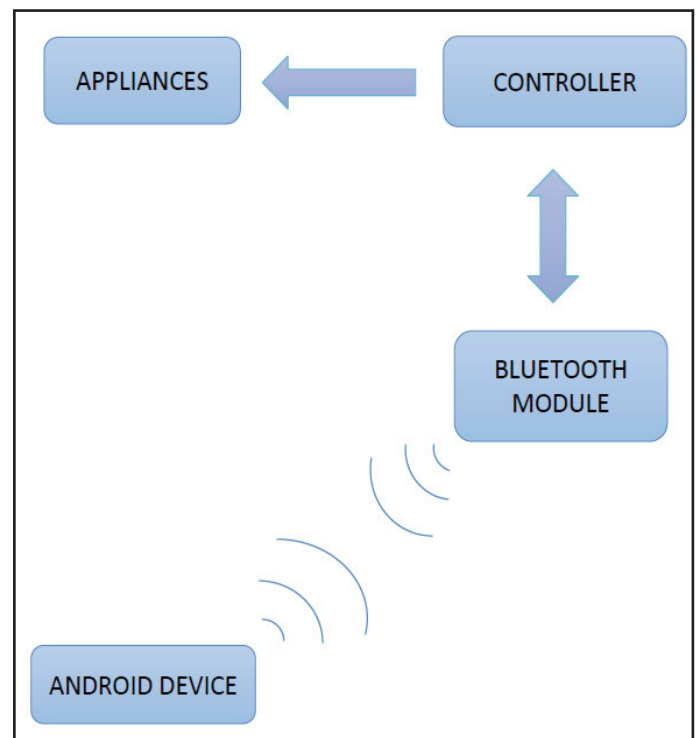


Fig. 1: Android Based Appliance Control Mechanism

III. Proposed Algorithm

A. For the Home Screen UI

The Home Screen UI would comprise of interfaces such as a toggle button to Enable/Disable the Bluetooth service, a button to scan for nearby devices, another button to cancel an ongoing scan, a list to display the devices found and a status display.

Step 1: Acquire the default Bluetooth adapter of the android device.

Step 2: If adapter cannot be acquired

Step 2.1: Notify user: “Your Device does not Support Bluetooth!!”

Step 2.2: Disable all the UIs of Home Screen.

Step 3: Else

Step 3.1: Set the status display, state of toggle button other buttons as per the status of the Bluetooth service on the android device.

Step 4: If the user chooses to Enable/Disable the Bluetooth service

Step 4.1: Notify the user accordingly using Toast notifications.
Step 4.2: Put the corresponding status message in the status display.

Step 5: If the user chooses to scan for new devices

Step 5.1: Clear the found devices list.
Step 5.2: Disable the scan for devices button & enable the cancel button.
Step 5.3: Initiate the Bluetooth device discovery service.
Step 5.4: For each device found
- Send message: "REQ_ID" to the device
- If reply is "0xBBBB" Store the device in an ArrayList "bluetoothDevices" of type BluetoothDevice. Display the device's name & address in found devices list.

Step 6: If the user chooses to cancel the ongoing scan

Step 6.1: Stop the Bluetooth device discovery service.
Step 6.2: Enable the scan for devices button & disable the cancel button.

Step 7: If the user taps on a device entry in the found devices list

Step 7.1: Get the Bluetooth device from "bluetoothDevices" list from that tapped position.
Step 7.2: If the device is not already paired
- Pair the device.
Step 7.3: If the device is pairing
- Wait for the pairing to complete.
Step 7.4: If the device is successfully paired
- Create Intent.
- Put the device into the intent
- Start the Intent to provide the user with a Switch Board User Interface.

B. For the Switch Board UI

The Switch Board UI would comprise of interfaces such as a toggle buttons to switch on/off the appliances, an input field to enter the rating (in Watts) of the appliance corresponding to that switch, a text field to display the units (in kWh) consumed by the device for the duration for which it was switched on & a text field to display the name of the device. Here the toggle buttons, rating input field & units consumed text field are organized in form of rows i.e. each row comprises of a toggle button, a rating input field & a text field for units consumed.

Step 1: Disable all rows.

Step 2: Extract the corresponding device from the Intent.

Step 3: Create an RfComm Socket corresponding to the device & connect to the device.

Step 4: Send Message: "DEVS_ATCHD" to the device.

Step 5: Retrieve the count value N from the device's reply.

Step 6: Enable the first N rows.

Step 7: If the user checks the toggle button

Step 7.1: Send Message: "x1" to the device.
(Where x is the row in which the toggle button is located and 1 infers to switching "on" the corresponding appliance)

Step 7.2: Get present system time in milliseconds.

Step 8: If the user unchecks the toggle button

Step 8.1: Send Message: "x0" to the device.

(Where 0 infers to switching "off" the corresponding appliance)

Step 8.2: Get the present system time in millisecond and subtract the previous time from it and divide it by 3600000 to get the time difference in hours

Step 8.3: If rating input field is not empty

- Multiply value in the rating input field by the (time difference/1000) to obtain the units consumed and update the unit's text field accordingly.

Step 8.4: Else

- Notify user: "Enter rating to obtain units consumed!"

Step 9: Close the socket upon closing of the switch board UI.

IV. Result & Discussion

Firstly, the user is presented with an option to turn on the mobile's (or their android device's) Bluetooth radio. After that the user is presented with an option to search the nearby devices (appliance controllers). The moment the user selects an appliance controller from the generated list, he / she is presented with a new interface comprising of all the switching mechanisms associated with that controller with an additional feature of assessing the units consumed by any appliance connected to that controller for a given time period if it's rating has been provided. The wireless control is implemented by sending a discrete string of data to the appliance controller over the Bluetooth whenever the user checks / unchecks any toggle button associated with any appliance in the application. The appliance controller uses a relay to switch an appliance on / off depending on the received string of data. The strings generated are different for each of the buttons associated with some appliance or the other. Depending on the value a certain pin of the microcontroller is made high / low.

For example:

When Toggle Button 1 is checked, the string "11" gets sent to the appliance controller.

On receiving that string, let us suppose the Appliance controller sets the pin 12 to a HIGH state, which is connected to a relay.

Thus the relay is switched on and it in turn connects the load to the source and hence switching on the appliance.

Similarly when Toggle Button 1 is unchecked, the string "10" gets sent to the appliance controller.

The controller upon receiving this sets its pin 12 to LOW state which in turn switches off the corresponding device.

The circuit diagram of the hardware controlling module is presented below:

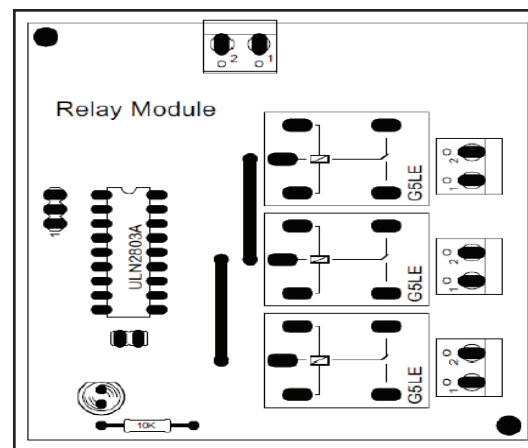


Fig. 2: Relay Module

V. Conclusion

It's an android based Bluetooth controlled application, hence it supports mobile technology.

It has been primarily focused upon the Classic Bluetooth feature. Classic Bluetooth effectively serves the purpose for more battery-intensive operations such as streaming and communicating between Android devices. GSM module can be used in place of the Bluetooth module, it can be accessed over the internet. That will be especially for industrial purposes. So comparing the said home appliance control devices it can be said that the proposed system is cost effective. It has already been portrayed, the power consumption and the power unit calculator in small prototype model. A total power management system in which a monthly power assumption manager (for limiting the range of electricity) might be introduced are in process for further development. As days are passing by we are edging more and more towards the next generation smarter world, where technologies will be more agile and astute. The expectations of the people are also going to break the bars. This is the sole reason behind the planning of this appliance control system.

VI. Acknowledgement

This work has been carried out at 'Center for Machine Learning and Intelligence' under Computer Science dept. of Supreme Knowledge Foundation Group of Institutions (SKFGI), Mankundu, Hooghly, WB, India.

References

- [1] [Online] Available: [http://www.developer.android.com / index](http://www.developer.android.com/index)
- [2] [Online] Available: [http://www.anddev.org/code-snippets-for-android-f33 / serial-over-Bluetooth-simple-test-client](http://www.anddev.org/code-snippets-for-android-f33/serial-over-Bluetooth-simple-test-client)
- [3] [Online] Available: [http://www.bravenewgeek.com / tag / Bluetooth](http://www.bravenewgeek.com/tag/Bluetooth)
- [4] [Online] Available: [http://www.google.com/patents/ US613809](http://www.google.com/patents/US613809)
- [5] N. Sriskanthan, Tan Karand, "Bluetooth Based Home Automation System", Journal of Microprocessors and Microsystems, Vol. 26, pp. 281-289, 2002.
- [6] E. Yavuz, B. Hasan, I. Serkan, 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.
- [7] Bluetooth based home automation system using cell phone R Piyare, M Tazil - ... (ISCE), 2011 IEEE 15th International ..., 2011
- [8] Pradeep.G, B.Santhi Chandra, M.Venkateswarao, "Ad-Hoc Low Powered 802.15.1 Protocol Based Automation System for Residence using Mobile Devices", Dept.of ECE, K L University, Vijayawada, Andhra Pradesh, India. IJCST Vol. 2, SP-1, December 2011.
- [9] AmulJadhav, S.Anand, NileshDhangare, K.S.Wagh "Universal Mobile Application Development (UMAD) On Home Automation" Marathwada Mitra Mandal's Institute of Technology, University of Pune, India Network and Complex Systems Vol 2, No. 2, 2012.