EMBEDDED BASED HOME AUTOMATION

Pradeep Rajan S*, Sivabalan S*, Dr. K. Fathima Patham***
& Sundar Ganesh C S**

* UG Student, Department of Robotics, PSG College of Technology, Coimbatore, Tamilnadu

** Assistant Professor, Department of Robotics and Automation Engineering, PSG College of Technology, Coimbatore, Tamilnadu

*** Professor, Department of Robotics, PSG College of Technology, Coimbatore, Tamilnadu

Abstract:

In this paper, a control mechanism is developed to optimize the power consumption in household devices. The need for energy efficient devices and declining resources for energy production has put in a state to look for new opportunities to efficiently manage and deliver optimize power. The control mechanism which has been developed is largely used at industry level helping them to optimize power. These control mechanisms are not at basic consumer level where the user is very large. This paper shows the method to make every home appliances such as like lights, air conditioners, fans, washing machines to work through the android application in our smart phone. This method certainly can serve a huge potential application to the handicapped and the bedridden patients who can control appliances staying where they are and it can reduces the overall tediousness of them as well.

Key Words: Home Automation, Embedded Systems, Artificial Intelligence, Bluetooth Module & Android

1. Introduction:

Home automation is the automation of the home or household activity. House automation includesthe centralized control of lighting, HVAC (heating, ventilation and air conditioning), appliances, and other devices to provide improved convenience, comfort, energy efficiency and security. The idea of concept of home automation has been around for a long time and the automation products have been available on the market for decades, though no one solution has broken through to the mainstream yet. The elderly and disabled can provide increased quality of life for the persons who might otherwise require caregivers or institutional care. It can also provide a remote interface to home appliances or the automation itself, via telephone line, wireless transmission or the internet, to provide control and monitoring. This paper presents the design and implementation of a low cost Android smart phone based home automation system. This design is based on the open sourced Arduino prototyping board where the sensors and electrical appliances are connected. To enhance the system responsiveness and to make it more dynamic, we have integrated a popular and open source RTOS, the scmRTOS. With the application of Android as a Smart Phone Operating System by Google Inc, Smartphones are becoming more popular around the world. Currently, Android has a rapid growth of 75% of Smart Phones user base. This mass adoption of Smart Phones created a demand for hard and soft applications.

Currently, Smart Phones are the important human interaction devices and the users want to control/accomplish most of their tasks from their Smart Phones rather than conventional ways. There are many wireless protocols that come embedded on a Smartphone has introduced a wireless lifestyle reliving people from the “wired” cable chaos [4]. Bluetooth feature can be found in almost all Android based devices which have been very popular over years for wireless data transmission with ease. Here, we propose a home automation system based on Bluetooth technology [8]. Some of the factors that influence the design of a home automation system include the scalability of the system, the ease of integrating new devices into the system and security. A cost effective system would qualify it for mass adoption.

The home automation systems are broadly classified into two categorize: locally controlled systems and remotely controlled systems. The systems that use an in-home controller to achieve home automation are called locally controlled systems.The globally- controlled systems are the systems that use an internet connection. Such type of systems can be controlled through personal computer, mobile devices, etc. The drawback of home automation system using Bluetooth, GPRS or RFID which requires a standalone hardware and software environment to be installed in each home. Also such systems provide the user with limited access as the access area is restricted only within a specific range. Cloud Network [6] and mobile devices reduces the need of installing and run applications on the customer's own computers and simplify maintenance and support. This project combines the remotely and locally and controlled systems with the use of a Cloud network. Cloud Computing [6] provides access on demand to resources online which requires less management effort and can be easily provisioned. An automated device hugely reduce theheavy amount of human energy, moreover humans are more prone to errors.During intensive conditions an automated device can work with diligence, versatility and with almost zero error when compared to human.

2. Android Design:

Android has a software library for mobile devices that has inbuilt in operating system, middleware and key applications. The Android SDK provides the necessary tools and Application Interfaces necessary to begin developing applications on the Android platform using the Java programming language. They include System C library, Media library, Surface Manager, LibWebCore, SGL, SQLite, Free Type and 3D libraries. The Android SDK compiles the necessary code along with any data and resource files into an Android package.

The Android has the capability for the Bluetooth stack, which allows a device to transfer and receive data with many Bluetooth devices without wire. TheBluetooth application framework has the full access to the Bluetooth functionality by the wireless Android Bluetooth Application interfaces. The Bluetooth APIs can perform the following Scan for other Bluetooth devices , establish Radio Frequency COMM channels, connect to other devices through directory ,transfer data to and from devices and Manage multiple connections. The program that scans and establishes a connection and sends data to the Bluetooth enabled Arduino board. The wireless medium get hold of the local Bluetooth adapter present in the smartphone and establish a connection with our external wireless module that is connected to the Arduino board. After the first connection is fully made, we transmit command from our mobile to the Arduino that are then utilized by the microcontroller to choose the specified appliances to switch ON/OFF. Android 4.1 and higher has support for voice output over a universal serial bus connection or Bluetooth An Android USB is necessary to Android Accessory Protocol, which defines the procedure for detecting the accessory and sets up communication with an Android powered device.

3. Embedded Module:

This project is developed using AVR series of microcontroller (ATMEGA 32) because of its advanced features such as inbuilt Analog to Digital Converter, EEPROM etc. The Atmel AVR ATMEGA 32 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. The interface between the Bluetooth modulescan be made directly with microcontroller pins since the power level of module is at 5V .We have to configure the microcontroller to work at 9600 baud rate. The TXD pin of Microcontrolleris connected to RX-IN pin of Bluetooth module. The RXD pin of MCU will go to TX- OUT pin of Bluetooth module Ground and +5V supply between Bluetooth module and microcontroller should be connected [6]. The objective of using Microcontroller is that maximum 30 appliances can be controlled by small modifications in the C program the GSM can be interfaced for remote control of home devices by using the Short Message Service with minimum modifications. Here we use Arduino which is a single-board microcontroller board based on Atmel's 8-bit microcontrollers. Fig. 1 shows the Arduino Uno board.

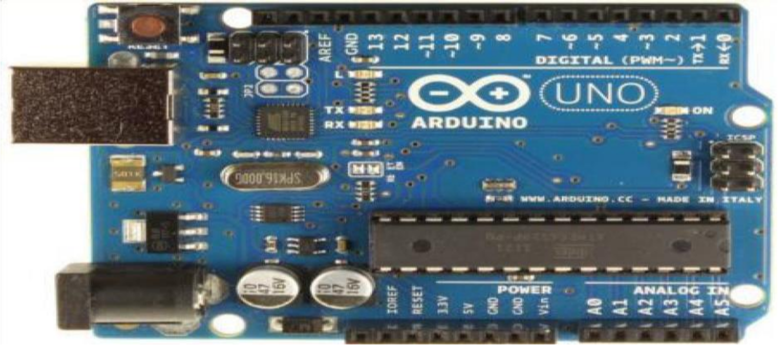


Figure 1: Arduino Uno Board

4. Bluetooth Module:

The module used for wireless transfer is the Bluetooth module allows us to wirelessly transmit and receive data. The module used in this project based on the Bluetooth V2.0 protocol and is having a range of 10 meters operating at a frequency of 2.4Giga Hz with a maximum data exchange rate of 2.1Mbps. We have used HC-06Bluetoothmodule. This particular Bluetooth module permits the microcontroller with a standard RS232 serial port to communicate with a PC or a smartphone equipped with a Bluetooth Master module. Bluetooth module is as robust as possible the operating band is divided into 1MHz-spaced data channels, each signaling data at 1Mega byte/second. This is done by using GFSK (Gaussian Frequency Shift Keying) modulation scheme.Both devices re-tune their radios to a different frequency, effectively hopping from radio channel to radio channel after each packet; this is known as FHSS (Frequency Hopping Spread Spectrum). Wherever a transmissionline is compromised by interference on any one channel, the retransmission will always be done on a different (hopefully clear) channel. For Each Bluetooth time slot lasts 625µs (625 microseconds) and generally devices hop once per packet, giving a hop rate of 1600 hops/second. If Bluetooth devices are to hop to new frequencies after each packet, there has to be some sort of agreement between the devices. The concept of Master and Slave appears; the Master is the Bluetooth device that decides the frequency hopping sequence. The Slave then synchronizes to the Masters in time and frequency by following the Master's frequency hopping sequence. Each Bluetooth device has a unique Bluetooth device address and a 28-bit Bluetooth clock. The baseband frequency part of the Bluetooth System uses a special algorithm, which calculates the frequency hop sequence from the master's clock and device address.

5. Circuit Diagram:

The wiring of the hardware system is shown below in the fig2. The circuit consists of an Arduino Board, Bluetooth module and a set of relays. The Arduino board is powered by connecting the Vcc and Gnd to 5V and Gnd of the power supply respectively. The Bluetooth module should also be powered and it's Vcc and Gnd and shorted with the power circuit of the Arduino Board. The NO of the relay is connected to the output ports of the arduino board and NC is connected to the 220 V AC Supply. When the output Ports of the Arduino Board turns ON then the relay is switched ON and the device which is connected to the relay is also switched ON.

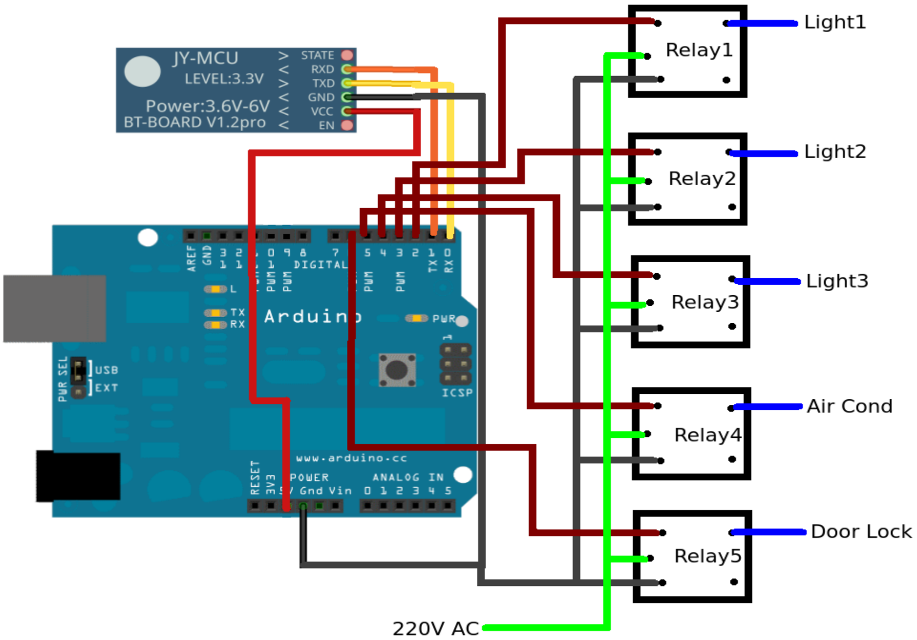


Figure 2: Circuit Diagram for Home Automation

6. Results and Discussion:

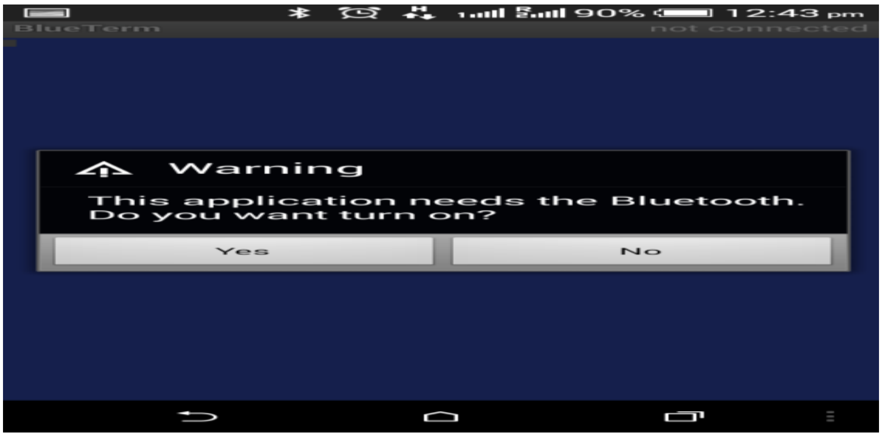
In this project, there are several steps in implementing the project like the design of Arduino Uno circuit with Bluetooth circuit, relay circuit and connecting the power supply circuit,.The Pairing of Smartphone and the Hardware, Sending Bluetooth Commands and controlling the devices is the other part.

Arduino microcontroller act as the brain of the whole system. The microcontroller can be connected with other circuits to perform required functions. The Arduino microcontroller using IC ATMega328P-PU and it works by entering the suitable program that is created. Bluetooth module used in this particular circuit is the HC-05, which requires a 3.3 V DC power drawn from the Arduino microcontroller circuit (pin 3.3 V), Pin (TX 1) is a pathway transmit / send data on the Bluetooth module HC-05 with microcontroller and Pin (Rx 0) as the receive path / receiver data on the HC-05 Bluetooth module with microcontroller while the path Ground is a line connecting the data between HC-05 Bluetooth module with the microcontroller circuit. This system has input from android Smartphone using Arduino software (v0.11), the overall system is controlled automatically and the output is a solenoid that connected to the Arduino microcontroller circuit.

No	System Block	Function
1	Arduino Microcontroller	As data processing center
2	Android Smartphone (Andruino v0.11)	As data input
3	Bluetooth Module Hc-05	As data receiver
4	Battery and Adaptor (12V)	As the power supply
5	Driver Relay	As switch

6	LED	As indicator
7	Solenoid	As system output

The App shown in the fig 3 & Fig.4 that is developed to integrate the user and hardware system should be paired in order to control the devices. The Bluetooth of the smartphone is switched ON and the app is enabled and it is made to scan the available Bluetooth devices around the system. It is made sure that the Bluetooth module is switched ON and it is scanable. The Bluetooth of the Smartphone scans and displays HC-05 if it is available. The pairing pin which is usually 1234 or 0000 is entered into the screed and both the devices are paired. The pairing is checked by connecting the android app with the Bluetooth module

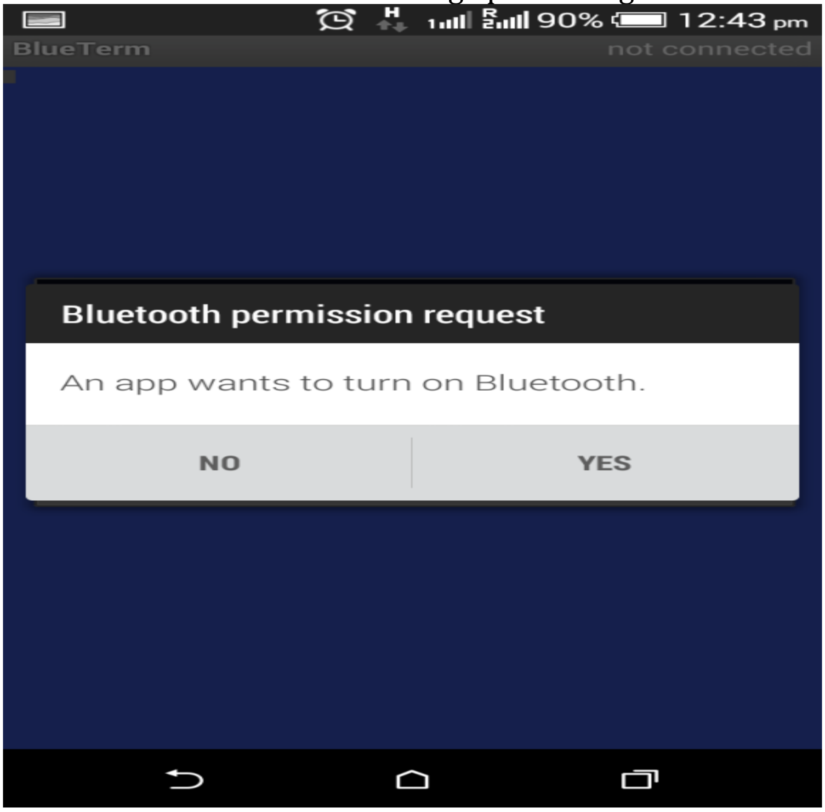


Warning

This application needs the Bluetooth. Do you want turn on?

Figure 3: App showing Bluetooth turned OFF

Open Source Computer Visionis a collection of programming functions which is stored as a library used particularly for real-time computer vision, developed by Intel Russia research center in Nizhny Novgorod. The library is cross-platform in nature. It focuses concentrated on real-time image processing.



Bluetooth permission request

An app wants to turn on Bluetooth.

Figure 4: App Showing Request to Bluetooth

7. Conclusion:

The prime objective of the project is to assist handicapped/old aged people. This project gives basic idea of controlling various home devices using Android phone. This project is based on Android platform which is Free Open Source Software. The total implementation cost of the project is very cheap and it is affordable by a common person. The consumer can interact with the application interface of the android phone and send control signal to the Arduino which in turn will control the other embedded devices.

We have introduced the design and implementation of a low cost Smartphone based home automation system. This home based automation system can be easily manufactured on a large scale for mass productionbecause of its simplicity and ease of design. Another advantage is that application software is based on Android, which has the largest Smartphone base. With improvements in technology of Android software which is open source, cheap Smart phones can be used as the controller in our project, making the total system cost affordable for mass production.

Further enhancements can be implemented on the system such as the incorporation of an intelligent controller that controls the various home devices based on various factors like pressure, humidity, temperature etc. we can also easily integrate Google's Cloud Messaging to control our home systems from the Internet, thus allows to monitor our home appliances from anywhere in the world.

We can also build cross platform system that can be implemented on various platforms like iOS, Windows. Security cameras can be controlled by allowing the user to observe the activity around a house. Security systems can include motion sensors that will detect any kind of unauthorized manual or object movement and notify the user immediately. Scope of this project can also be expanded to many areas other than house. It will be flexible to support various wired as well as wireless technologies like Bluetooth, Zigbee, Wi-Fi, World Wide Web.

8. References:

1. N. Sriskanthan and Tan Karande, "Bluetooth Based Home Automation Systems," Journal of Microprocessors and Microsystems, 2002, Vol. 26, pp. 281-289
2. 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
3. N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless Sensor Network Based Home Monitoring System for Wellness Determination of Elderly", in proc.IEEE Recent Adv.Intell. comput. Syst., june.2012,pp. 1-8
4. Hiroshi Kanma, Noboru Wakabayashi, Ritsuko Kanazawa &Hirimichi Ito, "Home Appliance Control System over Bluetooth with a Cellular Phone", in IEEE, 2003, pp. 1049-1053
5. S. K. Das, D. J. Cook, A. Bhattacharya, E. O. Heierman, III, and T.-Y. Lin, "The Role of Prediction Algorithms on the MavHome Smart Home Architectures," IEEE Wireless Communications (Special Issue on Smart Homes), Vol. 9, No. 6, pp. 77–84, Dec. 2002
6. Shepherd R, "Bluetooth Wireless Technology in the Home." Electronics & Communication Engineering Journal 13 (2001): 195-203. IEEE/IEE Electronic Library. 15 Oct. 2007
7. R. Piyare, M.Tazil" Bluetooth Based Home Automation System Using Cell Phone", 2011 IEEE 15th International Symposium on Consumer Electronics
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 Vo l. 2, SP 1, December 2011

9. Al-Ali, Member, IEEE & M. AL-Rousan,“Java-Based Home Automation System” IEEE Transactions on Consumer Electronics, Vol. 50, No. 2, MAY 2004
10. D. Naresh, B.Chakradhar, S.Krishnaveni, “Bluetooth Based Home Automation and Security System Using ARM9”, International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 9- Sep 2013.
11. R. A. Ramlee, M.H.Leong, R.S.S.Singh, M.M.Ismail, M.A.Othman, H.A.Sulaiman, M.H.Misran, M.A.Meor Said,(2013) “Bluetooth Remote Home Automation System Using Android Application” Vol. 2, pp 149-153.
12. Sharon panth, T.Lee and C.YeeSoh. “Internet-Based Monitoring of Distributed Control Systems-An Undergraduate Experiment”. IEEE Transaction on Education, Vol. 45, No. 2, May 2002
13. Syed Anwaarullah; Chao-Lin Wu. “An integrated, flexible, and Internet-based control architecture for home automation system in the internet era”. Proceedings ICRA `02. IEEE International Conference on Robotics and Automation, Vol. 2, pp.1101-1106, 2002

Appendix:

WEB PROGRAM:

Android_Manifest.xml:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="es.pymasde.blueterm">
<application android:debuggable="false"
android:icon="@drawable/icon" android:label="@string/app_name">
<activity android:configChanges="keyboardHidden|orientation"
android:label="@string/app_name" android:name=".BlueTerm">
<intent-filter>
<actionandroid:name="android.intent.action.MAIN"/>
<category
android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
<activity android:configChanges="keyboardHidden|orientation"
android:label="@string/select_device"
android:name=".DeviceListActivity"
android:theme="@android:style/Theme.Dialog"/>
<activity android:name="TermPreferences"/>
</application>
<uses-permission
android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permissionandroid:name="android.permission.BLUETOOTH"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
</manifest>
```

Custom_Title.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayoutandroid:gravity="center_vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
```

```
xmlns:android="http://schemas.android.com/apk/res/android">
<TextViewandroid:ellipsize="end" android:id="@id/title_left_text"
android:layout_width="wrap_content"
android:layout_height="fill_parent" android:singleLine="true"
android:layout_weight="1.0" android:layout_alignParentLeft="true"
style="?android:attr/windowTitleStyle" />
<TextViewandroid:textColor="#ffffffff" android:ellipsize="end"
android:id="@id/title_right_text" android:layout_width="wrap_content"
android:layout_height="fill_parent" android:singleLine="true"
android:layout_weight="1.0" android:layout_alignParentRight="true" />
</RelativeLayout>
```

Device_List.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutandroid:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent"
xmlns:android="http://schemas.android.com/apk/res/android">
<TextViewandroid:textColor="#ffffffff"
android:id="@id/title_paired_devices" android:background="#ff666666"
android:paddingLeft="5.0dip" android:visibility="gone"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/title_paired_devices" />
<ListViewandroid:id="@id/paired_devices"
android:layout_width="fill_parent"
android:layout_height="wrap_content" android:stackFromBottom="true"
android:layout_weight="1.0" />
<TextViewandroid:textColor="#ffffffff"
android:id="@id/title_new_devices" android:background="#ff666666"
android:paddingLeft="5.0dip" android:visibility="gone"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/title_other_devices" />
<ListViewandroid:id="@id/new_devices"
android:layout_width="fill_parent"
android:layout_height="wrap_content" android:stackFromBottom="true"
android:layout_weight="2.0" />
<Button android:id="@id/button_scan"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/button_scan" />
</LinearLayout>
```

Device_Name.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<TextViewandroid:textSize="18.0sp" android:padding="5.0dip"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
xmlns:android="http://schemas.android.com/apk/res/android" />
```