

Domotics Using Android Smartphone

Sumit Vikram Tripathi¹, Ritukar², Murthy B³, Dr. K. S. Jagadeesh Gowda⁴

U.G. Student, Department of Computer Science & Engineering, Sri Krishna Institute of Technology, Bengaluru, India¹

U.G. Student, Department of Computer Science & Engineering, Sri Krishna Institute of Technology, Bengaluru, India²

Assistant Professor, Department of Computer & Engineering, Sri Krishna Institute of Technology, Bengaluru, India³

Head of Department, Computer & Engineering, Sri Krishna Institute of Technology, Bengaluru, India⁴

ABSTRACT: Domotics using android smartphone is a research project completely focused on creating an eco-system capable of automating any android smartphone into universal remote controller by the help of some hardware and android application extension. In order to use this service; User need to install the application into his or her mobile phone and then using a custom developed a 8051 enabled microcontroller the entire hardware can be automated which in turn automates the home appliances.

KEYWORDS: Android, Domotics, Microcontroller.

I. INTRODUCTION

Domotics is intended to give the user the ability to manage all the equipments that are capable of being automated, from a single controlling source i.e. Android smartphone. The purpose of this project is to present an implementation model of automation with the help of LED bulbs using advanced Android mobile. The Android mobile acts as the "Universal Remote Controller". Bluetooth is selected as a medium of communicating the mobile phone with the components.



(Fig. 1: Domotics along with android smartphone and platform architecture)

Despite equipments from a variety of manufacturers, using a variety of automation protocols, are all abstracted into a common metaphor that allows it to be controlled easily by a single system. All the automation equipments are viewed in a consistent and uniform way in this project, by the help of software application that is developed and installed on the Android phone can automate the devices. Bluetooth is the technology used for communication with the devices. Some of the benefits of implementing Domotics using Bluetooth technology include low power, small form factor size, less cost, robust design, mobile phone connectivity, multi-vendor or manufacturer inter-operability, license-free, etc.

II. RELATED WORK

[1] Provides an insight into achieving the dream of living in a Smart Home.[2]Has described about the use of X10 technology; X10 technology uses electrical wires for powering the devices.[3]Has spoken about Home Automation using Wi-Fi. However, implementation Home Automation using Wi-Fi technology is not always secure. [4] Have spoken about Z wave technology;Z-Wave is a wireless communications protocol designed for home automation.The major drawback of Z wave is it can transmit only up to 75 feet.[5]Gives a training to the programmers on programming the 8051 microcontroller. [6]Provides an introduction to Bluetooth programming, with a specific focus on developing real code and with special emphasis on how they relate to other networking technologies.

III. CONCEPTS, PROPOSED APPROACH, SCENARIOS AND CONSTRAINTS

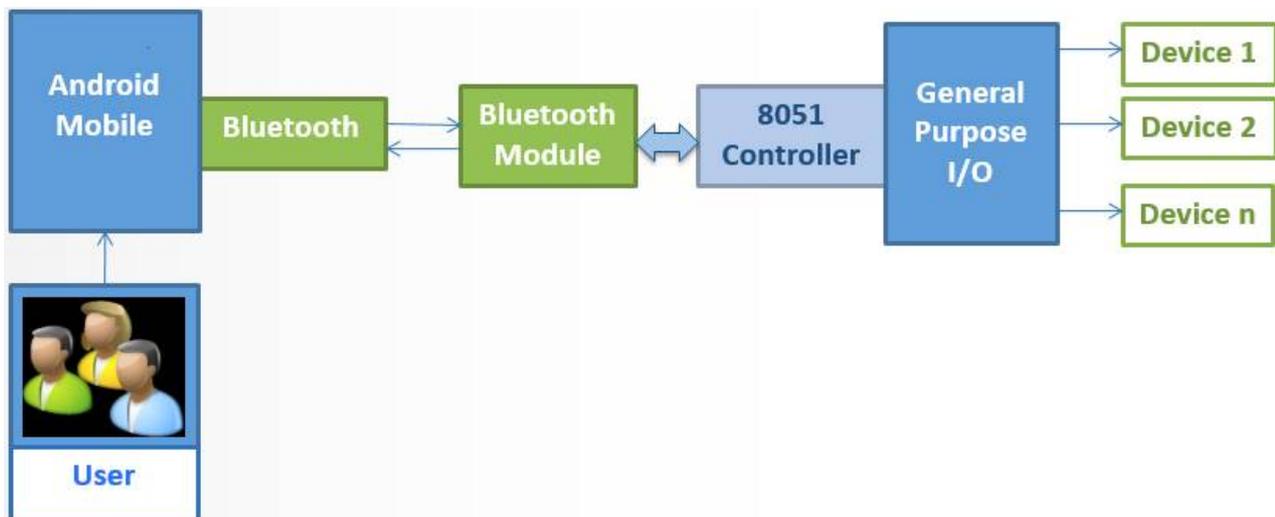
3.1 Problem Definition: To develop an Android application which can be used as a universal remote controller

Components required:

1. Bluetooth enabled Android smartphone
2. An Android application which will enable user to interact with the board
3. A Bluetooth chip connected to the board for establishing communication with Android smartphone
4. 8051 Microcontroller for processing request sent from Android smartphone
5. Relays, LED bulbs, which are the components being automated from a controlling source

Outputs: A “Universal Remote Controller” which is capable of turning ON/OFF devices connected to it.

3.2 Working Principle



(Fig. 2: Functional diagram of the system)

The implementation of Smart-Home automation is mainly divided into **three modules**:

- Module I** : Sender Side
- Module II** : Working of Sensor
- Module III** : Receiver Side

3.2.1: Module I: Sender Side

- Initiate the touch events in order to perform an operation.
- Know the UUID of the Android device.
- View the list of devices available for connecting.
- Once connected, send the request to the receiving side to perform an operation.

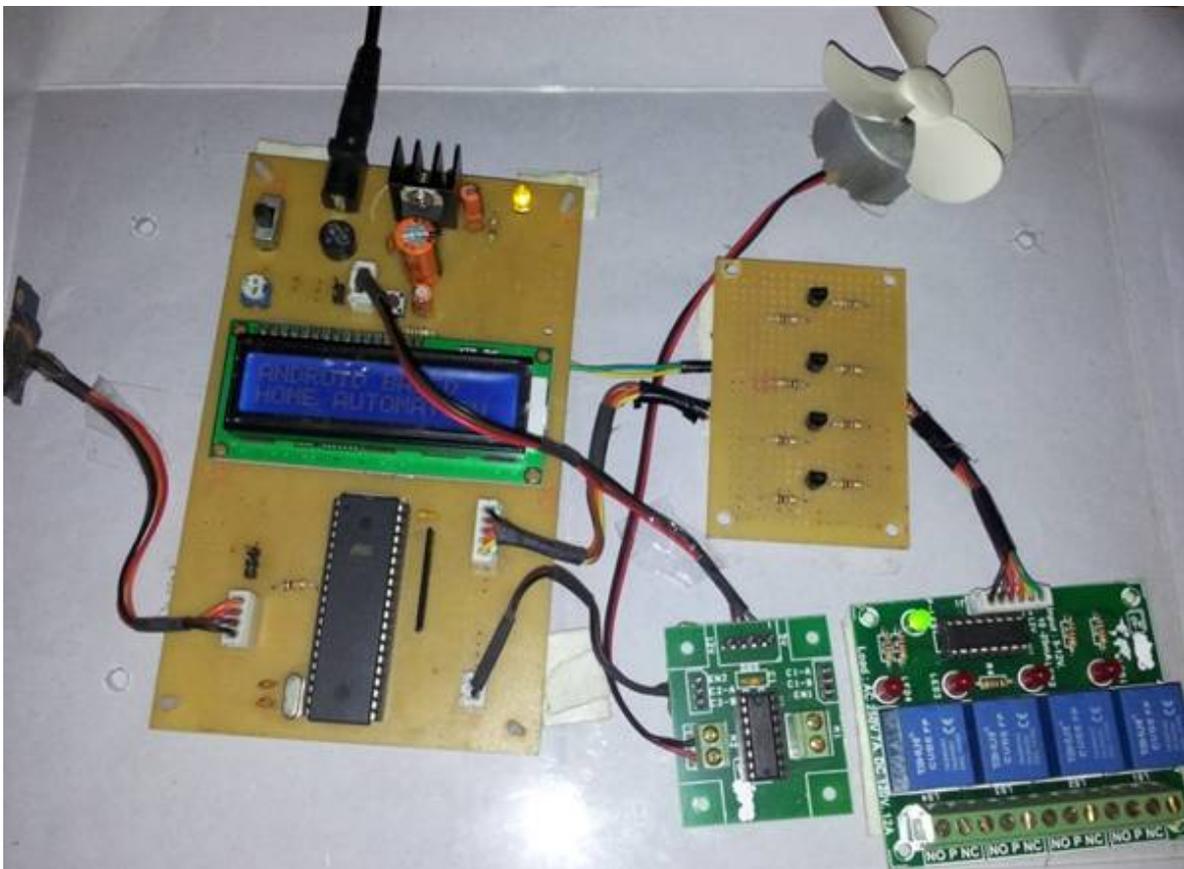
3.2.2: Module II: Working of Sensor

- Detect the number of devices that are currently visible.
- Pair devices that need to communicate.
- Check if any input or output data to be sent is got.
- Send input data or write output data if available, else report an error.

3.2.3: Module III: Receiver Side

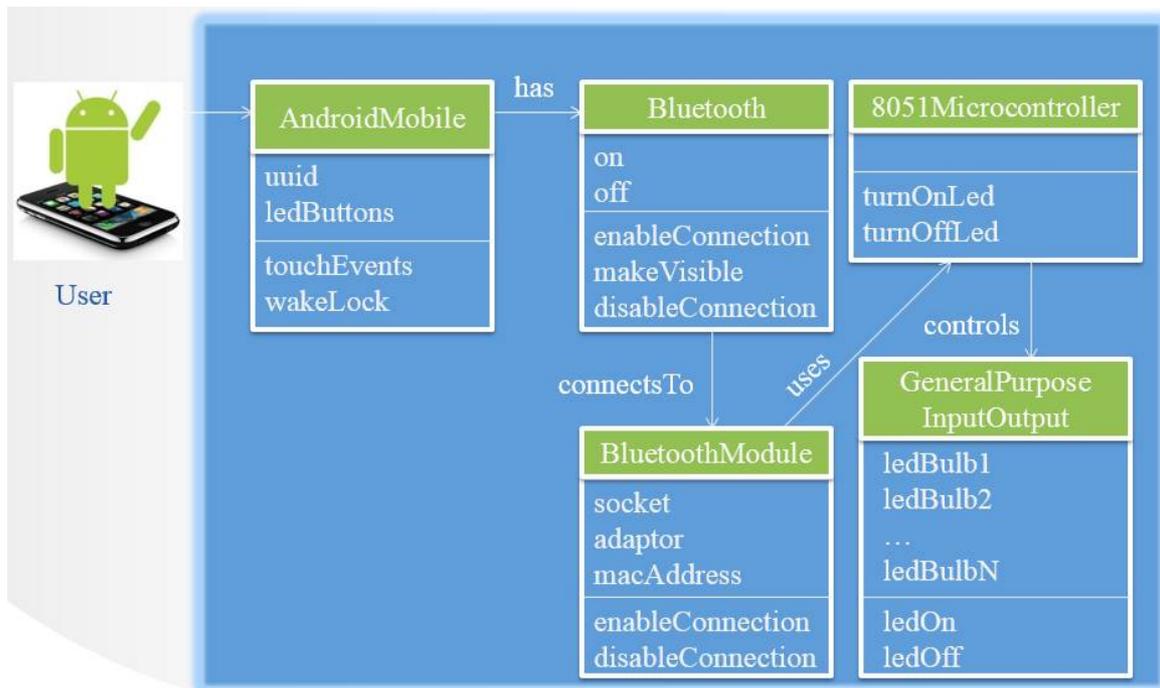
- Assign MAC address to the Bluetooth device.
- Check if the request to connect to a device has been received.
- Glow the LED bulb, if the state is "ON".
- Switch off the LED bulb, if the state is "Off".

3.3 The Proposed Model



(Fig.3:proposed modelHome automation system)

3.4 Class Diagram



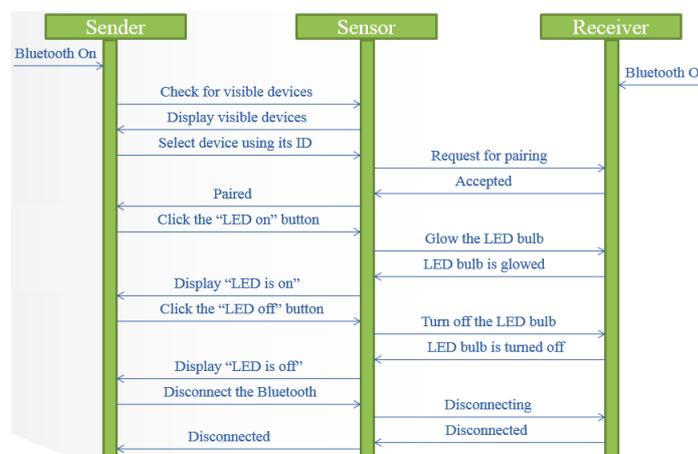
(Fig 4: Class Diagram)

3.5 Assumptions and Dependencies

One of main dependencies is that the Android SDK environment that is prepared using Eclipse as the IDE, with a plug-in for the SDK and the Android Platform for which it will be developed.

3.6 Scenarios

- **Scenario 1: Sequence diagram – Normal Scenario**

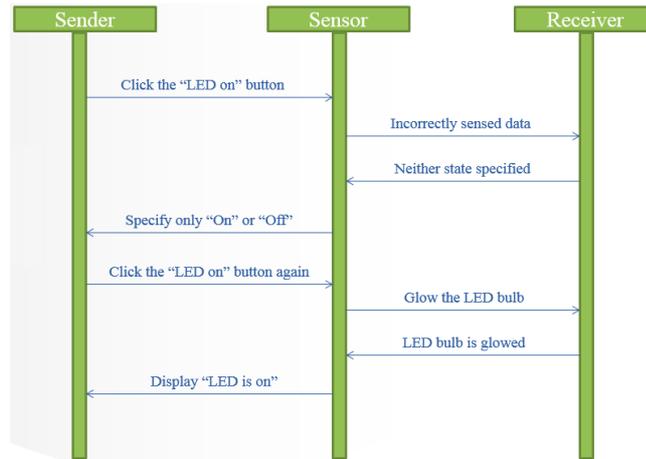


(Fig. 5: Sequence diagram – Normal Scenario)

The above figure shows the sequence diagram for a normal scenario. As shown in the figure, after the Bluetooth on both the sender and the receiver side is switched on, the sensor checks for visible devices. The required device is selected using its ID and paired between the sender and the receiver. The data is sensed correctly on the

receiver side if either ON or OFF state is specified by the user. Bluetooth is disconnected on either side after the required operation is performed.

• **Scenario 2:Exception Scenario**



(Fig. 6: Sequence diagram – Exception Scenario)

The above figure shows the sequence diagram for an exception scenario. This causes a problem for connection establishment between the sender and the receiver.

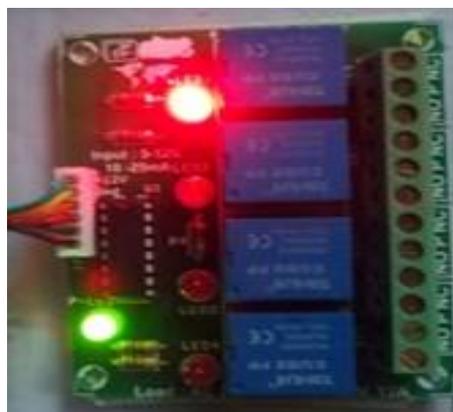
The sender acts as an exception handler. Once an exception is caused, the sender responds to it by specifying either one of the ON or OFF states.

3.7 Constraints

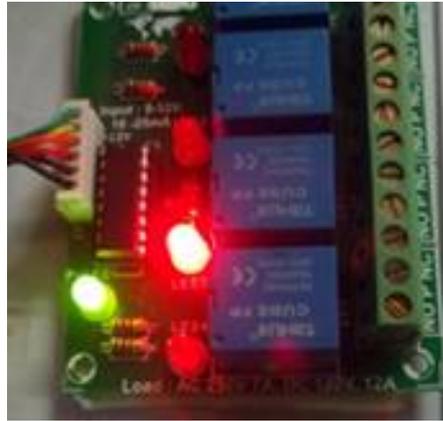
The constraints pertaining to my system are:

- Android Operating System must be installed on the device.
- Bluetooth has a limited range.

IV. EXPERIMENT AND RESULT



(Fig. 7: Turning ON Load-1 [Fan])



(Fig. 8: Turning OFF Load-1 [Fan])

ACKNOWLEDGEMENT

We would like to thank my HOD and project guide for helping us to find out the required resources in order to complete this project work.

REFERENCES

- [1] Richard Harper, in his book "Inside the Smart Home" (2003)
- [2] Kenneth J. Ayala, in his book "The 8051 Microcontroller" (2004)
- [3] Technica Pacifica, in the book "Easy X10 Projects for Creating a Smart Home" (2005)
- [4] Albert S. Huang, in his book "Bluetooth Essentials for Programmers" (2007)
- [5] Dr. Jose A. Gutierrez, in his book "Wireless Hart" (2010)
- [6] Paulo Novais, Kasper Hallenbog, Dante I. Tapia and Juan M, in the book "Ambient Intelligence Software and Applications" (2012)
- [7] Platform Architecture "<https://developer.android.com/guide/platform/index.html>"
- [8] Android architecture components "<https://developer.android.com/topic/libraries/architecture/index.html>"
- [9] Guide to Application architecture "<https://developer.android.com/topic/libraries/architecture/guide.html>"