A Project Report

On

Lab Automation Using GSM And Bluetooth

Submitted towards the partial fulfilment of the requirement for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

2012-2016

BY

Anirudh Kulkarni (1601-12-733-094)

Surya Teja Sharma (1601-12-733-121)

Under the supervision of

Mr.A.Mohan

Asst Professor

Mrs.Vimala Manohara Ruth

Asst Professor

Department of CSE

CBIT,Hyderabad.

Department of Computer Science and Engineering

Chaitanya Bharathi Institute of Technology, Gandipet, Hyderabad-500075

CERTIFICATE

This is to certify that the project work entitled “Lab Automation Using GSM And Bluetooth” is a bonafide work carried out by Mr Anirudh Kulkarni (1601-12-733-094) and Mr Surya Teja Sharma (1601-12-733-121) in partial fulfilment of the requirements for the award of degree of BACHELOR OF ENGINEERING in COMPUTER SCIENCE AND ENGINEERING by the OSMANIA UNIVERSITY, Hyderabad, under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guides

Mr.A.Mohan

Asst Professor

Department of CSE

CBIT, Hyderabad.

Mrs.Vimala Manohara Ruth

Asst Professor

Department of CSE

CBIT, Hyderabad.

Head of the Department

Dr. Y.Rama Devi

Professor and Head

Department of CSE

CBIT, Hyderabad.

DECLARATION

This is to certify that the project entitled "**Lab Automation Using GSM And Bluetooth**" is a record of work done by us in the Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Osmania University. The reports are based on the project work done entirely by us and not copied from any other source.

Anirudh Kulkarni

(1601-12-733-094)

Surya Teja Sharma

(1601-12-733-121)

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude and indebtedness to our project supervisor **Mr.A.Mohan**, Asst. Professor, **Mrs.Vimala Manohara Ruth**, Asst. Professor and the Head of the Department **Dr. Y. Rama Devi** whose valuable suggestions and interests played a major role in the successful completion of this project.

Anirudh Kulkarni

(1601-12-733-094)

Surya Teja Sharma

(1601-12-733-121)

LIST OF FIGURES

LIST OF SCREENSHOTS

Screenshot Number	Screenshot Name	Page Number

CONTENTS

ABSTRACT

The main objective of this project is to develop a home automation system using an Arduino board with Bluetooth and GSM being remotely controlled by any Android OS smart phone. As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralised control system, involving remote controlled switches.

Presently, conventional wall switches located in different parts of the house makes it difficult for the user to go near them to operate. Even more it becomes more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a most modern solution with smart phones.

In order to achieve this, a Bluetooth module and a GSM module is interfaced to the Arduino board at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF commands to the receiver where loads are connected.

The end user makes use of Bluetooth wireless connection or SMS that enables the system communicates with graphical user interface (GUI) on smart phone without cable. The target home appliances (in our case, a lamp) are controlled by the system Main Control Board as described above.

The mobile application is built using Hybrid Mobile Application development techniques. Hybrid development combines the best of both the native and HTML5 worlds. We define hybrid as a web app, primarily built using HTML5 and JavaScript, that is then wrapped inside a thin native container that provides access to native platform features.

CHAPTER 1

INTRODUCTION

1.1 What is Home Automation?

Home automation refers to the automatic and electronic control of household features, activity, and appliances. Various control systems are utilised in this residential extension of building automation. Home automation gives you access to control devices in your home from a mobile device anywhere in the world. The term may be used for isolated programmable devices, like thermostats and sprinkler systems, but home automation more accurately describes homes in which nearly everything including lights, appliances, electrical outlets, heating and cooling systems are hooked up to a remotely controllable network.

One of the greatest advantages of home automation systems is that users can be protected against break-ins and fires, while enjoying automations for lights, temperature, and more. The automation of features in one's home helps to promote security, comfort, energy efficiency, and convenience. Another benefit of home automation systems is the amount of labor, time, energy and materials that is saved.

Home automation system represents and reports the status of the connected devices in an intuitive, user-friendly interface allowing the user to interact and control various devices with the touch of a few buttons. Some of the major communication technologies used by today's home automation system include Bluetooth, WiMAX and Wireless LAN (Wi-Fi), Zigbee, and Global System for Mobile Communication (GSM).

1.2 Technologies Adopted

Our project uses two of the above mentioned technologies namely, Bluetooth and Global System for Mobile Communication (GSM).

1.Bluetooth

Wireless technologies are becoming more popular around the world and the consumers appreciate this wireless lifestyle which gives them relive of the well known “cable chaos” that tends to grow and uses a lot of space. Now with the embedded Bluetooth technology, digital devices form a network in which the appliances and devices can communicate with each other. Today, home automation is one of the major applications of Bluetooth technology. Operating over unlicensed, globally available frequency of 2.4GHz, it can link digital devices within a range of 10m to 100m at the speed of up to 3Mbps depending on the Bluetooth device class. With this capability of Bluetooth, we propose a home automation system based on Bluetooth technology.

Bluetooth is a radio standard and communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 1 meter, 10 meters, 100 meters) based on low-cost transceiver microchips in each device.
Bluetooth lets these devices communicate with each other when they are in range. The devices use a radio communications system, so they do not have to be in line of sight of each other, and can even be in other rooms, as long as the received transmission is powerful enough.

Class	**Maximum Permitted Power** (mW/dBm)	**Range (approximate)**
Class 1	100 mW (20 dBm)	~100 meters
Class 2	2.5 mW (4 dBm)	~10 meters
Class 3	1 mW (0 dBm)	~1 meter

Bluetooth is implemented in a variety of new products such as phones, printers, modems, and headsets. Bluetooth is acceptable for situations when two or more devices are in proximity to each other and don't require high bandwidth. Bluetooth is most commonly used with phones and hand-held computing devices, either using a Bluetooth headset or transferring files from phones/PDAs to computers.Bluetooth also simplifies the discovery and setup of services. Bluetooth devices advertise all services they provide. This makes the utility of the service that much more accessible, without the need to worry about network addresses, permissions and all the other considerations that go with typical networks.

2.Global System for Mobile Communication (GSM)

As mentioned previously, apart from Bluetooth, which has a range limited to few meters, our prototype also has a GSM module where in, one can control the home appliances using the simple GSM based phone, just by sending SMS through his phone.

GSM (Global System for Mobile Communications) is one of the most widely used cellular technologies in the world. With the increase in the number of GSM subscribers, research and development is heavily supported in further investigating the GSM implementation. The microcontroller acts as the bridge between the GSM network and sensors of the home automation system. Internet and wireless communications have also been utilised in parallel with GSM for home automations. Among the cellular technologies, GSM network is preferred for the communication between the home appliances and the user due to its wide spread coverage which makes the whole system online for almost all the time. Another advantage of using the GSM network in home automation is its high security infrastructure, which provides maximum reliability whereby other people cannot monitor the information sent or received. Hence, this research work implements SMS based control for home appliances using the GSM architecture without accessing the local network.

1.3 Microcontroller

Our project uses an Arduino microcontroller to process the user instructions and perform desirable actions. Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Some of the features of the Arduino microcontroller include -

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

1.3 Mobile Application

We developed a mobile application to establish communication with the prototype. The mobile application is developed using hybrid technology. Hybrid development combines the best of both the native and HTML5 worlds. We define hybrid as a web app, primarily built using HTML5 and JavaScript, that is then wrapped inside a thin native container that provides access to native platform features.

Native applications are developed for a specific platform and installed on a computing device. Web applications are generalised for multiple platforms and not installed locally

but made available over the Internet through a browser. Hybrid apps are often mentioned in the context of mobile computing.

We have used a framework named ionic framework for building our mobile application. Ionic is a complete open-source SDK for hybrid mobile app development. Built on top of AngularJS and Apache Cordova, Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova

Ionic provides all the functionality that can be found in native mobile development SDKs. Users can build their apps, customise them for Android or iOS, and deploy through Cordova. Ionic includes mobile components, typography, interactive paradigms, and an extensible base theme.

Using Angular, Ionic provides custom components and methods for interacting with them. One such component, collection repeat, allows users to scroll through a list of thousands of items without any performance hits. Another component, scroll-view, creates a scrollable container with which users can interact using a native-influenced delegate system.

Supported platforms : Ionic is focused on building for modern Web standards and for modern mobile devices. For Android, Ionic supports Android 4.1 and up. For iOS, Ionic supports iOS 7 and up. Ionic Framework, powered by Angular.js, supports BlackBerry 10 apps.

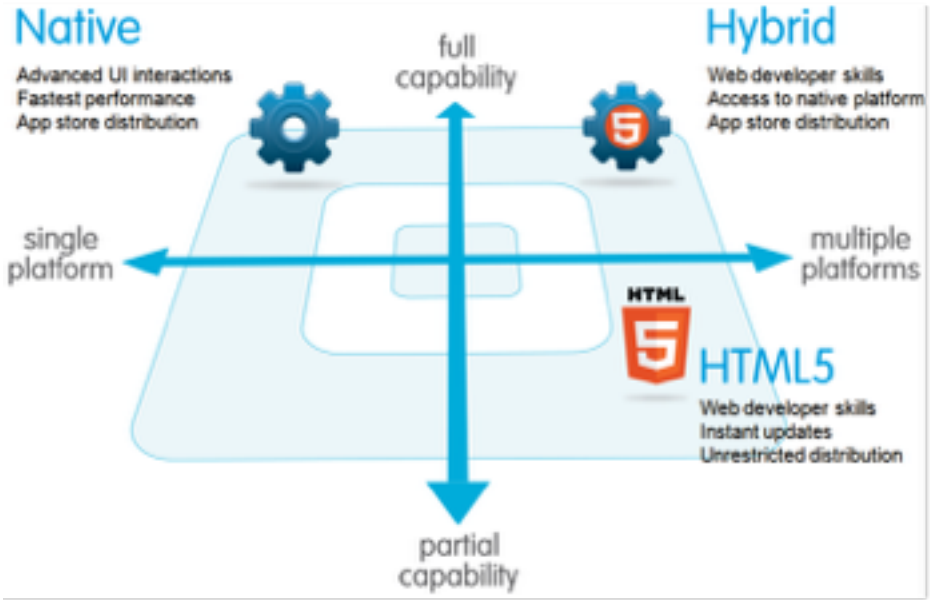


Figure 1.1 : Overview of Hybrid Applications

CHAPTER 2

LITERATURE SURVEY

CHAPTER 3

PROBLEM STATEMENT and REQUIREMENTS

3.1 Problem Statement

The field of Automation has well advanced in Industries, as majority of automobile industry plants as well as bottling plants have Automated assembly lines. But automation has not yet penetrated in the homes especially in India.

If automation was to be used in homes than everyday life would be get eased. Simple example of use of automation in home can be seen in the operation and control of electrical appliances like tube light, fan, television sets, motors etc. This process eases the every time effort the user has to put in for the operation of the same and save energy.

Also, people are getting more acquainted daily with the use of Smartphone and tablets which are capable of doing much of PC's work handy. So we have decided to make a low cost Embedded System in which the smart phones can be used to help automate entire home. In this system the user will have remote access and control over all the subsystems present in the house.

We can automate home using various wireless technologies such as Bluetooth, WI-Fi, Zig-Bee, GPS and GSM etc. For controlling as well as monitoring the entire home system we are going to use Bluetooth module, GSM Module and an Android phone. As android is open source software we can easily make an application on it as per our necessities and requirements. This application will provide an interface between users and the actual appliances which he wishes to control. By designing entire embedded system with the help of the controlling, receiver modems and the devices we would prepare an actual model of the home automation system which will be cheap and easy enough to implement in our homes and other places.

3.2 Hardware Requirements

Microcontroller	Arduino
Add-on Module - 1	GSM Module - SIM900A
Add-on Module - 2	Bluetooth Module - HC-05
Relay	4-channel Relay
Connectors	M-M, M-F
Testing Devices	Lamp

3.3 Software Requirements

Operating System	Android OS
APK File	Android application package

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 SYSTEM DESIGN - Hardware

4.1.1 System Components

1. Arduino Uno Board

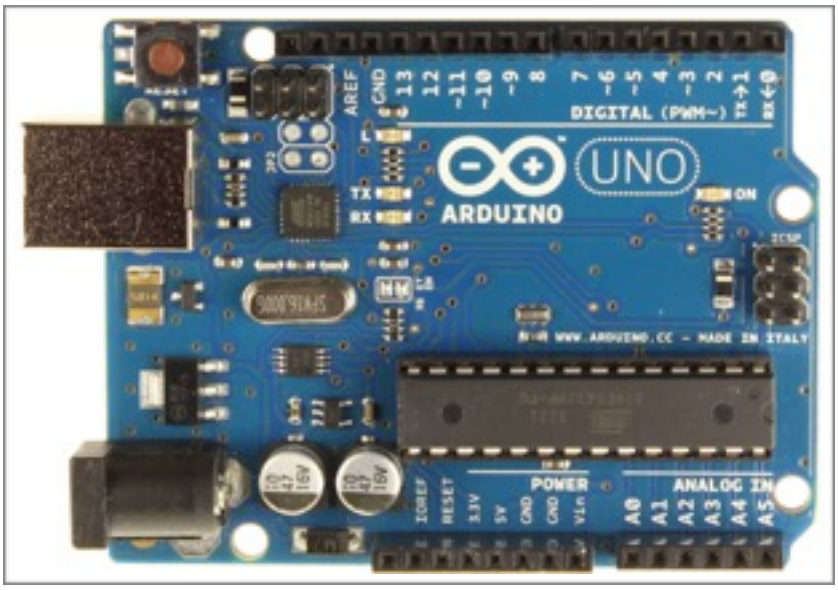


Figure 4.1 : Arduino Uno Board

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/ output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

- Technical Specifications

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
Length	68.6 mm
Width	53.4 mm
Weight	25 g

- Programming

The Uno can be programmed with the Arduino Software (IDE). Select "Arduino/Genuino Uno" from the Tools > Board menu (according to the microcontroller on your board).

The ATmega328 on the Uno comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP.

- Power

The Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- Vin. : The input voltage to the Uno board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V : This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3.3 V : A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Uno board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage

and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

- Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

- Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using **pinMode()**, **digitalWrite()**, and **digitalRead()** functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialised functions:

- Serial : 0 (RX) and 1 (TX).Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts : 2 and 3.These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM : 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

- SPI : 10 (SS), 11(MOSI), 12(MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED : 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI : A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

There are a couple of other pins on the board:

- AREF : Reference voltage for the analog inputs. Used with analogReference().
- Reset : Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

- Communication

The Uno has a number of facilities for communicating with a computer, another Uno board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but

not for serial communication on pins 0 and 1). A SoftwareSerial library allows serial communication on any of the Uno's digital pins.

2. Bluetooth Module (HC-05)

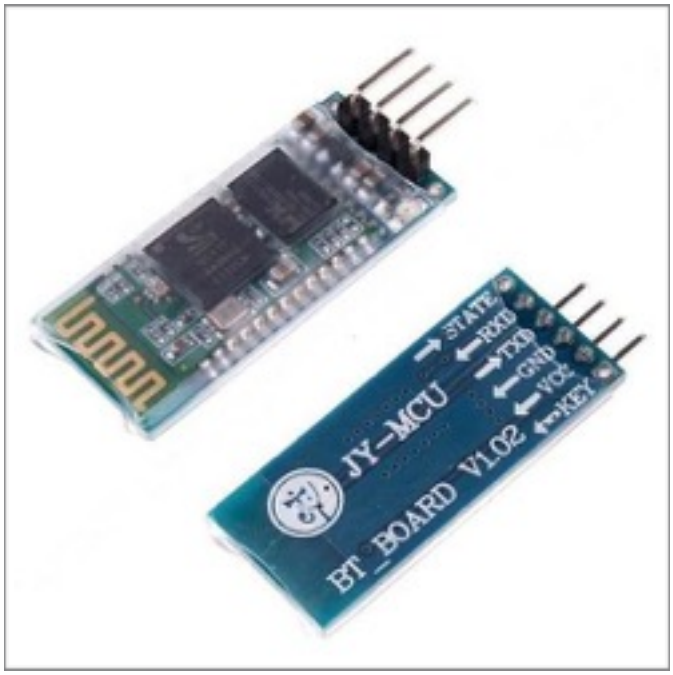


Figure 4.2 : Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm.

3. GSM Module (SIM900A)

Figure 4.3 : GSM Module

GSM/GPRS Modem-RS232 is built with Dual Band GSM/GPRS engine- SIM900A, works on frequencies 900/ 1800 MHz. The Modem is coming with RS232 interface,which allows you connect PC as well as microcontroller with RS232 Chip(MAX232). The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface.

The onboard Regulated Power supply allows you to connect wide range unregulated power supply . Using this modem,you can make audio calls, SMS, Read SMS, attend the incoming calls and internet through simple AT commands

4. Relay 4-channel

Figure 4.4 : 4-channel Relay

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal.

Relays are found hidden in all sorts of devices. In fact, some of the first computers ever built used relays to implement Boolean gates.

4.1.2 System Configuration

- Interfacing Bluetooth Module (HC-05) to Arduino

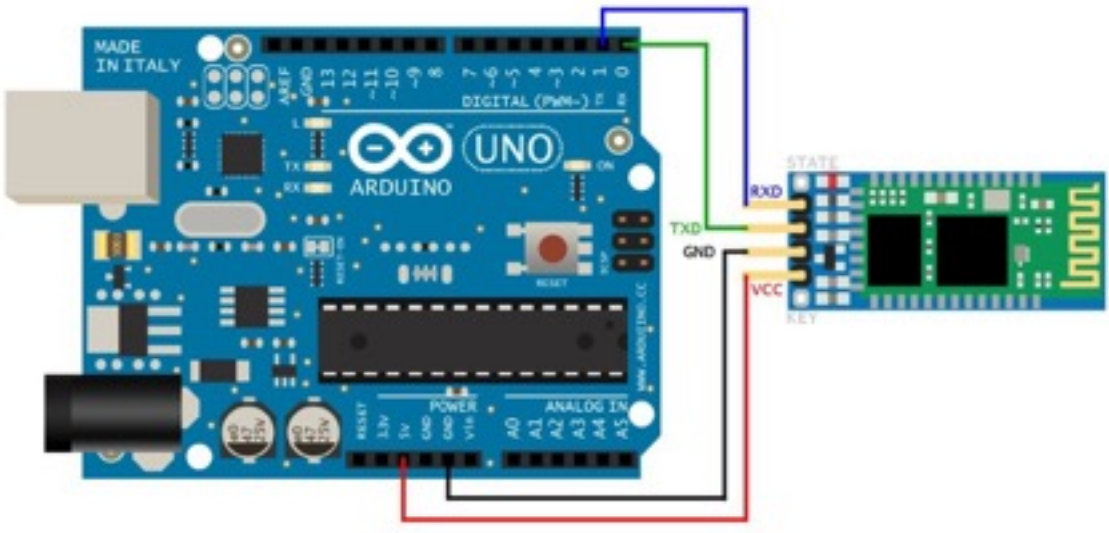


Figure 4.5 : Interfacing HC-05 and Arduino

The Bluetooth module is interfaced to the Arduino uno board as shown in the above diagram. To establish serial mode of communication between the bluetooth module and Arduino, the serial transmission pin of the bluetooth module is connected to the Rx pin (Pin no. 0) of the Arduino. Similarly, the serial transmission pin of the Arduino is connected to the Rx pin of the bluetooth module.

Apart for these, the GND pins and voltage pins of both, Arduino and bluetooth module are connected as shown above.

- Interfacing GSM Module (SIM900A) to Arduino

BOOTING UP THE GSM:

1. Insert the SIM card to module and lock it.

2. Connect the adapter to module and turn it ON!

3. Now wait for some time (say 1 minute) and see the blinking rate of ‘status LED’ (GSM module will take some time to establish connection with mobile network.

4. Once the connection is established successfully, the status LED will blink continuously every 3 seconds.

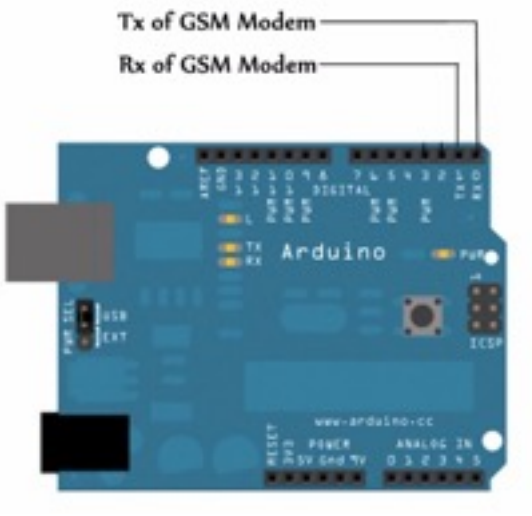


Figure 4.6 : Interfacing SIM900 and Arduino

There are two ways of connecting GSM module to Arduino. In any case, the communication between Arduino and GSM module is serial. We are supposed to use serial pins of Arduino (Rx and Tx). So if you are going with this method, you may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino.

Now connect the ground pin of Arduino to ground pin of GSM module. You made 3 connections and the wiring is over. Now you can load different programs to communicate with GSM module and make it work.

The problem with the above connection is while programming, Arduino uses serial ports to load program from the Arduino IDE. If these pins are used in wiring, the program will

not be loaded successfully to Arduino. So, we have to disconnect wiring in Rx and Tx each time you burn the program. Once the program is loaded successfully, we can reconnect these pins and have the system working.

To avoid this difficulty, We are using an alternate method in which two digital pins of Arduino are used for serial communication. We need to select two PWM enabled pins of Arduino for this method. So we chose pins 9 and 10 (which are PWM enabled pins). This method is made possible with theSoftwareSerial Library of Arduino. SoftwareSerial is a library of Arduino which enables serial data communication through other digital pins of Arduino. The library replicates hardware functions and handles the task of serial communication.

4.1.3 System Architecture

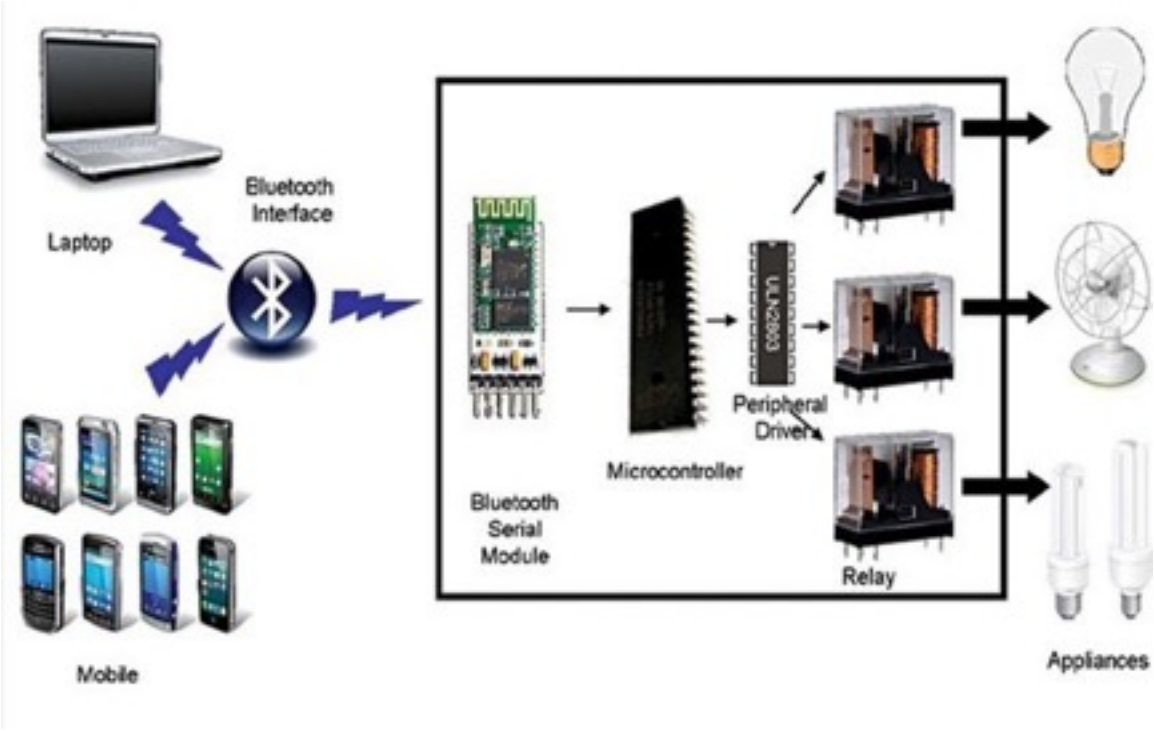


Figure 4.7 System Architecture

This home automation system consists of two main hardware components: the cell phone and the Arduino uno board connected to the bluetooth and GSM module. The cell phone hosts the hybrid application which enables the user to access the home appliances and also the control commands for the appliances. This hybrid application communicates with the Bluetooth and GSM module, and sets up an ad-hoc communication protocol between the two devices, which allows controlling the behaviour of the Arduino uno board and hence the electronic devices via relay.

An Arduino uno board is a 8-bit microcontroller board based on the ATmega328P. We connect this board to a HC-05 Bluetooth module and a SIM900A GSM module. Arduino supports wireless serial communication over Bluetooth and GSM. This board has 14

digital output ports, 5 analog input pins, 16kB of flash memory, pulse width modulator and extra hardware resources which makes it suitable for the required task.

The Arduino uno board can be programmed wirelessly over the Bluetooth connection using the high-level Arduino Programming language. The Bluetooth or the GSM antenna picks up the packets sent from the cell phone. Subsequently, these packets containing the appliance status commands are pipelined through ATmega328P microcontroller and the designed analogue circuitry according to the definition of each output. Different home appliances are connected to the digital output ports of the Arduino Uno board via relays to provide sufficiently high currents and voltage compatibility.

For test purposes, 25W, 240V lamps have been used. Sending commands from software to turn ON/OFF a device may not guarantee the successful operation of the device as the device may be defective.

4.2 SYSTEM DESIGN - Software

4.2.1 Modules

Our mobile application has the following modules -

- Bluetooth Module
- GSM Module
- Security Module
- Embedded Code for Arduino

4.2.1.1 Bluetooth Module

To establish communication between the mobile application and the bluetooth module (HC-05) we make use of Cordova plugin for bluetooth. A plugin is a bit of add-on code that provides JavaScript interface to native components. They allow your app to use native device capabilities beyond what is available to pure web apps. A Cordova plugin bridges a bit of functionality between the WebView powering a Cordova application and the native platform the Cordova application is running on. Plugins are composed of a single JavaScript interface used across all platforms, and native implementations following platform-specific Plugin interfaces that the JavaScript will call into. It should be noted that all of the core Cordova APIs are implemented using this exact architecture.

We used a plugin named Bluetooth Serial Plugin for PhoneGap. This plugin enables serial communication over Bluetooth. It was written for communicating between Android or iOS and an Arduino.

The process of **Installing** this plugin is as follows -

Install with Cordova cli

$ cordova plugin add cordova-plugin-bluetooth-serial

The list of **Supported Platforms** for this plugin include :

- Android
- iOS with RedBearLab BLE hardware, Adafruit Bluefruit LE, Laird BL600, or BlueGiga
- Windows Phone 8

The **Limitations** for this plugin are as follows :

- The phone must initiate the Bluetooth connection.
- iOS Bluetooth Low Energy requires iPhone 4S, iPhone5, iPod 5, or iPad3+.
- Will not connect Android to Android.
- Will not connect iOS to iOS.

The **Methods** that we used to establish connection include :

- bluetoothSerial.connect : To connect to a Bluetooth device.

  Method Prototype:

  bluetoothSerial.connect(macAddress_or_uuid, connectSuccess, connectFailure);

  Description : Function connect connects to a Bluetooth device. The callback is long running. Success will be called when the connection is successful. Failure is called if the connection fails, or later if the connection disconnects. An error message is passed to the failure callback.

- bluetoothSerial.connectInsecure

  Method Prototype:

  bluetoothSerial.connectInsecure(macAddress, connectSuccess, connectFailure);

Description : Function connectInsecure works like connect, but creates an insecure connection to a Bluetooth device.

- bluetoothSerial.disconnect

Method Prototype:

bluetoothSerial.disconnect([success], [failure]);

Description : Function disconnect disconnects the current connection.

- bluetoothSerial.write

Method Prototype:

bluetoothSerial.write(data, success, failure);

Description : Function write data to the serial port. Data can be an ArrayBuffer, string, array of integers, or a Uint8Array. Internally string, integer array, and Uint8Array are converted to an ArrayBuffer. String conversion assume 8bit characters..

- bluetoothSerial.available

Method Prototype:

bluetoothSerial.available(success, failure);

Description : Function available gets the number of bytes of data available. The bytes are passed as a parameter to the success callback.

- bluetoothSerial.read

Method Prototype:

bluetoothSerial.read(success, failure);

Description : Function read reads the data from the buffer. The data is passed to the success callback as a String. Calling read when no data is available will pass an empty String to the callback.

- bluetoothSerial.enable

Method Prototype:

bluetoothSerial.enable(success, failure);

Description : enable is only supported on Android and does not work on iOS or Windows Phone. If enable is called when Bluetooth is already enabled, the user will not prompted and the success callback will be invoked.

4.2.1.2 GSM Module

To establish communication between the mobile application and the GSM module (SIM900A) we make use of Cordova plugin for SMS.

We used a plugin named Cordova SMS Plugin. This plugin enables communication over SMS. It was written for communicating between Android or iOS and an Arduino.

The process of **Installing** this plugin is as follows -

Using the Cordova CLI and NPM, run:

cordova plugin add cordova-sms-plugin

The list of **Supported Platforms** for this plugin include :

- Android
- iOS
- Windows Phone 8

- Windows 10 Universal

The **Methods** that we used to establish connection include :

- sendSMS : To send a short message to a device.

  Method Prototype:

  sendSMS(address(s), text, successCallback, failureCallback);

  Description : Function send SMS to another GSM device. The callback is long running. Success will be called when the SMS delivery is successful. Failure is called if the delivery fails, or timeout occurs. An error message is passed to the failure callback.

4.2.1.3 Security Module

As more organisations explore ways to push functionality to mobile devices, there is a desire to move an increasing amount of sensitive data onto the device. In addition, there is a push to have more sensitive calculations performed on devices.

With the dramatic increase in the number of devices, their technical capabilities, and their use for information-rich transactions attached to back-end enterprise applications, mobile platforms have become an increasingly attractive security target. Adding to the importance of mobile application security is the fact that the networks used to access many mobile services are often unsecured and unencrypted, creating potential confidentiality breaches from traffic monitoring.

Also, it may lead to inappropriate and disastrous consequences. Security breachment gifts the intruder with the access to entire home leading to a huge loss.

To decrease the threat associated with this, we have installed a patten unlocking system as seen in many of the android devices. The user of the application cannot gain access as long as he enters the right pattern.

4.2.1.4 Embedded Code for Arduino

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

4.3 Data Flow Diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modelling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

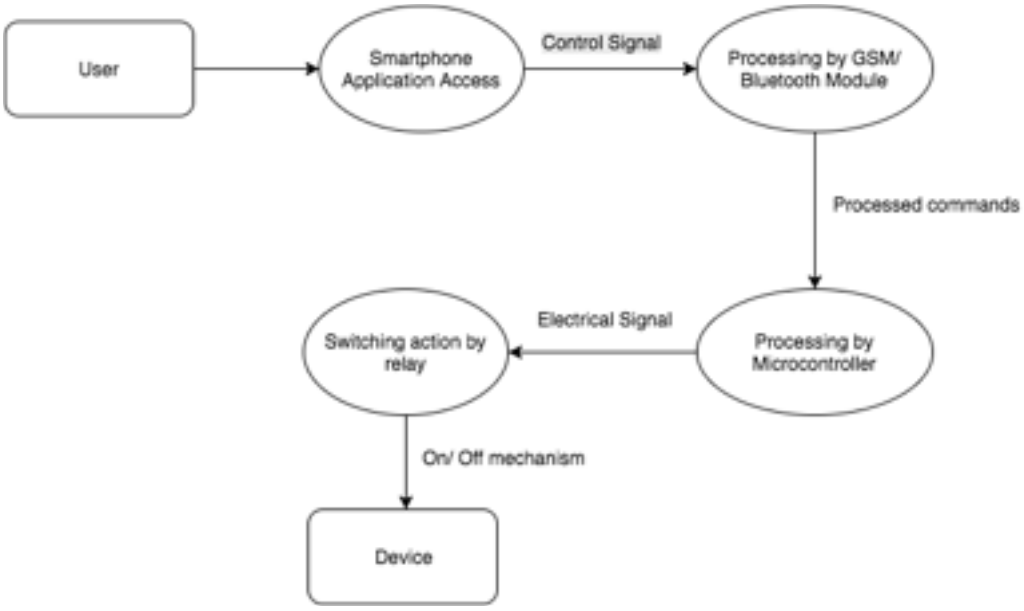


Figure 4.8 Data Flow Diagram

The Arduino uno board can be programmed wirelessly over the Bluetooth connection using the high-level Arduino Programming language.

The Bluetooth or the GSM antenna picks up the packets sent from the cell phone. Subsequently, these packets containing the appliance status commands are pipelined through ATmega328P microcontroller and the designed analogue circuitry according to the definition of each output. Different home appliances are connected to the digital output ports of the Arduino Uno board via relays to provide sufficiently high currents and voltage compatibility.

4.4 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardised general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualisation, Constructing and documenting the artefacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialisation mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.4.1 Use Case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

There are two main actors in our project, one being the end user with the mobile application and the other being the device itself.

The user of the mobile application has two options at his disposal for establishing communication with the microcontroller, and hence controlling the electronic devices. The first option for communication is via Bluetooth and the second one being short message service (SMS).

The use case diagram for our project is as follows :

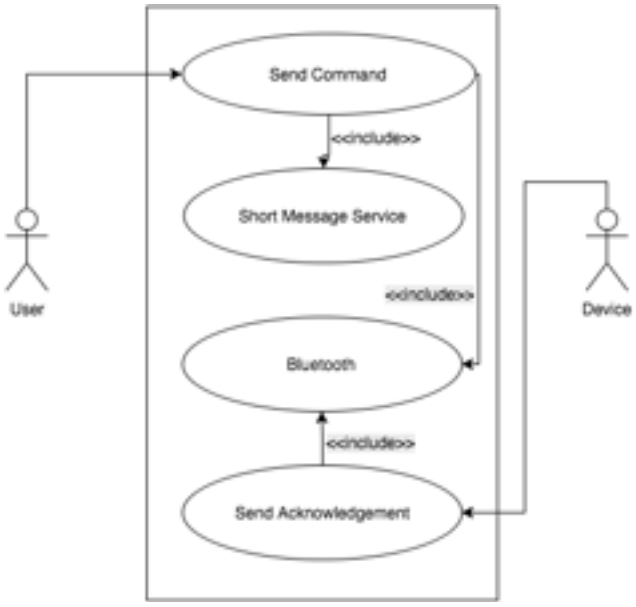


Figure 4.9 Use Case Diagram

4.4.2 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

The class diagram for our project has 5 main classes - Android Application, Microcontroller, Bluetooth Module, GSM Module and the device.

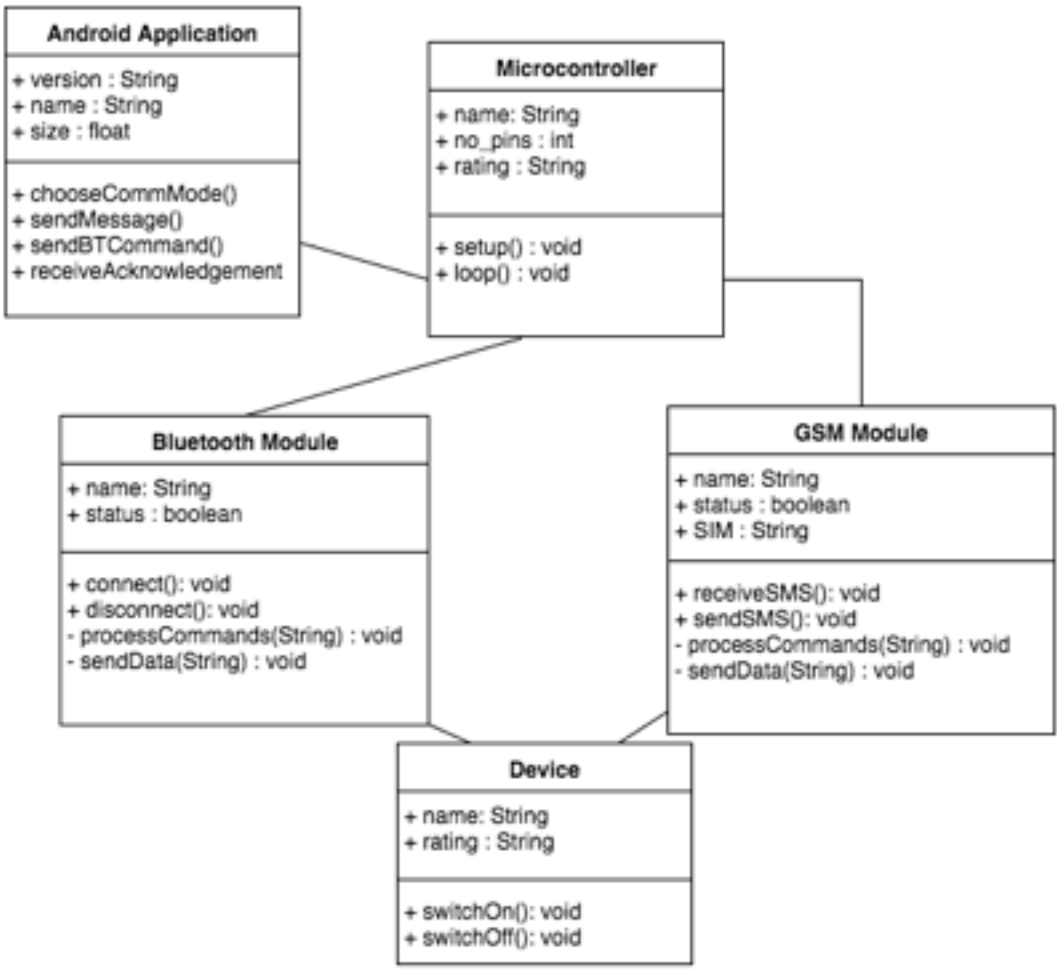


Figure 4.10 Class Diagram

4.4.3 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

The sequence diagram has 4 main objects namely - Android application, GSM/Bluetooth Module, Microcontroller and the Device that communicate among one another using different messages as shown below.

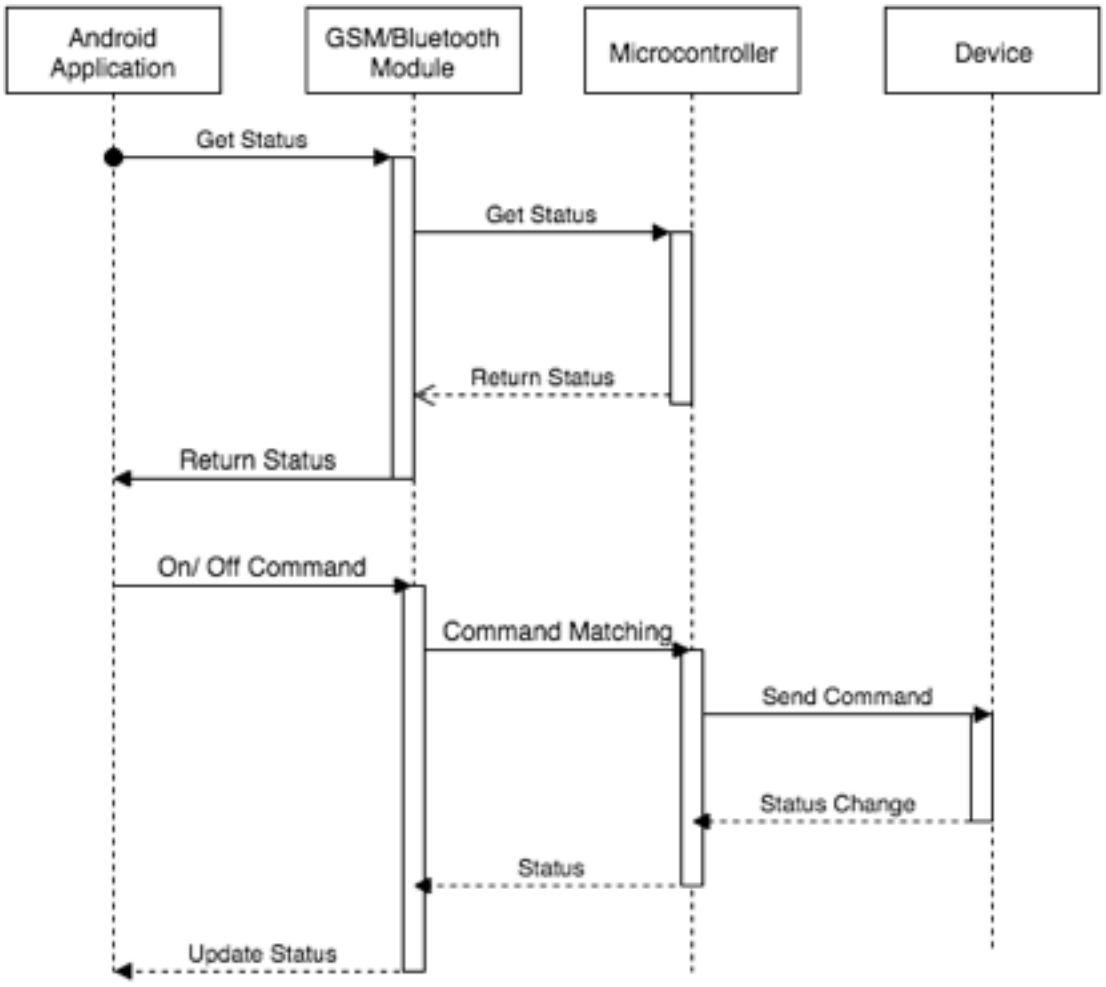


Figure 4.11 Sequence Diagram

4.4.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

The activity diagram is pretty straight forward and self explanatory.

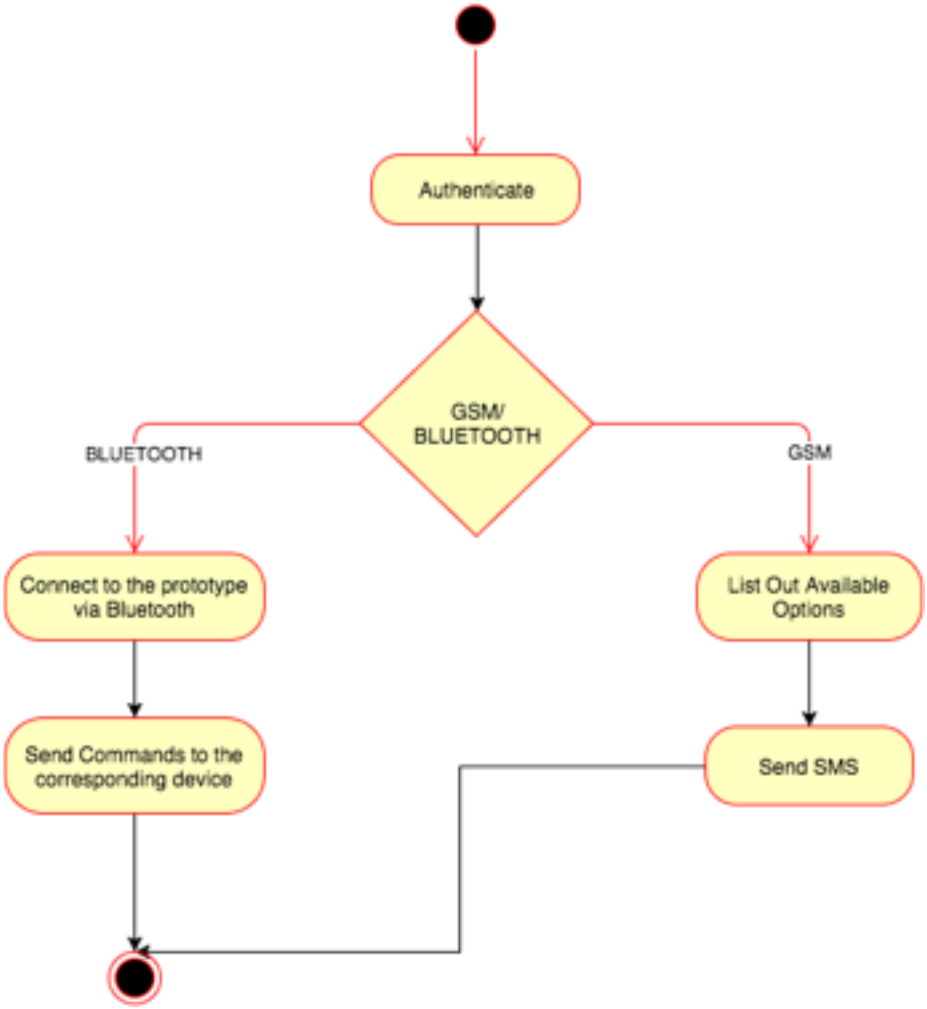


Figure 4.12 Activity Diagram

CHAPTER 5

TESTING AND RESULTS

5.1 TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

We conducted unit testing parallely with the development process. We divided the testing process into two parts. In the first part we made sure that the hardware was functioning as desired and in the second part, we performed various combinations of tests on our mobile application.

5.1.1.1 Testing the Hardware

This phase was divided into multiple small phases as follows,

- **Test Case 1** : The testing process started with troubleshooting the Arduino uno board itself. We made sure that the Arduino board was functioning correctly by writing a program to make the on-chip LED glow.
- **Test Case 2** : After the bluetooth module was interfaced with the Arduino, we made sure that the connections were right using a bluetooth enabled smart phone. We used a mobile application, Bluetooth Terminal to communicate with the Arduino. We wrote a program in ADE which enabled us to control the on-chip Arduino LED via bluetooth. Using the Bluetooth Terminal application on the mobile phone we were able to switch the LED ON/OFF.
- **Test Case 3 :** After the GSM module was interfaced with the Arduino, we made sure that the connections were right by sending SMS form a mobile phone. We used the in-built texting application to communicate with the Arduino via SMS. We wrote a program in ADE which enabled us to control the on-chip Arduino LED via SMS. We were able to switch the LED ON/OFF by sending text messages.
- **Test Case 4** : Next, we connected the relay to the Arduino board. In this test case we combined all the above test cases. We tested the connections by making the relay's on chip LED glow via bluetooth and SMS.
- **Test case 5** - Here, we connected a lamp to the relay and made it glow ON/OFF via bluetooth and SMS.

5.1.1.2 Testing the Mobile Application

We started off by developing the bluetooth module of our mobile application and the testing it in similar manner as the Test Case 2 in section 5.1.1.1.

We then integrated the SMS module into the mobile application and tested it in a manner similar to the Test Case 3 in section 5.1.1.1.

5.1.2 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

5.1.3 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

5.1.4 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

5.2 RESULTS

5.2.1 Screenshots

Screenshot 5.1 Home page

Screenshot 5.2 User Registration Page

Screenshot 5.3 Admin Login

Screenshot 5.4 User Login

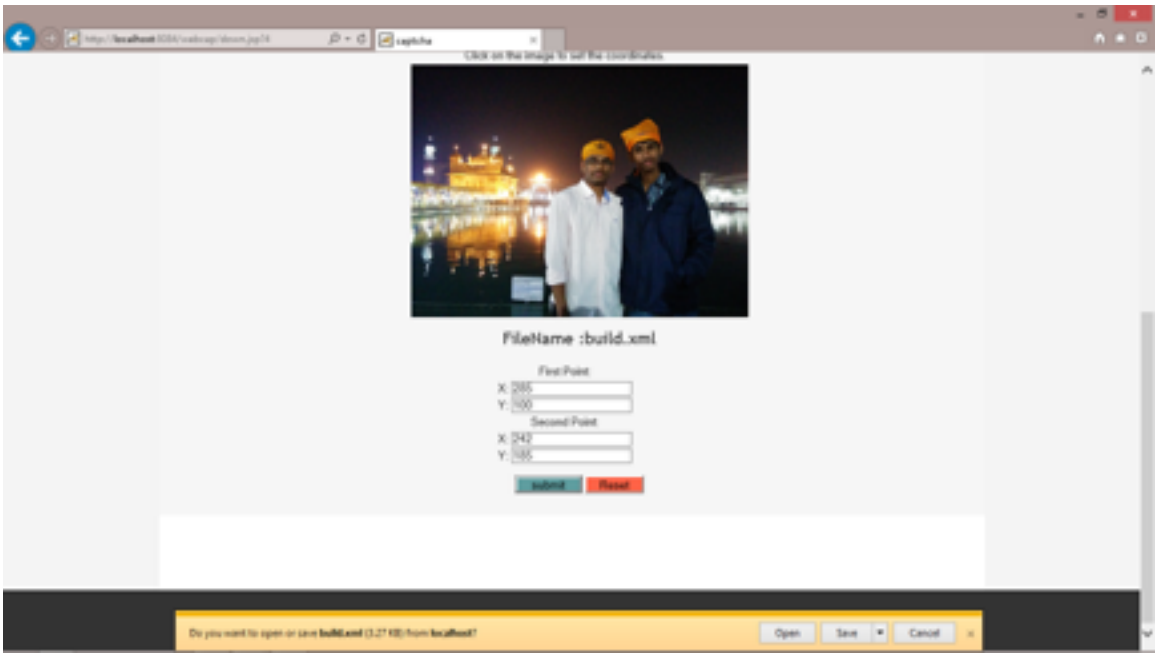
Screenshot 5.5 User Login Contd.

Screenshot 5.6 User Successful Login

Screenshot 5.7 File Upload

Screenshot 5.8 Uploaded Files ready for download

Screenshot 5.9 Download Process



Screenshot 5.10 File being Downloaded

Screenshot 5.11 Admin Managing Users

Screenshot 5.12 Upload History

Screenshot 5.13 Download History

CONCLUSION

Home automation systems have evolved into practical, almost essential, parts of a home. The days of wild science fiction speculation have transformed into systems that help homeowners monitor lighting, heating, and cooling systems. Homeowners have complete control over the security of their homes, from programming sensors to alert them whenever a potentially dangerous event is about to occur to allowing them to ensure that every door has been locked. Home automation systems enhance the quality of home theatres, by connecting audio and video devices with a central processing unit. They also provide homeowners with convenience, security, and above all, savings. Reduced utility bills are just one way homeowners recoup the initial large investment of a home automation system.

We have introduced design and implementation of a low cost, flexible and wireless solution to the home automation. The system is secured for access from any user or intruder. The users are expected to enter the unlocking pattern on cell phone to access the home appliances. This adds a protection from unauthorised users. This system can be used as a test bed for any appliances that requires on-off switching applications without any internet connection.

Our system has both short-range (bluetooth) and long-range (GSM) connectivity. Bluetooth is better for short-range communication as there is less delay in comparison to GSM mode of communication. But, GSM has global connectivity and hence, range is never an issue when it comes to data transmission.

The full functionality of the home automation system was tested and the wireless communication between the cellphone and Arduino was found to be limited to <50m in case of bluetooth and it was global in case of GSM.

REFERENCES

[1] R. Biddle, S. Chiasson, and P. C. van Oorschot, “Graphical passwords: Learning from the first twelve years,” *ACM Comput. Surveys*, vol. 44, no. 4, 2012.

[2] (2012, Feb.). *The Science Behind Passfaces* [Online]. Available: http://www.realuser.com/published/ScienceBehindPassfaces.pdf

[3] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, “The design and analysis of graphical passwords,” in *Proc. 8th USENIX Security Symp.*, 1999, pp. 1–15.

[4] H. Tao and C. Adams, “Pass-Go: A proposal to improve the usability of graphical passwords,” *Int. J. Netw. Security*, vol. 7, no. 2, pp. 273–292, 2008.

[5] S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon, “PassPoints: Design and longitudinal evaluation of a graphical password system,” *Int. J. HCI*, vol. 63, pp. 102–127, Jul. 2005.

[6] P. C. van Oorschot and J. Thorpe, “On predictive models and userdrawn graphical passwords,” *ACM Trans. Inf. Syst. Security*, vol. 10, no. 4, pp. 1–33, 2008.

[7] K. Golofit, “Click passwords under investigation,” in *Proc. ESORICS*, 2007, pp. 343–358.

[8] A. E. Dirik, N. Memon, and J.-C. Birget, “Modeling user choice in the passpoints graphical password scheme,” in *Proc. Symp. Usable Privacy Security*, 2007, pp. 20–28.

[9] J. Thorpe and P. C. van Oorschot, “Human-seeded attacks and exploiting hot spots in graphical passwords,” in *Proc. USENIX Security*, 2007, pp. 103–118.

[10] P. C. van Oorschot, A. Salehi-Abari, and J. Thorpe, "Purely automated attacks on passpoints-style graphical passwords," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 393–405, Sep. 2010.

[11] P. C. van Oorschot and J. Thorpe, "Exploiting predictability in clickbased graphical passwords," *J. Comput. Security*, vol. 19, no. 4, pp. 669–702, 2011.

[12] T. Wolverton. (2002, Mar. 26). *Hackers Attack eBay Accounts* [Online]. Available: http://www.zdnet.co.uk/news/networking/2002/03/26/hackers-attack-ebay-accounts-2107350/

[13] HP TippingPoint DVLabs, Vienna, Austria. (2010). *Top Cyber Security Risks Report, SANS Institute and Qualys Research Labs* [Online]. Available: http://dvlabs.tippingpoint.com/toprisks2010

[14] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in *Proc. ACM CCS*, 2002, pp. 161–170.

[15] P. C. van Oorschot and S. Stubblebine, "On countering online dictionary attacks with login histories and humans-in-the-loop," *ACM Trans. Inf. Syst. Security*, vol. 9, no. 3, pp. 235–258, 2006.

[16] M. Alsaleh, M. Mannan, and P. C. van Oorschot, "Revisiting defenses against large-scale online password guessing attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 1, pp. 128–141, Jan./Feb. 2012.

[17] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," in *Proc. Eurocrypt*, 2003, pp. 294–311.

[18] S. Chiasson, P. C. van Oorschot, and R. Biddle, “Graphical password authentication using cued click points,” in *Proc. ESORICS*, 2007, pp. 359–374.

[19] S. Chiasson, A. Forget, R. Biddle, and P. C. van Oorschot, “Influencing users towards better passwords: Persuasive cued click-points,” in *Proc. Brit. HCI Group Annu. Conf. People Comput., Culture, Creativity, Interaction*, vol. 1. 2008, pp. 121–130.

[20] D. Davis, F. Monrose, and M. Reiter, “On user choice in graphical password schemes,” in *Proc. USENIX Security*, 2004, pp. 1–11.

[21] R. Dhamija and A. Perrig, “Déjà Vu: A user study using images for authentication,” in *Proc. 9th USENIX Security*, 2000, pp. 1–4.

[22] D. Weinshall, “Cognitive authentication schemes safe against spyware,” in *Proc. IEEE Symp. Security Privacy*, May 2006, pp. 300–306.

[23] P. Dunphy and J. Yan, “Do background images improve ‘Draw a Secret’ graphical passwords,” in *Proc. ACM CCS*, 2007, pp. 1–12.

[24] P. Golle, “Machine learning attacks against the Asirra CAPTCHA,” in *Proc. ACM CCS*, 2008, pp. 535–542.

[25] B. B. Zhu *et al.*, “Attacks and design of image recognition CAPTCHAs,” in *Proc. ACM CCS*, 2010, pp. 187–200.