

East West University

Department of Electrical and Electronic Engineering

Project Report

on

Home Automation System with Remote Access through Web Server

by

Md. Merajul Islam	2010-2-80-008
A.E.M Obeydullah Siddique	2010-2-80-032
Mojibur Rahman	2010-2-80-037

Submitted to the
Department of Electrical and Electronic Engineering
Faculty of Sciences and Engineering
East West University
Dhaka, Bangladesh

In partial fulfillment of the requirements for the degree of Bachelor of Science in
Electrical and Electronic Engineering (B.Sc. in EEE)

Spring 2016

Approved by

Supervisor

Dr. MuhammedMazharul Islam

Chairperson

Dr. MuhammedMazharul Islam

Acknowledgment

At first, we are grateful to Almighty Allah for giving us the strength to do this project. We express our deep gratitude to our supervisor Dr. Muhammed Mazharul Islam, Chairperson, Department of Electrical and Electronic Engineering, East West University, Dhaka, for his helpful guidance, supervision, encouragement and constant support during this project work.

We would also like to thank our academic advisors Dr. AnisulHaque, Ms. SohanaTanzeem and also other faculty members, office staffs and East West University in general.

At last we want to thank our parents and all our friends for their moral support and helpful discussions during this project work.

Abstract

In our homes, we have to control our appliances for our comfort. As most of the controlling are done manually, a lot of power is wasted every day if sometimes we forget to switch-off any appliance, which is a very common phenomenon. If we control home appliances through a remote access, then we can easily abrogate the problem. We have designed a system where appliances can be controlled through a web server. This system is also helpful for people who are physically disabled, as the system can be accessed through cell phone or computer connected to the internet.

The proposed system has three sub-systems, i.e. (1) hardware for appliance control, (2) interface between hardware and web server and (3) web based server. We are using Arduino in the hardware level to control appliances. To work as an interface between the server and Arduino, we have developed an application in Python programming language. The web server serves up a website; we used PHP programming language to design this website through which we can control and also show the on/ off status of home appliances. The appliances are connected with an Arduino board and the Arduino board is connected with a local server (i.e. pc). The local server is communicating with the web server through internet. Through the server, a client gets the authority to monitor and control the on/ off states of the appliances. In order to ensure the security of the system, we have implemented password protection for the clients on the website.

Authorization

We hereby declare that we are the sole authors of this project. We authorize East West University to lend this project to other institutions or individuals for the purpose of scholarly research.

Md. Merajul Islam

Mojibur Rahman

A.E.M Obeydullah Siddique

We further authorize East West University to reproduce this project report by photocopy or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Md. Merajul Islam

Mojibur Rahman

A.E.M Obeydullah Siddique

East West University
Dhaka, Bangladesh
May, 2016

Table of Contents

	<u>Page Number</u>
Acknowledgment	ii
Abstract	iii
Authorization	iv
Chapter 1 Introduction	4
1.1 Objective	4
1.2 Home Automation Systems	4
1.3 Review of Existing Appliance Control Systems:.....	4
1.4.1 Proposed System	6
1.5 Features	7
Chapter 2 Hardware of the System	8
2.1 Introduction	8
2.2 Main Components of the Hardware	8
2.3.1 Arduino board	8
2.3.2 Relay.....	9
2.3.3 PC Connected to Internet	9
2.4 Connection Between Arduino and Home Appliances	10
2.5 Prototype Construction:	11
Chapter 3 Software of the System	13
3.1 Introduction.....	13
3.2 Program for Arduino Board	13
3.3 Program for Local Server.....	16
3.4 Web Server Software Details	18
3.6 System Communication Diagram.	21
3.7 Communication Between Arduino and Web Server.....	22

Chapter 4 Conclusion	23
Draw backs.....	23
Future Development.....	23
References	24
Appendix A: Codes	26
Appendix B: Screenshots of software	45

List of Figures

	<u>Page Number</u>
Figure 1.1: Working principle block diagram.....	6
Figure 3.1: Software flow chart for Arduino	14
Figure 3.2: Software flow chart for local server	15
Figure 3.3 : Operation of the website.....	17
Figure 3.4: Internal structure of database	18
Figure 3.5: Internal structure of user table in database	19
Figure 3.6: Internal structure of appliances status table in database	19
Figure 3.7: User home page of our system	20
Figure 3.8: Password change window	20
Figure 3.9: Communication flow between components of system.....	21
Figure 3.10: Successfully connection established.	22
Figure B.1: Window of successful run of exe file.	45
Figure B.2: Web server directory page.	45

Chapter1

Introduction

In our project, we control home appliances through a centralized controlling process which provide comfort, energy efficiency and security, i.e. the fundamental requirement of home automation [1].

1.1 Objective

The objective of this project is to implement a low cost reliable home automation system that can be used to remotely monitor and control household appliances. The system can remotely switch-on or switch-off any household appliance connected to it. Only authorized persons can observe the state of an appliance (i.e. whether the appliance is turned on or off) from remote place and easily control it.

1.2 Home Automation Systems

Remote controls of home appliances are especially helpful for people with physical disabilities, because for a physically handicapped person, the simple action of getting up to turn a light or fan on/off might be difficult to perform. The systems can also help to reduce utility bills and energy wastage, since we can easily verify and turn off machines that are left turned on in our absence. In fact, sophisticated home automation systems are being developed that can maintain inventory of household items using RFID (radio frequency identification) [2,3]. The system can also incorporate lighting, security, telecommunication and air conditioning in one single controlled system.

1.3 Review of Existing Appliance Control Systems:

There are different types of systems that are currently used for home automation, such as global system for mobile communication (GSM), wireless fidelity (Wi-Fi), serial communication using Bluetooth etc. In GSM based automation, the developer used GSM network and mobile phones for controlling electronics appliances. The GSM modem provides the communication media between the user and the system. A mobile phone is used to control household appliances and to receive a feedback by short messaging service (SMS) about the security and safety. The SMS consists of commands to be executed and the format of SMS is predefined. The message is sent to the GSM modem via the GSM public network. When the GSM modem receives the message, it executed a command through the microcontroller [4].

In Wi-Fi based automation systems, the home appliances are controlled and monitored by a pc and an android phone in the fixed area of Wi-Fi coverage range (i.e. approximately 200 meters). The controlling device for the Wi-Fi based automation is also a microcontroller. In this system, the data sent from pc/cell phone over Wi-Fi will be received by the Wi-Fi module connected with the microcontroller. The microcontroller receives the command from user through the Wi-Fi module and executed the command [5].

There is also a project on home automation based on android application. One of the system introduces an Android OS (operating system) based application for Tablet or Smartphone that communicates with the home appliances through a serial connection. In this project, they used Bluetooth for communication because Bluetooth device is built-in module of android phone. Bluetooth with globally available frequencies of 2400Hz is able to provide connectivity up to 100 meters and speed up to 3Mbps depending on different Bluetooth device classes. The Bluetooth connection is established between the system and the application which was designed and installed in the Android device. The users can easily access the Android application by simply tapping the buttons present on the touch screen of Android device. Android device send the data over Bluetooth which is received by Bluetooth module connected by microcontroller and microcontroller execute the command [6, 7].

In another project, a Bluetooth module (HC-05) is used for the home automation system which is secured and flexible to control the home appliances using an Android app from a cell phone. The Android app is developed in JAVA scripts according to the Bluetooth module specifications. At first, a connection is established between Bluetooth and the FPGA board. The FPGA board is used to control the home appliances which are connected through relays. The communication between the Bluetooth module and the cell phone is wireless and then that module is serially connected to the FPGA board through RS-232 port for the control of home appliances. Initially, the Bluetooth serial module has to be paired with the nearest Bluetooth devices. To control the appliances, an application is developed in FPGA and the password is used to authenticate the communication. After completing the pairing, the module receives the command from the mobile and then it transmits those commands to RS-232 port. All the appliances are controlled by the FPGA board through relays [8].

In another project, they have designed the Android app and pc application for controlling home system. Home appliances like fans, lights and other appliances with relays are connected to an embedded circuit board which is connected to the home pc which is used to provide the authentication to the system for various mobile users to access home appliances.

The communication between the home system and the Android application is made through cloud server using internet or wifi [9].

There are also some commercial companies who provide home automation facilities. One of them is ‘Loxone Home Automation’ system. They provide different types of features such as controlling blinds & shading, lighting, multi zone heating etc. In a ‘Loxone Smart Home’ every device is controlled and monitored by a central hub, called the ‘Miniserver’. Simple push buttons, sensors, as well as complex systems such as photovoltaic installations are fully integrated into ‘Loxone’. It recognizes energy savings potentials, shuts down electricity to stand-by power consuming devices. ‘Vivint Smart Home’ is another company which provides the same type of automation systems [10, 11].

1.4.1 Proposed System

In our system, the appliances to be controlled (such as electrical fans, lights etc.) are connected to an Arduino board. The Arduino board is connected to a computer which acts as a local server. The current state of the appliances can be monitored and the appliances can be controlled through the local server. To access this information, we developed a program which runs on a remote web server. Now, through the web server, users can log-on and view the states as well as control appliances remotely. Users need only a device with access to the internet (i.e. phone, personal computer (pc), tablets etc.) to be able to monitor and control the appliances. The connection is shown in Figure 1.1.

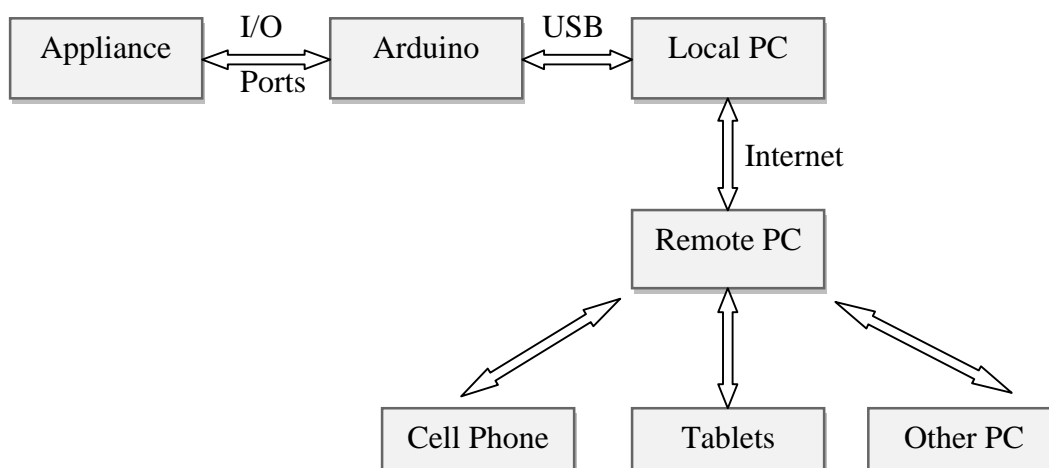


Figure 1.1: Working principle block diagram.

The web server access is password protected, so, the server is secured against unauthorized access. A prototype hardware was developed to demonstrate our system.

To implement the system, we needed to develop three types of programs, i.e.

- (i) a program for the 'Arduino' board to monitor and control the appliances,
 - this program resides in Arduino's memory
 - the programming language used is C
- (ii) a program to communicate between the local server pc (connected to the Arduino) and the web server,
 - this program resides in the local server pc
 - the programming language used is 'python'
- (iii) a program (web application) to communicate between the web server and local server
 - this program resides in the internet (i.e. cloud)
 - the programming language used is 'PHP'

When the user requests connection to the appliance through the internet, first the current state information of the appliance is collected by Arduino, and then this information is carried to the local server pc through USB port. Finally, this information is sent to the web server through http protocol.

1.5 Features

Our proposed system has the following features:

- There's no limitation of proximity in terms of accessing appliances due to the use of web server
- Appliance control is possible via multiple types of devices (e.g. phones, computers etc.) as long as the device has internet access
- Provides security and protection against unauthorized usage

Chapter 2

Hardware of the System

2.1 Introduction

In the home automation system, an Arduino board is used to control and monitor home appliances from remote places. This Arduino board is connected with server pc through USB port. Arduino is an open-source electronics board based on Atmel microcontrollers. The boards are very popular for building prototypes of systems.

2.2 Main Components of the Hardware

The main components used in our project are listed below:

- 1) Arduino board
- 2) Relays
- 3) PC connected to internet

These components are briefly described in the following section.

2.3.1 Arduino board

In this project we used the Arduino Mega 2560. It is based on the ATmega 2560 microcontroller manufactured by Atmel. It has 54 digital input/output pins (including 15 pulse width modulation (PWM) outputs), 16 analog inputs, 4 universal asynchronous receiver-transmitter (UART)s, a 16 MHz crystal oscillator, a USB connection, a power jack, in circuit serial programming (ICSP) header, and a reset button. Figure 2.1 shows the pin diagram of Arduino mega.

Memory:

The ATmega 2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Communication:

The Arduino 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega 2560 provides four hardware universal synchronous receiver transmitters (UART) for serial communication.

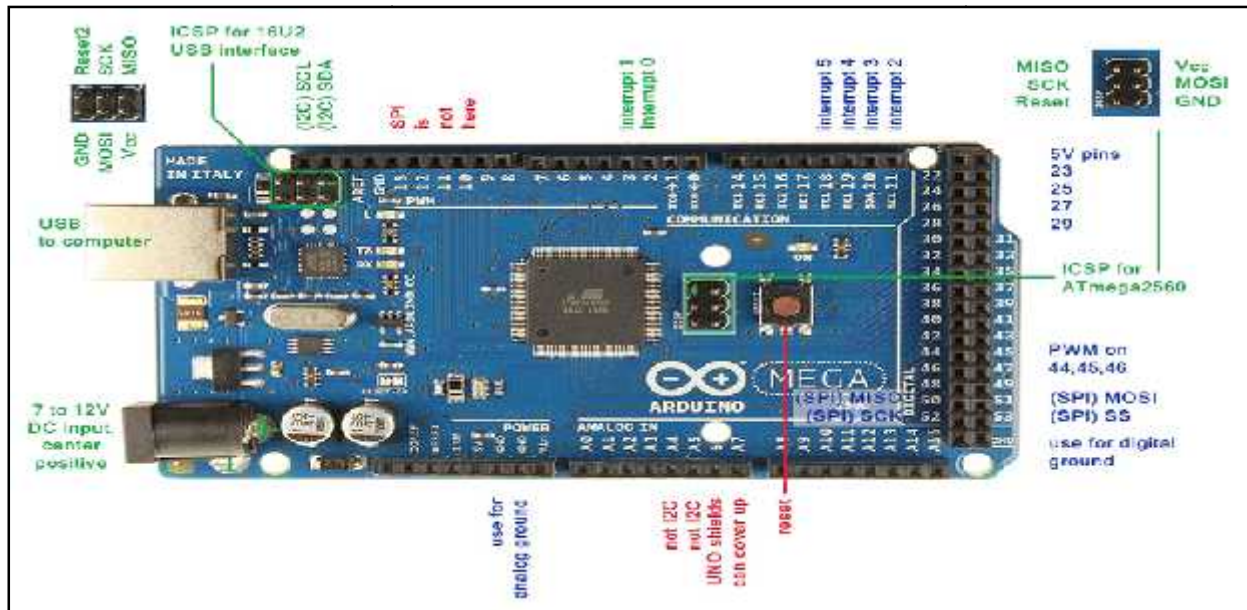


Figure 2.1: Pin diagram of Arduino [12].

2.3.2 Relay

Relays are electrically operated switches. Some relays use an electromagnet to mechanically operate the switch, but other operating principals are also used such as solid-state relays.



Figure 2.2: Typical relays [13]

Figure 2.2 shows typical relays. Relays are used where it is necessary to control a circuit with a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal.

2.3.3 PC Connected to Internet

The local server pc needs internet connections and the Arduino board is connected to local server through the built-in USB port of Arduino.

2.4 Connection Between Arduino and Home Appliances

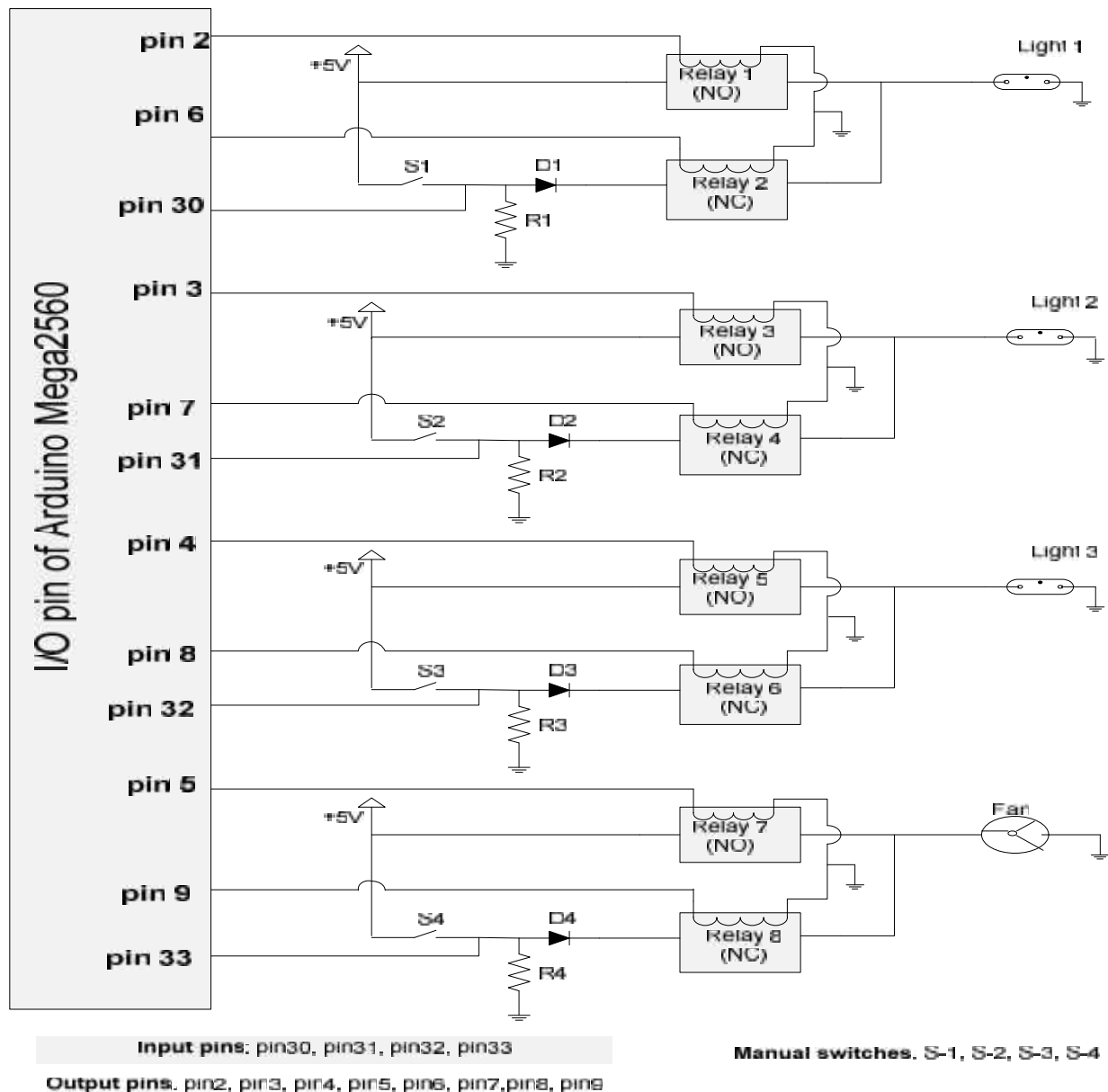


Figure 2.3: Circuit diagram for connecting lights and fan to Arduino through relays

Each of the appliances is connected to a separate digital I/O port of the Arduino board through relays. Figure 2.3 shows the circuit diagram showing eight (8) relays connected to the Arduino board. The digital input pins (i.e. pin nos. 30~33) are used to check the states of the appliances while the digital output pins (i.e. pin nos. 2~9) are used for energizing the relays. The USB port is used to communicate with the local server.

In our prototype we used two relays to execute switching operation for each light and fan. We used two relays to include both manual switching and switching operation from the web.

In these two relays, one is normally open (NO) and the other is normally closed (NC) type. The operating coils of each relay is connected with a digital output pin of Arduino. The voltage source (+5 V DC), manual switch, diode and NC relay is connected in series with appliances (i.e. lights/fans) but the NO relay is connected separately with voltage source and light/fan. If the manual switch remains open then for switching the light and fan we energize the NO relay and switching command is controlled by the user from web. Again, if the manual switch remains closed, the NO relay always remain open and by energizing the NC relay we control the switching operation of the light and fan through web command. By measuring the voltage between the switch and the diode, we can determine the condition of manual switch (i.e. whether on or off) and this voltage (+5 V) is measured by the digital input pin of Arduino. The diode is used to ensure that no voltage can return to the digital input pin when we energize the NO relays. We are also grounding the input pin of Arduino with high resistance (100k). This resistance is known as the pull down resistance and it is used to avoid the noise signal when no source is connected with input pin of Arduino [14].

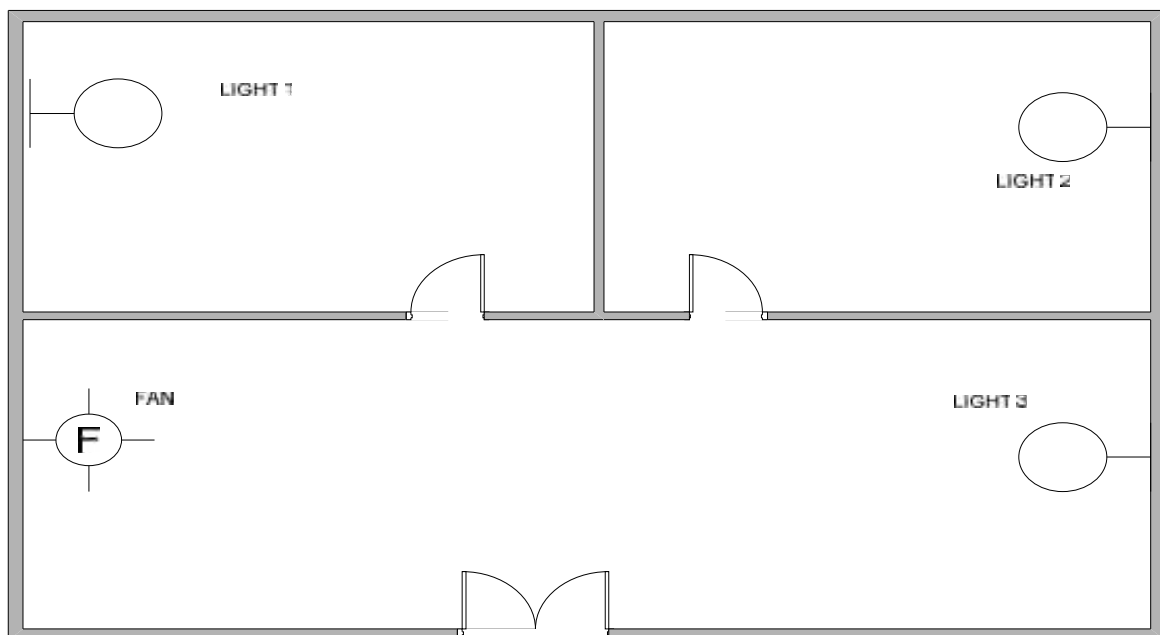


Figure 2.4: Layout of a typical home

2.5 Prototype Construction:

We designed a prototype to demonstrate our project. The prototype contains three lamps and a fan. The layout of the appliances for our prototype is shown in Figure 2.4. The dark grey lines indicate walls of rooms. We use plastic board to build the frame and used LEDs instead

of electric lamps. For the prototype, we used 5V DC relays since we are not using ac lamps in our prototype. In actual applications, however, we will need to drive ac loads, so we would use higher capacity ac relays. Some electric switches are included for manual control of appliances. Figure 2.5 shows a picture of the prototype. There are three lights and a fan in our prototype.

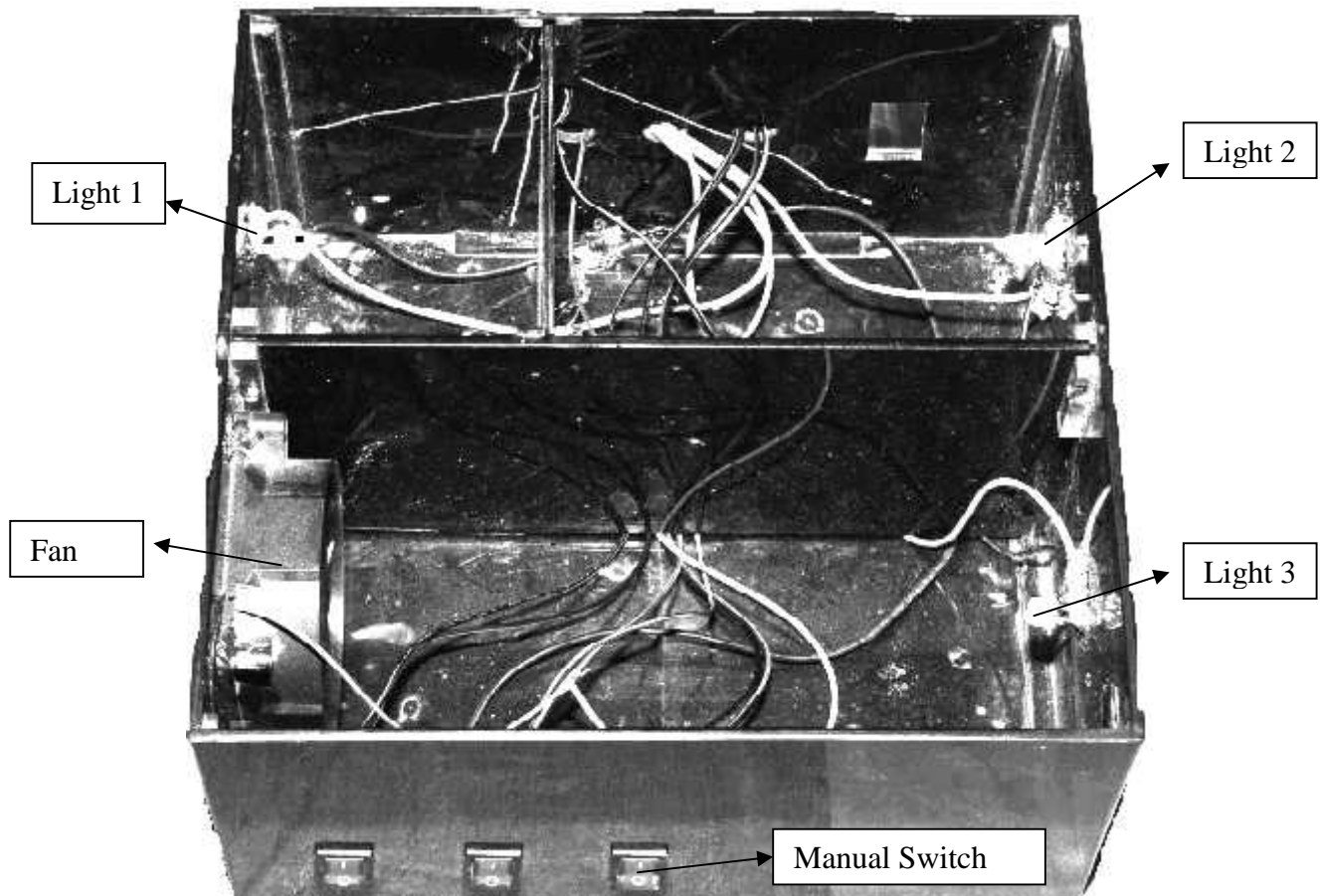


Figure 2.5: Developed prototype

Chapter 3

Software of the System

3.1 Introduction

There are three kinds of program on our system, i.e.

- (i) A program for the Arduino board to monitor and control the appliances. This program is used to learn the status of appliances and to execute switching commands, and also for communicating the operational command through universal serial bus (USB) port of Arduino from the local server.
- (ii) A program to communicate between the local server pc (connected to the Arduino) and the web server. The program also communicates with the Arduino board via the USB port.
- (iii) A program (web application) to communicate between the web server and local server, which receives information from both user and the local server pc. The program is also used for password protecting the access to web server. All the information that are fetched by the server are stored in a database.

3.2 Program for Arduino Board

The program flowchart for the Arduino board is shown in Figure 3.1. In the system, we need to use the digital input pins, digital output pins and the USB serial port of the Arduino Board. The I/O pins are first initiated in the program before the program enters the main loop.

- If a user changes the state of an appliance, a user interruption is requested through the web server. This request is communicated to the USB port via the local server pc and a data string is available in the serial buffer of the local server pc.
- If a string is available in the serial buffer, then the program executes the switching operation as requested by the web server. This is done by making corresponding output pins high or low (to turn the appliance ON or OFF) to activate the relays. The new state of appliances are then sent to the local server via the serial port.
- If there are no interruption from the web server, there will be no data strings in the serial buffer. In that case, the Arduino continuously sends the appliance states to the local server pc.

- Since the program continuously monitors the states of the appliances, if the state is changed when someone turns on/off an appliance, the new state is also detected by the program.

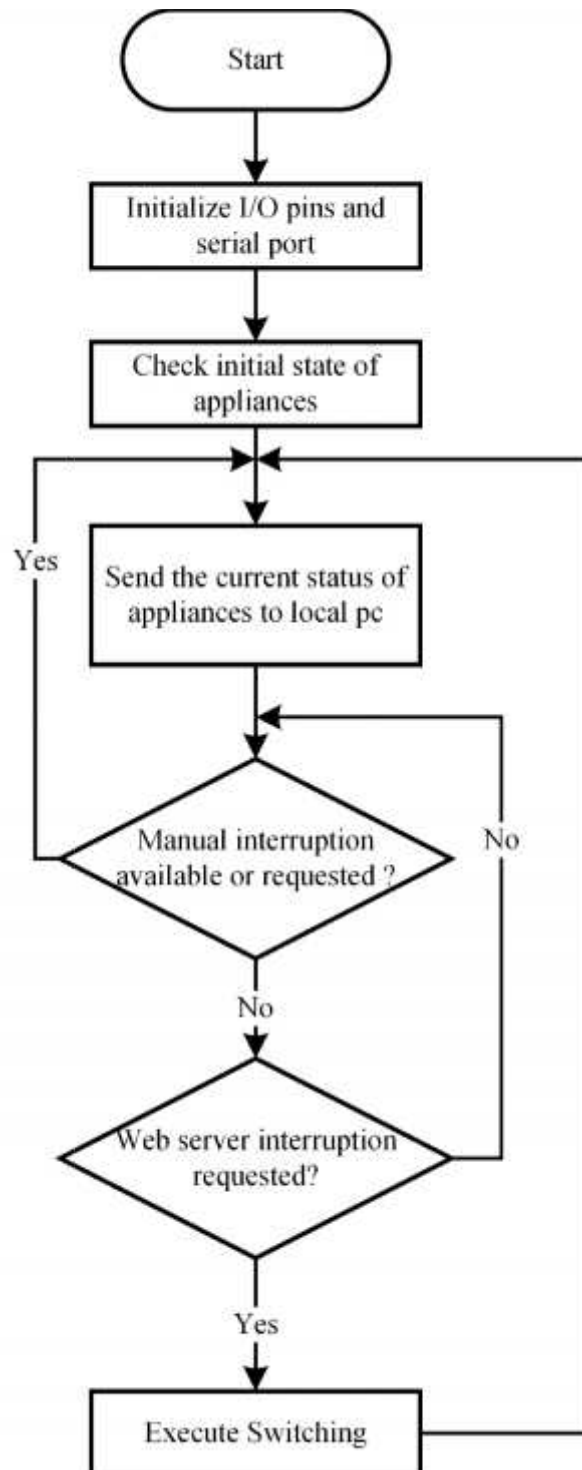


Figure 3.1: Software flow chart for Arduino

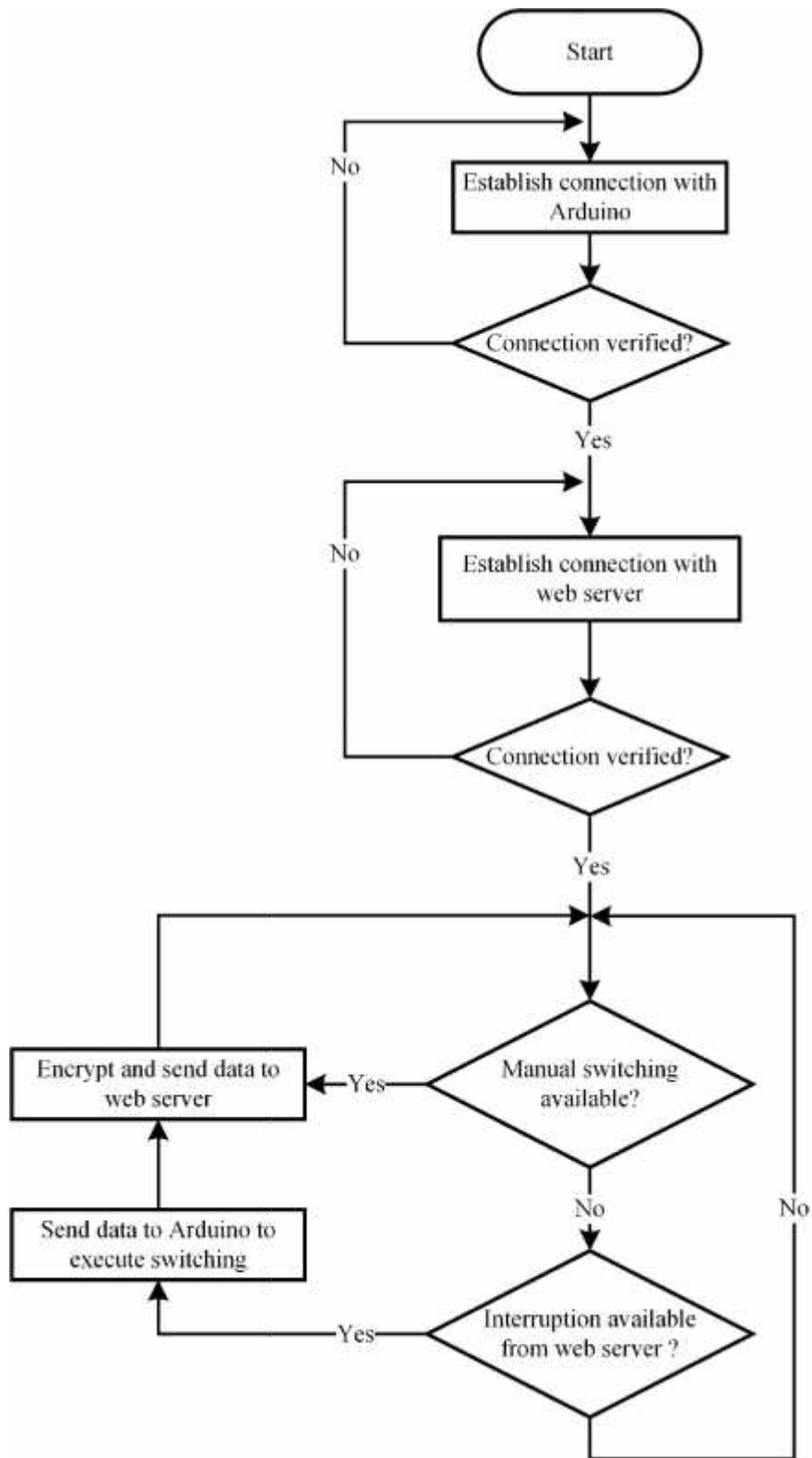


Figure 3.2: Software flow chart for local server

3.3 Program for Local Server

The software of local pc server is developed to communicate back on forth with both Arduino and the web server. Figure 3.2 shows the flow chart of the software. We used a programming language called Python for programming. Python is a programming language developed for quickly prototyping large software [15]. Python is open source software and is distributed freely even for commercial use.

After establishing connections with both the Arduino board and the web server, the program executes an endless loop.

- At the 1st cycle of the loop it gathers the string from Arduino, saves it to a variable and then sends the string to web server to update the database of the server. This task is done to synchronize the appliance state with web server.
- For the rest of the loop it grabs the string from the Arduino and compares both the string. Here, the string represents the user interruption in the web server. If the string gathered from Arduino is different than the one from the previous loop, then it means that the manual switch has been operated. If the manual switch is operated then the software sends command to update the database of the web server.
- If the fetched string from the database of the web server is different from the previous loop, then that means a user interruption is present on the web server. Then program sends the switching command as string to the Arduino.
- The first priority of the program is to detect the operation of manual switches and send them to the web server and the second priority is to detect user interruption on web server.
- The 'PySerial' module is used to communicate with Arduino through the USB port and the 'mechanize' module is used to communicate with web server. Both are open source library module of python.
- The communication between the web server and the program has data encryption protection and password protection to prevent hacking. This password is not visible to the user and also not changeable by the user because of security reasons.

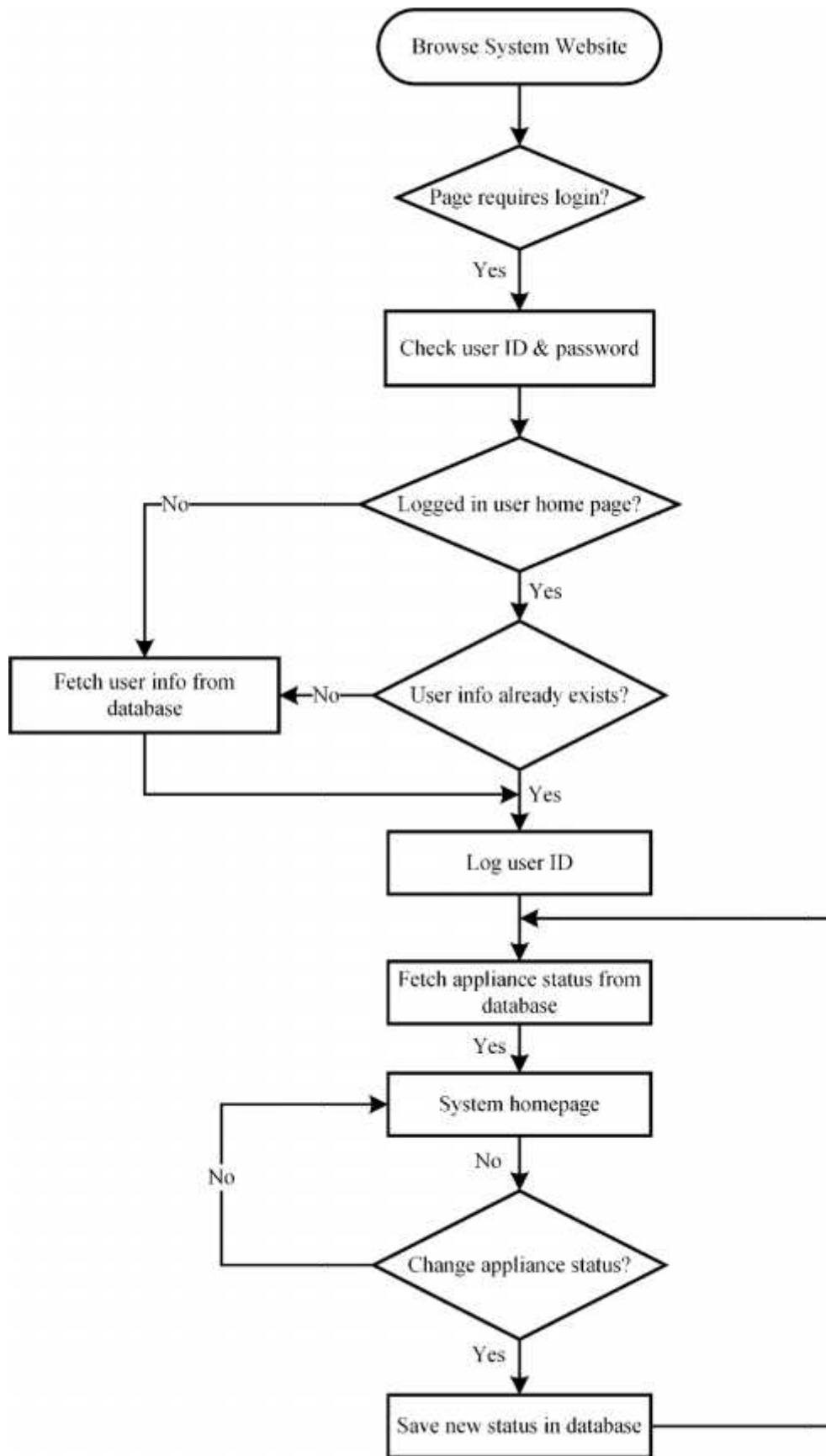


Figure 3.3: Operation of the website

3.4 Web Server Software Details

Figure 3.3 shows the operation of the web server. The software is basically an application that runs on the web server. For building the prototype, we registered for space in a freely available hosting service provider. The web server can be accessed from any computer around the world with internet connection, while the local server can only be accessed from local area networks (LAN). This site is developed with PHP, which is a web scripting language.

Web scripting language:

Web scripting languages are server-side scripting languages that manipulate the data, usually in a database, on the server.

Advantages of PHP:

- It is an open source scripting language.
- It is relatively fast and easy to use and it is very easy to create website scripts in PHP.
- Debugging is easy.
- Functional modules such as PDF, Graph etc. are available in the PHP library.
- Database can be connected easily using PHP, and can be used frequently. This largely reduces the development time of the web apps.
- Can be run on many platforms, including Windows, Linux and Mac.
- It is easy for users to find hosting service providers.

Basic PHP Syntax:

Our web server file hosting is also password protected and this is only accessible by the programmer. We do also have database to save the appliance states and the login username and password of the users. There are two tables in the database, as shown in Figure 3.4.

Table	Action	Records	Type	Collation
onoff		1	MyISAM	latin1_general_c
tbl_users		1	MyISAM	latin1_general_c
2 table(s) Sum		2	MyISAM	latin1_general_ci

Figure 3.4: Internal structure of database [16].

The “tbl_users” is used for storing the user names and passwords and has the architecture shown in Figure 3.5:

id	username	password
1	mksmmi	mcrjnal1

Figure 3.5: Internal structure of user table in database [16].

For testing purposes, our website is hosted on 000webhost, a free webhosting provider, with free domain names. When a user is added by the admin/ programmer, then the rows of the user table are added with the user name and password and if an admin deletes a user row, the corresponding user name will be deleted. The admins/ programmers of the site are the only ones allowed to add users, but the users can change their password by clicking the change password link.

The on/ off table is used for storing the on/off states of the appliances, and also for storing the user interrupts. This table is also accessed by the local server pc to determine any user interrupts from the remote pc and also to update when manual switches operate. Figure 3.6 shows the on/off table structure. This table is also a part of the internal architecture of the web app and is not visible by the users.

id	room1fan1	room1light1	room2fan1	room2light1	room3fan1	room3light1	room4fan1	room4light1
1	1	0	0	0	0	0	0	0

Figure 3.6: Internal structure of appliances status table in database [16].

3.5 User Interface:

If a user wants to see the status of appliances or control the appliances from remote devices, then he/she has to visit the web address of the web server. This website is the user interface which is shown in Figure 3.7.

Login:

For logging in to the server, at first we have to visit' <http://ewuproject.net16.net/>'. Then we have to enter the username and the password. All the information will be stored in a database. If the information is authentic, then the user will be able to access the server. In the server we can check and change the state of the appliances. When we want to leave the page, we need to click on the logout option.

User Home page:

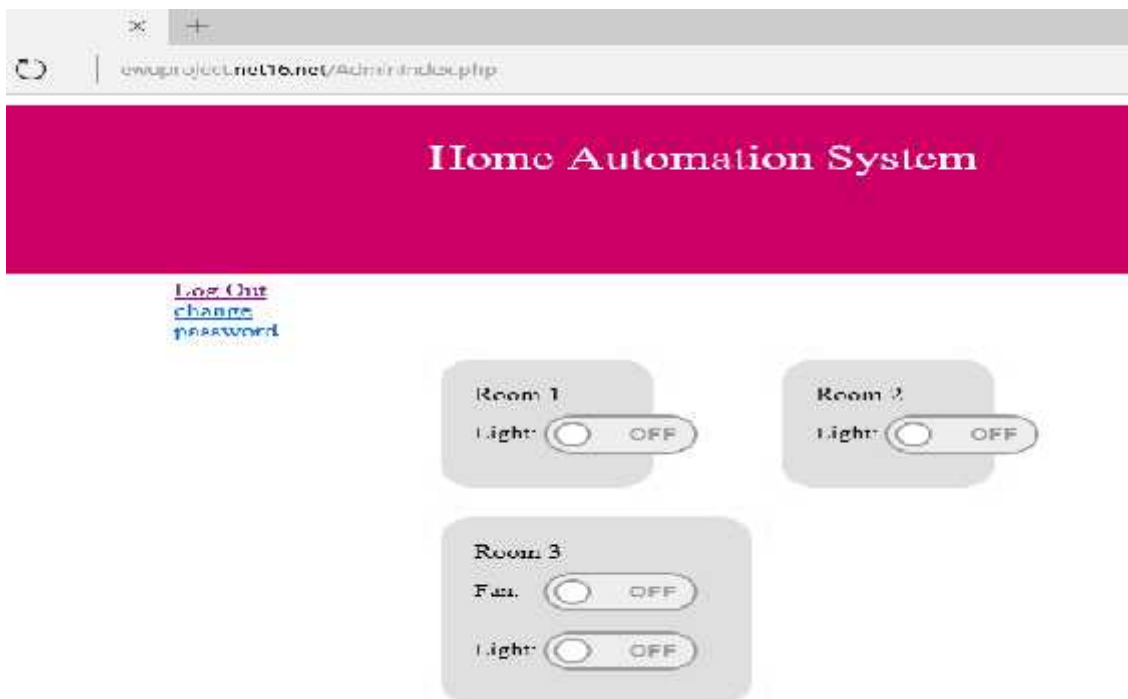


Figure 3.7: User home page of our system [16].

Change password:

In order to change the password, at first we need to login to the server. Then, we will see an option to ensure the old password once and the new password twice. Thus, we can easily change the password and establish a strong security.



Figure 3.8: Password change window [16].

Our system can be accessed through any remote computers (pcs), web servers & database, tablets etc. that has access to the internet. The login information is authenticated and validated from the database through the web server.

3.6 System Communication Diagram.

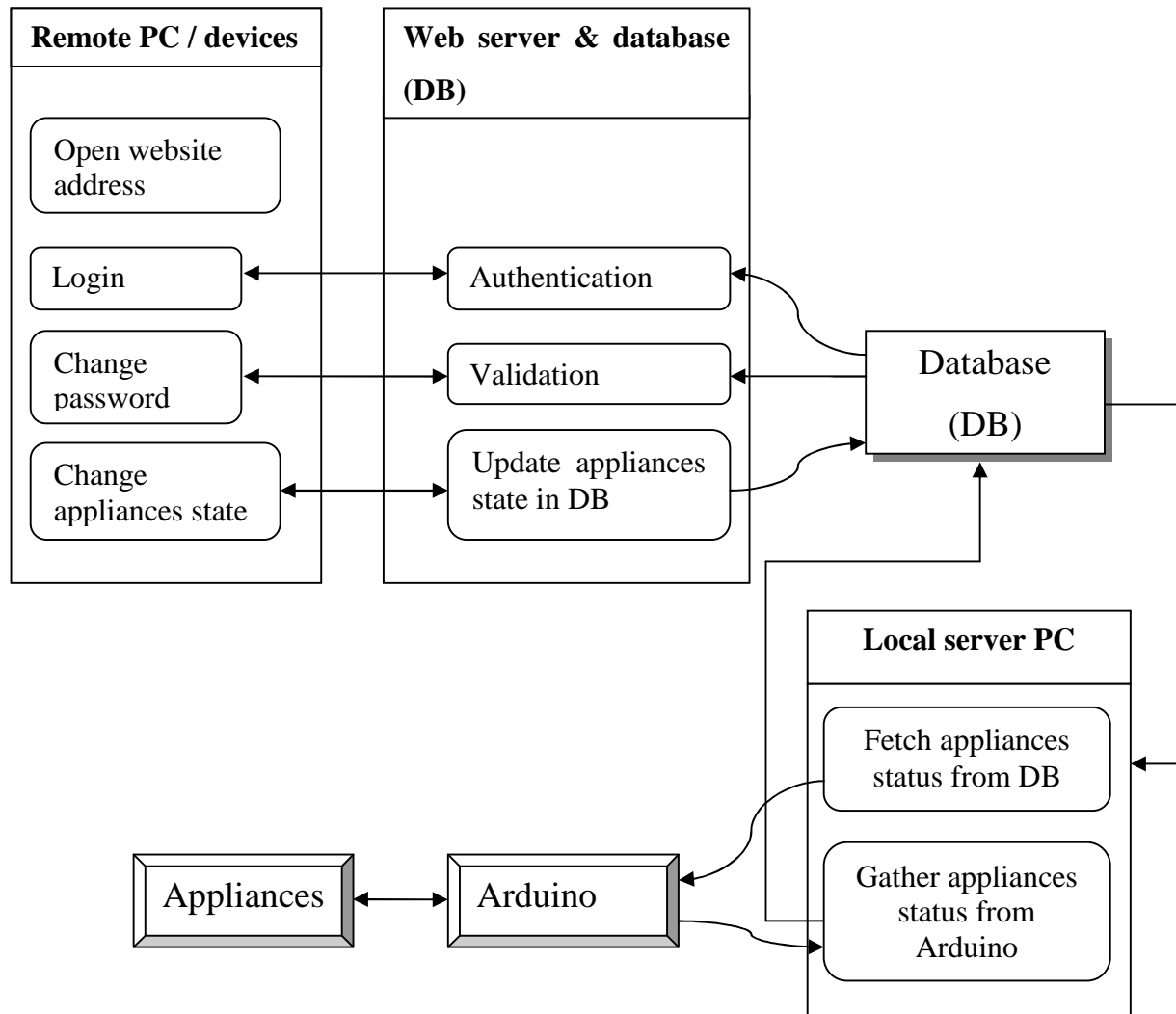


Figure 3.9: Communication flow between components of system

Figure 3.9 shows the communication between the various components of our system. After running the software in the local server, it establishes connection between Arduino and web server. If the execution fails, then it starts the procedure again. If the connection is successful then the software reads the present state of the appliances and sends all the information to the web server and after that, it updates the database. If there is a change in the state of the manual switch, then it updates the server again. But when the manual switch state is

unchanged by the user, then it starts to seek for the command of users who is operating the device from remote server. If a command is received from the user then it executes the command and updates the server again.

3.7 Communication Between Arduino and Web Server

Arduino is connected to the local server via USB cable. The local server software collects and sends data to Arduino via the USB serial buffer. Python has a library function called 'pyserial' [17] to collect and send data to the serial buffer of a computer. We are using this module of python to establish communication with Arduino and pc. We are also using 'python mysql connector' [18] for receiving and sending the data to the web server/ web address from the local server where is Arduino is connected . Figure 3.10 shows a screenshot of the software after successful connection.



Figure 3.10: Successfully connection established.

Chapter 4

Conclusion

We developed a home automation system with remote access through web server. We tested the system through a prototype built by us. This system can be really helpful for physically disabled persons to control home appliances. Furthermore, if anyone forgets to switch off their home appliances before leaving home, they can control and monitor those appliances from anywhere using our system through the web server from remote devices. Through using this system, we can get endless conveniences such as reduce the electric power consumption of our home etc.

Draw backs

A drawback of the system is that since we need to run the local server pc continuously, power failure would disrupt the system. Also, to operate our system, internet access is a necessity.

Future Development

In the future, our plan is to use the 'Raspberry pi' module instead of the local server pc. This will help to minimize the system cost. Furthermore, we plan to enhance our system by adding different types of sensors.

References

1. (2015, January) Current System.[Online].Available:
<http://www.slideshare.net/olafusimichael/500project1>
2. (2015, January) Automation Basic.[Online].Available:
https://en.wikipedia.org/wiki/Home_automation
3. (2015, February) Why Automation.[Online].Available:
<http://www.control4.com/blog/2013/04/7-reasons-why-home-automation-is-made-for-the-technically-challenged>
4. (2015, February) Home Automation Current System.[Online].Available:
<https://www.elprocus.com/gsm-based-home-security-system-working-with-applications/>
5. (2015, March) Home Automation Current System wifi .[Online].Available:
<http://www.mycollegeproject.com/Abstracts/SET-409.%20WiFi%20based%20advanced%20Home.pdf>
6. (2015, March) Current Project Work of Automation.[Online].Available:
https://www.worldwidejournals.com/ijar/file.php?val=July_2014_140422319157.pdf
7. (2015, April) Current project work of Automation.[Online].Available:
<https://www.irjet.net/archives/V2/i3/Irjet-v2i3317.pdf>
8. (2015, July) Bluetooth Based Automation System.[Online].Available:
<http://www.jatit.org/volumes/Vol77No3/12Vol77No3.pdf>
9. (2015, March) Automation Current System.[Online].Available:
<http://pnrsolution.org/Datacenter/Vol3/Issue3/41.pdf>
10. (2015, April) Commercial Review of Home Automation .[Online].Available:
<http://www.vivint.com/packages/home-automation>
11. (2015, May) Commercial Review of Home Automation.[Online].Available:
<http://www.loxone.com/enen/start.html>
12. (2015, June)Arduino Mega Pinouts.[Online].Available:
<http://forum.arduino.cc/index.php?topic=125908.0>

13. (2015, July) Relay Picture.[Online].Available:

<http://www.reuk.co.uk/buy-6V-10A-RELAY.htm>

<http://www.ebay.com/itm/10pcs-SONGLE-Power-Relay-SRD-5VDC-SL-C-Mini-5V-DC-10A-PCB-Type-T73-/251205051922>

14. (2015, July) Pull Down Effect.[Online].Available:

<http://playground.arduino.cc/CommonTopics/PullUpDownResistor>

15. (2015, July) python.[Online].Available:

<http://mcsp.wartburg.edu/zelle/python/python-first.html>

16. (2015, July) Our System Website: [Online].Available

<http://ewuproject.net16.net/>

17. (2015, July) Pc Interfacing.[Online].Available:

<http://pythonhosted.org/pyserial/pyserial.html>

18. (2015, July) Pc Interfacing.[Online].Available:

<https://pypi.python.org/pypi/mysql-connector-python/2.0.4>

19. 2015, June)Arduino.[Online].Available:

<https://www.arduino.cc/en/Guide/Environment>

20. (2015, July) Pc Interfacing.[Online].Available:

<http://www.ladyada.net/learn/arduino/lesson4.html>

21. (2015, July) Python Programming for Arduino.[Online].Available:

<https://books.google.com.bd/books?id=OOPfBgAAQBAJ&pg=PP7&lpg=PP7&dq=py>

Appendix A: Codes

PHP and HTML code for the web site :

Login authentication check :

```
<?php
session_start();
session_name("MY_SESSION");
include("php/configPDO.php");

if(isset($_SESSION['username'])){
header("location: AdminIndex.php");
}
//print_r($_SESSION);
?>
<html>
<head>
<title>PHPRO Login</title>
</head>

<body>
<h2>Login Here</h2>
<form action="CheckLogin.php" method="post">
<fieldset>
<p>
<label for="inputName">Username</label>
<input type="text" id="inputName" name="inputName" value="" maxlength="20" />
</p>
<p>
<label for="inputPassword">Password</label>
<input type="password" id="inputPassword" name="inputPassword" value=""
maxlength="20" />
</p>
<p>
<input type="submit" value="Login" />
</p>
</fieldset>
</form>
</body>
</html>
<?php
```

Checks if the password entered by the user is correct:

```
//session_name("main");

session_start();
session_name("MY_SESSION");
```

```
include("php/configPDO.php");
//Define $myusername and $mypassword
$UserName=$_POST['inputName'];
$Password=$_POST['inputPassword'];
//echo $UserName;
// We Will prepare SQL Query
    $STM = $dbh->prepare("SELECT username,password FROM tbl_users WHERE id= 1");
// bind paramenters, Named paramenters always start with colon(:)
    $STM->bindParam(':UserName', $UserName);
    $STM->bindParam(':Password', $Password);
// For Executing prepared statement we will use below function
    $STM->execute();
// Count no. of records
$count = $STM->rowCount();
//just fetch. only gets one row. So no foreach loop needed :D
$row = $STM -> fetch();
//print_r($row);
//echo $row[0];

//User Redirect Conditions will go here
    if($UserName==$row[0] && $Password==$row[1])

        {

        $_SESSION['username']=$row[0];
        //$_SESSION[Region]=$row[1];
        //$_SESSION[myusername]=$UserName;
        header("Location:AdminIndex.php");
        }
    //else { header( "index.php");
        }
else
    {
    echo "Error Password or Username" ;
    header("index.php");
    }

// Closing MySQL database connection
    //$dbh = null;
//?>
```

Change existing password for web login:

```
<?php
session_start();
session_name("MY_SESSION");

include("php/configPDO.php");

//print_r($_SESSION) ;
if ($_SESSION['username']=="" )
```

```
{
header("Location:index.php");
}

?>
<html>
<head>
<title>change pass</title>
</head>

<body>
<h2>Change password </h2>
<form action="changeeps.php" method="post">
<fieldset>

<p>
<label for="inputPassword1"> Old Password</label>
<input type="password" id="inputPassword1" name="inputPassword1" value=""
maxlength="20" />
</p>
<p>
<label for="inputPassword2">New Password</label>
<input type="password" id="inputPassword2" name="inputPassword2" value=""
maxlength="20" />
</p>
<p>
<label for="inputPassword3">Confirm New Password</label>
<input type="password" id="inputPassword3" name="inputPassword3" value=""
maxlength="20" />
</p>
<p>

<input type="submit" value="Login" />
</p>
</fieldset>
</form>
</body>
</html>
```

Code for the Homepage after login the site :

```
<?php

session_start();
print_r($_SESSION) ;
if ($_SESSION['username']== "")
{
header("Location:index.php");
}
$page = $_SERVER['PHP_SELF'];
$sec = "10";
```



```
        include("php/dbconnn.php");
        $conn = dbConnect();
$stmt = $conn->prepare('SELECT * FROM onoff WHERE ID = 0');
$stmt->execute();
$row = $stmt->fetch();
$r1f1=$row ["1"];
$r1l1=$row ["2"];
$r2f1=$row ["3"];
$r2l1=$row ["4"];
$r3f1=$row ["5"];
$r3l1=$row ["6"];
$r4f1=$row ["7"];
$r4l1=$row ["8"];
if ($r1f1==1)
    {
        $r1f1= "checked";
    }

else
    {
        $r1f1="";
    }
if ($r1l1==1)
    {
        $r1l1= "checked";
    }
else
    {
        $r1l1="";
    }

if ($r2f1==1)
    {
        $r2f1= "checked";
    }
else
    {
        $r2f1="";
    }
if ($r2l1==1)
    {
        $r2l1= "checked";
    }
else
    {
        $r2l1="";
    }
if ($r3f1==1)
    {
        $r3f1= "checked";
```

```
}
else
{
    $r3f1="";
}
if ($r3l1==1)
{
    $r3l1= "checked";
}

else
{
    $r3l1="";
}if ($r4f1==1)
{
    $r4f1= "checked";
}
else
{
    $r4f1="";
}
if ($r4l1==1)
{
    $r4l1= "checked";
}
else
{
    $r4l1="";
}
?>
<!DOCTYPE html>

<html lang="en">
<head>
<meta http-equiv="refresh" content="<?php echo $sec?>;URL='<?php echo $page?>'">
    <meta charset="UTF-8" />
<title>On/OFF</title>
    <link rel=stylesheet href="css/style.css">
    <link rel=stylesheet href="css/button-room1.css">
    <link rel=stylesheet href="css/button-room2.css">
    <link rel=stylesheet href="css/button-room3.css">
    <link rel=stylesheet href="css/button-room4.css">
    </head>
<body>

<div id="container">
<div class="header"><p class="text-head">>Home Automation System</p></div>
<div class="mainbody">
<div id="menu"></div>
    <div id="control">
```

```

<form action="php/ps.php" method="post">
<div id="logout">
<a href="logout.php">Log Out</a>
</div>
<div id="content1">
<p id="d1">Room 1</p>
    <label id="text1">Fan:</label>
<div class="room1fan1">
<input type="checkbox" name="room1fan1" value="1" class="room1fan1-checkbox"
id="myroom1fan1" onClick="javascript: submit()" <?php echo $r1f1; ?>>
<label class="room1fan1-label" for="myroom1fan1">
<span class="room1fan1-inner"></span>
<span class="room1fan1-switch"></span>
</label>
</div>

<label id="text1">Light:</label>
<div class="room1light1">
<input type="checkbox" name="room1light1" value="1" class="room1light1-checkbox"
id="myroom1light1" onClick="javascript: submit()" <?php echo $r1l1; ?>>
<label class="room1light1-label" for="myroom1light1">
<span class="room1light1-inner"></span>
<span class="room1light1-switch"></span>
</label>
</div>

    </div>
<div id="content2">
<p id="d1">Room 2</p>
    <label id="text1">Fan:</label>
<div class="room2fan1">
<input type="checkbox" name="room2fan1" value="1" class="room2fan1-checkbox"
id="myroom2fan1" onclick="javascript: submit()" <?php echo $r2f1; ?>>
<label class="room2fan1-label" for="myroom2fan1">
<span class="room2fan1-inner"></span>
<span class="room2fan1-switch"></span>
</label>
</div>

<label id="text1">Light:</label>
<div class="room2light1">
<input type="checkbox" name="room2light1" value="1" class="room2light1-checkbox"
id="myroom2light1" onClick="javascript: submit()" <?php echo $r2l1; ?>>
<label class="room2light1-label" for="myroom2light1">
<span class="room2light1-inner"></span>
<span class="room2light1-switch"></span>
</label>
</div>
</div>

```

```

<div id="content3">
<p id="d1">Room 3</p>
    <label id="text1">Fan:</label>
<div class="room3fan1">
<input type="checkbox" name="room3fan1" value="1" class="room3fan1-checkbox"
id="myroom3fan1" onClick="javascript: submit()" <?php echo $r3f1; ?>>
<label class="room3fan1-label" for="myroom3fan1">
<span class="room3fan1-inner"></span>
<span class="room3fan1-switch"></span>
</label>
</div>

<label id="text1">Light:</label>
<div class="room3light1">
<input type="checkbox" name="room3light1" value="1" class="room3light1-checkbox"
id="myroom3light1" onClick="javascript: submit()" <?php echo $r3l1; ?>>
<label class="room3light1-label" for="myroom3light1">
<span class="room3light1-inner"></span>
<span class="room3light1-switch"></span>
</label>
</div>
    </div>
    <div id="content4">
<p id="d1">Room 4</p>
    <label id="text1">Fan:</label>
<div class="room4fan1">
<input type="checkbox" name="room4fan1" value="1" class="room4fan1-checkbox"
id="myroom4fan1" onclick="javascript: submit()" <?php echo $r4f1; ?>>
<label class="room4fan1-label" for="myroom4fan1">
<span class="room4fan1-inner"></span>
<span class="room4fan1-switch"></span>
</label>
</div>

<label id="text1">Light:</label>
<div class="room4light1">
<input type="checkbox" name="room4light1" value="1" class="room4light1-checkbox"
id="myroom4light1" onClick="javascript: submit()" <?php echo $r4l1; ?>>
<label class="room4light1-label" for="myroom4light1">
<span class="room4light1-inner"></span>
<span class="room4light1-switch"></span>
</label>
</div>
</div>
</form>
</div>
</div>
</div>
</body>
</html>

```

Create connection with database:

```
<?php
function dbConnect (){
    $conn =null;
    $host = 'localhost';
    $db = 'meraj';
    $user = 'root';
    $pwd = 'merajnai1';
    try {
        $conn = new PDO('mysql:host='.$host.'.dbname='.$db, $user, $pwd);
        //echo 'Connected succesfully.<br>';
    }
    catch (PDOException $e) {
        echo '<p>Cannot connect to database !!</p>';
        echo '<p>'.$e.'</p>';
        exit;
    }
}
return $conn;
}
?>
```

Saving switching operation in database by user:

```
<?php
include ("dbconnn.php");
    $conn = dbConnect();
//1. room1fan1
    {
if (isset($_POST['room1fan1'])) {
    $name = $_POST['room1fan1'];

$checkbox = "UPDATE onoff SET room1fan1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
        $error = $stmt->errorInfo();
    //echo $error[2];
    }
else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room1fan1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
    $error = $stmt->errorInfo();
    //echo $error[2];
}
}
}
```

```

//2. room1light1
    {
if (isset($_POST['room1light1'])) {

    $name = $_POST['room1light1'];

$checkbox = "UPDATE onoff SET room1light1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
        $error = $stmt->errorInfo();
    //echo $error[2];
}
else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room1light1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
        $error = $stmt->errorInfo();
    //echo $error[2];
}
}
// 3.room2fan1
    {
if (isset($_POST['room2fan1'])) {

    $name = $_POST['room2fan1'];

$checkbox = "UPDATE onoff SET room2fan1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
        $error = $stmt->errorInfo();
    //echo $error[2];

}

else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room2fan1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
        $error = $stmt->errorInfo();
}
}

```

```

//echo $error[2];
}
}
//4. room2light1
{
if (isset($_POST['room2light1'])) {

$name = $_POST['room2light1'];
$checkbox = "UPDATE onoff SET room2light1=($name) WHERE id=0 ";
$stmt = $conn->prepare($checkbox);
// Now we execute the query passing an array toe execute();
$results = $stmt->execute(array($name));
// Extract the values from $result
$error = $stmt->errorInfo();
//echo $error[2];
}
else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room2light1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
    $error = $stmt->errorInfo();
    //echo $error[2];
}
}
// 5.room3fan1
{
if (isset($_POST['room3fan1'])) {
    $name = $_POST['room3fan1'];

$checkbox = "UPDATE onoff SET room3fan1=($name) WHERE id=0 ";
$stmt = $conn->prepare($checkbox);
// Now we execute the query passing an array toe execute();
$results = $stmt->execute(array($name));
// Extract the values from $result
$error = $stmt->errorInfo();
//echo $error[2];
}
else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room3fan1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array toe execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
    $error = $stmt->errorInfo();
    //echo $error[2];
}
}
}

```

```

}
//6. room3light1
{
if (isset($_POST['room3light1'])) {

    $name = $_POST['room3light1'];

    $checkbox = "UPDATE onoff SET room3light1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array to execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
    $error = $stmt->errorInfo();
    //echo $error[2];
}
else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room3light1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array to execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
    $error = $stmt->errorInfo();
    //echo $error[2];
}
}
//7. room4fan1
{
if (isset($_POST['room4fan1'])) {

    $name = $_POST['room4fan1'];

    $checkbox = "UPDATE onoff SET room4fan1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array to execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
    $error = $stmt->errorInfo();
    //echo $error[2];

}

else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room4fan1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array to execute();
    $results = $stmt->execute(array($name));
}
}

```



```

        // Extract the values from $result
            $error = $stmt->errorInfo();
        //echo $error[2];
    }
}

//7. room4light1
{
if (isset($_POST['room4light1'])) {

$name = $_POST['room4light1'];
$checkbox = "UPDATE onoff SET room4light1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array to execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
        $error = $stmt->errorInfo();
    //echo $error[2];

}
else {
    $name = '0';
    $checkbox = "UPDATE onoff SET room4light1=($name) WHERE id=0 ";
    $stmt = $conn->prepare($checkbox);
    // Now we execute the query passing an array to execute();
    $results = $stmt->execute(array($name));
    // Extract the values from $result
        $error = $stmt->errorInfo();
    //echo $error[2];
}
}

?>

```

Creating database:

```

CREATE TABLE IF NOT EXISTS `tbl_users` (
  `id` INT(20) NOT NULL,
  `username` INT(20) NOT NULL,
  `password` INT(20) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `onoff` (
  `id` INT(20) NOT NULL,
  `room1fan1` INT(20) NOT NULL,
  `room1light1` INT(20) NOT NULL,
  `room2fan1` INT(20) NOT NULL,
  `room2light1` INT(20) NOT NULL,
  `room3fan1` INT(20) NOT NULL,
  `room3light1` INT(20) NOT NULL,

```

```
`room4fan1` INT(20) NOT NULL,  
`room4light1` INT(20) NOT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

Arduino controlling relay and Appliance state read codes :

Arduino Code :

```
int Loopcount =0;  
String inputString = "";    // a string to hold incoming data  
boolean stringComplete = false; // whether the string is complete  
//defining input pin  
int inPin30 = 30;  
int inPin31 = 31;  
int inPin32 = 32;  
int inPin33 = 33;  
//defining output pin  
int outPin2 = 2;  
int outPin3 = 3;  
int outPin4 = 4;  
int outPin5 = 5;  
int outPin6 = 6;  
int outPin7 = 7;  
int outPin8 = 8;  
int outPin9 = 9;  
// global variable set for the memory input pin  
int in00 =0;  
int in11 =0;  
int in22 =0;  
int in33 =0;  
void setup() {  
    digitalWrite(outPin2, LOW);  
    digitalWrite(outPin3, LOW);  
    digitalWrite(outPin4, LOW);  
    digitalWrite(outPin5, LOW);  
    digitalWrite(outPin6, LOW);  
    digitalWrite(outPin7, LOW);
```

```
    digitalWrite(outPin8, LOW);
    digitalWrite(outPin9, LOW);
// initialize serial:
Serial.begin(9600);
// reserve 200 bytes for the inputString:
inputString.reserve(200);
}
void loop() {
    int in0= digitalRead(inPin30);
    int in1= digitalRead(inPin31);
    int in2= digitalRead(inPin32);
    int in3= digitalRead(inPin33);

    String str=String(in0)+String(in1)+String(in2)+String(in3)+'\n';
    String str2=String(in00)+String(in11)+String(in22)+String(in33)+'\n';
if (Loopcount =0){
    Serial.println(str);
    }
    delay(1000);
if (Loopcount !=0 && str!=str2){

    Serial.println(str);
    }
Loopcount=1;
// print the string when a newline arrives:
if (stringComplete) {
    appliance_st(inputString);
    // clear the string:
    inputString = "";
    stringComplete = false;
}
delay(1000);
}
/*
```

SerialEvent occurs whenever a new data comes in the

hardware serial RX. This routine is run between each time loop() runs, so using delay inside loop can delay response. Multiple bytes of data may be available.

```

*/
void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    inputString += inChar;
    // if the incoming character is a newline, set a flag
    // so the main loop can do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}

String appliance_st ( String content) {
  //Appliance 1
  // for turn on the appliance
  if (content[0]=='1' && digitalRead(30) == HIGH )
  {
    digitalWrite(outPin2, LOW);
  }
  if (content[0]=='1' && digitalRead(30) == LOW )
  {
    digitalWrite(outPin6, HIGH);
  }
  //for turn off the appliance
  if (content[0]=='0' && digitalRead(30) == HIGH )
  {
    digitalWrite(outPin2, HIGH);
    digitalWrite(outPin6, LOW);
  }
  if (content[0]=='0' && digitalRead(30) == LOW )

```

```
{
digitalWrite(outPin6, LOW);
}
//Appliance 2
// for turn on the appliance
if (content[1]=='1' && digitalRead(31) == HIGH )
{
digitalWrite(outPin3, LOW);
}
if (content[1]=='1' && digitalRead(31) == LOW )
{
digitalWrite(outPin7, HIGH);
}
//for turn off the appliance
if (content[1]=='0' && digitalRead(31) == HIGH )
{
digitalWrite(outPin3, HIGH);
digitalWrite(outPin7, LOW);
}
if (content[1]=='0' && digitalRead(31) == LOW )
{
digitalWrite(outPin7, LOW);
}
//Appliance 3
// for turn on the appliance
if (content[2]=='1' && digitalRead(32) == HIGH )
{
digitalWrite(outPin4, LOW);
}
if (content[2]=='1' && digitalRead(32) == LOW )
{
digitalWrite(outPin8, HIGH);
}
//for turn off the appliance
if (content[2]=='0' && digitalRead(32) == HIGH )
```

```
{
digitalWrite(outPin4, HIGH);
digitalWrite(outPin8, LOW);
}
if (content[2]=='0' && digitalRead(32) == LOW )
{
digitalWrite(outPin8, LOW);
}
//Appliance 4
// for turn on the appliance
if (content[3]=='1' && digitalRead(33) == HIGH )
{
digitalWrite(outPin5, LOW);
}
if (content[3]=='1' && digitalRead(33) == LOW )
{
digitalWrite(outPin9, HIGH);
}
//for turn off the appliance
if (content[3]=='0' && digitalRead(33) == HIGH )
{
digitalWrite(outPin5, HIGH);
digitalWrite(outPin9, LOW);
}
if (content[3]=='0' && digitalRead(33) == LOW )
{
digitalWrite(outPin9, LOW);
}
//switching done and idle the switching command
}
```

Pc Communicating with Arduino and Server :

```
#library import
import serial, time
import urllib
import sys
```

```
import mysql.connector
import mechanize
import cookielib
from BeautifulSoup import BeautifulSoup
import html2text
from timeout import timeout
def try_connect_Arduino():
    arduino = serial.Serial('COM3', 9600, timeout=.1)
    arduino.setDTR(level=False)
    time.sleep(0.5)
    arduino.flushInput()
    arduino.setDTR()
    time.sleep(0.5)
def try_connect_db():
    cnx = mysql.connector.connect(user='mksmmi', password='merajnai1',
    host='http://ewuproject.net16.net/',
    database='a8142278_meraj')
connected_Arduino = False
while not connected_database:
    try:
        try_connect_Arduino()
        if True:
            connected_Arduino = True
            print " Successfully connected with Arduino at COM3"
    except:
        pass
connected_database = False
while not connected_database:
    try:
        try_connect_db()
        if True:
            connected_database = True
            print " Successfully connected to Web"
    except:
        pass
```

```
data = arduino.readline()
br.open('http://ewuproject.net16.net/php/submitbin.php')
br.select_form(nr=0)
br.form['newcon'] = data
br.submit()
time.sleep(.5)
x= data
while True:
    sock = urllib.urlopen('http://ewuproject.net16.net/php/binout.php')
    htmlSource = sock.read()
    z = str(htmlSource)
    v=z[0]+z[4]+z[8]+z[10]
    sock.close()
    try :
        data = arduino.readline()
        br.open('http://ewuproject.net16.net/php/submitbin.php')
        br.select_form(nr=0)
        br.form['newcon'] = data
        br.submit()
        time.sleep(.5)
    else:
        if v !=x:
            ser.write(v)
            time.sleep(1)
```


Appendix B: Screenshots of software

_ctypes.pyd	5/23/2015 9:40 AM	PYD File	86 KB
_hashlib.pyd	5/23/2015 9:41 AM	PYD File	889 KB
_socket.pyd	5/23/2015 9:40 AM	PYD File	45 KB
_sqlite3.pyd	5/23/2015 9:40 AM	PYD File	47 KB
_ssl.pyd	5/23/2015 9:40 AM	PYD File	1,285 KB
bz2.pyd	5/23/2015 9:40 AM	PYD File	67 KB
CRYPT32.dll	7/10/2015 3:00 PM	Application extens...	1,488 KB
driver.exe	11/30/2015 9:06 AM	Application	23 KB
library.zip	11/30/2015 9:06 AM	WinRAR ZIP archive	2,915 KB
pyexpat.pyd	5/23/2015 9:40 AM	PYD File	125 KB
python27.dll	5/23/2015 9:40 AM	Application extens...	2,402 KB
select.pyd	5/23/2015 9:41 AM	PYD File	10 KB
sqlite3.dll	5/23/2015 9:39 AM	Application extens...	417 KB
unicodedata.pyd	5/23/2015 9:40 AM	PYD File	670 KB
wfpopen.exe	5/23/2015 9:39 AM	Application	49 KB

Figure B.1: Window of successful run of exe file.

file-manager.000webhost.com/file-manager/index.php

100% Script finished in 0.00 seconds

ewuproject.net16.net

Language: English

All	Name	Type	Size	Owner	Group	Perms	Mod time	Actions
	..	Directory	1096	u8140278	u8140278	rw-rw-rw-	Nov 29 09:19	
	...	Directory	1096	u8140278	u8140278	rw-rw-rw-	Nov 29 09:19	
	..	Directory	1096	u8140278	u8140278	rw-rw-rw-	Nov 29 15:00	
	..	Directory	1096	u8140278	u8140278	rw-rw-rw-	Nov 29 09:07	
	admin.php	HTML text	91	u8140278	u8140278	rw-rw-rw-	Nov 29 09:07	View Edit Open
	adminindex.php	HTML text	448	u8140278	u8140278	rw-rw-rw-	Nov 29 09:05	View Edit Open
	checklogin.php	PHP script	170	u8140278	u8140278	rw-rw-rw-	Nov 29 09:17	View Edit Open
	login.php	PHP script	67	u8140278	u8140278	rw-rw-rw-	Nov 29 09:07	View Edit Open
	logout.php	HTML text	51	u8140278	u8140278	rw-rw-rw-	Nov 29 09:15	View Edit Open
	logout.php	HTML text	507	u8140278	u8140278	rw-rw-rw-	Nov 29 09:19	View Edit Open
	index.php	HTML text	667	u8140278	u8140278	rw-rw-rw-	Nov 29 09:10	View Edit Open

Directory total: 4
Files: 7 / 12.35 kB
Sym links: 0

Figure B.2: Web server directory page.