Master’s Thesis 2015

Pratik Gadtaula

Home Automation

Telemark University College
Faculty of Technology

Telemark University College
Faculty of Technology
Kjølnes ring 56
3918 Porsgrunn
Version 0.9

http://www.hit.no

Abstract

The thesis is on HomeAutomation. It covers the area of monitoring and controlling appliances in home as per users configuration and control. As the automation is performed on Raspberry Pi device along with Arduino board, it combines the overall benefits from both devices and thus useful in implementing our tasks.

It primarily focus on safety and then other facilities extended along with it. Services like knowing temperature reading, lights On/Off condition, fan On/Off and other services are featured in this HomeAutomation. The Alarm system is also major part in HomeAutomation which secure the home and update user with right information in right time to avoid accident and loss.

The controlling section is great importance in HomeAutomation. User will have automatic settings to control the appliances. Further, this service is good and one of the reliable way to encapsulate home from internal and external danger. People in job or outside home can work freely and smartly having control to their home. They can just sit and login browser and see what is going on in their home in just a second and feel that their home is with them all time.

HomeAutomation is truly one of the needs in today's world. People rely and feel safe and warmth in their home with their family. HomeAutomation brings more closer and more safer to them.

Abbreviation

O.S	Operating System
DBMS	Database Management System
PC	Personal Computer
PID	Proportional-Integral-Derivative
DB	Data Base
LED	Light Emitting Diode

Tables of Contents

Tables of Figures

Preface

The Home Automation Thesis is final year Master Thesis for System and Control Engineering. The final year thesis is mandatory for all students in System and Control Master programme at the final semester.

The thesis is carried out under supervision of Mr. Hans-Petter Halvorsen lecturer , lecturer in Faculty of Technology, Telemark University College. He has been very helpful in this thesis and provided , a guidance and feedback throughout the thesis period. I like to thank Hans also for providing sensors and control devices along with number of Raspberry Pi and Arduino devices for ease access to work in thesis. Also, i would like to thank my friend Mr. Kishan Prajapati for helping programming part in thesis.

I like to thank supervisor for giving an opportunity to learn new things during thesis to know real data monitoring system with Arduino device and Database system in Raspberry Pi with some PHP based programming. It was worth more and hope to do find more in this topic in future days.

Porsgrunn, June 3, 2015

Pratik Gadtaula

1 Introduction

The Master Thesis is on Home Automation. HomeAutomation is today's one of the growing needs in people's life. HomeAutomation actually refers to automation of home appliances which can be viewed from safety, automatic monitoring and controlling view.

1.1 Project Task Description

Information is everything in today's real life. Particularly when these information are prone to affect their life in close way, information about such things are of great importance. People are away from home for work or for other reasons and there is always insecure of what's happening in home and things related to home appliances whether they are operating or turned On/Off and several other things associated to their house.

HomeAutomation is sought to solve these problems and provide user with the instant monitoring and controlling of home appliances. There are several devices for such monitoring and controlling. In this Master Thesis, we are focused on monitoring and controlling the home appliances with Raspberry Pi in extension with Arduino. As these devices are cheaper and simpler to use and handle, they are used in co-operation to widen the work and service functionality and comparatively.

The Thesis is focused on building HomeAutomation which supports basic monitoring and controlling of lights, Fan and Raspberry Pi Camera. With these tasks to accomplish, the internal understanding of Database(SQL and MySQL), Programming Languages(PHP), Web-Server are of equal importance to deal with as these are the internal core on which HomeAutomation is build upon.

HomeAutomation provides a feel of secure when in/out of house. The system appliances run automatic. For example: The lights are turned On and Off when there is evening and morning. Also when away from home these things can be monitored in webpage or application, which allows to know what are the condition in home, whether light is On or whether fan is Off or door is locked. And through automatic controlling action we can control the light to On/Off, fan to On/Off and be in safe zone

1.2 Goals of the Project

The goals are to complete the tasks and in systematic order finish the given task. The main goal of the project are listed below:

- Log, Monitor and Control appliances in Homes.
- Develop PID control for temperature control in Homes.
- Develop central Server 'SQL Server'.
- Develop 'Dashboard for Lab VIEW' application for control and monitoring

These are the important goals of the project. The Logging, Monitoring and Controlling of sensors and appliances are done .Temperature are monitored, Lights and Fans are controlled and logged accordingly. Camera along with Alarm are activated.

Similarly, a PID controller is developed and used for controlling room heater in house. Depending on temperature sensor data recorded, the input is tuned for constant output temperature specified at reference point.

These tasks were of interesting , however were time consuming due to beginners in Raspberry Pi device and programming language. Also devices were not working according to guidance from reference book. Troubleshooting of such problems took more time than expected. It was time consuming to find solutions , thus restricting to go further tasks. Furthermore webpage design and coding is new and took lot of time in understanding and synchronizing with MySQL Database server.

Dashboard applications tasks and central server tasks i.e SQL server system tasks were incomplete in the thesis period. Also, Central Server tasks needed more time to create, it was unfortunately undone. And dashboard application tasks require Central server tasks to be done. Without it, dashboard application was useless so it was too undone.

Although the tasks were exciting and was interesting to work with Raspberry device and Arduino along with sensors and devices.

2 Theory

2.1 Home Automation

Home Automation is automation of home, housework or household activity. In other words it refers to use of IT/computer to control home appliances. It integrates electrical devices in a house with each other. For example: It can include centralized control of lighting, appliances, security lock of gates & doors to provide improved convenience, comfort, energy, efficiency and safety.

In today's IT world, home automation is being popular due to easiness, flexible means of viewing/monitoring and controlling the appliances and other things according to users comfort and needs. The challenging part lies in simplicity and cost of installing them in home and varies with increasing number of services to be monitored and controlled.

2.2 Raspberry Pi

Raspberry Pi is credit-card sized single board computer used for programming and designing. It is operating on Linux-Kernel based Operating System. It function similar to computer except that it do not have screen monitor, mouse and keyboard. It costs around € 45.00. We can do most of work in Raspberry Pi like we can browse internet or play video. The only exception is to program so that it can function and play the video or surf internet.

It consists of CPU, GPU and RAM like that of Computer. The physical outlets consists of Power plug , Ethernet plug, HDMI, USB, Card Reader, Audio Out, RCA Video Out, GPIO respectively.[1]

The frontal and back part of Raspberry Pi Circuit board is shown in Figure 2-1 and Figure 2-2 along with the major parts indication.

Figure 2-1: Front part of Circuit board of Raspberry Pi

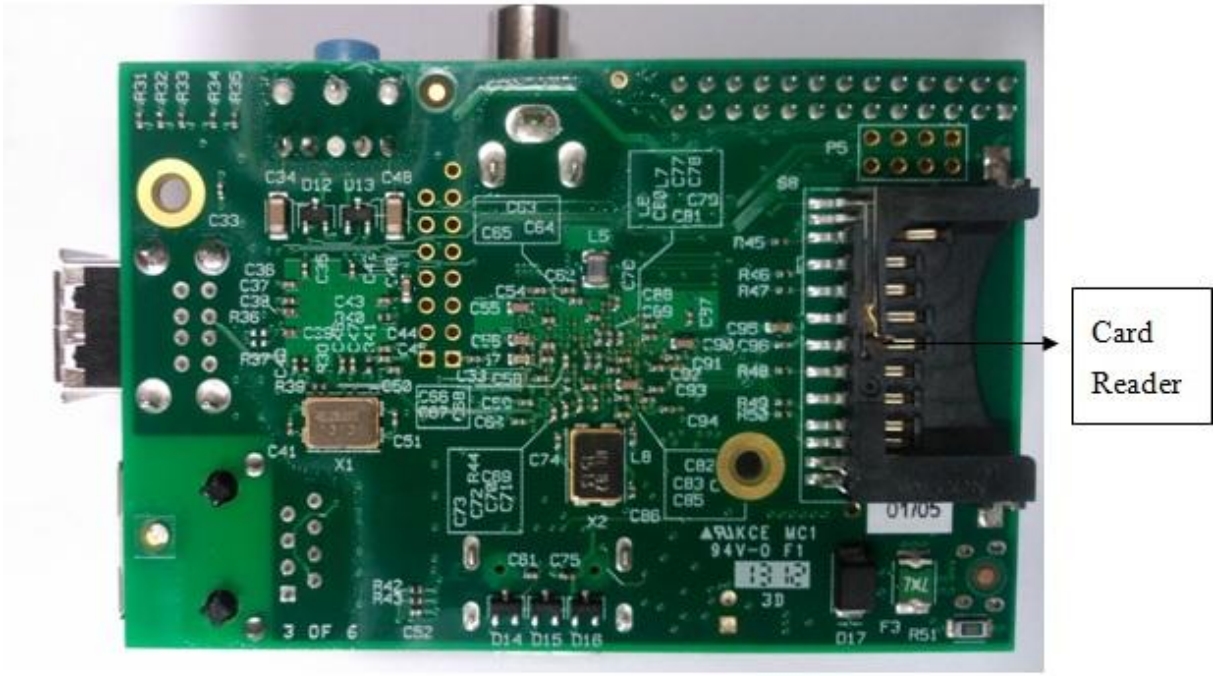


Figure 2-2: Back part of Circuit board of Raspberry Pi

The working of Raspberry Pi is explained below:

Raspberry Pi is credit-card sized computer which supports Linux based Operating System(OS). The foremost thing is to install O.S in the memory card as Raspberry Pi do not have any Hard-Disk attached by. The O.S can be downloaded from website and installed on Secure Digital(SD) card. There are different types of Linux based O.S, however choose 'Raspbian Debian Wheezy' O.S to install.[2]

Raspberry Pi have 2 USB ports through which Keyboard and Mouse are plugged in. Similarly, HDMI port through which Monitor is connected to display and work. There is Ethernet port to connect the Raspberry Pi in network or Internet. And the GPIO port is used to connect other physical devices as peripherals.

Now, programming can be done in Linux windows for various simple applications. It can be used to control, log and monitor several devices by programming and connecting such devices. Hence, a good knowledge of programming is pros to user who wants to have knowledge and program to various applications.

Once the installation is completed, the windows opens and desktop contains number of icons for user facility like web browser, LTX terminal, Control program, games and others. To connect other devices GPIO pins are used for external connection so that either the data can be read from them for monitoring or send to them for controlling part.

The user must install necessary software that supports LINUX O.S to program for logging, monitoring and controlling section.

2.3 Arduino and Arduino IDE

Arduino is open-source hardware, where the programming can be done to control the electronics devices. Generally, Arduino is used for sensing sensor/devices and controlling them. There are different types of Arduino such as Arduino Uno, Arduino Mega 2560, Arduino Mega ADK and so on. Arduino Uno Rev3 Board is available in market and costs around € 20.00 and Arduino Wi-Fi Shield cost around € 69.00. The one we have used in this project is Arduino Uno Board.[3]

Arduino consists of microcontroller, Reset switch. The physical outlets consists of Power plug , USB connector, Analog pins, Digital pins respectively. The frontal view of Arduino Uno Circuit board is shown in the Figure 2-3 along with the major parts indication. Also, frontal view of Wi-Fi Shield Arduino is shown in Figure 2-4.

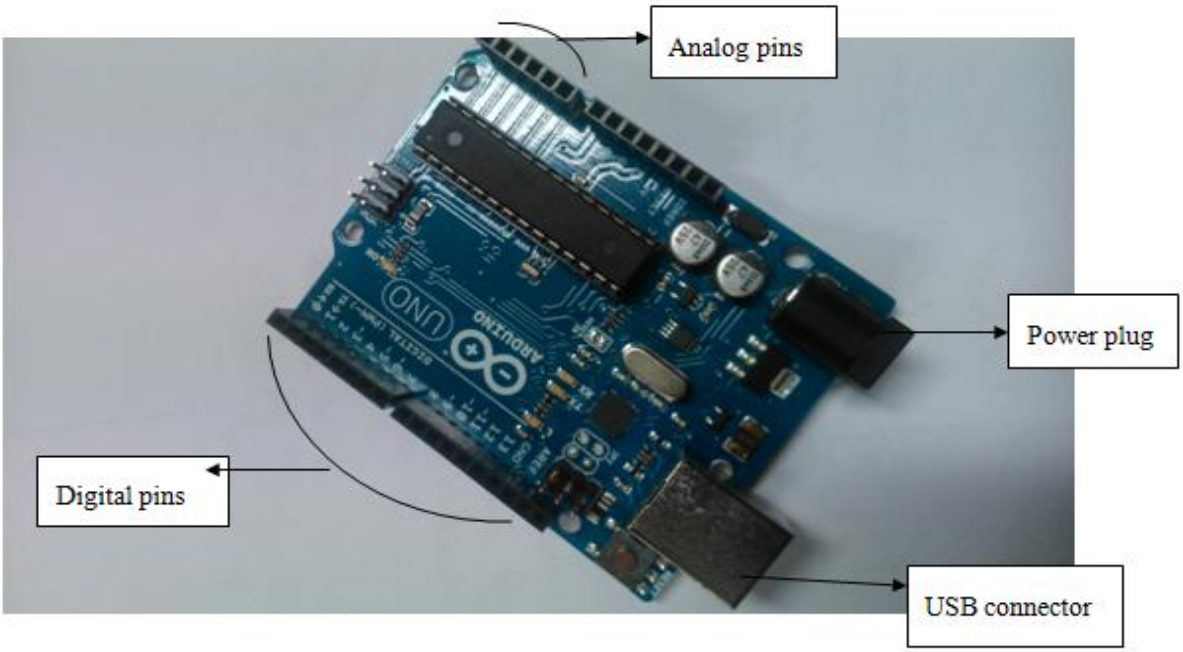


Figure 2-3: Front part of Arduino Uno Board

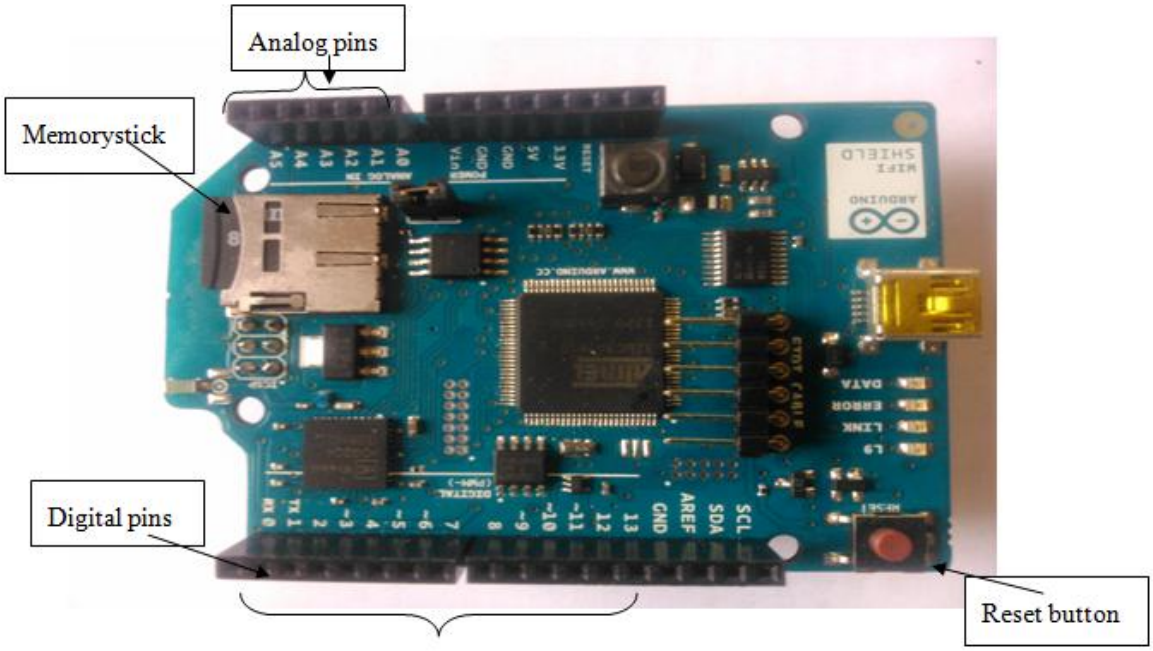


Figure 2-4: Front part of Wi-Fi Shield Arduino

Arduino is microcontroller controlled hardware for interacting the physical world entities like pressure, temperature and controlling other devices accordingly. It needs set of instructions to carry out the tasks which is embedded in microcontroller.

Arduino have power plug to power up the board. Through USB connector, it is connected to computer where the programmed instruction is burned into microcontroller. The 6 Analog input pins receives analog inputs from sensors while 14 Digital input/output pins receive/send digital signals.

Arduino Wi-Fi shield connects Arduino Uno Board to Internet wirelessly. It can be connected to wireless network through programming on Arduino Board. It has micro-SD card slot which can store files. Arduino Wi-Fi shield requires Wi-Fi library function to support wireless functions. It has open pins at back part which can be insert into Arduino Uno Board and then the whole Arduino Board acts as Wi-Fi Shield Arduino device. In HomeAutomation, Wi-Fi Shield Arduino is used so that it can be placed anywhere in house and access information accordingly.

The programming part is essential in Arduino as it is the brain behind Arduino functioning the tasks. For programming, Arduino IDE software is installed which is available on Arduino official site. These set of instructions are programmed by user through Arduino software and burned in microcontroller.

Arduino IDE is the open-source software for coding and uploading it to the Arduino board. It can be downloaded from Arduino official website and is free of cost. This software can run on O.S like Windows, Linux, MAC. This software can be written in any Arduino Board hence applicable for all. The coding in this software is simple and easy to use. The programming is relatively simple and easy to understand. The Basic programming model format consists of two functions:

void setup() {} // We write initialization variable or initial things before main function.

void loop() {} // We write loopable tasks or things to control in this function.

2.4 MySQL Database Server

It is an open-source relational database application that uses structured query languages. It is one of the widely used database used in web applications.[4]

With MySQL, we can add, access and manage content in database. SQL offers fast processing, ease, scalable and flexibility in use. Also MySQL is part of open-source PHP application. It is client-server system that consists of multi threaded SQL server that supports different client programs, wide range of applications programming interfaces.

For example, a database name homeautomation2 is created, but we have to be sure for installing MySQL application prior following below commands on LXTerminal.

```
SUDO MYSQL -U ROOT -P
ENTER PASSWORD:      (you must enter root password here)
CREATE DATABASE  homeautomation2;          (create database by giving name)
USE homeautomation2;  (use your database  now)
CREATE  TABLE  weather (create table for your database )
(
data  INT(11)  NOT NULL,
regtime TIMESTAMP
);
INSERT  INTO  weather (data.regtime)VALUES (150,now()); (Insert data for your table)
SELECT * FROM weather; (display inserted data  that is stored in table)
```

This is the basic commands for creating database and tables in MySQL Database Server.

2.5 Apache-Web server

Apache is one of the most used WebServer software. This application can be downloaded from Internet and is free of cost. It is available for different O.S such as Linux, Windows, Unix etc. It supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. [5]

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs). It also supports password authentication and digital certificate authentication.

2.6 Database & Database Management Systems

Database is a collection of information that is organized in systematic way. It is organized in a manner that the stored information can be easily retrieved, updated, analyzed and output as and when required. The data can be stored in the computer/server in the form of tables, text, scripts, reports, graphics etc.[6]

Database Management Systems is application/programs enabling user to store, organize, select and analyse data in a database. Hence, DMS is needed for accessing information from a database. Some of the DBMSs are MySQL, Microsoft SQL Server, Oracle, IBM DB2 etc.

There are different types of DBMSs according to the management of database structures. Some of the types of DBMS are: Hierarchical DBMS, Network DBMS, Relational DBMS, Object-oriented DBMS.

The most common used is the relational database. In relational database, data are stored in table form. The table contains list of columns in which related data are entered ,stored and also retrieve when required. MySQL Database Server uses Relational database model for managing data.

2.7 PID Controller

PID (Proportional-Integral-Derivative) Controller is a type of controller used for controlling the process or devices to desired operation value so that output is always fixed. PID Controller uses feedback mechanism process to minimize the error for desired output. PID uses 3 parameters called PID where P is Proportional, I is Integral and D is Derivative for tuning input to desire output.[7]

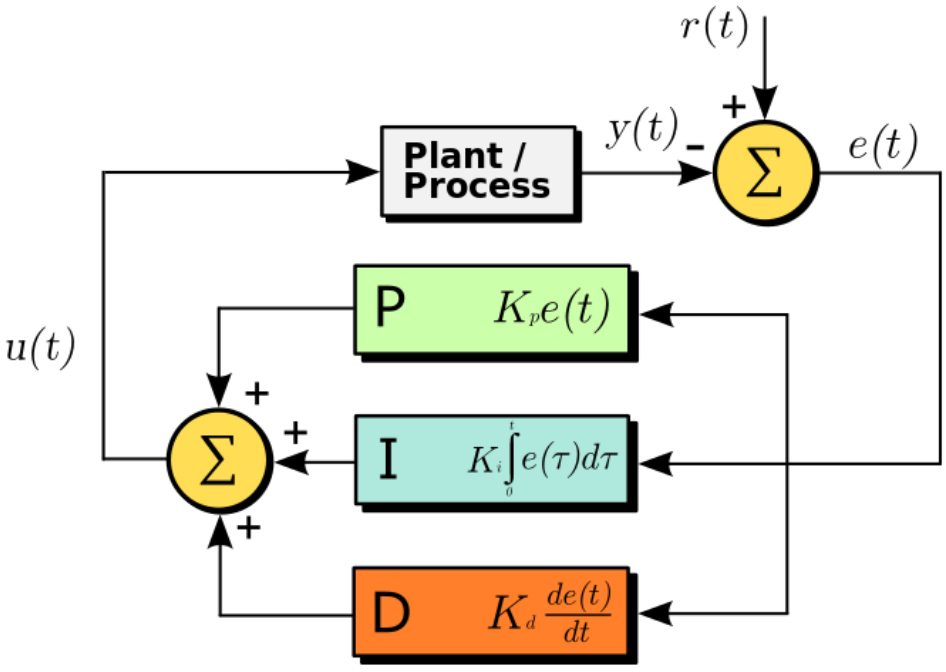


Figure 2-5: Figure of PID Controller working mechanism[7]

The Figure 2-5 shows the mechanism by which Plant/Process is controlled and as we can see there is manipulated input to plant process from PID Controller after it detects error with the reference input to be fed in the Plant/Process for desired output.[26]

2.8 Putty and Xming/Xlaunch

Putty is free Telnet and SSH open-source terminal software. Putty is developed for both Windows and Linux. It supports network protocols like SSH, Telnet, RLogin, Serial connection.[8]

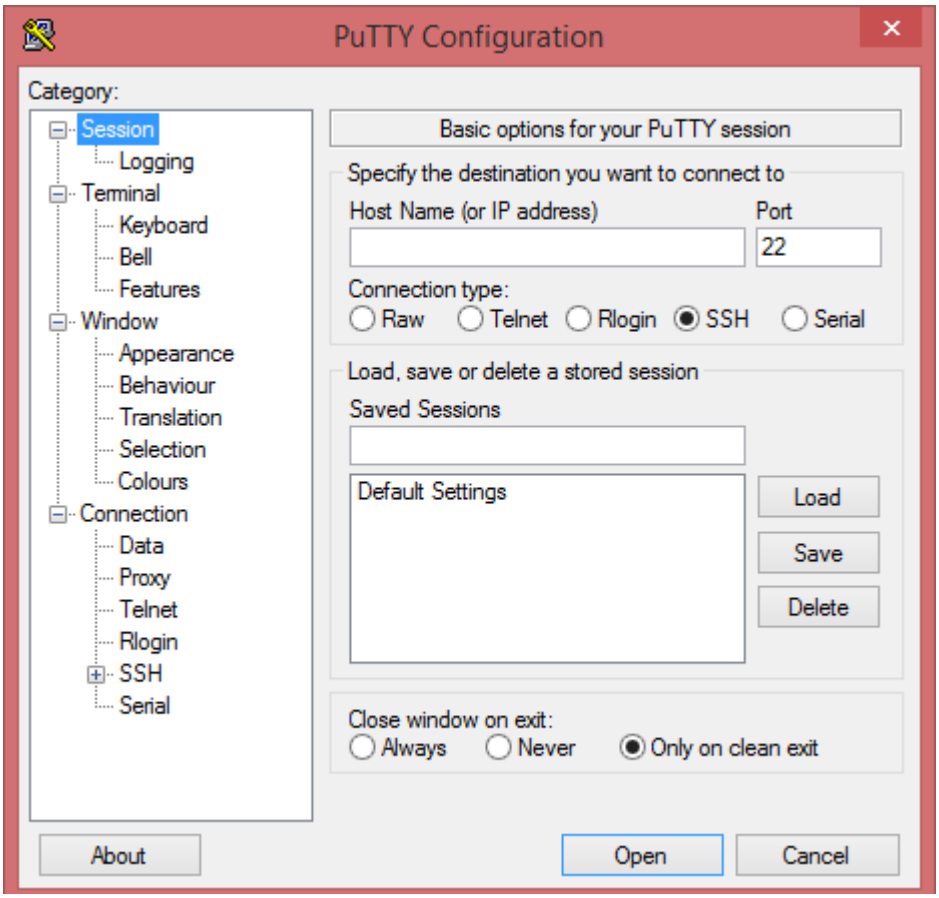


Figure 2-6: Figure of Putty software

The IP address is used for connection to specified network device. In other words, Putty is program to work on remote machine from other device. We can connect by entering the hostname of the machine. Figure 2-6 shows the application page of putty software. Xming is Windows System Server for Micorsoft Windows.[9] Xming allows to connect multiple windows in host PC. Running Xlaunch application allows features to connect multiple windows.

In HomeAutomation, both putty and Xlauch/Xming service is necessary as to control Raspberry Pi via remote PC. It is not always possible to sit in front of Raspberry Pi device and do work. Putty and Xming/Xlaunch offer us with virtual desktop of Raspberry Pi which helps us to access Raspberry Pi device at any place and time.

3 Methods

3.1 Procedure Process

The procedure for basic installing of Raspberry Pi and connection to Arduino is important for HomeAutomation tasks. We have different sources guidelines to install the Raspberry Pi and use it to connect Arduino. The most common book reference we used are 'Home Automation with Raspberry Pi ' and 'Beginning Sensors Network with Arduino and Raspberry Pi '. The screenshot of the respective books to indicate guidelines is shown Figure 3-1 & Figure 3-2.

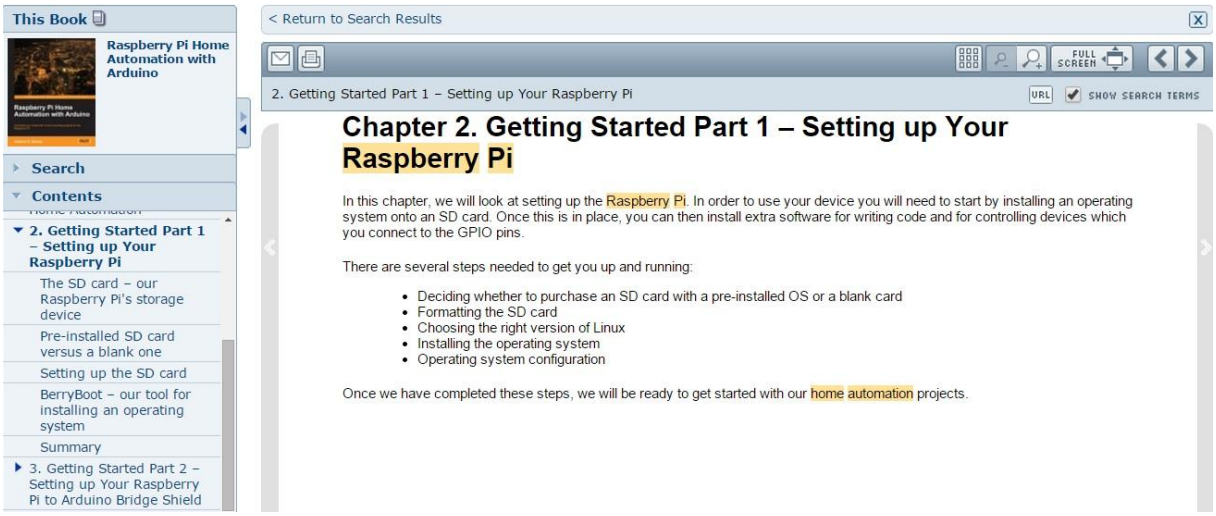


Figure 3-1: Book reference for basic Raspberry Pi installation[10]

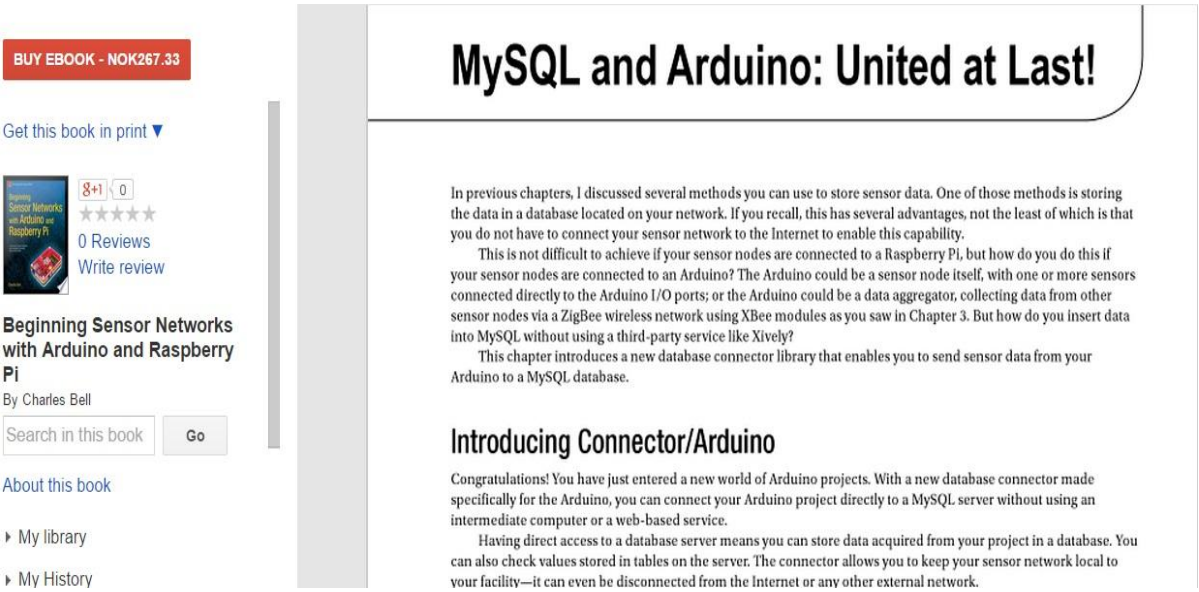


Figure 3-2: Book reference for basic Arduino and MySQL connection[20]

3.1.1 Installation of Operating System and other Programs

The HomeAutomation tasks begin with installing O.S in memory stick card in Raspberry Pi device. The O.S was downloaded from Internet through home page of Raspberry Pi. The web page provides a number of O.S and we choose to install 'Raspbian Debian Wheezy' O.S [10]. Since with 'NOOBS' O.S, the output window screen of Raspberry Pi was inverted and also it was difficult to access from outside using Putty application.

The O.S was successfully installed under guidance from sources from Internet. Later we installed many features/services in O.S to support other features such as Arduino IDE software, MySQL Database Server, PHP5, Apache-WebServer and several others program necessary for our tasks. For such features to install, Internet connection was established in Raspberry Pi device.

Then, the Raspberry Pi device was ready for use and operation. Depending on programs requirement and problems faced, necessary software installment can be upgraded and installed later.

The procedures in steps are pointed as below:

1. Install Raspberry Pi O.S on Raspberry Pi. The O.S downloaded from website is zip(compressed) file[1]. We unzipped it and transferred the unzipped O.S to Raspberry Pi SD card. The unzipped folder snapshot is shown in Figure 3-3 :

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
defaults	File folder					01/02/2015 04:32
os	File folder					01/02/2015 04:41
bootcode	VLC media file (.bin)	9 KB	No	18 KB	54%	01/02/2015 03:15
BUILD-DATA	BBiitSavear	1 KB	No	1 KB	30%	01/02/2015 04:37
INSTRUCTIONS-README	Text Document	1 KB	No	3 KB	57%	01/02/2015 04:28
recovery	Disc Image File	2,044 KB	No	2,053 KB	1%	01/02/2015 03:14
recovery.cmdline	CMDLINE File	1 KB	No	1 KB	0%	01/02/2015 03:15
recovery.elf	ELF File	345 KB	No	541 KB	37%	01/02/2015 03:15
recovery.rfs	RFS File	17,755 KB	No	19,763 KB	11%	01/02/2015 03:14
RECOVERY_FILES_DO_NOT_EDIT	BBiitSavear	0 KB	No	0 KB	0%	01/02/2015 03:15
recovery7	Disc Image File	2,090 KB	No	2,098 KB	1%	01/02/2015 03:15
riscos-boot	VLC media file (.bin)	1 KB	No	10 KB	99%	01/02/2015 02:45

Figure 3-3: Snapshot of folders and application of Raspberry Pi O.S

2. Thereafter, O.S starts to work in Raspberry Pi device. The outlook of O.S on screen/monitor is shown in Figure 3-4.

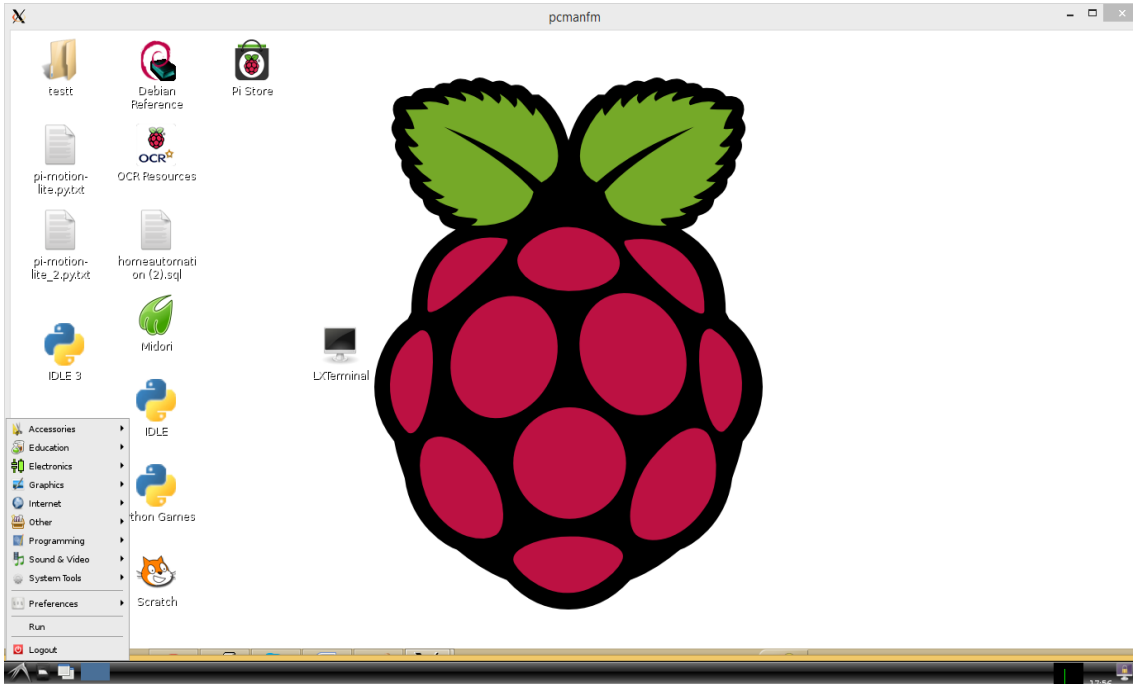


Figure 3-4: Snapshot of windows screen of Raspberry Pi

3. Now, we look for access to Internet. Raspberry Pi have Ethernet port to support network communication. By default, Raspberry Pi device has automatic setting for DHCP network communication. Therefore, when there is network cable connection, Raspberry pi device assigns Dynamic IP address and there is network connection.

   However in many cases, there needs to assign Static IP address to Raspberry Pi device. This can be done easily by entering command line to make Static IP. For this, we open LXTerminal command and enter the following commands [11]:

   sudo nano /etc/network/interfaces

   Then it displays as below:

   iface eth0 inet dhcp (for dynamic ip address)

   To change this dynamic address configuration to static address, Enter following commands:

   #iface eth0 inet dhcp (To deactivate dhcp server for static ip, put # in front as shown)
   Then type below commands:
   auto eth0
   auto eth0 inet static
   address 192.168.0.25 (Give your Static IP address)
   netmask 255.255.255.0 (Give your network netmask address)

   Next, edit resolv.conf file by entering following commands.
   sudo nano /etc/resolv.conf

   Then enter commands as below:
   nameserver 8.8.8.8
   nameserver 8.8.4.4

Then save it by pressing ctrl-o and Enter followed by ctrl-x. Now we restart the networking service to activate Static IP address that we have just entered in command line. The networking service is restarted by entering commands below:

sudo service networking restart

We can check if the configuration has worked by entering command as below:
ifconfig (displays ip address with full information)

4. Now, we install necessary software for HomeAutomation project. The software are installed by entering respective commands on LXTerminal. The software installation begins with WebServer-Apache installation, MySQL Database Server, PHP installation, Arduino IDE installation and others. It should be noted that while installing the software the network(Internet) connection must be established otherwise the installation halts.

5. Installation of WebServer -Apache Server.[12] This software application is necessary for establishing web server in Raspberry Pi device. Enter the following commands on LXTerminal to install it:

   Step1: **sudo apt-get update && apt-get upgrade**
   Step2: **sudo apt-get install apache2 -y**
   Step3: **sudo reboot**

   Then, test the WebServer running on Raspberry Pi by browsing on local browser as http://localhost/. The webpage responds to WebServer as shown in Figure 3-5

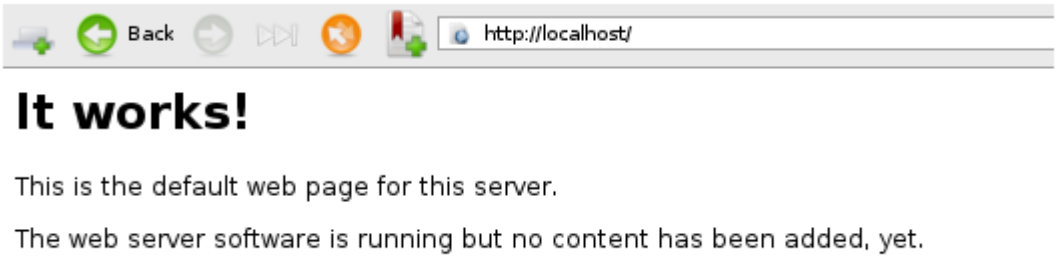


Figure 3-5: Screenshot of browser showing WebServer working in Raspberry Pi

6. Installation of MySQL Server[13]. This software application is necessary for local server database in Raspberry Pi device for HomeAutomation. Enter the following commands on LXTerminal to install it:

   Step1**: sudo apt-get update && apt-get upgrade**
   Step2: **sudo apt-get install mysql-server --fix-missing**
   Step3: **sudo apt-get install mysql-client php5-mysql**
   Step4: **sudo reboot**

7. Installation of Arduino IDE.[14] This software application is necessary for coding Arduino code and uploading code in Arduino device. Enter the following command on LXTerminal to install it:

   Step1**: sudo apt-get update && apt-get upgrade**
   Step2: **sudo apt-get install arduino**
   Step3: **sudo reboot**

   After installation, the Arduino IDE software appears on Start menu of Raspberry Pi windows on Electronics-Arduino IDE. Similarly, Sketchbook folder for storing Arduino files is present in /home/pi.

8. After these installation, we can move on doing main HomeAutomation monitoring, logging and controlling section.

3.1.2 Database Design and Implementation in MySQL Server

After 3.1.1, we focus on Database implementation in MySQL Server in Raspberry Pi. For this, the database design was necessary as to construct database tables and their respective columns and define the relationship that can exist among different tables.

As from task description, HomeAutomation deals with information that has to be logged in database and monitored through web page on browser through laptop or tabloid, it was necessary to build the model that fulfill the task requirement and also tie the relationship between. So, database design was started to link on how these system coordinates and function together to accomplish task.

The first action done was design of Database model i.e make E/R relationship in Erwin Software for HomeAutomation. For that, we must download Erwin software from Internet[15]. The E/R relationship in Erwin features Table with Columns, Primary & Foreign Key, and other necessary tools. Here, the basic design was developed using Tables & Columns so to include information to be record and processed. The E/R diagram of HomeAutomation database design is shown in Figure 3-6:

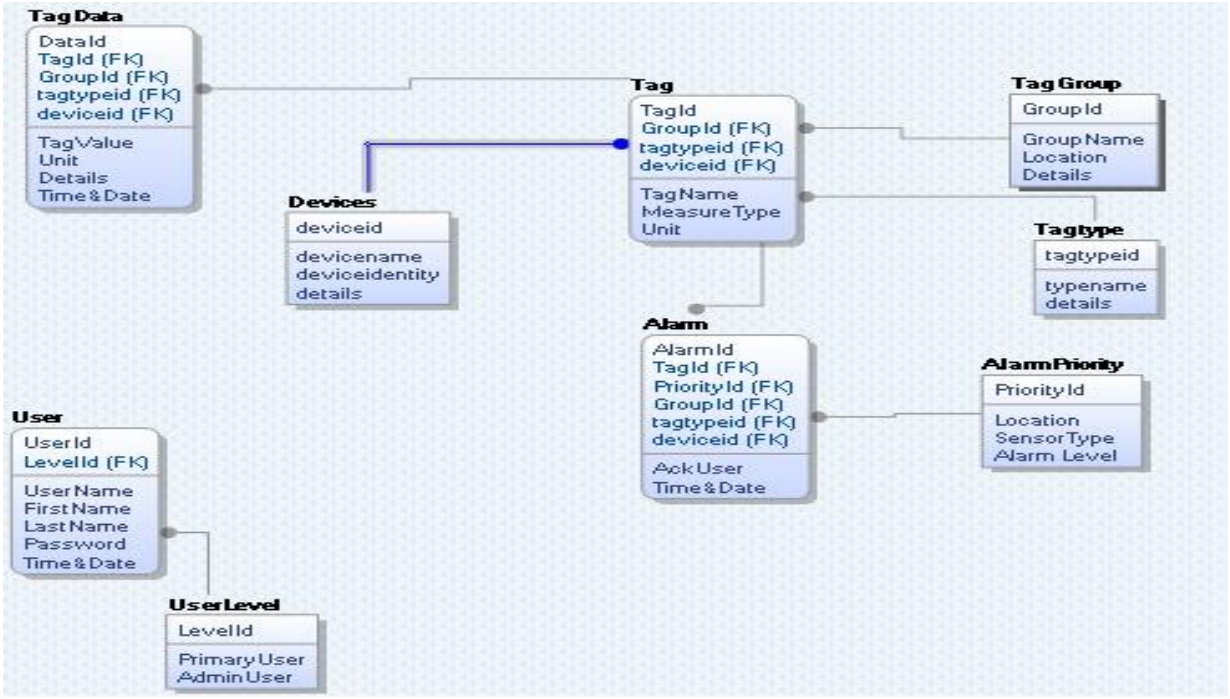


Figure 3-6: E/R diagram of Home Automation database

Under this relationship, this database design has to be implemented in MySQL Server in Raspberry Pi device for HomeAutomation task. Tables were created and columns were added together with indication of primary key and foreign key relationship based on their link with each other table. Under table section, several columns were created for the information to be recorded or managed.

Now the database has to be created in MySQL Server. There are two methods to implement the database with tables and columns :

a) The first method is by using LXTerminal and entering the command necessary to create the required database and its table & columns.
b) The second method is to install PhpMyAdmin in Raspberry Pi and create database along with tables and columns. The second way is preferred as it is much easier and simpler to use just by browsing link to localhost/PhpMyAdmin/ and creating database with the list of option available.

3.1.2.1 Install PhpMyAdmin in Raspberry Pi to handle administration of MySQL Database Server

We then install PhpMyAdmin in Raspberry Pi so that we can easily administrate MySQL Database Server in browser to manage database. This feature enabled to create database simply by logging on local PhpMyAdmin web page on browser and then click on create

panel to create new database we desire for. Further, tables and columns can be created, edited or as well delete and other properties related to implement of database was available. The feature made more simple to manage MySQL Database Server in Raspberry Pi. The problem of using command line on LXTerminal to manage database was sorted out, which is great relief for beginners and with weak PHP coders. Hence, the database for HomeAutomation was successfully implemented. The Figure 3-7 shows the PhpMyAdmin page in browser:

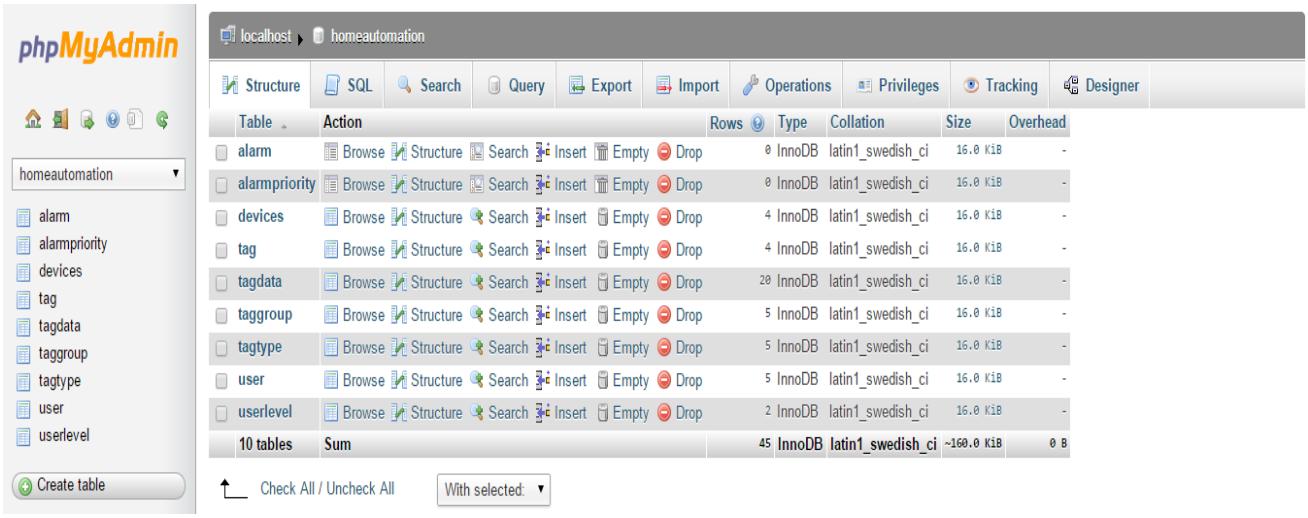


Figure 3-7: PhpMyAdmin in browser to administer database

The list of Tables created for database 'HomeAutomation' are as shown in Figure 3-7. The list of tables created for database 'HomeAutomation' are: User, Userlevel, Tag, Tagtype, Tagdata, Tagroup, Devices Alarm and Alarmpriority.

The snapshots of some of the Tables with Column for HomeAutomation Database is shown in Figure 3-8, Figure 3-9, Figure 3-10, Figure 3-11, Figure 3-12, Figure 3-13, Figure 3-14.

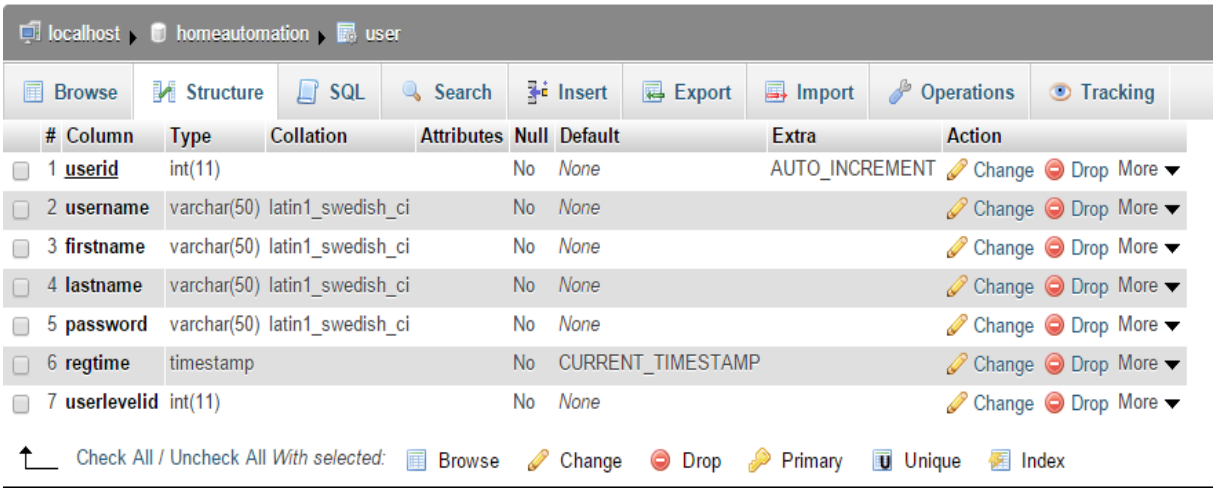


Figure 3-8: Snapshot of User Table & Column in PhpMyAdmin

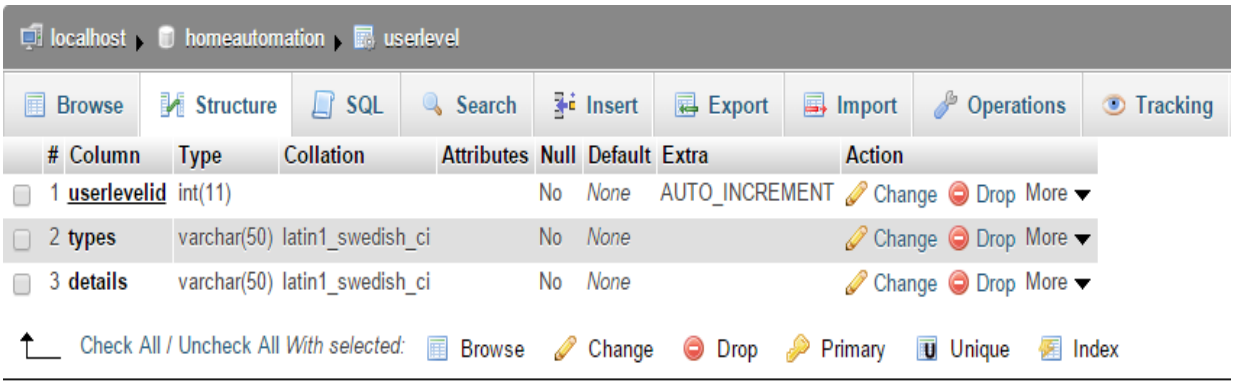


Figure 3-9: Snapshot of Userlevel Table & Column in PhpMyAdmin

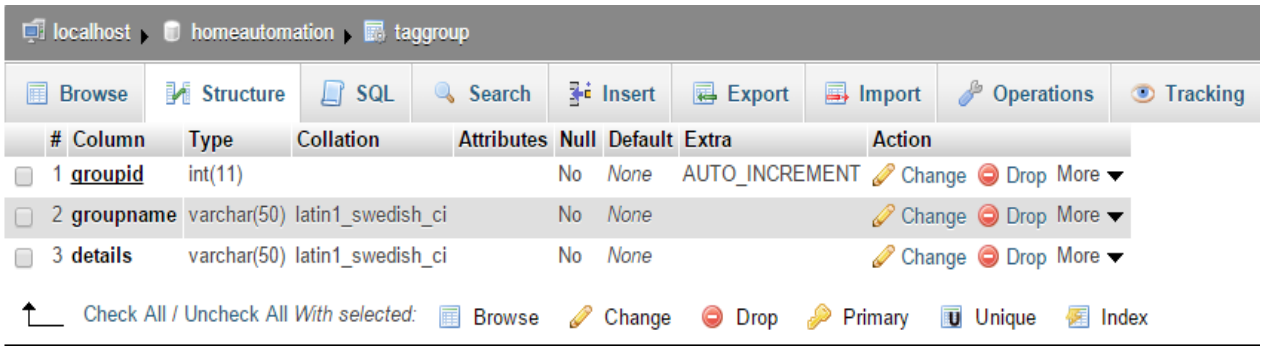


Figure 3-10: Snapshot of Tagroup Table in PhpMyAdmin

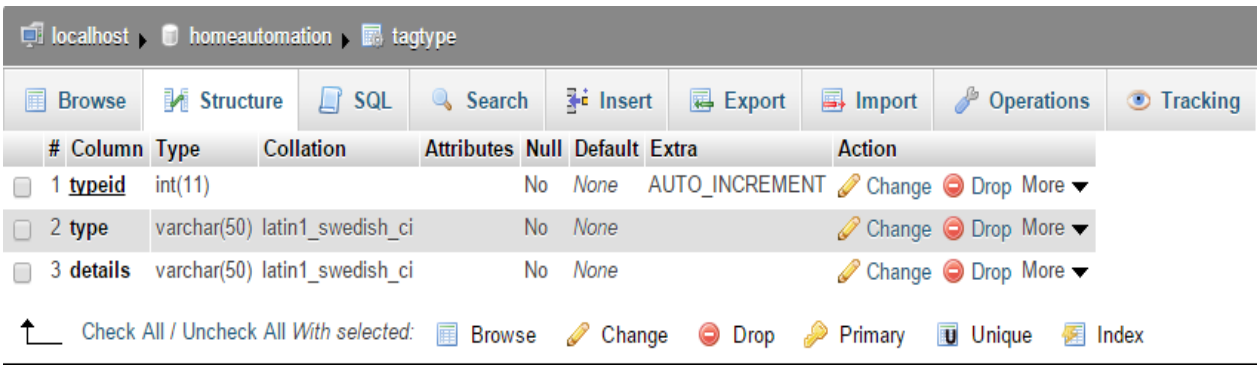


Figure 3-11: Snapshot of Tagtype Table & Column in PhpMyAdmin

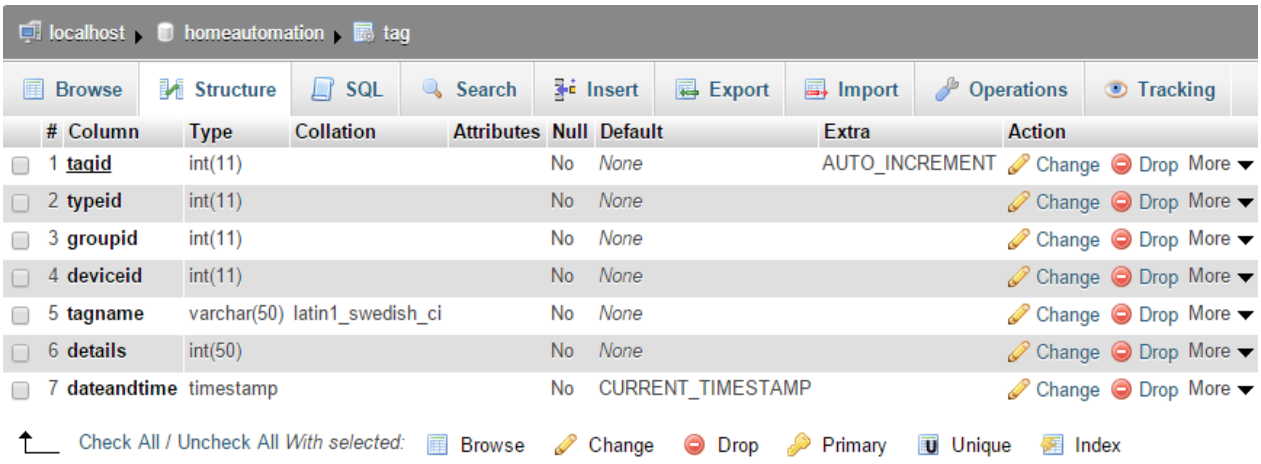


Figure 3-12: Snapshot of Tag Table & Column in PhpMyAdmin

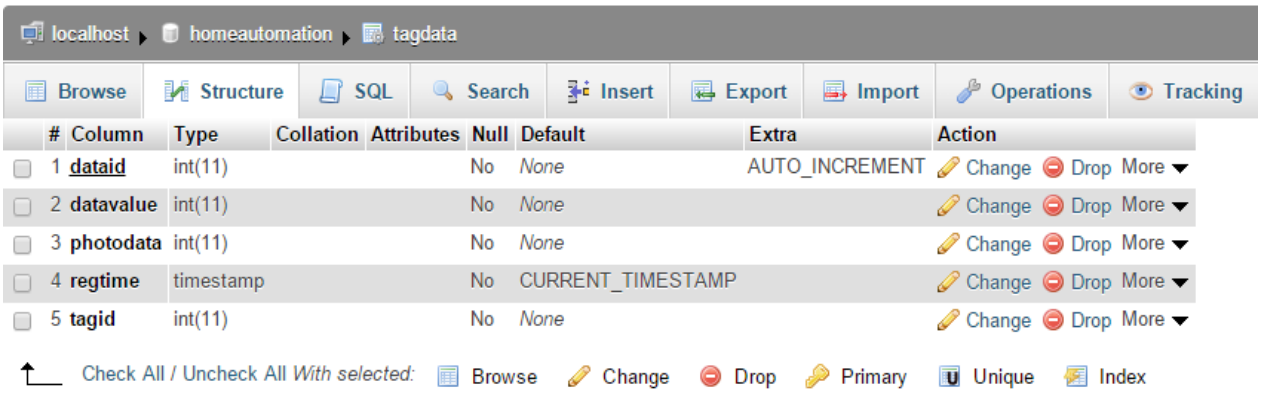


Figure 3-13: Snapshot of Tagdata Table & Column in PhpMyAdmin

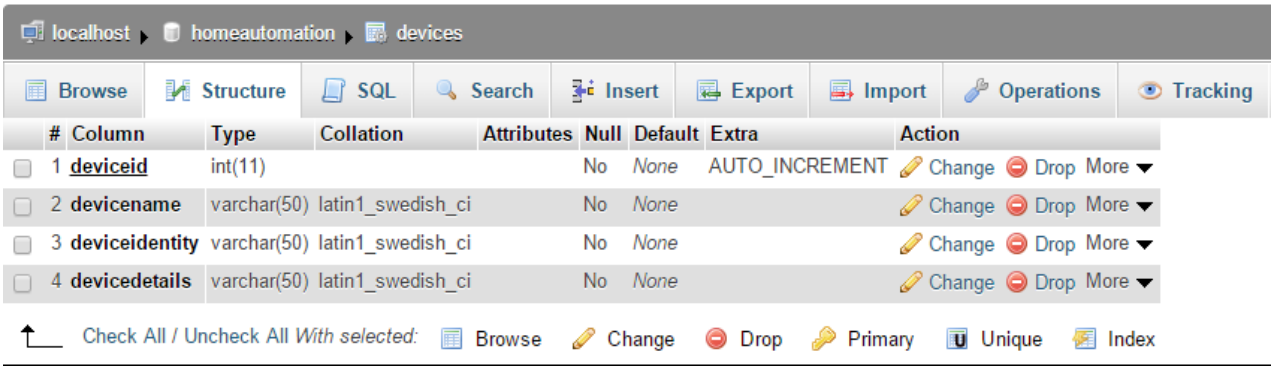


Figure 3-14: Snapshot of Devices Table & Column in PhpMyAdmin

3.1.3 Install of Apache Web-Server and Design of Webpage

After 3.1.2, the next important and challenging part in HomeAutomation was to monitor and control the different appliances in house. There has to be medium/platform through which user can access the information stored in database and monitor them. The webpage can be such medium where the all the information can be displayed at one place and can be seen from anywhere in the worlds.

For that, we must install Apache Web-server application in Raspberry Pi[12]. It enables us to upload designed webpage in WebServer in Raspberry Pi. Using Apache Web-Server, any web pages that are under it can be accessed globally provided that Raspberry Pi device have unique static IP address. In Raspberry Pi, this WebServer is located on File Manger-var/www.

The web page design part were more of challenging in the sense that they had to be programmed, designed beautifully & display the required information from database. For this, we must install Dreamweaver[16] and Xampp application[17] in our local computer. Dreamweaver is used to design, create web page and Xampp to create local database and local WebServer for testing purpose before we finally transfer and implement in Raspberry Pi.

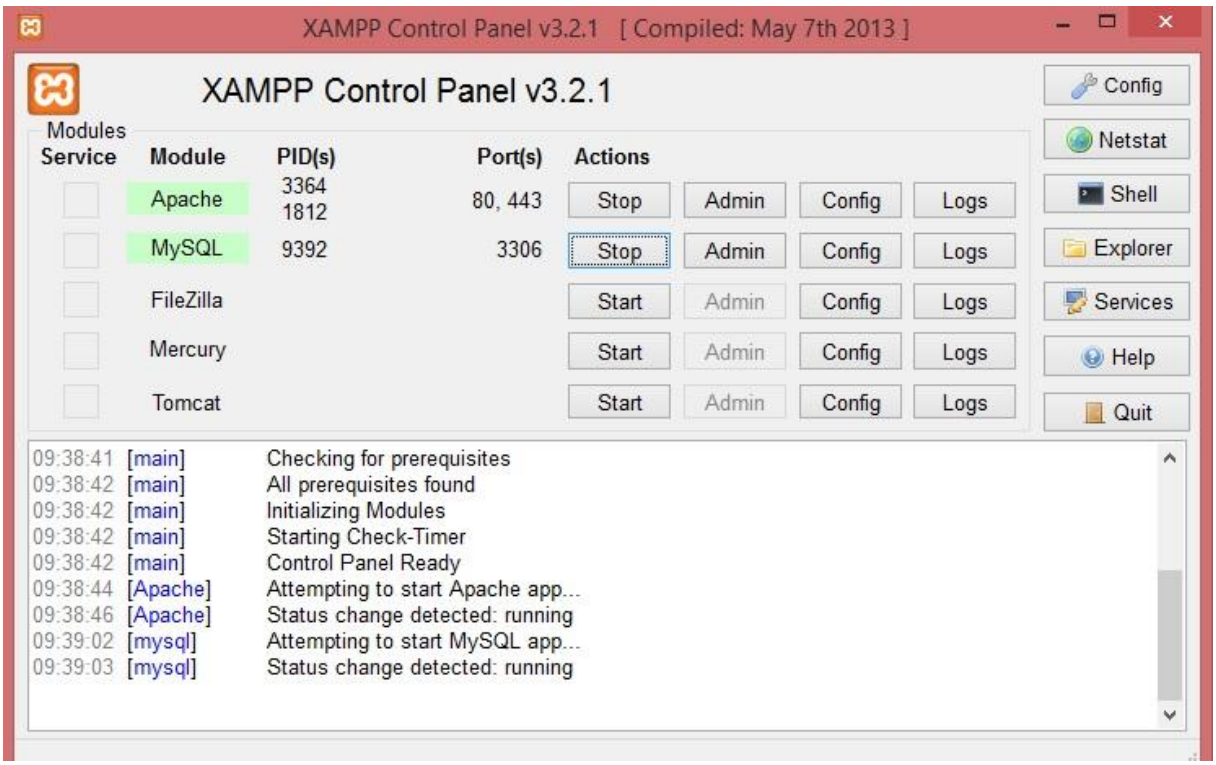


Figure 3-15: Xampp Sever installed in local computer and run

When installation of XAMPP application in PC is completed, then a file appears as shown in above Figure 3-15. Here, we have several application that XAMPP supports like Apache web server, MySQL, FileZilla etc.

As we need only Apache WebServer and MySQL application to test the created web page in PC, we click on Start option of Apache and MySQL service. There is Admin option next to Start option as shown in Figure 3-15, through which we can have admin settings of those services. Hence, through XAMPP which have local WebServer and MySQL service, the web page which we created/designed can be viewed directly on real-time so that the quick view of created web page can be done AND also created/designed webpage have access to database service(via MySQL). As a result of which database information can be dragged and shown in web page.

Dreamweaver allows us to see the designed web page look in different browser. Also that it is far more fast and quick to design web page in our local computer than in Raspberry Pi. Through MySQL service in Xampp, we have feature to access database and implement the web page we have created.

An example of web page designed in Dreamweaver is shown below in Figure 3-16:

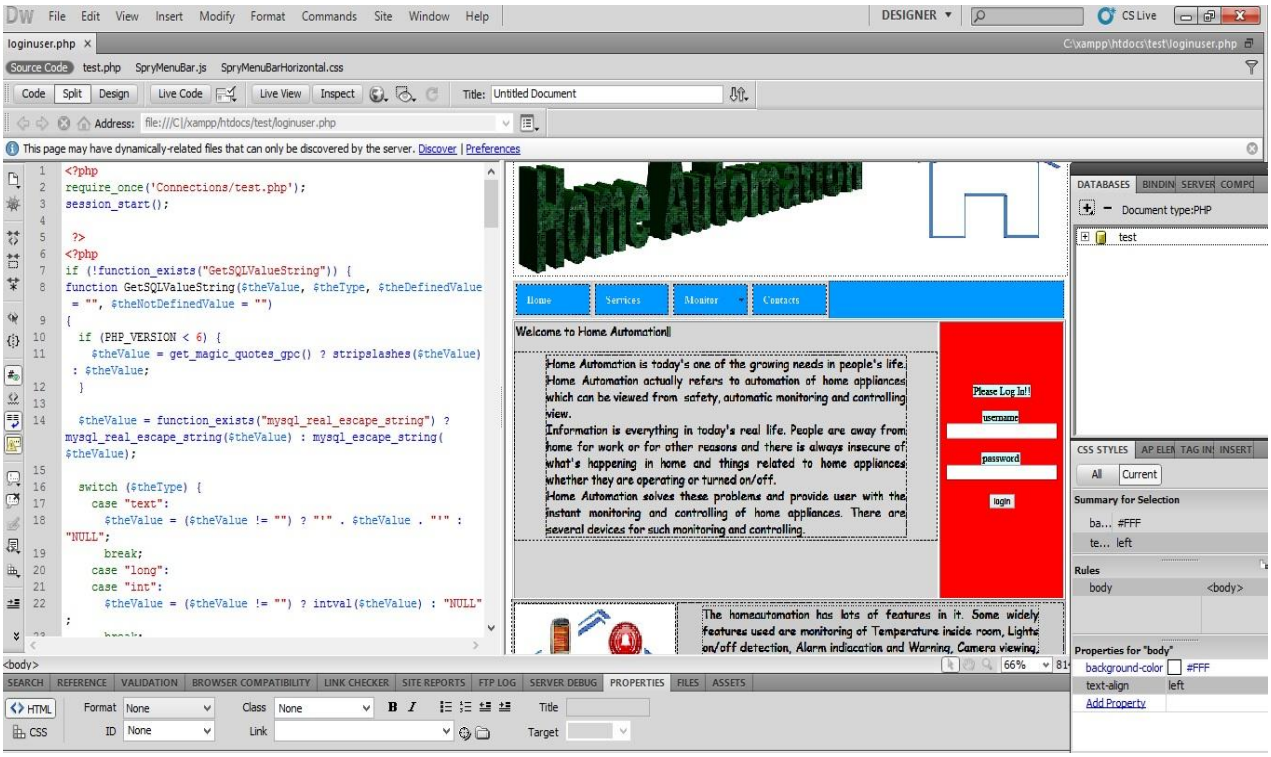


Figure 3-16: Designing web page called index.php in Dreamweaver for HomeAutomation

Later on, when the overall web pages for HomeAutomation task is completed, it must be transferred to WebServer of Raspberry Pi device. The transfer of files from our PC to Raspberry Pi device follows steps as below:

- First, we open localhost/phpmyadmin/ in our local browser in PC. Then we select database that we have created for HomeAutomation task. The database created is called **homeautomation.** Now, we need to transfer two files from PC to Raspberry Pi device.
  These two files are:
  a) The folder which contains php web pages designed for HomeAutomation task.
  b) The file(.sql type) which is downloaded by clicking under export as shown in Figure 3-17:

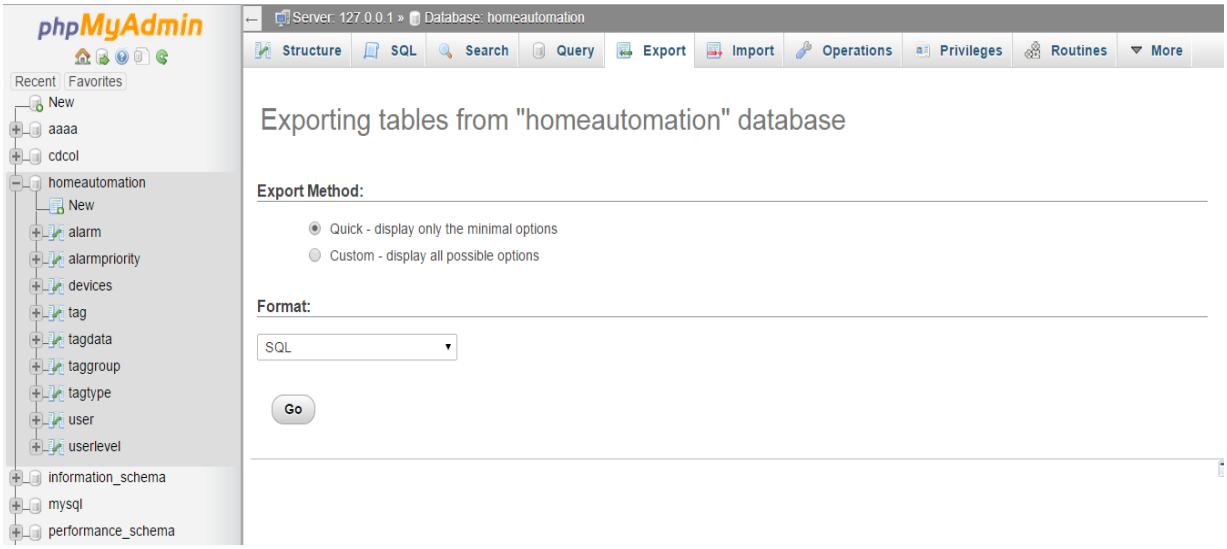


Figure 3-17: Exporting Tables to MySQL Database in Raspberry Pi

- These two files are transferred to Raspberry Pi device. The first file i.e folder which contains web pages is transferred to File Manager inside /var/www and saved there. The next file i.e .sql file is however placed in any folder for some time.

- Then in Raspberry Pi device, we open localhost/phpmyadmin in browser similar to that we did in PC. We create new database and name it similar to that of PC database. The database is created under name **homeautomation**. Under that **homeautomation**, we can see import button in right side. Click on it and it will ask to choose file. Then, direct it to the .sql file in the folder that we saved earlier and click on Go button. The whole database is saved. The snapshot for showing import procedure is shown in Figure 3-18:

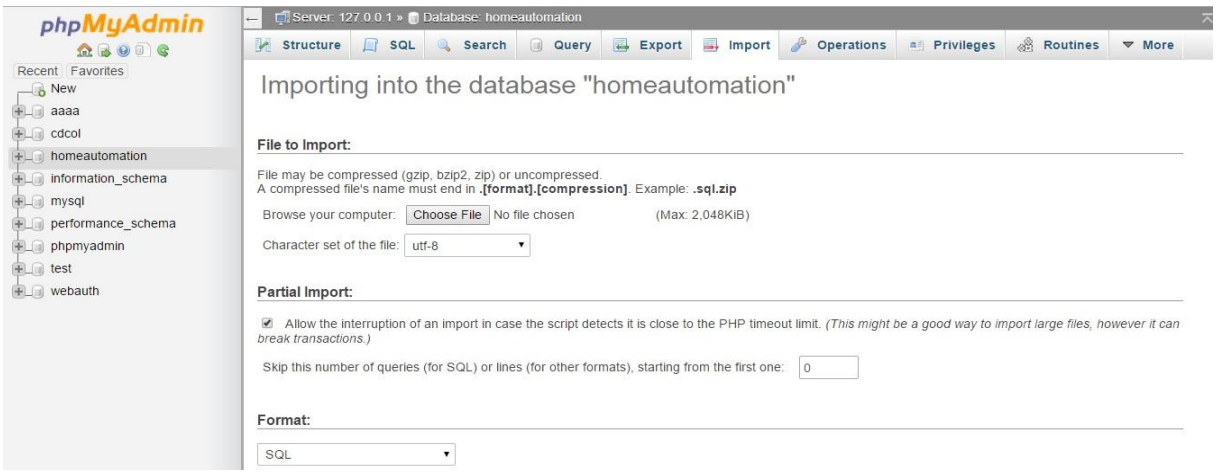


Figure 3-18: Importing Tables from PC MySQL Database

- Now the folder which we saved inside WebServer can access to database of MySQL server and is ready to use.

The problem still lies in addressing and making the web page look good and systematic. The reason behind lies in being new to this kind of programming and lots of time is consumed to know and implement it. The connection php web page files is added in the Appendix,

3.1.4 Putty and Xming

Putty is open software for connecting to other device. In HomeAutomation task, it is used for remote connecting to Raspberry Pi device. Since, it is not always possible to work in front of Raspberry Pi device for several reasons, Putty is a good means to connect Raspberry Pi in our personal computer. We can enter the IP Address of Raspberry Pi in Putty and connect it in our laptop. But before that we must run the Xlaunch application in our personal computer to support multiple windows[18].

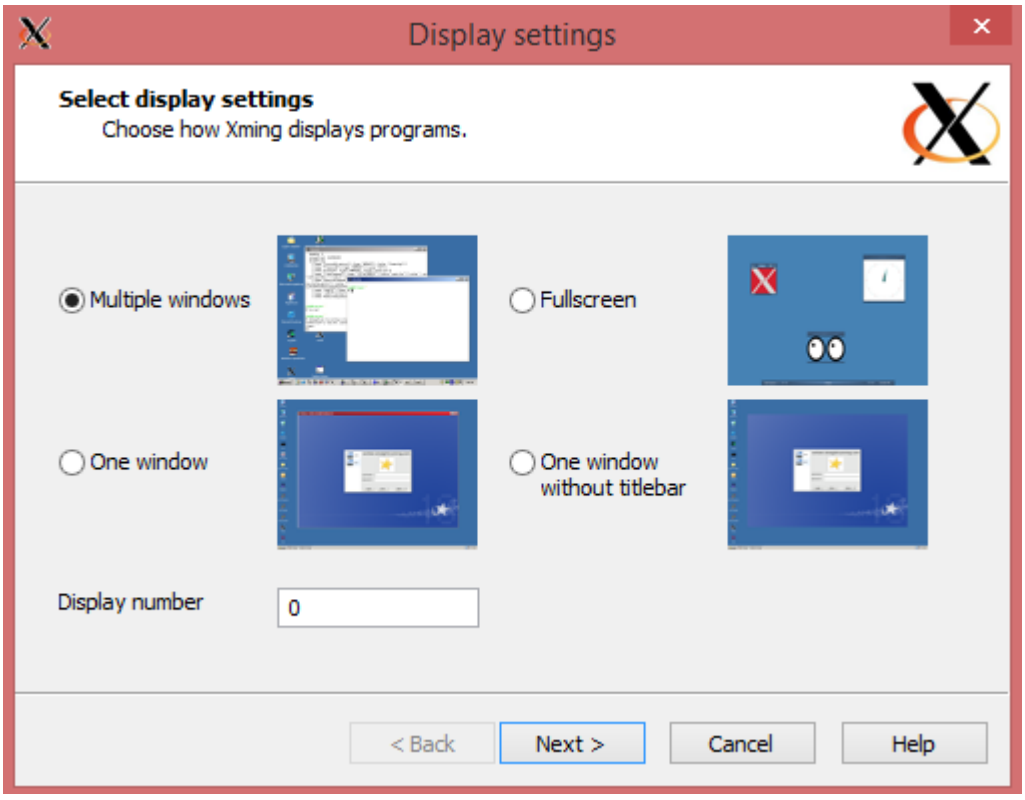
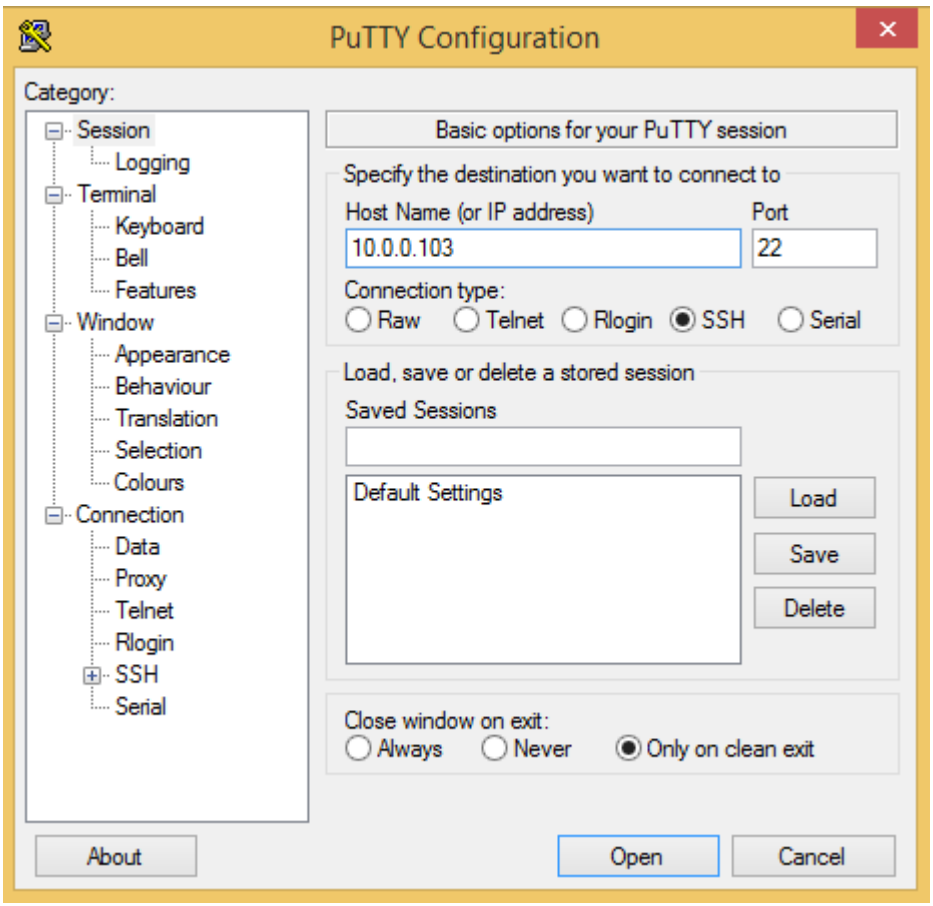
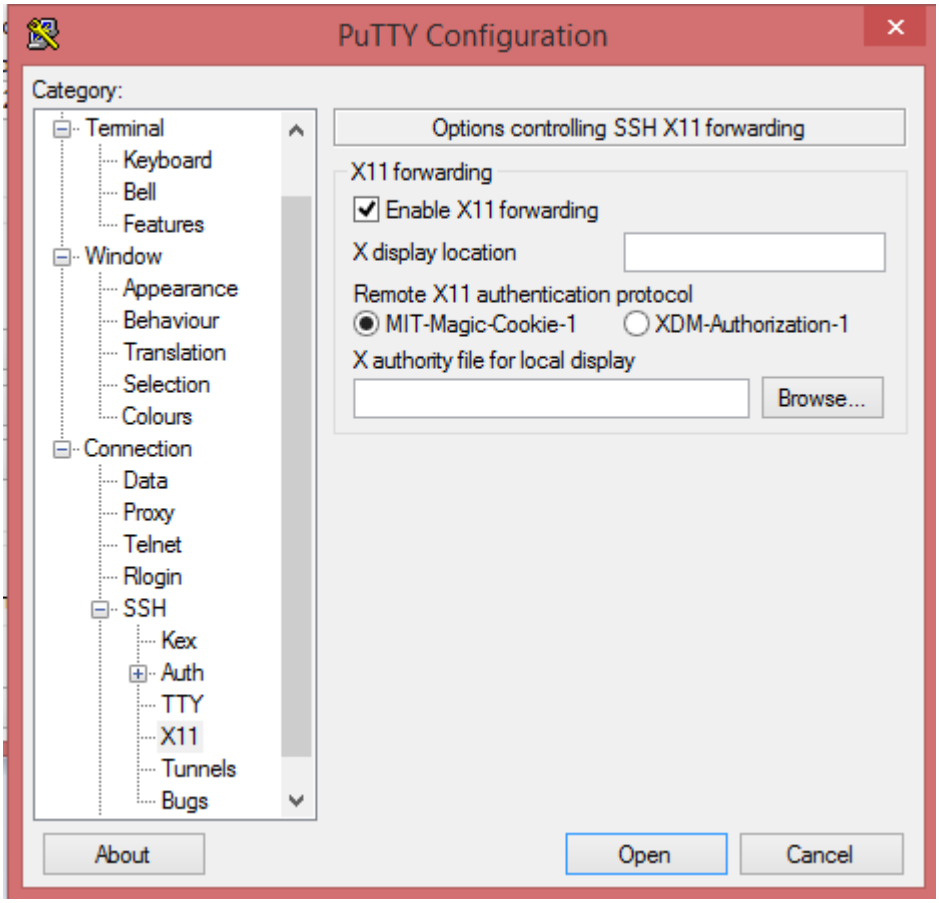


Figure 3-19: Figure of Xluanch starting for multiple windows on PC


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
10.0.0.103
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel


PuTTY Configuration
Category:
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Auth
TTY
X11
Tunnels
Bugs
Options controlling SSH X11 forwarding
X11 forwarding
Enable X11 forwarding
X display location
Remote X11 authentication protocol
MIT-Magic-Cookie-1
XDM-Authorization-1
X authority file for local display
Browse...
About
Open
Cancel

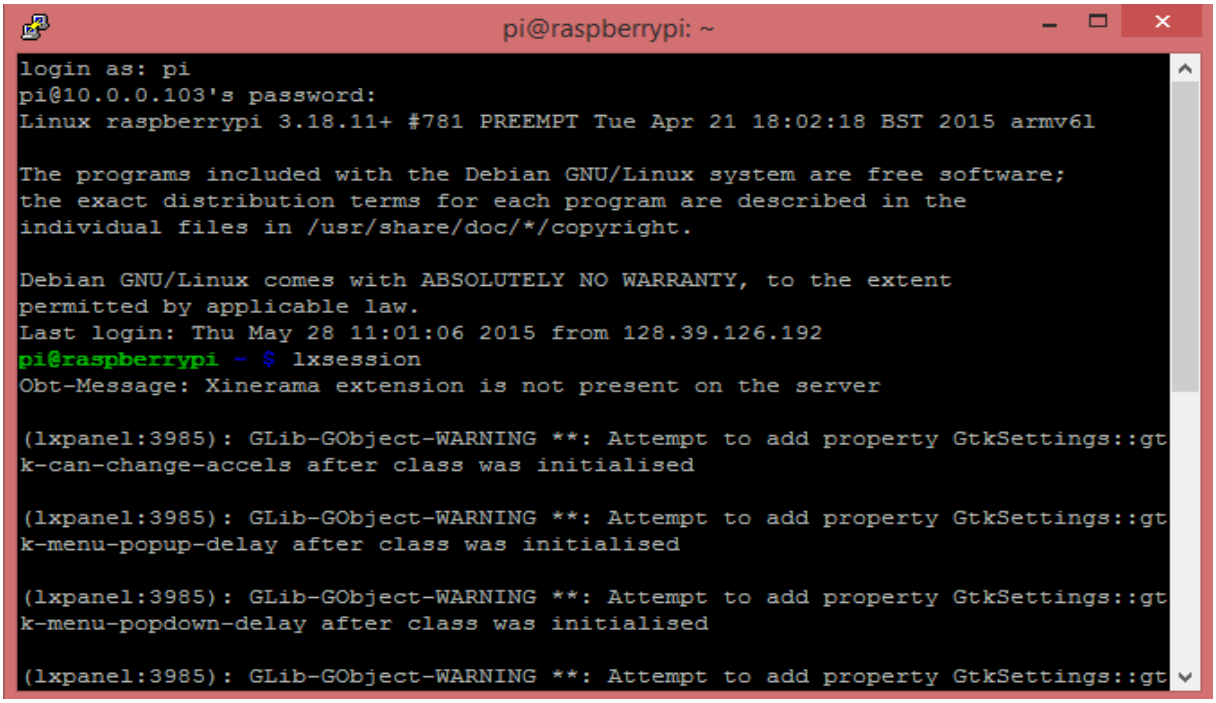


Figure 3-20: Snapshot of Putty software step by step instruction to run in PC

Figure 3-21: Snapshot of Raspberry Pi window on PC window

The list of steps to use this service in remote desktop is given below:

Part I(Procedures for use of Xlaunch application)

- First run Xlaunch application.
- Click on Multiple windows - Next - Start no client - Next- Clipboard -Next- Finish.

Part II(Procedure for use of Putty application)

- Then run putty application.
- Enter Host IP Address (i.e IP Address of Raspberry Pi as shown in Figure 3-20)
- Click on SSH option below IP Address.
- Click on SSH option on left side- Click X11
- Click on Enable X11 forwarding and Open
- A command window appears asking for login name & password.(Enter Raspberry username and password)
- Then at last type lxsession
- Then virtual window of Raspberry Pi opens on your remote PC.

3.1.5 Arduino - Monitor Sensors/ Control Devices

After 3.1.4, the next major and important task was actual real time monitoring, logging and controlling appliances. The challenge was to implement in real practice and program code in Arduino Board to read sensor values from different appliances and send these information to MySQL Database Server in Raspberry Pi device.

There are different version of Arduino Board available. Arduino Uno Board is used in HomeAutomation task. Some of the methods used for collecting information using Arduino Board are :

- Using GPIO pins of Raspberry Pi to connect Serial pins on Arduino Uno Board to collect information from sensors.
- Using Ethernet Shield Arduino to collect information from sensors
- Using Wi-Fi Shield Arduino to collect information from sensors.

As for Home Automation, we use Wi-Fi Shield Arduino to monitor and control appliances since there are several places in home where the appliances have to monitored. We can't use Ethernet based Arduino as there will not be Ethernet plug in every room to connect them. Besides lots of wire through use of Ethernet shield Arduino in home look bulky and complex. And almost every house have router supporting Wi-Fi Service, which can be used for network connection for Wi-Fi shield Arduino. Wi-Fi Arduino facilitates to connect appliances from any corner of room. With Wi-Fi Shield Arduino, data are send serially through port 3306 of Raspberry Pi device into MySQL Database Server.

The programming is necessary for addressing sensors input and directing them to appropriate place. The programming code is written in sketch which is later uploaded to Arduino device through Arduino IDE. This Arduino IDE is software for writing code and uploading. The sketch of coding for Wi-Fi is shown below in Figure 3-22.

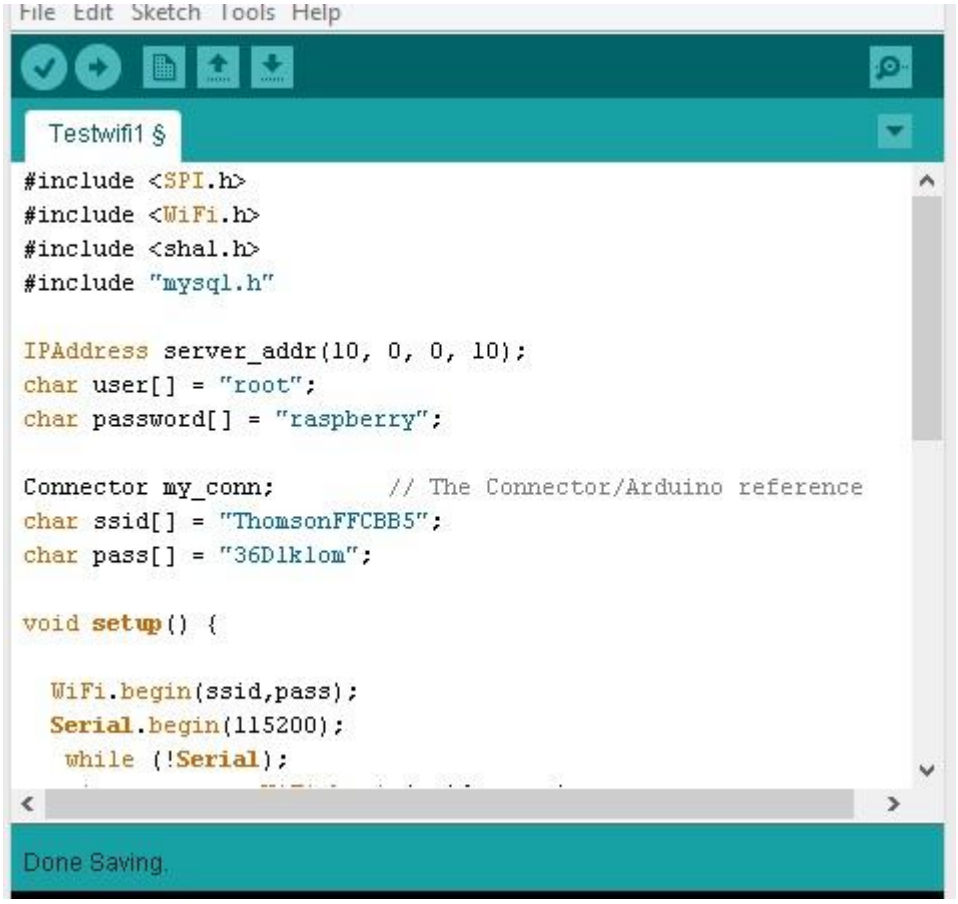


Figure 3-22:A part of Arduino code for Wi-Fi Shield Arduino showing mysql connector code to connect to MySQL Database Server in Raspberry Pi

Both Ethernet Shield Arduino and Wi-Fi Shield Arduino scenarios were tested to read sensor values and connect to local MySQL Database Server in Raspberry Pi. As shown in Figure 3-22, while programming code for Wi-Fi four libraries function are used which are SPI.h, WiFi.h, sha1.h and mysql.h respectively. Out of these four, we have to download two libraries function from Internet which are "sha1.h" and "mysql.h" function.[20] These function are needed to operate the code and function properly. The "mysql.h" function is required to connect to MySQL Database Server in Raspberry. For Wi-Fi based Arduino, we alter network changes in mysql.h function i.e comment on Ethernet.h library function and comment out WiFi.h library function as shown in Figure 3-23.[20] The Wi-Fi Shield Arduino code for HomeAutomation is attached in the Appendix. After that through Webpage, the information in database are displayed.

```
//#include <Ethernet.h> // Uncomment out this for use with the Ethernet shield

#define WITH_SELECT  // Comment out this for use without SELECT capability
                     // to save space.

#define WIFI       // Uncomment out this for use with the WiFi shield
#include <WiFi.h>  // Uncomment out this for use with the WiFi shield
```

Figure 3-23: snapshot of portion of code in MySQL.h function to edit for WiFi Shield Arduino

However, several problems were encountered while connecting Arduino to MySQL Database in Raspberry Pi device. The very first and important problem was that it could not connect to port in Raspberry Pi through which data can be transferred to Raspberry Pi from Arduino. The Arduino code was successfully compiled and uploaded in Arduino chip but still data did not transmitted[21]. The Serial output in Arduino results displaying Connection fail to Server as shown in Figure 3-24.

Figure 3-24: Snapshot of failed MySQL connection in Output Monitor while running Arduino Code

There are number of reasons for Connection fail such as (a) Wrong Server_Address (b) Wrong Username and Password (c) Raspberry Pi port '3306' not open (d) Wrong tables and columns. Out of these, port '3306' is the major problem to solve and connect.[21] This can be identified by going to LXTerminal and typing following command as below:

mysql -u root -p -h IpAddresssOfRaspberryPi --port 3306

Then we need to enter the Raspberry Pi password and below we can see whether the above command connects to MySQL Database Server or not. However, in my case it does not access to MySQL Database Server and indicates ERROR as shown in Figure 3-25:

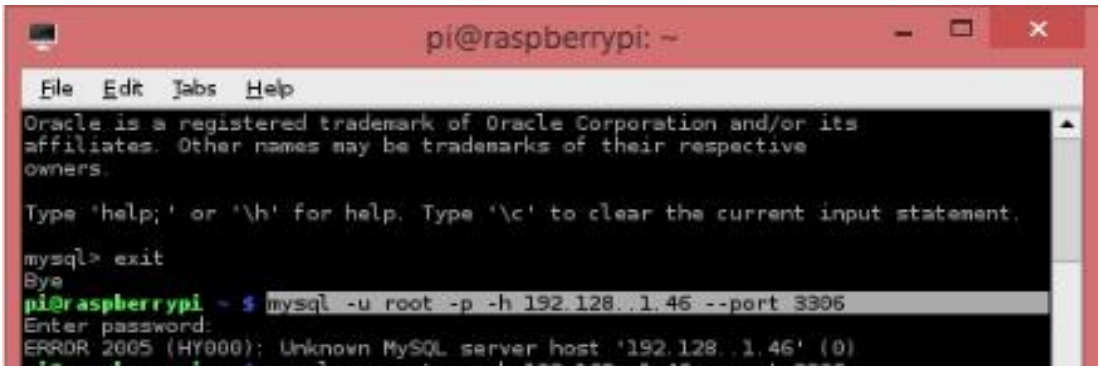


Figure 3-25: Snapshot of failed connection to MySQL Database Server in Raspberry Pi from LXTerminal

This problem can be solved by typing following commands in LXTerminal as below. [21].

Step1: **chown pi /etc/mysql/my.cnf**
Step2: **nano /etc/mysql/my.cnf**
Step3: **mysql -u root -p**
Enter password:

Then the commands is re-entered as earlier to see if it connects to MySQL, then it really worked and connected to MySQL Database Server.

Then it displayed as below in LXTerminal:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 47

Server version: 5.5.43-0+deb7u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> **GRANT ALL PRIVILEGES ON * . * TO 'root'@'%' IDENTIFIED BY 'password';**

Query OK, 0 rows affected (0.00 sec)

In this way, the problem was solved for connecting to MySQL server in Raspberry Pi.

3.1.6 Raspberry Pi Camera Module

Raspberry Pi Camera Module is a camera module attached to interface in Raspberry Pi to take still photographs as well as high-definition videos. There are in numerous function in this Raspberry Pi Camera Module. This camera module weighs only 3 grams and measures 25*20*9 mm and comes with 15cm ribbon cable. This camera sensor is 5MP and produces resolution up to 2592*1944 still images. It comes with video support of 1080p at 30fps,720p at 60fps [22]. The installation process of this camera module is very easy comparative to other camera module [23]. The installation procedure is explained as below:

I. First of all open Raspberry Pi Camera Module and attach the silver connectors of cable of Raspberry Pi Camera Module into Raspberry Pi. The connector lies between Ethernet and HDMI ports, with silver connectors facing the HDMI port as shown in Figure 3-26:

Figure 3-26: Snapshot of Raspberry Pi camera module connected to Raspberry Pi

II. Then open LXTerminal and enter the following commands as below:

sudo apt-get update (update Raspberry Pi device)
sudo apt-get upgrade -y (upgrade Raspberry Pi device)

III. Then open the raspi-configuration to enable camera module in Raspberry Pi. Type following command on LXTerminal.

sudo raspi-config

Here, a dialogue box will appear as shown in Figure 3-27 and click under Enable Camera to enable the Camera settings in Raspberry Pi and finish it.

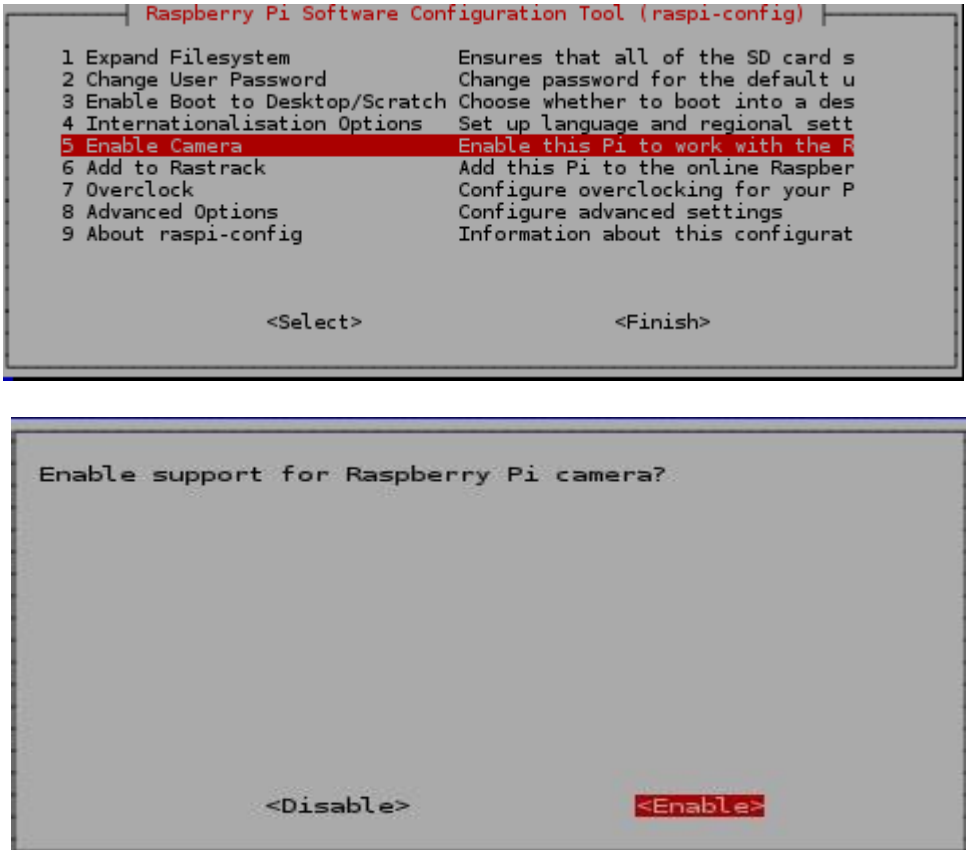


Figure 3-27: Snapshot of Raspberry Pi Camera Module installation in Raspberry Pi

IV. Now, we are ready for Raspberry Pi Camera Module use. As indicated above, Raspberry Pi Camera Module can be used for taking still snaps and videos of different quality. To take picture image, we type the following command on LXTerminal as below.
This is the basic command line application for capturing image with camera module.

raspistill -o image.jpg (Here o refers to output and image refers to name of image)

V. Similarly, Raspberry Pi Camera module can be used for recording videos of different quality and size. The basic command line application for recording videos with Camera module is shown in below which is later typed on LTX terminal to run.

raspivid -o -t 10000 songvideo.h264 (Here o refers to output, t refers to video recording time and songvideo refers to name of video)

VI. Similarly, Raspberry Pi Camera module can be used for live streaming of videos of different quality, resolution, frame per sec. The basic command line application for

live streaming of video from Raspberry Pi Camera module is shown below which is later typed in LXTerminal to execute.[24]

raspivid -vf -hf -o --t 0 -n -w 600 -h 400 -fps 12 | cvlc -vvv stream:///dev/stdin --sout'#standard{access=http,mux=ts,dst=:8080/}':demux=h264
where, -o is output, -n is nonpreview, -w is width, -h is height,-fps is framepersec

Or, the above command can be written in script file so that it becomes easy to run simple command on LXTerminal and begin live streaming of video from camera. Let the script file name is stream. Enter following command below to create script file in LXTerminal.[28]

Part1(Procedure code for writing script file for video streaming)
Step1: **nano ~/stream**
Step2:

```
#!/bin/bash
while :
do
  raspivid -o - -t 0 -n -w 600 -h 400 -fps 12 | cvlc -vvv stream:///dev/stdin--
sout'#standard{access=http,mux=ts,dst=:8080/}' :demux=h264
  sleep 1
  done
```

Step3: **chmod +x ~/stream**

Part2(Begin and End stream using LXTerminal)

Step4: **~/stream (for begin stream)**
Step5: **Cntrl+C (to end stream)**

VII. Then open VLC media player in computer and under Media-Open Network Stream, type following commands http://IPAddressOfRaspberryPi:8080 to execute and run live streaming videos in computer.
OR, we can also view by browsing on browser. Just create a web page and save the following below commands on the web page. Then run the web page to view the streaming videos from your browser[24]. The list of codes to save in webpage is shown below:

```
<!DOCTYPE html>
<html><body>
<OBJECT classid="clsid:9BE31822-FDAD-461B-AD51-BE1D1C159921"
codebase="http://downloads.videolan.org/pub/videolan/vlc/latest/win32/axvlc.cab
"
 width="600" height="400" id="vlc" events="True">
 <param name="Src" value="http:// IpAddressOfRaspberryPi:8080/" />
 <param name="ShowDisplay" value="True" />
 <param name="AutoLoop" value="False" />
 <param name="AutoPlay" value="True" />
```

<embed id="vlcEmb" type="application/x-google-vlc-plugin"
version="VideoLAN.VLCPlugin.2" autoplay="yes" loop="no" width="640"
height="480"
target="http://IpAddressOfRaspberryPi:8080/" ></embed>
</OBJECT>
</html></body>

Then, view the live streaming from browser in your PC.

VIII. In normal situation the LED glow red on Raspberry Pi Camera module indicating that the Camera is On. There are cases when we need to disable the red LED on Raspberry Pi Camera module for several reasons like reflection problem, power consumption, security reasons and so on. To solve it, enter following commands to edit config.txt file on LXTerminal.[27]

sudo nano /boot/config.txt
disable_camera_led=1 (add this line at the last line of config.txt file)

Then, save it by pressing Cntrl+O followed by enter and Cntrl+X to exit. After that reboot the system by entering **sudo reboot** on LXTerminal. Incase, we want to enable red light, just edit disable_camera_led=0 in config.txt file.

IX. Important thing to know: The camera is operating til the power plug is On. When the Raspberry Pi device is re-powered on power plug, command to operate camera must be entered again i.e enter **~/stream** on LXTerminal and the camera begins to stream live video. Also important thing to note is that, while changing IP address of Raspberry Pi, camera IP address also must be changed. This change is done by opening php page of camera and changing IP address according to IP address of Raspberry Pi device.

3.1.7 Sketch Design for Home Automation

The overview diagram of HomeAutomation is important to get quick idea of how Raspberry Pi, Arduino and sensor/devices work in connection to achieve the tasks of monitoring, logging and controlling device. The sketch design for HomeAutomation is shown in Figure 3-28.

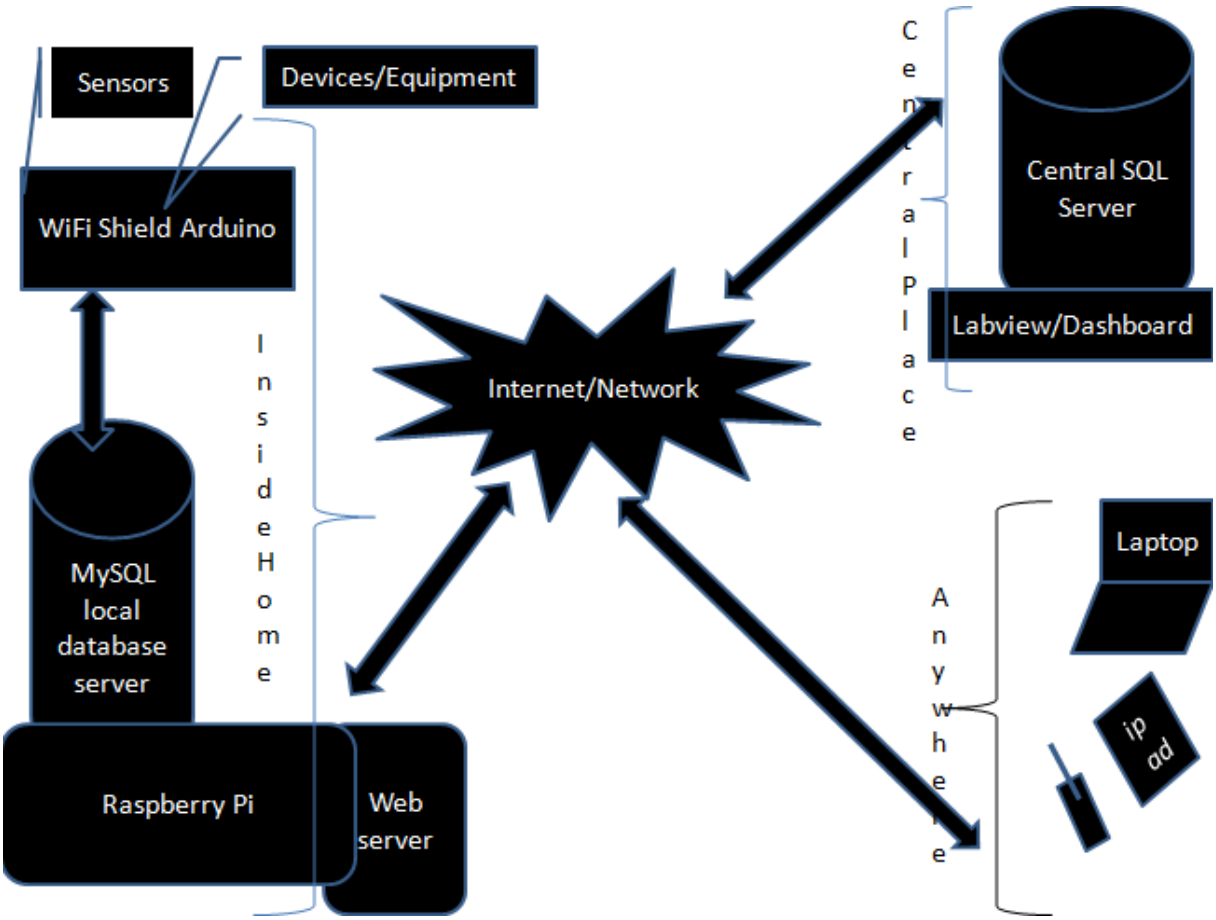


Figure 3-28: Block Diagram of Model for HomeAutomation

HomeAutomation deals with automation of devices inside home. All the sensors and devices are within the peripheral of Home Area. As shown in Figure 3-28, the block diagram is divided into 3 main parts: Home zone, Central Server and Monitoring Devices. Inside Home, we have main Raspberry Pi device and Arduino device working in coordinate with sensors/devices for receiving sensor information and controlling devices accordingly. Services like MySQL Database Server and WebServer in Raspberry Pi provided access for Database management and viewing created webpage in browser.

Similarly, the Central Server reflects the Server outside Home, where all the necessary data of Home are saved and managed. The Lab view features is added to view these data in Dashboard application so that through use of Tabloid and Iphone, home appliances and information can be monitored.

The next is monitoring home from electronic devices like Laptop, Ipad etc. By using web service, the information is monitored on browser screen of these devices and information can be known from anywhere and anytime.

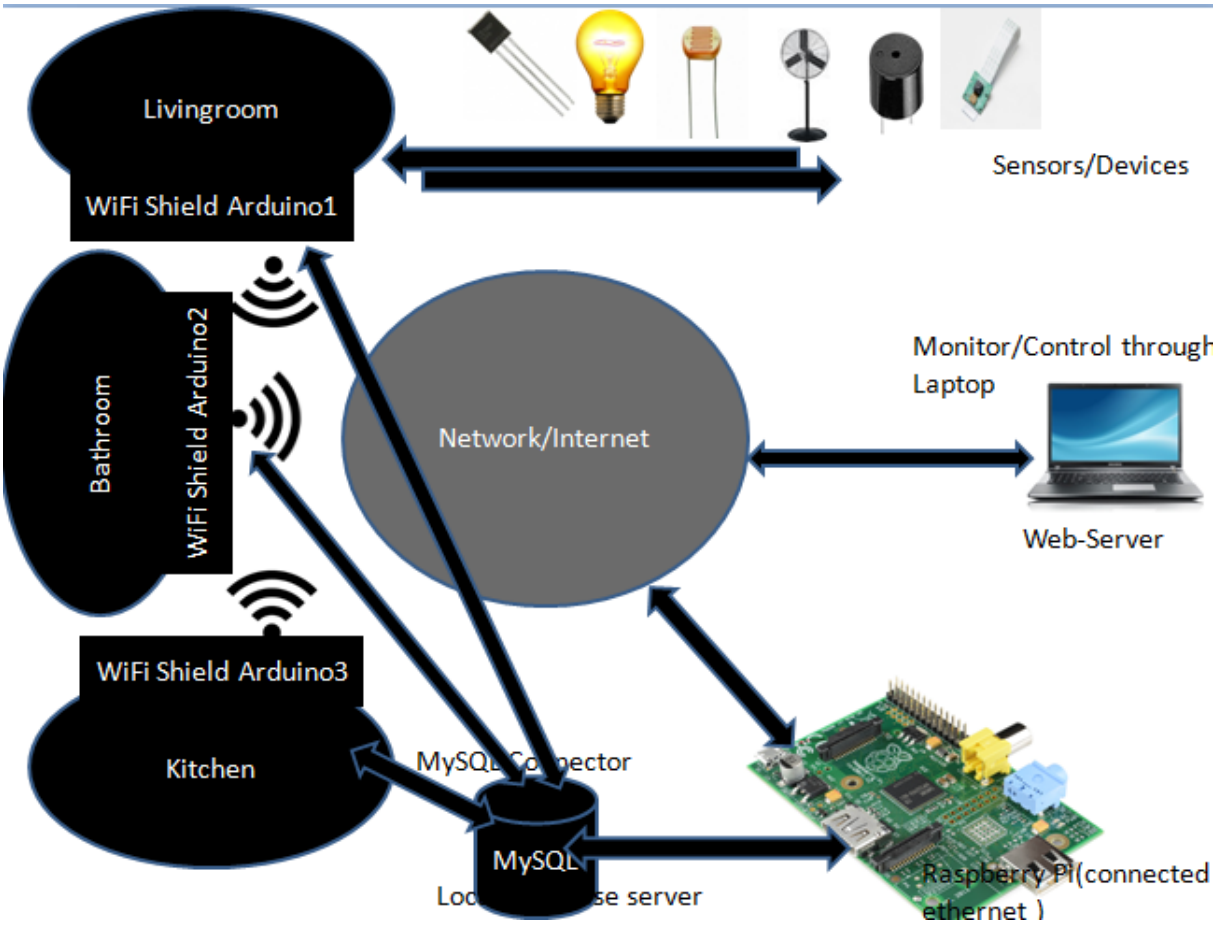


Figure 3-29: Block Diagram of Multiple Rooms & their connection to Raspberry Pi in Home

As shown in sketch Figure 3-29, we have different rooms in House(Living room, Bed room, Kitchen etc). And there are different appliances in each room(Air heater, Fan, Lamp etc) that needs monitoring and control. Hence, we must allocate one Wi-Fi Arduino Shield in each room to connect appliances. The Wi-Fi Arduino Board connects to Raspberry Pi device through network connection.

Since, Arduino is Wi-Fi shield and connected to Internet/Network connection, the information from sensors are send over network to Raspberry Pi device. Arduino Wi-Fi Board and Raspberry Pi do not have direct physical connection by any means of wire. They transmit data through network connection.

As shown, Raspberry Pi has MySQL Server and Web Server running on it. These are two basic and important applications of our Home Automation. Through MySQL server, database are created and the information from Arduino Wi-Fi Board are managed. Similarly the other most important thing Web Server contains html/php files that connects database information

and other information to users via web page. The user can simply connect to that php files and open in their browser in Tabloid or PC and then monitor/Control the Home appliances.

4 Results

The HomeAutomation task is to monitor, control and log the data from respective sensors & devices.

As indicated in Procedures, the first major implementation is done with database model for linking various parts. The web design is of utmost important in HomeAutomation systems as it displays the output in structured form for user interface. By entering IP_Address_Of_Raspberry_Pi/HomeAutomation on browser, we have access to web page of HomeAutomation that displays list of sensors/devices we keen on to monitor.

The home page of HomeAutomation in browser is as shown in Figure 4-1 that contains Login, Monitoring, Services and Contact section. Under Login section, we enter username and password to access inside next web page for managing tag, tagtype, devices and so on.

Figure 4-1: Snapshot of home page for HomeAutomation

The main important part in HomeAutomation is Monitoring section. In Monitoring section, we find a number of lists of sensor/devices we like to monitor on. These lists include Temperature Monitoring, Light Monitoring, Fan Monitoring, Alarm Monitoring and Camera Monitoring. Figure 4-2 shows the basic list of devices and sensors we like to monitor on.

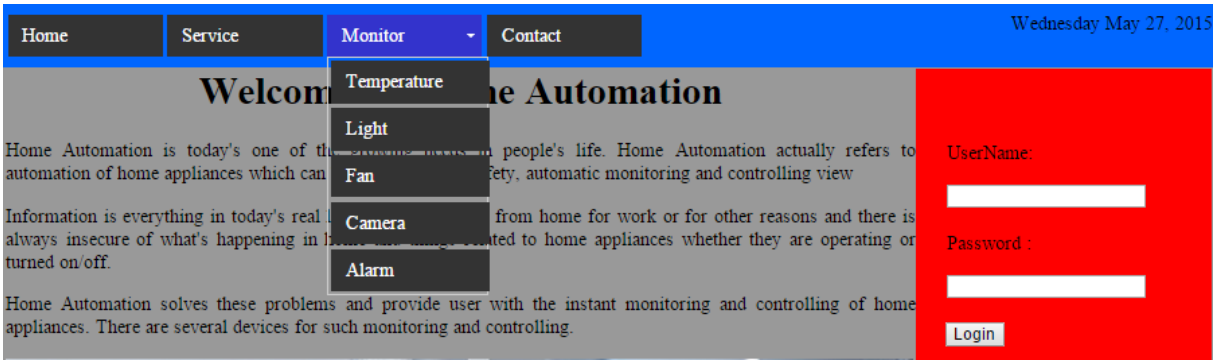
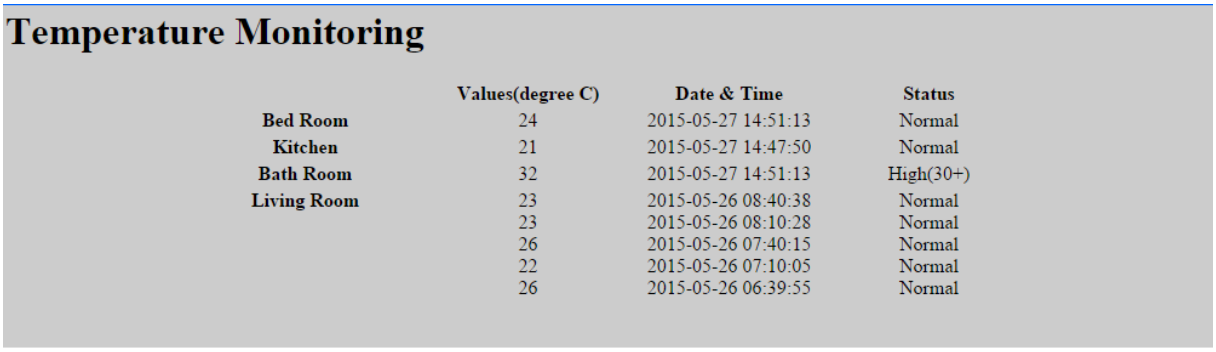


Figure 4-2:Snapshot of list of sensor/devices in Monitoring section

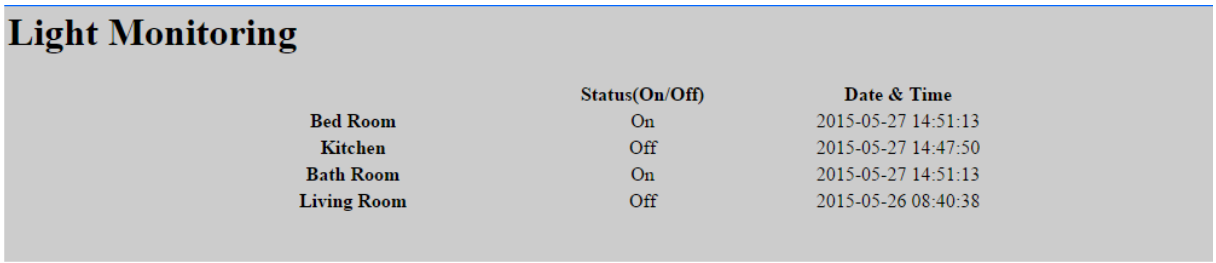


Temperature Monitoring

	Values(degree C)	Date & Time	Status
Bed Room	24	2015-05-27 14:51:13	Normal
Kitchen	21	2015-05-27 14:47:50	Normal
Bath Room	32	2015-05-27 14:51:13	High(30+)
Living Room	23	2015-05-26 08:40:38	Normal
	23	2015-05-26 08:10:28	Normal
	26	2015-05-26 07:40:15	Normal
	22	2015-05-26 07:10:05	Normal
	26	2015-05-26 06:39:55	Normal

Figure 4-3: Snapshot of Temperature Monitoring

The first thing we measure is temperature inside room. In Temperature Monitoring, we monitor the temperature inside all rooms in the house. As there will be number of rooms in house, we need temperature sensor in each room. Figure 4-3 shows temperature data information dragged in web page to have clear idea of temperature inside each room. Bedroom shows temperature of 24, while Kitchen shows 21, Bathroom shows 32 and Living room shows 23 degree Celsius at latest time. Also there is status to indicate High/Low/Normal temperature so that user will have information if there is high or low temperature along with the value.



Light Monitoring

	Status(On/Off)	Date & Time
Bed Room	On	2015-05-27 14:51:13
Kitchen	Off	2015-05-27 14:47:50
Bath Room	On	2015-05-27 14:51:13
Living Room	Off	2015-05-26 08:40:38

Figure 4-4: Snapshot of Light Monitoring

Similar is the case with Light Monitoring. Depending on LDR sensor sensitivity, light inside room will be controlled automatically. If enough light falls on LDR sensor, it will have low resistance, then light inside room will be turned Off and vice-versa. Also, threshold limit and location of LDR affects Light operation. Threshold limit is defined for each room. Depending on threshold limit, decisions are made on when to turn light On and Off. We are monitoring lights in different rooms. As shown in Figure 4-4, Bedroom & Bathroom light is On while Kitchen & Living room light is Off.

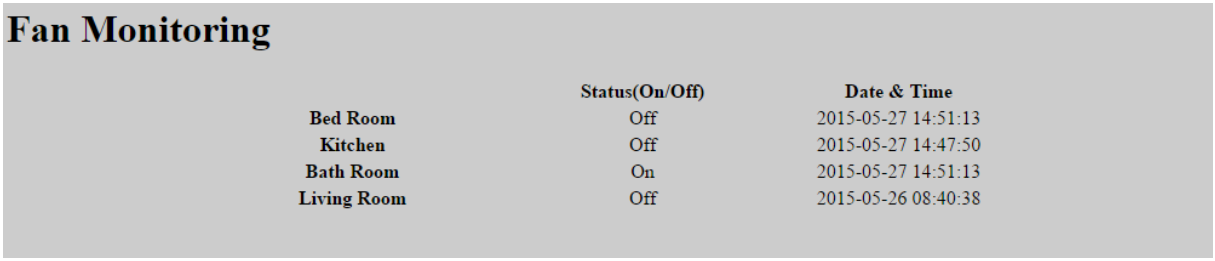


Figure 4-5: Snapshot of Fan Monitoring

Next is Fan Monitoring as shown in Figure 4-5, where the fan inside room is monitored and controlled automatically depending on temperature inside room. If the temperature sensor reads high temperature than normal value, then fan will automatically turn On and if temperature is within normal then fan is turned Off and these information are updated in database and displayed in browser for user information. As shown in Figure 4-5, fan of Bedroom, Kitchen and Living room is Off while fan inside Bathroom is On since Bathroom records a high temperature.

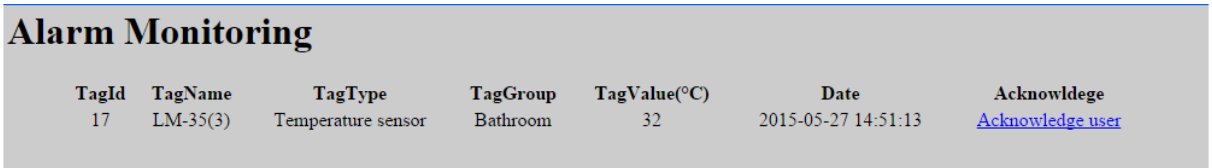
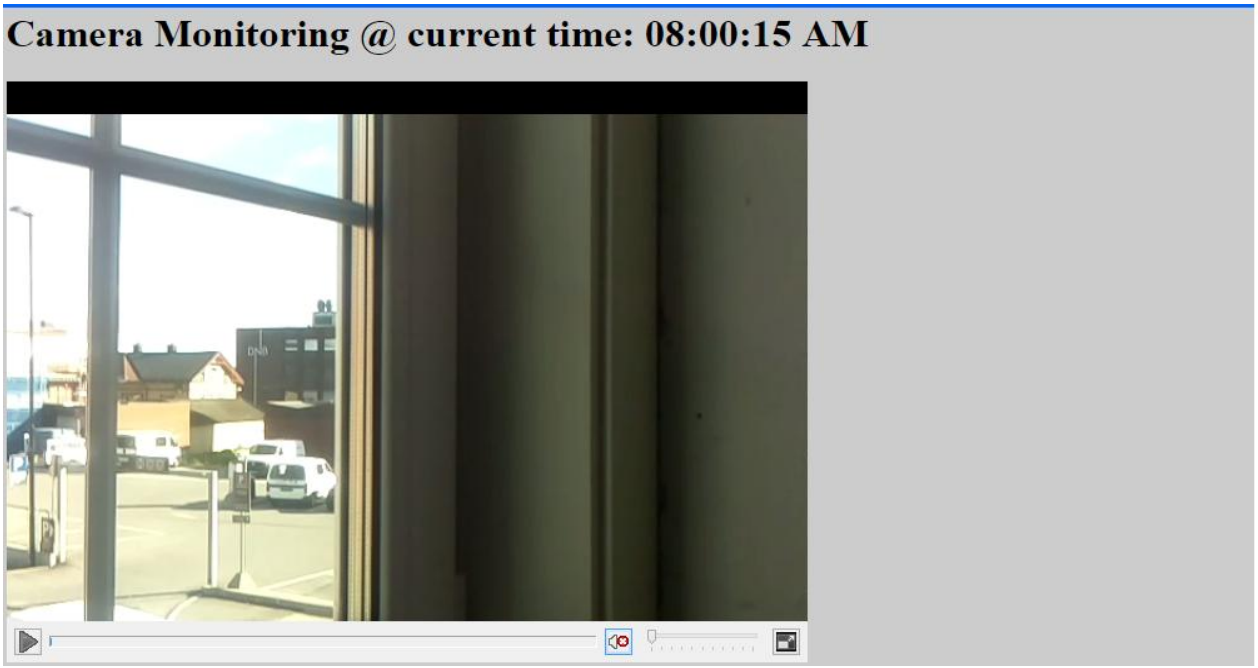
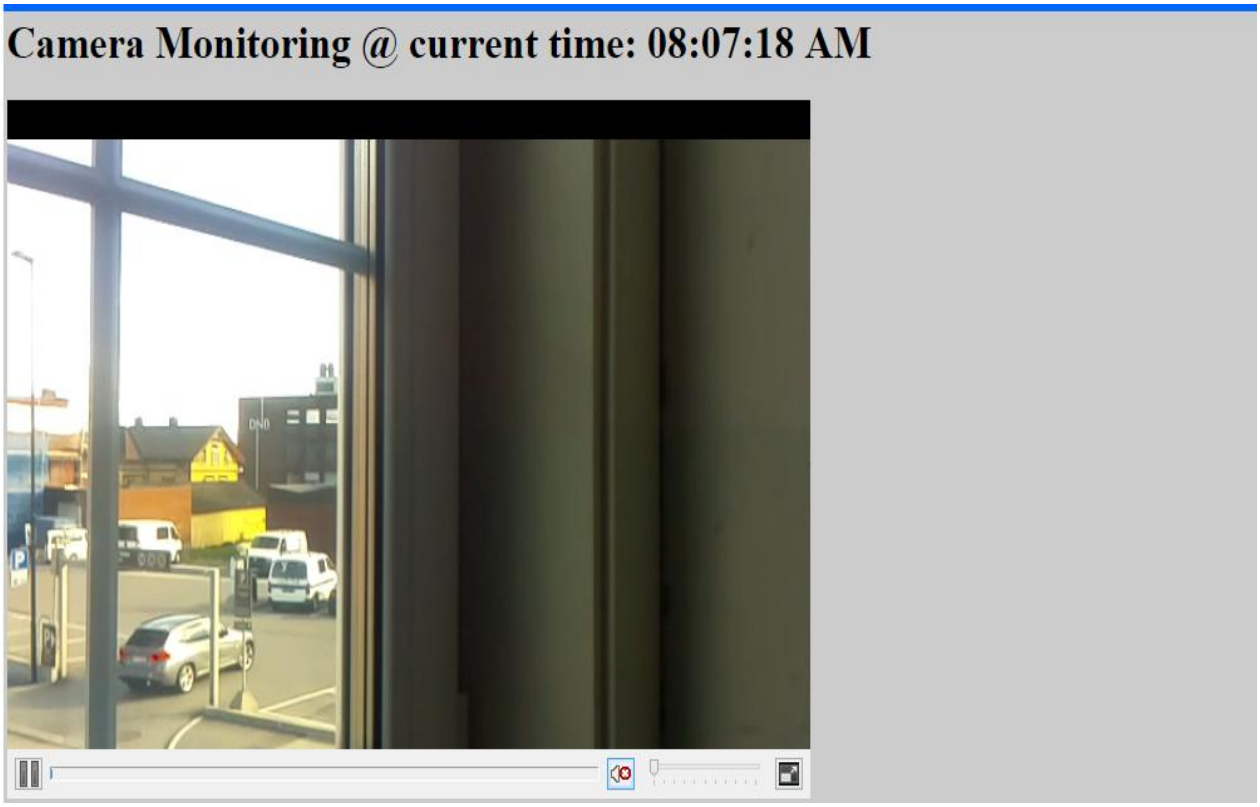


Figure 4-6: Snapshot of Alarm Monitoring

With Alarm Monitoring, we monitor if the temperature inside room is high or low. If the temperature reading shows high or low value, then alarm will go On. The alarm is an essential part in HomeAutomation as it secures and provide alert if high or low temperature is recorded

and user can respond by Acknowledging it. Figure 4-6 shows the high temperature recorded in Bathroom.





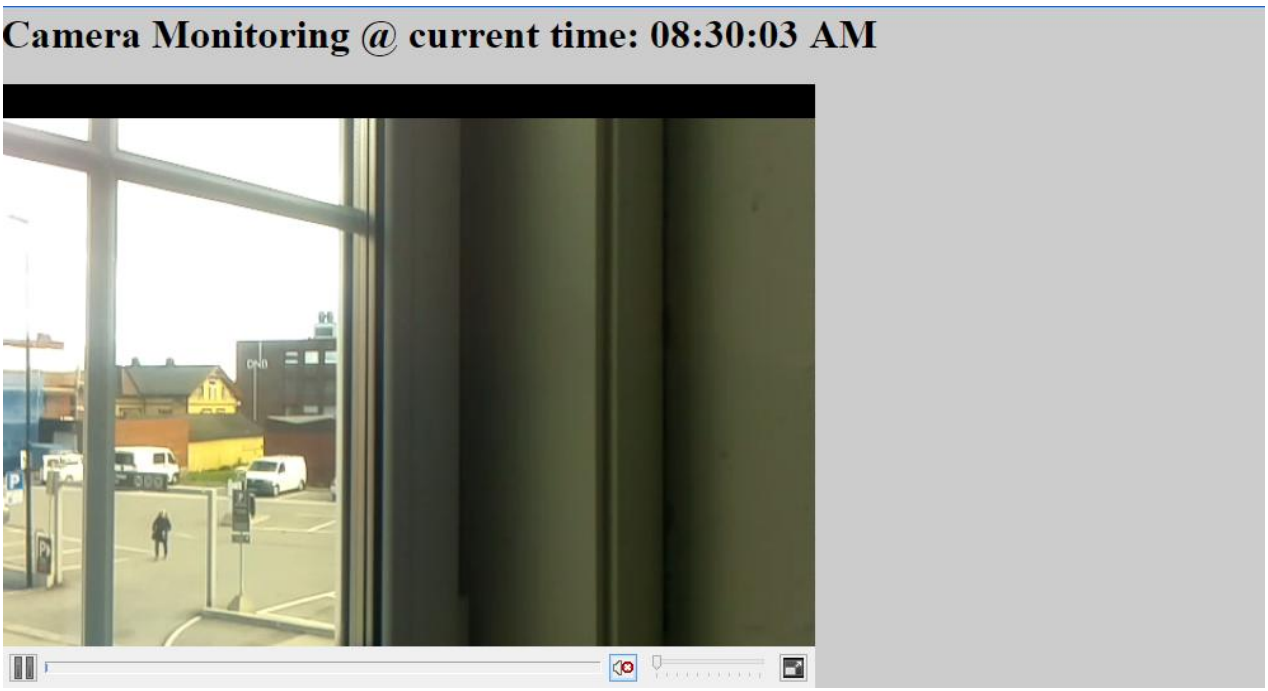


Figure 4-7: Some Snap shots of Camera Monitoring outside house

The next is Camera Monitoring where we monitor video surveillance. Raspberry Pi Camera module can be kept above main door of house that one can be updated with what is going outside house. The camera can also be used for Baby monitoring if baby is left alone in house. Camera Monitoring displays live streaming video, with the help of which user can be updated with momentary action going in their house. Figure 4-7 shows some snapshots of Camera Monitoring outside house.

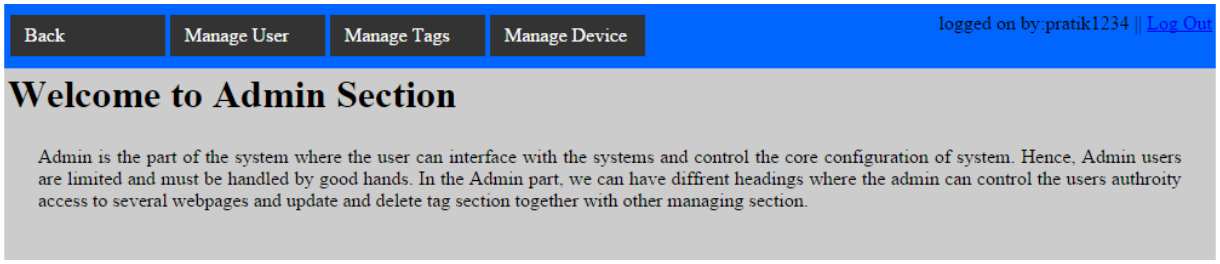


Figure 4-8: Snapshot of Admin section for HomeAutomation system

The very next important thing in HomeAutomation is managing sort of things like Managing Users, Managing Tags & Devices etc. Since, as a user of this HomeAutomation service, one

likes to change settings according to his/her preference. To enter into Manage or Admin section we have to Login from homepage of the website by using username and password.

The Login part is important for managing some of the important features in HomeAutomation. After one is logged as user, the webpage for Admin opens. Login is required to enter in Admin web page as wrong use of such Managing settings can make Raspberry Pi database system crash. The Admin section is for managing users, tags, devices and so on. As shown in Figure 4-8 Admin section has link to Manage User, Manage Tags, Manage Device.

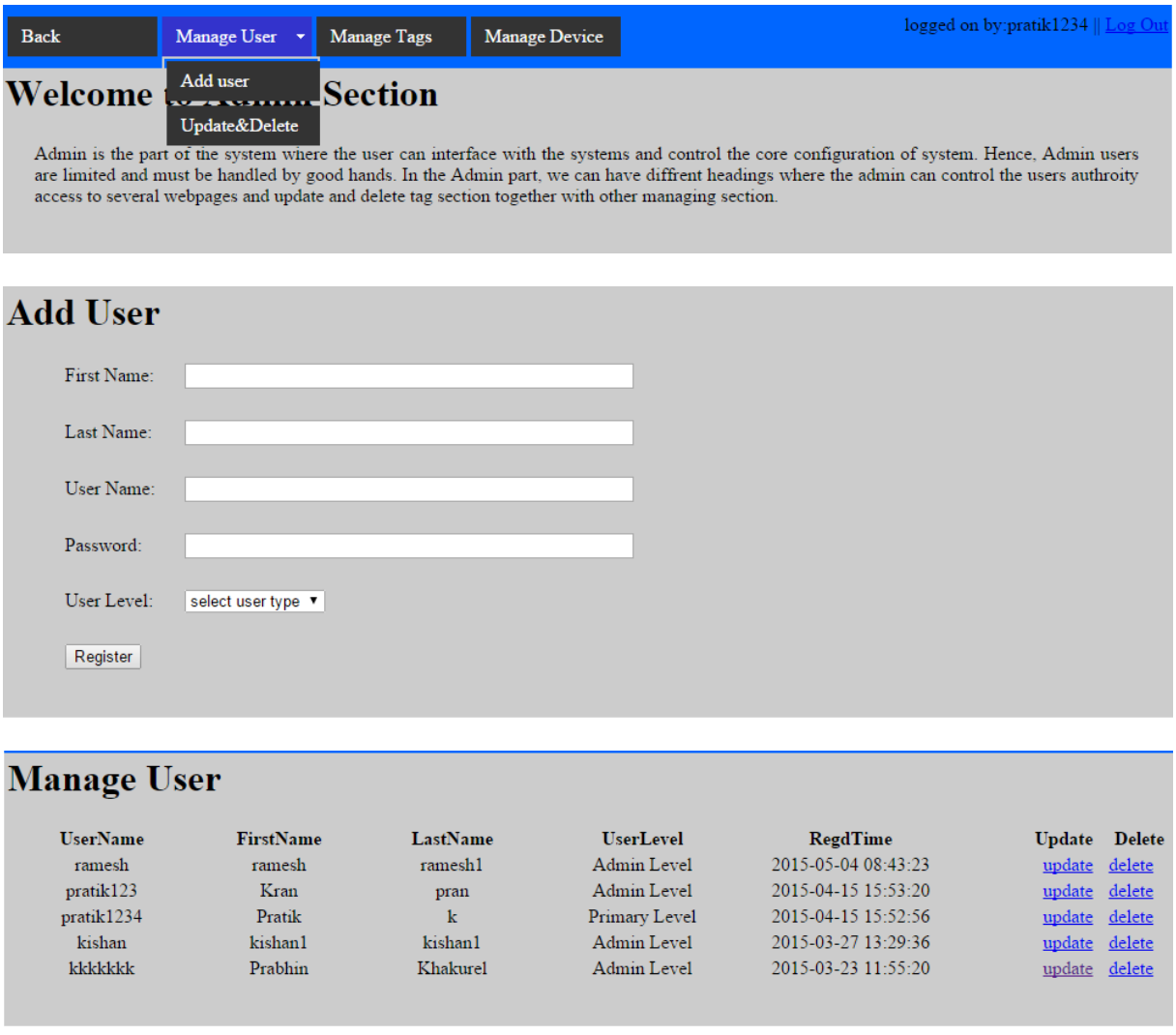


Figure 4-9: Snapshot of Manage User (Add, Update,Delete)

The first is Managing Users. Managing Users allows to add, update and delete users. One can add user, update and delete them. Since, there can be more than one users in home and they need to monitor and manage the HomeAutomation, it is required that they can be able to

create such users and monitor the sensor's data and devices. As shown in Figure 4-9, users can be added by filing & saving the specified columns and later on they can be updated for some changes or deleted if such users are inactive or unwanted.

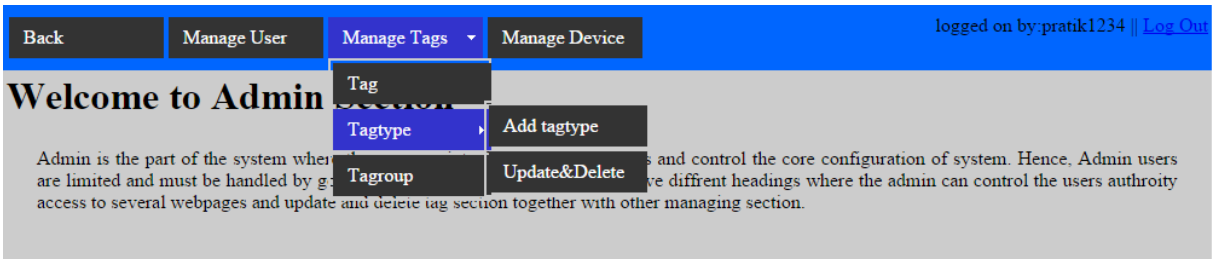


Figure 4-10: Admin showing Tag,Tagtype and Tagroup

Figure 4-11: Snapshot of Manage Taggroup(Add,Update&Delete)

Here in Admin Section as shown in Figure 4-10, Manage Tags allows to add, update and delete tag, taggroup and tagtype.

Under Taggroup section for example in a house, we can add group name such as Kitchen, Bathroom etc etc as group to indicate location of sensors/devices. Later on, changes can be made with update selection. Similarly, delete option allows to delete such group and create new one according to user desire. The Figure 4-11 lists some of the added groups in HomeAutomation.

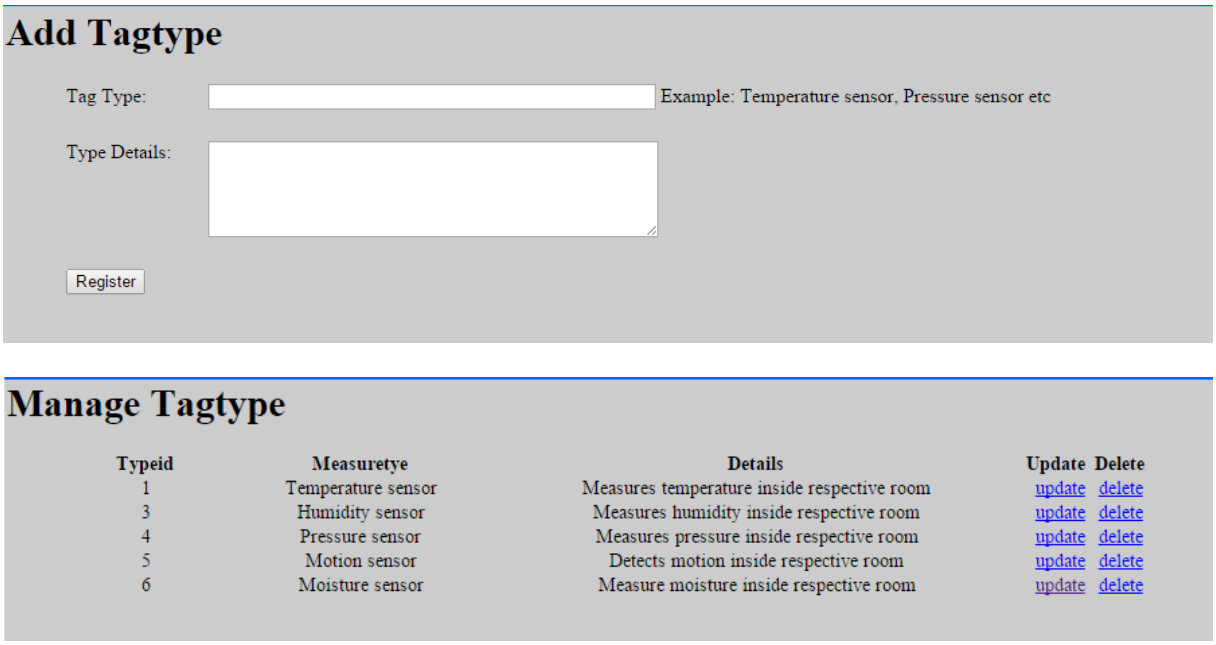
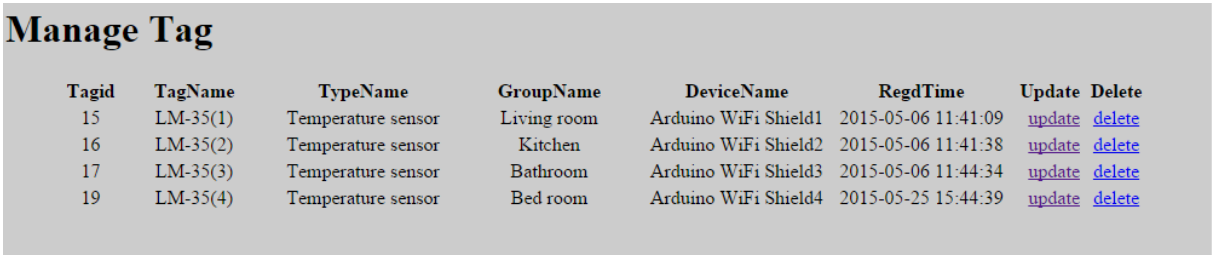


Figure 4-12: Manage Tagtype(Add,Update&Delete)

Tagtype allows us to add type of tag. In Tagtype section for example in house, we can add sensor types i.e Pressure sensor, Temperature sensor, Humidity sensor etc as common name of sensors so that user can have general idea of what type of sensor is used and what it measures.

As shown in Figure 4-12, a number of sensor types is added in HomeAutomation. However, only Temperature sensor type is used in HomeAutomation.

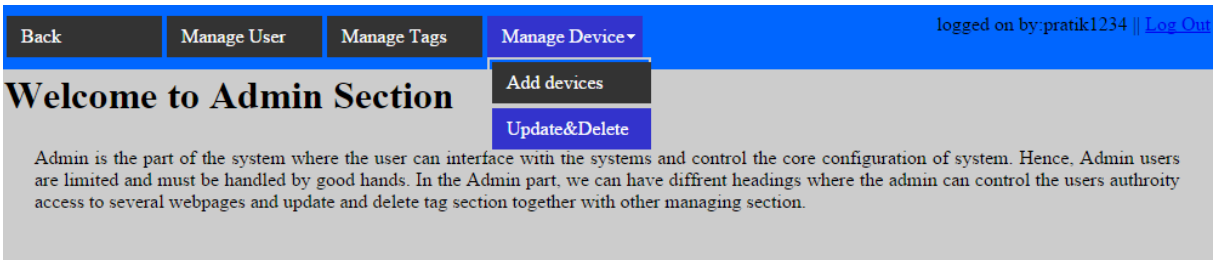


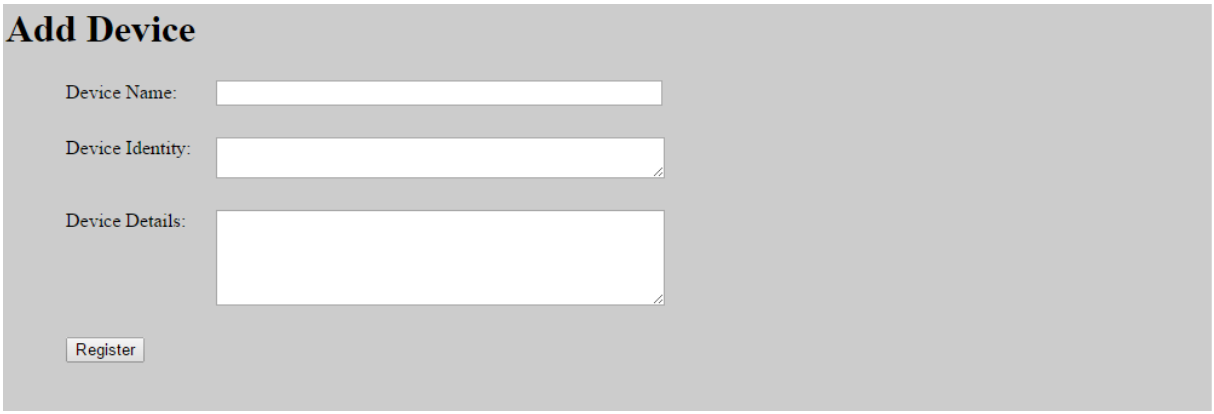
Manage Tag

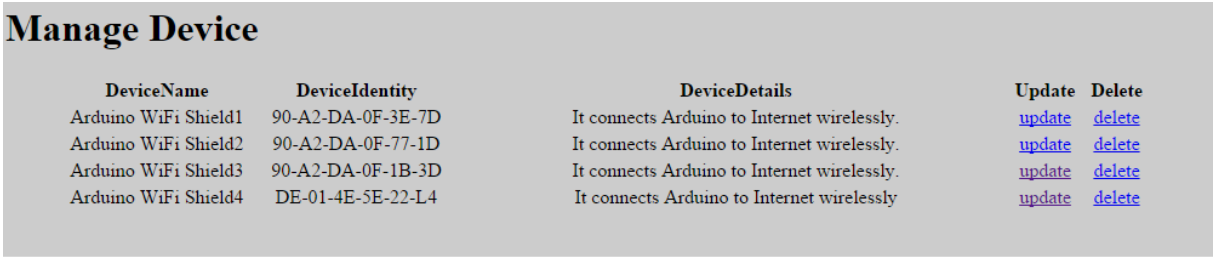
Tagid	TagName	TypeName	GroupName	DeviceName	RegdTime	Update	Delete
15	LM-35(1)	Temperature sensor	Living room	Arduino WiFi Shield1	2015-05-06 11:41:09	update	delete
16	LM-35(2)	Temperature sensor	Kitchen	Arduino WiFi Shield2	2015-05-06 11:41:38	update	delete
17	LM-35(3)	Temperature sensor	Bathroom	Arduino WiFi Shield3	2015-05-06 11:44:34	update	delete
19	LM-35(4)	Temperature sensor	Bed room	Arduino WiFi Shield4	2015-05-25 15:44:39	update	delete

Figure 4-13: Snapshot of Manage Tag(Add, Update,Delete)

Mange Tag allows us to manage tag. In Tag section for example, we can add sensors but in particular form as TMP36, PT-100 and also link with Tagtype and Tagroup to indicate the sensors type and location of where the sensor lies. As shown in Figure 4-13, we have option to choose sensor type in Tagtype and also option of sensor location in Tagroup along with option to choose kind of devices in Devices option.







Manage Device

DeviceName	DeviceIdentity	DeviceDetails	Update	Delete
Arduino WiFi Shield1	90-A2-DA-0F-3E-7D	It connects Arduino to Internet wirelessly.	update	delete
Arduino WiFi Shield2	90-A2-DA-0F-77-1D	It connects Arduino to Internet wirelessly.	update	delete
Arduino WiFi Shield3	90-A2-DA-0F-1B-3D	It connects Arduino to Internet wirelessly.	update	delete
Arduino WiFi Shield4	DE-01-4E-5E-22-L4	It connects Arduino to Internet wirelessly	update	delete

Figure 4-14: Snapshot of Manage Devices(Add,Update&Delete)

Similarly Manage Device allows us to add, update and delete the devices used in our project. As we can have number of Arduino Wi-Fi Shield working together in monitoring and controlling process. It is required that adding device information allows users to know which devices are used and with other supplementary information like location and sensors attached to that device. The snapshots of Devices webpage is shown in

Figure 4-14.

PID Controller

PID controller is used in controlling room heater in home. It uses general feedback mechanism to control heater in room.

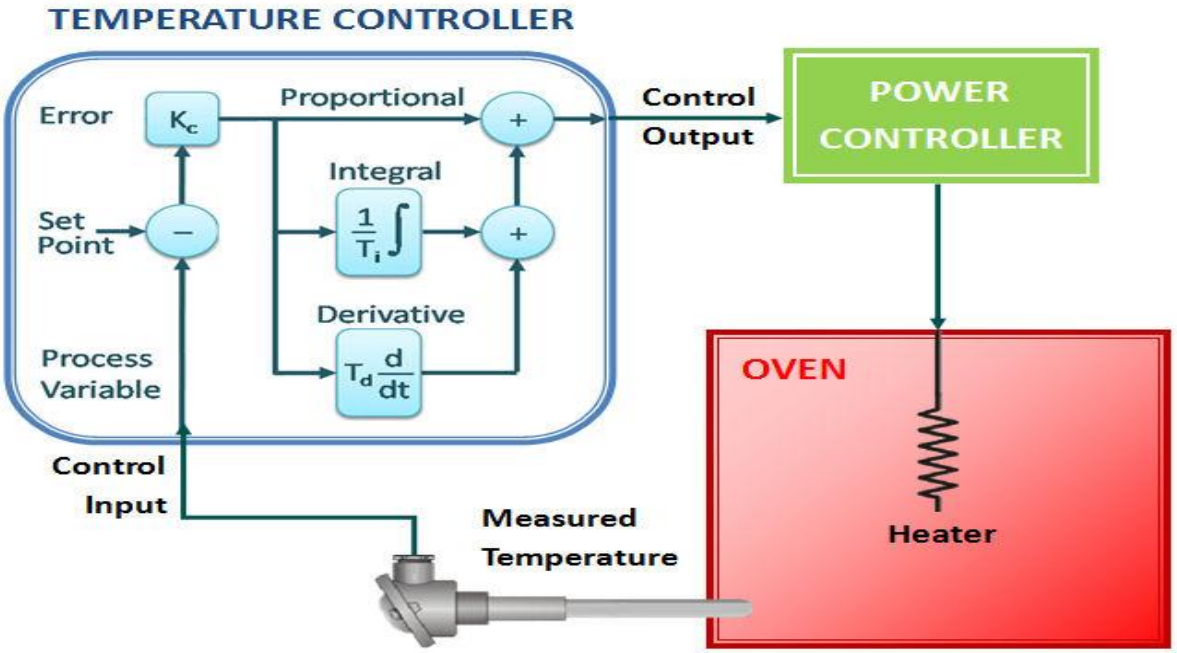


Figure 4-15: Temperature controlled heater[25]

As the Figure 4-16 shows the scenario of heater controlled according to temperature measured by sensor and according to set point. The Temperature sensor LM-35 collects temperature information from Heater and feeds as input to Process. The PID controller detects error between input and reference value and is tuned to that degree where any deflection is minimized and output is almost constant. We have equation as shown in Equation 1 to describe more on how PID controller works to give constant output.

Equation 1: Mathematical expression for PID controller loop

$$u(t) = K_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{d}{dt} e(t)$$

$$K_p = Proportinal\ gain$$

$$k_i = Integral\ gain$$

$$k_d = Derivative\ gain$$

e= Error, t= time of instantaneous

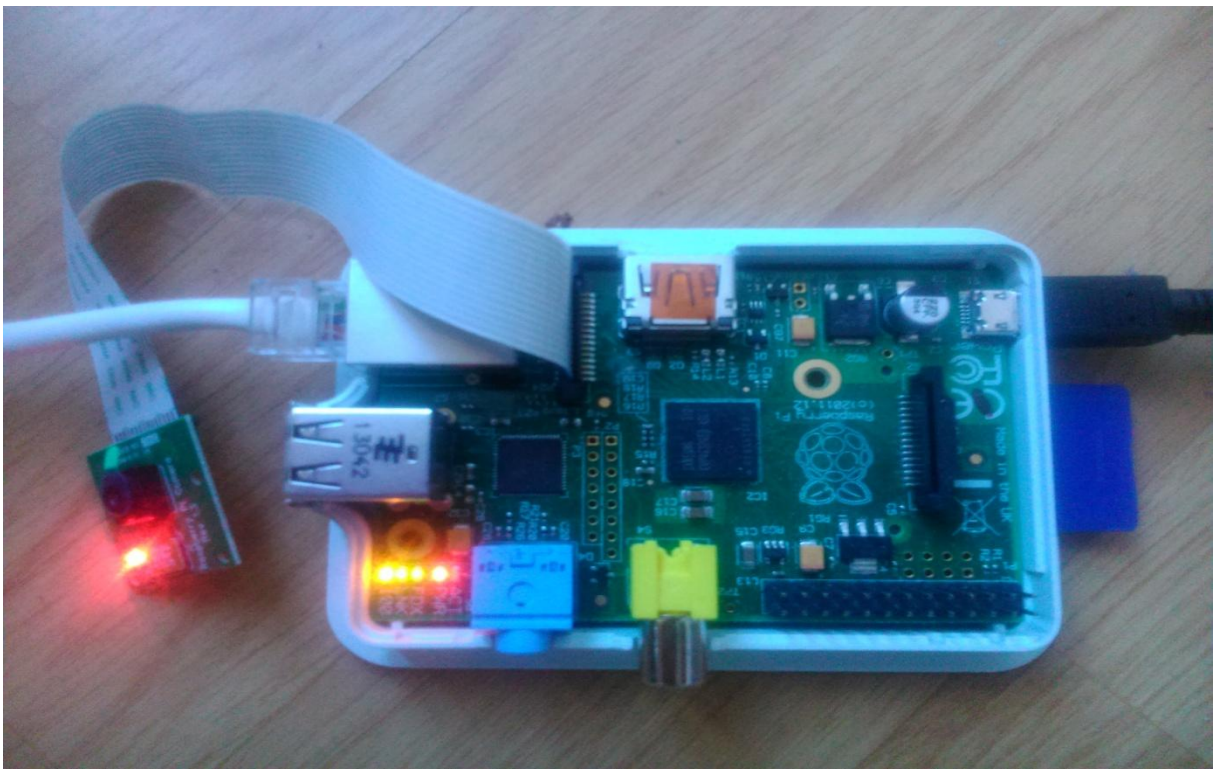


Figure 4-16: Image of Raspberry Pi with Camera

The Wi-Fi shield Arduino, Raspberry Pi and sensors used in HomeAutomation tasks is shown in Figure 4-16, *Figure 4-17*, Figure 4-18. As shown in Figure 4-16, the Raspberry Pi device is powered and connected to Ethernet cable along with Raspberry Pi Camera module. As Raspberry Pi is connected to network, the web files stored in WebServer of Raspberry Pi can be accessed by browsing IP_Address_of_RaspberryPi along with the file name.

The Camera module in Raspberry Pi can be used for monitoring a place in house. The Raspberry Pi camera module streams live video. Depending on where the camera needs to be monitored, live streaming video of that place can be monitored live 24*7.

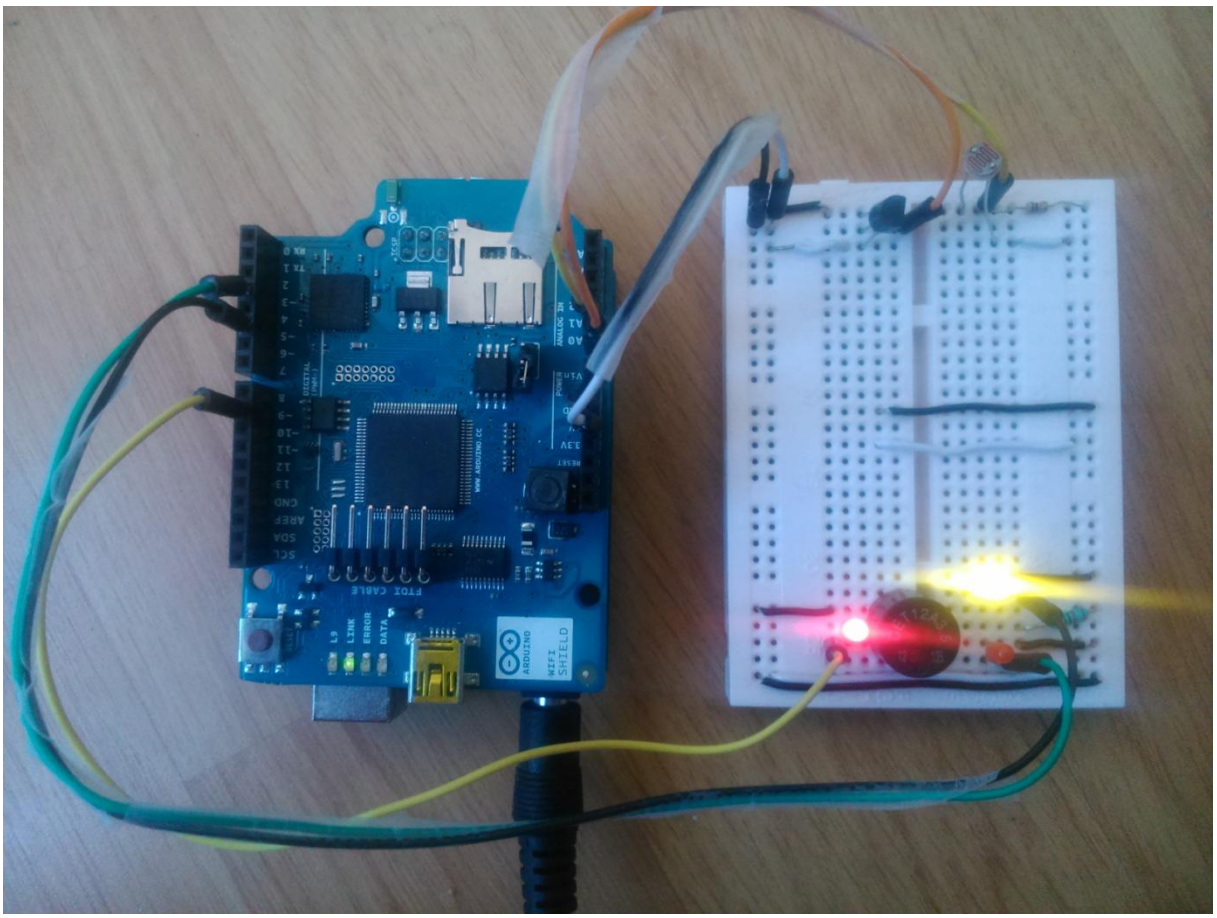


Figure 4-17; Snapshot of Wi-Fi Arduino connected to sensors & devices.

Also the Figure 4-17 shows the breadboard consists of Temperature sensor TMP-35, LDR sensor, Alarm, Different Led connected to Wi-Fi shield Arduino. The TMP-35 sensor collects temperature data while LDR sensor collects light intensity data. Now we use these information from sensors and control the devices attached to Wi-Fi Arduino Shield.

The Fan and Alarm depends on temperature sensor for functioning. When the temperature records high temperature, the Fan device will go On from Off state. Similarly when the temperature detects very high temperature, an Alarm will go On indicating high temperature in the room.

Similarly, when the LDR records intensity of different level, the Light device will go On or Off. Usually in day when there is more brightness, LDR records high analog value and so this value is used for controlling light devices.

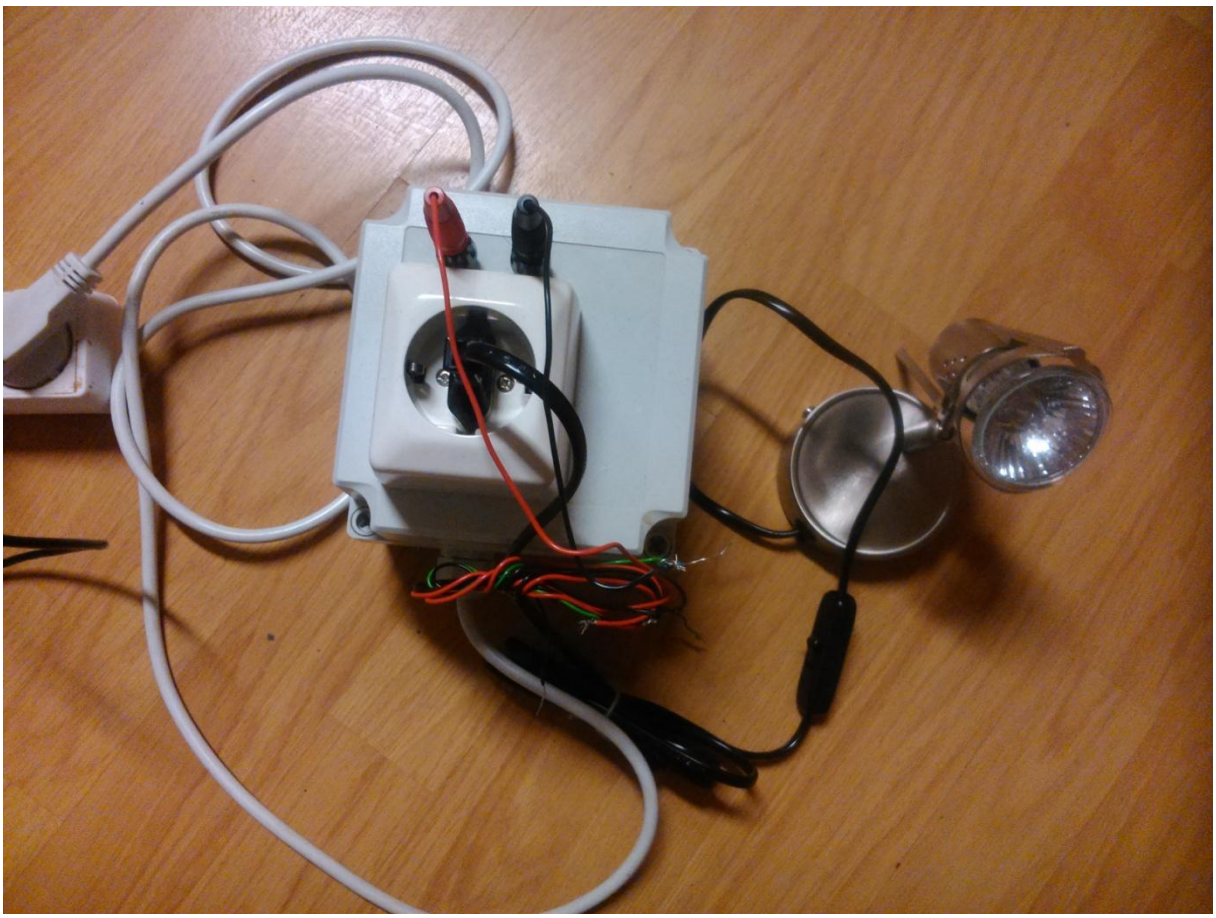


Figure 4-18: Snapshot of Light used in HomeAutomation

The Figure 4-18 shows light device, which is used in HomeAutomation. The figure shows light connected to control box which has one main supply and extra 2 wires(red & black) for controlling light through Arduino connectors. These red and black pins are connected in replacement of LED. When the Arduino device is powered, then the devices connected in replacement of LED output at pin 3 and pin 5 works accordingly as coded and then turn Light On/Off and fan On/Off accordingly.

Similarly, we can control the room heater in home. The PID Controller used in programming allows to control the room heater and maintain to specified temperature at all time so that room can be at constant temperature always.

5 Discussion

In HomeAutomation, the sensors were used for information and then control devices accordingly. Information were logged on in local database MySQL in Raspberry Pi device.

In HomeAutomation, Temperature Monitoring recorded temperature inside respective room, Light Monitoring recorded light status inside respective room, Fan Monitoring recorded fan status inside respective room, Alarm Monitoring recorded High Alarm status from respective tag and room. Similarly Camera shows live video monitoring.

As it was new completely on different matters like database, webpage design and so on. The desire time was much more than expected. Even for small programming and configuring in Raspberry Pi took lots of time and so could not do lots of things as assigned.

But in future, different features can be extended and make HomeAutomation reliable and safe and cheaper to use. In this HomeAutomation, there are not sufficient sensors to collect more information from house such as humidity measurement, door lock/unlock state and more on. These features can be added on next level. Similarly more devices can be monitored controlled such as Fan speed, Light Brightness etc.

Also Dashboard application can be built on with LabView software so that through gadgets like iphone, ipad, smartphones etc , the HomeAutomation can be viewed from these devices.

6 Conclusion

The Thesis is on HomeAutomation with aim of monitoring, logging and controlling home appliances. The primary focus is to make safe and secure the home we live in and have information on status of electronic devices.

As prime target of Thesis, some sensors were monitored and then used the information for controlling devices. Use of several software's like PhpMyAdmin, Arduino IDE, PHP, Xampp were integral part in completing the desired monitoring and controlling tasks.

Though it was completely new to certain devices and programming languages, it was exciting to play with sensors and devices. As due to problems like devices problem, information problem and as beginners to use of new software, some tasks were left out as there was not enough time to sort out the rest tasks. Major time were used in troubleshooting devices problem, coding, webpage design so only the main monitoring, logging and controlling of sensor/devices was done.

The thesis topic has lots of tasks which are challenging and exciting. In future, more improvements can be done in monitoring and controlling part. More sensors and devices can be connected and monitored on. Also, Camera module can be adjusted to different modes making more option in future work. Controlling manually from webpage can be made so that user can have more option in controlling device. Central server system could be developed and through Lab View based dashboard, information can be monitored through gadgets like Iphone, Ipad etc. These features can be added and improved in future work.

7 References

[1] Raspberry Pi.(2015). What is Raspberry Pi? Available: *http://www.raspberrypi.org*

[2] Wikipedia.(2015). Raspberry Pi. Available: *http://en.wikipedia.org/wiki/Raspberry_Pi*

[3] Wikipedia.(2015).Arduino. Available: *http ://en.wikipedia.org/wiki/Arduino*

[4] Wikipedia.(2015).MySQL. Available: *http://en.wikipedia.org/wiki/MySQL*

[5] Wikipedia.(2015). Apache HTTP Server. Available: *http://en.wikipedia.org/wiki/Apache_HTTP_Server*

[6] Wikipedia.(2015). Database. Available: *http://en.wikipedia.org/wiki/Database*

[7] Wikipedia.(2015). PID Controller. Available: http*://en.wikipedia.org/wiki/PID_controller*

[8] Wikipedia.(2015). PuTTY. Available: *http://en.wikipedia.org/wiki/PuTTY*

[9] Wikipedia.(2015). Xming. Available: *http://en.wikipedia.org/wiki/Xming*

[10] Andrew K.Dennis, Raspberry Pi Home Automation with Arduino, February 05,2013.

[11] Raspberry Pi Tutorials.(2015). Raspberry Pi Static IP Address. Available: *http://www.suntimebox.com/raspberry-pi-tutorial-course/week-3/day-5/*

[12] Raspberry Pi.(2015).Setting up an Apache Web Server on a Raspberry Pi. Available: *https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md*

[13] Raspipress.(2015). Tutorial-Install-MySQL-server-on-Raspbian. Available: *http://www.raspipress.com/2014/06/tutorial-install-mysql-server-on-raspbian/*

[14] CY's Tech Talk.(2015).Installing the Arduino IDE on Raspberry Pi. Available: *http://tech.cyborg5.com/2013/05/30/irlib-tutorial-part-3d-installing-the-arduino-ide-on-a-raspberry-pi/*

[15] ERwin.(2015).Software Trials. Available: *http://erwin.com/resources/software-trials*

[16] Adobe Dreamweaver CC.(2015).Dreamweaver Help. Available: *https://helpx.adobe.com/dreamweaver/topics-cs6.html?promoid=KFPHA*

[17] Xampp.(2015).Download. Available: *https://www.apachefriends.org/download.html*

[18] Raspberry Pi Community Projects.(2015).How To – Remote Desktop with Xming and Putty. Available: *http://www.raspians.com/Knowledgebase/how-to-remote-desktop-with-xming-and-putty/*

[19] Arduino.(2015).Examples. Available: *http://www.arduino.cc/en/Tutorial/HomePage*

[20] launchpad.(2015).MySQL Connector/Arduino. Available: *https://launchpad.net/mysql-arduino*

[20] Charles Bell, Beginning Sensors Network with Arduino and Raspberry Pi

[21] Chuck's Blog.(2015).Introducing MySQL Connector/Arduino. Available: *http://drcharlesbell.blogspot.no/2013/04/introducing-mysql-connectorarduino_6.html*

[22] Raspberry Pi.(2015).Raspberry Pi Camera Module. Available: *https://www.raspberrypi.org/documentation/raspbian/applications/camera.md*

[23] techradar pro.(2015). How to install Raspberry Pi Camera Module. Available: *http://www.techradar.com/news/computing-components/peripherals/how-to-install-the-raspberry-pi-camera-module-1172034*

[24] raspberry pi pod.(2015).Stream the #Raspberry Pi camera module through your browser. Available: *http://www.recantha.co.uk/blog/?p=4165*

[25]CD Automation uk.(2015). De-mystifying PID feedback loop control. Available: *http://www.cdautomation.co.uk/blog/wp-content/uploads/2013/02/PID_Block_Diagram.jpg*

[26]Fermentation Riot.(2015).DIY Arduino PID .Available: *http://fermentationriot.com/arduinopid.php*

[27] Raspberry Pi.(2015).how to disable the red led on the pi camera module. Available: *http://www.raspberrypi-spy.co.uk/2013/05/how-to-disable-the-red-led-on-the-pi-camera-module/?utm_source=feedly*

[28] ustream. (2015). Raspberry Pi Streaming video to Ustream. Available:

https://ustream.zendesk.com/entries/63723444-Raspberry-Pi-Streaming-video-to-Ustream

8 Appendix

8.1 Appendix: Thesis Task Description

Telemark University College
Faculty of Technology

FMH606 Master's Thesis

Title: Home Automation

TUC supervisor: Hans-Petter Halvorsen

External partner: National Instruments

Task description:

Home automation has greatly increased in popularity over the past several years. Home Automation may include centralized control of lighting, heating, ventilation and air conditioning, appliances, security locks of gates and doors and other systems, to provide improved convenience, comfort, energy efficiency and security.

The following topics could be investigated in this project:

- Study of existing Home Automation Systems and explore the Arduino and the Raspberry Pi platforms to see how they can be integrated and used for Home Automation
- Development of PID control (or other control strategies like MPC) for temperature control in houses.
- Logging, Monitoring and Control of typical Data in Homes, e.g., Temperature, etc.
- Create a Home Automation Prototype based on Arduino/Raspberry Pi (e.g., with Temperature Control, Security Cameras, Alarms, Remote Control, …)

Adress: Kjølnes ring 56, NO-3918 Porsgrunn, Norway. **Phone:** 35 57 50 00. **Fax:** 35 55 75 47.

4. Using Arduino/Raspberry Pi for Monitoring and Datalogging using Web Services and "Data Dashboard for LabVIEW" (App for Smartphones and Tablets)
5. Development of PID control (or other control strategies like MPC) for temperature control in houses.
6. Create a Home Automation Prototype based on Arduino/Raspberry Pi (Temperature Logging and Control)
7. Explore integration with Apple HomeKit, Android Home Automation or Microsoft HomeOS. These are frameworks for communicating with and controlling connected accessories in a user's home.
8. Explore Wireless Communication, such as WiFi, Bluetooth, XBee, RFID together with Raspberry Pi and Arduino for Publishing and Monitoring Home Data
9. Using OPC UA together with Raspberry Pi
10. Use of e.g., the ZigBee Home Automation standard for Sensor Integration
11. Using Arduino within LabVIEW; LabVIEW LINX

Based on the students interest, he should select some of the topics above in collaboration with the supervisor for further investigation.

References:

- Smart Homes (in Norwegian): http://www.dinside.no/931315/smarthjem-i-kraftig-vekst
- The Arduino platform: http://arduino.cc
- The Raspberry Pi platform: http://www.raspberrypi.org, http://www.raspberrypi.org/tag/home-automation/
- Apple HomeKit: https://developer.apple.com/homekit
- Microsoft HomeOS: http://research.microsoft.com/en-us/projects/homeos/
- ZigBee Home Automation: http://www.zigbee.org/Standards/ZigBeeHomeAutomation/Overview.aspx
- LabVIEW LINX (using Arduino with the LabVIEW platform): https://www.labviewhacker.com/doku.php?id=learn:libraries:linx:getting_started
- Xively: https://xively.com

Task background:
Home Automation Systems have been very popular today and many vendors have solutions within this area. This must also be seen in connection with Internet of Things (IoT).

Arduino is a low cost open-source electronics prototyping platform. The Arduino has analog and digital I/O.

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and may be used with a standard keyboard and mouse.

Student category: SCE students
Practical arrangements: None

Signatures:
Supervisor (date and signature): 2/6-15 Hasp. Halvorsen
Students (date and signature): June-02,2015 Rashik

8.2 Appendix : Abstract Page

Telemark University College

Faculty of Technology
M.Sc. Programme

MASTER THESIS ON HOMEAUTOMATION

Students: **Pratik Gadtaula**

Thesis title: **HomeAutomation**

Signatures: . . . . . Pratik. . . . . .

Number of pages: <85>

Keywords: . . . . . .

. . . . . .

. . . . . .

Supervisor: Hans Petter-Halvorsen sign.: . . . . . .

2nd Supervisor: <name> sign.: . . . . . .

Censor: <name> sign.: . . . . . .

External partner: <name> sign.: . . . . . .

Availability: <Open/Secret>

Archive approval (supervisor signature)**:** sign.: . . . . . **Date :** . . . . . .

Abstract:

The thesis is on HomeAutomation. It covers the area of monitoring and controlling appliances in home as per users configuration and control. As the automation is performed on Raspberry Pi device along with Arduino board, it combines the overall benefits from both devices and thus useful in implementing our tasks.

It primarily focus on safety and other facilities extended along with it. Services like knowing temperature reading, lights on/off condition, fan on/off and other services are featured in this HomeAutomation. The Alarm system is major part in HomeAutomation which secure the home and update user with right information in right time to avoid accident or loss.

The controlling section is great importance in HomeAutomation. User will have automatic settings to control the appliances. Further , this service is good and one of the reliable way to encapsulate home from internal and external danger. People in job or outside home can work freely and smartly having control to their home. They can sit and just login browser and see what is going on in their home in just a second and feel that their home is with them all time.

HomeAutomation is truly one of the needs in today's world. People rely and feel safe and warmth in their home with their family. HomeAutomation brings more closer and safer to them.

8.3 Appendix: LM-35 DATASHEET

TEXAS INSTRUMENTS

LM35
SNIS159E – AUGUST 1999 – REVISED JANUARY 2015

LM35 Precision Centigrade Temperature Sensors

1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60-µA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

3 Description

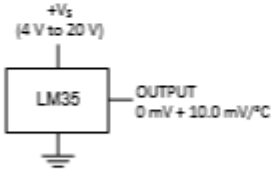
The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temperature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

Device Information[1]

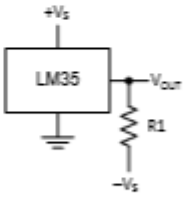
PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM35	TO-CAN (3)	4.699 mm × 4.699 mm
	TO-92 (3)	4.30 mm × 4.30 mm
	SOIC (8)	4.90 mm × 3.91 mm
	TO-220 (3)	14.986 mm × 10.16 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Basic Centigrade Temperature Sensor (2°C to 150°C)



Full-Range Centigrade Temperature Sensor



Choose $R_1 = -V_S / 50\ \mu A$
$V_{OUT} = 1500$ mV at 150°C
$V_{OUT} = 250$ mV at 25°C
$V_{OUT} = -550$ mV at −55°C

Table of Contents

4 Revision History

Changes from Revision D (October 2013) to Revision E — **Page**

- *Added Pin Configuration and Functions* section, *ESD Ratings* table, *Feature Description* section, *Device Functional Modes*, *Application and Implementation* section, *Power Supply Recommendations* section, *Layout* section, *Device and Documentation Support* section, and *Mechanical, Packaging, and Orderable Information* section ... 1

Changes from Revision C (July 2013) to Revision D — **Page**

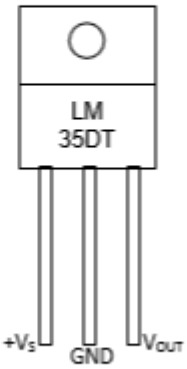
- Changed *W* to *Ω* ... 1
- Changed *W* to *Ω* in *Abs Max* tablenote ... 4

5 Pin Configuration and Functions

NDV Package
3-Pin TO-CAN
(Top View)

Case is connected to negative pin (GND)

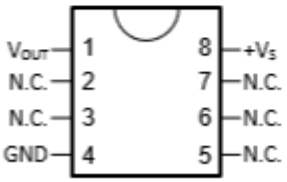
NEB Package
3-Pin TO-220
(Top View)



Tab is connected to the negative pin (GND).

NOTE: The LM35DT pinout is different than the discontinued LM35DP

D Package
8-PIN SOIC
(Top View)



N.C. = No connection

LP Package
3-Pin TO-92
(Bottom View)

Pin Functions

PIN					TYPE	DESCRIPTION
NAME	TO46	TO92	TO220	SO8		
V_{OUT}	—	—	—	1	O	Temperature Sensor Analog Output
N.C.	—	—	—	2	—	No Connection
	—	—	—	3		
GND	—	—	—	4	GROUND	Device ground pin, connect to power supply negative terminal
N.C.	—	—	—	5	—	No Connection
	—	—	—	6		
	—	—	—	7		
$+V_S$	—	—	—	8	POWER	Positive power supply pin

Electrical Characteristics: LM35A, LM35CA (continued)

Unless otherwise noted, these specifications apply: −55°C ≤ T_J ≤ 150°C for the LM35 and LM35A; −40°C ≤ T_J ≤ 110°C for the LM35C and LM35CA; and 0°C ≤ T_J ≤ 100°C for the LM35D. V_S = 5 Vdc and I_{LOAD} = 50 µA, in the circuit of Full-Range Centigrade Temperature Sensor. These specifications also apply from 2°C to T_{MAX} in the circuit of Figure 14.

PARAMETER	TEST CONDITIONS		LM35A			LM35CA			UNIT
			MIN	TYP	MAX	TYP	TYP	MAX	
Quiescent current(6)	V_S = 5 V, 25°C			56			56		µA
		Tested Limit(2)			67			67	
		Design Limit(3)							
	V_S = 5 V, −40°C ≤ T_J ≤ 125°C			105			91		
		Tested Limit(2)							
		Design Limit(3)			131			114	
	V_S = 30 V, 25°C			56.2			56.2		
		Tested Limit(2)			68			68	
		Design Limit(3)							
	V_S = 30 V, −40°C ≤ T_J ≤ 125°C			105.5			91.5		
		Tested Limit(2)							
		Design Limit(3)			133			116	
Change of quiescent current(5)	4 V ≤ V_S ≤ 30 V, 25°C			0.2			0.2		µA
		Tested Limit(2)			1			1	
		Design Limit(3)							
	4 V ≤ V_S ≤ 30 V, −40°C ≤ T_J ≤ 125°C			0.5			0.5		
		Tested Limit(2)							
		Design Limit(3)			2			2	
Temperature coefficient of quiescent current	−40°C ≤ T_J ≤ 125°C			0.39			0.39		µA/°C
		Tested Limit(2)							
		Design Limit(3)			0.5			0.5	
Minimum temperature for rate accuracy	In circuit of Figure 14, I_L = 0			1.5			1.5		°C
		Tested Limit(2)							
		Design Limit(3)			2			2	
Long term stability	T_J = T_{MAX}, for 1000 hours			±0.08			±0.08		°C

(6) Quiescent current is defined in the circuit of Figure 14.

6.6 Electrical Characteristics: LM35A, LM35CA

Unless otherwise noted, these specifications apply: −55°C ≤ T_J ≤ 150°C for the LM35 and LM35A; −40°C ≤ T_J ≤ 110°C for the LM35C and LM35CA; and 0°C ≤ T_J ≤ 100°C for the LM35D. V_S = 5 Vdc and I_{LOAD} = 50 µA, in the circuit of Full-Range Centigrade Temperature Sensor. These specifications also apply from 2°C to T_{MAX} in the circuit of Figure 14.

PARAMETER	TEST CONDITIONS		LM35A			LM35CA			UNIT
			MIN	TYP	MAX	TYP	TYP	MAX	
Accuracy(1)	T_A = 25°C			±0.2			±0.2		°C
		Tested Limit(2)			±0.5			±0.5	
		Design Limit(3)							
	T_A = −10°C			±0.3			±0.3		
		Tested Limit(2)							
		Design Limit(3)						±1	
	T_A = T_{MAX}			±0.4			±0.4		
		Tested Limit(2)			±1			±1	
		Design Limit(3)							
	T_A = T_{MIN}			±0.4			±0.4		
		Tested Limit(2)			±1				
		Design Limit(3)						±1.5	
Nonlinearity(4)	T_{MIN} ≤ T_A ≤ T_{MAX}, −40°C ≤ T_J ≤ 125°C			±0.18			±0.15		°C
		Tested Limit(2)							
		Design Limit(3)			±0.35			±0.3	
Sensor gain (average slope)	T_{MIN} ≤ T_A ≤ T_{MAX}			10			10		mV/°C
		Tested Limit(2)			9.9				
		Design Limit(3)						9.9	
	−40°C ≤ T_J ≤ 125°C			10			10		
		Tested Limit(2)			10.1				
		Design Limit(3)						10.1	
Load regulation(5) 0 ≤ I_L ≤ 1 mA	T_A = 25°C			±0.4			±0.4		mV/mA
		Tested Limit(2)			±1			±1	
		Design Limit(3)							
	T_{MIN} ≤ T_A ≤ T_{MAX}, −40°C ≤ T_J ≤ 125°C			±0.5			±0.5		
		Tested Limit(2)							
		Design Limit(3)			±3			±3	
Line regulation(5)	T_A = 25°C			±0.01			±0.01		mV/V
		Tested Limit(2)			±0.05			±0.05	
		Design Limit(3)							
	4 V ≤ V_S ≤ 30 V, −40°C ≤ T_J ≤ 125°C			±0.02			±0.02		
		Tested Limit(2)							
		Design Limit(3)			±0.1			±0.1	

(1) Accuracy is defined as the error between the output voltage and 10 mv/°C times the case temperature of the device, at specified conditions of voltage, current, and temperature (expressed in °C).
(2) Tested Limits are ensured and 100% tested in production.
(3) Design Limits are ensured (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.
(4) Non-linearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the rated temperature range of the device.
(5) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

6.5 Electrical Characteristics: LM35A, LM35CA Limits

Unless otherwise noted, these specifications apply: −55°C ≤ T_J ≤ 150°C for the LM35 and LM35A; −40°C ≤ T_J ≤ 110°C for the LM35C and LM35CA; and 0°C ≤ T_J ≤ 100°C for the LM35D. V_S = 5 Vdc and I_{LOAD} = 50 µA, in the circuit of Full-Range Centigrade Temperature Sensor. These specifications also apply from 2°C to T_{MAX} in the circuit of Figure 14.

PARAMETER	TEST CONDITIONS	LM35A			LM35CA			UNIT
		TYP	TESTED LIMIT(1)	DESIGN LIMIT(2)	TYP	TESTED LIMIT(1)	DESIGN LIMIT(2)	
Accuracy(3)	T_A = 25°C	±0.2	±0.5		±0.2	±0.5		°C
	T_A = −10°C	±0.3			±0.3		±1	
	T_A = T_{MAX}	±0.4	±1		±0.4	±1		
	T_A = T_{MIN}	±0.4	±1		±0.4		±1.5	
Nonlinearity(4)	T_{MIN} ≤ T_A ≤ T_{MAX}, −40°C ≤ T_J ≤ 125°C	±0.18		±0.35	±0.15		±0.3	°C
Sensor gain (average slope)	T_{MIN} ≤ T_A ≤ T_{MAX}	10	9.9		10		9.9	mV/°C
	−40°C ≤ T_J ≤ 125°C	10	10.1		10		10.1	
Load regulation(5) 0 ≤ I_L ≤ 1 mA	T_A = 25°C	±0.4	±1		±0.4	±1		mV/mA
	T_{MIN} ≤ T_A ≤ T_{MAX}, −40°C ≤ T_J ≤ 125°C	±0.5		±3	±0.5		±3	
Line regulation(5)	T_A = 25°C	±0.01	±0.05		±0.01	±0.05		mV/V
	4 V ≤ V_S ≤ 30 V, −40°C ≤ T_J ≤ 125°C	±0.02		±0.1	±0.02		±0.1	
Quiescent current(6)	V_S = 5 V, 25°C	56	67		56	67		µA
	V_S = 5 V, −40°C ≤ T_J ≤ 125°C	105		131	91		114	
	V_S = 30 V, 25°C	56.2	68		56.2	68		
	V_S = 30 V, −40°C ≤ T_J ≤ 125°C	105.5		133	91.5		116	
Change of quiescent current(5)	4 V ≤ V_S ≤ 30 V, 25°C	0.2	1		0.2	1		µA
	4 V ≤ V_S ≤ 30 V, −40°C ≤ T_J ≤ 125°C	0.5		2	0.5		2	
Temperature coefficient of quiescent current	−40°C ≤ T_J ≤ 125°C	0.39		0.5	0.39		0.5	µA/°C
Minimum temperature for rate accuracy	In circuit of Figure 14, I_L = 0	1.5		2	1.5		2	°C
Long term stability	T_J = T_{MAX}, for 1000 hours	±0.08			±0.08			°C

(1) Tested Limits are ensured and 100% tested in production.
(2) Design Limits are ensured (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.
(3) Accuracy is defined as the error between the output voltage and 10 mV/°C times the case temperature of the device, at specified conditions of voltage, current, and temperature (expressed in °C).
(4) Non-linearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the rated temperature range of the device.
(5) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.
(6) Quiescent current is defined in the circuit of Figure 14.

6 Specifications

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)[1][2]

		MIN	MAX	UNIT
Supply voltage		−0.2	35	V
Output voltage		−1	6	V
Output current			10	mA
Maximum Junction Temperature, T_J max			150	°C
Storage Temperature, T_{stg}	TO-CAN, TO-92 Package	−60	150	°C
	TO-220, SOIC Package	−65	150	

(1) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
(2) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions.

6.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001[1]	±2500	V

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
Specified operating temperature: T_{MIN} to T_{MAX}	LM35, LM35A	−55	150	°C
	LM35C, LM35CA	−40	110	
	LM35D	0	100	
Supply Voltage (+V_S)		4	30	V

6.4 Thermal Information

THERMAL METRIC[1][2]		LM35				UNIT
		NDV	LP	D	NEB	
		3 PINS		8 PINS	3 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance	400	180	220	90	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	24	—	—	—	

(1) For more information about traditional and new thermal metrics, see the *IC Package Thermal Metrics* application report, SPRA953.
(2) For additional thermal resistance information, see *Typical Application*.

6.7 Electrical Characteristics: LM35, LM35C, LM35D Limits

Unless otherwise noted, these specifications apply: −55°C ≤ T_J ≤ 150°C for the LM35 and LM35A; −40°C ≤ T_J ≤ 110°C for the LM35C and LM35CA; and 0°C ≤ T_J ≤ 100°C for the LM35D. V_S = 5 Vdc and I_{LOAD} = 50 µA, in the circuit of Full-Range Centigrade Temperature Sensor. These specifications also apply from 2°C to T_{MAX} in the circuit of Figure 14.

PARAMETER	TEST CONDITIONS	LM35			LM35C, LM35D			UNIT
		TYP	TESTED LIMIT(1)	DESIGN LIMIT(2)	TYP	TESTED LIMIT(1)	DESIGN LIMIT(2)	
Accuracy, LM35, LM35C(3)	T_A = 25°C	±0.4	±1		±0.4	±1		°C
	T_A = −10°C	±0.5			±0.5		±1.5	
	T_A = T_{MAX}	±0.8	±1.5		±0.8		±1.5	
	T_A = T_{MIN}	±0.8		±1.5	±0.8		±2	
Accuracy, LM35D(3)	T_A = 25°C				±0.6	±1.5		°C
	T_A = T_{MAX}				±0.9		±2	
	T_A = T_{MIN}				±0.9		±2	
Nonlinearity(4)	T_{MIN} ≤ T_A ≤ T_{MAX}, −40°C ≤ T_J ≤ 125°C	±0.3		±0.5	±0.2		±0.5	°C
Sensor gain (average slope)	T_{MIN} ≤ T_A ≤ T_{MAX}, −40°C ≤ T_J ≤ 125°C	10	9.8		10		9.8	mV/°C
		10	10.2		10		10.2	
Load regulation(5) 0 ≤ I_L ≤ 1 mA	T_A = 25°C	±0.4	±2		±0.4	±2		mV/mA
	T_{MIN} ≤ T_A ≤ T_{MAX}, −40°C ≤ T_J ≤ 125°C	±0.5		±5	±0.5		±5	
Line regulation(5)	T_A = 25°C	±0.01	±0.1		±0.01	±0.1		mV/V
	4 V ≤ V_S ≤ 30 V, −40°C ≤ T_J ≤ 125°C	±0.02		±0.2	±0.02		±0.2	
Quiescent current(6)	V_S = 5 V, 25°C	56	80		56	80		µA
	V_S = 5 V, −40°C ≤ T_J ≤ 125°C	105		158	91		138	
	V_S = 30 V, 25°C	56.2	82		56.2	82		
	V_S = 30 V, −40°C ≤ T_J ≤ 125°C	105.5		161	91.5		141	
Change of quiescent current(5)	4 V ≤ V_S ≤ 30 V, 25°C	0.2	2		0.2	2		µA
	4 V ≤ V_S ≤ 30 V, −40°C ≤ T_J ≤ 125°C	0.5		3	0.5		3	
Temperature coefficient of quiescent current	−40°C ≤ T_J ≤ 125°C	0.39		0.7	0.39		0.7	µA/°C
Minimum temperature for rate accuracy	In circuit of Figure 14, I_L = 0	1.5		2	1.5		2	°C
Long term stability	T_J = T_{MAX}, for 1000 hours	±0.08			±0.08			°C

(1) Tested Limits are ensured and 100% tested in production.
(2) Design Limits are ensured (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.
(3) Accuracy is defined as the error between the output voltage and 10 mv/°C times the case temperature of the device, at specified conditions of voltage, current, and temperature (expressed in °C).
(4) Non-linearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the rated temperature range of the device.
(5) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.
(6) Quiescent current is defined in the circuit of Figure 14.

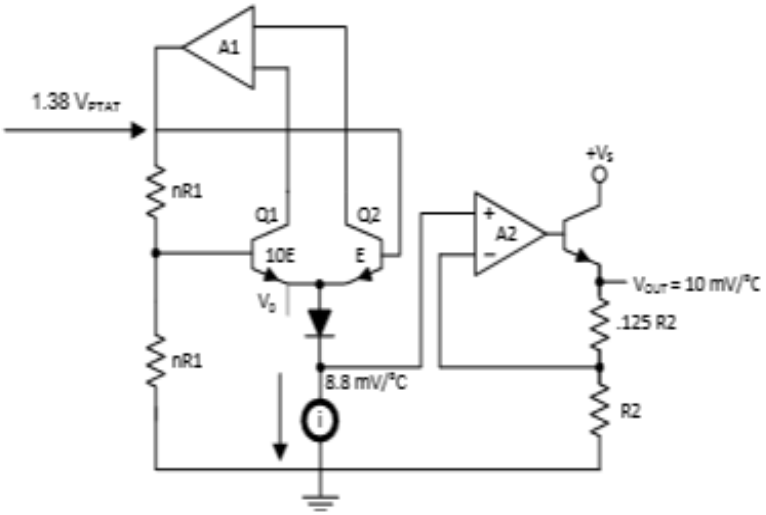
7 Detailed Description

7.1 Overview

The LM35-series devices are precision integrated-circuit temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ± ¼ °C at room temperature and ± ¾ °C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temperature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy). The temperature-sensing element is comprised of a delta-V BE architecture.

The temperature-sensing element is then buffered by an amplifier and provided to the VOUT pin. The amplifier has a simple class A output stage with typical 0.5-Ω output impedance as shown in the *Functional Block Diagram*. Therefore the LM35 can only source current and it's sinking capability is limited to 1 µA.

7.2 Functional Block Diagram



7.3 Feature Description

7.3.1 LM35 Transfer Function

The accuracy specifications of the LM35 are given with respect to a simple linear transfer function:

$V_{OUT} = 10\ mv/°F \times T$

where

- V_{OUT} is the LM35 output voltage
- T is the temperature in °C

(1)

7.4 Device Functional Modes

The only functional mode of the LM35 is that it has an analog output directly proportional to temperature.

8 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

8.1 Application Information

The features of the LM35 make it suitable for many general temperature sensing applications. Multiple package options expand on it's flexibility.

8.1.1 Capacitive Drive Capability

Like most micropower circuits, the LM35 device has a limited ability to drive heavy capacitive loads. Alone, the LM35 device is able to drive 50 pF without special precautions. If heavier loads are anticipated, isolating or decoupling the load with a resistor is easy (see Figure 12). The tolerance of capacitance can be improved with a series R-C damper from output to ground (see Figure 13).

When the LM35 device is applied with a 200-Ω load resistor as shown in Figure 16, Figure 17, or Figure 19, the device is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input and not on the output. However, as with any linear circuit connected to wires in a hostile environment, performance is affected adversely by intense electromagnetic sources (such as relays, radio transmitters, motors with arcing brushes, and SCR transients), because the wiring acts as a receiving antenna and the internal junctions act as rectifiers. For best results in such cases, a bypass capacitor from V_{IN} to ground and a series R-C damper, such as 75 Ω in series with 0.2 or 1 µF from output to ground, are often useful. Examples are shown in Figure 13, Figure 24, and Figure 25.

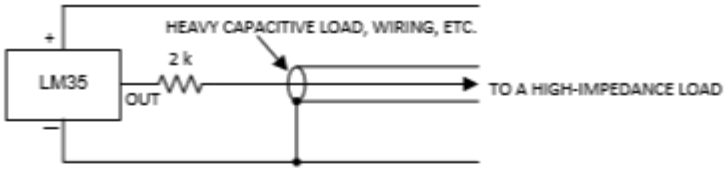


Figure 12. LM35 with Decoupling from Capacitive Load

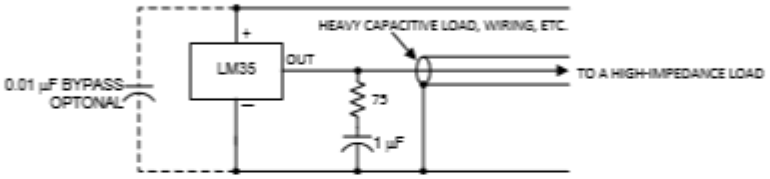


Figure 13. LM35 with R-C Damper

8.4 Appendix: LDR DATASHEET

Data pack F *Issued March 1997* **232-3816**

Data Sheet

Light dependent resistors

NORP12 RS stock number 651-507
NSL19-M51 RS stock number 596-141

Two cadmium sulphide (cdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (Lux)
Moonlight	0.1
60W bulb at 1m	50
1W MES bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000

Circuit symbol

Light memory characteristics

Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (**RS** stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	320V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Electrical characteristics

T_A = 25°C. 2854°K tungsten light source

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	1000 lux 10 lux	– –	400 9	– –	Ω kΩ
Dark resistance	–	1.0	–	–	MΩ
Dark capacitance	–	–	3.5	–	pF
Rise time 1	1000 lux 10 lux	– –	2.8 18	– –	ms ms
Fall time 2	1000 lux 10 lux	– –	48 120	– –	ms ms

1. Dark to 110% R_L
2. To 10 × R_L

R_L = photocell resistance under given illumination.

Features

- Wide spectral response
- Low cost
- Wide ambient temperature range.

Dimensions

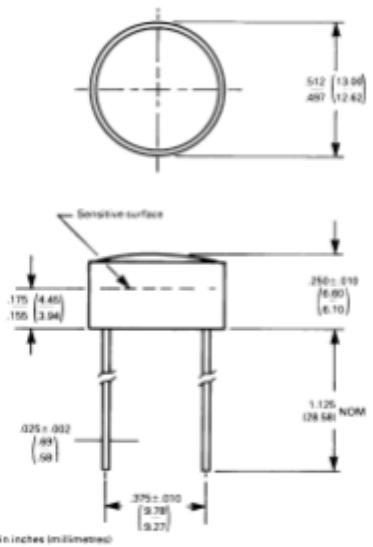


Figure 1 Power dissipation derating

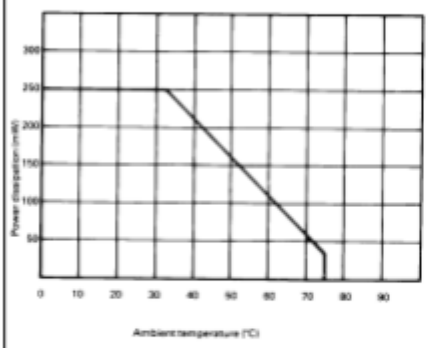
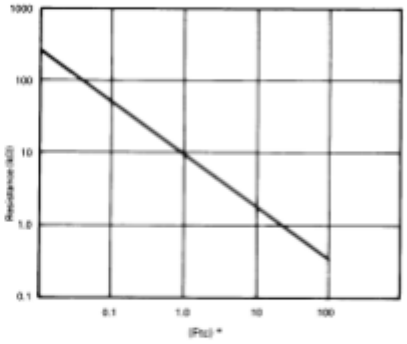
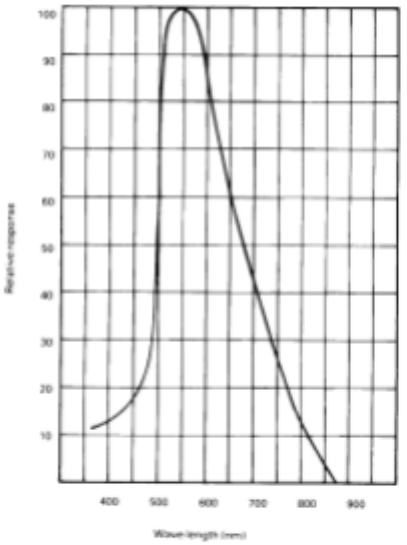


Figure 3 Resistance as a function of illumination



*1Fc=10.764 lumens

Figure 2 Spectral response



Absolute maximum ratings

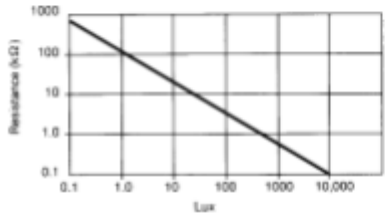
Voltage, ac or dc peak _____ 100V
Current _____ 5mA
Power dissipation at 25°C _____ 50mW*
Operating temperature range _____ -25°C +75°C

*Derate linearly from 50mW at 25°C to 0W at 75°C.

Electrical characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	10 lux 100 lux	20 -	- 5	100 -	kΩ kΩ
Dark resistance	10 lux after 10 sec	20	-	-	MΩ
Spectral response	-	-	550	-	nm
Rise time	10fc	-	45	-	ms
Fall time	10fc	-	55	-	ms

Figure 4 Resistance as a function illumination



Dimensions

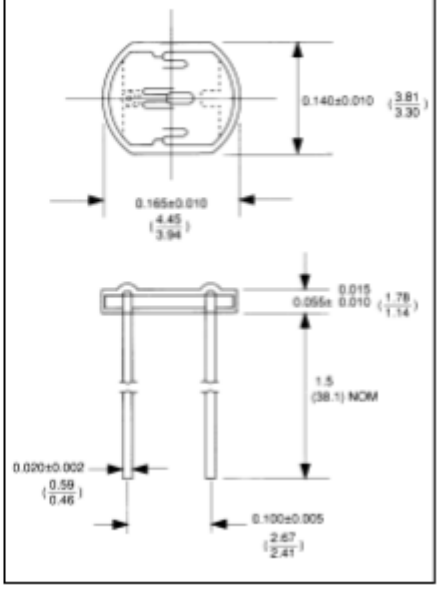
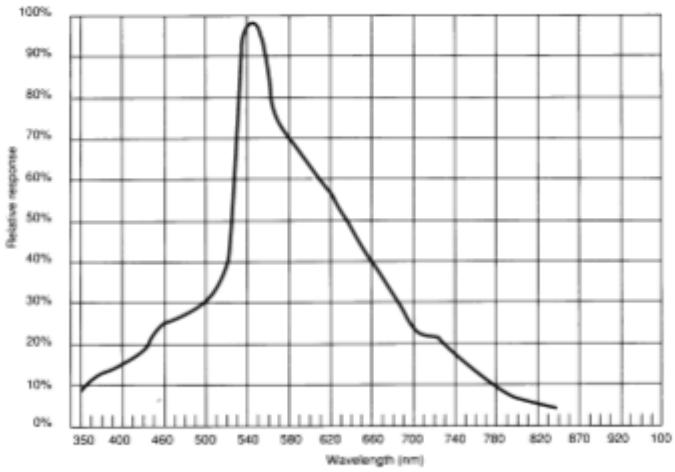


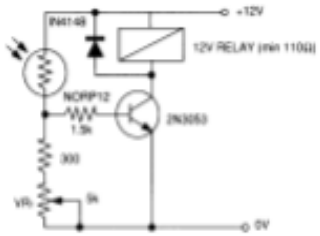
Figure 5 Spectral response



232-3816

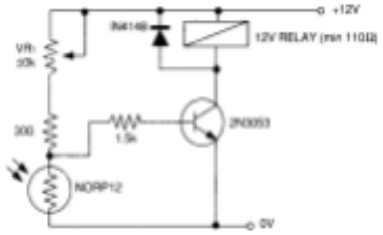
Typical application circuits

Figure 6 **Sensitive light operated relay**



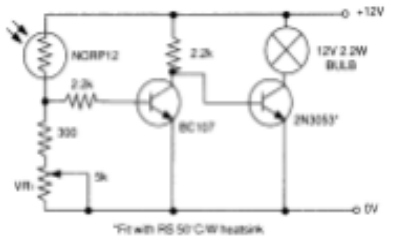
Relay energised when light level increases above the level set by VR_1

Figure 7 **Light interruption detector**



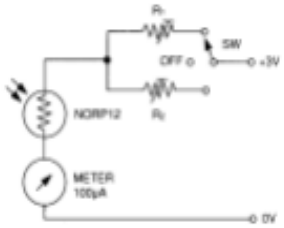
As Figure 6 relay energised when light level drops below the level set by VR_1

Figure 8 **Automatic light circuit**



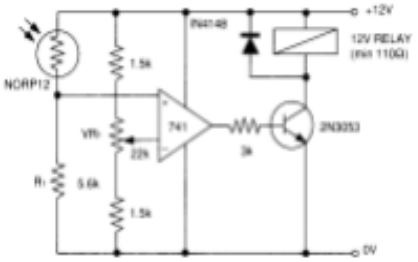
Adjust turn-on point with VR_1

Figure 9 **Logarithmic law photographic light meter**



Typical value $R^1 = 100k\Omega$
$R^2 = 200k\Omega$ preset to give two overlapping ranges.
(Calibration should be made against an accurate meter.)

Figure 10 **Extremely sensitive light operated relay**



(Relay energised when light exceeds preset level.)
Incorporates a balancing bridge and op-amp. R_1 and NORP12 may be interchanged for the reverse function.

RS Components, PO Box 99, Corby, Northants, NN17 9RS
An Electrocomponents Company
Telephone: 01536 201234

8.5 Appendix: ARDUINO CODE For HomeAutomation

```
#include <sha1.h>
#include <SPI.h>
#include <WiFi.h>
#include <mysql.h>

IPAddress server_addr(10,0,0,103);
char user[] = "root";
char password[] = "raspberry";
Connector my_conn;        // The Connector/Arduino reference
char ssid[] = "ThomsonFFCBB5";
char pass[] = "36***AD4C";
int status = WL_IDLE_STATUS;
WiFiClient client;

int sensorPin = A0;
int fanOutput = 5;
int sensorPin1 = A1;
int lightOutput = 3;
int alarmOutput =9;
int threshold = 400;

void setup() {
 pinMode(fanOutput, OUTPUT);
 pinMode(lightOutput, OUTPUT);
 pinMode(alarmOutput, OUTPUT);
 Serial.begin(115200);
                while (!Serial);
                int status = WiFi.begin(ssid, pass);
                 if ( status != WL_CONNECTED) {
                 Serial.println("Couldn't get a wifi connection");
                 while(true);
                  }
                else {
                Serial.println("Connected to network");
                IPAddress ip = WiFi.localIP();
                Serial.print("My IP address is: ");
                Serial.println(ip);
                 }}
```

```
void loop(){
    int read = analogRead(sensorPin);
    float volt = (read/1024.0)*5.0;
    int t = (volt - 0.5)*100;
    Serial.println(t);
    delay(1000);
    int val = analogRead(sensorPin1);
    Serial.println(val);

    if (t<30){
              digitalWrite(fanOutput, LOW);
               if (val < threshold) {
                  digitalWrite(lightOutput, HIGH);
                  if (my_conn.mysql_connect (server_addr, 3306, user, password)){
                  char buf[128];
                  sprintf(buf,"INSERT INTO homeautomation.tagdata(datavalue,photodata,tagid) VALUES('%d','%d','%d')",t,val,15);
                  my_conn.cmd_query(buf);
               }}

             else {
                   digitalWrite(lightOutput, LOW);
                    if (my_conn.mysql_connect (server_addr, 3306, user, password)){
                    char buf[128];
                    sprintf(buf,"INSERT INTO homeautomation.tagdata(datavalue,photodata,tagid) VALUES('%d','%d','%d')",t,val,15);
                    my_conn.cmd_query(buf);
                    }}}
```

```
    else if (t>30) {
                    digitalWrite(fanOutput, HIGH);
                    analogWrite(alarmOutput, 20);
                    delay(5000);
                     analogWrite(alarmOutput, 0);
                     delay(5000);
                   if (val < threshold) {
                       digitalWrite(lightOutput, HIGH);
                        if (my_conn.mysql_connect (server_addr, 3306, user, password)){
                         char buf[128];
                         sprintf(buf,"INSERT INTO homeautomation.tagdata(datavalue,photodata,tagid) VALUES('%d','%d','%d')",t,val,15);
                         my_conn.cmd_query(buf);
                        }}

                    else {
                          digitalWrite(lightOutput, LOW);
                       if (my_conn.mysql_connect (server_addr, 3306, user, password)){
                       char buf[128];
                        sprintf(buf,"INSERT INTO homeautomation.tagdata(datavalue,photodata,tagid) VALUES('%d','%d','%d')",t,val,15);
                       my_conn.cmd_query(buf);
                        }}}
 delay(1800000);
}
```

8.6 Appendix: PID CODE for Room Heater

```
#include <PID_v1.h>
const int lm35 = A0;   // LM-35 input pin
const int pot = A1;        // Potentiometer input pin
const int heater = 9;         // Heater output pin
double tempLevel;         // Indirectly store the light level
// Tuning parameters
double aggKp=4, aggKi=0.2, aggKd=1;

float Kp = 200;              // Proportional gain
float Ki = 50;             // Integral gain
float Kd = 0;              // Differential gain
// Record the set point as well as the controller input and output
double Setpoint, Input, Output;
// Create a controller that is linked to the specified Input, Ouput and Setpoint
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup() {
 tempLevel = analogRead(lm35);                // Read the set point
 // Arduino has an analogueRead() resolution of 0-1023 and an analogueWrite() resolution of 0-255
 Input = map(tempLevel, 0, 1023, 0, 255);        // Scale the input
 Setpoint = map(analogRead(pot), 0, 1023, 0, 255);  // Scale the set point
 Serial.begin(9600);                                // Initialise serial communications at 9600 bps
 myPID.SetMode(AUTOMATIC);                          // Turn on the PID control
 }

void loop()  {
 Setpoint = map(analogRead(pot), 0, 1023, 0, 255);  // Continue to read and scale the set point
 tempLevel = analogRead(lm35);    // Read the light level
 Input = map(tempLevel, 0, 1023, 0, 255);   // Scale the input to the PID
 double gap = abs(Setpoint-Input); //distance away from setpoint

 if(gap<10) {  //we're close to setpoint, use conservative tuning parameters
    myPID.SetTunings(Kp, Ki, Kd);
  }
  else {
     //we're far from setpoint, use aggressive tuning parameters
     myPID.SetTunings(aggKp, aggKi, aggKd);
  }
```

```
 myPID.Compute();                                // Calculates the PID output at a specified sample time
 analogWrite(heater, Output);                       // Power the Heater
   Serial.print("tempLevel = ");
 Serial.print(tempLevel);
 Serial.print("Setpoint = ");
 Serial.print(Setpoint);
 Serial.print(" Input = ");
 Serial.print(Input);
 Serial.print(" Output = ");
 Serial.print(Output);
 Serial.print("\n");
 // The tuning parameters can be retrieved by the Arduino from the serial monitor: 0,0.5,0 set Ki to 0.5.
 }
```

8.7 Appendix: PHPCODE(MySQL Connection Procedure)

```php
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_test = "localhost";
$database_test = "homeautomation";
$username_test = "root";
$password_test = "raspberry";
$test = mysql_pconnect($hostname_test, $username_test, $password_test) or trigger_error(mysql_error(),E_USER_ERROR);
?>
```