4

IMPLEMENTATION

CONTENT

CHAPTER 4

IMPLEMENTATION

4.1 Overview

As mentioned the proposed framework, the implementation of designed automation system is depend on the different modules. They are the server software design, database design and the hardware interface design. In a general view, the whole proposed framework needs some implementation requirements [48].

1) **User friendly interface**: All the users of the present system can easily handle the home automation system locally or remotely through an easy and user friendly graphical user interface.
2) **Security and authentication**: Only valid authorized user can manage, control and monitor the system (locally or remotely) as users have to log in with their user-id and password.
3) **Low cost per node / High node count:** More than hundred nodes may be needed to provide automation in case of building automation. Anyway, society needs competitive performance to be delivered at a low system cost. Along with this, also protocols need to measure to high node count e.g., ensuring message delivery
4) **Large area coverage**: As there is no limitation of area of locations, users can manage the home automation system from a remote place of anywhere. So, the implementation part of the system should have the availability of large area coverage.
5) **System Scalability**: Scalability is the ability of a system, network, or process, to handle growing amount of working a capable manner or its ability to be enlarged to accommodate that growth. For example, system upgrade/downgrade by adding/removing hardware interface module should be easy and systematic task.

The implementation of complete proposed system is based on some systematic way of-

a) Home Server Set Up

b) The user module Initialization

c) Receiving the user's delivered command or instruction by the server.

d) Designing the database.

e) Maintaining the common platform for all four communication mechanisms.

f) Firing the commands to microcontroller

g) Microcontroller programming design

h) Electrical circuit design

i) Providing security features

4.2 Home Server Set Up

Since Home Server is nothing but a PC, So Administrator has to open the channel for the communication with the user through all mechanisms on the server so that users can control the appliances connected with the server from a remote place. In the server interface, users have to activate GSM modem that is connected with the server for the Communications.

4.3 Receiving Command through SMS

At the time of activating the server, the administrator has to open the channel for GSM communication. In the server interface, when user press the "Connect GSM modem" button then two AT commands will generate automatically as per as the programming part. At the beginning, AT+CMGF=1 i.e. the message is appeared in text format as the messages stored in SIM are in binary format i.e. AT+CMGF=0 by default. Then, AT+CNMI=1,2,0,0,0 i.e when GSM modem receives the SMS [58]. It enables the notifications or information about whether the message is reached to the node or not where the GSM modem is connected. After that, the command, AT+CMGR reads the SMS from the particular selected port by using +CPMS command. But, the message has not only the action command, it has also different parameters like source number of the message, status of the message (read/unread), users' name and password etc. The message is usually arrived as mentioned below-

"*message_status,address,[address_text],service_center_time_stamp[,address_type,TPDU_first_octet,protocol_identifier,data_coding_scheme,service_center_address,service_center_address_type,sms_message_body_length]<CR><LF>sms_message_body*".

So, from those, the predefined programming package present in the PC in the server side extracts the proper command sent by the users and if it is valid, dumps it into the database .

The all commands used for the GSM communication is described briefly as follows [49].

- **New Message Indications to Terminal +CNMI**

Description: This command handles enabling of unsolicited notifications to the terminal when an SMS is received by the GSM modem. After sending an unsolicited response to the TE, the G24 will expect a +CNMA (new message acknowledgement) from the TE within a predefined timeout of 60 seconds [59]. The GSM modem will not send another unsolicited response to the TE before the previous one is acknowledged. If acknowledged within the timeout, the new SM is not saved in the message storage. If not, the new SM is saved in the message storage and +CNMI parameters are set to 0.

Syntax: AT+CNMI= [<mode> [,<mt>[,<bm> [,<ds>[,<bfr>]]]]]
Defined values:

<mode>: controls the processing of unsolicited result code

<mt>: sets the result code indication routing for SMS-DELIVERs. Default is 0.

<bm>: set the rules for storing received CBMs (Cell Broadcast Message) types depend on its coding scheme, the setting of Select CBM Types (+CSCB command) and Default is 0

<ds> : for SMS-STATUS-REPORTs. Default is 0.

<bfr>: Default is 0

- **Preferred Message Format + CMGF**

Description: The message formats supported are *text mode* and *PDU mode*. In PDU mode, a complete SMS Message including all header information is given as a binary string (in

hexadecimal format).Therefore, only the following set of characters is allowed: {‘0’,’1’,’2’,’3’,’4’,’5’,’6’,’7’,’8’,’9’, ‘A’,‘B’,’C’,’D’,’E’,’F’}. Each pair or characters are converted to a byte (e.g.: ‘41’ is converted to the ASCII character ‘A’, whose ASCII code is 0x41 or 65). In Text mode, all commands and responses are in ASCII characters. The format selected is stored in EEPROM by the +CSAS command.

Syntax: AT+CMGF=<index>

As example, AT + CMGF=1 for text mode

AT + CMGF=0 for PDU mode

- **Read message +CMGR**

Description: This command allows the application to read stored messages. The messages are read from the memory selected by +CPMS command.

Syntax: AT+CMGR=<index>

- **Send message +CMGS**

Description: The <address> field is the address of the terminal to which the message is sent. To send the message, simply type, <ctrl-Z> character (ASCII 26). The text can contain all existing characters except <ctrl-Z> and <ESC> (ASCII 27). This command can be aborted using the<ESC> character when entering text. In PDU mode, only hexadecimal characters are used(‘0’…’9’,’A’…’F’).

Syntax:Command syntax in text mode*:*

AT+CMGS= <da>[,<toda>] <CR>

text is entered <ctrl-Z / ESC >

Command syntax in PDU mode*:*

AT+CMGS= <length><CR>

PDU is entered <ctrl-Z / ESC >

Parameters used in the syntax are given below-

<da>: Destination address in quoted string. This field contains a single MIN number.

<toda>: Type of DA. Value between 128-255 (according to GSM 03.40.9.1.2.5). If this field is not given and first character of <da> is '+', <toda> will be 145, otherwise 129.

<length>: Size of message in PDU mode format, in octets, excluding SMSC data.
<mr>: Sent message reference number.

4.3 Receiving Command through Web

In order to achieve interaction with the home automation network from the outside, the other option is to use the Internet. To accomplish this, a web server is built to take requests from remote clients. The system is modelled with three different units. The first unit is the PC side which is formed of a user interface component, the database and the web server components. The user interface and the internet front end are connected to a backend database server. The clients/users can send requests to the server machine through a web page. A web page is constructed as an interactive interface where commands can be submitted by the client to change and also monitor the status of the devices. The web page is designed by the high level scripting language PHP. When user gives the command in the web page by pressing the respective button present in the web page, then the command line with receiving system, system date and action is received by the server database designed by MySql. The two machine, user's PC and home server communicate each other through internet using a real IP address (Internet IP). The LAMP server has the main responsibility for communication among the server database and user's web page.

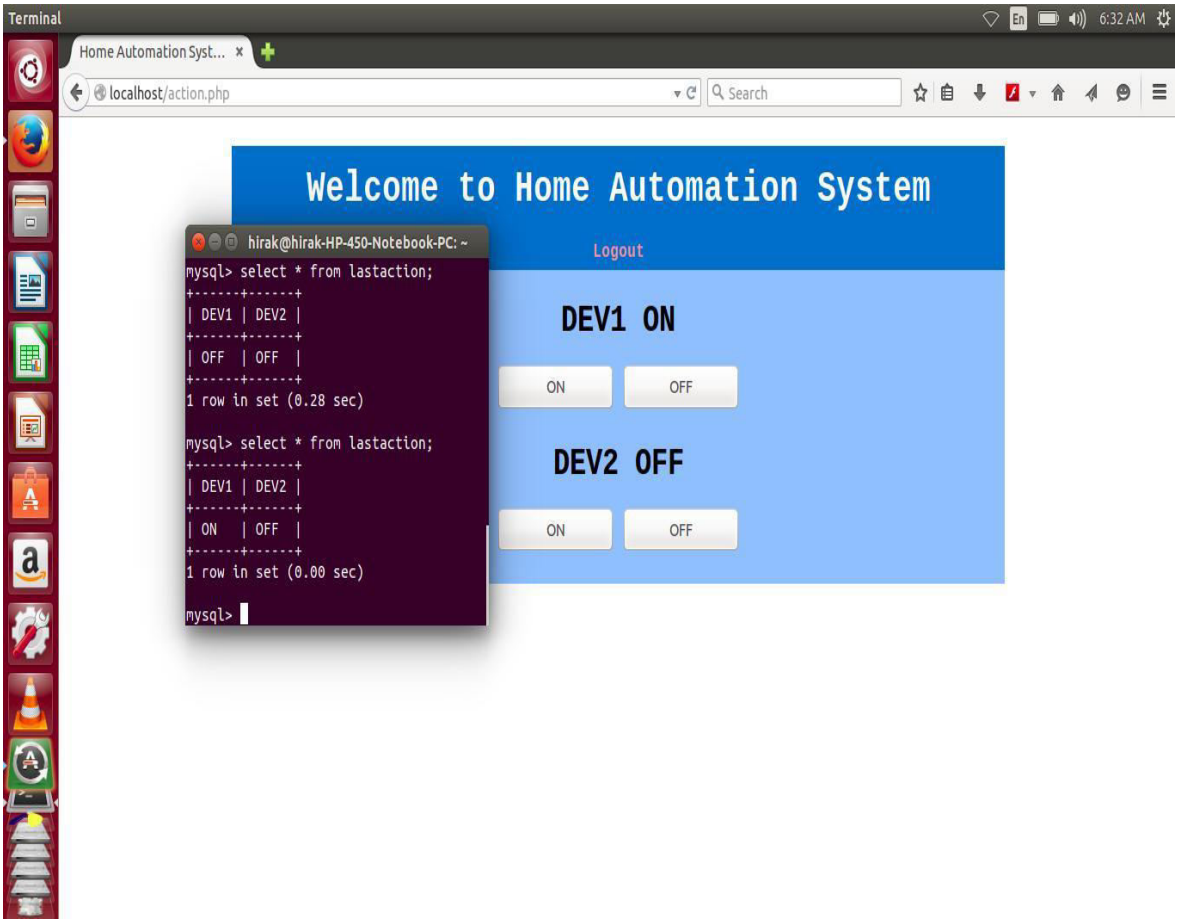

Figure 4.1 Screenshot of Web Interface

4.5 Receiving Command through Speech

As mentioned the proposed framework, the implementation of designed system is depend on mainly two modules. The first one is speech recognition and another one is hardware interface for controlling Devices. The speech Recognition module consists of different sub-modules such as data collection, data preparation, data training, data testing, execution etc. The hardware module provides an interface to get the recognized command that are to be used for device controlling by using a microcontroller, heart of the switching circuit. The main task is to develop interface using relevant speech technologies so that the most noise of users is able to get the relevant information with minimum human intervention. The speech-interface will also be designed to take into speaker independent voice responses as well as errors of the ASR system. The Automatic Speech Recognition engines containing sub-word Hidden Markov Model (HMM) based ASR engines for recognition of appliances name to be controlled [50]. The Speech recognition toolkit HTK is used for generation.

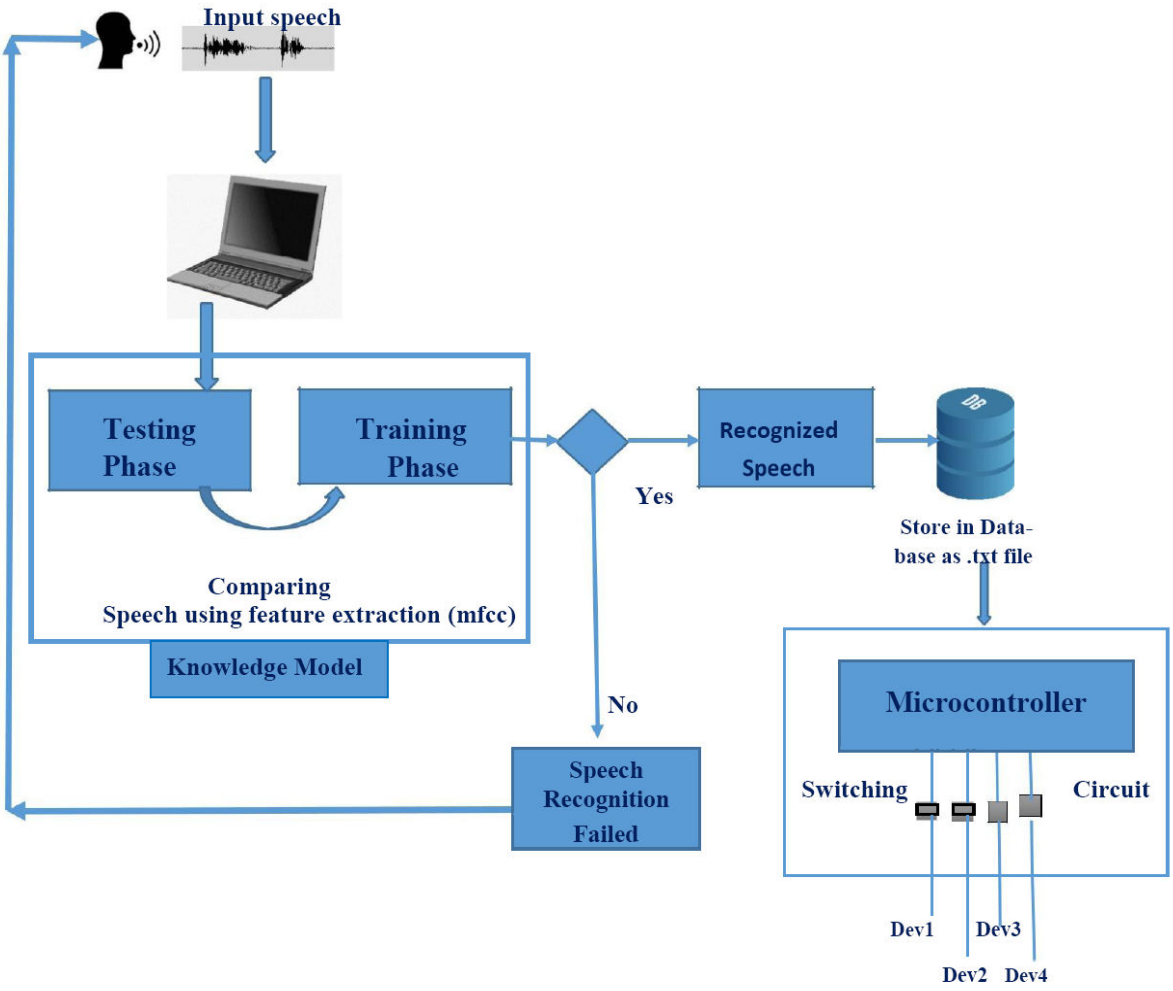

Figure 4.2: Design of Speech Implementation [46]

4.5.1 Input Speech

In our proposed system we use speech as command for controlling electronic device. Therefore we have collected our required voice data from different people in different environment. For collection our required data, we have used microphone and wave surfer as a voice recording toolkit. We have tried to avoid the noise in our recording environment. Still in our collected data, we found some noisy samples. Depending on noisy utterance, we have ignored such samples for further processing. We have saved all voice data in .wav file extension with 16000 bit rate. Collection of voice data for speech recognition is very challenging because the accuracy of recognition depends on this collected input speech as well as collecting training data.

Table 4.1: Data Collection Parameter with its values

Parameter	Value
Sampling Rate	16 kHz,16 bits
Wave Format	Mono, Wav
Speakers	200(140 M +60 F)

Development of Interactive Voice Response System for controlling Devices

For designing an interactive voice response system the developer must design a call flow chart for his program. The authorized users interact with the server system with a predefined call flow. As the Home Automation system mainly deals with some isolated words only hence system seeks isolated word from the user. Otherwise the performance of the system will be decreased. As the IVR system is designed to give the name of the devices that should be controlled and actions to be applied for the devices [51]. The user has to prompt yes/no for increasing the confidence of the reorganization engine. The user will get the information of the status for the current appliances connected with the hardware interface if the name of the devices name provided the recognition of the queries success. If the recognition of the device name or the action (ON/OFF) is failed then the system will be terminated. The following figure 4.2 shows flow of the system is the actual frame work for the IVR system.

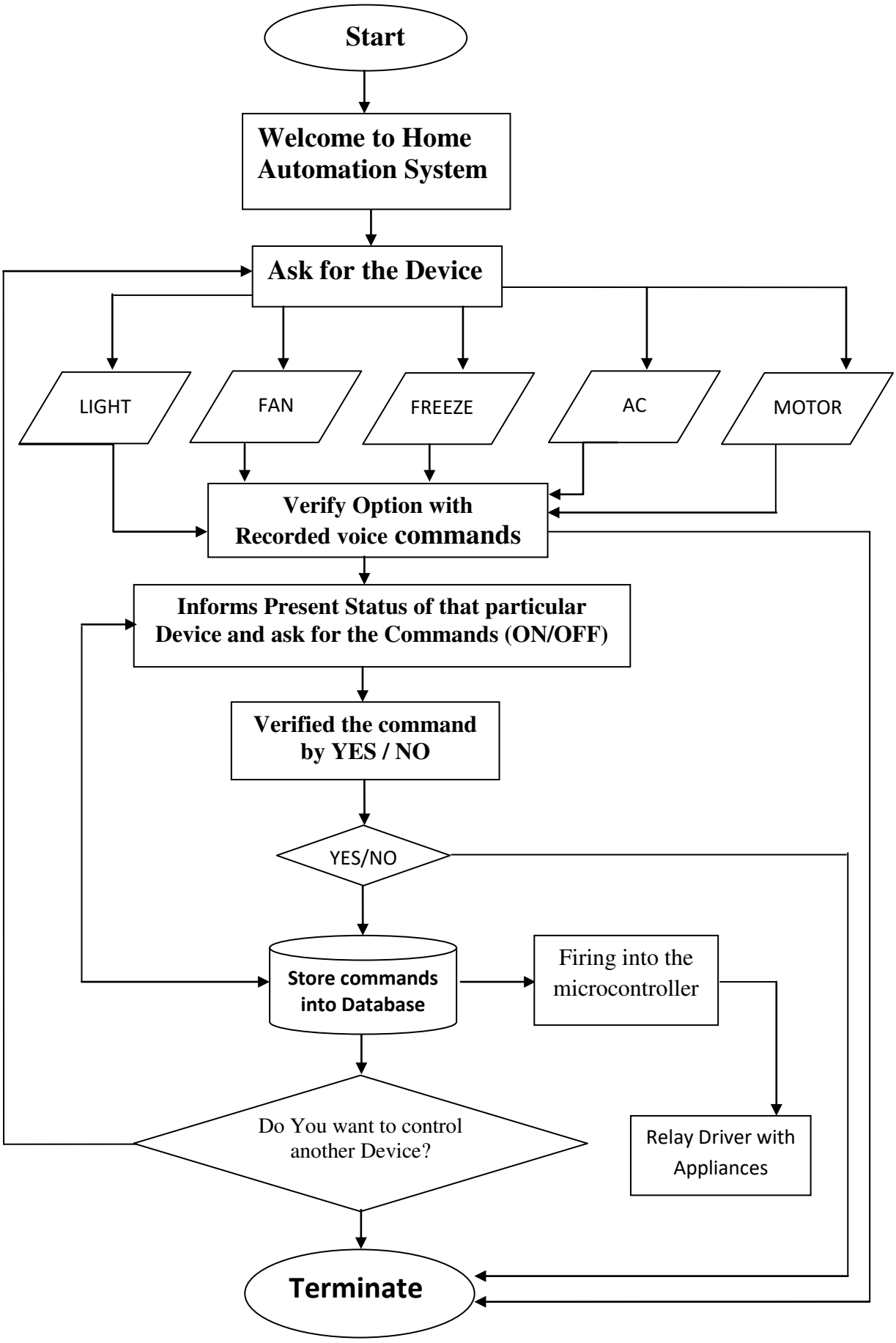

Figure 4.3: Call Flow Diagram for IVR System

4.5.2 IVR design

An IVR system containing a spoken dialog manager is a computer agent that interacts with users by understanding their speech spoken in different languages. The input in spoken dialog systems consist of small set of spoken words like action commands (ON/OFF), appliances name ,yes and no. In this work, a focus is mainly concentrated on building of dialog manger and evaluating the performance of speech recognizer engine. In this designed system user can take the input by making a call, recognize to word with the predefined HTK based recognizer and then language model generate the word based on our recognizer output. Dialog manger takes the output from the language analyzer module and passes it to the language generator. It also controls the structure of the dialogue [52].

The entire server side system consists of Bluetooth enabled mobile phone, Asterisk server, speech interface, dialog manager and Mysql database. Asterisk is open source (Unix/Linux based platform) communication toolkit that enables the computer to be used as a telephone network server. It consists of computer telephone interface (CTI) card and IVR module. This designed system has used Bluetooth connectivity in place of CTI card. The developed system mainly transmits the pre-recorded voice responses and records and processes the voice query from the user. In this proposed framework, user first call to a mobile number which is connected to our asterisk server. After that server mobile get the call then its blue tooth adaptor connect the asterisk server, with the help of Chan mobile module. Chan mobile is asterisk module which helps to connect asterisk server with Bluetooth enabled mobile and it will help to transfer the communication. Chan mobile transfer the call to Dahdi, It is a collection of open source drivers, for Linux, which are used to interface with a variety of telephony related hardware [51]. Here dahdi will identify the channel API and finally land the call to dialplan. The configuration file named as extension.conf is stored in dialplan. Asterisk gateway interface (AGI) program, shell script is used to integrate Asterisk server with speech recognizer module.

To convert the call flow into dialog manger we have used PHP-AGI script which is integrated with extention.config file. Once the call landed to extention.config file the script will activate and start working according to flow.

4.5.3 ASR design

Since a speaker Independent home automation system has been designing and developing, so as per requirement of the designed system users need mainly three types of data, Device name, action command and yes/no. The speech corpus consist of 9 (5+2+2) English words, out of which 5 device names, 2 action commands (ON/OFF) and Yes/No word. We take 70 males speaker and 30 female speaker of different age group for data recording purpose. A total of 7908 words in speech data were collected digitized at 16 bits/sample at a sampling rate of 16 kHz. In this section, the experimental work on word recognition from a connected word speech corpus will be presented. Wave surfer is used for data recording, praat is used for data analysis, Hidden Markov Toolkit (HTK) is used for feature extraction, training and recognition steps [53].

4.5.4 Recognizing Toolkit – HTK

The HTK tools are best introduced by going through the processing steps involved in building a sub-word based continuous speech recogniser.

There are 4 main phases:

- Data preparation
- Training
- Testing
- Analysis.

4.5.4.1 Data preparation

For building a set of HMMs, a set of speech data files and their associated transcriptions are required. Very often speech data will be obtained from database archives. Before it can be used in training, it must be converted into the appropriate parametric form and any associated transcriptions must be converted to have the correct format and use the required phone or word labels. For data preparation, we have collected nine commands as word (“ON”, “OFF”, “YES”, “NO”, “LIGHT”,”FAN”,”FREEZE”,”MOTOR”,”AC”) from 200 people out of which approximately 140 recordings are of male and approximately 60 recordings are of female speakers. The data is recorded with the help of unidirectional microphone using a recording tool wave surfer in .wav extension. The .wav files recorded are saved as HTK transcription. The sampling rate used for recording is 16 KHz. A labeling

tool wave surfer is used to label the speech waveforms. The label files are used in acoustic model generation phase of the system. Following are the command words that we have collection.

The raw speech data is in the form of wave files. This needs to be converted to MFCC speech vectors. This is a form of spectral analysis of the raw waveform, and can be performed by using the tool HCopy. HCopy is used to copy one or more source files to an output file. Normally, HCopy copies the whole file, but a variety of mechanisms are provided for extracting segments of files and concatenating files. The MFC file can be viewed by using the HTK tool HList [66]. A transcript is needed. This needs to be converted to Unicode, and then to label files. Label files are of the format

<Start> <end> <label>

Where start and end correspond to the beginning and end of the part of the waveform to which this label is being assigned. In case of word level it will be the whole file and thus 0 and -1.

In case of phone level recognition, a phonetic dictionary will be needed to convert the words into phones. Then the start and end values will be that of the various component phones.

4.5.4.2 Data Preparation Tools

Although all HTK tools can parameterise waveforms *on-the-fly*, in practice it is usually better to parameterise the data just once. The tool HCopy is used for this. HCopy is used to copy one or more source files to an output file. Normally, HCopy copies the whole file, but a variety of mechanisms are provided for extracting segments of files and concatenating files. By setting the appropriate configuration variables, all input files can be converted to parametric form as they are read-in. Thus, simply copying each file in this manner performs the required encoding.

Transcriptions will also need preparing. Typically the labels used in the original source transcriptions will not be exactly as required, for example, because of differences in the phone sets used. Also, HMM training might require the labels to be context-dependent. The tool HLEd is a script-driven label editor which is designed to make the required transformations to label files. HLEd can also output files to a single *Master Label File* MLF which is usually more convenient for subsequent processing. Finally on data preparation,

HLStats can gather and display statistics on label files in preparation for building discrete probability HMM system.

4.5.4.3 Training and Testing

Firstly, during training, an initial set of models must be created. If there is some speech data available for which the location of the sub-word (i.e. phone) boundaries has been marked.

In training phase HTK system verify the speeches in different phases.

Phase I: It checks dictionary and filler dictionary with phone list file.

A dictionary is explained to the pronunciations of the words as linear sequence of phonemes in such a way that each line in the dictionary contains exactly one pronunciation specification. The dictionary file is save as .dic extension. In the dictionary file have spacing are regular in each line and there are not extra blank lines in the end .The pronunciation is completely case-insensitive, that is, it is not possible to have two different pronunciations ON and on in the dictionary.

- We create a dictionary of all the words that are there in the vocabulary of the recognizer and save it as .dic extension file. Here the entries of the dictionary will be

ON

OFF

LIGHT

FAN

FREEZE

MOTOR

AC

YES

NO

We make sure that the spacing are regular in each line and there are not extra blank lines in the end.

In this file create a filler dictionary with all non-speech sounds with the name .filler extension file. In our case silence is the only non-speech sound. This filler dictionary

includes the special beginning-of-sentence and the end-of-sentence tokens <s> and </s> respectively as well as the silence word <sil>. However, all of them have the same SIL (silence-phone) as their pronunciation.

Example:-

<s> SIL

</s> SIL

<sil> SIL

It is important to make sure that there are no blank spaces after any line in the dictionary.

Phase II: It checks to make sure that there is not any duplicate entry in the dictionary.

Phase III: It checks the general format and utterance length of the training file. Utterance length must be positive.

This file included the word (or utterances), consisting of the spoken text to the corresponding audio files in the transcription file format. We create list of transcription file and save them as a .transcription extension file. We maintain equal spacing between all the words of the lines.

A list of transcription of all the wave files is created. For example if the contents of the wav files are on, off and status respectively then the transcription will have

<s> on </s> (on202)

<s> off </s> (off121)

It should maintained equal spacing between all the words in

Phase IV: The matching of total number of lines between the transcript file and the control file is done

Phase V: It checks the amount of training data and verifies that the number of n_tied_states is reasonable for those training data or not.

Phase VI: It verifies that all the transcript words are present in the dictionary or not.

Phase VII: It verifies all the phone present in the transcript file is appear in the phone list or not. Phoneme is the basic or the smallest unit of sound in any language. In the phone set that we have used to develop the speech recognition system, the phone set consists of 4 phonemes.

4.5.4.4 Recognition and Analysis

HTK provides a recognition tool called HVite that allows recognition using language models and lattices.HTK provides a recognition tool called HVite which uses the token passing algorithm described in the previous chapter to perform Viterbi-based speech recognition. HVite takes as input a network describing the allowable word sequences, a dictionary defining how each word is pronounced and a set of HMMs. It operates by converting the word network to a phone network and then attaching the appropriate HMM definition to each phone instance. Recognition can then be performed on either a list of stored speech files or on direct audio input. As noted at the end of the last chapter, HVite can support cross-word triphones and it can run with multiple tokens to generate lattices containing multiple hypotheses. It can also be configured to rescore lattices and perform forced alignments.

Once the HMM-based recognizer has been built, it is necessary to evaluate its performance. This is usually done by using it to transcribe some pre-recorded test sentences and match the recognizer output with the correct reference transcriptions. This comparison is performed by a tool called HResults which uses dynamic programming to align the two transcriptions and then count substitution, deletion and insertion errors.

For evaluating performance of speech system we have used the following equations [54]

$PC = (N - D - S)/N \times 100$ where PC gives word correction rate.

$PA = (N - D - S - I)/N \times 100$ where PA gives word accuracy rate.

Word Error Rate (WER) = 100% − Percentage Accuracy (PA)

Where, N = the number of words in test set.

D = the number of deletions.

S = the number of substitutions.

I = the number of insertions.

4.6 Receiving Command through Wi-Fi

The design of the whole system is implemented by following the modular approach for programming. Modules for distinct functionalities that the system must possess were builtstep by step before being put together. Coding was accompanied by regular debugging

and circuit stimulation so as to eliminate errors and to ensure smooth functioning of the system.

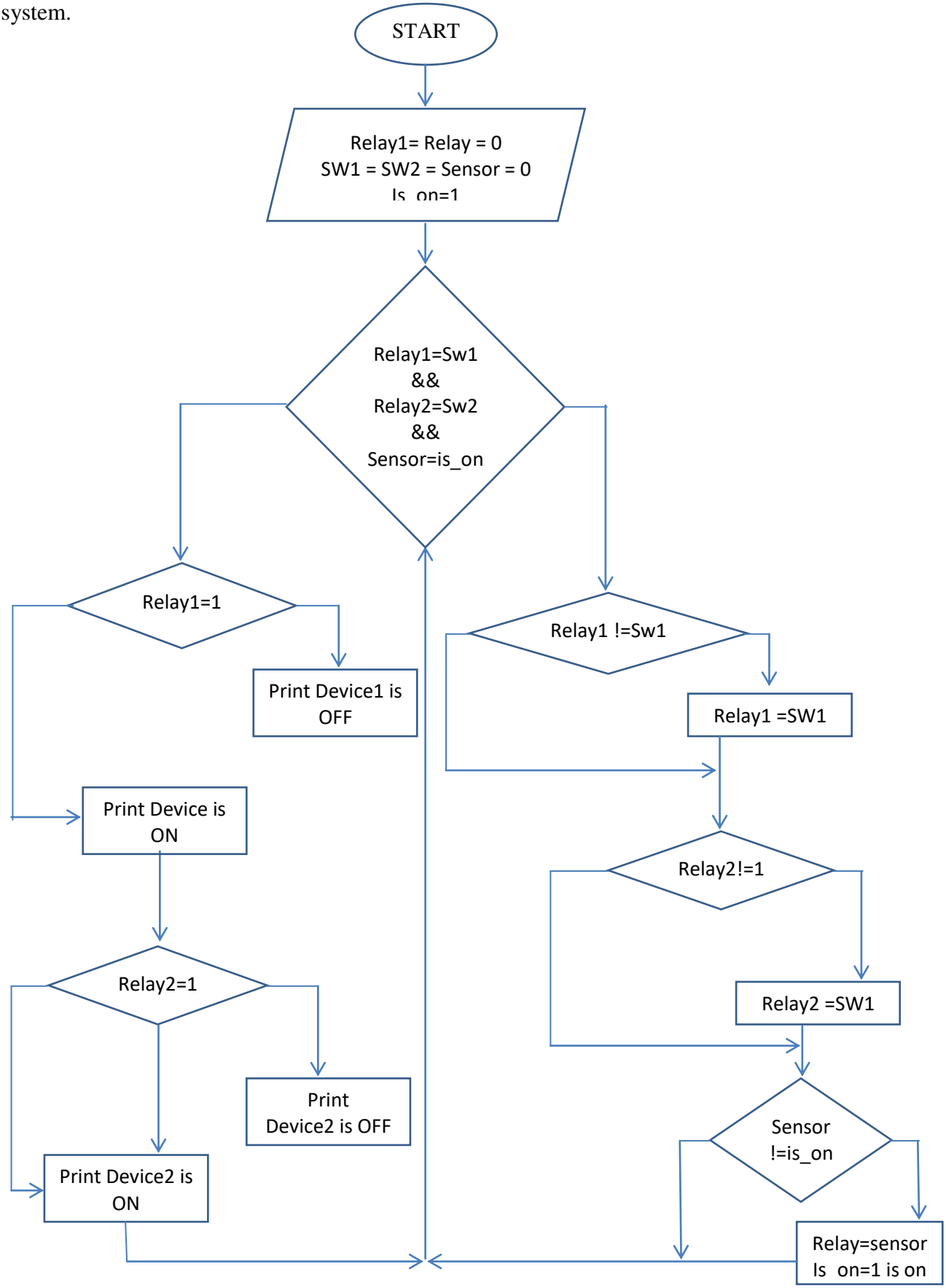

Figure 4.4: Flow Diagram of the designed System [47]

4.6.1 Observation

This IoT based system is very useful in automation as it provides the option of introducing different sensors based on the requirements of the user and automating the control of power supply of the load accordingly with respect to the sensory data. This system also provides some facilities as follows

- Remotely control the power supply of the load.
- Introduce automation on the basis of the parameters that are sensed by the sensor.
- Introduce other sensors in the system which automates the control of the power supply of the load depending upon the value of the sensory data, thereby making the device support modularity and generality.

It also leaves the control to override the flow of automation to the user and remotely control the device according to his or her will.

4.7 Database Design

The database is the main back end of the home server. In this system work, a special focus is given for designing the database as a common integrated platform to store the instructions sent by the user through web page as well as SMS. The database is designed using MySql. The name of the database is LABDA and it is connected to MySql server by using a data source name (DSN). There are basically three tables present in the LABDA database as DEVICE, USERS and a relational table ACTION by which the two table are connected in a many to many relationship.

4.7.1 E-R Diagram

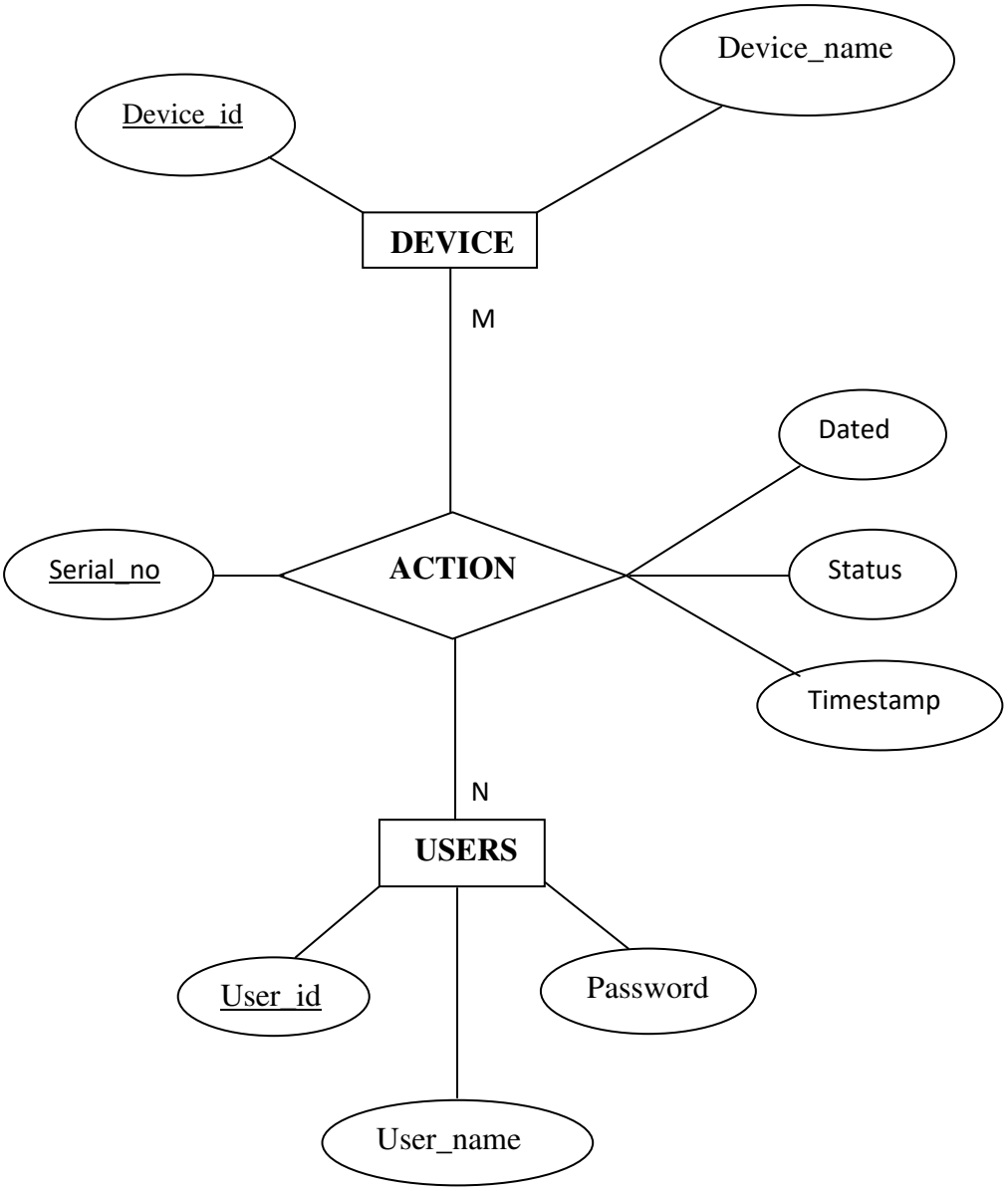

Figure 4.5: Entity- relationship diagram of the database

4.7.2 Relational Schema

The USERS table has three attributes fields – user_id, user_name and password where user_id is a primary key. In users table, user_id, user_name and staus are the attributes where user_id is the primary key. These two table are joined through a relational table action in a many to many relationship. Action table has serial_no,dated,timestamp,devic _id from device, user_id from users table and status attributes. In this table, serial_no, device_id,

user_id are primary key which are act as a composite key. Here, serial_no is auto-incremented because last command (having maximum serial_no) has to be given in the LASTACTION fields. So, LASTACTION table contains only one command at a time because a command is automatically removed when a new command is appeared according to the server programming. But, the LASTACTION table is not included in E-R diagram because it is only used in application level

At first, Database 'labda' is created and used, Then create the Device, Users, Action table. The Device table is shown in tabular form below-

Table 4.2: Device table

Field	Types	Null	Key	Defaults	Extra
Device_id	int (5)	NO	PRI	NULL	
Device_name	varchar(15)	YES		NULL	

Table 4.3: Users table

Field	Types	Null	Key	Defaults	Extra
User_name	varchar(10)	YES		NULL	
User_id	varchar(10)	NO	PRI	NULL	
Password	varchar(10)	YES		NULL	

Table 4.4: Action table

Field	Types	Null	Key	Defaults	Extra
serial_no	int(5)	NO	PRI	NULL	Auto_in
user_id	varchar(20)	NO	PRI		crement
dated	date	YES		NULL	
timestamp	varchar(20)	YES		NULL	
device_id	int(5)	NO	PRI	0	
status	varchar(20)	YES		NULL	

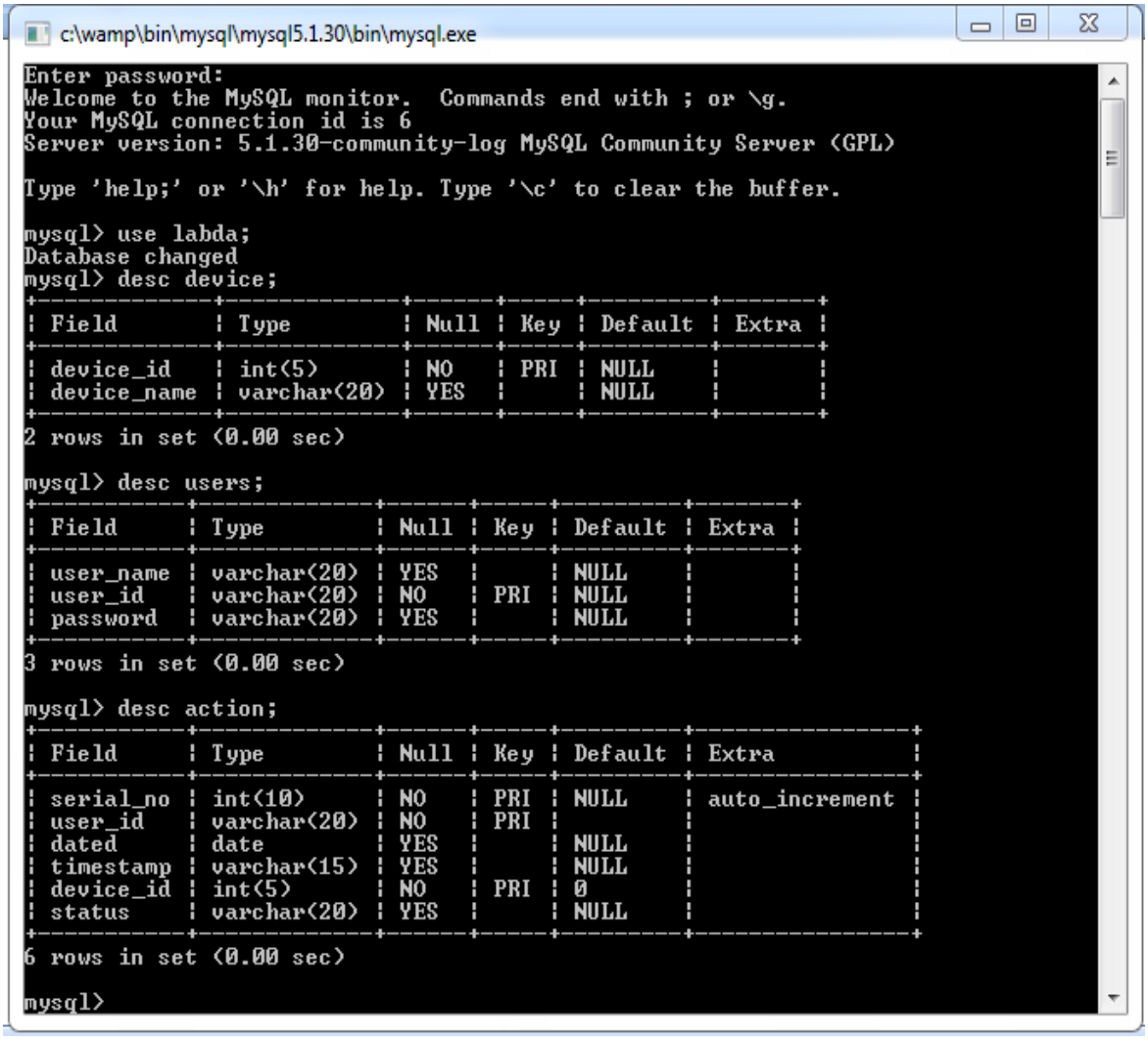

Figure 4.6: Screen shot of MySql having table Description

4.8 Integration of Common Platform

The main aim of this project work is to develop a common platform for both controlling mechanism in terms of an integrated database. At the beginning of the system when the command is arrived from Web page, GSM modem, wi-fi user and speech user in the server, the information details about the commands are stored only in the common database. It first stores in the ACTION table and then stores the command having maximum serial_no in the LASTACTION table as the serial_no of ACTION table is auto- incremented. The feedback or response to the users (Web user, SMS user, wi-fi user and speech user) has always arrived from the same server database. As a result, when the user controls devices via GSM modem, then the same information regarding the device control are available for Web users too. In this way a common integrated platform for the whole systems is maintained by the home server

4.9 Firing Command to AT89S52 from Database

The software package present in the server is mainly responsible for selecting the required command from the ACTION table. The last command stored in the database can be selected easily as the serial_no of ACTION table is auto-incremented. At every time when a message is stored in the ACTION table, then the action having maximum serial_no is stored in the LASTACTION table. The LASTACTION table is the middleware between the database and the microcontroller for firing process. So, the command is arrived at the port of home server from the LASTACTION table. The message is received by the port of the server machine using the port number which is already defined in the server programming. After that, the message is fired by the server program to MAX 232 through the DB-9 serial port using RS232 protocol. The MAX232 convertor fires the signal into the AT89S52 microcontroller and the microcontroller will be programmed in an appropriate way to understand this signal and thus convert it to an electrical signal and transmitted to the switch controlling the relay drivers which will switch the home appliances. The end result will be a simple action like: switching on a light.

4.10 AT89S52 PROGRAMMING DESIGN

The software package present in the hardware interface is responsible for switching the relay driver, controlling the LCD display, sending the switching status from the devices to serial port. The software package is designed by embedded C programming using the Keil compiler. The command/action sent by the user is already stored in a predefined manner such that the microcontroller programming can match it with its own. If it matches with incoming one then a function is called to reset the port as the instruction. As example, 'LIGHTON' is the command and it is matched with predefined command of programming, then, reset the port as 1 where the light is connected and simultaneously same is displayed in the LCD screen. Administrator of the system can see also the latest action done by the remote user over the system locally in the LCD. The Programming code has also designed in such a way that it can send the status of the switching devices when the users want. The embedded C programming is burnt into the microcontroller using the universal burner software.

4.11 HARDWARE INTERFACE DESIGN

The Hardware interface comprises of microcontroller AT89S52, TTL converter MAX232, an oscillator circuit, a voltage regulator circuit and a few discrete components. Microcontroller AT89S52 is at the heart of the circuit. It is a low power, high-performance, 8-bit microcontroller with 4KB of PEROM used as on-chip program memory, 128 bytes of RAM used as internal data memory, 32 individually programmable I/O lines divided into four 8-bit ports, two 16-bit programmable timers/counters, a five-vector two level interrupt architecture, on-chip oscillator and clock circuitry. Capacitor and resistor from the power-on reset circuit, while push-on reset switch is used for manual reset. Pin no. 7 to 14 of LCD is configured to get the result to be displayed from the Port pins P1.0 to P1.7 of the microcontroller respectively. Port pins P3.0 and P3.1 of the microcontroller are configured to get the input from 11 and send reply to12 pin of MAX232 respectively.

The figure 4.7.depicts the whole electronic circuit along with the relay driver board below-

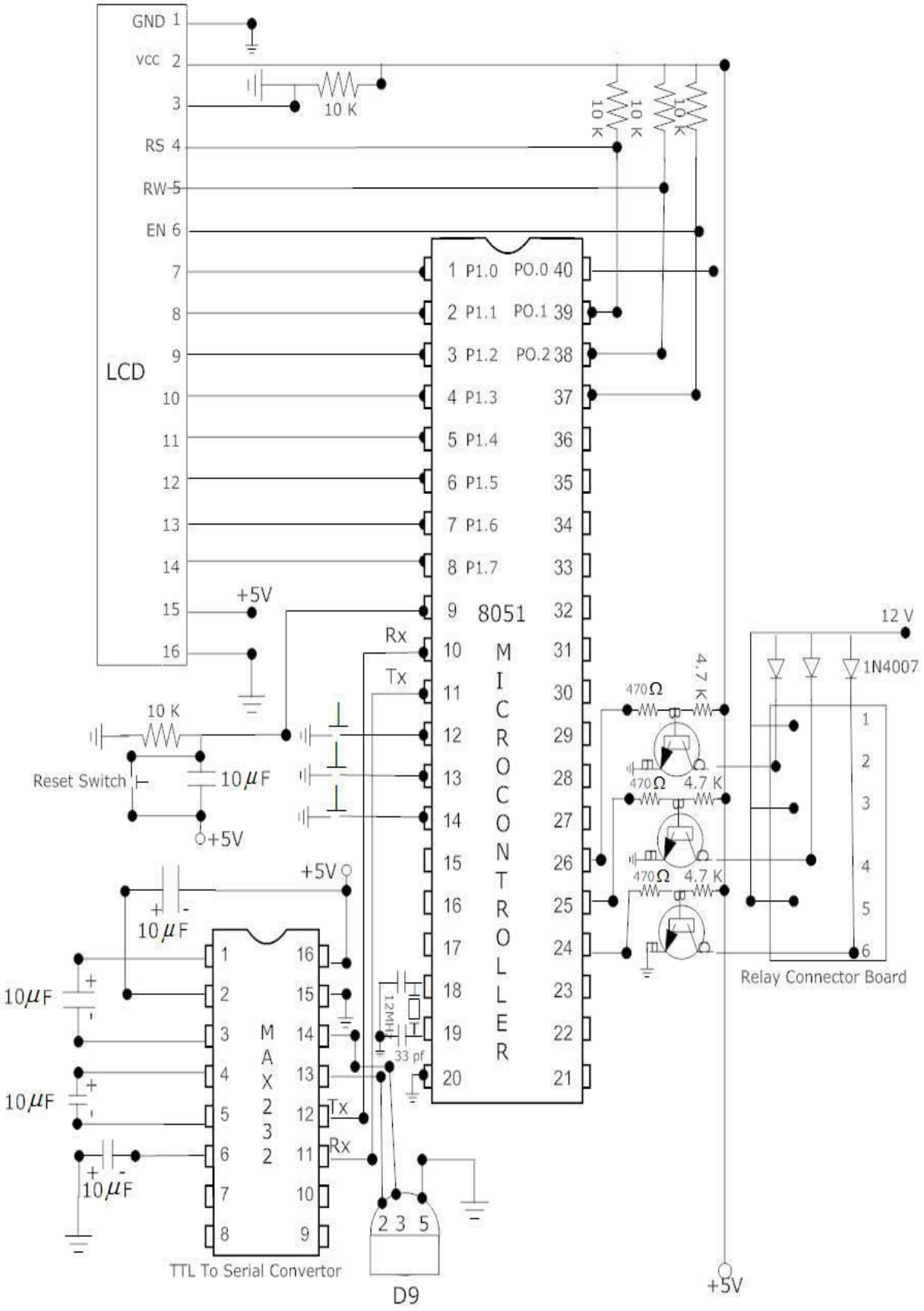

Figure 4.7: Circuit diagram of present hardware Interface

P2.3, P2.4 and P2.5 pins of the microcontroller are connected into the relays of the relay board through 2N2222 switching transistor and 1N4007 for removing the inverse current. Port pin P0.0 is connected to pin 4 of LCD to register selectors, port pin P0.1 is connected to pin 5 of LCD to read and write and port pin P0.2 is connected to pin 6 to enable the communication. The 18 and 19 pin of the microcontroller is connected into an oscillator circuit. The 2 and 3 pin of the DB-9 port is configured with 14 and 13 pin of MAX232 to give the input from the server.

The values and ratings of the component that are used in the implementation of the hardware interface in the table 4.5.

Table 4.5: Component values and ratings [55]

Serial No.	Name of the Components	Values	Ratings
1	AT89S52	8K,flash memory CMOS, 8 bit	0-20 MHz, 5V±15%
2	MAX232	RS232 to TTL Convertor	5V,8ma
3	7805	Three Terminal Positive Voltage Regulator	1A,5V
4	1N4007 diode	Rectifier Diode	Peak reverse voltage:1000v, peak rev current:5uA
5	Resistors	10K,4.7K, 470 ohm	¼ W, ±5% CARBON
6	Capacitors	33pF	Ceramic
7	Capacitors	10uF	16v, electrolytic
8	Crystal oscillator	11.0592 MHz	
9	LCD	Dot-matrix liquid Crystal Display	2.7 to 5.5 V, low power operation
10	2N2222	NPN Switching Transistor	High Current= max. 800mA,Low Voltage= max. 40 V
11	ESP8266	Wi-Fi Module	

4.12 Introducing the Security Features

In the proposed system, the security features should be included as it is a personal switching and controlling device system. Only authorised user can access the system from a remote place at any time. The user can control the system through communication mechanism based web page and SMS technology. So, in the proposed, to control appliances, web user (Internet user) has to log in with his/her user_id and password which are already stored in users table of server database. If it is correct, then only user can access the devices which are connected in the server end. Otherwise, the user machine will display as 'Invalid user_name and password' and not permit the user to access the system. Similarly, the SMS user has to send the user_id and password along with the command at the time of sending the action to the server. The server program extracts the user_id and password from that and performs the same process like the previous one. On the other hand, in case of speech communication, IVR system will work with the registered GSM modem either in server end or in user end. Along with the user has to know the command which is already trained and tested in the speech recognition system. From this point of view, this communication is more secured quite. For IoT communication, User has to operate the home appliances through a web interface designed previously and put their userid and password, otherwise they cannot work with the system. Therefore, only registered users can only work. So, all four communication mechanisms are secured in this way from the unauthorised users.