

Voice activated Smart Home Design and Implementation Alexa

R.Deepthi¹, S. Ashok Reddy², S. Mahaboob Basha³

¹P.G. Scholar, ²Assistant Professor, ³Head of the department
^{1,2,3} BRANCH . ECE (ES)

^{1,2,3} Geethanjali College of Engineering and Technology

Email id . ¹deepthirachagouni@gmail.com, ²singasaniashokreddy@gmail.com

Abstract.

For over 10 years, smart systems have had a significant impact in human day by day life. With the utilization of current advances, smart sensors, processors and telephones, the present smart home systems upgrade conveyed stimulation, family control, home vitality checking, home security and reconnaissance. The services gave incorporate voice controlled cautions, customized schedules with climate figure, and news report updates. It plans to improve their scholarly presentation by giving a superior nature of living. The system can likewise be worked as a PC and supports combination with different sensors through its General-purpose Input/Output pins (GPIO). Clients can alter their needs through this system. The displayed system incorporates family unit machine control, excitement system, lobby offices checking and home security. It is controlled by means of Telegram on smart telephones and Alexa Voice Service (AVS) and Amazon Developer Console bolstered by Amazon.

Keywords . raspberry Pi, Python, smart home, Telegram, Alexa, Alexa skill sets

INTRODUCTION

A Voice Command System essentially means a system that processes voice as an input, decodes or understands the meaning of that input processes it and generates an appropriate voice output. Any voice

command system need three basic components which are speech to text converter, query processor and a text to speech converter. Voice has been a very integral part of communication nowadays. Since, it is faster to process sound and voices than to process written text, hence voice command systems are omnipresent in computer devices. There have been some very good innovations in the field of speech recognition.

Some of the latest innovations have been due to the improvements and high usage of big data and deep learning in this field. These innovations have attributed to the technology industry using deep learning methods in making and using some of the speech recognition systems, Google was able to reduce word error rate by 6% to 10% relative, for the system that had the word error rate of 17% to 52%. Text to speech conversion is the process of converting a machine recognized text into any language which could be identified by a speaker when the text is read out loud. It is two step processes which is divided into front end and back end. First part is responsible for converting numbers and abbreviations to a written word format. This is also referred to as normalization of text. Second part involves the signal to be processed into an understandable one.

Speech Recognition is the ability of machine for instance a computer to

understand words and sentences spoken in any language. These words or sentences are then converted to a format that could be understood by the machine. Speech recognition is basically implemented using vocabulary systems. A speech recognition system may be a Small Vocabulary-many user system or a Large Vocabulary- small user system.

RELATED WORK.

There are a lot of commercial companies which are into Home Automation like Schneider, Home Automation Inc., Logitech, Honeywell, etc., They have made successful attempts in integrating the home appliances to a central server and controlling them using the same. Certain companies include Security also. Natural Language Processing is a real difficult part in this domain. Many legends including Google, Amazon, Apple, Microsoft have NLP in action in the form of Google Now, Alexa, Siri and Cortana respectively.

The NLP is based on machine learning which breaks down the words to the smallest unit called phonemes and compare it with the pre-recorded speech which had been trained previously. Speech to Text also comes under the NLP. The primary aim of NLP is to make the human to computer interaction easier by making it more like human to human interaction. By the term human to human interaction, we mean the communication through native languages like English, French, Spanish, etc., The computers usually communicate through artificial languages like C, C++, Java, Python, etc., Our project aims to combine all of this and create a Virtual Assistant which would be handy in Home Automation also.

Devices like Amazon Alexa, Google Home are existing systems which are dedicated personal assistants which comes with a lot

of features. But they are equally expensive which cannot be affordable by everyone. This system is developed to be less expensive than the existing system providing features that they provide along with add-ons.

Few years back, Facebook CEO Mark Zuckerberg tried developing an Artificially Intelligent personal assistant named 'Jarvis' inspired from the movie 'Iron Man'. He quite succeeded in the project. It took so much effort and so much coding to make it possible. These are some of the related works in the field which inspired us to start with this project.

This work presents the survey on virtual assistant, google assistant, siri, cortana, alexa. This technology focuses in worldwide like smart phone, laptop, computer, etc. The aim of this work is to test voice recognition and contextual understanding between user and human interaction. For completing this aim they analyses on the voice recognition and human interaction. To recognize the voice, it was necessary to know that the Virtual Assistant regularly understood the words that the users are referring to. The idea of giving a feedback or estimation.

Alexa sounds and music are good, Alexa is not easy with basic questions, Core is not basic. This was the first issue of human interaction that survey found that the hand-free connection to the neglected user was a real use case. In these special circumstances, the disruption ability was a huge obstacle. When they are given requests that they are asked to select or select an option by dragging the touch screen using the sensor on the outside of the intelligent virtual assistive screen or by using the discourse.

This work includes voice activated smart home design and implementation. The aim of this paper is to design and implement a quality home assistant that can be controlled by voice and text commands. Sound commands are captured by the Amazon Voice Service, also known as alexia, is processed via Amazon Developer Console. Before reaching the control terminal, Alexa skills and Ngrok services can access and manage these services via raspberry pi user Amazon Developer Console. This work present voice enabled home automation using Amazon echo. Home Automation enabled current paper with Amazon Paper using Amazon Echo. In this field, the research focuses on how to create a robust, economical system with the latest developments available, and smart non-smart and large-scale applications. Amazon Power N is used. And to provide smart features for the smart phone, Raspberry Pi3 is used as a hardware component. They describe different components of their product and work effectively to switch their systems and shut down applications.

I discussed a smart home system will be needed in the near future, with continually improved Google homepage and Amazon Echo. This work introduces the design and implementation of a low-cost Voice Active Smart Home system that can be combined with many basic subsystems and individual needs can be met. With Alexa Skills Kits and Raspberry Pi, each device can be controlled anywhere without having to interact directly with them. It is full with the possibility of developing simple and user-friendly Alexa Skills Kit and Amazon Ib Services. Future tasks for upgrading Smart Home systems can be done to accommodate large numbers of users.

This work present the voice controlled personal assistant using raspberry

pi. The purpose of this work can serve many as the purpose of implementing the voice command system as an intelligent personal assistant. In this system the user started the system and used the microphone to send it to the input. Then users receive input and the computer is given it for further processing, then the input volume is conveyed to the text converter, which converts the audio input text output from the computer and after that the keywords of that content are analyzed and searched, the text searches the text to match the system word, and once the keyword is matched then it give required output. This output is in text format and then converted to speech output using speech converter which uses optical character recognition system.

This article present the next generation of virtual assistant (Microsoft cortana, apple siri, amazon alexa and google home).The goals of this work is the realization of natural dialogue between the human and machine for this purpose multimodal is used. Multimodal dialogue system which process toe or more combination of user input modes such as image, video, touch, head and body movement in order to design next generation of VPAS model. This system can support to wide range of application such as business enterprises, education, government, health care. In this work, the Amazon Voice Service, also called Alexa is not it Voice recognition work only in this smart phone, but the brains of millions of tools Amazon Echo - Smart Home Device has developed Amazon uses the proposed Smart Home System Amazon because of its voice recognition system and the value of proto typing, voice services .I have a general understanding of personal assistants. A person (or agent) who is able to provide specific help in the context of given time and given action.

After the various study and literature survey, communication system is one of the active sectors Companies use to design and improve their new system. According to CMM Research, 2030, millions of people using "voice" to interact with the machine and to voice-play Smart phones will become part and portions of smart phones Specs, Home Centers, Kitchen Appliances, TVs, Game Console.

INTRODUCTION TO EMBEDDED SYSTEMS

3.1 Application Areas

Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on.

Consumer appliances At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc. Today's high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now becoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general-purpose tasks such as playing games and word processing.

Office automation. The office automation products using em embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

Industrial automation. Today a lot of industries use embedded systems for process control. These include

pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

Medical electronics. Almost every medical equipment in the hospital is an embedded system. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners, equipment used in blood analysis, radiation, colonoscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

Computer networking. Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end systems (desktop computers) we use to access the networks, are embedded systems

Telecommunications. In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Disassemblers (PADs), satellite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

Wireless technologies. Advances in mobile communications are paving way for many interesting applications using embedded systems. The mobile phone is one of the marvels of the last decade of the 20th century. It is a very powerful embedded system that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centers are also powerful embedded systems.

Insemination. Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all embedded systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio communication test set etc. are embedded systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming portable facilitating easy testing and measurement in the field by field-personnel.

Security. Security of persons and information has always been a major issue.

We need to protect our homes and offices, and also the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of the processors that are manufactured end up in~ embedded systems. Embedded systems find applications in every industrial segment-consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

Finance. Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded as Any Time Money) machines. Smart card, of the size of a credit card, has a small micro-controller and memory, and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society. Well, the list goes on. It is no exaggeration to say that eyes wherever we go, we can see, or at least feel, the work of an embedded system.

HARDWARE IMPLEMENTATION OF THE PROJECT

This chapter briefly explains about the Hardware Implementation of the project. It discusses the design and working of the

design with the help of block diagram and circuit diagram and explanation of circuit diagram in detail. It explains the features, timer programming, serial communication, interrupts of atmega328 microcontroller. It also explains the various modules used in this project.

Project Design

The implementation of the project design can be divided in two parts.

- Hardware implementation
- Firmware implementation

Hardware implementation deals in drawing the schematic on the plane paper according to the application, testing the schematic design over the breadboard using the various IC's to find if the design meets the objective, carrying out the PCB layout of the schematic tested on breadboard, finally preparing the board and testing the designed hardware.

The project design and principle are explained in this chapter using the block diagram and circuit diagram. The block diagram discusses about the required components of the design and working condition is explained using circuit diagram and system wiring diagram.

POWER SUPPLY

The device is powered by a 5V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what you connect to it. We have found that purchasing a 1.2A (1200mA) power supply from a reputable retailer will provide you with ample power to run your Raspberry Pi.

Typically, the model B uses between 700-1000mA depending on what peripherals are connected, the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1 Amp. If you need to connect a USB device that will take the power requirements

above 1 Amp, then you must connect it to an externally-powered USB hub.

The power requirements of the Raspberry Pi increase as you make use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely, distributed across all the pins, an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250mA, and keyboards and mice can take as little as 100mA or over 1000mA! Check the power rating of the devices you plan to connect to the Pi and purchase a power supply accordingly.

BACKPOWERING

Back powering occurs when USB hubs do not provide a diode to stop the hub from powering against the host computer. Other hubs will provide as much power as you want out each port. Please also be aware that some hubs will backfeed the Raspberry Pi. This means that the hubs will power the Raspberry Pi through its USB cable input cable, without the need for a separate micro-USB power cable, and bypass the voltage protection. If you are using a hub that backfeeds to the Raspberry Pi and the hub experiences a power surge, your Raspberry Pi could potentially be damaged.

USB

OVERVIEW

The Raspberry Pi Model B is equipped with two USB2.0 ports. These are connected to the LAN9512 combo hub/Ethernet chip IC3, which is itself a USB device connected to the single upstream USB port on BCM2835.

On the Model A, the single USB2.0 port is directly wired to BCM2835.

The USB ports enable the attachment of peripherals such as keyboards, mice, webcams that provide the Pi with additional functionality.

There are some differences between the USB hardware on the Raspberry Pi and the USB hardware on desktop computers or laptop/tablet devices.

The USB host port inside the Pi is an On-The-Go (OTG) host as the application processor powering the Pi, BCM2835, was originally intended to be used in the mobile market. i.e. as the single USB port on a phone for connection to a PC, or to a single device. In essence, the OTG hardware is simpler than the equivalent hardware on a PC.

OTG in general supports communication to all types of USB device, but to provide an adequate level of functionality for most of the USB devices that one might plug into a Pi, the system software has to do more work.

SUPPORTED DEVICES

In general, every device supported by Linux is possible to use with the Pi, subject to a few caveats detailed further down. Linux has probably the most comprehensive driver database for legacy hardware of any operating system (it can lag behind for modern device support as it requires open-source drivers for Linux to recognise the device by default).

If you have a device and wish to use it with a Pi, then plug it in. Chances are that it'll "just work". If you are running in a graphical interface (such as the LXDE desktop environment in Raspbian), then it's likely that an icon or similar will pop up announcing the new device.

If the device doesn't appear to work, then refer to the Troubleshooting section.

GENERAL LIMITATIONS

The OTG hardware on Raspberry Pi has a simpler level of support for certain devices, which may present a higher software processing overhead. The Raspberry Pi also

has only one root USB port. all traffic from all connected devices is funnelled down this bus, which operates at a maximum speed of 480mbps.

The USB specification defines three device speeds - Low, Full and High. Most mice and keyboards are Low-speed, most USB sound devices are Full-speed and most video devices (webcams or video capture) are High-speed.

Generally, there are no issues with connecting multiple High-speed USB devices to a Pi.

The software overhead incurred when talking to Low- and Full-speed devices means that there are soft limitations on the number of simultaneously active Low- and Full-speed devices. Small numbers of these types of devices connected to a Pi will cause no issues.

INSTALLING OPERATING SYSTEM IMAGES

How to install a Raspberry Pi Operating System image on an SD card. You will need another computer with an SD card reader to install the image. We recommend most users download NOOBS which is designed to be very easy to use. However more advanced users looking to install a particular image should use this guide.

DOWNLOAD THE IMAGE

Official images for recommended Operating Systems are available to download from the Raspberry Pi

website. raspberrypi.org/downloads.

Alternative distributions are available from third party vendors.

WRITING AN IMAGE TO THE SD CARD

With the image file of the distribution of your choice, you need to use an image writing tool to install it on your SD card.

See our guide for your system.

INSTALLING OPERATING SYSTEM IMAGES ON LINUX

Please note that the use of the dd tool can overwrite any partition of your machine. If you specify the wrong device in the instructions below you could delete your primary Linux partition. Please be careful.

- Run `df -h` to see what devices are currently mounted.
- If your computer has a slot for SD cards, insert the card. If not, insert the card into an SD card reader, then connect the reader to your computer.
- Run `df -h` again. The new device that has appeared is your SD card. The left column gives the device name of your SD card, it will be listed as something like `/dev/mmcblk0p1` or `/dev/sdd1`. The last part (p1 or 1 respectively) is the partition number but you want to write to the whole SD card, not just one partition. Therefore you need to remove that part from the name (getting, for example, `/dev/mmcblk0` or `/dev/sdd`) as the device for the whole SD card. Note that the SD card can show up more than once in the output of `df`, it will do this if you have previously written a Raspberry Pi image to this SD card, because the Raspberry Pi SD images have more than one partition.
- Now that you've noted what the device name is, you need to unmount it so that files can't be read or written to the SD card while you are copying over the SD image.
- Run `umount /dev/sdd1`, replacing `sdd1` with whatever your SD card's device name is (including the partition number).
- If your SD card shows up more than once in the output of `df` due to having multiple partitions on the SD card, you should unmount all of these partitions.
- In the terminal, write the image to the card with the command below, making sure you replace the input

file `if=` argument with the path to your .img file, and the `/dev/sdd` in the output file `of=` argument with the right device name. This is very important, as you will lose all data on the hard drive if you provide the wrong device name. Make sure the device name is the name of the whole SD card as described above, not just a partition of it, for example `sdd`, not `sdds1` or `sddp1`, or `mmcblk0`, not `mmcblk0p1`.

`dd bs=4M if=2014-06-20-wheezy-raspbian.img of=/dev/sdd`

- Please note that block size set to 4M will work most of the time, if not, please try 1M, although this will take considerably longer.
- Also note that if you are not logged in as root you will need to prefix this with `sudo`.
- The `dd` command does not give any information of its progress and so may appear to have frozen, it could take more than five minutes to finish writing to the card. If your card reader has an LED it may blink during the write process. To see the progress of the copy operation you can run `pskill -USR1 -n -x dd` in another terminal, prefixed with `sudo` if you are not logged in as root. The progress will be displayed in the original window and not the window with the `pskill` command, it may not display immediately, due to buffering.
- Instead of `dd` you can use `dcfldd`, it will give a progress report about how much has been written.
- You can check what's written to the SD card by `dd`-ing from the card back to another image on your hard disk, and then running `diff` (or `md5sum`) on those two images. There should be no difference.
- Run `sync`, this will ensure the write cache is flushed and that it is safe to unmount your SD card.

- Remove the SD card from the card reader.

APT

The easiest way to manage installing, upgrading, and removing software is using APT (Advanced Packaging Tool) which comes from Debian. If a piece of software is packaged in Debian and works on the Raspberry Pi's ARM architecture, it should also be available in Raspbian. To install or remove packages you need root user permissions, so your user needs to be in sudoers or you must be logged in as root. Read more about users and root.

To install new packages, or update existing ones, you will need an internet connection.

Note that installing software uses up disk space on your SD card, so you should keep an eye on disk usage and use an appropriately sized SD card.

Also note that a lock is performed while software is installing, so you cannot install multiple packages at the same time.

SOFTWARE SOURCES

APT keeps a list of software sources on your Pi in a file at /etc/apt/sources.list. Before installing software, you should update your package list with apt-get update.

sudo apt-get update

INSTALLING A PACKAGE WITH APT

sudo apt-get install tree

Typing this command should inform the user how much disk space the package will take up and asks for confirmation of the package installation. Entering Y (or just hitting Enter, as yes is the default action) will allow the installation to occur. This can be bypassed by adding the -y flag to the command.

sudo apt-get install tree -y

Installing this package makes tree available for the user.

USING AN INSTALLED PACKAGE

tree is a command line tool which provides a visualisation of the directory structure of the current directory, and all it contains.

- Typing tree runs the tree command. For example.

tree

```
..
├── hello.py
├── games
│   ├── asteroids.py
│   ├── pacman.py
│   ├── README.txt
│   └── tetris.py
```

- Typing man tree gives the manual entry for the package tree
- Typing whereis tree shows where tree lives.

tree. /usr/bin/tree

UNINSTALLING A PACKAGE WITH APT

REMOVE

You can uninstall a package with apt-get remove.

sudo apt-get remove tree

The user is prompted to confirm the removal. Again, the -y flag will auto-confirm.

PURGE

You can also choose to completely remove the package and its associated configuration files with apt-get purge.

sudo apt-get purge tree

UPGRADING EXISTING SOFTWARE

If software updates are available, you can get the updates with sudo apt-get update and install the updates with sudo apt-get upgrade, which will upgrade all of your packages. To upgrade a specific package, without upgrading all the other out-of-date packages at the same time, you can use sudo apt-get install somepackage (which may be useful if you're low on disk space or you have limited download bandwidth).

SEARCHING FOR SOFTWARE

You can search the archives for a package with a given keyword with apt-cache search. apt-cache search locomotive
sl - Correct you if you type `sl' by mistake
You can view more information about a package before installing it with apt-cache show.

```
apt-cache show sl
Package. sl
Version. 3.03-17
Architecture. armhf
Maintainer. Hiroyuki Yamamoto
<yama1066@gmail.com>
Installed-Size. 114
Depends. libc6 (>= 2.4), libncurses5 (>= 5.5-5~), libtinfo5
Homepage. http://www.tkl.iis.u-
tokyo.ac.jp/~toyoda/index_e.html
Priority. optional
Section. games
Filename. pool/main/s/sl/sl_3.03-
17_armhf.deb
Size. 26246
SHA256.
42dea9d7c618af8fe9f3c810b3d551102832b
f217a5bcdba310f119f62117dfb
SHA1.
b08039acccecd721fc3e6faf264fe59e56118e
74
MD5sum.
450b21cc998dc9026313f72b4bd9807b
Description. Correct you if you type `sl' by
mistake
Sl is a program that can display animations
aimed to correct you
if you type 'sl' by mistake.
SL stands for Steam Locomotive.
```

UPDATING AND UPGRADING RASPBIAN

There are two steps to upgrading. First, run `sudo apt-get update` in order to synchronise the database of available software packages and the versions available. Next, run `sudo apt-get upgrade` which will cause any packages with newer versions available to be updated.

Generally speaking, doing this regularly will keep your installation up to date, in other words, it will be equivalent to the latest released image. However, there are occasionally times where a change is made in the Foundation's Raspbian image that require your intervention to reproduce. A recent example is the addition of the `xserver-xorg-video-fbturbo` X.Org driver to the standard image, this requires users running older images to manually install the package to benefit from it. Cases like this are documented on the relevant image update announcement on the Raspberry Pi blog.

UPDATING THE KERNEL AND FIRMWARE

The kernel and firmware are installed as a Debian package, and so will also get updates when using the procedure above. These packages are updated infrequently (after extensive testing), if you want to try more recent experimental software, it's also easy to update to the latest available version using `rpi-update`. This is pre-installed on the current Raspbian image, so you can just use `sudo rpi-update` to try the latest firmware, this will sometimes be suggested when troubleshooting. If you receive errors about invalid certificates, then run `sudo apt-get update & sudo apt-get install rpi-update` to upgrade to the latest `rpi-update` version.

RUNNING OUT OF SPACE

When running `sudo apt-get upgrade`, it will show how much data will be downloaded and how much space it will take up on the SD card. It's worth checking with `df -h` that you have enough disk space free, as unfortunately apt will not do this for you. Also be aware that downloaded package files (`.deb` files) are kept in `/var/cache/apt/archives`. You can remove

these in order to free up space with sudo apt-get clean.

RASPI-CONFIG

raspi-config is the Raspberry Pi configuration tool written and maintained by Alex Bradbury. It targets Raspbian.

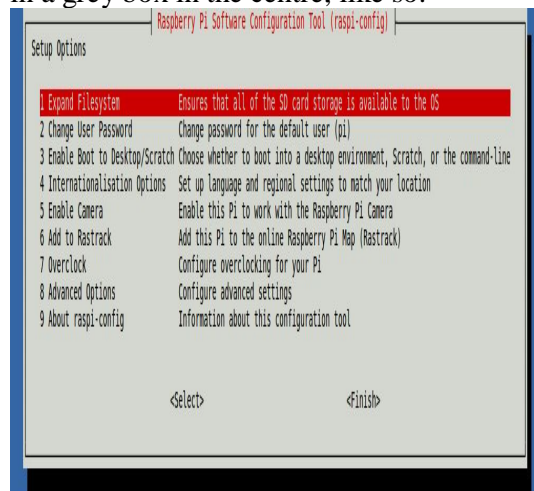
USAGE

You will be shown raspi-config on first booting into Raspbian. To open the configuration tool after this, simply run the following from the command line.

```
sudo raspi-config
```

The sudo is required because you will be changing files that you do not own as the pi user.

You should see a blue screen with options in a grey box in the centre, like so.



It has the following options available.

Raspberry Pi Software Configuration Tool (raspi-config)

Setup Options

1. Expand Filesystem- Ensures that all of the SD card storage is available to the OS
2. Change User Password -Change password for the default user (pi)
3. Enable Boot to Desktop/Scratch- Choose whether to boot into a desktop environment, Scratch, or the command line

4. Internationalisation Options - Set up language and regional settings to match your location
5. Enable Camera-Enable this Pi to work with the Raspberry Pi Camera
6. Add to Rastrack - Add this Pi to the online Raspberry Pi Map (Rastrack)
7. Overclock- Configure overclocking for your Pi
8. Advanced Options- Configure advanced settings
9. About `raspi-config`- Information about this configuration tool

<Select><Finish>

MOVING AROUND THE MENU

Use the up and down arrow keys to move the highlighted selection between the options available. Pressing the right arrow key will jump out of the options menu and take you to the <Select> and <Finish> buttons. Pressing left will take you back to the options. Alternatively, use the Tabkey to switch between these.

Note that in long lists of option values (like the list of timezone cities), you can also type a letter to skip to that section of the list. For example, entering L will skip you to Lisbon, just two options away from London, to save you scrolling all the way through the alphabet.

WHAT RASPI-CONFIG DOES

Generally speaking, raspi-config aims to provide the functionality to make the most common configuration changes. This may result in automated edits to /boot/config.txt and various standard Linux configuration files. Some options require a reboot to take effect. If you changed any of those, raspi-config will ask if you wish to reboot now when you select the <Finish> button.

SUDO

You will not normally log into to the computer as root, but can instead use the sudo command to provide access as the superuser. If you log into your Raspberry Pi as the pi user then you are logging in as a normal user. You can run commands as the root user by using the sudo command before the program you want to run. For example if you want to install additional software on Raspbian then you normally use the apt-get tool. To be able to update the list of available software then you need to prefix the apt-get command with sudo. sudo apt-get update. You can also run a superuser shell by using sudo su. When running commands as a superuser there is nothing to protect against mistakes that could damage the system. It is like disabling the safety guards on a machine. It makes it easier to access the insides, but the risk of damage is far greater. It is recommended that you only run commands as the superuser when required and to exit a superuser shell when it is no longer needed.

KERNEL

The Raspberry Pi kernel is stored in GitHub and can be viewed at github.com/raspberrypi/linux, it follows behind the main linux kernel. The main Linux kernel is continuously updating, we take long-term releases of the kernel, which are mentioned on the front page, and integrate the changes into the Raspberry Pi kernel. We then create a 'next' branch which contains an unstable port of the kernel, after extensive testing and discussion we push this to the main branch.

- Updating your kernel
- Building a new kernel
- Configuring the kernel
- Applying patches to the kernel
- Getting the kernel headers

GETTING YOUR CODE INTO THE KERNEL

There are many reasons you may want to put something into the kernel.

- You've written some Raspberry Pi-specific code that you want everyone to benefit from
- You've written a generic Linux kernel driver for a device and want everyone to use it
- You've fixed a generic kernel bug
- You've fixed a Raspberry Pi-specific kernel bug

Initially you should fork the Linux repository and clone that on your build system, this can be either on the Raspberry Pi or on a Linux machine you're cross-compiling on. You can then make your changes, test them, and commit them into your fork.

Next, depending upon whether the code is Raspberry Pi-specific or not.

For Pi-specific changes or bug fixes, submit a pull request to the kernel.

For general Linux kernel changes (i.e. a new driver) these need to be submitted upstream first. Once they've been submitted upstream and accepted, submit the pull request and we'll receive it.

Building Qt 5 on Raspberry Pi

This is a HOW TO guide for building Qt 5 for the Raspberry Pi, and building and deploying Qt 5 apps using Qt Creator. This guide will be using Raspbian "Wheezy", a Debian based distro designed for the Raspberry Pi. This guide also assumes the use of Linux or UNIX on the workstation side.

Install a Toolchain

To build on the Raspberry Pi we need a cross-compile toolchain. The toolchain will contain compilers, linkers and other tools that run on the host workstation but create executables for the target Raspberry Pi.

For embedded development, one normally uses a vendor-supplied toolchain, but in the

case of the Raspberry Pi, there is no official vendor supplied toolchain. There are several generic ARM toolchains that will suffice, however, we have chosen to use the same one the *bakeqtpi* script uses. This is a Linaro based toolchain for the ARMv6 platform with hard floating-point support.

Alternatively, we could have built our own toolchain. We will create a working directory to use named "raspberry". Our first step is to get and install a cross compiling toolchain.

```
$ mkdir ~/raspberry
$ cd ~/raspberry
$ wget http://swapt.smt.eu/gcc-4.7-linaro-rpi-gnueabi.hf.tbz
$ tar xvf gcc-4.7-linaro-rpi-gnueabi.hf.tbz
```

ETHERNET MODULE

This board utilizes the new Microchip ENC28J60 Stand- Alone Ethernet Controller IC featuring a host of features to handle most of the network protocol requirements. The board connects directly to most microcontrollers with a standard SPI interface with a transfer speed of up to 20 MHz. This module contains what you need to enable ethernet on your next microcontroller project. It includes a ENC28J60 ethernet controller, an RJ45 socket with link/activity lights and integrated transformer. This ethernet module enables you to connect a particular embedded device (equipped with SPI support) on to a network. By using this Ethernet module (along with a microcontroller running a small TCP/IP stack) applications like Embedded Web server can be easily developed.

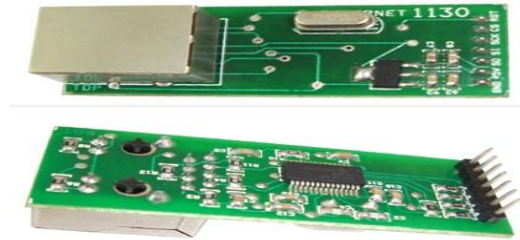


Fig: Ethernet Module

Use this module to enable Ethernet interface for your product. It works with any microcontroller operating at 3.3V or 5V. This module works at 5V and is compatible with 5V interface lines. Use SPI process to interface with it. Host web server, ping the module or add it to home automation via internet. Heart of this module is ENC28J60 Ethernet controller from Microchip. This board is assembled with all components as shown in the picture.

The circuit board includes all required components for the ethernet controller, plus a 3.3V Voltage Regulator and a RJ-45 jack with integrated transformer and built-in Link and Activity LEDs for connection to an Ethernet Local Area Network.

Features

1. Microchip ENC28J60 Ethernet Chip supporting Microchip's TCP/IP protocol stack, with IPv4,
2. UDP, TCP, DHCP, ICMP, FTP, and HTTP support
3. On board voltage regulator for 3V
4. Compatible with all controllers having an SPI interface, SPI interface takes only few pins to add Ethernet interface to your microcontroller project
5. Suitable for Either 5V or 3.3V Interface
6. LAN connector with build in transformer
7. Link and Activity LED's on Ethernet connector
8. 8 pin standard interface connector with spacing of 0.1" (2.54 mm) row pins
9. IEEE 802.3 Compatible with Integrated MAC and 10Base-T PHY

10. Fully Compatible with 10/100/1000Base-T Networks
11. Supports One 10Base-T Port with Automatic Polarity Detection and Correction
12. Supports Full and Half-Duplex modes
13. Programmable Automatic Retransmit on Collision
14. Programmable Padding and CRC Generation
15. Programmable SPI Interface with Clock Speeds Up to 20 MHz

Applications

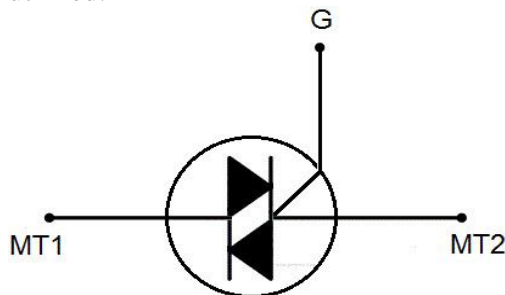
- a. Remote control and monitoring
- b. Data capture and logging
- c. Industrial Automation
- d. Building Automation
- e. Appliance remote management
- f. Robotics
- g. Microcontroller Research and Development
- h. TCP/IP Research and Development

Specification

- a. Board Size. 2.3" x 0.94" (60x24mm)
- b. Operating Voltage. +5V DC
- c. Current Consumption. 160mA
- d. Pin Spacing. 0.1" (2.54mm)
- e. Output wire length. 12"(304mm)

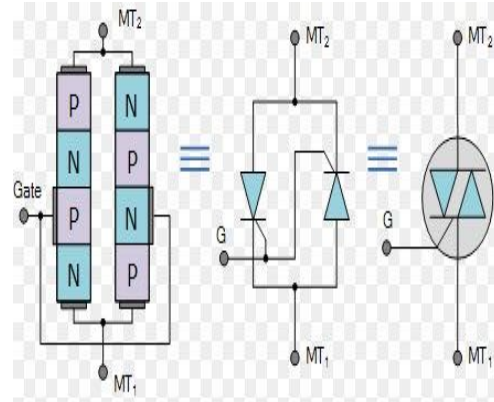
TRIAC

The TRIAC is an ideal device used for the AC switching applications. This can control the flow of current over both halves of an alternating cycle. The Thyristor can control only over the one-half of a cycle. The other remaining half no conduction occurs and accordingly only half the waveform can be utilized.



Construction of TRIAC

The TRIAC consists of four layers like PNP is in the positive direction and the negative direction consists of NPN as we can see in the figure. The three-terminal bidirectional device blocks the current in the OFF state and it will act as an open circuit switch.



The four triggering modes of operations can operate by using the I-V characteristics of TRIAC.

Quadrant I operation. VMT2, positive, VG1 positive

Quadrant II operation. VMT21 positive, VG1 negative

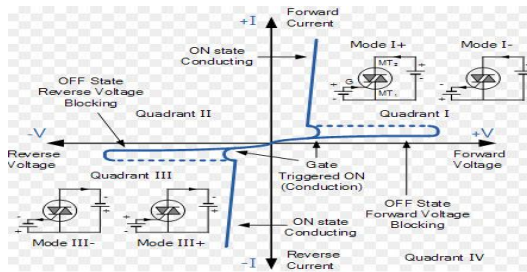
Quadrant III operation. VMT21 negative, VG1 negative

Quadrant IV operation. VMT21 negative, VG1 positive

where VMT21 and VG1 are the voltages of the terminal of MT2 and gate with respect to terminal MT1.

I-V Characteristics

The following figure shows the typical TRIAC characteristics. The Triode For Alternating Current consists of ON and OFF state characteristics which are similar to the SCR. Now the characteristics are applicable to both the voltages of positive and negative.



I-V Characteristics of TRIAC

Advantages of TRIAC

- In the TRIACs, there will be single gate control conduction in both the directions.
- If the voltage is decreased to zero the TRIAC turns OFF
- Basically, it is bidirectional device and in the both the directions it will conduct.
- High voltage rating are available

Disadvantages

- It doesn't fire symmetrically on both sides of the waveforms
- It doesn't appropriate for the DC power
- It has very high switching delay.

Applications

The TRIACs are using many electrical switching applications like

- The electric fan speed control
- Smaller motor controls
- Domestic light dimmer
- Control of small AC powered domestic appliances

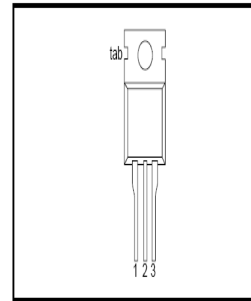
BT136 Description

Glass passivated triacs in a plastic envelope, intended for use in applications requiring high bidirectional transient and blocking voltage capability and high thermal cycling performance. Typical applications include motor control, industrial and domestic lighting, heating and static switching.

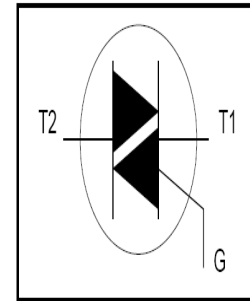
PIN	DESCRIPTION
1	main terminal 1
2	main terminal 2
3	gate
tab	main terminal 2

Pin description

PIN CONFIGURATION



SYMBOL



DC Motor

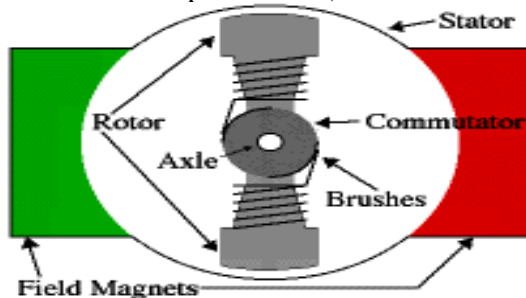
DC motors are configured in many types and sizes, including brush less, servo, and gear motor types. A motor consists of a rotor and a permanent magnetic field stator. The magnetic field is maintained using either permanent magnets or electromagnetic windings. DC motors are most commonly used in variable speed and torque.

Motion and controls cover a wide range of components that in some way are used to generate and/or control motion. Areas within this category include bearings and bushings, clutches and brakes, controls and drives, drive components, encoders and resolves, Integrated motion control, limit switches, linear actuators, linear and rotary motion components, linear position sensing, motors (both AC and DC motors), orientation position sensing, pneumatics and pneumatic components, positioning stages, slides and guides, power transmission (mechanical), seals, slip rings, solenoids, springs.

Motors are the devices that provide the actual speed and torque in a drive system. This family includes AC motor types (single and multiphase motors, universal, servo motors, induction, synchronous, and gear motor) and DC motors (brush less, servo motor, and gear motor) as well as linear, stepper and air motors, and motor contactors and starters.

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field, when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

Let's start by looking at a simple 2-pole DC electric motor (here red represents a magnet or winding with a "North" polarization, while green represents a magnet or winding with a "South" polarization).

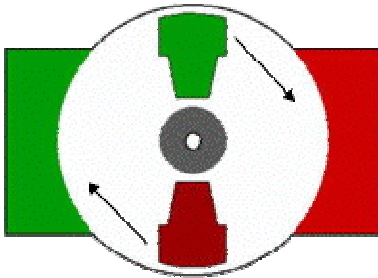


Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that Beamers will see), the external magnetic field is produced by high-strength permanent magnets¹. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a

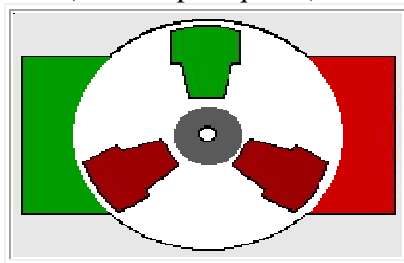
common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, and driving it to continue rotating.

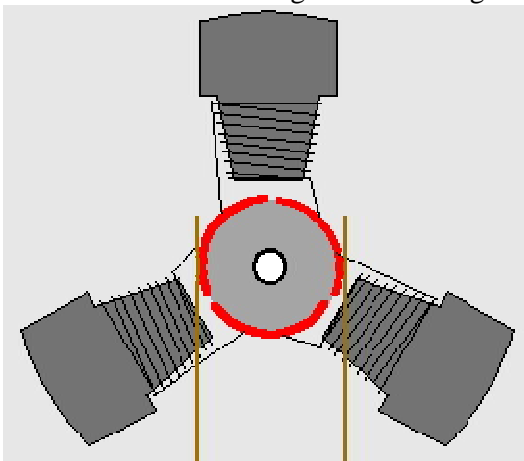
In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).



So since most small DC motors are of a three-pole design, let's tinker with the workings of one via an interactive animation (JavaScript required).



You'll notice a few things from this -- namely, one pole is fully energized at a time (but two others are "partially" energized). As each brush transitions from one commutator contact to the next, one coil's field will rapidly collapse, as the next coil's field will rapidly charge up (this occurs within a few microsecond). We'll see more about the effects of this later, but in the meantime you can see that this is a direct result of the coil windings' series wiring.



There's probably no better way to see how an average dc motor is put together, than by just opening one up. Unfortunately this is tedious work, as well as requiring the destruction of a perfectly good motor.

FIRMWARE IMPLEMENTATION OF THE PROJECT DESIGN

This chapter briefly explains about the firmware implementation of the project. The required software tools are discussed in the following section.

FIRMWARE IMPLEMENTATION

Raspbian is a competent and versatile operating system that gives your Raspberry Pi all the comforts of a PC. a command line, a browser, and tons of other programs. You can use a Raspberry Pi running Raspbian as a cheap and effective home computer, or you can use it as a springboard and turn your Raspberry Pi into any of countless other functional devices, from wireless access points to retro gaming machines. Here's how to install Raspbian on the Raspberry Pi.

How to install Raspbian on the Raspberry Pi

Installing Raspbian on the Raspberry Pi is pretty straightforward. We'll be downloading Raspbian and writing the disc image to a microSD card, then booting the Raspberry Pi to that microSD card. For this project, you'll need a microSD card (go with at least 8 GB), a computer with a slot for it, and, of course, a Raspberry Pi and basic peripherals (a mouse, keyboard, screen, and power source). This isn't the only method for installing Raspbian (more on that in a moment), but it's a useful technique to learn because it can also be used to install so many other operating systems on the Raspberry Pi. Once you know how to write a disc image to a

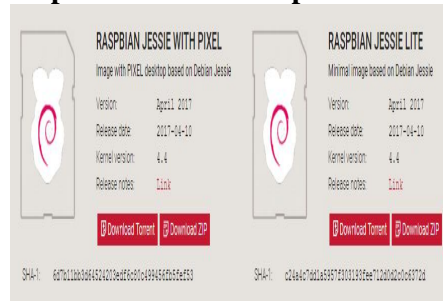
microSD card, you open up a lot of options for fun Raspberry Pi projects.

A word about NOOBS

It's worth noting that the method described here isn't your only option for installing Raspbian. You can also opt to use NOOBS, an operating system installation manager that makes it easy to install Raspbian, as well as a few other operating systems.

If you really want to make things easy, you can even buy SD cards that come pre-loaded with NOOBS.

Step 1. Download Raspbian



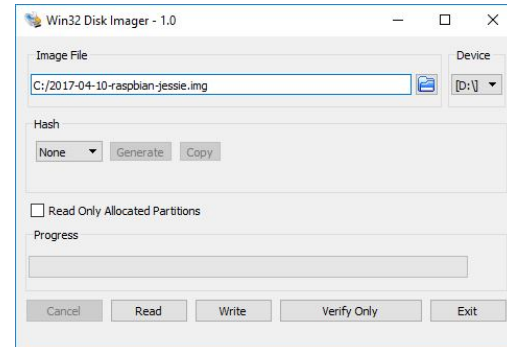
First things first. hop onto your computer (Mac and PC are both fine) and download the Raspbian disc image. **You can find the latest version of Raspbian on the Raspberry Pi Foundation's website here.**

Step 2. Unzip the file

The Raspbian disc image is compressed, so you'll need to unzip it. The file uses the ZIP64 format, so depending on how current your built-in utilities are, you need to use certain programs to unzip it. If you have any trouble, try these programs recommended by the Raspberry Pi Foundation.

- Windows users, you'll want **7-Zip**.
- Mac users. **The Unarchiver** is your best bet.
- Linux users will use the appropriately named **Unzip**.

Step 3. Write the disc image to your microSD card



Next, pop your microSD card into your computer and write the disc image to it. You'll need a specific program to do this. Windows users, your answer is **Win32 Disk Imager**. Linux people, **Etcher**, which also works on Mac and Windows is what the Raspberry Pi Foundation recommends.

The process of actually writing the image will be slightly different across these programs, but it's pretty self-explanatory no matter what you're using. Each of these programs will have you select the destination (make sure you've picked your microSD card!) and the disc image (the unzipped Raspbian file). Choose, double-check, and then hit the button to write.

Step 4. Put the microSD card in your Pi and boot up

Once the disc image has been written to the microSD card, you're ready to go. Put the card into your Raspberry Pi, plug in the peripherals and power source. The current edition to Raspbian will boot directly to the desktop. Your default credentials are username **pi** and password **raspberry**.

Working procedure.

The purpose of this project is to demonstrate the implementation of a Voice Command System as a Voice activated Smart home design and implementation that can perform numerous tasks or services for an individual. These tasks or services are based

on user input, location awareness and the ability to access information from a variety of online sources.

The working of the design starts when the user wants to control the loads using the voice command. Raspberry pi is used to implement this task.

The main reasons why we have chosen Raspberry pi as specific element are the high processing capacity, relatively low price, and its ability to adapt in different programming modes. The device uses Linux as an operating system, which has access to a large number of libraries and applications compatible with it. Raspberry Pi has an Ethernet port allowing us a network connection, as long as we are in the same subnet with the device we want to access and manage, 4 USB ports used to connect devices like a keyboard, mouse, camera, and other devices that connect through a USB port, and an HDMI port giving us access to the interface of the operating system installed, and can also be used the first time while installing the devices.

It has 40 pins that allow us to receive and send signals. They are divided in half into two groups. the 3V, and the 5V group. Therefore, one side of the microcontroller gives a voltage of 3V, and the other 5V. Besides the 40 voltage pins, it has pins that are used to receive signals, which in our case was used to connect the button, that will send the signal for the face identification. Raspberry Pi does not have an operating system previously installed, but that can be downloaded from the Raspberry website, and transferred to an SD card.

When the user gives the inputs in the form of voice, the system will receive this input and directly connect to the Amazon Alexa to control the loads. The controlling of the

loads according to the user input is written in the python script. The system compares the user input and the inputs written in the python script and if both match, the particular load will be controlled. Not only controlling the loads, the system will provide any information the user wishes to know.

CONCLUSION AND FUTURE SCOPE

CONCLUSION

The smart home space has a lot of interesting challenges to be solved. One very important problem that I tried to address in this work is that of non-smart. I made use of one of the hottest smart devices available today in the market, Amazon Echo- Alexa and coupled it with the Raspberry Pi. The module I used for performing home automation tasks worked ill. Further, on testing the application on Alexa in real-time, I obtained promising results.

FUTURE SCOPE

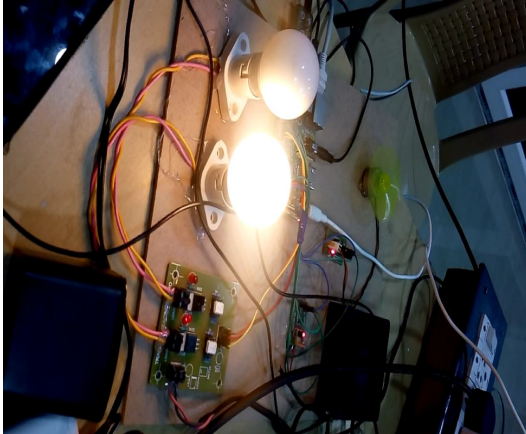
I believe that this a step towards a cost-effective smart homes. I aim to build similar applications in the future for Google Home and other personal assistants that can revolve around using simple voice commands to provide a cost effective solution for non-smart homes.

EXPERIMENTAL RESULTS

This is the setup of Raspberry Pi for recording voice, the user will give the commands to Alexa through the microphone which is connected to the Raspberry Pi in order to access the services provided by Amazon Alexa service.

Say "Alexa" with a phrase which is assign by the user to turn on and off the devices. It will react permitting to the directives concluded Echo IP Address. For example, if you want to turn on and off the light by using a command , so you have to

say “Alexa trigger turn on the light” or “Alexa trigger turn off the light” and so on.



My application relies on Raspberry Pi as it satisfies the hardware requirements and also does all the computation. The Raspberry Pi has a Wi-Fi and it uses the relay to switch on and switch off the appliances.

Reference.-

1. Markets and markets, “Smart home market by product (lighting control (timer, daylight sensor, occupancy sensor), security & access control, HVAC, entertainment, home healthcare and smart kitchen), software & service (behavioral, proactive), and geography – Global Forecast to 2022, May 2016.
2. Amrutha S, Aravind S, Ansu Mathew, Swathy Sugathan, Rajasree R, Priyalakshmi S, “Voice controlled smart home,” ISSN 2250-2459, ISO 9001:2008 Certified Journal, vol. 5, Issue 1, January 2015.
3. Sonali Sen, Shamik Chakrabarty, Raghav Toshniwal, Ankita Bhaumik d, “Design of an intelligence voice controlled home automation system,” International Journal of Computer Applications (0975 – 8887), vol. 121 – No.15, July 2015.
4. Gerhart, J. (1999). Home automation and wiring. McGraw-hill (ISBN 0070246742).
5. Mann, W. C. (Ed.). (2005). Smart technology for aging, disability, and

independence. The state of the science. John Wiley & Son

6. Yerrapragada, C., & Fisher, P. S. (1993, June). Voice controlled smart house. In International Conference on Consumer Electronics, 1993. Digest of Technical Papers. ICCE., (pp. 154-155). IEEE.
7. John Carlsen. (22 April 2014). Home Automation Systems Review. From home-automation-systems-review.toptenreviews.com/
8. JKA-Electronics. (April 2006). HM2007 datasheet ver. 1. Retrieved 2014, March
9. Atmel Corporation. (2014) Atmel AVR 8-bit and 32-bit Microcontrollers. Retrieved 2014, May