HOME AUTOMATION FROM ANYWHERE

Ayandeep Dutta

School of Electronics, Kalinga Institute of Industrial Technology
ayandeepd1@gmail.com

Abstract-

With the advancement of society there is always a constant need to reduce manual work to save time and effort as well as to achieve a certain efficiency and accuracy so as to let us deal with more important issues. This has led to the need of automated or semi automatic systems which are required to control a wide variety of devices and appliances present in large numbers, ensure timely functioning of said appliances, as well as monitor the same for their health and maintenance. Hence this project aims to implement solutions for above problems in a smaller scale. The said system has the ability to operate ordinary appliances, present in a small room, running on electrical power on the requirement of the user. Any user can operate this system based on a web portal based remote control as well as using voice commands. This project has future prospects in implementation in Industries for machine automation and monitoring as well as for domestic home or office automation.

Index Terms- AUTOMATION, HOME, HTTP, IOT, REMOTE, RECOGNIZE, VOICE.

I. INTRODUCTION

This project aims to operate multiple electrical appliances at the will of the user by taking commands from a remote web portal in form of virtual button clicks as well as voice commands of the user present locally.

This is project falls under the category development of IOT or Internet Of Things.

The need of this kind of system is in a variety of places like:

i. In industries for real time monitoring and control of various variables from remote locations as a trust worthy source for checking conditions for people not present locally.[1]

ii. At home for automated lighting, heating and cooling systems to work in coordination and synchronization or even at user will.[1]

iii. Provide a viable solution to old or differently abled people to operate all appliances at their place of stay with minimal effort.[1]

iv. Provide real time security systems with automated calling system to warn and ward homes as well as neighborhoods.[1]

v. Fitness trackers to regular people or athletes to track and update their physical training advancements as well as check on physical conditions such as cholesterol, blood glucose[2][3], pulse rate and blood oxygen level [4]which are made possible by use of modern sensors.

This project has made use of a large number of technologies. The prime ones are as follows:

- Esp8266 generic board.
- Esp8266 Arduino core.
- DC to AC Relay .
- NPN BJT transistor(bc548).
- Signal preamplifier(lm358)
- Wireless Fidelity.
- The Internet network.
- Http protocol.
- Hosting Services.
- Python.
- PHP.
- Google recognize API.

This project will help the society as a whole and as individuals too. As we eventually advanced we have seen as individuals that our work loads are increasing day by day as our own advancement is in our hands, for which other equally important tasks and duties such as socializing and meeting up new people, doing our hobbies and taking rest are getting overlooked. Thus this type of systems come in to play in which we do the jobs which are exclusively meant for us to do and cant be replaced by anyone and in return we get more time out of our life for our more important tasks.

II. Study of similar projects or technology\ literature review

While researching on this project a variety of such systems were commercial available with varied functionality. From various discussions on online threads and user reviews it was found that almost all of these major systems had various amounts of functionality and were considerably accurate and efficient. The technologies behind all these systems was looked into as closely as possible. Most of these use an external service to recognize voice and have some kind of remote control in form of Android or IOS app.

This project is heavily inspired by the above concepts which as later found out to be necessary for the successful functioning of the system as a whole.

However the commercial products use different appliances all together to pair up with main IOT control unit. Whereas this project uses already available appliances. This is a major advantage as less hardware overhead is required which should show decrease in cost as number of appliances increase.

III. BASIC CONCEPTS/ TECHNOLOGY USED

The concepts and technologies used:

ESP8266 generic board

The ESP8266 is a Wi-Fi capable microchip which is capable of behaving as a micro controller. This chip has a full TCP/IP network stack and software due to which it is fully capable of joining a Wi-Fi network and connect to the internet. This development board has an on board micro USB port and a Converter IC which is capable of converting the USB transmission protocol to serial protocol necessary for programming the main WiFi chip. The board also has a on board led for various uses.

This board can be programmed by using Arduino software(IDE). With help of programming we can make the ESP8266 work at our will and do different tasks including constantly polling certain predefined pin or outlet for voltage levels or changing the voltage level of certain pins depending on voltage level of other pins or based on predefined algorithms not involving external inputs of any kind. The chip does this by running in a loop and doing the predefined tasks in order in each iteration before starting a new iteration to do the same tasks in order. This process goes on indefinitely. This chip supports networking and so supports hosting of a server on itself as well as making different kind of requests to other servers.

DC to AC relay module

A relay is a device used to switch a larger load using a weaker signal. The relay module used is specifically manufactured to use a digital signal to switch loads running on AC.

Microphone

A microphone is a device used to take energy in form of sound and produce an electrical analog output in form of electrical signal.It needs to be connected to a voltage source using various passive components in a specific configuration to receive output from it.

Wireless Fidelity

It is a technology which enables wireless and handheld devices connect to any type of network. It works in a client server method in which a node becomes a server(called access point) and to this node other devices or nodes(called stations) can connect using wireless signals as client. The server always decides whether to allow a client access a network or not.

HTTP protocol

The HTTP protocol is a way of communicating on a network over the internet supporting Internet Protocol(IP) . This protocol is one of the ways we can send messages and data between different devices or nodes across a network.

Hosting Services

The hosting services are different organizations and companies running large servers to host server software for the use of anyone who requires their software to be online for a long period of time or indefinitely as per the case for free or while charging money at different rates.

Two such hosting services used by this are **www.000webhost.com** and **aws.amazon.com.**

Google Assistant API

This is a service provided by **www.google.com.** This service accepts an audio file in wave form and recognizes if the audio has any speech in it and returns text interpretation of the audio if speech was found.

IV. PROPOSED MODEL / ARCHITECTURE / METHODOLOGY/ MODEL TOOL

The proposed model is as follows:

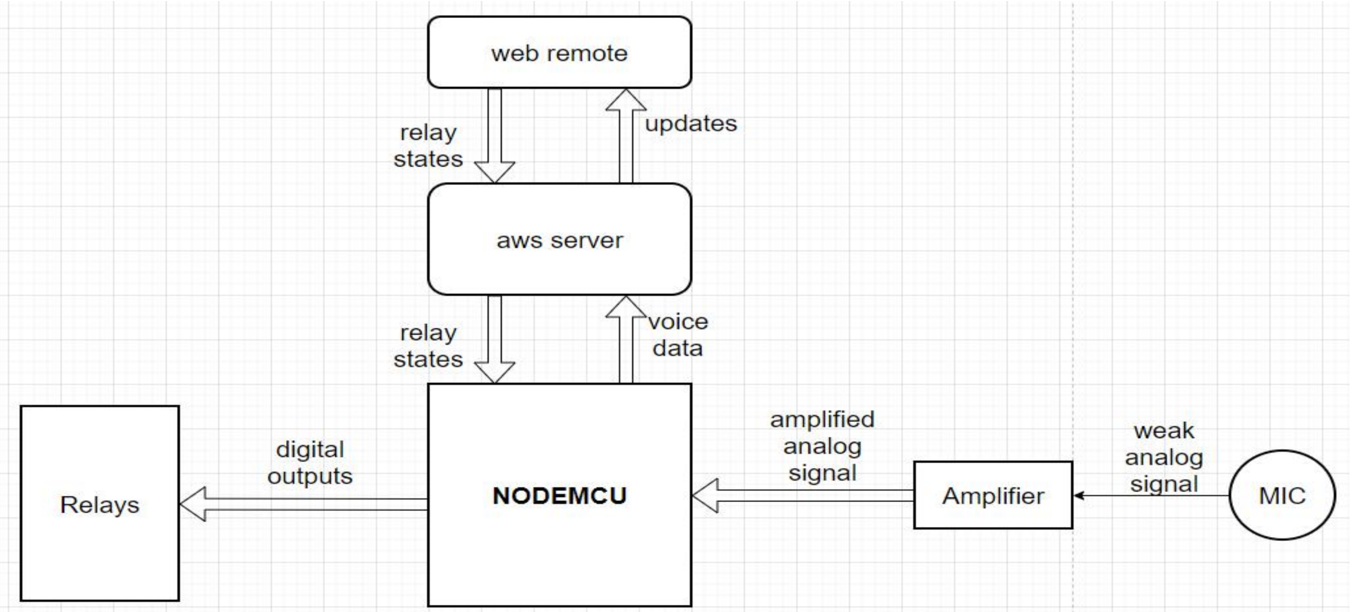
The esp8266 is capable of sending data over network but is not powerful enough to recognize audio on its own(using suitable machine learning models and algorithms). Thus to solve this incapability we use an external server hosted on Amazon Web Services which can accept our audio data and give us back the ESP8266 I/O pin states.

Thus this method can be applied as:

1. There should be a remote user portal on 000webhost.com for interacting with our system. This web portal will house a website for user interaction, a script which will take in user input from aforementioned website and log it into several files for each of the appliances, and this same script will also give us back the status of each appliance whenever requested.

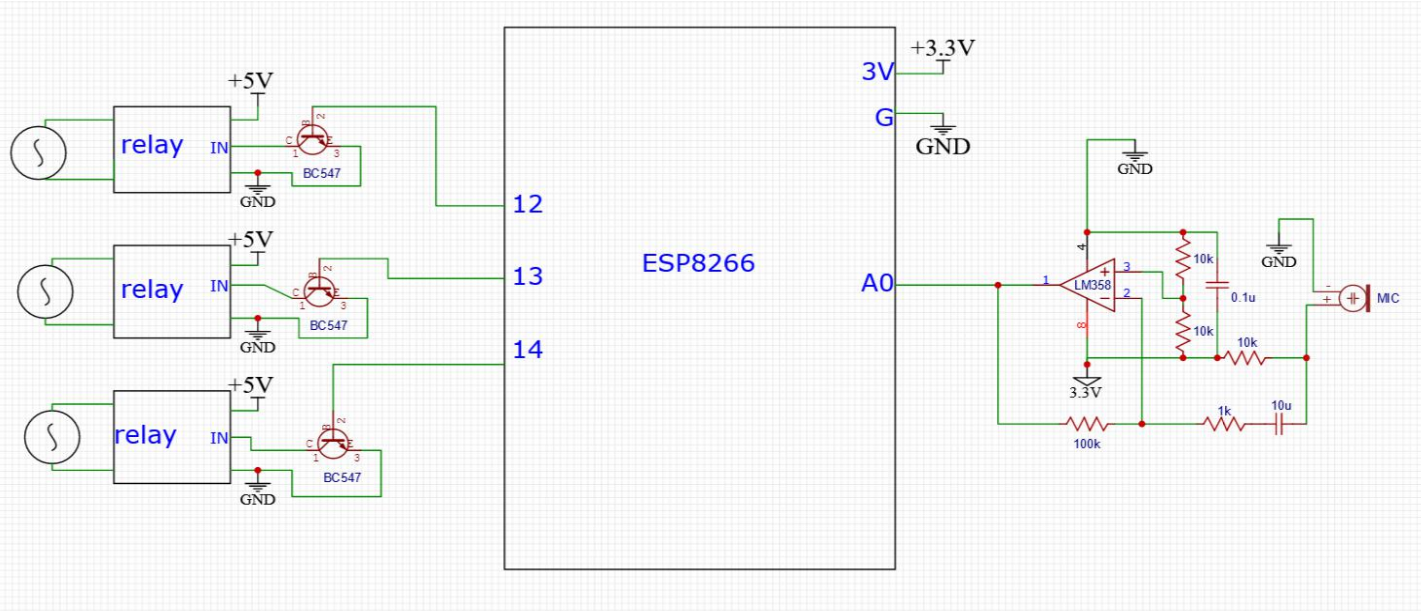
2. A python server on aws.amazon.com which will accept the audio data from ESP8266 and process it and send it to google assistant API for recognition. If successfully recognized and the command is valid then a valid response is sent back else an error code should be sent back for any kind of error.

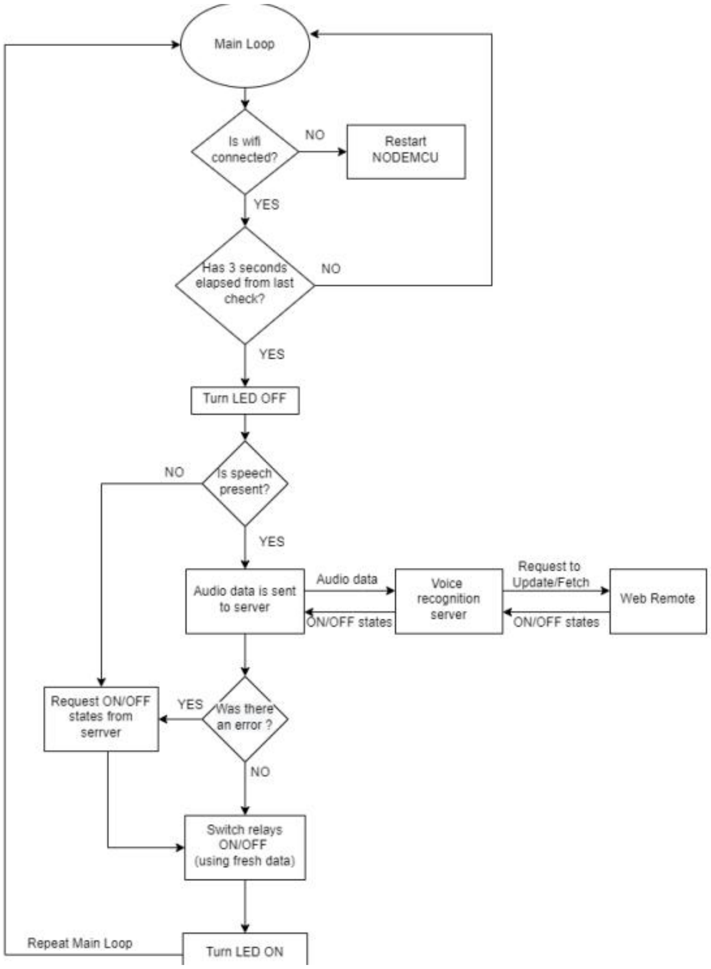
3. We have our ESP8266 board which acts as our local IOT device. It will function to change its output states according to the received response from either of the servers, check the microphone connected to its ADC outlet at (specific time intervals so as to record audio data), detect if any audio is present and empty data is not sent, will request responses from the web server at regular intervals. The ESP8266 covers all these tasks one at a time in a repetitive manner indefinitely.

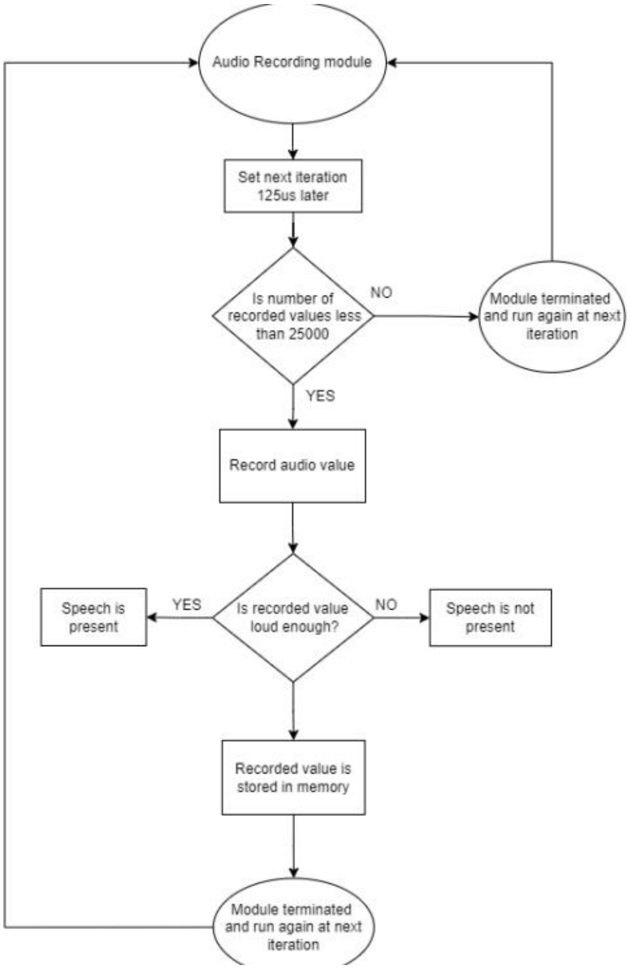


V. IMPLEMENTATION AND RESULTS

The diagram for ESP8266 wiring:







Audio Recording module
Set next iteration 125us later
Is number of recorded values less than 25000
NO
Module terminated and run again at next iteration
YES
Record audio value
Is recorded value loud enough?
YES
Speech is present
NO
Speech is not present
Recorded value is stored in memory
Module terminated and run again at next iteration

The above proposed model is implemented in a more detailed manner as follows:

1. First of all a website on 000webhost.com is kept running always. This website has three buttons and a welcome message. The website includes a password protection system also for validation. On clicking any one of the buttons a post request is

sent to the server side script which writes the opposite value of the one stored in the respective file. For example if file for light A stored 1 then the script will rewrite the file with 0 thus toggling the state of the appliance.

2. The server side script in above server has the job to update values that are coming in form of post requests. It does this job by first checking if the password in the request is correct then proceeds to toggle the stored value of the appliance whose value is to be changed. It also serves the job of sending the IP of the recognition server supposed to be hosted on aws platform as well as the states of the connected appliances.

3. A voice recognition server hosted in aws.amazon.com. This server's main role is to accept audio data values from the Esp8266, convert to audio file object and send this object to google recognition service using the Google API. Once the audio has been processed, if no speech was recognized then an error code is sent along with states of appliances other wise the data is processed and checked for command validity. If the commands were valid then latest device states are retrieved from the web portal server, altered and sent back as response otherwise the these are sent as is, also a post request is sent to the web portal server so that the necessary values are reflected over there.

4. Finally we have our locally running ESP8266.

a) The esp8266 working involves two nearly independent processes. The main process targets the networking, audio transfer and device control while the other's function is to record audio samples after fixed duration.

b) The main process checks if WiFi is connected or not, turns the led off when communication to any server is started and turns back on when completed, send the audio data if loud enough noise is received, receive appliance states and apply received states on appropriate appliances.

c) The second process is dependent on the inbuilt timer of the esp8266 which can be called to run a specific task after a fixed interval. This process is set to run every 125micro seconds, each time checking total samples recorded, takes an audio sample from the microphone puts a check if the sample is loud enough or not, stores the sample and setting its own timer for next run.

Notes on usage

1. Any interactions to the web portal are saved immediately and reflect on devices after the immediate next iteration is done.

2. Voice interaction is to be timed between two successive requests. To understand when the Esp is requesting data the on board led is set to light up when requesting starts and light off when requesting ends. Thus any commands to be said should be completed in between two led blinks which is a approximately 3 seconds. To aid this, commands have been kept short and any interaction will toggle the device state.

VI. CONCLUSION

The project is built successfully and functions as described throughout .

The conclusions drawn from testing this project are:

1. The ESP8266 board can connect to the WiFi network but takes some time to do so.

2. When voice is not used the state of appliances(on or off states) are retrieved from the server with little to no delay and are successfully reflected in the physical appliances as they switch on or off without any error.

3. The voice recognition is a slow process and commands are to be said with care at specific intervals thus reducing user friendliness.

4. Audio data is difficult to initiate sending and results in a network error occasionally however once sent it registers without error in server and the resulting response can also be applied to the pins responsible and thus reflect on the appliances as they switch on or off.

5. Audio recording process occasionally causes overload during slow network conditions and eventually leads to crashes.

Some adjustments made to the project to counter above errors:

1. The WIFI server is brought closer to the ESP chip so that network error can be minimized.

2. It was discovered that keeping a network idle cause the ESP8266 to not recognize it later hence giving 404 errors, hence it was later adjusted to make get requests and post requests both to the aws server rather than only post requests. The aws server was responsible to fetch values from the 000webhost server and forward it to our ESP8266 chip.

Some future additions can be made:

1. More appliances can be added in a more flexible way.

2. Status of the appliances can be showed in the web portal such that appliances can be monitored from remote location.

3. Appliances requiring varied controls other than on/off can be added using external hardware expansions(I2C devices, demultiplexers, so on).

References

circuit Basics for microphone reference:

https://www.circuitbasics.com/how-to-use-microphones-on-the-arduino/

ESP8266 http class reference:

https://links2004.github.io/Arduino/dd/d8d/class_h_t_t_p_client.html

circuits4you.com for timer usage example:

https://circuits4you.com/2018/01/02/esp8266-timer-ticker-example/

flaviocopes.com for python http server example:

https://flaviocopes.com/python-http-server/

W3schools.com for various php, javascript and html references:

https://www.w3schools.com/php/default.asp

Google speech recognition api reference for python:

https://github.com/Uberi/speech_recognition/blob/master/reference/library-reference.rst

Appendix A: Project Code

(Github link: https://github.com/ayandeepd1/iot_esp_automation.git)