

Echo-Mote

Design Document

Dec1720

Gary Tuttle

Sarah Huber - Team Leader

Alex Christenson - Communication Leader

Ross Reicks - Idea Holder, Web Master

dec1720@iastate.edu

<http://dec1720.sd.ece.iastate.edu/>

Revised: 3/09/17 Version 1.0

Table of Contents

Introduction	2
Project statement	2
Purpose	2
Goals	3
Deliverables	3
Design	3
System Specifications	3
Non-functional	3
Functional	4
Standards	4
Proposed Design Method	5
Design Analysis	5
Testing/Development	5
INTERFACE specifications	5
Hardware/software	6
Process	6
Results	6
Conclusions	6
References	7

Introduction

Project statement

Echo-mote is a hardware module that the Amazon Echo can communicate with to perform basic operations on your television. These operations include turning the volume up and down, changing the channel up, down or to a specific channel as well as turning off and on your television. A product of this type has not yet been created and we hope to be first to market.

Purpose

The Echo-Mote is a project primarily aiming to allow visually impaired people to be able to operate their television on their own. There are handicapped people all over the world that need assistance with daily tasks--this serves as a simple solution

Goals

We aim to create a device that anyone can use. This includes those with physical handicaps. We hope to deliver a product that is not only incredibly simple, but also physically appealing. We hope to have compatibility with all televisions. We will allow control using everyday language. It must be easier to use than a remote.

Deliverables

We will deliver a hardware module, a power adapter, an application to configure the remote and a user manual. We will create a design for the packaging the remote will be marketed in. We will deliver a kickstarter style website in order to generate potential crowdfunding.

Design

To meet our goal of a minimum viable product by the end of the semester, we have had to narrow down our focus and concentrate our time on the crucially important steps. For our viable product we need an amazon echo to talk to our device and our device needs to turn on and off a television. Our MVP does not need to be pretty or bug free but it must work. So far, we have had to do a bunch of research on remote protocols and the amazon alexa SDK. We have built a small device that is capable of sending commands to a tv. We can program it to turn on, off power, up and down channels and volume. We have built a raspberry pi web server the can communicate with this device. Our next plan is to be able to make requests to this web server with endpoints such as “turn-up-volume” etc. and these endpoints will send the correct command to our device.

System Specifications

Non-functional

The project will also consist of many preliminary tasks that will not be seen by the user.

Connect To Wifi - The device needs to be able to be configured to the user's current in house wifi network so that alexa can communicate with it. The only interface that the amazon echo currently supports is wifi.

Creating a compact device - We want our device to be able to sit on someone's coffee table. We want it to be circular with about the same diameter of the current Amazon Echo. This could be a challenge because our current implementation utilizes a raspberry pi and an arduino uno. We will have to build our own hardware to get it to fit in this small of a space.

Infrared power - In the EE 230 lab we build a remote control and receiver but we had troubles with distance and line of site. We don't want this to be an issue. We have done some research on how to increase the power of the signal to get further and indirect distances.

Power - This device will need power. Battery powered devices struggle with consistent power and users would have to change batteries. For this reason we are leaning towards a wall powered device. Efficiency is also an issue. If we make a device that is power hungry users might not necessarily want to buy our device.

Heat - We are shoving a lot of hardware into a small space. Heat could be an issue. We will have to consider this if this problem arises. This should not be an issue if we make an efficient board that is not as power hungry.

Functional

In order to implement the following commands, we will need to utilize amazon's open source software and apply our commands that we would like to the code. The software will then have to communicate with the hardware to tell the TV what to do.

The amazon echo will initially be able to do all basic remote functions via a voice command through the amazon echo.

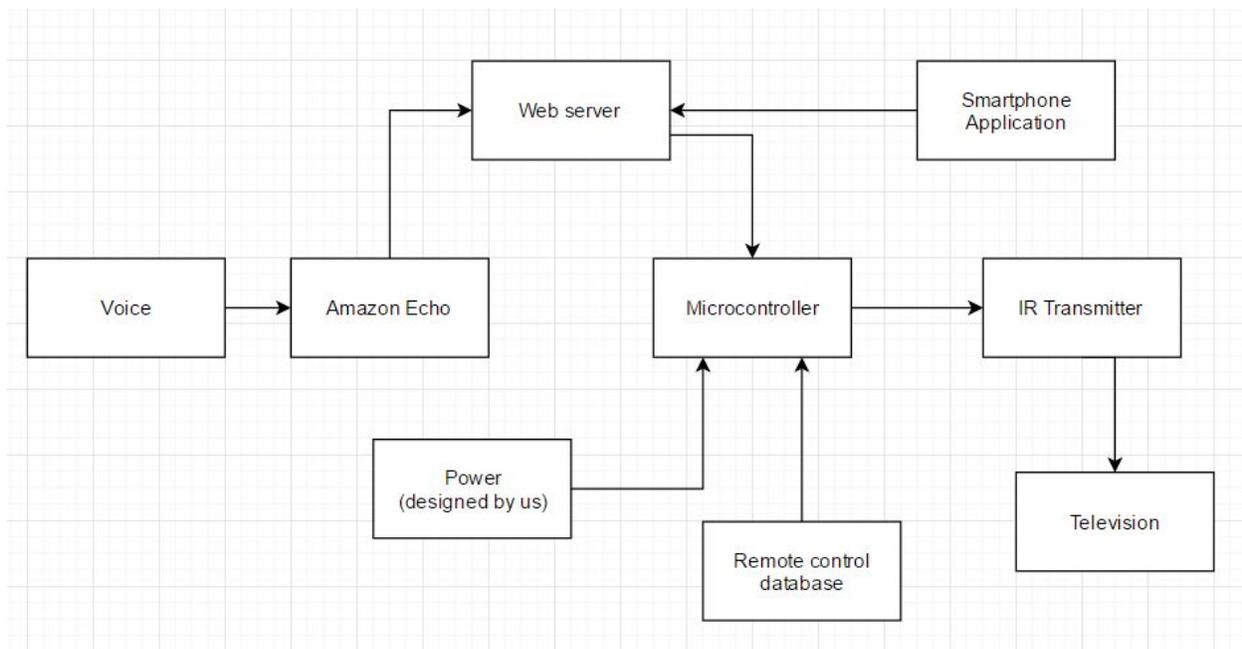
Functions:

- Power On/Off
- Input
- Channel Up/Down
- Change to channel X
- Volume Up/Down
- Volume Up/Down Increment
- Mute
- Other options will be added with lower priority

Standards

We have found a device that is similar to our product called the anyMote (see references for link). Their device cost \$140 and the voice commands are not as natural. We aim to make our product better and for a more affordable price.

Proposed Design Method



Design Analysis

The proposed design method above will allow for an efficient and effective way to implement our project. In order to implement our proposed design, we will work backwards. Our first stage is to turn on the TV using an IR transmitter. We will use a temporary microcontroller and then create a final concise microcontroller for our final project. Once we are able to interact with the TV, we will work on interacting with the web server and then adding the voice commands.

Testing/Development

INTERFACE specifications

For this project, the interface will be via voice commands. There will be a list of common commands that we will implement. The following initial codes are:

1. "Tell TV to turn On/Off"
2. "Tell TV to change channel"
3. "Tell TV to turn volume up/down"
4. "Tell TV to turn volume up/down X amount"
5. "Tell TV to tune through channels"
 - a. "Tell TV to stop tuning"
6. "Tell TV to change input"

Future inputs could be:

1. "Tell TV to turn on Apple TV"
2. "Tell TV to open Netflix/Amazon/Hulu/etc"
3. Be able to change TV settings

Hardware/software

Testing certain aspects of the Echo-Mote components is easily done with any commercial television and the Amazon Echo. This makes the testing relatively easy to do since there are few instruments needed in order to produce a working prototype.

TV:

The television we use to test is a standard Samsung commercial TV. It only tests the IR protocol specific to the Samsung brand, but since all other IR transmission is very similar, testing with other models and brands is unnecessary in the prototype phase.

DMM:

Since it is important to be able to transmit over a relatively long range, we need to have the largest possible current output to the LED transmitter. The DMM is useful to test when attempting to achieve the output on the datasheet.

Process

Results

Thus far in our project, we have been able to implement some of the basic commands. Using an arduino we are able to send the command to turn on/off, change volume, etc. Our next steps are to implement this by communicating via wifi and then send commands via voice.

Conclusions

In conclusion, this device will be able to control a television fully from the Amazon Echo. It will act as other products that already interact with the Echo via wifi. This product will be used by all people but will open the door for people with seeing disabilities to be able to control their television. The final product will look like something that would be purchased at a store, the product will not only be functional, but physically appealing. Finally this project will teach us how to create a piece of hardware that interacts with wifi, how to utilize open source software, and how to plan, design, and create a product.

References

AnyMOTE- <https://www.anymote.io/tutorials/tutorial-how-to-control-your-tv-with-amazon-echo>

Codes- <https://sourceforge.net/p/lirc-remotes/code/ci/master/tree/remotes/>

Protocols- http://www.techdesign.be/projects/011/011_waves.htm

IR protocol explained-

<https://rusticengineering.com/2011/02/09/infrared-room-control-with-samsung-ir-protocol/>