

**Using Speech Recognition to Control the Home Environment:**

David Harrington, Ryan Henderson, Peter Mason, Dhruva Seelin

Penn State University

Learning Factory Project

Spring 2017

All opinions are those of the authors

## Amazon Echo Setup

- 1) Plug in the Echo into an outlet (The echo should be blue on top)
- 2) Install the Amazon Echo App on a mobile device
- 3) Sign in with your Amazon Account
  - a) Echo should now say “Hello, your Amazon Echo is now ready for Setup”
- 4) Connect to the Amazon wifi on your mobile device
- 5) In the Amazon Echo App, connect to your home wifi
- 6) The Amazon Echo is now setup

## Alexa Skill Setup

### Getting Started

- 1) Go to [developer.amazon.com](https://developer.amazon.com)
- 2) Sign in on the top right
- 3) Click the Alexa option at the top of the screen
- 4) Click “Get Started” under Alexa Skills Kit
- 5) Click the “Add a New Skill” button on the top right

### Initial Skill Set Up

- 1) The Name the name of the skill displayed to the customer
  - a) Ex. Youtube Data API
- 2) The Invocation name is the phrase the user needs to utter to use the skill
  - a) Ex. youtube data (you could say “Alexa, ask youtube data for \_\_\_\_”)

### Interaction Model

- This model consists of 3 parts: the Intent Schema, Custom Slot Types, and Sample Utterances
- 1) Intent Schema
    - a) This is where you actually list all the different actions the skill will be able to perform.

```

1  {
2  "intents": [
3  {
4  "intent": "GetSubscriberCount"
5  },
6  {
7  "intent": "GetVideoViewCount"
8  },
9  {
10 "intent": "GetVideoViewCountSinceDate",
11 "slots": [
12 {
13 "name": "SinceDate",
14 "type": "AMAZON.DATE"
15 }
16 ]
17 }
18 ]
19 }

```

- b)
- 2) Custom Slot Types
    - a) Slots are certain data structures you may want to use. While amazon already offers many pre-made slots (such as date, time, etc.), you have the option to create your own slot as well.
    - b) As seen in the example above, if the user wanted to ask “Get video view count this month”, the value of “this month” would be translated into the first of the month and would be assigned to the variable SincDate.

### 3) Sample Utterances

- a) Sample Utterances are what you want the user to say in order to activate the intent. The syntax of sample utterances are shown below. First you type the intent followed by the phrase(s) the user needs to call to activate that specific intent.

```

1  GetSubscriberCount current subscriber count
2  GetSubscriberCount subscriber count
3
4  GetVideoViewCountSinceDate current video view count {SinceDate}
5  GetVideoViewCountSinceDate video view count {SinceDate}
6
7  GetVideoViewCount current video view count
8  GetVideoViewCount video view count

```

- b)
- After completing the interaction model, click next to be taken to the configuration model. This is when you need to create and link a lambda function for the skill. Leave the developer.amazon.com page open and go to aws.amazon.com/console in another tab.

### Creating and Linking a Lambda Function

- 1) While having the skill creation tab open, go to aws.amazon.com/console in another tab
- 2) Click “Sign In to the Console” on the top right.
- 3) After logging in, Click the “Services” drop down in the top left.
- 4) Select “Lambda” under the “Compute” category
- 5) Select the “Functions” Option on the left

- 6) Select the “Create a Lambda function” button on the top left
- 7) This is where you can select from several pre-made blueprints for a lambda function. You can search “Alexa” in the filter option to see what blueprints there are for Alexa skills. If you are creating you first Alexa skill, it is better to just create a skill without a blueprint.
- 8) Select the “Configure Triggers” option on the left
- 9) Click the empty dashed box in the middle of the screen to reveal a drop down menu.
- 10) Select the “Alexa Skills Kit” options and select next on the bottom right.
- 11) Enter a name (E. youtubeDataAPIAlexaSkill), description and leave the default Node.js option.
- 12) After entering your Node.js or Python code in the code box, scroll down and select index.handler, choose an existing role, and basic\_lambda\_execution for Handler, Role, and Existing Role respectively.
- 13) After ignoring the Advanced Settings section and clicking next at the bottom of the page, click Create to create the skill.
- 14) Once the skill is created, you should see a ARN number at the top right of your screen that looks something like this:  
**ARN - arn:aws:lambda:us-east-1:431180689513:function:youtubeDataAPIAlexaSkill**
- 15) Copy-Paste that ARN into the ARN space on the developer.amazon.com page. Click Next. This links the Amazon Skill to the Lambda function you created.
- 16) At this stage, the function has now been deployed to the associated amazon account. If everything was done correctly, you should be able to test the skill on Alexa by saying “Alexa” followed by the Invocation Phrase.

If something didn't work correctly or if you want a more detailed overview of a skill setup, I would suggest watching this video: <https://www.youtube.com/watch?v=zt9WdE5kR6g&>

## Creating the Alexa Gmail Skill

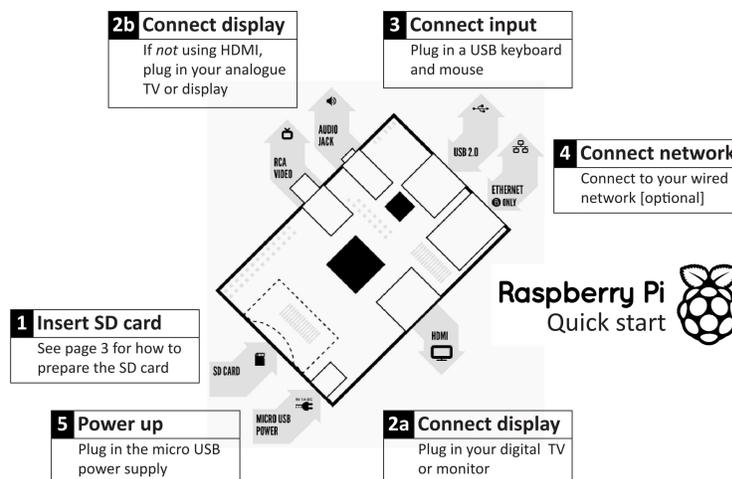
The process of creating the Alexa Gmail Skill was enlightening in many aspects. Before deciding on creating an Alexa Gmail Skill, the group had actually set a goal of creating a Facebook Skill. This Skill was supposed to read Anthony any notifications that was sent to his facebook account, also allowing him to read and post to his profile without needing to go to his computer. As Anthony is an avid facebook user, this skill was supposed to help streamline his social media lifestyle. While we were able to implement posting and reading to/from Anthony's timeline, we ran into several obstacles. Only relatively recently, Facebook removed API access to a user's feed and notification hub for security reasons. This was a major setback, as this eliminated the ability to have any interaction with the user's facebook feed and the possibility of announcing the users notifications. We also discovered that facebook made posts that were created via third party apps “hidden”. This means that only Anthony himself would be able to see the post he made, essentially negating the purpose of the Skill. After examining the mistakes

and challenges we faced with the Facebook Skill, we decided to pivot our efforts toward making a Gmail Skill.

Compared to Facebook’s API, Gmail’s is much more liberal in regards to how much information is accessible. We were able to connect our Amazon account to the users Gmail via an access token. With this Gmail Skill, Alexa was able to read unread emails in the user’s inbox, delete an email, or reread and email if the user needs. While the Gmail Skill was more successful than the Facebook Skill, there were still some challenges we encountered. One was that Alexa wasn’t being friendly with the AAC device. A problem Anthony told us about was that Alexa only waited about 7 seconds after asking for an input before shutting down the skill. An example of this is after reading an email, Alexa will ask “What would you like to do?” If Anthony isn’t able to have his AAC device say the answer within seven seconds, Alexa will shut down. Unfortunately, there is nothing in the Alexa API that allows us to manipulate the time delay. Another problem we faced had to do with Alexa’s inability to properly read certain emails. If the email was purely simple plaintext, Alexa would have no problem reading it. If there was any special formatting or characters, however, Alexa would respond saying the email contains an error and is thus unreadable.

## Raspberry Pi Setup

We began our work creating a Pi-Alexa proof of concept by setting up the Raspberry Pi, which we document here. After obtaining the [Raspberry Pi 3 Model B](http://www.raspberrypi.org/qsg) (the starter kit is more convenient, but not required), visit <http://www.raspberrypi.org/qsg> for detailed instructions for first-time setup. Below are a list of all required items if not using the starter kit, as well as a blueprint for connecting power, ethernet, an SD card, and USB devices. A 220-330 ohm resistor



1

and an LED are also required for later on in the process.

Additionally ensure that your microSD card has the NOOBS operating system installed

## To set up your Raspberry Pi you will need:

	Item	Minimum recommended specification & notes
1	SD card	<ul style="list-style-type: none"> <li>Minimum size 4Gb; class 4 (the <i>class</i> indicates how fast the card is).</li> <li>We recommend using branded SD cards as they are more reliable.</li> </ul>
2a	HDMI to HDMI / DVI lead	<ul style="list-style-type: none"> <li>HDMI to HDMI lead (for HD TVs and monitors with HDMI input).</li> <li><b>OR</b></li> <li>HDMI to DVI lead (for monitors with DVI input).</li> <li>Leads and adapters are available for few pounds -- there is no need to buy expensive ones!</li> </ul>
2b	RCA video lead	<ul style="list-style-type: none"> <li>A standard RCA composite video lead to connect to your analogue display if you are not using the HDMI output.</li> </ul>
3	Keyboard and mouse	<ul style="list-style-type: none"> <li>Any standard USB keyboard and mouse should work.</li> <li>Keyboards or mice that take a lot of power from the USB ports, however, may need a powered USB hub. This may include some wireless devices.</li> </ul>
4	Ethernet (network) cable [optional]	<ul style="list-style-type: none"> <li>Networking is optional, although it makes updating and getting new software for your Raspberry Pi much easier.</li> </ul>
5	Power adapter	<ul style="list-style-type: none"> <li>A good quality, micro USB power supply that can provide at least <b>700mA at 5V</b> is essential.</li> <li>Many mobile phone chargers are suitable—check the label on the plug.</li> <li>If your supply provides less than 5V then your Raspberry Pi may not work at all, or it may behave erratically. Be wary of very cheap chargers: some are not what they claim to be.</li> <li>It does not matter if your supply is rated at <i>more</i> than 700mA.</li> </ul>
6	Audio lead [optional]	<ul style="list-style-type: none"> <li>If you are using HDMI to HDMI then you will get digital audio via this.</li> <li>If you are using the analogue RCA connection or a HDMI to DVI cable, stereo audio is available from the 3.5mm jack next to the RCA connector.</li> </ul>

### Know your leads:



HDMI connector



HDMI to DVI lead



RCA composite video connector

## 2

on it. If this is not the case, download it from <https://www.raspberrypi.org/downloads/noobs/> and put the unzipped files on the microSD card. Once complete, insert the microSD card into the Pi as located above.

After successfully turning the Pi on and connecting a display via an HDMI cable and a USB mouse and keyboard, begin the operating system installation process. NOOBS allows the user to choose one of several operating systems. Your choice here does not impact the rest of the Pi's functionality, but for ease of use we recommend Raspbian Jessie. Wait until installation to completes and you are taken to a desktop background.

Two important settings must be enabled before beginning the work with Amazon Alexa. First, connect the Pi to the internet, either by connecting an ethernet cable from a router to the Pi, or by enabling WiFi from the top right corner of the screen. Second, enable SSH by going to Preferences -> Raspberry Pi Configuration -> Interfaces, then selecting "Enabled" next to "SSH." Selection "Ok" to confirm and exit the preferences window. Finally, open the terminal window, log in with the default username and password, then type "ifconfig" and press enter. Save the inet information for later. SSH enables the user to access the Pi remotely, namely from

their own computer. From this point onward, we no longer need the HDMI cable or the USB mouse and keyboard.

## Raspberry Pi and Alexa Interface

After reaching the point where you have enabled SSH on the Raspberry Pi, install an SSH client on your computer. There are numerous free options for Windows, Mac, and Linux. Start the SSH client, enter the inet information saved from earlier for Host Name/IP Address, and begin the session. Confirm that the information is valid and that you can successfully log in to the Pi, otherwise see [here](#) for help with SSH. From this point, closely follow the guide at <http://www.instructables.com/id/Control-Raspberry-Pi-GPIO-With-Amazon-Echo-and-Pyt/>, which provides a concise set of instructions for using Alexa to turn an LED on and off via the Raspberry Pi.

## Conclusion

Looking back, there were many obstacles to overcome to reach the final state of the project. While we originally set out to create an all-you-can-do Alexa Facebook Skill, we eventually pivoted to create a relatively extensive Alexa Gmail Skill. Throughout the entire process, the question we kept asking ourselves was “How would this benefit Anthony?” While this caused us to throw out many of the ideas we came up with in our original brainstorm, the question guided us toward making a more productive and useful Alexa Skill. An example of this guiding process is how we implemented the different functionalities of the skill. Because Alexa reads all text at the same speed and inflection, some words can be hard to hear for user. This is the reason we implemented the ability to ask Alexa to repeat emails after each reading. Another major concern for our specific client, was Alexa’s ability to properly comprehend Anthony’s Speech Generating Device. Luckily, after Anthony was able to purchase and test the Echo, he confirmed that the Echo was able to understand his device about 95% of the time.

One of the biggest causes for delay in the project was not having a full understanding of the abilities of the Amazon Echo and other API’s the skill would be interacting with. This is what caused our group to pursue the Facebook skill idea well into the semester before realizing it wasn’t a feasible idea. This is why we recommend thoroughly researching the hard set limits of the Echo and other API’s used. We also recommend setting hard deadlines to ensure not too much of the project is spent on purely the research stage.