## 



Final Progress Report

**Alexa Printing Project - 12/1/2019**

Maximilian Moore, Nathan Cox, Kevin Valle

COSC 439

Professor Dehlinger

Towson University

# Objective

We will use a raspberry pi to create a print server that can be controlled with Alexa commands. An alexa skill will send instructions to the raspberry pi, which is connected to a printer, enabling it to print files with audible commands from wherever you may be.

## Examples:

Realistic uses:

1. Printing common documents such as resumes and cover letters.
2. Streamline the process for less tech savvy people.
3. College students running late and need to print without pulling out their laptop.
4. Print a document for a friend or family member when you are not home.

# Work Distribution

1. **Research and find needed resources:** Kevin
2. **Acquire the following components:** Max, Nathan, Kevin
3. **Research how the following software runs:** Max, Nathan, Kevin
4. **Configure Raspberry Pi with Rasbian:** Nathan
5. **Install Common Unix Printing System (CUPS) on the Pi:** Nathan,Max
6. **Add the printer to CUPS:** Nathan,Max
7. **Install Python and Ngrok on Raspberry Pi:** Nathan
8. **Develop the Alexa App:** Nathan, Max, Kevin
9. **Set up a CGIHTTPServer on the PI:** Nathan
10. **Write a file searching program in Java:** Max
11. **Test Product:** Kevin, Max, Nathan
12. **Troubleshoot and Debug:** Max, Kevin, Nathan
13. **Final Touches:** Kevin, Max, Nathan

# Work Distribution Details / Timeline

## Research and find needed resources

* Build our understanding of how to create a server to send our request to our raspberry pi. We found a couple different approaches:
  + Using an S3 bucket through AWS
  + Using a DynamoDB
  + Creating our own mySQL server
* Set up an Alexa Development console

## Acquire the following components

* 1. Printer (completed)
  2. Alexa capable device (completed)
  3. Raspberry pi (completed)
  4. Create repository (completed)

## Research how the following software runs

* 1. **Common Unix Printing System (CUPS):** We had to change the cups configure file to get the software to work with the PI.

\*Completed.

* 1. **Internet Printing Protocol (IPP)**

\*Completed.

* 1. **Alexa skill development:** We had to create an Alexa skill using the Alexa developer tools. From the skill needed to create all the possible voice commands and retrieve a string from those voice commands that will be the document and store this single string in s3.

\*Completed.

* + - 1. We have not started. This is our next step to figure out after developing the Alexa skills.
  1. **File searcher on the pi:** Create a program that will take the document string and search the pi files for a match or to return false if the document is not present.
     1. If the file searcher returns true we send the matched document to Cups and print.
        1. We confirm if it printed then return true or false if it did or did not.

\*Completed.

## Configure Raspberry Pi with Rasbian

\*This step of the process has been completed.

## Install Common Unix Printing System (CUPS) on the Pi

\*This step has been completed.

## Add the printer to CUPS

\*This step has been completed.

## Develop the Alexa Skill

* 1. The skill will be developed in the Amazon Developer console using Python

\*This step has been completed

## Set up a web server on the raspberry pi to receive the JSON object sent by the Alexa Skill.

\*This step has been completed.

## Test product

* 1. Test 1: Give Alexa voice commands to print a document on the computer.

\*This test was completed.

* 1. Test 2: Give Alexa voice commands to print a document that is not on the computer.
  2. Test 3: Give Alexa a command to print a file type such as an mp3 file that cannot be printed.

## Troubleshoot / Debug

1. Use the debugger in the Alexa development console.
2. Troubleshoot by using the interactive environment in Alexa console.

## Final touches

1. Ensure that we have responses for all possible scenarios in the Alexa skill.
2. Remove any redundant code.

# Expected Results

We expect our product to print any needed documents through voice commands via Alexa. The voice commands will be spoken into the Alexa app, which in turn will send a signal to the raspberry pi, instructing the printer to print the documents. We expect to be able to print from our printer via Alexa from any location. Hence, we will be testing our product while being in the same room as the printer and while being in a different location away from the printer and the raspberry pi.

# Evaluation Methodology

Our product must simply print the needed documents when the command comes in from the user. If the documents are printed, we will then know our product does the required task. That is how we will evaluate our success of the product.

# Deliverables

By the end of this project we should have a functioning printer via raspberry pi and alexa. In doing so we should have a good understanding of drivers so we can send instructions from the raspberry pi to the printer. We also need to become proficient in the alexa development api. We also need to understand IPP and CUPS. In doing those last three tasks we will have a finished project and a deeper understanding of device drivers.

# Implementation Details

First, the web server on the Raspberry Pi is started, and tunneled through the firewall using ngrok. Once the Alexa still is invoked by an audible command, the user tells Alexa what they would like to print. The skill then sends a JSON object to the web server invoking a CGI script called alexa.sh. This script then executes the linux print command, and sends a success response back to the Skill.

# Challenges and Problems Met

* Setting up a webserver on the Pi with the right permissions.
* Enabling the web server to be accessible to the internet.
* Creating a second endpoint on our original Alexa skill.
* Accessing the JSON object sent to our web server.
* Passing a file variable into the bash script

# Result Analysis

As stated in our expected result section, our Raspberry Pi was able to communicate with the Alexa skill using the http web server, and has printed the required document we asked for. We have attached a short video that shows how our product works in our final presentation powerpoint, which has been compressed into the zip file. While we were able to achieve a successful run, there are still many dynamics we would have liked to implement in our project. These are outlined in the following Future Work section.

# Future Work

In terms of the project, we have completed the required task. If we do decide to return back to this, one thing we wish to improve is our file searcher and the abilities of the Alexa Skill. Although it does the job, it could be more efficient. As of right now, we have a file searcher that searches through all the files in the user’s directory. But this file searcher has not been implemented into the Alexa Skill due to unforeseen challenges. We must add more dynamics to the skill, enabling it to pass the file name to the shell script, which will then execute the file searcher. We also would add more code to handle files that are not found, or files that cannot be printed. As of now, the Alexa skill will simply respond with “There was a problem with the selected skill’s response.

# Repository URL

https://gitlab.com/mmoore54/finalproject439.git

# References

Apple. (2019, September 5). apple/cups. Retrieved from <https://github.com/apple/cups>

Teach, Learn, and Make with Raspberry Pi – Raspberry Pi. (n.d.). Retrieved from <https://www.raspberrypi.org/>

Johnson, Dave. “How to Host a Raspberry Pi Web Server on the Internet with Ngrok.” *ThisDaveJ*, 29 Jan. 2019, <https://thisdavej.com/how-to-host-a-raspberry-pi-web-server-on-the-internet-with-ngrok/>.

“Raspberry Web Server.” *Raspberry Pi Web Server - Writing CGI Scripts on a Raspberry Pi*, <http://raspberrywebserver.com/cgiscripting/writing-cgi-scripts-on-a-raspberry-pi.html>.