RASPBERRY PI DJ

Iulia JURCUȚA, Adrian LAZĂR, Politehnica University of Timișoara

May, 2018

1. **Repository**

Related files, as well as code can be found on:

<http://github.com/adrianl911/MS>

1. **User requirements**
2. The project should detect a user’s mood based on two questions asked by Alexa.
3. The project should detect four moods: happy, sad, angry, neutral but it is open for extension.
4. The system must turn on LEDs according to the song that it is played.
5. The system must provide a database with a user’s moods.
6. **System overview**

**AUDIO**

**SPEECH**

**ALEXA**

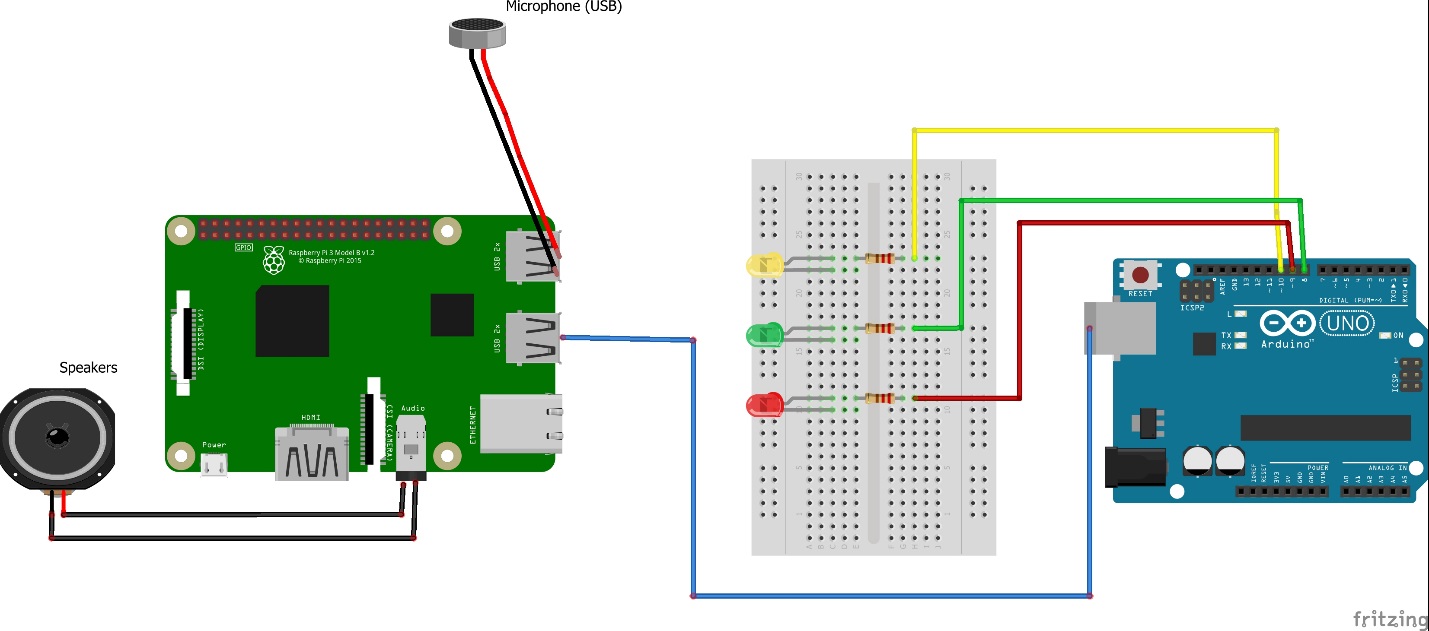
**PYTHON**

**ARDUINO**

Speech is done using a USB microphone connected to the Raspberry Pi. The user must invoke the custom skill, using the intent name (mood dj). Then, using an audio system (speakers, HDMI), Alexa asks the questions needed to calculate the mood.

After Alexa sends an event containing the classification result to the server, python classifies the answers into one of the situations: happy, sad, etc and it is ready to calculate the result.

Finally, the system is ready to play some music and light up some LEDs.

1. **Circuit design**

In order to develop our project, we used:

* **Raspberry Pi 3 Model B+**

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

We turned a Raspberry Pi3 Model B+ into an Alexa Echo using Alexa Skills Kit. Also, a python server running on Arduino manages the conversation between Alexa and the user.

* **Arduino UNO**

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

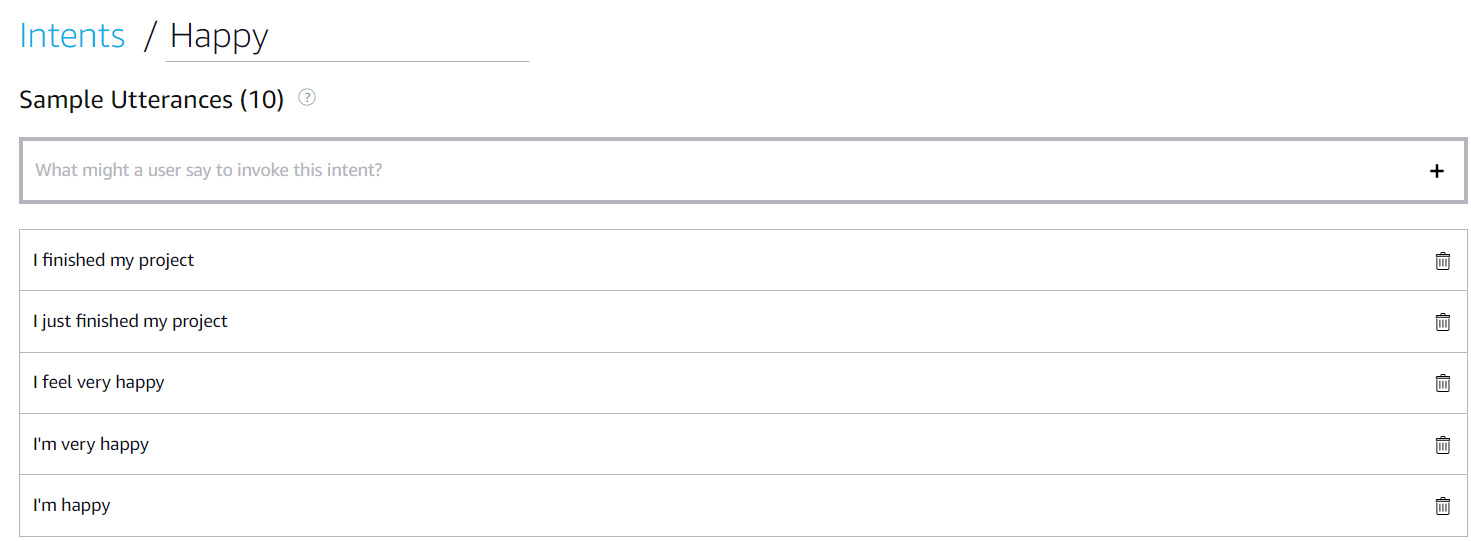
We used it in this project in order to light up some LEDs in syncronization with the song that is being played.

1. **Software design**

* **Amazon Alexa**

The first thing we did was to develop custom skills using Alexa’s services in order to manage the mood comprehension.

We used the skill builder in order to define four states: Happy, Sad, Angry and Neutral. These moods are declared as intents, and every intent has a list of utterances that represent the corresponding mood.

Finally, we created the endpoint to which the requests made to Alexa have to be sent in order to be processed. We used HTTPS as "Service Endpoint Type" and ngrok services in order to provide us the URL for exposing our local web server.

* **Raspberry configuration**

We followed a tutorial to install Alexa Voice Service to our Raspberry, turning it into a sort of homemade Alexa Echo.



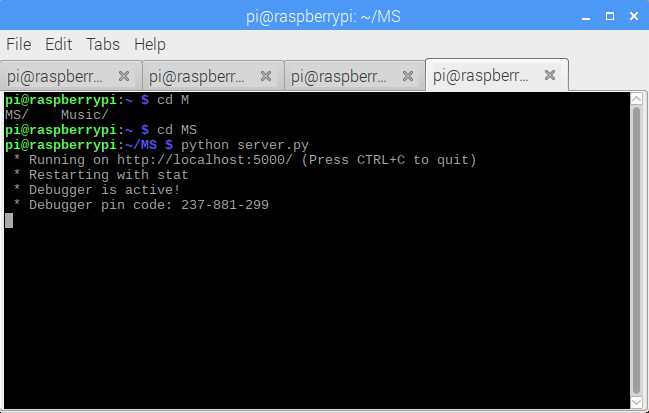
After that, we started developing the server that processes the informations received from the custom skill.

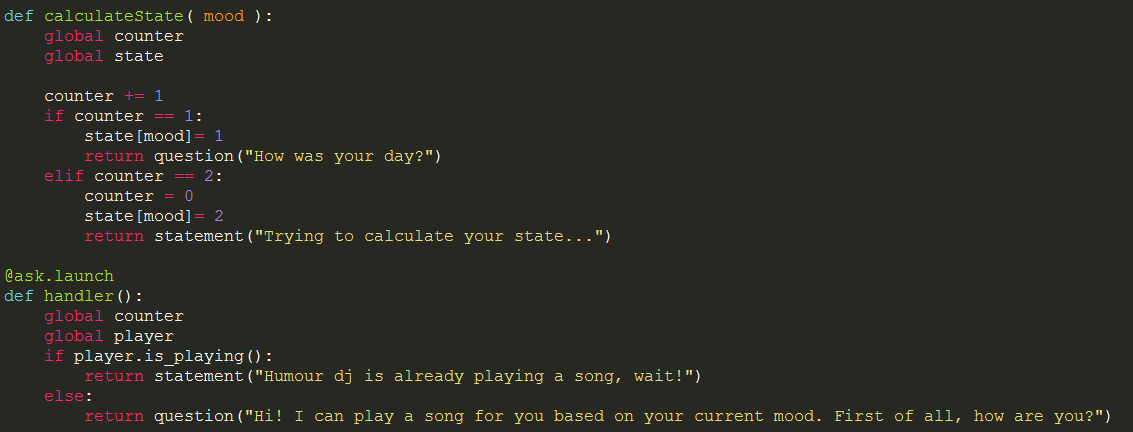
We used python as programming language and Flask as web framework. Flask is used for managing statements and questions.

Moving on, we followed another tutorial on how to import and use the vlc media player that we will use for playing songs.

Finally, we managed to make the server to listen on localhost:5000, exposed by ngrok.

When someone invokes the custom skill, the server intercepts the "Launch" event and returns a welcome message followed by the first question chosen to understand the user's mood: "How are you?"



What the user answers will be classified and the server will handle the event fired from the custom skill. Then, it will calculate the gathered results, it will play a song according to the mood and will send a signal to Arduino, using serial.

* **Arduino**

The algorithm for the LEDs part uses the Fast Fourier transform of the audio signal from the mp3 to analyze certain frequencies, and lights an LED if certain ranges of frequencies go above the allotted "sensitivity" value.

1. **Further work**

When we first started the project, we were willing to also make a database for a user to be able to see a report: how happy, sad, angry, he/she was in an amount of time. For this, we thought about using Firebase we did not manage to develop a solution.

All in all, out project provides a fun way of spending time for a user, and for us is was a great way to learn to overcome obstacles (a lot of errors and challenges from the Raspberry Pi) and to improve our knowledge.

**References:**

1. <https://github.com/alexa/alexa-avs-sample-app/wiki/Raspberry-Pi>
2. <http://www.instructables.com/id/How-to-Make-LEDs-Flash-to-Music-with-an-Arduino/>
3. <https://blog.computerbacon.com/playing-audio-in-python-with-libvlc.html>