BU

Senior Design

ENG EC 463

Memo

To: Professor Pisano

From: Arley Trujillo, Shivani Bhatia, Laura Salinas, Priya Kapadia, Steven Graham

Team: 14

Date: 11/19/17

Subject: First Deliverable Test Report

[··]vobot

1. **Project Objective**

1.1. The overall objective of this device is to help children on the autistic spectrum develop their articulation and communication skills.

2. **Test Objective & Significance**

2.1. Android Application

2.1.1. The objective of the Android application is to create an interactive way for users to control the robots behaviors and to choose the word their child is currently practicing. The test for this deliverable is simply to run through a loop of the application being handled by a user and the robot displaying behavior accordingly. Specifically, the app should be able to send a command to the bluetooth mic/speaker and make it prompt the child to say a word. The app should listen for the child's response, compare the word said to the model word, and command the robot to provide positive reinforcement.

2.1.2. This deliverable is essential because it ties together the speech recognition aspect of the project and the bluetooth communication to the robot. Integration of these two aspects of the project is a foundation for the rest of the work that will be done moving forward. While the different components that come together in the application may change and become more complex, the essence of the application will remain unchanged. The only feature that would change in the future would be that the recording of the child's response would be from the bluetooth mic/speaker instead of the Android application itself.

2.2. Speech Recognition API

2.2.1. The speech recognition API is essential to our project because it is required to assess the child's progress. We need a way to quantify how the child's vocal skills are changing, and hence output a similarity score between the child's word, and the model word. Testing the speech recognition API involved downloading, modifying and compiling the code necessary to listen for user articulations and display to screen text understood. In addition to stand alone speech to text being tested, the API needed to be integrated with the Android application already built.

2.2.2. This deliverable was important for us to test because without it there would be no way to dictate what behavior the robot would have. As it stands the robot performs a behavior only if the received spoken word by user matches the model word exactly. It does not accept partial words or similar sounding words. Moving forward, speech recognition is also one of the most important technical aspects of the project. Building on from this, we could provide statistics on the child's progress over time.

2.3. Arduino Wireless Communication

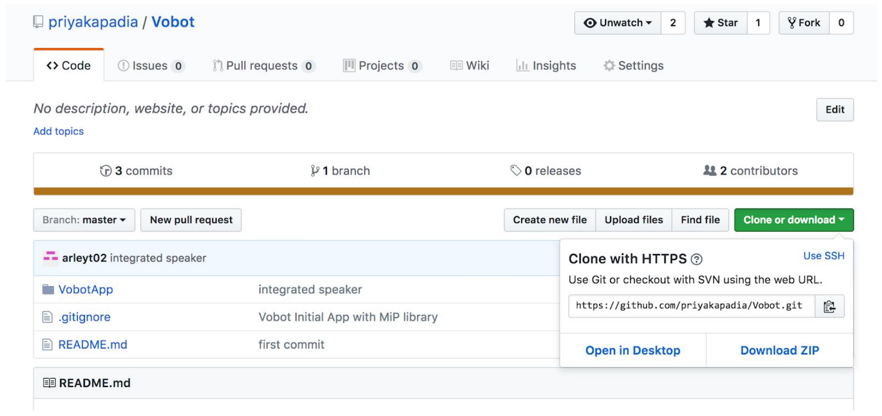
2.3.1. In order to compensate for the potential problem of two way communication using a bluetooth microphone and speaker module, we devised a way to listen to and display user articulations using an Arduino Uno and a microphone amplifier. By running speech recognition tests across multiple platforms we're able to have a backup plan incase we run into problems with the bluetooth module to robot communication pipeline.

2.3.2. Testing the Arduino speech recognition and wireless communication is important because we wanted to make sure we could interpret and send words a user is saying using a method other than the phone application using speech to text API, and having to rely on the bluetooth speaker module to listen for input. Having this method of speech recognition and wireless communication working gives us a backup option in case the bluetooth speaker is not able to listen for microphone input on command.

3. Equipment and Setup

3.1. Android App

1. To get started with the Android App, clone the repository from github: https://github.com/priyakapadia/Vobot



priyakapadia / Vobot

Unwatch 2 | Star 1 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

No description, website, or topics provided. Edit

Add topics

3 commits | 1 branch | 0 releases | 2 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

arleyt02 integrated speaker

VobotApp	integrated speaker
.gitignore	Vobot Initial App with MiP library
README.md	first commit

README.md

Clone with HTTPS | Use SSH

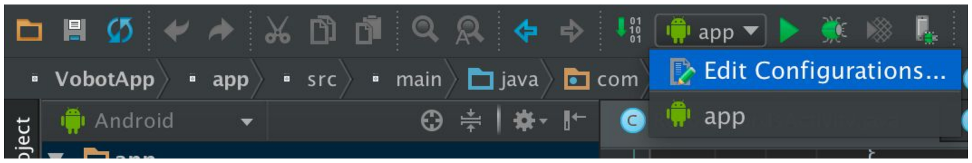
Use Git or checkout with SVN using the web URL.

```
https://github.com/priyakapadia/Vobot.git
```

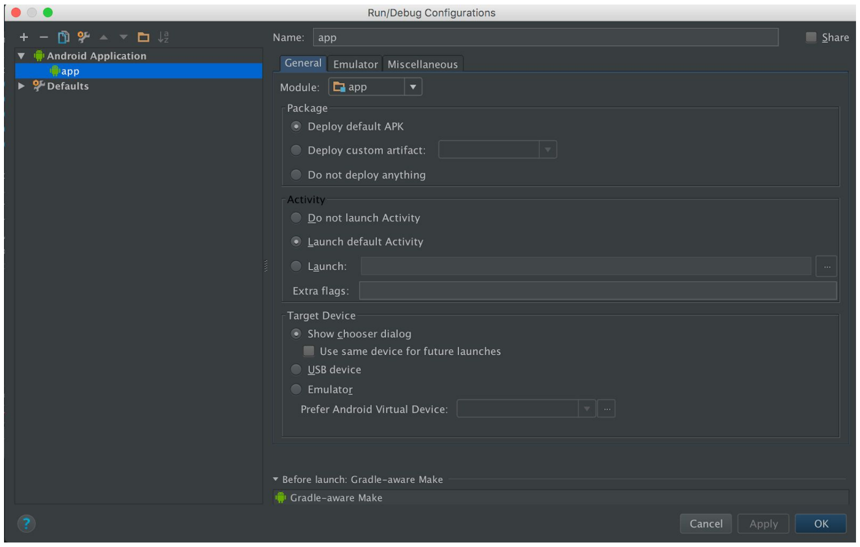
Open in Desktop | Download ZIP

   You may do so by clicking "Download ZIP" as shown above, or copying the url and typing "git clone https:..." on terminal. To do the latter, github needs to be installed on the machine.

2. Next, download Android Studio and import the project that you have just cloned. (File → Import Project)
3. Once you import the project, Android Studio should give you an option to sync Gradle files on the top right.
4. Then, edit configurations as shown in the image below:



5. Create a new android application build with the following specifications:



Run/Debug Configurations

Android Application
app
Defaults

Name: app | Share

General | Emulator | Miscellaneous

Module: app

Package
- [✓ Deploy default APK]
- Deploy custom artifact:
- Do not deploy anything

Activity
- Do not launch Activity
- [✓ Launch default Activity]
- Launch:

Extra flags:

Target Device
- [✓ Show chooser dialog]
  - Use same device for future launches
- USB device
- Emulator

Prefer Android Virtual Device:

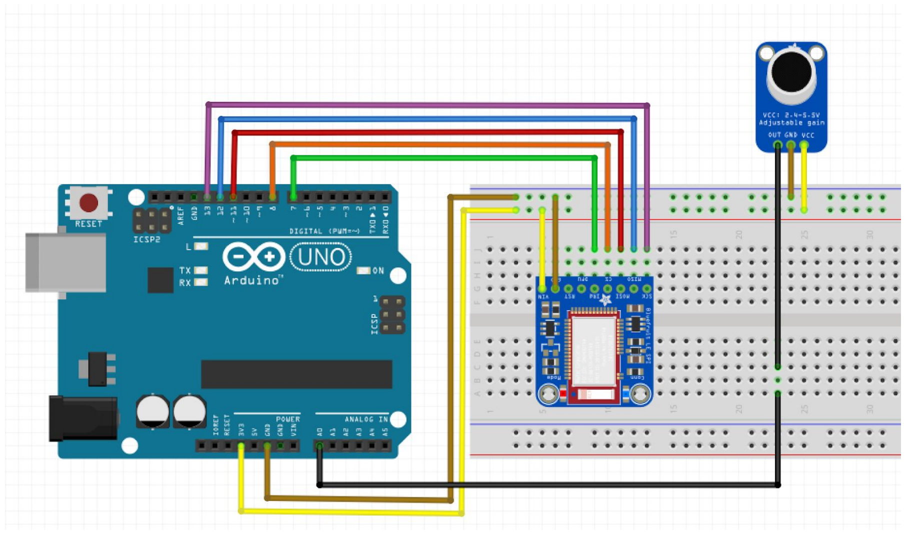
Before launch: Gradle-aware Make
Gradle-aware Make

Cancel | Apply | OK

6. Run the program using the play button and select the device you'd like to upload your app onto. Open the app on your phone, and start playing.

3.2. Arduino Wireless Communication

1. In order to implement wireless communication via the Arduino, begin by setting up your circuit as shown in the circuit diagram and pinout table below. You will need an Adafruit Bluefruit LE SPI Friend, Adafruit MAX4466 Electret Microphone Amplifier, and an Arduino Uno.

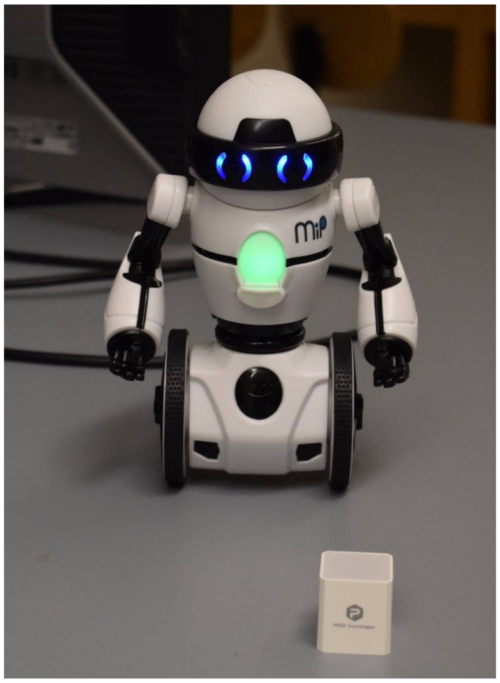


Arduino Pinout	Bluefruit SPI Friend Pinout	MAX4466 Microphone Pinout
13	SCK	-
12	MISO	-
11	MOSI	-
8	CS	-
7	IRQ	-
A0	-	OUT
3.3V	Vin	Vcc
GND	GND	GND

2. Once your circuit is assembled, clone the Arduino BLE repository from Github. You may access the repository at https://github.com/scbhatia/Arduino-BLE/.
3. Install the uSpeech and Bluefruit LE libraries for the Arduino found in the cloned repository. You can do this by copying the library directories in the cloned repository into the libraries directory for Arduino installed on your local computer.
4. On your phone, go to the app store and download the Adafruit Bluefruit application.

5. Open the Arduino sketch (found under VobotArduinoBLE -> communication). Follow the uSpeech library tutorial found at https://goo.gl/1Dr7Tt to calibrate the Phoneme values programmed in the setup() function in the sketch.
6. Once the phoneme values are calibrated, run the program. Open the Adafruit Bluefruit LE application on your phone and connect to the Adafruit Bluefruit SPI friend module. Once connected, click on UART.

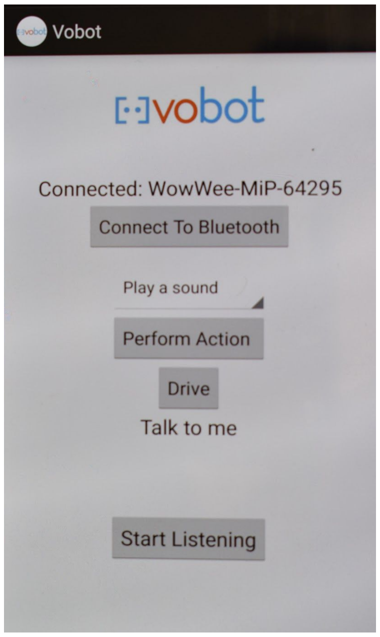
3.3. MiP Robot and Pred SmartCube Bluetooth Module



Shown on the right are the MiP Robot and Pred SmartCube Bluetooth Module, the main equipment we hope to use for our product. The MiP robot connects to the application and the Pred SmartCube via bluetooth and will respond accordingly based on the vocalizations determined by the speech recognition software and the reaction set by the phone application. The Pred SmartCube is what we hope to use for our wireless communication. The SmartCube would act as both our bluetooth microphone and speaker module.

4. Measurements and Data

4.1. Android Application

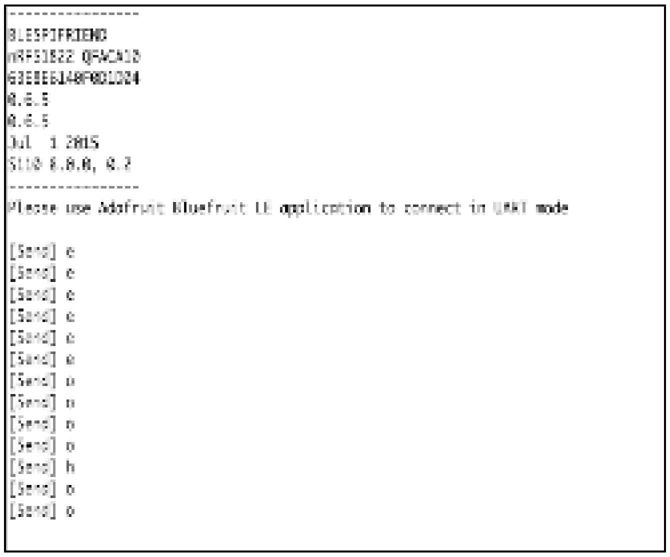


Connected: WowWee-MiP-64295

Talk to me

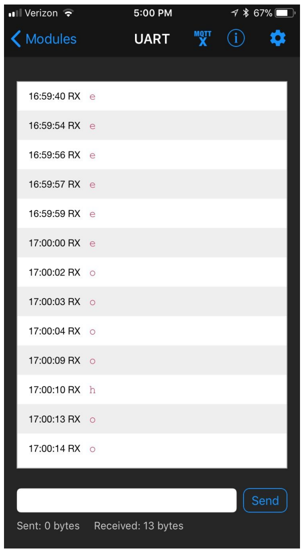
The app, as shown, has functionality to connect to the MIP robot. It can make the robot say something, or turn around. The app also has the added feature of turning the robot into drive mode and controlling it. Last, the start listening functionality prompts the microphone to say “Say the word mother”, record the user’s audio, and output a response to the robot if the audio matches the model word.

4.2. Arduino Wireless Communication



```
-----------------
BLESPIFRIEND
nRF51822 QFACA10
63E8E6140F0D1D24
0.6.5
0.6.5
Jul  1 2015
S110 8.0.0, 0.2
-----------------
Please use Adafruit Bluefruit LE application to connect in UART mode

[Send] e
[Send] e
[Send] e
[Send] e
[Send] e
[Send] e
[Send] o
[Send] o
[Send] o
[Send] o
[Send] h
[Send] o
[Send] o
```



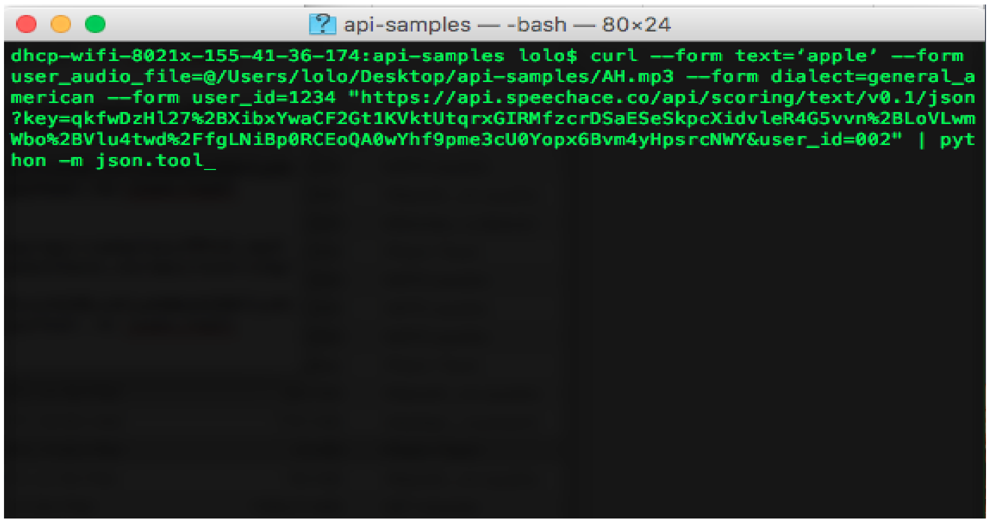
```
16:59:40 RX  e
16:59:54 RX  e
16:59:56 RX  e
16:59:57 RX  e
16:59:59 RX  e
17:00:00 RX  e
17:00:02 RX  o
17:00:03 RX  o
17:00:04 RX  o
17:00:09 RX  o
17:00:10 RX  h
17:00:13 RX  o
17:00:14 RX  o

Sent: 0 bytes    Received: 13 bytes
```

Left: Serial Monitor View from Arduino showing phonemes sent from Arduino Bluefruit LE module to the phone application. Right: Screenshot of received phonemes on phone application from Arduino Bluefruit LE module.

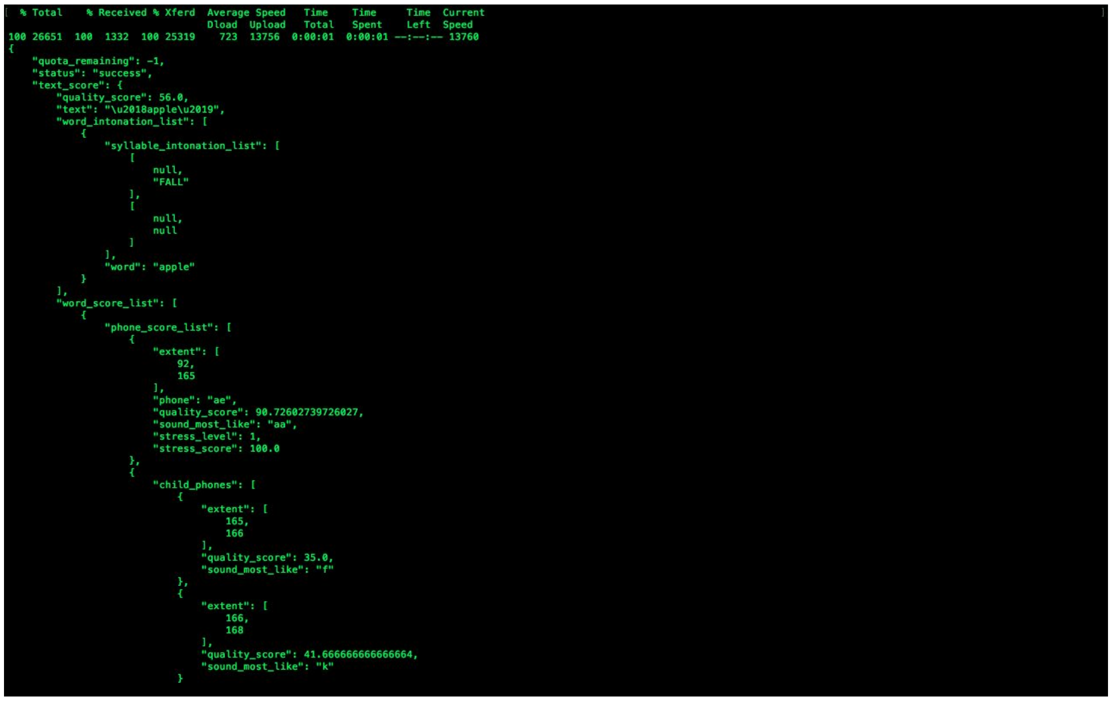
The communication system, as shown, is able to send information seamlessly from the Arduino Bluefruit LE system to the phone application (shown on an iPhone). The microphone amplifier is able to take in vocalizations, and the system is able to process them into text. The processed text is then sent via Bluetooth LE to the phone application. As shown, the received phonemes on the phone application match the sent phonemes on the serial monitor.

4.3. SpeechAce API



```
dhcp-wifi-8021x-155-41-36-174:api-samples lolo$ curl --form text='apple' --form
user_audio_file=@/Users/lolo/Desktop/api-samples/AH.mp3 --form dialect=general_a
merican --form user_id=1234 "https://api.speechace.co/api/scoring/text/v0.1/json
?key=qkfwDzHl27%2BXibxYwaCF2Gt1KVktUtqrxGIRMfzcrDSaESeSkpcXidvleR4G5vvn%2BLoVLwm
Wbo%2BVlu4twd%2FfgLNiBp0RCEoQA0wYhf9pme3cU0Yopx6Bvm4yHpsrcNWY&user_id=002" | pyt
hon -m json.tool_
```

Terminal Command to get similarity score for voice sample "AH" compared to word "Apple"



```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 26651  100  1332  100 25319    723  13756  0:00:01  0:00:01 --:--:-- 13760
{
    "quota_remaining": -1,
    "status": "success",
    "text_score": {
        "quality_score": 56.0,
        "text": "\u2018apple\u2019",
        "word_intonation_list": [
            {
                "syllable_intonation_list": [
                    [
                        null,
                        "FALL"
                    ],
                    [
                        null,
                        null
                    ]
                ],
                "word": "apple"
            }
        ],
        "word_score_list": [
            {
                "phone_score_list": [
                    {
                        "extent": [
                            92,
                            165
                        ],
                        "phone": "ae",
                        "quality_score": 90.72602739726027,
                        "sound_most_like": "aa",
                        "stress_level": 1,
                        "stress_score": 100.0
                    },
                    {
                        "child_phones": [
                            {
                                "extent": [
                                    165,
                                    166
                                ],
                                "quality_score": 35.0,
                                "sound_most_like": "f"
                            },
                            {
                                "extent": [
                                    166,
                                    168
                                ],
                                "quality_score": 41.666666666666664,
                                "sound_most_like": "k"
                            }
```

Detailed report of where differences in pronunciation are found

5. Conclusions

5.1. Android Application

5.1.1. Overall the app that we developed works seamlessly with the MiP robot and gives us the control functions we need to choose a word to work on and receive user input analyzed by Google Speech API. At the current stage we can hear audio from the bluetooth module but cannot record user articulations on it. A speaker's words are being received from the onboard phone microphone and in future implementations should come from the external bluetooth speaker.

5.2. Speech Recognition

5.2.1. In its current stage, the speech recognition aspect of the project cannot discern unintelligible speech. Fragmented articulations like “ahh”, “oo” will either throw an error or return the next best approximation to a known word. Speech recognition is successful in testing but will need to be modified or reworked into something that can break down phonemes for articulations that are not yet complete words.

5.2.2. We have found another speech recognition API (SpeechAce) that returns similarity scores compared to a model text. The software generates a detailed breakdown of what syllables and phones were mispronounced in addition to assessing a quality score for input. Moving forward this

SpeechAce API is what we will attempt to integrate the Android app with in order to handle incomplete articulations.

5.3. Arduino Wireless Communication

5.3.1. At the moment, the Arduino wireless communication works smoothly to transmit data from the Bluefruit SPI Friend to the application; however, transmission of data from the application to the Arduino based on data received has not yet been implemented. In the future, the open-source application should be modified to send messages based on the data received as opposed to input in the UART module. Additionally, the functionality of the uSpeech library is limited. The program only detected ‘e’, ‘f’, ‘h’, ‘o’, and ‘s’ phonemes with 40% accuracy. As a result, these are the only letters from the speech-to-text conversion that are transmitted. Ideally, this library would be more refined with the integration of the speech recognition software of our project.