CSCI 4126/6609, Summer 2017

**Milestone 4: Drumstick**

Instructor: Dr. Derek Reilly

Submission date: July 27, 2017

**Group 1**

Scott Bailey B00636779

Yanan Zhang B00728680

Athi Narayanan B00736197

Xuanyu Wang B00750530

Nupur Shah B00756907

Priya Gosain B00763986

Linhui Zhao B00575548

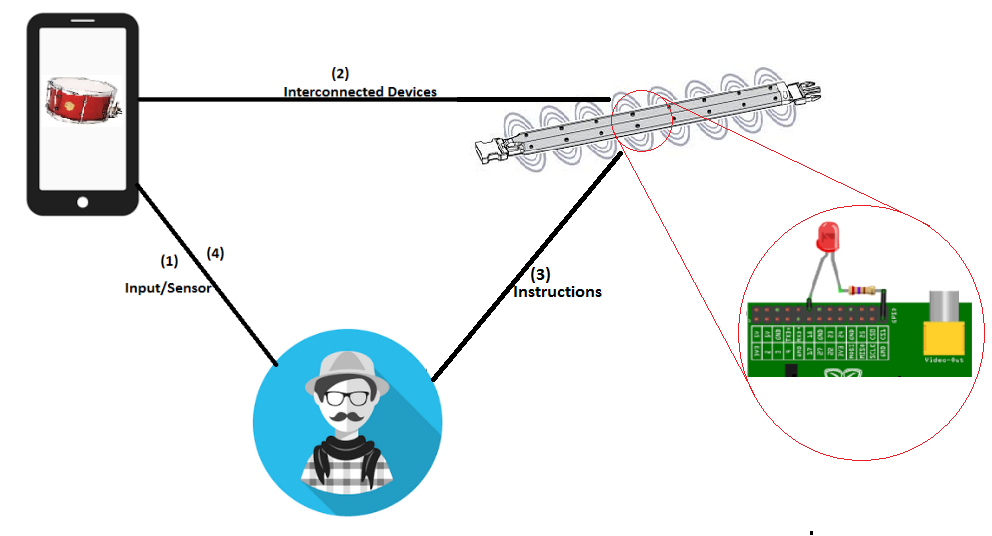
**Smart Drumstick**

# **Introduction**

Smart Drumstick is device that aims to revolutionize the music learning, practising, and playing experience. Smart Drumstick uses a combination of modern gadgets and unconventional mode of interaction that provide an unique experience to users. Smart Drumstick is designed to achieve a learnability that is on par with the traditional method while conferring users with the higher level of comfort. Smart Drumstick is feature rich application that utilizes the omnipresent mobile technology coupled with a low budget electronic bracelet.

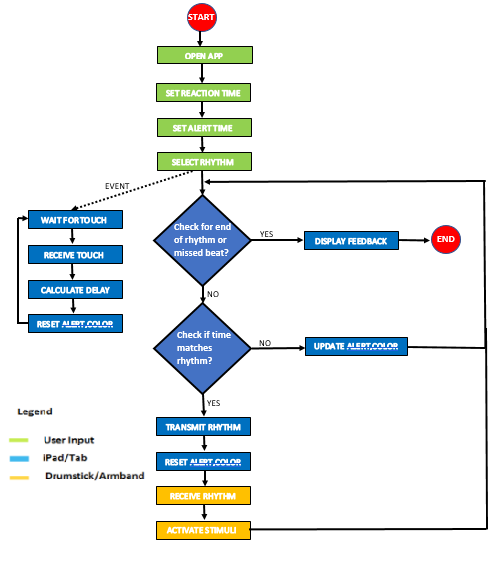
# **Architecture**

The Smart Drumstick consists of two physical components, the mobile phone/tab and the wearable unit. The mobile phone/tab can be any touch based display device that runs on Android operating system. The wearable unit is made of Raspberry Pi, LED, Vibration motor and its power source. The mobile phone acts as a primary interface with user. The user can input response time, select rhythms and view feedbacks/alerts at the mobile. The selected rhythm is transferred to the wearable unit to instruct the user. The devices can be either communicated through a common WiFi network available or through tethering functionality of the mobile. Serialized sockets are used for data transfer. LED and vibrators are used to instruct users on playing the rhythm. Using the rhythm user then hits the touch sensors of the mobile as an action of playing the drum.



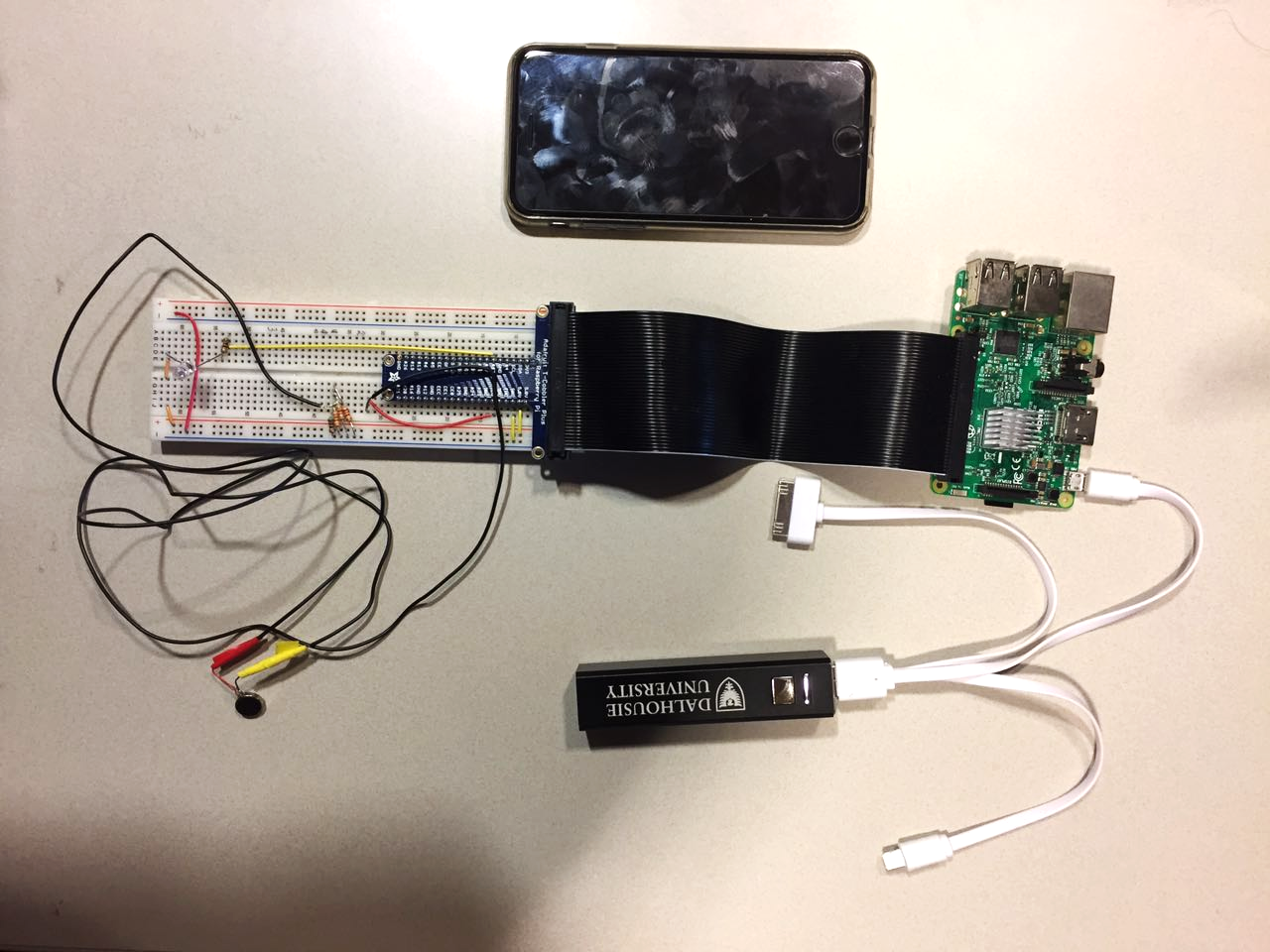
# **Functional Design**

The primary functionality of the Smart Drumstick is to instruct users to hit the drum according to the selected rhythm. The Smart Drumstick also allows additional functions of setting expected stimuli-to-response time, dynamic alerts to users on increase in delay and practice statistics at the end. Stimuli-to-response time will offset the delay calculation of the device to include standard human reaction time. Dynamic alert changes the colour of the screen to indicate delay of every beat. Practise statistics at the end provides better understanding of user’s performance.



# **Deployment**

## Hardware Setup



## Equipment Required

|  |  |
| --- | --- |
| Android Mobile Phone | 1 |
| Rasbperry Pi 3 | 1 |
| Pi T-Cobbler Plus - GPIO Breakout Ribbon Cable | 1 |
| Standard Electronic Breadboard | 1 |
| 5V Battery Pack - Micro USB | 1 |
| Micro USB Cable | 1 |
| LED | 1 |
| Vibration Motor (1 cm disk) | 1 |
| Insulated Wire | 1 meter |
| Resistors | 2 X 510Ω  2 X 820Ω  1 X 120Ω |

## Setting Up the Raspberry Pi

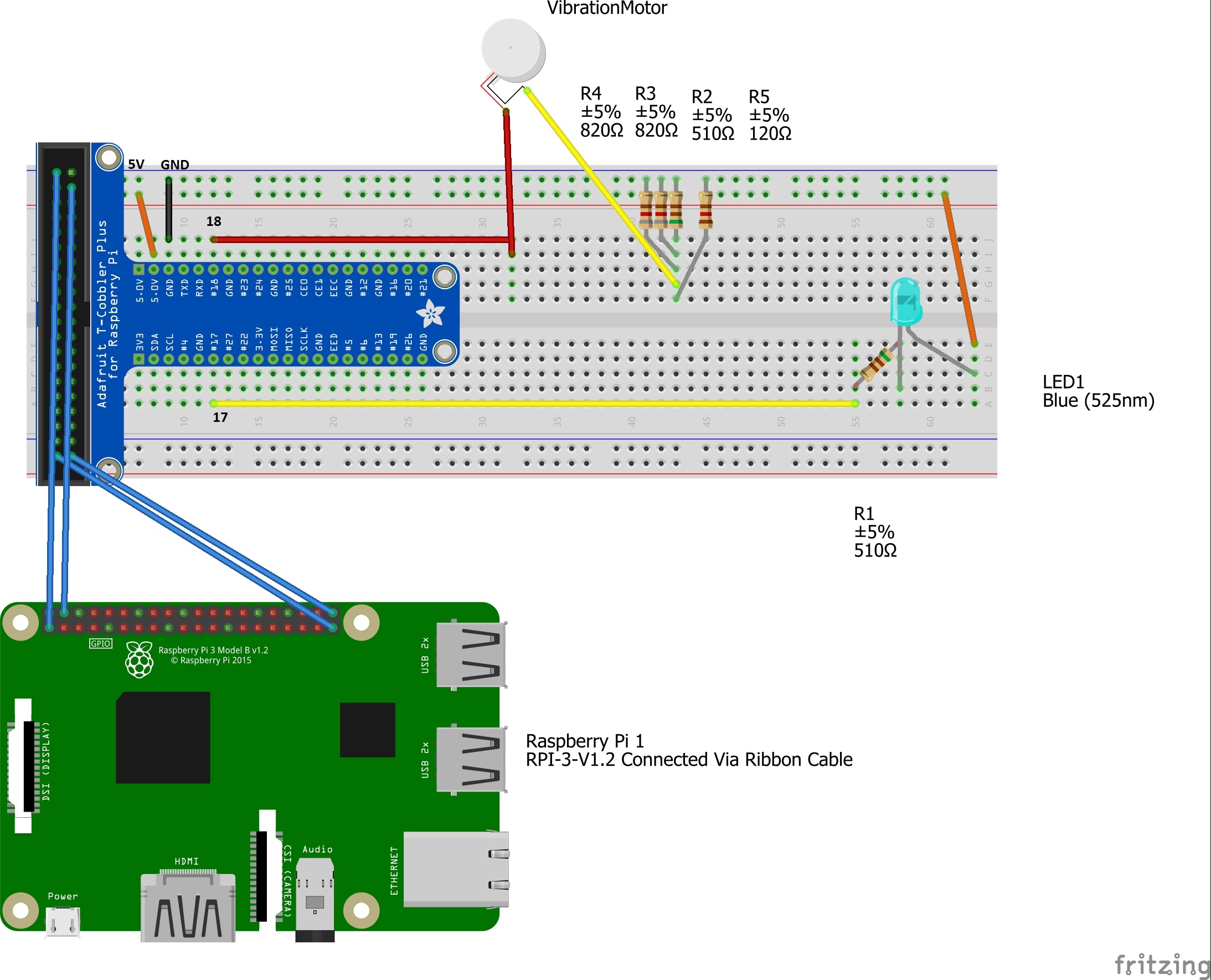
There are some steps you need to follow.

1. The Raspberry Pi provided in the Smart Drumstick kit will be pre-loaded with the required operating system and applications.
2. To power the Raspberry Pi on, simply connect the battery pack to the device via the MicroUSB cable and turn the battery pack on. Do this step once all the wiring connections are made. *As an alternative to the battery pack, any typical 5V cellphone charger which uses MicroUSB can be used to power the system from a wall outlet.*
3. Enable a hotspot on your Android device and the Raspberry Pi will automatically connect to it via WiFi.
4. The prompting device server application is automatically started when the device boots.
5. You will know that the application is ready when the vibration motor and LED pulse once after about 30 seconds from first powering on the Raspberry Pi.

## Setting Up the Prompting Devices

1. The prompting devices provided in the kit are the LED and the vibration motor.
2. Begin by connecting the GPIO ribbon cable to the Raspberry Pi’s GPIO pins.
3. Connect the male side of the GPIO ribbon to the breadboard.
4. Connect the LED and Motor as shown in the wiring diagram below.

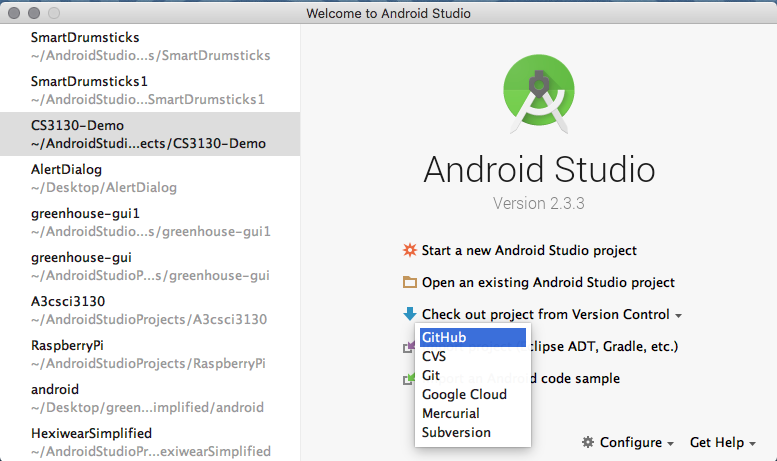
* Note that for the vibration motor, the yellow and red diagram wires correspond to the yellow and red taped leads in the real system.



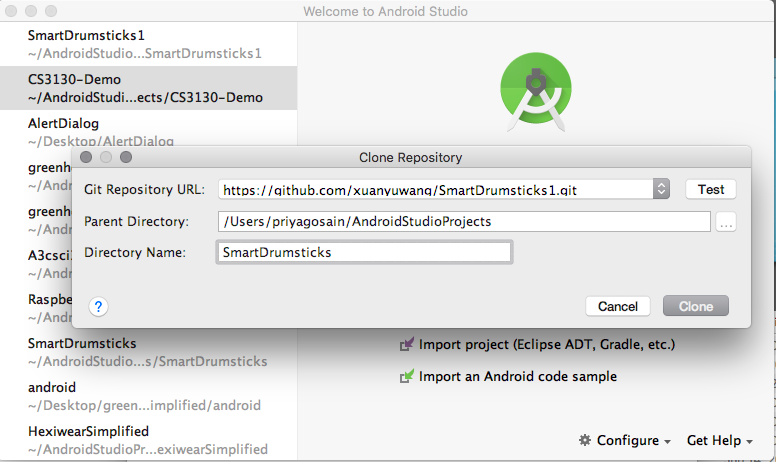
## Installing and Starting the Android Application

#### From Github repository

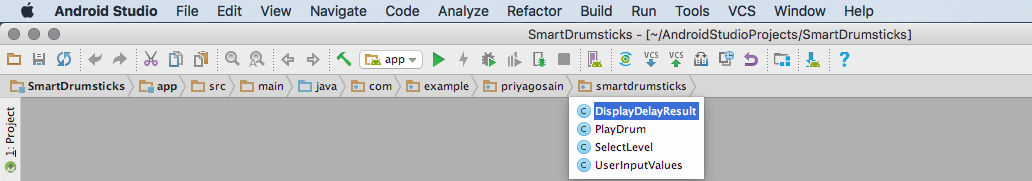
Step-1: Open Android Studio and select GitHub from the “Check out project from Version Control” option.

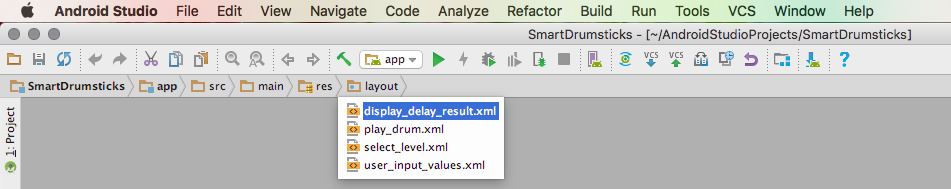


Step-2: Select the repository in the Git Repository URL from where the code has to be cloned. '<https://github.com/xuanyuwang/SmartDrumsticks1>’. Select Yes to checkout from Version Control.



Step-3 : SmartDrumsticks is cloned in the android studio. Access the activity java files and layout xml files as shown below.

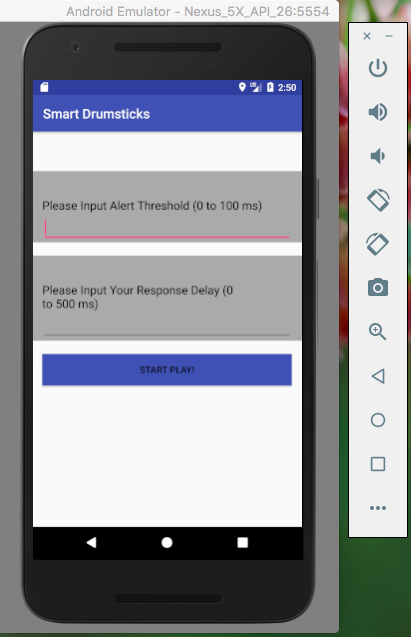




A pre-configured and optimized Android Virtual Device testing on the emulator is recommended to test the working of application.

Please Note : In the activity file - PlayDrum, the IP address of the system being used is to be changed in order to make connection between the host and client servers. The IP and port we have used is "192.168.43.10" and “8080”

Step-4 : Now run the application. It should successfully connect with the emulator as shown below taking the user to the first screen of the application.



#### Mobile App:

First, download the apk file on your computer and then connect your android phone.

Here is the link to apk file: https://github.com/xuanyuwang/SmartDrumsticks1/releases.

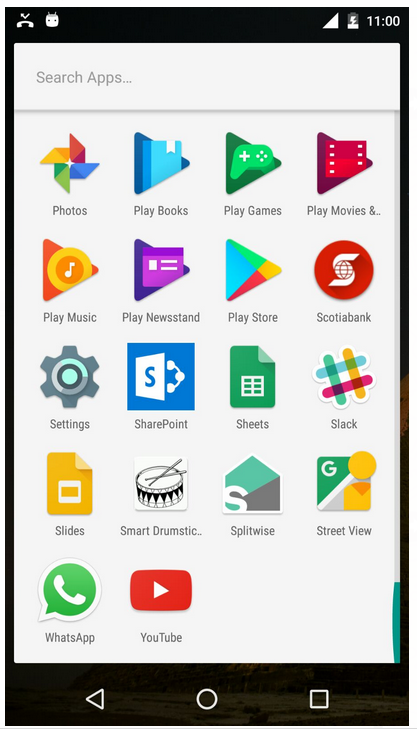
Choose the phone to be simply charged or used it as a media device. Copy the apk file into the folder of your choice in your android phone. Find the apk file and then tap install. Make sure to switch on the tethering to build a connection between the android phone and raspberry pi.

# USER MANUAL

Once all the hardware and applications are set up as detailed in the setup section, you are now ready to use the Smart Drumsticks learning system.

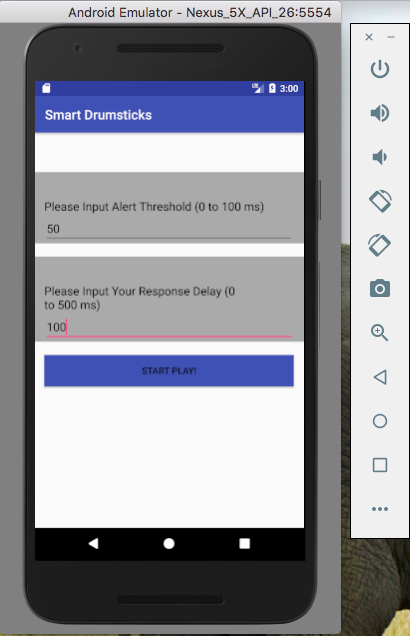
### Step 1

Open the Smart Drum mobile application.



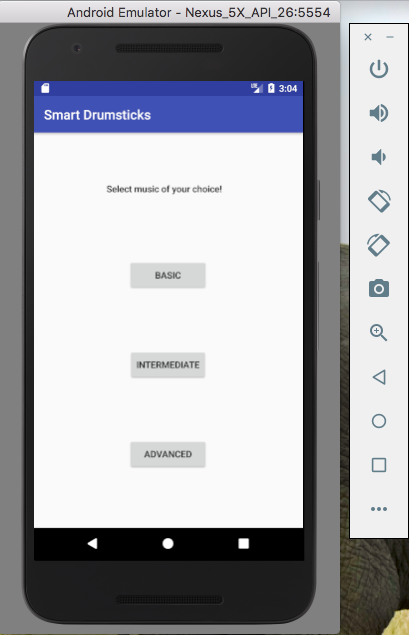
### Step 2

Select the input alert threshold and response delay in the ranges mentioned.

****

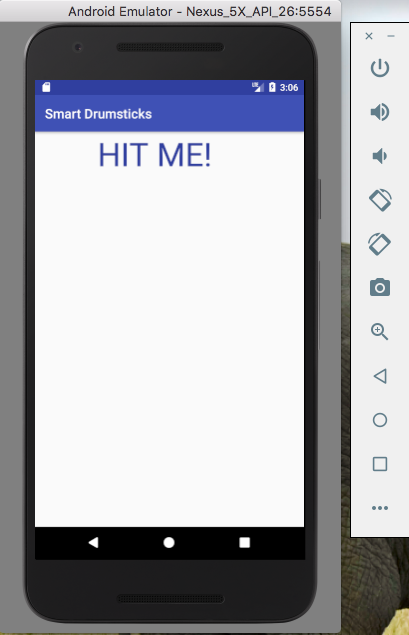
### Step 3

Select the level of rhythm.



### Step 4

Wait for the LED to flash or the vibrations as these two (LED and vibrators) indicate the time to hit the screen based on the selected rhythm. Hit the screen where it shows the text “Hit Me”.



### Step 5

Once the rhythm is over, the screen will calculate the average/maximum/minimum delay of all the hits. The delay for each hit is the difference between the expected time of hitting the screen and actual time of hitting the screen.

The lower the delay the better the drummer is.

