
COMP4801 Final Year Project

FYP15027 Final Report

Internet of Thing Application

TOUCH3

Department of Computer Science, The University of Hong Kong

Supervisor

Dr. Lau Vincent

Student

Poon Chi Hin (2012587555)

17-04-2016

Abstract

Robots play a significant role in human development. Human usually remotely manipulate robots to finish different dangerous and inaccessible tasks. However, the remote-control system with physical buttons is not appropriate in certain situations, such as doctors in operating room and astronauts in space. And this project is to provide a gesture-control system to solve this problem. This paper describes the design, implementation and evaluation of TOUCH3, a web application that interconnect MYO armband and robots. With TOUCH3, users are able to use hand gestures to manipulate robots and monitor the connecting devices. This paper introduces the four main functions of TOUCH3. Also, it discusses the major problem in this project, the limited amount of hand gestures provided by MYO armband. The solution, combining hand gestures with arm motions, will be mentioned to solve this major problem. Currently, a beta version of TOUCH3 is released. All in all, TOUCH3 can provide an unexpected touch-free experience to user in robot control, which may contribute to the future robot development.

Acknowledgement

This project is a final year project for computer science student. And it consumed huge amount of work and research. However, this project would not have been possible without a support of many individuals and organizations. Therefore, I would like to extend my sincere gratitude to all of them.

Thank you Dr. Lau Vincent for his supervision and his valuable advises. Also, thank you Mr. Lau Brian from Globalactive Technology Limited for lending iDroid robot. Thank you the Department of Computer Science, the University of Hong Kong for providing different items and financial support. Finally, thank you everyone's opinion and suggestions for making TOUCH3 better.

Table of Content

1. INTRODUCTION	5
2. MOTIVATION	5
3. OBJECTIVES	6
4. PREVIOUS WORK	8
5. DELIVERABLES	9
6. APPROACH AND METHODOLOGY	10
6.1 Use-Case.....	11
7. IMPLEMENTATION DETAIL	12
7.1 User Interface Design.....	12
7.1.1 Status Tab UI design.....	14
7.1.2 Recording Tab UI design.....	15
7.1.3 Log Tab UI Design	16
7.1.4 Customization Tab UI Design	17
7.2 Implementation of Five Functions	18
7.2.1 To Interconnect MYO Armband And Robots	18
7.2.1.1 To Connect With MYO Armband.....	19
7.2.1.2 To Connect With Robot (iDroid).....	20
7.2.2 To Match Hand Gestures To iDroid's Motions.....	21
7.2.2.1 To Detect User's Hand Gestures.....	22
7.2.2.2 To Match Hand Gestures to iDroid's Motions	25
7.2.2.3 To Play Sound Files As Command	26
7.2.3 To Customize User Gestures.....	26
7.2.4 To Present The Information To The Users In A Proper Manner.....	26
7.2.5 To Perform A Sequence of iDroid's Motions By Using A Simple Hand Gesture....	27
7.3 Summary	28
8. RESULTS	28
8.1 User Experience Assessment.....	29
8.1.1 First User Experience Assessment For UI Design Prototype	29
8.1.2 Second User Experience Assessment For Final Product Of TOUCH3.....	29
8.2 Accuracy of Detecting Hand Gestures	32
8.2.1 First Accuracy Test For TOUCH3 Detecting Hand Gestures.....	32
8.2.2 Second Accuracy Test For TOUCH3 Detecting Hand Gestures.....	33
9. PROBLEMS ENCOUNTERED	34
9.1 Self-defined Hand Gestures	34
9.2 Accuracy Of Detecting Hand Gestures.....	36
9.3 Running Environment	37
9.4 Incomplete myo.js Library.....	38
10. CONCLUSION	38
10.1 Future Work.....	39
11. LIST OF FIGURES	41
12. LIST OF TABLES	41
13. LIST OF EQUATIONS	41

14. TOUCH3 USER GUIDE	42
14.1 Launch TOUCH3.....	42
14.2 Status Tab	42
14.3 Recording Tab.....	43
14.4 Log Tab.....	43
14.5 Customization Tab	44
14.6 Terminate TOUCH3.....	45
15. REFERENCE.....	46

1. Introduction

Most of the current remote-control systems are using physical buttons to control different electric devices. This project is trying to make a change on robot control. Instead of using remote-control systems with physical buttons, an advanced technology, MYO armband is used. MYO armband is remote with a gesture-control system. It has a powerful sensor, called electromyography sensor [1]. This sensor is able to read the electrical signals from muscles. Therefore, MYO armband can recognize users' hand gestures after detecting the electrical signals in forearm. This project is to make use of MYO armband and to develop a web application, TOUCH3.

TOUCH3 aims to connect MYO armband and robot and manipulate a robot by gesture-control system, so as to provide a touch-free experience to users and be suitable for scenarios where touch is not preferred. For example, doctors interact with different medical devices by MYO armband in operating room, and astronauts complete their dangerous mission with MYO armband's assistance.

In this final report, the implementation details, final results and problems are reported, so the final product can be evaluated and the improvements can be provided.

2. Motivation

Human make use of robots to complete different dangerous and inaccessible tasks though remote-control system, but most of the remote control systems use many physical buttons or many commands to complete a simple task. Therefore, the development of robot is facing a bottleneck due to the complex and inconvenient control system. So MYO armband has been developed to solve this problem, and this project is to develop a web application TOUCH3. The purpose of TOUCH3 is to provide a touch-less control system for user to control the robot in different scenarios. And there are two examples showing gesture-control system is significantly useful.

First, doctors in operating room are required to locate the affected part by controlling the diagnostic imagery. Due to avoiding any infection, they cannot touch anything with their hands, so doctors need nurses to help and more manpower is needed to complete the surgery. With the help of MYO armband, doctors will be able to navigate the diagnostic imagery by using hand gestures [2]. In addition, they can also interact with other medical devices without nurses' help, so nurses can focus on other duties. As a result, the efficiency and the success rate of surgery will be increased.

Apart from medical purpose, astronauts can also complete their mission with the gesture control system. Under zero gravity state and wearing space suit, using remote with physical buttons is not user-friendly to them, and reduces the working efficiency. So gesture control system is significantly useful in this case. Astronauts can easily control different devices by using a simple hand gesture, so they can work faster and reduce the time of exposure to space. As a result, the risk of the mission will be reduced and the safety of astronauts will be secured.

Hence, using gesture control system to remotely control different devices is especially useful under certain scenarios. And this project is to show the great potential of MYO armband by developing a web application, TOUCH3, which can use MYO armband to manipulate robot.

3. Objectives

This project aims to remotely control robots by MYO armband, and aims to develop a user-friendly web application, TOUCH3, which allows users to monitor robot and define customer gestures. It provides four main functions:

1. To interconnect MYO armband and robots
2. To match hand gestures to robot's motions
3. To customize user gestures
4. To present the information to the users in a proper manner

5. To manipulate robots performing a sequence of actions by using one gesture

First, TOUCH3 is required to connect MYO armband with robot. Since MYO armband supports the Bluetooth Smart (Bluetooth 4.0 Low Energy), it is able to communicate with the smart devices having Bluetooth radio. Applying BLE technology, TOUCH3 can exchange information between MYO armband and robot. As a result, TOUCH3 acts as a platform. It is able to handle the data input from MYO armband and data output to robot, so it can accommodate interconnection and intercommunication with MYO armband and robots.

Second, TOUCH3 is to match hand gestures to robot's motions. An ARM Cortex M4 processor in MYO armband provides a strong processing power to recognize muscle activities and sends detected gestures to smart devices. Therefore, TOUCH3 can read the gesture input from MYO armband and then sends commands to robot, so as to control the robot.

Third, this project would like to develop a function, which allows users to define their gesture control setting. Since MYO armband contains two types of sensors, TOUCH3 can make use of them to recognize hand gestures. Firstly, MYO armband has eight medical grade stainless electromyography (EMG) sensors, thus it can monitor the electrical signal in forearm muscle. Secondly, it contains a highly sensitive nine-axis inertial measurement unit, which are able to detect the forearm motion in any direction. With the powerful sensors, MYO armband is able to recognize numerous gestures, and provides lots of gestures to user. So TOUCH3 allows users to customize their own gesture control setting.

Fourth, TOUCH3 should be able to present different information to users in a proper manner. Therefore, a concise and user-friendly user interface will be developed and implemented in website. Also, MYO armband contains vibration motor. It is able to produce several types of vibration. TOUCH3 makes use of this feature, and provides haptic feedback to users. Therefore, users are not required to hold their smart devices all the time during controlling robot.

Fifth, TOUCH3 would like to provide a smooth robot manipulation to users, so a function that control the robots to perform a sequence of actions by using a simple gesture is developed. Users only have to input one gesture, TOUCH3 will send a sequence of commands to robots, and then the robots will react according to the commands.

The goal for this project is to develop a web application, TOUCH3, which is able to have above four main functions with a user-friendly user interface. As a result, users are able to remotely control robot with MYO armband. Also, robot can perform a sequence of complicated actions by a simple gesture, so as to enhance the efficiency of gesture control system and to perform a smooth robot's manipulation.

4. Previous Work

MYO armband has an application store. It likes the Apple app store and the Google market. The application store is open to public. There are generally two types of application, one is for remote control, and one is for entertainment.

In the MYO market, there are some applications make the MYO armband to replace the remote control. For example, a PowerPoint connector can connect the MYO armband with the PowerPoint, so users are able to use hand gestures to switch the slideshow. Also, there are other connectors, which are able to connect the MYO armband with different music players. Therefore, users can enjoy the convenience brought by gesture control.

Apart from connection application, there are also some applications for entertainment. For example, users play Fruit Ninja or Minecraft with MYO armband, so they do not need any mouse or touch screens to control, instead, the users can enjoy touch-less gaming experience. Also, there are some applications, which connect MYO armband with the toys, so MYO armband will acts as a controller to manipulate the toys.

Moreover, there are also many great projects developed with MYO armband. For instance, researchers from Thalmic Labs and Johns Hopkins University developed an application, which successfully helped a man who lost his arm to cancer wirelessly control his prosthetic arm. The application receives the data of muscle activities in arm from MYO armband, processes the information and then sends commands to the prosthetic. This project proved that MYO armband can integrate with different smart objects, doing something we cannot be done in the past.

The product of this project TOUCH3 is a connector, but also is a controller. It has a fundamental function as similar as the previous works. However, it also aims to enhance the efficiency of gesture control system and to perform a smooth robot's control. For example, the robot can perform a sequence of complicated action by a simple gesture. This feature is not provided by previous works, therefore, this project would like to achieve this goal and contribute to the future robot development.

5. Deliverables

Deliverables of phase one (inception)

1. Detail Project Plan
2. Project Website
3. User Interface of TOUCH3

Phase one was completed. All deliverables of phase one was finished. The detail project plan and a project web page were uploaded to the project website, <http://i.cs.hku.hk/fyp/2015/fyp15027>, also they were submitted to supervisor, Dr. Lau Vincent. The prototype of user interface was designed and a user experience assessment was conducted.

Deliverables of phase two (elaboration)

1. Implementation of Four Functions
2. Detailed Interim Report

3. Modification of UI Design

Phase two was completed. The progress was discussed in the interim report. After phase two, TOUCH3 was released with the modified UI design. It is able to detect the hand gestures and the self-defined hand gestures. An interim report was also provided to assess the progress and the achievement.

Deliverable of phase three (construction)

1. Implementation of Beta Version Application
2. Final Report

After phase three, a beta version application is released. It is able to control the robot by using gesture control. The project objectives and progress will be evaluated in the section 7 and 9, and a final report is provided.

6. Approach and Methodology

This project is to develop a web application with four functions to control robots by MYO gesture-control system. To elaborate on the system design, a use-case diagram is showing below.

6.1 Use-Case

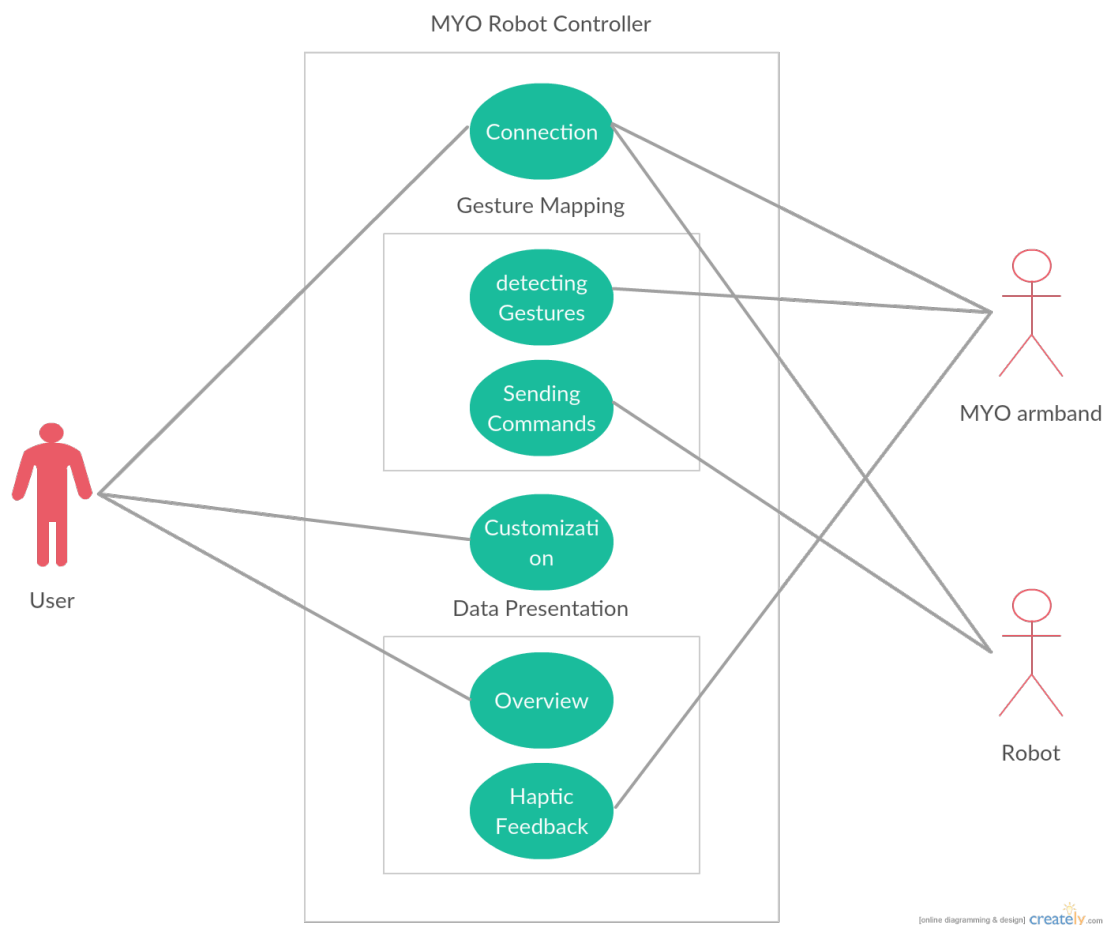


Figure 1: TOUCH3 Use-Case Diagram

Figure 1 is the use-case diagram for TOUCH3. There are one primary actor on left hand side, and two secondary actors on right hand side. In the middle, it is TOUCH3 with the four functions. This use-case diagram can explain the interaction between functions and actors.

The primary actor for TOUCH3 is the user. The user is interested in controlling robot by MYO armband, so they would like to perform three functions. They would like to connect certain MYO armband with certain robot, so the BLE technology is applied to interconnect these devices. In this project, BLE supports TOUCH3 acting as a platform of input and output data while connecting with different devices. With the software development kit (SDK) of MYO armband and robot, TOUCH3 forms a platform to handle the communication between two devices. Also, the user would like to define the gesture control setting, so

TOUCH3 is developed this function with a Node.js library, MYO.js. Finally, the user is able to interact with the UI mentioned before, to monitor the robot and MYO armband.

The secondary actors in this use case are MYO armband and robots with a BLE radio. MYO armband is able to send the detected gestures to smart devices. And also TOUCH3 is able to request the MYO armband to vibrate, so as to give a haptic feedback to the user. Robot also contains a BLE device, thus it is able to receive the commands from TOUCH3, and performs certain motion.

7. Implementation Detail

7.1 User Interface Design

After phase one, a prototype of user interface was designed. Figure 2 shows the prototype. There are three tabs. First, the connection tab is used to select the connection of MYO armband and robot. Second, the gestures tab is used to display the gesture control setting. Third, the setting tab is for users to modify the gesture control setting and haptic feedback.

Also, a user experience assessment was conducted to collect the user opinion. The prototype was tested by six engineering students, after reading the project plan and visiting the project website. Most of them mentioned about the style of UI is too concise and unattractive, since the application lacks in animation and does not have any icons. Although it provides the necessary functionalities, it is plain. However, they said the website design is concise but also fashionable, they suggested applying the style of website into the mobile application.

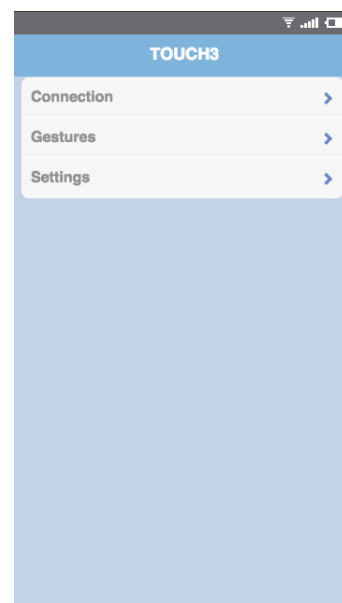


Figure 2: TOUCH3 UI Prototype

After analyzing the user opinion, the UI design was modified. Also, since the application was developed as a web application instead of mobile application, the reason will be discussed in the section 9, the UI design was re-designed with user opinion, similar style of the project website was applied.

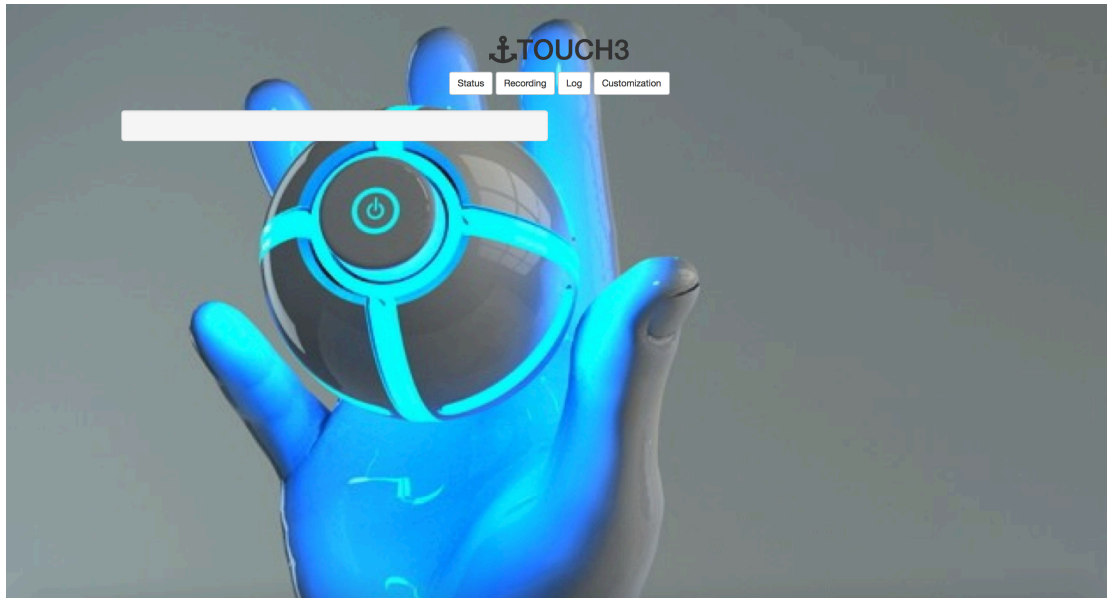


Figure 3: Modified TOUCH3 UI Design

Figure 3 shows the modified UI design of TOUCH3. It is a one-page web application developed on Node.js. It has four tabs. First, the Status tab shows the information of connected MYO armband. Second, the Recording tab is used to modify a sequence of robot's motion, it provides a function that users can define a sequence of robot's motion and use one gesture to execute the sequence of motions. Third, the Log tab shows the detected hand gestures in word. Fourth, the Customization tab is used to modify the gesture control setting. This UI is designed with the user opinion, and another user experience assessment is conducted to collect user new opinion. The result will be showed and discussed in the section 8.1.2.

7.1.1 Status Tab UI design

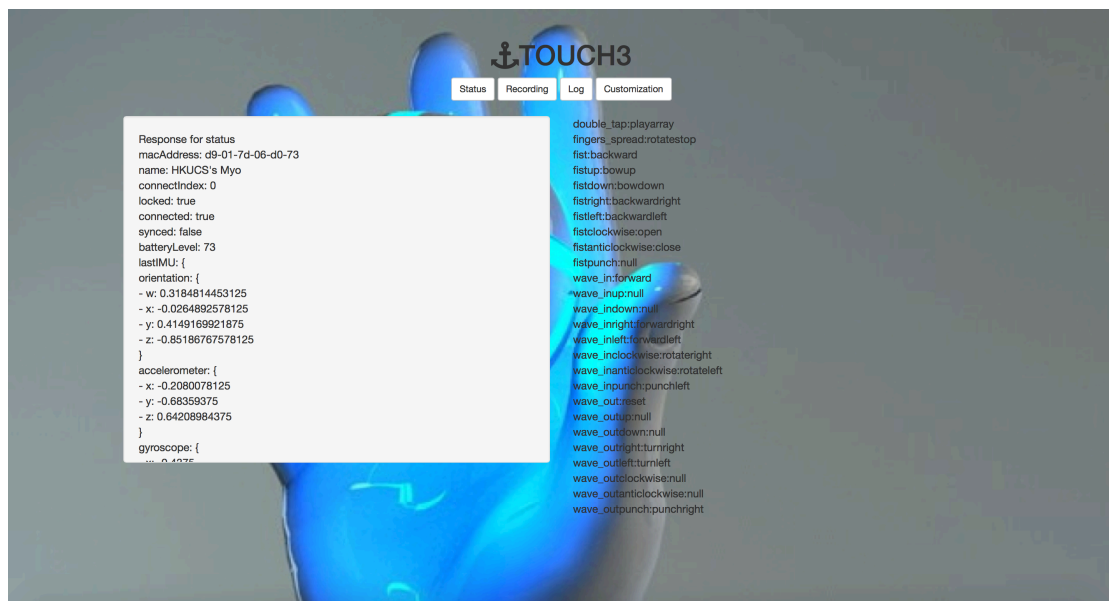


Figure 4: Status Tab UI Design

Figure 4 shows the UI design of Status tab. The left text box shows the information of connected MYO armband. There are two kinds of information. The first one is the hardware information, for example, the name of MYO armband, battery level and the connection information. Second information is the sensors data, for example, the data from orientation sensors, accelerometer, and gyroscope.

The right text box shows the gesture control setting. For example, if user makes a double tap gesture, TOUCH3 will send a play array command, and the robot will perform a sequence of motions recorded in Recording Tab.

Since it is a one-page website, the elements for Status tab load and be hidden before users pressing the Status button. Once the users press the button, the attributes of this two html elements will be changed and will become visible. Also, the event of clicking Status button is caught by client side, and client will send request to server side for getting the information of connected MYO armband and the default gesture control setting. Server receives the request and emits the required information to client side. Finally, the information will be displayed by client side.

7.1.2 Recording Tab UI design

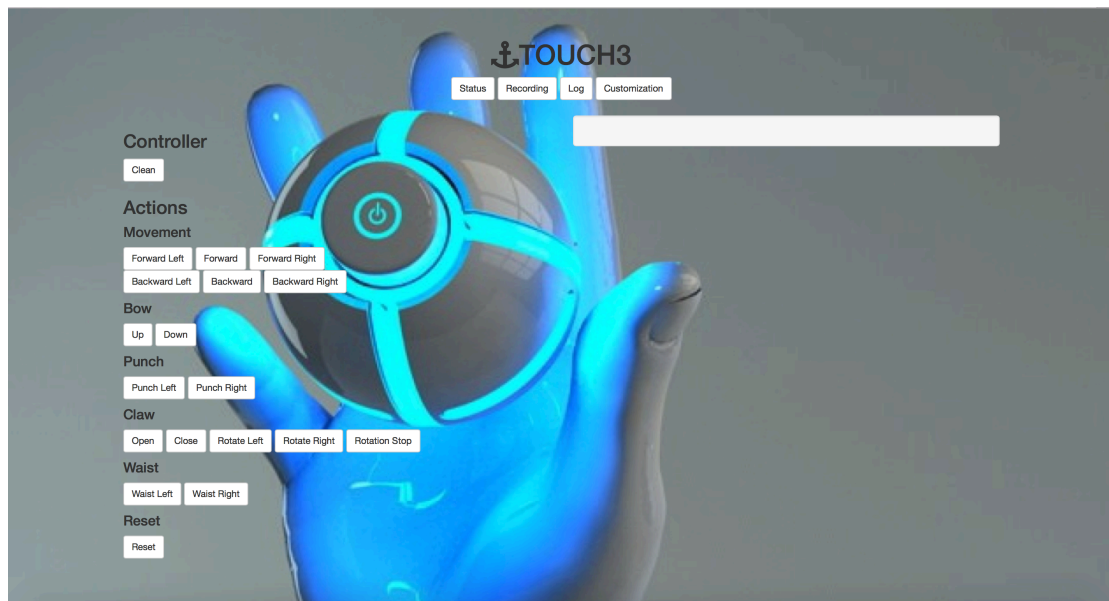


Figure 5: Recording Tab UI design

Figure 5 shows the UI design for Recording Tab. When users click the Recording button, this click event will be caught by client. And client will change the attributes of the html elements to be visible. After that, the website will display two area, control area and text box area.

In the left hand side, there is a control area. Users can clean the sequence of actions saved before by press Clean button. Once users press the button, client side will send a request to server for deleting the whole saved sequence of robot's motions. Also, users can add robot's motions into the sequence of action by pressing the action's buttons. Once users press the button, client side will send a request to server by button id for adding robot's motion. Users can also using MYO armband to input the robot's motion by corresponding hand gesture. Since server is listening for MYO armband input, once MYO armband data input, the server will add the robot's motion according to the gesture control setting.

In the right hand side, there is a text box, which shows the stored sequence of robot's motions. Once the server stores the input action from client, the client will display the new sequence in this text box.

7.1.3 Log Tab UI Design

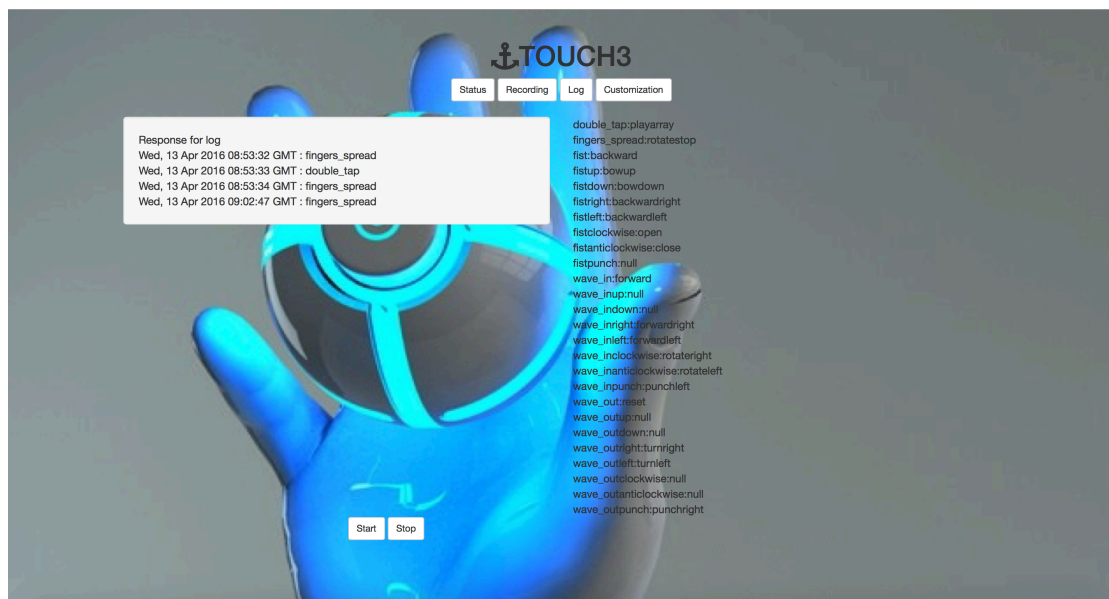


Figure 6: Log Tab UI Design

Figure 6 shows the UI design of the Log Tab. There are three major elements. In the left hand side, there is a text box. In the right hand side, it is show gesture control setting. The bottom part is control buttons. Once users click the Log button, this event will be caught by client and will send the request to the server for obtaining the necessary information. And then server will send the information back to the client, the client will display the information on the website.

The text box in left hand side, it is a logbook to record the gesture input from the connected MYO armband. Since server is listening for the connected MYO armband, once it receives gesture input, it will store in an array and will emit the array to client side. Finally, the client will display the array. The array includes the time receiving gesture and the gesture name in figure 6.

In the right hand side, it is showing the gesture control setting. Since the server stores an array of the gesture control setting, once the client requests the array, server will emit the array to client, and client will display it in this area. If users forget the setting, users can read it and manipulate the robot by MYO armband.

In the bottom part, there are two buttons, Start and Stop. They are used to start and stop the manipulation of robot. Once users press start, server will send command to robot if server receives gesture input from MYO armband. Once users press stop, server will not send any commands to robot, unless users press start to resume the manipulation.

7.1.4 Customization Tab UI Design

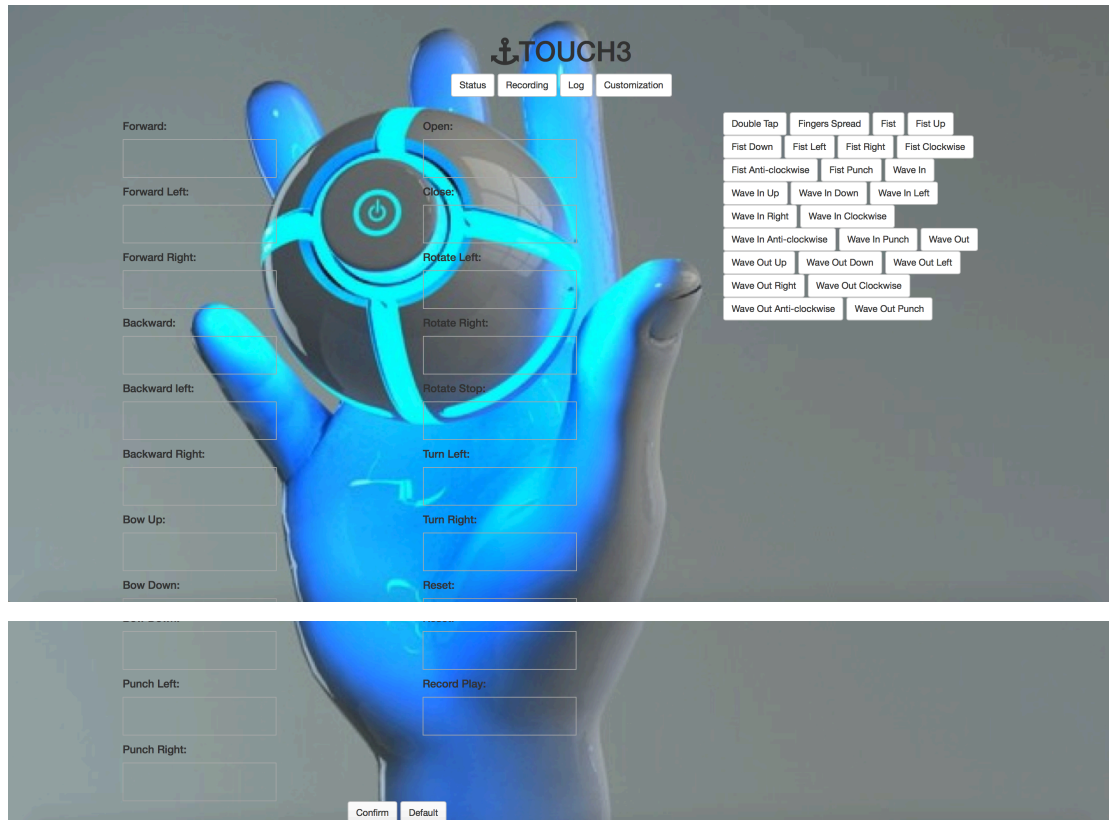


Figure 7: Customization Tab UI Design

Figure 7 shows the UI design of Customization Tab. There are two major elements, a drag and drop area and two control buttons. Once users click Customization button, this event will be caught by client, client changes the attributes of the html elements and makes them visible.

In the top part, it is a drag and drop area. In left hand side, there are 19 droppable areas for placing gesture buttons. They represent different robot's actions. In right hand side, there are 26 draggable gesture buttons. They represent 26 different gestures. Users are allowed to grab a gesture button, and

drop it in a droppable area, as shown as figure 8. Once users drop a button in a droppable area, client will store it in an array according their id.

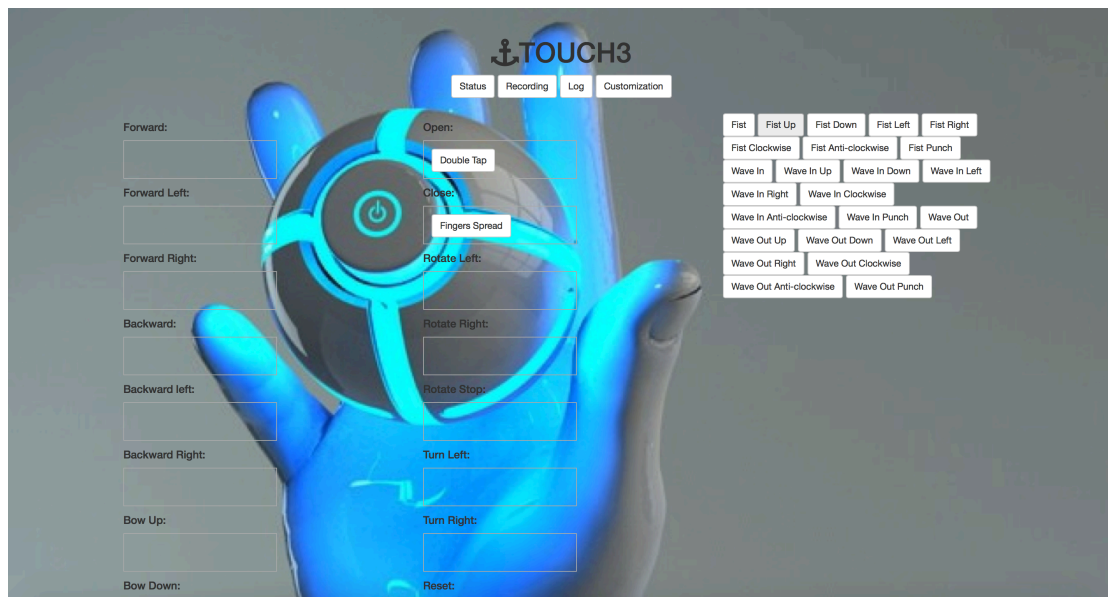


Figure 8: Drag and Drop Function for Customization of Gesture Control Setting

In the bottom part, there are Confirm and Default buttons. For Confirm button, once users press it, the client will send the array storing the drag and drop data to server. And then, server will receive it and will replace the current gesture control setting. For Default button, once users press it, the client will send a request to server. The server will reset the gesture control setting to the default gesture control setting.

7.2 Implementation of Five Functions

Based on the system design shown as figure 1, the four functions are implemented in the web application.

7.2.1 To Interconnect MYO Armband And Robots

TOUCH3 now is able to connect with MYO armband through BLE. It can accurately identify MYO armband even there are different devices with BLE, user can select a specific one MYO armband to connect with TOUCH3. TOUCH3 can receive the information and data input from MYO armband. Under the BLE environment, MYO armband can be connected within 10 meters. Although the

maximum theoretical range is about 100 meters, it may be affected due to the power output of the devices. But 10-meter-range is long enough for manipulating the robot and fulfill the requirement in this project. To increase the range, the power consumption of MYO armband and the connected device will be increased, and the manipulation time will be reduced.

7.2.1.1 To Connect With MYO Armband

In this project, the web application is implemented and developed by node.js. The producer of MYO armband, Thalmic Labs develop a MYO javascript binding, myo.js. It allows users to interact with MYO armband by web sockets. It is able to listen the data input from different sensors in MYO armband and gesture events, and control vibration and locking of MYO armband.

To connect the MYO Armband with TOUCH3, the MYO armband has to be paired, connected and synchronized. First, the MYO armband has to be paired with the computer. Since there may have more than one MYO armband, users have to select one MYO armband to pair up with the computer. The pairing task can be done in MYO armband manger application, which is provided by Thalmic Labs, and it helps connect the MYO with computer.

Second, after pairing with computer, a function `Myo.connect()` in `myo.js` can be called. `Myo.js` will create an instance to the paired MYO armband, and will make it being connected. And now MYO armband can send the data from different sensors to TOUCH3, but it cannot send the gesture event, because it is not synchronized.

Third, once the MYO armband is putted on user's forearm, the MYO armband will synchronize with the computer. And after that, it will send the gesture input to TOUCH3.

Once the above connection is done, TOUCH3 can get the data from connected MYO armband. The status information of MYO armband will be display on the left hand side of Status tab as shown as figure 4. The gesture input from MYO

armband will be sent to TOUCH3. TOUCH3 will listen for gesture input and will record them in an array. TOUCH3 will display the array on the left hand side under Log tab as shown as figure 6. Also, TOUCH3 will record the gesture input for the sequence of robot's motions under Recording tab as shown as figure 5.

7.2.1.2 To Connect With Robot (iDroid)

In this project, a robot developed by Globalactive Technology Limited, called iDroid, is used. iDroid is the first robot in the world controlled by smartphone since 2010. Since it has the Bluetooth radio, it is able to connect with different smart objects by BLE. Therefore, TOUCH3 can connect iDroid by BLE and send commands to iDroid.

iDroid is able to perform 6 types of robot's action, they are human-like movement, bow, twisting the waist, launching missiles, clip performance and punches. In the table 1, it is showing all the robot's motions, which can be performed by iDroid. In total, there are 20 robot's motions. However, the iDroid borrowed from Globalactive Technology Limited does not provide the missiles, so the iDroid cannot launch the missiles. Also, TOUCH3 provides an extra command for manipulating the iDroid to perform a sequence of motions. As a result, TOUCH3 have to send 19 commands to fully control the iDroid.

Table 1: iDroid Robot's Motions

	Robot's Motion				
	Forward	Forward Right	Forward Left		
Movement	Backward	Backward Right	Backward Left	Reset	
Bow	Bow Up	Bow Down			
Twisting	Twist Right	Twist Left			
Clip	Open	Close	Rotate Right	Rotate Left	Rotate Stop
Punches	Punch Right	Punch Left			
Missiles	Launch One	Launch Three			

Although an iDroid has a Bluetooth radio, the Bluetooth module in the borrowed iDroid is removed, so an audio cable is used to connect TOUCH3 with iDroid. Since the iDroid can be simply controlled through Bluetooth audio by playing different sound as command, TOUCH3 can play different sound files to manipulate the iDroid, and iDroid will react correspondingly. For example, TOUCH3 plays a sound file called “forward.wav”, the connected iDroid will move forward.

However, the iDroid is not able to provide any information about its status. It can only receive audio from other smart objects, but cannot communicate with other smart objects. Therefore, TOUCH3 cannot get any data from iDroid, and cannot display any information about iDroid.

7.2.2 To Match Hand Gestures To iDroid’s Motions

This function can be divided into three parts, detecting hand gestures, converting hand gestures into commands and sending command to iDroid.

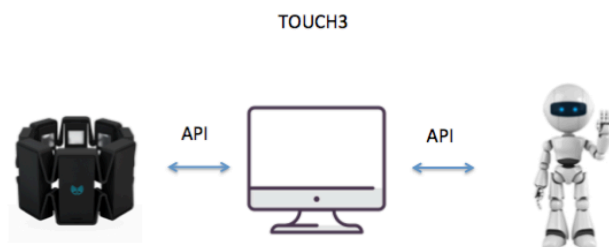


Figure 9: TOUCH3 Matching Hand Gestures To iDroid's Motion

In figure 9, it is showing the procedures of TOUCH3 matching hand gestures to iDroid’s Motion. In general, first, TOUCH3 connects with MYO armband and iDroid. Second, MYO armband detects user’s hand gestures and sends the gesture event to TOUCH3. Third, TOUCH3 matches the gesture event with iDroid’s motion according to the gesture control setting. Finally, TOUCH3 plays a sound file as a command, and iDroid reacts correspondingly.

7.2.2.1 To Detect User's Hand Gestures

MYO armband has five pre-set hand gestures. They are double tap, fingers spread, fist, wave in and wave out. Since iDroid has 19 robot's motions to perform, TOUCH3 is required to send 19 kinds of command, so MYO armband is required to detect at least 19 hand gestures. However, MYO armband only provides five pre-set hand gestures, more hand gestures have to be created and detected.

To create more hand gestures, there are two approaches. MYO armband is able to detect electrical signals by eight medical grade stainless electromyography (EMG) sensors. So MYO armband is able to recognize more hand gestures by EMG data. However, it is not feasible solution in this project. The reasons will be discussed in the section 9.1. Therefore, another approach is applied to solve this problem, combining the five pre-set hand gestures with arm motion.

Since MYO armband does not only contain eight EMG sensors, it also contains a group of sensors, they are gyroscopes, accelerometers and magnetometers, TOUCH3 is able to get the arm motion and arm position by calculating the input IMU data. To determine the arm motion and arm position, the angles of roll, pitch, yaw and the acceleration of three axes are required to be calculated. However, myo.js does not provide any functions to get the angles of roll, pitch and yaw, and the acceleration of three axes, as a result, four functions are required to develop for calculating those necessary data.

By using the function `myo.on('orientation', function(data){...})` in myo.js, TOUCH3 can get the orientation data in the form of quaternion. The data structure of orientation data is:

orientation : {
 x: NUM,
 y: NUM,
 z: NUM,
 w: NUM
 }

To obtain the angles of roll, pitch and yaw, TOUCH3 have to convert the quaternions into the Euler angles [5].

Equation 1: Obtaining The Euler Angles From Quaternions

$$\begin{aligned} \phi &= \tan^{-1} \left(\frac{2(xy + zw)}{1 - 2(y^2 - z^2)} \right) \\ \theta &= \sin^{-1} (2(xz - wy)) \\ \psi &= \tan^{-1} \left(\frac{2(xw + yz)}{1 - 2(z^2 + w^2)} \right) \end{aligned}$$

Equation 1 is the equation of calculating the Euler angles from quaternions, ϕ is the angle of roll, θ is the angle of pitch and ψ is the angle of yaw. However, the function of arctan and arcsin implemented in computer program only gives results between $-\pi/2$ and $\pi/2$. So they do not cover all possible orientations, a function call atan2 is used to replace arctan function to get all the orientations as shown as equation 2.

Equation 2: Obtaining The Euler Angles From Quaternions By Using atan2 Function

$$\begin{aligned} \phi &= \text{atan2}(2(xy + zw), 1 - 2(y^2 + z^2)) \\ \theta &= \sin^{-1} (2(xz - wy)) \\ \psi &= \text{atan2}(2(xw + yz), 1 - 2(z^2 + w^2)) \end{aligned}$$

Based on the equation 2, three functions are implemented in TOUCH3, so TOUCH3 is able to calculate the angles of roll, pitch and yaw, they are getRoll(), getPitch() and getYaw(). By comparing the initial angles of roll, pitch and yaw with the current angles, TOUCH3 can determine the arm motion, rolling clockwise or anticlockwise, and the arm position, raising up or putting and placing right or left.

To obtain the acceleration data of three axes, TOUCH3 have to obtain the acceleration data from MYO armband by myo.js. By using a function,

myo.on('imu', function(data){...}), TOUCH3 can receive the acceleration data. The data structure of acceleration is:

```
acceleration : {
    x: NUM,
    y: NUM,
    z: NUM
}
```

By using the acceleration data, TOUCH3 can implement a function to determine the punch motion. To simplify the manipulation of iDroid, TOUCH3 is only able to detect the punch motion across the x axis, namely punch forward. To detect the punch motion, TOUCH3 have to calculate the acceleration and the punch time. Once the acceleration and the punch time reach the threshold, TOUCH3 will detect a punch motion.

As a result, TOUCH3 is able to determine the arm motion and the arm position, so more gestures is created by combining the five pre-set hand gestures with the arm motions. And now, 21 self-defined gestures are created, and there are 26 gestures in total with the five pre-set hand gestures as shown as table 2.

Table 2: Detected Gestures In TOUCH3

Pre-set Hand Gestures	Self-Defined Hand Gestures						
Double Tap							
Fingers Spread							
Fist	Fist Up	Fist Down	Fist Right	Fist Left	Fist Clockwise	Fist Anticlockwise	Fist Punch
Wave In	Wave In Up	Wave In Down	Wave In Right	Wave In Left	Wave In Clockwise	Wave In Anticlockwise	Wave In Punch
Wave Out	Wave Out Up	Wave Out Down	Wave Out Right	Wave Out Left	Wave Out Clockwise	Wave Out Anticlockwise	Wave Out Punch

To recognize different arm motions, TOUCH3 is using a binary tree structure to classify the data. In each level, only one arm motion or arm position will be classified and outputted.

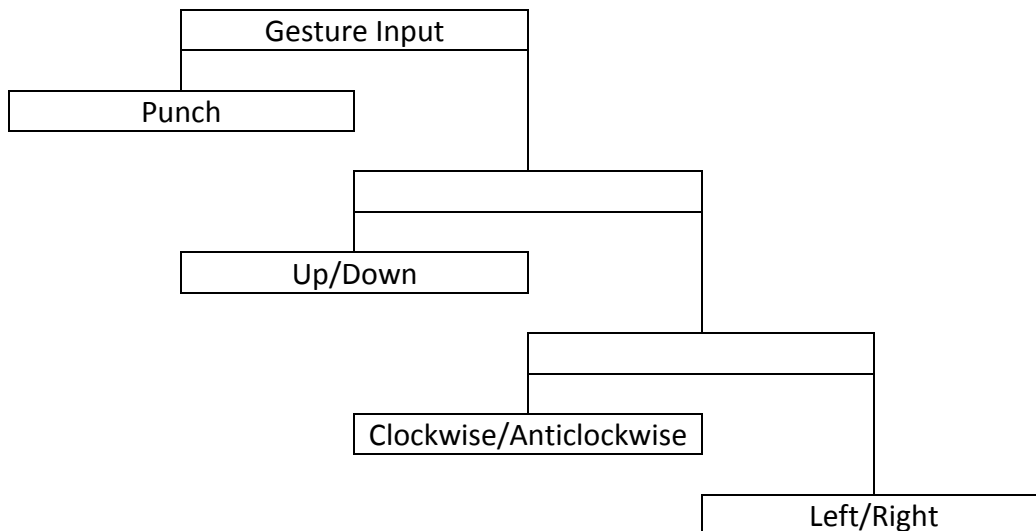


Figure 10: Binary Tree Structure Of Gesture Classifier

Figure 10 is showing the binary tree structure of gesture classifier. By using this two-stage tree classifier, TOUCH3 is able to determine the type of arm motion or arm position.

Also, using the function `myo.on('pose_off', function(pose_name){...})`, TOUCH3 can listen to the gesture event. This function is providing a gesture data, when MYO armband catch the end of a pose, and it will fire a gesture event to TOUCH3. And then, TOUCH3 combines the result from arm motion and arm position classifier and the gesture event, TOUCH3 can determine a gesture input from users.

7.2.2.2 To Match Hand Gestures to iDroid's Motions

TOUCH3 contains two array, one is used to store the path of sound files, another one is used to store the matching of gestures and iDroid's motions. Once TOUCH3 recognize a gesture input, it will search the matching array to determine the corresponding iDroid's motion. And then, TOUCH3 will search the array storing the path of sound files to get the path.

7.2.2.3 To Play Sound Files As Command

After getting the path of sound file, TOUCH3 makes use of a node module, play-sound.js, to play the audio. Since the iDroid is connect to TOUCH3 by an audio cable, the audio signal will send to the iDroid as a command, and finally, iDroid will react correspondingly.

7.2.3 To Customize User Gestures

This function is used for users to customize the gesture control setting. Since TOUCH3 defined 26 hand gestures, users can customize the gestures matching to certain command by themselves. For example, the default gesture to make the iDroid moving forward is waving the palm inside, but user is allow to change the setting, and using waving outside orders the iDroid moving forward.

TOUCH3 has a tab, called Customization, as shown as figure 7. It allows users to change the setting by drag and drop. After customization as shown as figure 8, users have to press Confirm button to establish the setting. Once the client side catches this event, it will store the matching in an array according to id, and send the array to server side. And then, server will replace the current matching array by this new setting. Furthermore, users are able to restore the default setting by pressing Default button. Once the client side catches this event, it will send the request to server, and server will replace the match array by the default matching array.

7.2.4 To Present The Information To The Users In A Proper Manner

TOUCH3 now is able to present the information of MYO armband, for example, the battery and the synchronization status of MYO armband as shown as figure 4. The modified UI design is implemented into TOUCH3 as shown as figure 3, so the layout of TOUCH3 is concise and user-friendly.

Apart from presenting the information of MYO armband, TOUCH3 should be able to display the information of connected robots. However, the iDroid borrowed is

not able to output any data or information, TOUCH3 cannot get any information of iDroid and cannot present the status to users.

In the proposed idea, TOUCH3 is required to provide a haptic feedback to users, when the robot receives a command. However, this function does not implement into TOUCH3, although a function called `vibrate()` in `myo.js` is able to provide haptic feedback to users, it is too annoying, if the MYO armband vibrate everytime, when the robot receives commands. Therefore, this idea does not apply into this project. And TOUCH3 does not provide any haptic feedback to users.

7.2.5 To Perform A Sequence of iDroid's Motions By Using A Simple Hand Gesture

This function is used to simplify the robot's manipulation. As mentioned in the section 3, TOUCH3 would like to simplify the manipulation, it allows users to use one single hand gesture to control the iDroid performing a sequence of motions, for example, dancing.

To perform a sequence of motions, iDroid has to perform the motions one by one. Therefore, TOUCH3 is required sending the commands one by one and orders the iDroid to react. To achieve this task, TOUCH3 is required to store a list of motions in order and plays the corresponding sound files.

TOUCH3 has a tab, called Recording. It allows users to store or clean the sequence of motions. Users can either pressing the button on the left hand side as shown as figure 5, or using hand gestures to input the motions into the sequence. Once users press the buttons or input hand gestures, client side will send the data to server, and server will store them into an array. Also, users are able to clean the array by pressing Clean button. Once the client side catches this event, it will send the request to server, and server will delete the entire array.

After storing the array, users are able to use a simple hand gesture for ordering iDroid and iDroid will perform the corresponding reactions. The default hand gesture to perform the sequence of motion is double tap. After double tap the

finger, TOUCH3 will get the sequence from the array, and search the array storing the path of sound file. And then, TOUCH3 play a list of sound file, so the iDroid will react according to the playlist.

7.3 Summary

Table 3: Progress of Five Functions

Function	Subtask	Status	
1. To Interconnect MYO Armband And iDroid	MYO Armband	Completed	
	iDroid		
2. To Match Hand Gestures To iDroid's Motions	To Detecting Hand Gestures		
	To Match Hand Gestures to iDroid's Motions		
	To Play Sound File As Command		
3. To Customize User Gestures	NA		
4. To Present The Information To The Users In A Proper Manner	UI Design		
	Haptic Feedback		Cancelled
5. To Preform A Sequence Of iDroid's Motions By Using A Simple Hand Gesture	NA		Completed

To summary, table 3 shows the progress of the five functions. Most of them are completed. Only the function that provides haptic feedback is cancelled, because it is very annoying, if the MYO armband virbrate everytime, when the iDroid receives commands. In general, all functions are completed and meet the requirements.

8. Results

To ensure TOUCH3 meeting the requirement and enhance the quality, two user experience assessments and two tests are conducted to collect the useful opinion and data.

8.1 User Experience Assessment

There are two user experience assessments conducted. The first assessment is conducted after phase one, and the second assessment is conducted in phase three.

8.1.1 First User Experience Assessment For UI Design Prototype

After phase one, a user experience assessment is conducted to collect users' opinion on the prototype of TOUCH3 UI design, as shown as figure 2. 6 engineering students participated in this assessment. Before doing the assessment, they have to read the project proposal and the project website. To make sure their understanding to this project, they are allowed to ask any questions about this project. And then, they did the assessment. Three questions are asked to collect their opinion about visual design, functionality and improvement.

Most of them mentioned about the style of UI is too concise and unattractive, since the application lacks in animation and does not have any icon. In term of functionality, although it is plain, it provides the necessary functions. They suggested that the style of project website is concise and fashionable, the style should be applied into the application.

8.1.2 Second User Experience Assessment For Final Product Of TOUCH3

After finishing the UI in phase three, 7 students are invited to participate in an assessment. To enhance their understanding of this project, they are required to read the project proposal and project website before the assessment. Also, they are allowed to ask any questions about this project.

In this assessment, participants have to answer 6 questions and give score, minimum 1 to maximum 5. 1 score means very disagreed, and 5 score means very agreed.

Question 1: Do you agree that the UI is concise and fashionable?

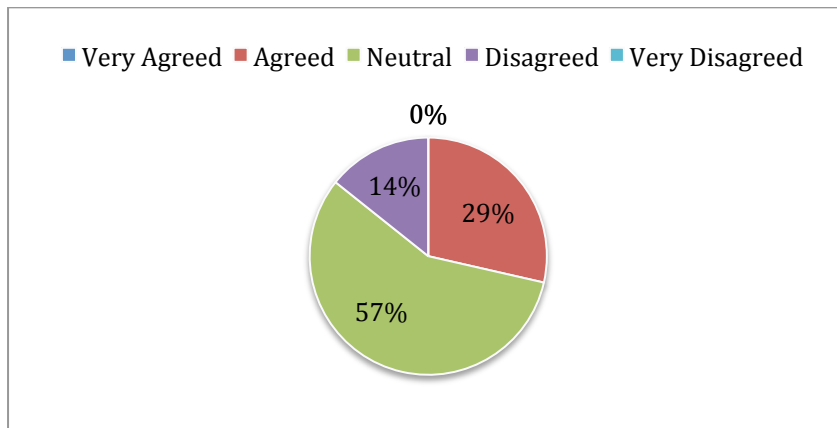


Figure 11: Pie Chart Of Agreeing that UI Design Is Concise and Fashionable

Question 2: Do you understand the function of each button?

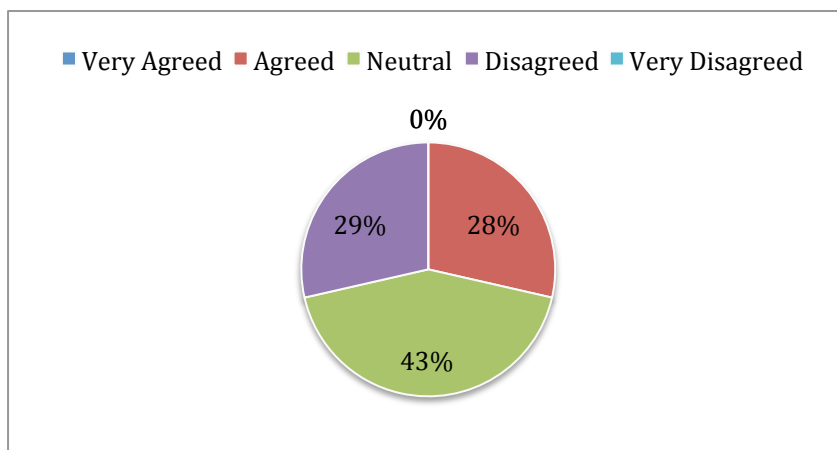


Figure 12: Pie Chart Showing The User's Understanding To TOUCH3 Buttons

Question 3: Do you agree that the Status tab showing the necessary information of MYO armband?

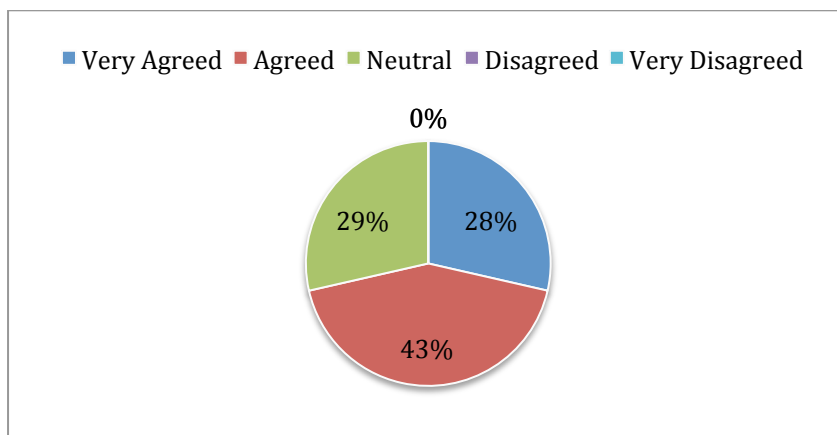


Figure 13: Pie Chart Showing Status Tab Provides Necessary Information Of MYO Armband

Question 4: Do you agree that the Recording function under Recording Tab is user-friendly?

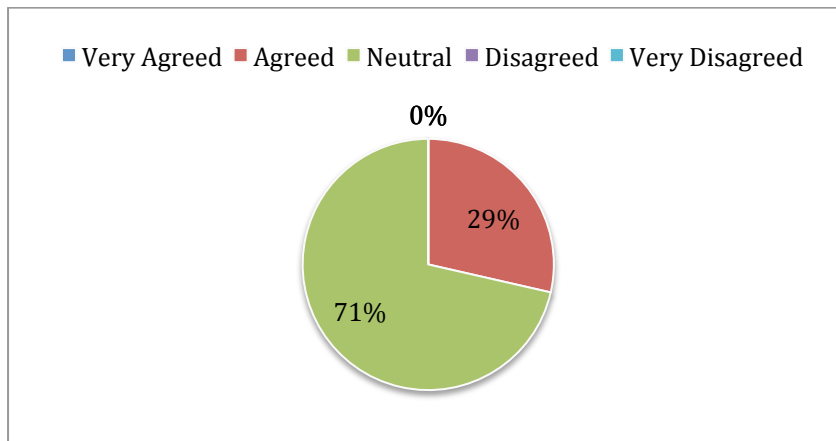


Figure 14: Pie Chart Showing Recording Function Is User-friendly

Question 5: Do you agree that the Log tab provides the necessary information?

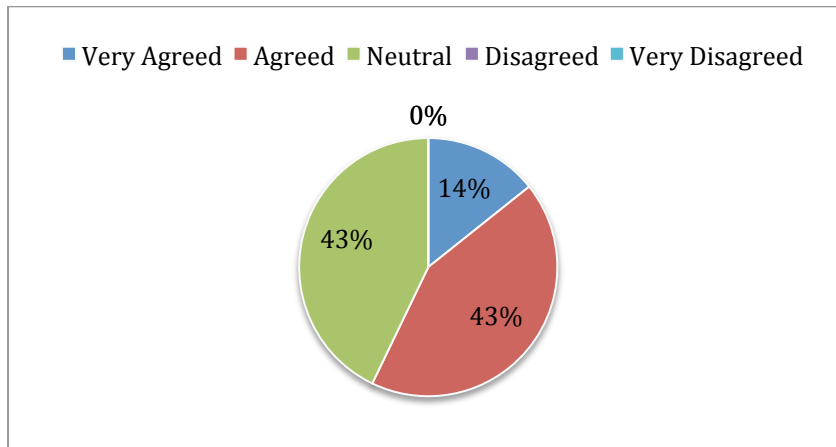


Figure 15: Pie Chart Showing Log Tab Provides Necessary Information

Question 6: Do you agree that using drag and drop to customize the setting is user-friendly?

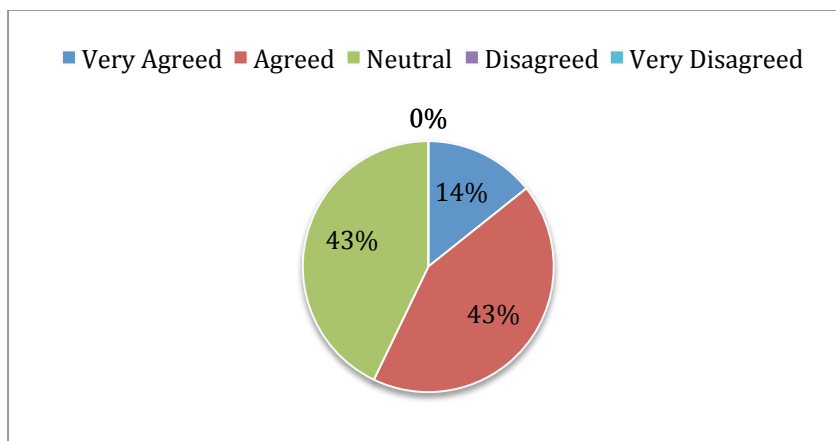


Figure 16: Pie Chart Showing Drag and Drop Function Is User-friendly

More than a half of participants are neutral to the concise and fashionable UI design as shown as figure 11. 71% of them understand the usage of each buttons in TOUCH3 as shown as figure 12. 43% of them agree the information provided under Status tab is enough as shown as figure 13. More than a half of participants feel neutrally to the recording function under Recording tab as shown as figure 14. 43% of them agree that the Log tab provides necessary information of gesture detection as shown as figure 15. 43% of them agree that the matching function under Customization tab is user-friendly as shown as figure 16. In general, participants are satisfied with the UI design of TOUCH3 and agree that the instructions provided are enough for users manipulating TOUCH3.

Comparing to the first assessment, the satisfaction of TOUCH3's UI design is increased. And it shows that TOUCH3 is user-friendly and easy to use.

8.2 Accuracy of Detecting Hand Gestures

In this project, there are two tests conducted to collect the data about the accuracy of detecting hand gestures by TOUCH3. The first one is conducted in phase two. And the second one is conducted in phase three.

8.2.1 First Accuracy Test For TOUCH3 Detecting Hand Gestures

In the end of phase two, an accuracy test is conducted. Two students participated in this test. First, participants have to read the instructions of gesturing. Second, participants have five minutes to do a trial by themselves. Finally, participants do the test.

In this test, participants are required to input 52 hand gestures in order, 26 types of hand gestures detected by TOUCH3 are included in the test. And then, the results will be compared with the order. If TOUCH3 can detected the similar gesture to the order, the input gesture will be determined as correct, otherwise, will be determined as incorrect.

The average accuracy is 71.2%. The accuracy of the first participant is 82.7%, and that of the second participant is 59.6%.

The accuracy is not very high, but the result is better than the experiment conducted by Cecille Freeman from University of Waterloo in 2014. Freeman gets 35.0% accuracy by using a one-stage tree structure to classify 8 types of hand gestures using MYO armband[6].

However, in this test, the difference of accuracy between two participants is large, 23.1%. Since the first participant is the developer of TOUCH3, he has a large experience of inputting different gestures, he can perform the gestures in a correct way, so the accuracy of him is higher than that of the second participant who does not have any MYO armband and TOUCH3 experience. But the accuracy should be able to increase.

8.2.2 Second Accuracy Test For TOUCH3 Detecting Hand Gestures

In the end of the phase three, another accuracy test is conducted. In this test, three students participated after reading the instructions of gesturing. And then, they have five minutes to have a trial. Finally, the test is start.

In this test, participants are required to input 26 types of hand gestures in order. The result will be compared with the order, if the detected gesture is similar to the order, it will be determined as correct, otherwise, it will be determined as incorrect. After the test, participants are required to do a calibration of hand gestures provided by MYO manager. It trains the MYO armband to detect the hand gestures. And then, participants are required to input 26 types of hand gestures in order one more time.

Table 4: The Result Of Second Accuracy Test

Participant	Trial	
	1st	2nd
1	92.3%	92.3%
2	65.4%	76.9%
3	61.5%	73.1%
average	73.1%	80.8%

In table 4, it shows the result of the second accuracy test. In the first trial, the average accuracy is 73.1%. Since participant 1 is a MYO armband experienced user and participant 2 and 3 is not, the accuracy of participant 2 and 3 is lower than that of participant 1. However, the individual accuracy and average accuracy is higher than the first accuracy test. Moreover, the individual accuracy and average in second trial is higher than first trial. The reason for which the accuracy is increased will be discussed in the section 9.2.

All in all, the improvement of accuracy is significant, increases 9.6%, and it proves that TOUCH3 is able to manipulate iDroid by hand gesture effectively.

9. Problems Encountered

In this project, there are four major problems. They are defining new hand gesture by using EMG data, low accuracy of detecting hand gestures by TOUCH3, poor performance on Android environment and incomplete myo.js library.

9.1 Self-defined Hand Gestures

In this project, there is a big challenge. The current MYO armband applications are mainly using five pre-set hand gestures to control different devices. But this project aims to develop a web application, which is able to fully and smoothly control the robot. In general, it requires 19 hand gestures to order iDroid performing different motions, but MYO armband can only recognize the five pre-set hand gestures. Therefore, TOUCH3 needs more hand gestures, and self-defined hand gestures are required. There are two solutions to solve this problem. They are capturing the EMG data of self-defined hand gestures and combining the five pre-set hand gestures with arm motion and arm position.

The first solution, capturing the EMG data of self-defined hand gestures is not a good solution. To recognize more gestures, the data of electrical signals from muscles in forearm have to be collected and analyzed. It requires to collect a huge amount of EMG data from different people. However, there are many factors affecting the EMG data. Since the MYO armband is skin mounted, the

electrical signal is not directly detected, but through a layer of skin and fat, so the signal may be affected. Also, the MYO armband is spatially removed from the muscles, the sensors cannot pick up the exact muscle that generate useful EMG data, they have to pick the signals from a group of muscles at the same time, so they cannot guarantee to be placed correctly and receive the useful signals. Since different people have different amount of fat and hair, the EMG data from different people have a slightly difference, this slightly difference affects the accuracy and the efficiency of recognition of self-defined hand gestures.

According to a research about using MYO armband for medical application conducted by Mahmoud Abduo and Matthias Galster from University of Canterbury in 2015, they would like to investigate that the accuracy of EMG data from MYO armband is sufficient to assist the hand amputees. They are trying to collect a number of EMG data from different people performing self-defined hand gestures. And then, they use those data to classify the gestures. However, the result is not satisfied [7].

Since the cost of collecting large amount of EMG data and low accuracy of self-defined hand gestures from EMG data, using EMG data to define hand gestures is not a good approach to create more hand gestures.

The second solution, combining the five pre-set hand gestures with arm motion is better than the first one. Since MYO armband also includes gyroscope, accelerometer and magnetometer, TOUCH3 can make use of their input data to detect the arm motion. By calculating the data input from three types of sensors, TOUCH3 is able to get the roll angle, yaw angle and the acceleration. This data is used to determine the arm motion and arm position. Therefore, TOUCH3 defined 21 self-defined hand gestures and can detect totally 26 hand gestures.

As a result, the limited amount of hand gestures can be solved by combining the five pre-set hand gesture with arm motion and arm position. And TOUCH3 applied this method to increase the number of hand gesture. With the MYO armband firmware updating from Thalmic Lab, more and more hand gestures

will be added in the future, so users can make use of them to have a better experience of robot manipulation.

9.2 Accuracy Of Detecting Hand Gestures

In the first accuracy test, the average accuracy is 71.2%, it is not satisfied, so another classify algorithm should be used to enhance the accuracy. In the first test, TOUCH3 used a single-stage classifier, which means this classifier will determine the arm motion or arm position from all possible arm motion or position. Therefore, the accuracy is low. According a research conducted by Cecille Freeman in 2014, the accuracy of using single-stage classifier is 35%. And the accuracy of using two-stage classifier, which is similar to figure 10, is 71.0%. So TOUCH3 applies his theory, and a binary tree structure of gesture classifier is designed. In each level, there are two inputs, one is an arm motion or arm position, one is entering the second level. By using this method, the classifier can eliminate the incorrect arm motion and arm position. TOUCH3 now is using this binary tree to classify the input. Therefore, in second accuracy test, the result is increased significantly to 80.8%.

Also, users can do the gesture calibration before running TOUCH3. Since the calibration provided by MYO manger can train the MYO armband to detect the hand gestures from specific user, the accuracy of MYO armband detecting that user can be increased. In second test, the result shows that calibration can increase the accuracy from 73.1% to 80.8%.

In phase two, TOUCH3 uses a function `myo.on('pose_on')` to get the gesture event. However, it increases the difficulty of inputting self-defined hand gestures, because it emits the gesture event when the users start do the gesture. At that moment, users may not finish the arm motion or place the arm to a correct position, so users will input a wrong self-defined hand gesture if using `myo.on('pose_on')`. Instead of it, TOUCH3 in phase three uses `myo.on('pose_off')`. It emits the gesture event, when users finish doing the gestures. Using `myo.on('pose_off')` makes the manipulation being easy, and increases the correctness of gesture input.

Due to these three results, the accuracy of TOUCH3 now is 80.8%, it is efficient to manipulate iDroid.

9.3 Running Environment

Apart from the low accuracy of detecting hand gestures, the running environment is also a problem in this project. The original designed platform is Android environment, however, the accuracy and the efficiency of detecting hand gestures are unsatisfactory. Therefore, a web application, TOUCH3 is developed, instead of a mobile application. And there are two reasons for which changing running environment.

First, the support of developing an Android mobile application from Thalmic Lab is insufficient. Although the mobile application can connect with MYO armband easily, the connection of MYO armband is without any synchronization and gesture calibration function. Therefore, the recognition of hand gestures and the performance are poor, it is not sufficient to control iDroid. However, the support in PC version is better. Users have to synchronize their MYO armband before they using it. Also, the MYO armband will keep track on the muscle signal and learn from it, so after a time for warming up, the recognition of hand gestures is significantly increased. Moreover, users can calibrate the gesture and have their own calibration profile. As shown as table 4, the accuracy with calibration is much higher than without calibration. As a result, the overall recognition performance in PC is better than in smartphone.

Second, Node.js has many advantages on developing a web application. Since TOUCH3 is a real-time application, the concurrency problem and synchronization problem have to be avoided. Node.js is using event-driven architecture, it is good at handling these problems. Also, Node.js have many open source libraries. These libraries are very useful in this project. For example, TOUCH3 is using play-sound.js to play the sound file as iDroid's command. As a result, using Node.js to develop TOUCH3 is efficient and effective.

9.4 Incomplete myo.js Library

TOUCH3 is using myo.js to connect and receive data from MYO armband. However, it does not provide the every function comparing MYO library in other programming language. As a result, to implement different functions in TOUCH3, more functions are required to develop for getting a data from MYO armband. For example, the getRoll, getPitch and getYaw functions are provided in c++, so these functions have to be developed. Although there are many open source library in Node.js, the documentation of these libraries are not clear enough, more time is spent on reading and understanding the coding.

10. Conclusion

MYO armband is a high potential smart device. Making use of it, users can achieve a touch free experience and apply in different dangerous and inaccessible situation. This project is trying to combine MYO armband with iDroid, the main vision is to provide an unexpected user experience in robot control.

The development of TOUCH3 is finished, and the progress is satisfied. The table 5 shows the current status and the working schedule. All of the tasks were completed on time, although there are two major difficulties in this project. Since the MYO armband can only detect five pre-set hand gestures, the number of hand gesture is not enough for the requirement, fully manipulating iDroid. The solution is to combine the pre-set hand gestures with arm motion, so the number of hand gesture can be increased, and provide more gestures for users to manipulate the iDroid. Moreover, the accuracy of detecting hand gestures by TOUCH3 have to be increased. To increase the accuracy, a binary tree structure classifier is implemented and used to classifier the arm motion and arm position. Also, using the gestures calibration function provided by MYO manager can increase the accuracy. So the accuracy is increased from 71.2% to 80.8%, it can provide a better manipulation experience to users.

All in all, the objective of this project, to remotely control robot by MYO armband, is successfully achieved. In the end of phase three, a beta version of TOUCH3 is released. Hopefully, after collecting more users' opinion, the final product of TOUCH3 will be powerful and may contribute to the future robot development.

Table 5: Schedule of TOUCH3 Development

Tasks		Due Date	Status
Inception	Project Plan	10/4/15	Done
	Project Website		
System Design	Review Preliminary	10/11/15	
	Develop UI Prototype	10/18/2015	
	Review UI Prototype	10/21/2015	
	Review System Architecture	10/25/2015	
Elaboration	Implement UI	11/4/15	
	Develop Connection of MYO armband Functionality	12/31/2015	
	Combine pre-set hand gestures with arm motion	1/4/16	
	Release Demo application	1/9/16	
	Interim Report	1/20/2016	
	Website Maintaining	1/20/2016	
Construction	Develop Connection of Robot Functionality	1/31/2016	
	Develop Remaining Functionalities	2/29/2016	
	Debugging	3/31/2016	
Testing	Release Bate-version application	4/4/16	
	Testing	4/10/16	
Documentation	Project Documents	4/13/2016	
	Final Report	4/15/2016	
Deployment	Final Checking	4/17/2016	
	Release final application	4/17/2016	

10.1 Future Work

In future, TOUCH3 can have many improvements. There are two goals should be completed in future. First, the accuracy should be increased to 90%. A possible way to achieve this goal is using two MYO armbands, so TOUCH3 can receive 2 input data, and compare them to output a correct hand gesture. In many researches, they used two MYO armbands to recognize the self-defined hand

gestures, and the accuracy is higher than using one MYO armband. So it is a possible way to increase the accuracy. Second, another robot should be used in this project. Since the iDroid can only receive audio as command, it cannot input or send any data to TOUCH3, so if using other robot that can truly communicate with TOUCH3, TOUCH3 can well-demonstrate its functions. Third, TOUCH3 can make use of `myo.on('pose_on')` and `myo.on('pose_off')` to get two IMU data. By comparing two data, TOUCH3 can recognize more arm motions, for example, moving from right to left and moving from up to down. So more self-defined hand gestures can be created.

11. List Of Figures

FIGURE 1: TOUCH3 USE-CASE DIAGRAM.....	11
FIGURE 3: MODIFIED TOUCH3 UI DESIGN.....	13
FIGURE 4: STATUS TAB UI DESIGN.....	14
FIGURE 5: RECORDING TAB UI DESIGN.....	15
FIGURE 6: LOG TAB UI DESIGN.....	16
FIGURE 7: CUSTOMIZATION TAB UI DESIGN.....	17
FIGURE 8: DRAG AND DROP FUNCTION FOR CUSTOMIZATION OF GESTURE CONTROL SETTING.....	18
FIGURE 9: TOUCH3 MATCHING HAND GESTURES TO IDROID'S MOTION.....	21
FIGURE 10: BINARY TREE STRUCTURE OF GESTURE CLASSIFIER.....	25
FIGURE 11: PIE CHART OF AGREEING THAT UI DESIGN IS CONCISE AND FASHIONABLE.....	30
FIGURE 12: PIE CHART SHOWING THE USER'S UNDERSTANDING TO TOUCH3 BUTTONS.....	30
FIGURE 13: PIE CHART SHOWING STATUS TAB PROVIDES NECESSARY INFORMATION OF MYO ARMBAND.....	30
FIGURE 14: PIE CHART SHOWING RECORDING FUNCTION IS USER-FRIENDLY.....	31
FIGURE 15: PIE CHART SHOWING LOG TAB PROVIDES NECESSARY INFORMATION.....	31
FIGURE 16: PIE CHART SHOWING DRAG AND DROP FUNCTION IS USER-FRIENDLY.....	31

12. List Of Tables

TABLE 1: IDROID ROBOT'S MOTIONS.....	20
TABLE 2: DETECTED GESTURES IN TOUCH3.....	24
TABLE 3: PROGRESS OF FIVE FUNCTIONS.....	28
TABLE 4: THE RESULT OF SECOND ACCURACY TEST.....	33
TABLE 5: SCHEDULE OF TOUCH3 DEVELOPMENT.....	39

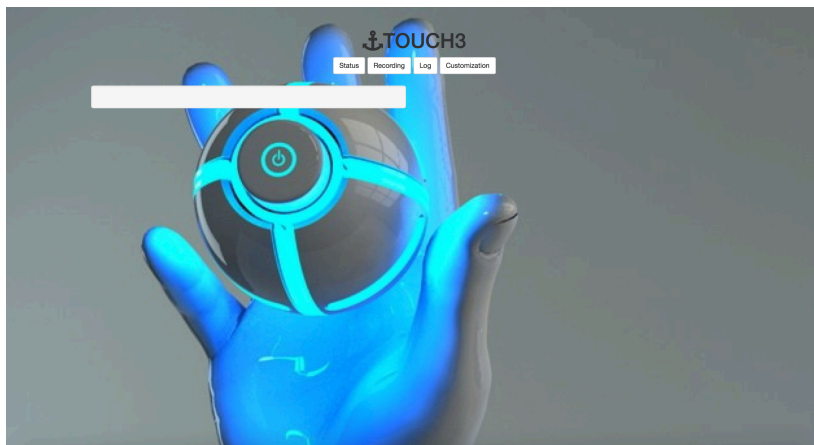
13. List Of Equations

EQUATION 1: OBTAINING THE EULER ANGLES FROM QUATERNIONS.....	23
EQUATION 2: OBTAINING THE EULER ANGLES FROM QUATERNIONS BY USING ATAN2 FUNCTION.....	23

14. TOUCH3 User Guide

14.1 Launch TOUCH3

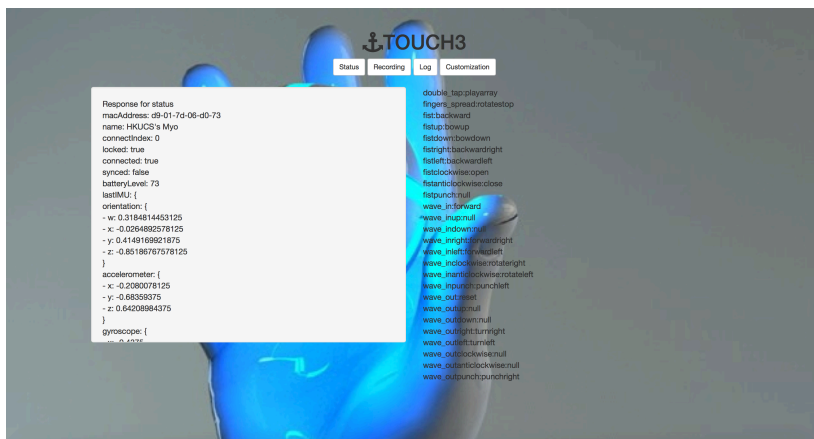
1. Connect MYO armband with computer
2. Wear MYO armband to a suitable position of forearm
3. Connect iDroid by using audio cable
4. Increase the volume to maximum to ensure iDroid can receive the sound
5. Place you arm facing forward as a initial position
6. Open TOUCH3 file in terminal by using command “cd TOUCH3”
7. Launch TOUCH3 in terminal by using command “node server.js”
8. Using web browser and visit “localhost:3000”



9. Launch successfully if appear the above picture

14.2 Status Tab

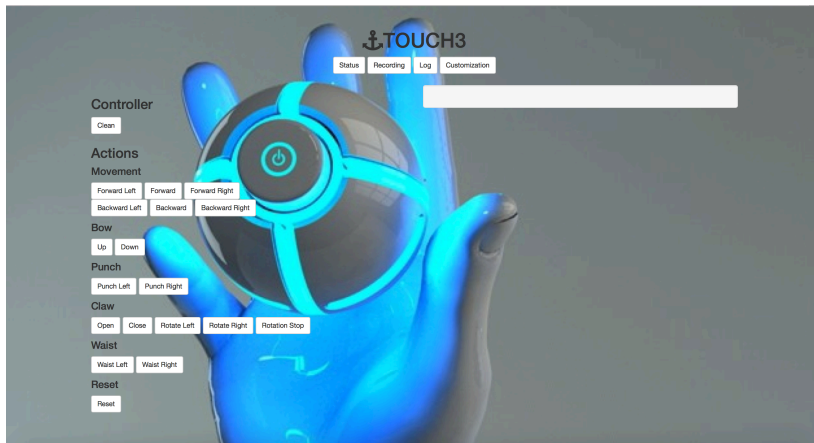
1. Press Status button to enter Status tab



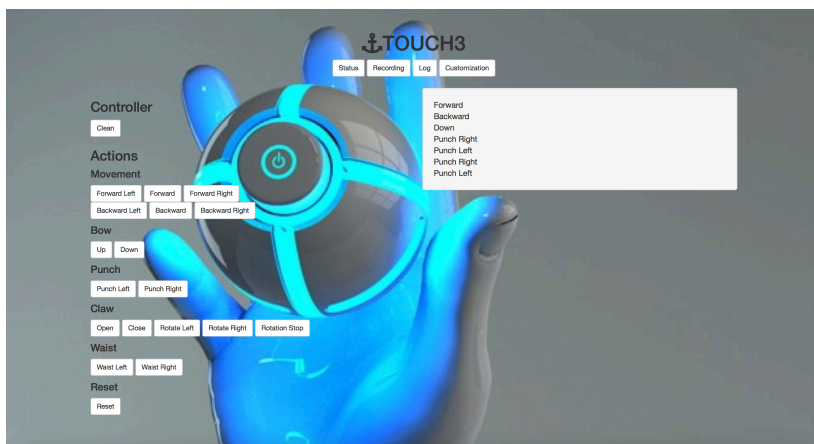
2. Enter Status tab successfully if appear the above picture
3. The left text box is the information of MYO armband
4. The right text area is the gesture control setting

14.3 Recording Tab

1. Press Recording button to enter Recording tab



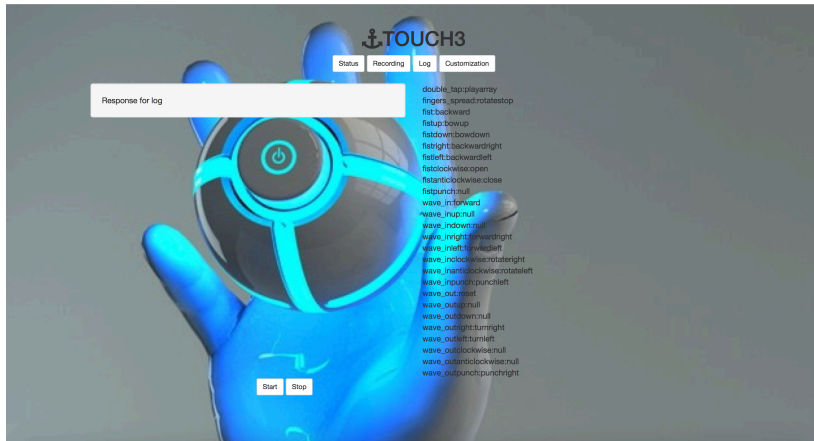
2. Enter Recording tab successfully if appear the above picture
3. The left area with buttons is operation control, the Clean button can clear the sequence of actions. The Action buttons can add different iDroid's motions into the sequence of actions. Also, users can use MYO armband to add different iDroid's motions according to the gesture control setting.



4. Adding iDroid's motions successfully if appear the above picture

14.4 Log Tab

1. Press Log button to enter Log tab



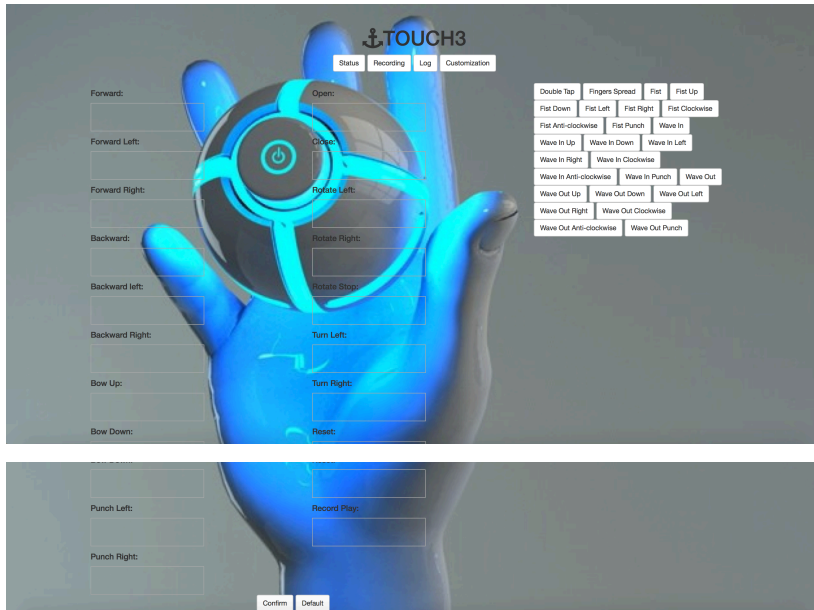
2. Enter Log tab successfully if appear the above picture
3. The left text is displaying the input of recognized hand gestures
4. The right text area is the gesture control setting
5. The button on the bottom is operation control button, press Start to start manipulating iDroid, after starting manipulation, the input hand gestures convert into command and send to iDroid, iDroid react correspondingly, press Stop to stop manipulating iDroid.



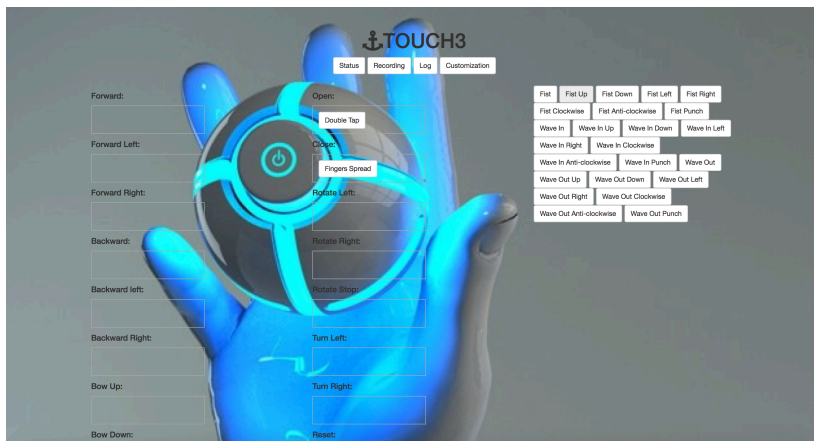
6. Recognize hand gestures by MYO armband successfully if appear the above picture

14.5 Customization Tab

1. Press Customization button to enter Customization tab



2. Enter Customization tab successfully if appear the above picture
3. The left hand side is droppable area representing iDroid's motions
4. The right hand side is draggable button representing hand gestures
5. The buttons in the bottom is operation control buttons, press Confirm button to store the new gesture control setting, press Default button to restore the default gesture control setting



6. Match the iDroid's motions with hand gestures successfully if appear the above picture

14.6 Terminate TOUCH3

1. Quit the web browser
2. Press control + C to terminate TOUCH3 server

15. Reference

- [1] MYO [Internet]. MYO; [cited 2015 Sep 30]. Available from: <https://www.myo.com/>
- [2] Forbes [Internet]. How Gesture Control And Wearable Tech Will Revolutionize Our Digital Lives; [updated 2013 Jul 17; cited 2015 Oct 2]. Available from: <http://www.forbes.com/sites/toyota/2013/07/17/how-gesture-control-and-wearable-tech-will-revolutionize-our-digital-lives/>
- [3] ThalmicLabs [Internet], Myo SDK Manual; [cited 2015 Oct 1]. Available from: https://developer.thalmic.com/docs/api_reference/platform/index.html
- [4] Bluetooth [Internet], Bluetooth Smart Technology: Powering the Internet of Things; [cited 2015 Oct 1]. Available from: <http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx>
- [5] Chrobotics [Internet], Understanding Quaternions; [cited 2016 Mar 15]. Available from: <http://www.chrobotics.com/library/understanding-quaternions>
- [6] Cecille Freeman [Paper], Feature Selection And Hierarchical Classifier Design With Applications To Human Motion Recognition; [cited 2016 Mar 23]. Available from: https://uwspace.uwaterloo.ca/bitstream/handle/10012/8480/Freeman_Cecille.pdf?sequence=3
- [7] Mahmoud Abduo, Matthias Galster [Paper], Myo Gesture Control Armband for Medical Application; [cited 2016 Mar 26]. Available from: https://www.cosc.canterbury.ac.nz/research/reports/HonsReps/2015/hons_1502.pdf
- [8] MYO Blog [Internet], Myo Unleashed: Myo.js; [cited 2015 Dec 20]. Available from: <http://developerblog.myo.com/myo-unleashed-myo-js/>
- [9] NPM [Internet], play-sound.js; [cited 2016 Feb 25]. Available from: <https://www.npmjs.com/package/play-sound>
- [10] Kristian Nymoen, Mari Romarheim Haugen, Alexander Refsum Jensenius [Paper], MuMYO – Evaluating and Exploring the MTO Armband for Musical Interaction; [cited 2016 Feb 28]. Available from: <https://nime2015.lsu.edu/proceedings/179/0179-paper.pdf>

[11] Ali Boyali, Naohisa Hashimoto, Osamu Matsumoto [Paper], Spectral Collaborative Representation Based Classification by Circulants and its Application to Hand Gesture and Posture Recognition from Electromyography Signals; [cited 2016 Jan 24]. Available from: <http://worldcomp-proceedings.com/proc/p2015/IPC6206.pdf>

[12] Sergey Lobov, Vasiliy Mironov, Innokentiy Kastalskiy, Victor Kazantsev [Paper], A Spiking Neural Network in sEMG Feature Extraction; [cited 2016 Apr 10]. Available from: <http://www.mdpi.com/1424-8220/15/11/27894/pdf>