GESTURE CONTROL ARMBAND

User Experience Guidelines

Revision 3 – November 27, 2014

Introduction

The Myo™ armband has the potential to transform the way people interact with their digital world. But without an ecosystem of Myo-enabled applications created by our developer community, they won't have a digital world to control. These guidelines are designed to help you create an amazing and consistent gesture control experience for your users.

Determining if the Myo armband is on the left or right arm

The handedness of the user is determined when they perform the **sync gesture**. The Myo SDK returns which arm the Myo armband is on.

The Myo armband xDirection (the direction the USB port is facing) is also returned by the Myo SDK. You may need to take this into account if your application uses the raw orientation data from the Myo SDK. In the smart thermostat example, the roll would need to be inverted in some cases so that rotating the arm always changes the temperature in the same direction.

Wave Left/Right vs Wave In/Out

Currently the SDK communicates the wave gestures as wave in and wave out. In many applications, including those cited in this guide, you should map wave in and wave out to wave left or wave right, depending on which arm is currently wearing the armband. Please see the table below for more details.

	Left Arm	**Right Arm**
Wave In	Wave Right	Wave Left
Wave Out	Wave Left	Wave Right

Use Common Gesture Mappings When Appropriate

The Myo SDK provides a set of gestures you can build into your application. You're encouraged to push the boundaries — but there are a few standardized gesture mappings that will ensure your users have the best experience with your application and

the Myo armband. In situations where you are using actions mentioned in this document, your gesture mappins should conform to these guidelines.

Lock/Unlock

The Myo SDK features an unlock mode, which is enabled by default. If you are making an application which uses unlocking functionality, you should use this mode by default. The **double tap** gesture is intended for use in situations where you want to activate the recognition of gestures from the Myo armband. For example:

- In a music player application, use this gesture to start recognition. When in the inactive state, the user will be prevented from making accidental gestures such as moving to the next track

Move Forward/Advance

Use the **wave right** gesture if your application has a forward or advance command. For example:

- Advance to the next slide in a presentation
- Move to the next track in an audio player

You can also use wave right to fast forward or fast advance by having the user hold the gesture.

Move Backward/Reverse

Use **wave left** when you want the user to move backward or reverse in your application. Similar to wave right, you can have the user wave left to move to the previous track of an audio player and hold wave left to fast rewind — especially useful in video or music players.

Select/Hold

The **fist** gesture is used for mapping to a button press or for manipulating an additional control element. For example:

- Map button clicks or select to the fist gesture
- For a dial or rotary control, pair fist with the orientation data in the Myo armband by mapping the relative roll from when they started performing the fist to a change in the variable being controlled

Play/Pause

The **finger spread** gesture is a universal play/pause control. If you don't have a play/pause function in your application, finger spread works well as a secondary button click or select, similar to fist. For example:

- If the Myo is a secondary controller for a game, fingers spread could reload or fire a secondary weapon.

Avoid requiring the user to remember state

Your application should not require the user to remember what state or mode they are in — unless there is visual or haptic feedback available. An example would be fast forward in an audio player application. Your application should either:

1. Use wave right and hold to fast forward.
2. Use wave right to enter a fast forward mode, then display a visual indicator in your application displaying this state to the user.

Do not require users to hold gestures for more than a few seconds

Avoid requiring the user to hold a gesture to engage an action. For example, if you have a flight simulator application, you should not make the user hold **fist** to engage flight for the entire duration of flight. Using rest states prevents exhausting the user.

Avoid relying on absolute orientation

Your application should use the relative orientation of the Myo armband on the user's arm. In the smart thermostat example, the user makes and holds a **fist** and then rotates their arm to change the temperature. The application looks at the change in roll from the time at which **fist** was initiated.

Graphic Assets for Poses and Gestures

The graphic assets can be downloaded from developer.thalmic.com/branding.

Revision History

Rev	Date	Notes
1	July 25, 2014	Initial release
2	October 9, 2014	- Renamed "setup gesture" to "sync gesture" - Added reference to haptic feedback for use in remembering state - Added pose assets
3	November 27, 2014	- Changed thumb to pinky gesture to double tap - Added details and recommendation for unlock modes