Robotic Arm
Technical Report

By: Kirolos Kaldas

Summary:

The aim of the robotic arm project is to build a LEGO EV3 robotic arm and control it using a Myo Gesture Control Armband. The EV3 brick has to be connected with the Myo armband to get motions and gestures from the arm to the robotic arm to execute different functions and make certain movements.

Introduction:

The Myo Gesture Control Armband is aimed to give motion control over devices via Bluetooth. The Myo is a lightweight elastic armband consisting of a number of metal contacts that reads the electrical activity of the muscles in the forearm and gives touch-free control of technology with hand gestures and motion. It has the ability to remotely control presentations like PowerPoint, and it can also replace the mouse and the keyboard.

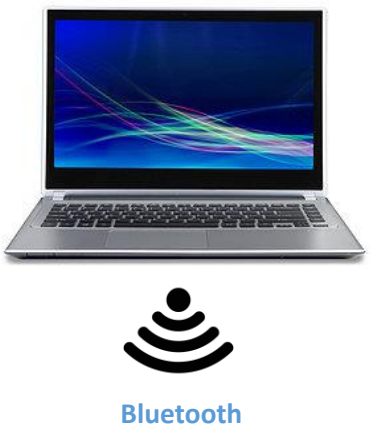
Lego Mindstorms EV3 is the third generation robotics kit in the Lego's Mindstorms technology. The technological advances in the programmable brick of the EV3 has been the biggest change from any other Lego Mindstorms. It also has a stronger CPU that runs Linux. Lego Mindstorms EV3 set comes with Lego pieces to build certain robots or structures, motors and sensors. The motors and sensors are connected to the EV3 brick to be controlled in a certain way according to the program on the EV3 brick.

Method:

- Steps of the Robotic Hand Project:
  1. Install Myo Connect to laptop
  2. Connect the Myo armband with laptop
  3. Install Maven Integration for Eclipse
  4. Go on https://developer.thalmic.com/forums/topic/541/
  5. Under Current Tools and Bindings List, Choose Myo Java
  6. Download the zip file from the GitHub link that you will be directed to
  7. Extract the files
  8. Import into Eclipse as an Existing Maven Project which is found under the Maven option. (May take a couple of minutes to create the workspace)
  9. Go to the Hub.java source file and find the loadX64ResourcesFromSysPath() function to edit the line; System.loadLibrary("JNIJavaMyoLib"); to System.loadLibrary("JNIJavaMyoLib64"); and find loadWin32ResourcesFromSysPath() function to edit the same line to System.loadLibrary("JNIJavaMyoLib32");
  10. Install LeJOS on the EV3 to program java on it, follow the steps on: http://sourceforge.net/p/lejos/wiki/Installing%20leJOS/
  11. Connect the laptop to the EV3 using Bluetooth, Wi-Fi or USB Ethernet
  12. Go on http://sourceforge.net/p/lejos/ev3/code/ci/master/tree/ and download the snapshot that is provided.

13. Extract the files
14. Import ev3classes, DBusJava and jna files into Eclipse as Existing Projects into Workspace
15. After importing, right click on the myo project that was imported and go to Build Path then click on Configure Build Path. Then click on the Projects tab and add the ev3classes project
16. Now the EV3 brick is connected with the Myo armband and you are ready to start writing a program to control the EV3 brick using the Myo armband. For a library on how to use the LeJOS functions, visit http://www.lejos.org/ev3/docs/overview-summary.html
17. Build the robotic arm
18. Test your program

- Architecture of the Connection between the Myo to EV3:



The idea behind this architecture is that the myo would send data such as gestures and arm movements to the laptop using Bluetooth then the laptop would send actions to the EV3 brick to execute using the motors and sensors. Other devices can be used instead of a laptop such as iOS, android, Mac devices. There are also other ways to connecting the laptop and EV3 together such as using WI-FI or USB Ethernet.

For a class diagram of the program architecture, visit https://www.lucidchart.com/documents/view/6c078fb1-5b38-4bc2-8eb3-f8f3e8aca449

Discussion:

There were a number of challenges throughout the project that were difficult to overcome. The first challenge was getting started with programming with myo because the only way to get started was through using maven which was a great learning curve. The second challenge was getting the myo and the EV3 connected together. A direct connection between them was attempted but it would have been very difficult and time consuming for a library for the myo was needed to be coded in the EV3 directly. Another challenge was getting familiar with the LeJOS programming interface to code the interaction between the EV3 brick and the myo. Many improvements could be made to the projects however one of the most important ones would be getting the myo to connect directly to the EV3 to send the data in a more efficient manner, to execute functions more rapidly and become more sensitive to arm movements..

Conclusion:

In conclusion, the idea behind this project is for cheap alternatives for artificial limbs and it can also be used by people who suffer from Parkinson's disease to be able to control an affordable hand to be able to grip anything that they cannot grip.