Human-Human Interface

Boniface Bassey/Bascharat Martin/Manisha Jadav

2. Februar 2016

Supervised by:
Prof. Kristof Van Laerhoven and Philipp M. Scholl
Wearable Computing Laboratory
Embedded Systems
University of Freiburg
Georges-Köhler-Allee 010
79110 Freiburg

Correspondence:
Manisha Jadav: *manisha_jadav@gmail.com*
Boniface Bassey: *basseyboniface@gmail.com*
Bascharat Martin: *basharatmartin01@live.com*
github: *https://github.com/Basharatmartin/myohhiproject*

1 Introduction

Humans often think, make decisions and are in control of their actions. Some of these actions are the physical movements of the fingers, wrist, arm, shoulder, neck, legs, blinking of the eyes, etc. They are all coordinated by the brain. The brain is the central control unit for the nervous system whose signal can e.g command the muscles to move. If one could read out a specific command signal of the brain, it could be used for various purposes and applications not limited by imagination either to investigate the responsiveness of a nervous system to a defected part of a body or command robots to perform certain actions simply by imitation.

In this experiment we read the brain signal *(EMG - ElectroMyoGraphy)* for controlling the movement of ones arm and by processing this signal, we are able to take control of the movement of another person's arm.

2 Hardwares

2.1 Myo

The Myo armband is a wearable gesture control and motion control device from Thalmic Labs. It has the following features;

- eignt EMG sensors,
- nine axis IMU containing gyroscope,
- three axis accelerometer and three axis magnetometer and
- Bluetooth LE 4.0.

2.2 RFduino

RFduino is an Arduino compatible microcontroller, with a Bluetooth 4.0 Low Energy built-in module. It provides 7 GPIO lines, which all supports digital I/O, analog input (10 bit ADC), I2C, SPI, UART and PWM, MCU Nordic nRF51822 Cortex M0 @ 16 MHz, Memory 16 KB on-chip SRAM, Storage 256 KB on-chip Flash and low power consumption.

2.3 BeagleBone Black

BeagleBoard is a low-power open-source hardware single-board computer by Texas Instruments. It is used as a control unit to control the flow of data from Myo to RFduino as well as a graphical visualization of the data received from Myo. Of relevance to this project is its Linux Kernel, Debian GNU/Linux to run our python program without much ado.

2.4 TENS device

TENS stands for (Transcutaneous Electrical Nerve Stimulation), a dual channel analogue device predominately used to prevent, or reduce, muscle atrophy. Its controls comes with three modes of operation namely, Continuous, Burst and Modulation. Its frequencies range is between 2 to 150 Hz with an adjustable impulse width of range between 50 to 250 microseconds duty cycle. The device intensity is continuously adjustable and can range from 0 to 90 mA peak.

3 The Human Interface - Implementation

3.0.1 Basic Idea

Our project is basically based on the idea from the website `https://backyardbrains.com`, in which they have used their own built hardware (spiker board), arduino microcontroller, tens device, and electrodes both at the input and output sides.

3.0.2 Modification

By following the idea from *backyardbrains* we have modified and replaced spikerboard with our own built hardware together with Myo (wirelessly)at the input and RFduino as well as beaglebone black. Replacing the one side of the project totally with wireless Myo gave us understanding about the Electromyography activity of the human body, different gestures of human body in our case *"movement of a humans arm"*.

3.0.3 Implementation:

The project consists of an input side(sensor), a control-unit and an output side(actuator).

Input Side (Myo Sensor)

Myo armband is used at the input side. The raw EMG data is obtained from its eight sensors at the frequency of 200Hz. Its means that, fresh EMG raw data is captured after every five milisecond of time via bluetooth (BLE 4.0) on the control-unit (beaglebone). By evaluating these eight different EMG sensors data, we are able to analyse the EMG activities of ones arm. This gives us insights of the brain controlling over our muscles. The values we get are basically in decimal ranging from

0 to 1024 or sometimes higher depending on the person (different people have either high or low EMG thresholds respectively).

Control Unit (beaglebone or a PC)

Beaglebone black microcontroller is used as a control unit in our project. It has two Bluetooth dongle connected to it. One is for receiving the data from Myo armband and another for sending the data to RFduino (BLE 4.0). RFduino is attached to our own built hardware through which the TENS device is further connected. Inside of the control unit, data acquisition and visualization are performed. Python language is the main source of acquisition and visualization of EMG raw data. Basic APIs *(PyoConnect_v2.0)* of Myo armband written in Python language is provided by Thalmic labs. With the help of these APIs, all kinds of data from Myo armband could be gotten. Nevertheless, the raw EMG data was our primary target.

Output side (Actuator TENS Device with electrodes)

TENS device together with our own built hardware is used at the output side. RFduino (BLE) is attached to the hardware in order to receive the emg data from the control-unit. Finding the EMG threshold is performed in RFduino microcontroller which eventually activates the TENS device. TENS device is caused to actuate the humans arm through electrodes which are placed on the surface of a human body in case of our project is humans arm.

4 Results

Starting of our project, we set two different milestones. If we were not able to achieve the human-human interface then we would be heading out to achieving human-machine interface. But with a teamwork and a support from Prof. Kristof Van Laerhoven and Philipp M. Scholl we have achieved our goal and were able to achieve three different movements:

1. fingers
2. hand left
3. whole arm

by corresponding different EMG signals responsible for each of these movements.

5 Challenges

5.1 Software

As we have used python as one of the programming environments. We did not have any experience with python language. However, we started learning it and performed different tasks on it. `PyConnect_v2.0` by Thalmic labs which is basically written in python language.

1. It was difficult for us to program the bluetooth data communication/synchronisation between RFduino and control unit in python language, we had problems with data lost and buffer overflow.

2. We tried a lot visualize the EMG signals graphically by using pyqtgraph and even matplotlib, but there was always a problem being trigged. But we finally got plotting by the help of pygame which was already inside of `PyConnect_v2.0` and it needed to be modified a bit and worked fine later on.

5.2 Hardware

Relay is used to switch the output of the TENS device, but the clicking sound it makes while switching was somewhat annoying. Experimenting with some transistors resulted to distortion of the TENS signal. Hence, due to time constraint and workload, we couldn't research further and therefore had to bear the noise the relay produced.

Acknowledgements

We cannot express enough gratitude to Prof. Kristof Van Laerhoven and Philipp M. Scholl for offering us the opportunity to do this project in their laboratory. Their unending supports and encouragements paved us the right way to conduct our project; they have passed a lot of time to this respect and were cordial to solving our problems. Thank you for believing in us and providing us with all the information we needed to go further with the project.

Literatur

[1] Experiment: Take Someone's Free Will.
https://backyardbrains.com/experiments/humanhumaninterface.

[2] Wearable Computing Systems, Lab course / Praktikum.
https://earth.informatik.uni-freiburg.de/teaching/lab-course-1

[3] RFduino
http://www.rfduino.com/product/rfd22102-rfduino-dip/index.html

[4] MYO armband
https://www.myo.com/

[5] Beaglebone Black
http://beagleboard.org/BLACK

[6] Impulse 3000
https://www.tensunitdepot.com/product/impulse-3000