Bionic Clicker Lab Session

Elliott Magee and Tom Carlson
Aspire Create, University College London, UK

1 Bionic Clicker

1.1 Background

The aim of the bionic clicker lab is to guide you through the development of a simple neuroprosthetic device. This device will use a myoware muscle sensor as its input and will output to a computer as a wireless keyboard.

The lab consists of one 2 hour session, three 3 hour sessions and a lab report.

This lab is based on the Bionic Clicker MkII device developed to demonstrate electromyography (EMG) based control to a lay audience. For further information watch the video at the following link: Bionic Clicker MkII.

Note: For sessions 2-4 please bring along your own laptop or device capable of receiving a bluetooth keyboard input (such as a smartphone or tablet)

1.2 Learning objectives

This lab has the following learning objectives:

Session 1 - EMG:

1. Set-up a muscle sensor for recordings
2. Use the picoscope USB scope
3. Take EMG recordings
4. *Basic EMG analysis and processing*

Session 2 - Control Circuit:

1. Design a basic control circuit
2. Write arduino code
3. Output to PC over bluetooth
4. *Produce a circuit diagram*

Session 3 - Bionic Controller:

1. Integrate control device with EMG input
2. Testing, fault-finding and debugging

Session 4 - Evaluation of Device:

1. Accuracy
2. Confusion Matrices
3. Effect of threshold
4. *Analysis of the device*

2 Session 1 - Analysing EMG activity

In the first session we will be looking at recording muscle activity using a myoware muscle sensor, as seen in figure 1, and a scope. We will be comparing the muscle the signals generated by RAW and SIG.

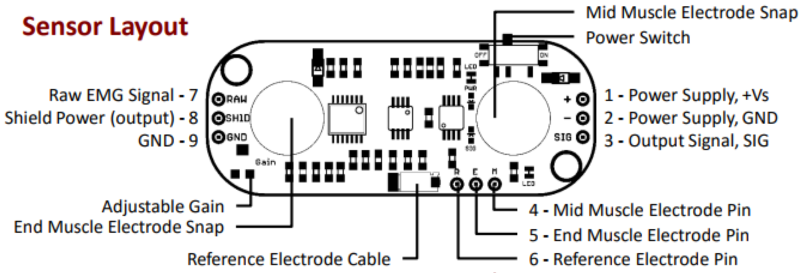


Figure 1: Diagram of the Myoware Muscle Sensor. From Advancer Technologies 2016.

2.1 Part 1 - Muscle and electrode selection

The myoware muscle sensor is designed so that it can be configured to be used in one of two ways: standard (see figure 2), where you use the electrode spacing of the device; and alternate, where you attach your own electrode cables for muscles where the standard spacing does not work. In these labs we will be using the standard configuration.

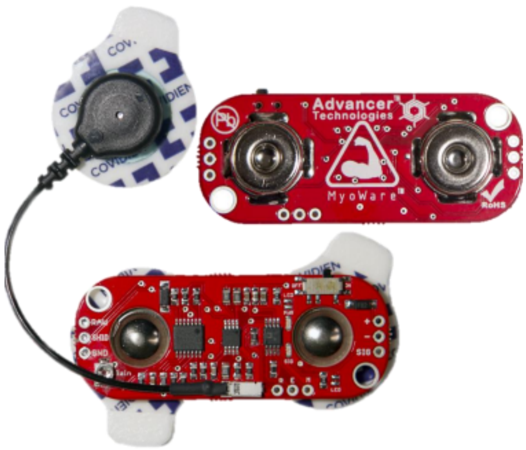
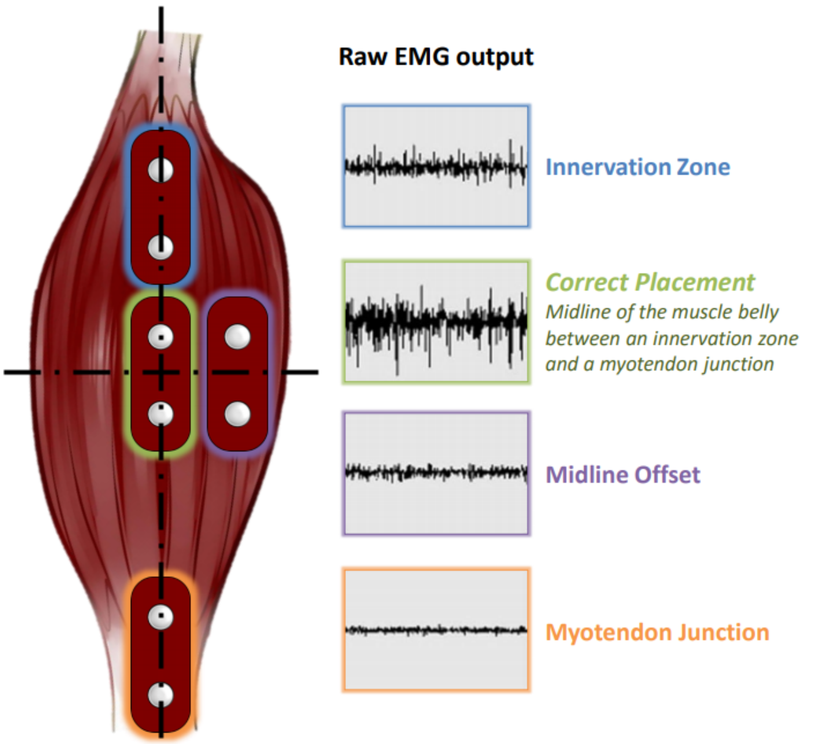


Figure 2: Picture of the Myoware Muscle Sensor. From Advancer Technologies 2016.

Discussion - Which muscles might be used for EMG analysis - why? What makes a muscle good for a control input - why? What can we do to make a muscle useful for control input - why?

2.2 Part 2 - Placing the electrode

Once you have clipped the electrode pads into the myoware, place the electrodes on the muscle. Figure 3 shows the ideal placement on a muscle and figure 4 shows the correct placement on the bicep.



Position and orientation of the muscle sensor electrodes has a vast effect on the strength of the signal. The electrodes should be place in the middle of the muscle body and should be aligned with the orientation of the muscle fibers. Placing the sensor in other locations will reduce the strength and quality of the sensor's signal due to a reduction of the number of motor units measured and interference attributed to crosstalk.

Figure 3: Placement of the Electrode. From Advancer Technologies 2016.

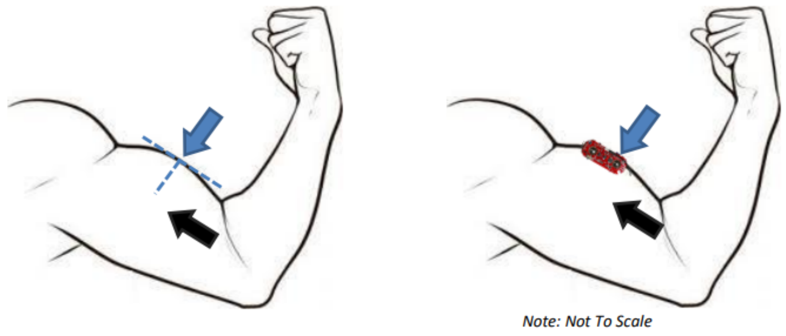


Figure 4: Example sensor location for bicep. From Advancer Technologies 2016.

2.3 Part 3 - Power the device

Attach the power shield to the myoware and place the batteries.

2.4 Part 4 - Attach the picoscope

ELECTRICAL HAZARD - DISCONNECT THE LAPTOPS FROM THE MAINS BEFORE PROCEEDING, ENSURE THERE IS NO PATH BETWEEN YOUR SKIN AND MAINS ELECTRICITY

Attach the wires provided to the Myoware shields. Next attach the Picoscope probes to the Myoware wires. Connect the grounds of two probes to the ground wire, the A probe to the RAW wire and the B probe to the SIG wire.

Turn on the picoscope software and make sure that you can observe signals from both pins.

2.5 Part 5 - Compare the signals

Periodically flex the muscle to which the Myoware is connected and observe how the signals change between the relaxation and contraction.

Discussion - What is the difference between RAW and SIG - what do you think is being done to the signals?

2.6 Part 6 - Calibrate the myoware

Adjust the gain switch on the Myoware until you see a large response on SIG, but do not get any plateauing. This is the device becoming oversaturated.

2.7 Part 7 - Recording the signals

We will be using the picoscope *record* function to record a series of EMG signals.

There are four scenarios to be recorded. Each will consist of a contraction period, where the muscle is contracted and a rest period, where the muscle is at rest. A single contraction and rest period is one cycle and each scenario consist of 20 cycle.

1. Scenario A: 1 second contraction, 1 second rest.
2. Scenario B: 1 second contraction, 5 second rest.
3. Scenario C: 5 second contraction, 1 second rest.
4. Scenario D: 5 second contraction, 5 second rest.

After each scenario (20 cycles) save the recorded file as a spreadsheet and name it SCENARIO_MUSCLE_YOURNAME for example A_BICEP_ELLIOTT. These files will be shared between all students on the SURGGRO3 / SURGMR03 module.

2.8 Part 8 - In your report

In your report you will be expected to analyse these data sets, with simple statistics such as mean and standard deviation of the signal for different periods (contraction/rest) and conditions (scenarios A-D). Which variables you compare will be up to you. Yo may use whichever software you prefer for the analysis (e.g. MATLAB, SPSS etc). A good comparison will further include a test to see whether or not there is a statistically significant difference between the conditions.

3 Session 2 - Control Circuit Design

In the second session we will be designing and building a basic control circuit. This will use basic electronic components and a "Feather" Arduino style device. We will then be adding bluetooth functionality to the control circuit via the feather to give control of some aspect of laptop functionality.

3.1 Part 1 - Input selection

First we need to select the input used for the device. This design is to be integrated with the Myoware next session so we can't have too many inputs.

Discussion - How can we have increased functionality with limited inputs?

3.2 Part 2 - Output Selection

Now is also a good time to select the desired "Keyboard" output of your device when connected to a PC, although we won't be using this immediately it is good to consider it now.

Discussion - What might be useful outputs for an assistive device that controls limited functionality on a PC?

Possible outputs can be found on this list of keyboard inputs: List of keyboard codes

3.3 Part 3 - Build the control circuit

Assemble the physical components required for your control circuit.

3.4 Part 4 - Write your code

Write the code for your control circuit, the following code chunks are provided to help with the writing code for a feather rather than a standard arduino and for serial communication. This code should allow the control inputs to output something over the serial communication to the laptops. This part can be done simultaneously with Part 5. All the following "chunks" of code are available to downlaod in a clean form as a txt file on Moodle, which will save you time copy-pasting (and subsequently editing) from this PDF.

Feather and serial initialisation

```
#include <Arduino.h>
#include <SPI.h>
#include "Adafruit_BLE.h"
#include "Adafruit_BluefruitLE_SPI.h"
#include "BluefruitConfig.h"

#define MINIMUM_FIRMWARE_VERSION    "0.6.6"


// Create the bluefruit object, as hardware serial
Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_CS,
   BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);

// The setup routine runs once when you press reset:
void setup() {
  Serial.begin(115200);
}
```

The following bits of code are provided to aid with serial communication. They give examples of printing strings, as well as the carriage return and new line commands.

Serial Print - BOOM

```
Serial.println("BOOM");
```

Serial Print - Carriage Return

```
Serial.println(\r);
```

Serial Print - New Line

```
Serial.println(\n);
```

3.5 Part 5 - Test and debug the code

Connect the control circuit to the feather and test the functionality of the code. If the code does not have the desired outcome then either the physical circuit or the code has a bug. Debugging is an iterative part of the design. Debugging is best done by flipping between parts 3,4 and 5. Only change one thing at a time and check the effect of the change, before testing the next potential problem.

3.6 Part 6 - Arduino code for feather and bluetooth

Now that we have the control circuit working over serial it is time to get it to work over bluetooth. Your code will need to be modified by replacing the old example code chunks with the following new chunks.

Below are some Arduino code chunks for using bluetooth communication with an arduino feather.

Feather and Bluetooth Serial Initialisation

```
#include <Arduino.h>
#include <SPI.h>
#include "Adafruit_BLE.h"
#include "Adafruit_BluefruitLE_SPI.h"
#include "BluefruitConfig.h"
#define FACTORYRESET_ENABLE         0
#define MINIMUM_FIRMWARE_VERSION    "0.6.6"


// Create the bluefruit object, as hardware serial
Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_CS,
   BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);

void setup()
{
  //if ( !ble.begin(VERBOSE_MODE) ) {
  ble.begin(VERBOSE_MODE); {
  }
  // Disable command echo from Bluefruit
  ble.echo(false);
  // Print Bluefruit information
  ble.info();
```

```
  // Change the device name to make it easier to find
  if (! ble.sendCommandCheckOK(F( "AT+GAPDEVNAME=name
     of device" )) ) {
  }
  // Enable HID Service
  if ( ble.isVersionAtLeast(MINIMUM_FIRMWARE_VERSION) )
  {
    if ( !ble.sendCommandCheckOK(F( "AT+BleHIDEn=On" ))
       ) {
    }
  } else
  {
    if (! ble.sendCommandCheckOK(F( "AT+BleKeyboardEn=
       On" ))) {
    }
  }
}
```

When writing your code replace *name of device* with the name you want for your device.

Replace the serial print commands with the bluetooth version of the commands. At this point you can select the actual desired outputs for control of the laptops and replace the serial print commands with the new commands.

Bluetooth Serial Print - BOOM

```
        ble.println("AT+BLEKEYBOARD=BOOM");
        //Print "BOOM"
```

Bluetooth Keyboard Code - Right Key

```
        ble.println("AT+BLEKEYBOARDCODE=00-00-4F
            -00-00-00-00");
        //Right key press down
        ble.println("AT+BLEKEYBOARDCODE=00-00");
        //keypress release
```

Bluetooth Keyboard Code - Left Key

```
ble.println("AT+BLEKEYBOARDCODE
    =00-00-50-00-00-00-00");
//Left key press down
ble.println("AT+BLEKEYBOARDCODE=00-00");
//keypress release
```

When you write your code, select a different function then the left or right key. For a list of possible keyboard functions check the following link: List of keyboard codes.

3.7 Part 7 - Test and debug the code

Disconnect the feather from the laptop and connect it to the batteries provided. Test the functionality of the code over bluetooth. If this doesn't do what you expect, then it is likely the the code has a bug. Debugging is an iterative part of the design. This is best done by flipping between parts 6 and 7. Again, only change one thing at at time and check the effect of the change, before testing the next potential problem. Progress from something simple amd gradually build in the additional complexity.

3.8 Part 8 - In your report

In your report you will be expected to include a circuit diagram of the control circuit you produced. You will not be expected to include your code. However you will be expected describe the functionality of the finished device.

4 Session 3 - Building a Bionic Controller

In this session the previous control circuit will be integrated with the Myoware muscle sensor.

Discussion - How do we use the EMG signal as an input? How do we use it for more than one control function?

4.1 Part 1 - Modify the control circuit

Now that we are integrating the Myoware with the Arduino, it is a good idea to have an override for the EMG input. This makes debugging and usage easier as the EMG can be deactivated when we need to move the electrodes or the EMG is misbehaving. A simple switch can be used for this function.

4.2 Part 2 - Modify the code

Modify your code for the EMG input and add in the EMG override. It is a good idea to keep the the initial button functionality intact.

4.3 Part 3 - EMG output code

Write an additional simple bit of code that can directly output the myoware EMG signal over the serial connection.

4.4 Part 4 - Test EMG output

ELECTRICAL HAZARD - DISCONNECT THE LAPTOPS FROM THE MAINS BEFORE PROCEEDING, ENSURE THERE IS NO PATH BETWEEN YOUR SKIN AND MAINS ELECTRICITY

Connect the myoware to the feather and monitor your EMG output at different positions. Make a note of the output values during contraction, movement and rest. These values will then be used as calibration for your device. You will want the EMG to trigger a command only when it rises above a certain threshold.

Discussion - What should the EMG threshold value be and why?

4.5 Part 5 - Finish the code

Put an appropriate threshold into your code (given the values you saw during the previous part and the subsequent discussion with your classmates).

4.6 Part 6 - Test the device

Connect all parts of the device together and test the functionality of EMG based control over the bluetooth connection. If everything works, well done! Otherwise it is time to debug the device again.

4.7 Part 7 - In your report

A photograph and description of your finished device, its intended use, functionality and limitations can be included.

5 Session 4 - Evaluation of Device

In the final session we will be evaluating the device.

Discussion - How can we evaluate the device? How would you define accuracy?

5.1 Part 1 - Confusion tables

A confusion matrix can be used to evaluate a classifier, in this case it will describe the actual output of your device, given your intended output for a (typically large) number of trials. If we imagine your device has only one input then we can use a simple confusion matrix. We then have two classes, yes or no. We can compare the actual states (intended) to the predicted (output) states. If your device triggers when we want it to then this is a **True Positive** (TP), if it doesn't trigger when we want it to this is a **False Negative** (FN), if it triggers when we don't want it to this is a **False Positive** (FP) and if doesn't trigger when it isn't meant to this is a **True Negative** (TN). See Figure 5 for a sample confusion matrix.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Figure 5: A sample confusion matrix from Markham 2014, simply recording the number of occurrences of each actual class (ground truth, in your case: your intended output), given the predicted class (in your case the output decoded by your Bionic Clicker device.)

5.2 Part 2 - Terminology of confusion

Below is some terminology to do with the the analysis of data in a confusion matrix. This is not an exhaustive list, but does contain many of the common terms.

1. **Accuracy** - Overall, how often is the classifier correct?
2. **Misclassification Rate** - Overall, how often is the classifier wrong?
3. **True Positive Rate** - When it is actually yes, how often does it predict yes?
4. **False Positive Rate** - When it is actually no, how often does it predict yes?
5. **Specificity** - When it is actually no how often does it predict no?
6. **Precision** - When it predicts yes, how often is it correct?
7. **Prevalence** - How often does the yes prediction actually occur in our sample?

Discussion - Using the confusion matrix, how would you calculate each of these metrics? I.e. can you derive each of the formulae? For your application, which of these metrics are most important and why - you should include the most relevant metric(s) in your lab report!

5.3 Part 3 - Effect of threshold

When calibrating your device you selected the threshold. This could have lie anywhere within a range of 'reasonable' numbers.

Discussion - What do you think the effect of your choice of threshold would be? What would be the effect of a 'low' threshold? What would be the effect of a 'high' threshold?

5.4 Part 4 - Replacement Code

If we want to analyse the accuracy of your bionic devices, we must first collect data. To do so we will conduct a simple experiment. First we must modify the code your device runs, so that it is easier to collect the data. Replace the code for the primary output of your device with a 'Print x', and if you have a secondary output the 'Print y' command.

Print x

```
ble.println("AT+BLEKEYBOARD=x");
//Print "x"
```

Print y

```
ble.println("AT+BLEKEYBOARD=y");
//Print "y"
```

ELECTRICAL HAZARD - DISCONNECT THE LAPTOPS FROM THE MAINS BEFORE PROCEEDING, ENSURE THERE IS NO PATH BETWEEN YOUR SKIN AND MAINS ELECTRICITY

5.5 Part 5 - Experiment set-up

The experiment will be a simple test of the devices. We will use an excel spreadsheet to record activation of the device, see table 1 for the layout of the spreadsheet. We will do the experiment once for each output with your initial threshold and once for each output with an adjusted threshold.

Name	*Elliott*	Primary Threshold	*700*
Muslce	*Bicep*	Secondary Threshold	*600 or N/A*
x	y *(or x)*	*Insert New Threshold*	*Insert New Threshold*
x	*y*	*xx*	*yy*

Table 1: Example data layout

5.5.1 Calibrate device

Take some time to check that the calibration of your device is still valid.

5.5.2 Experiment timings

The experiment will be broken into blocks of five seconds, with three blocks forming a single cycle. In the first block in a cycle you will attempt to trigger the device once, in the second block you will be stationary to avoid triggering the device, and in the third block you will move but not attempt to trigger the device. After each block of five seconds you will press enter to move down a cell in the column, each block of five seconds will be marked by a beep from the tv. The experiment will consist of 20 cycles and will be conducted once for each output

The spreadsheets will be shared with each other for analysis.

5.5.3 Adjusted Threshold

After conducting the inital experiment you will each change the threshold of you device by 20%. Half of you will increase it and the other half will lower it. The experiment will then be repeated with these new values for each output. The data from this experiment will then allow for analysis of the effect of threshold on the confusion matrix.

5.6 Part 7 - In your report

Your report should explain how you have decided to evaluate your device (which variables to calculate) and why they are important for your application. You should include the relevant results (e.g. confusion matrices and appropriate metrics derived from them. Given your experimentation you should be able to propose a method for calibrating the device (i.e. selecting a good threshold) for your particular application and explain the issues that may occur if the device is incorrectly calibrated.

Finally your report should wrap up the key scientific findings of your lab sessions and explain what they mean in the wider context. What limitations exist, what might you change or develop further if you were to continue the project?

Remember, you have already been given the lab report marksheets, which clearly define all the criteria against which your report will be assessed, so make sure you keep these criteria in mind whilst writing your report.

References

Advancer Technologies, L L C (2016). *MyoWare Muscle Sensor Manual.* URL: https://github.com/AdvancerTechnologies/MyoWare_MuscleSensor/raw/master/Documents/AT-04-001.pdf (visited on 10/06/2017).

Freebsd. *List of keyboard codes.* URL: http://www.freebsddiary.org/APC/usb_hid_usages.php.

Magee, Elliott. *Bionic Clicker MKII.* URL: https://www.jove.com/video/55705/the-bionic-clicker-mark-i-ii.

Markham, Kevin (2014). *Simple guide to confusion matrix terminology.* URL: http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/.