

# MyoSensor Application

## Introduction

This document outlines the set-up and installation to build and run the MyoSensor Application.

The MyoSensor application has been designed to provide an affordable solution to engage and stimulate an amputee during the process of training to use a myoelectric prosthesis. The application will display live sensor readings and dynamically draw a line graph, the keyboard control allows each sensor to be mapped to any key or mouse control opening up endless possibilities from controlling computer games, writing in a text document or controlling the virtual prosthetic hand simulation.

The set-up and instructions will include the live data reading and recording and keyboard control.

## Hardware requirements

- Arduino – (Mega 2560 tested)
- USB 2.0 Type A Male to Type B Male Power Data Transmission Cable
- Two MyoWare Muscle Sensor (Advancer Technologies used for the build tested)
- Arduino Software IDE
- Jumper Cables
- Breadboard (Optional)

## Arduino set-up

### Arduino and MyoWear hardware

Full instructions provided Advancer Technologies

(<https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf>)

Simple set-up but as highlighted by Advancer Technologies warning as no isolation included, use with caution. This set up would require Arduino to have two ground and 5V ports.

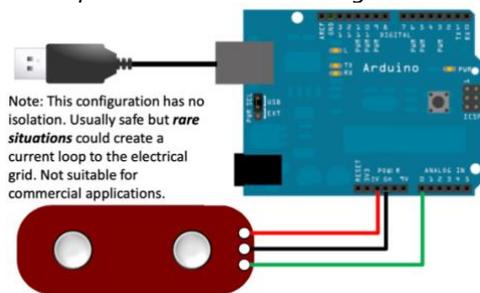


Figure 1. Arduino MyoWear Configuration (Advancer Technologies, 2015)

Circuit configuration with bread board (clean design) still requires warning of no isolation, use with caution.

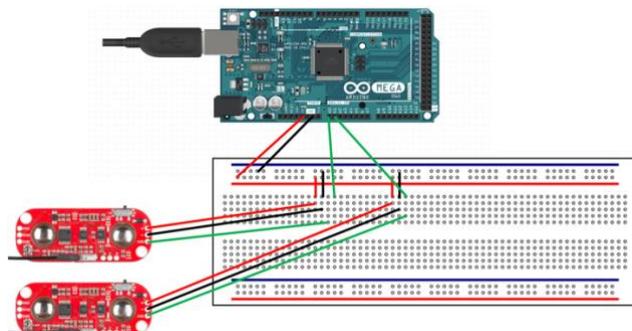


Figure 2. Arduino MyoWear Configuration

## Arduino Software IDE

1. Install the Arduino Software IDE
2. Open "TwoSignalReadSerial.ino" or copy and paste code below

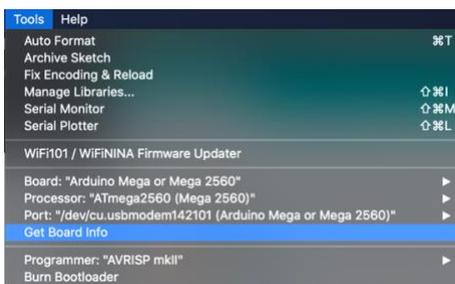
```
//declare two int's with pin values
int pinOne = A0;
int pinTwo = A1;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // initialise A0 and A1 pins and declare as inputs
  pinMode(pinOne, INPUT);
  pinMode(pinTwo, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0(pinOne) and one(pinTwo):
  int sensorValueOne = analogRead(pinOne);
  int sensorValueTwo = analogRead(pinTwo);

  //declare char array text and store values
  char text[40];
  sprintf(text, "%d,%d\n", sensorValueOne, sensorValueTwo);
  //print text array
  Serial.println(text);
  //delay 50ms
  delay(50);
}
```

3. Ensure under tools tab in Arduino IDE Board, Processor and Port is set and selected. If you have difficulties select the Port the board is connected to and click GetBoard Info which will auto generate the settings in Board and Processor.



4. Verify (Tick Icon) the code and then upload (Arrow Icon) by clicking the icons in the top left-hand corner:

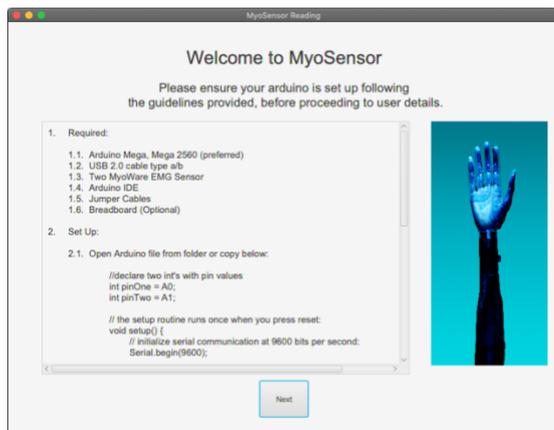


Now the board and Arduino IDE is set-up, you may wish to confirm by checking the serial plotter or serial monitor which will display live readings from the myo-sensor readings within the Arduino IDE. Both serial plotter and monitor found under Tools tab as shown above in point three.

## MyoSensor Application

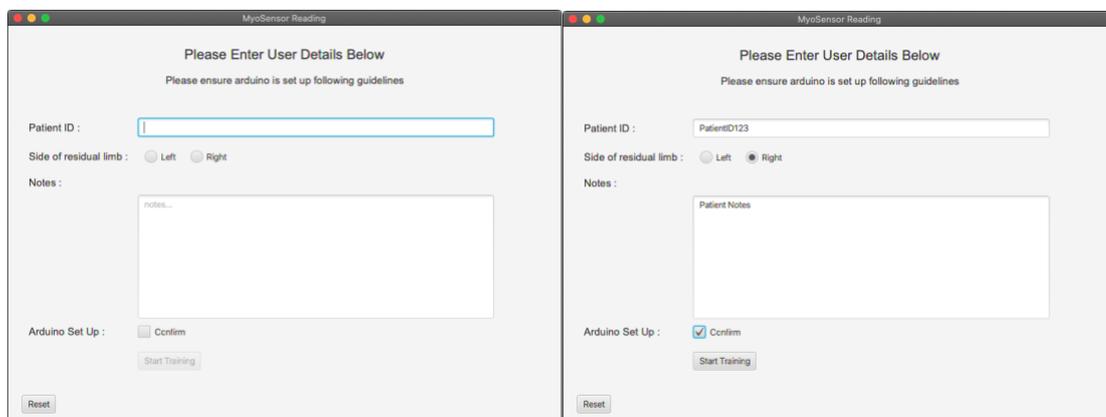
### 1. Open Welcome Window

- i. Displays welcome message and reminder to have Arduino connected
- ii. Light instructions provided
- iii. Click Next button to progress



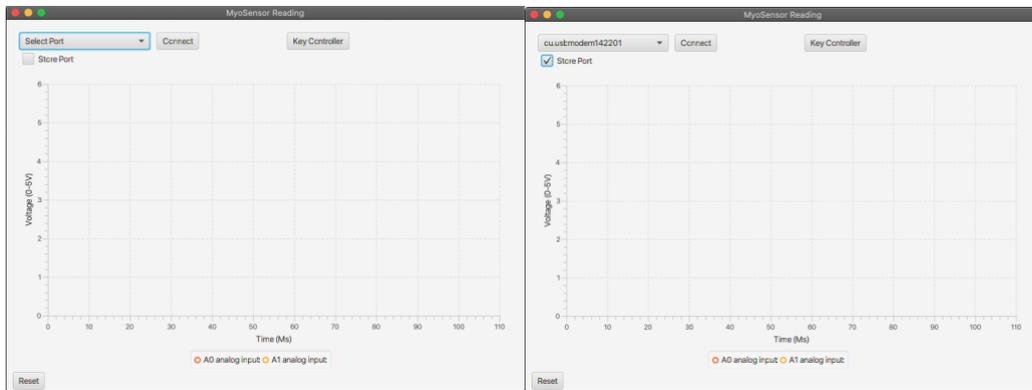
### 2. User Details Window

- i. User details saved and written to json file
- ii. Start Training button will only show once all fields are validated
- iii. Fields:
  - i. Patient ID: 2-20 characters
  - ii. Side of residual limb: tick-box Left or Right (only one can be selected)
  - iii. Notes: 1-unlimited characters
  - iv. Arduino Set-Up: tick-box Confirm
  - v. Start Training: Proceed to training (only active when all fields validated)
  - vi. Reset: Return to Welcome Window



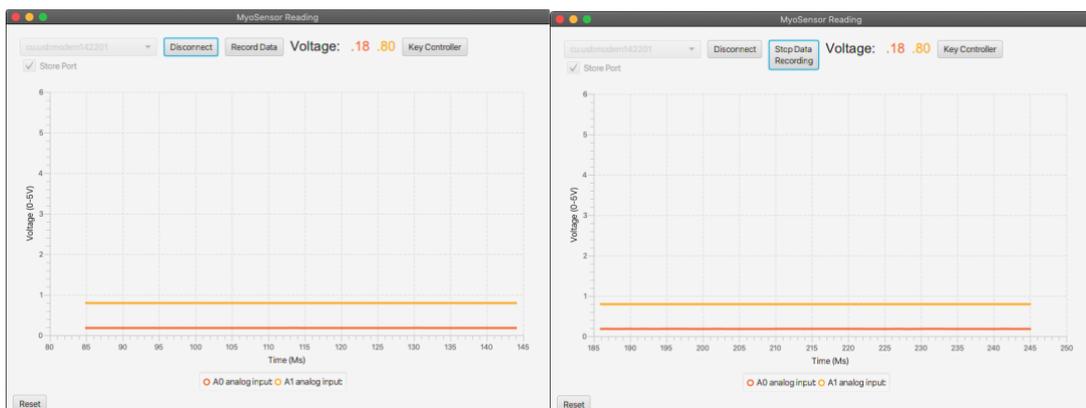
### 3. Training Window (Port Selection)

- i. Select serial port to connect read Arduino data
- ii. Button interaction:
  - i. Select Port: dropdown menu will display available serial ports
  - ii. Store Port: tick-box, if selected will store the port and automatically select on next session if the Arduino is connected.
  - iii. Connect button: Starts to display live readings
  - iv. Reset: Return to Welcome Window (useful to reload port list if Arduino not connected)



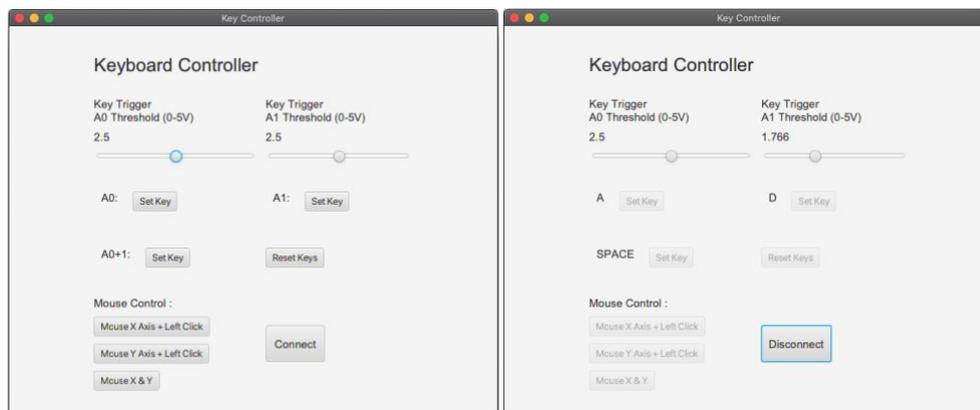
### 4. Training Window (Connect Clicked)

- i. Live display of electrode sensor readings:
  - i. Sensor One (Pin Ao - orange)
  - ii. Sensor Two (Pin A1 – yellow)
  - iii. Sensor signal displayed in Voltage range (0-5V) as Arduino circuit produces 5V
  - iv. Displayed live on line graph and top-bar Voltage:
- ii. Button interaction:
  - i. Disconnect: stop reading live data from serial port
  - ii. Record Data: record and store data
    1. Stop Data Recording: displayed when Record Data clicked
    2. If Disconnect clicked while recording, Stop Data Recording automatically selected
    3. Sensor One and Two y-axis (Voltage (0-5V)) recorded along with x-axis time stamp (Time (Ms))
  - iii. Key Controller: Opens Key Controller Window
  - iv. Reset: Return to Welcome Window (if clicked does not record data to file)



## 5. Key Controller Window

- i. Keyboard Controller window allows mapping of three keys to sensor signals or control of the mouse movement and function.
- ii. Threshold:
  - i.  $A_0$  and  $A_1$  threshold automatically set at 2.5V using slider this can be set between 0-5V
  - ii. Once signal is greater than threshold key or mouse control mapped is activated
- iii. Set Keys:
  - i. Click Set Key then click the key on the keyboard to be mapped, this will then replace the signal reference and display the mapped key.
    1.  $A_0$  - Once signal greater than  $A_0$  threshold the mapped key will be pressed
    2.  $A_1$  - Once signal greater than  $A_1$  threshold the mapped key will be pressed
    3.  $A_0+1$  (co-contraction) - Once signal greater than  $A_0 + A_1$  threshold the mapped key will be pressed
  - ii. Reset Keys: reset all mapped keys
- iv. Mouse Control:
  - i. Mouse X Axis + Left Click: signals control x-axis movement and co-contraction clicks the left mouse button
  - ii. Mouse Y Axis + Left Click: signals control y-axis movement and co-contraction clicks the left mouse button
  - iii. Mouse X & Y: signals control x & y axis movement direction, ENTER key used to cancel out of connection
- v. Connect:
  - i. Connect will activate key or mouse control
  - ii. If mouse control activated key-control not active
  - iii. Connect will display disconnect when active



## 6. Recorded Data:

- i. Please see Report Generator manual for automatic report creation.
- ii. Recorded data will be stored in a folder created using the Patient ID and current Date
- iii. Files:
  - i. CSV:
    1. Contains x-axis Time(MS) and y-axis Signal $A_0$  and Signal $A_1$
    2. Can be imported directly into Mac and Windows Excel
  - ii. JSON:
    1. Contains user details and recorded data
    2. Can be imported directly into Windows Excel
  - iii. TXT:
    1. Contains user details
    2. Can be viewed in general text editor
- iv. Example of file location with PatientID ASmith:



## References

- *Advancer Technologies (2015) 3-lead Muscle / Electromyography Sensor for Microcontroller Applications, Available at: <https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf> (Accessed: 9th August 2020).*