Experiment: Advanced Neuro-Prosthetics-Take Someone's Free Will

Have you ever had the desire to lose your free will? Or, alternatively, have you ever had the desire to take someone's free will? Now you can with our new "Human-Human-Interface Experiment."

Time 2 hours

Difficulty Advanced

What will you learn?

In a continuation of our 1) Neuroprosthetics Experiment and 2) EMG SpikerShield Experiment, you will learn how to use your muscle signal to control other devices, in this case, a TENS (transcutaneous electrical nerve stimulator) to excite and contract the muscle of another human.

Prerequisite Labs

- EMG SpikerShield - You should become familiar with how to use your EMG SpikerShield and an Arduino

Equipment

Human-Human Interface Kit

Making the Connection:

The Human-Human Interface is available as an easy to use complete kit! It comes with everything you need, including an arduino board already programmed with the HHI code! For the fastest way to begin imparting your will upon others, check it out in our store! If you're a budding neuro-prosthetic designer, and you're still interested in building and gathering the devices yourself, check out our DIY page for further instructions!

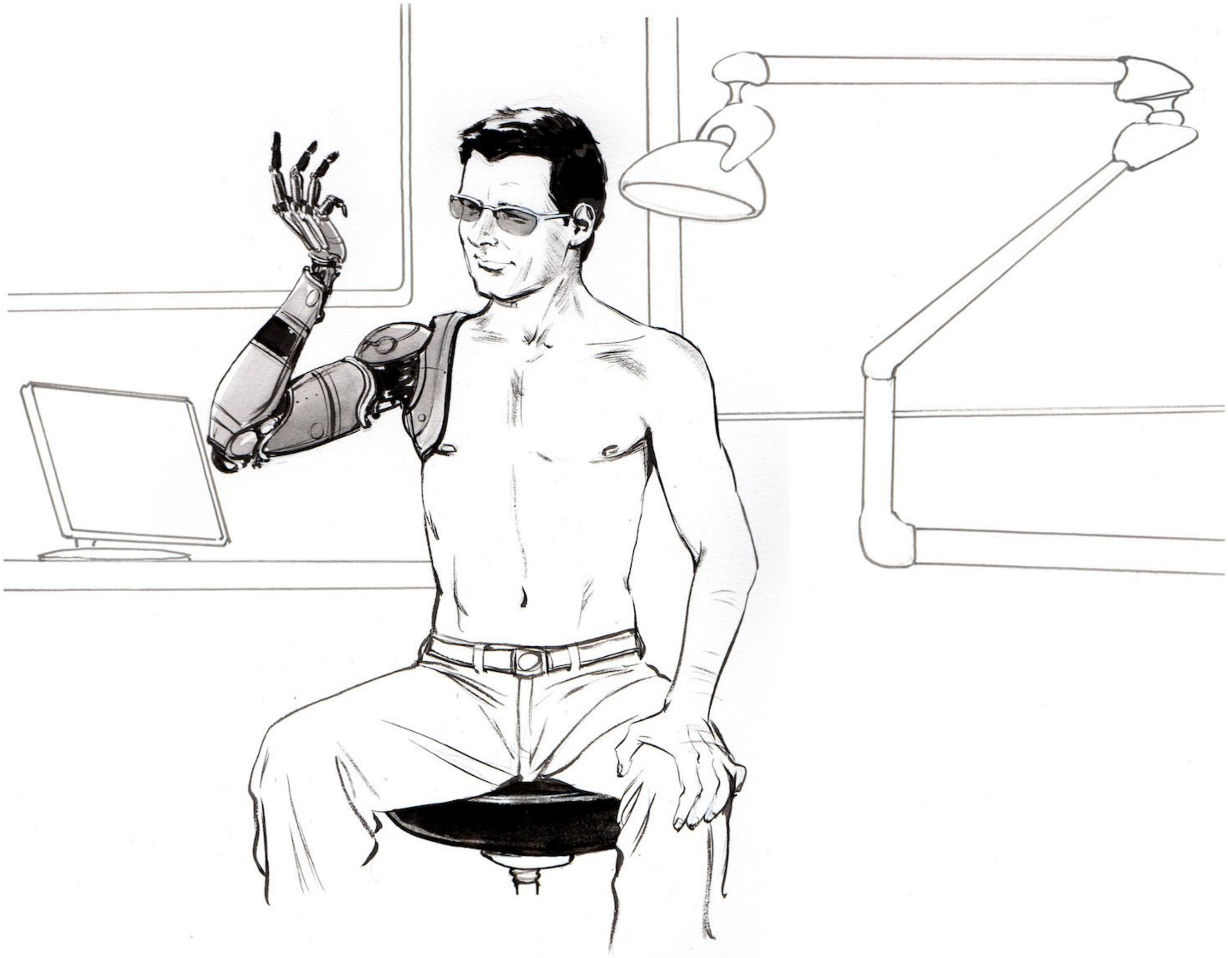
Background

Note: A TENS unit by definition delivers enough current to cause muscle contraction. ***Do not place electrodes across the muscles of the throat or the chest.*** *This experiment is appropriate for college students. Adult supervision is needed for younger (high school) users.*

We have previously discussed "neuroprosthetics," that is, designing a machine that interfaces with living neurons to control a device or for sensory substitution. But what about muscles? If people have damage to their spinal nerves, the muscles themselves can be stimulated, and this line of research is called "functional electrical stimulation."

For example, functional electrical stimulation can often be used to help someone stand up, or to improve walking by helping to swing a foot forward.

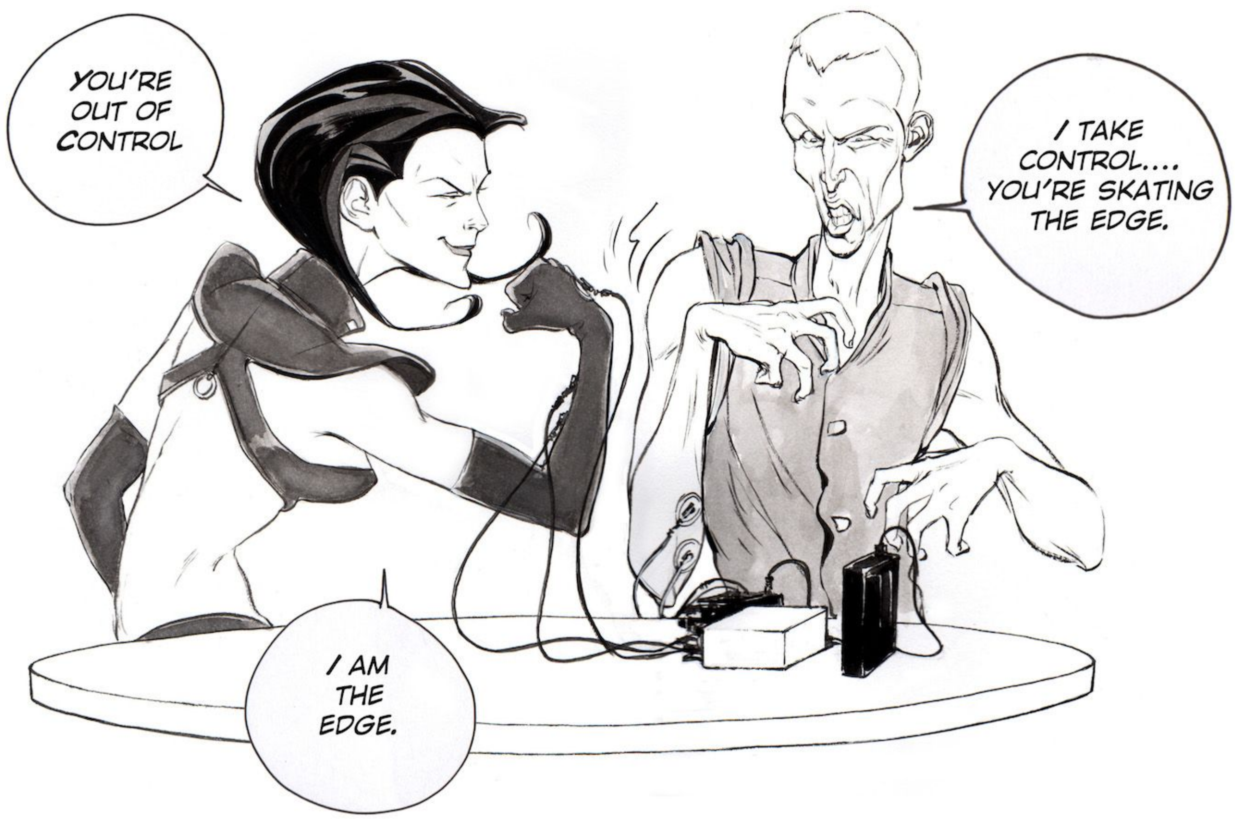
In addition, if someone has an artificial limb, such as a robotic arm, they can use the electrical activity generated by the electromyographic signals in their pectoral muscles to control the motors and control systems in the robotic arm. A notable example is Jesse Sullivan, who lost his arms during an electrical installation. There is also a company called RSLSTEEPER that has developed a bionic arm called the"bebionic3" which functions on very similar principles, but uses muscular activity in the amputee's stump to control the wrist and hand movements.



Relation to RoboRoach

Fans of the RoboRoach have often asked us: "When will you get this working on humans?" We responded and solved this customer request by launching our "Society" invention. But many people were left unsatisfied. They wanted something more direct.

On the other hand, critics of the RoboRoach have often asked us: "How would you like it if someone controlled you with electrical stimulation?" Now, after months in the research lab, we finally now have the answer, and it is...



"...it depends.""

If you like who is controlling you, the feeling can be quite nice.

Downloads

Your HHI arduino comes pre-loaded with the HHI code. If you use your arduino for other projects, if you accidentally press the reset button, or would like to adjust the thresholds, you will have to re-upload the code! You can do this with the Arduino software. If you are new to Arduino and need to learn how to upload code, check out this helpful guide

Arduino Software

Download our Arduino(.ino) Human-Human-Interface Sketch

HHI_switches_gripper.ino

Video

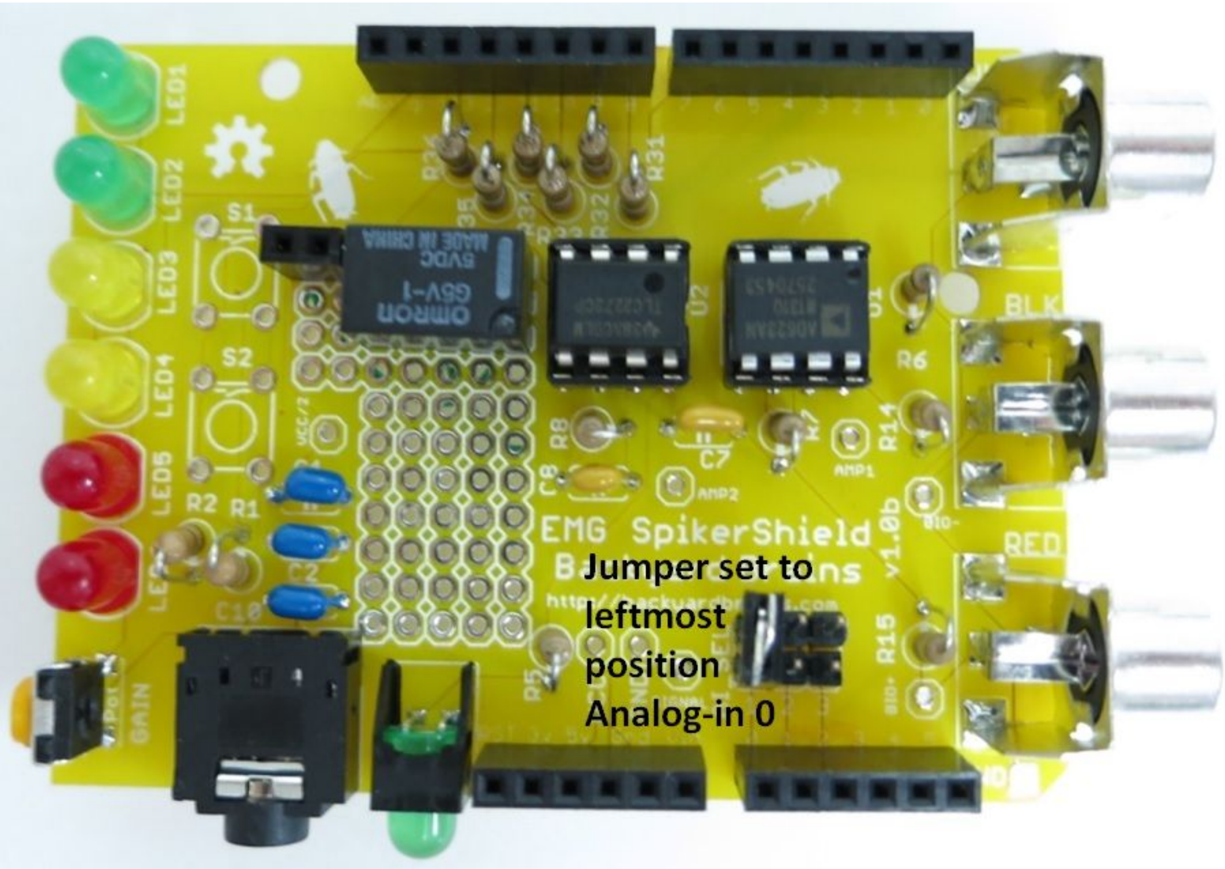
https://youtu.be/fGMDq3ay7Ro

The Code

Explanation of the HHI_switches_gripper.ino code

Some things to note:

1. We have the EMG input set to analog-in 0. This is done with a jumper. You can change the jumper position to set the analog-in to analog-in 0 (left), analog-in 1 (middle), or analog-in 2 (right). This allows you to stack three EMG SpikerShields at once should you prefer to do so for other experiments recording from multiple muscle groups...


Jumper set to leftmost position Analog-in 0

2. The LEDs are incorporated into the circuit through the jumper to the chip

3. We have the Stimulation output (to activate the relay) set to Digital Out

In the code, the **analogRead** function is used to convert the EMG signal into a number between 0 and 1024. This number, which updates every 100 ms, is called "finalReading". We then set an "if" statement threshold with the line " **if (finalReading > 3)**". Final reading is then matched against the threshold to determine how many LEDs will light up and whether or not the relay circuit gets tripped. When the EMG signal goes above a "3" we activate the relay with the Digital Out 3. This sends a 1 second digital high pulse, activating the relay, allowing current to flow through the TENS unit and stimulate the muscle of your partner. You can change the threshold from "3", found at the beginning of the code as **#define Threshold 3** to a lower number to make stimulating someone easier (less EMG amplitude), or a higher number to make stimulating someone harder (more EMG amplitude).

The Code

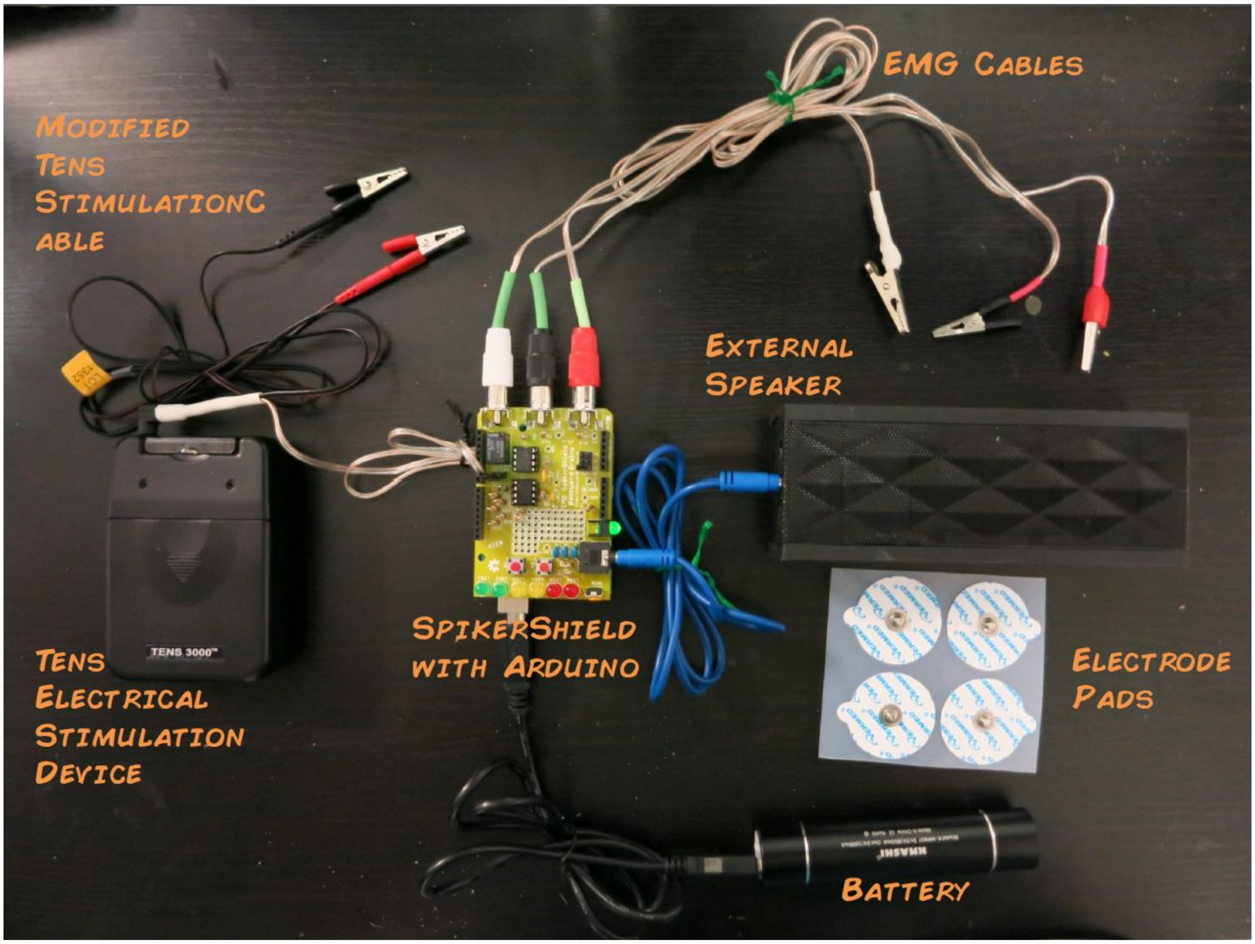
```
/* Arduino Code for an EMG SpikerShield to control a TENS device, LED lights, and a
gripper */
```

```
#include  //Includes the Gripper Servo Library
Servo ServoGripper; //Declares the Name of the Servo to be ServoGripper -- this is all
if you want to control a gripper hand as well

#define NUM_LED 6  //sets the maximum numbers of LEDs
#define MAX_Low 100   //for people with low EMG activity
#define MAX_High 254//for people with high EMG activity
#define Threshold 3 // this sets the light to activate TENS
#define threshold_degrees 10 //Number of steps the Servo will have

int reading[10];
int finalReading;
int StimPin = 3; // TENS Digital 3
int SwitchPin = 4;   // pushbutton connected to digital pin 4 -  this will momentarily
turn on TENS
int SwitchThreshold = 7; //pushbutton connected to digital pin 7 - this will switch
thresholds
int GripPin = 2; //Digital Out that controls Servo Motor Gripper Hand
int SwitchState = 0;
int SwitchThresholdState = 0;
int MAX = 0;
byte litLeds = 0;
byte multiplier = 1;
byte leds[] = {8, 9, 10, 11, 12, 13};
int aQ1 = 11;
int aQ2 = 13;
int aQ3 = 8;
```

```
const int UpdateTime = 200; // (number of milliseconds between updating servo position
-- if too low you will burn motor out)
unsigned long OldTime = 0;
int old_degrees = 0;
int new_degrees = 0;

void setup(){
  Serial.begin(9600); //begin serial communications
  ServoGripper.attach(GripPin); //Declare the Servo to be Connected to GripPin
  pinMode(StimPin, OUTPUT); // Set TENS output to StimPin
  pinMode(SwitchPin, INPUT);      // sets the digital pin 4 as the switch input
  pinMode(SwitchThreshold, INPUT);      // sets the digital pin 7 as the threshold
changer
  for(int i = 0; i < NUM_LED; i++){ //initialize LEDs as outputs
    pinMode(leds[i], OUTPUT);
  pinMode(aQ1, OUTPUT);

  }
  MAX = MAX_High; //This sets the default to people with high EMG activity.



}

void loop(){

    //note the serial print stuff is sjust debugging tools
    SwitchThresholdState = digitalRead(SwitchThreshold);
    if (SwitchThresholdState == HIGH){ // This will allow the switching between a low threshold and high threshold state on S2
    if (MAX == MAX_High){ MAX = MAX_Low; digitalWrite(aQ3, HIGH);}
    else{
     MAX = MAX_High; digitalWrite(aQ2, HIGH); }
    while (SwitchThresholdState == HIGH) { // This will pause the program while the person is touching the threshold button,
    SwitchThresholdState = digitalRead(SwitchThreshold);
    delay(10); }                         // so it doesn't flip back and forth while button is pushed
    }

    SwitchState = digitalRead(SwitchPin);
    //Serial.println(SwitchState);
    if (SwitchState == HIGH){ // This will activate the TENS if the PushButton is pressed on S1
    digitalWrite(StimPin, HIGH), digitalWrite(aQ1, HIGH);
    //Serial.println('Stim');
     while (SwitchState == HIGH) { // This will pause the program while the person is touching TENS test button
    SwitchState = digitalRead(SwitchPin);
    delay(10); }
    }

  for(int i = 0; i < 10; i++){    //take ten readings in ~0.02 seconds
    reading[i] = analogRead(A0) * multiplier;
    delay(2);
  }
  for(int i = 0; i < 10; i++){   //average the ten readings
    finalReading += reading[i];
  }
  finalReading /= 10;
  for(int j = 0; j < NUM_LED; j++){  //write all LEDs low and stim pin low

    digitalWrite(leds[j], LOW);
    digitalWrite(StimPin, LOW);
  }
  Serial.println(finalReading);
  //Serial.print("\t");
  finalReading = constrain(finalReading, 0, MAX);
  litLeds = map(finalReading, 0, MAX, 0, NUM_LED);
  //Serial.println(litLeds);

  for(int k = 0; k < litLeds; k++){
    digitalWrite(leds[k], HIGH); // This turns on the LEDS
    if (k >= Threshold){
      digitalWrite(StimPin, HIGH); // This turns on the TENS as a function of which LED
is lit
    }
  }

  new_degrees = map(finalReading, 0 ,MAX, 165, 0); //Translate the analog reading to
degrees for the servo (from 165° to 0°).

if (millis() - OldTime > UpdateTime){
  if(abs(new_degrees-old_degrees) > threshold_degrees){
    ServoGripper.write(new_degrees); //Move the servo according to the degrees
calculated
  }
  OldTime = millis();
  old_degrees = new_degrees;
}
   //delay(10);
  //for serial debugging, uncomment the next two lines.
  //Serial.println(finalReading);
  //delay(100);
}
```

Experimental Procedure

Set Up

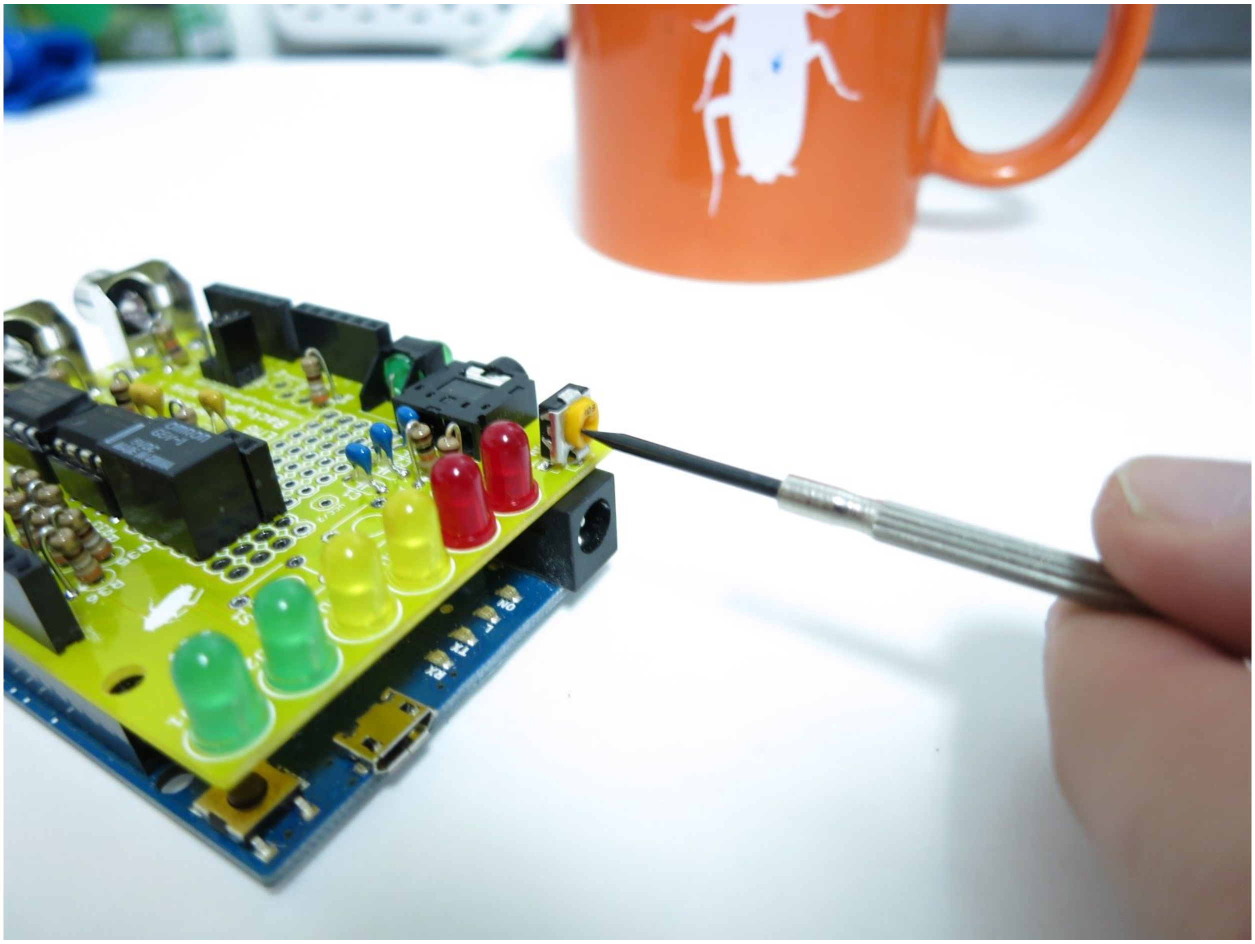

MODIFIED TENS STIMULATIONC ABLE
EMG CABLES
EXTERNAL SPEAKER
SPIKERSHIELD WITH ARDUINO
TENS ELECTRICAL STIMULATION DEVICE
ELECTRODE PADS
BATTERY

The Controller

1. Plug the EMG electrode cables (RCA) into the EMG SpikerShield, matching the red, black, and white.
2. Place two EMG electrodes on the forearm, one on the back of the hand, and clip on the EMG cables using the alligator clips. Red and black on the forearm muscle, and the ground (white/bare metal clip) on the back of the hand.
3. Now, have the controller flex their forearm muscles. With a sufficient strong flex, your controller should be able to light up all the LEDs. If the LEDs are not lighting up, you can:

a) change the numberical value of the **#define Threshold 3** statement in the Arduino code (using the Arduino Software) to a lower value to make the threshold lower. Note that you will then have to reupload this code to the Arduino if you want to change this value.

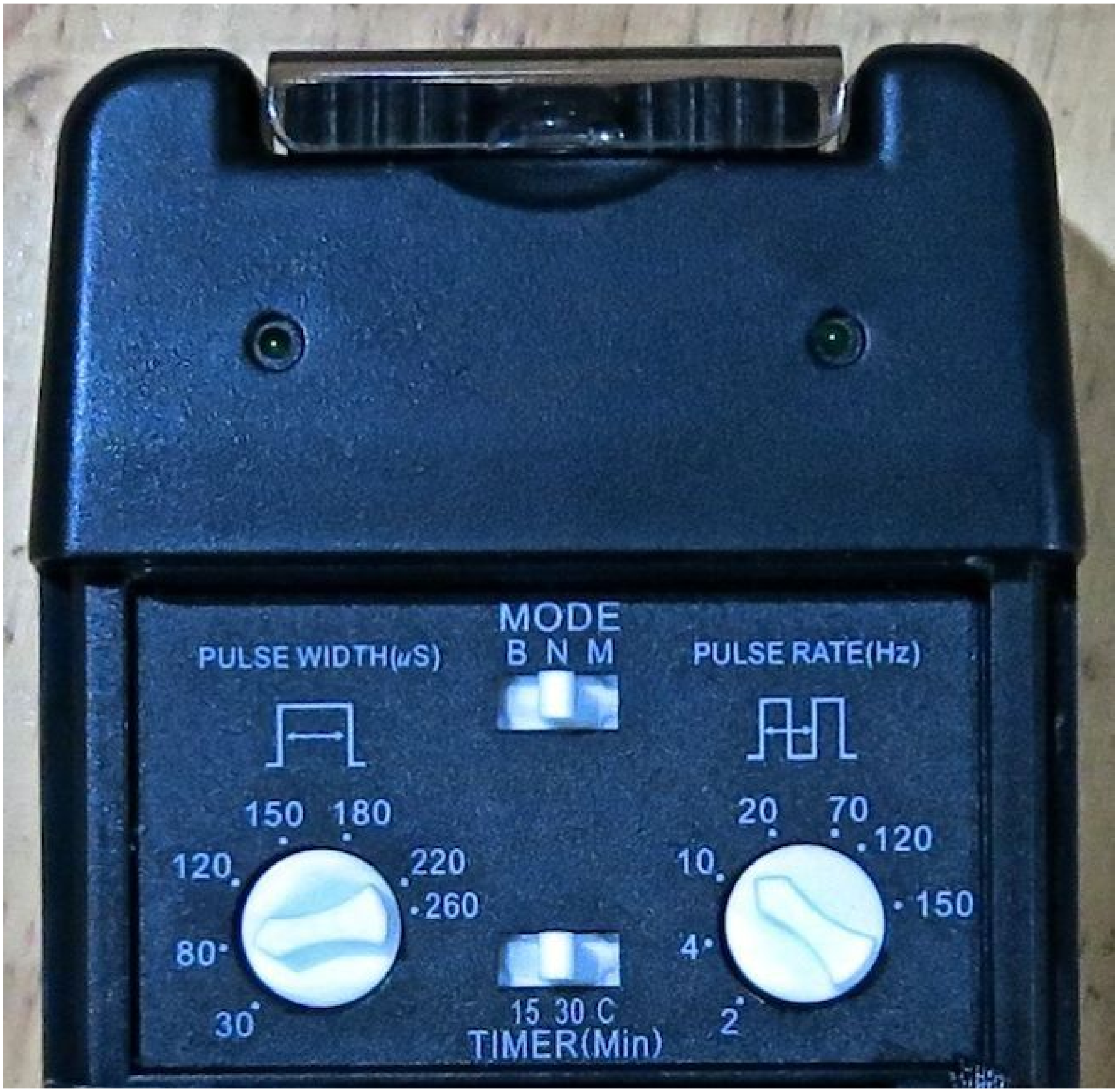
b) turn up the gain on the EMG SpikerBox by turning the orange potentiometer screw counterclockwise.



Now you are ready to hook up the "Controlled Human."

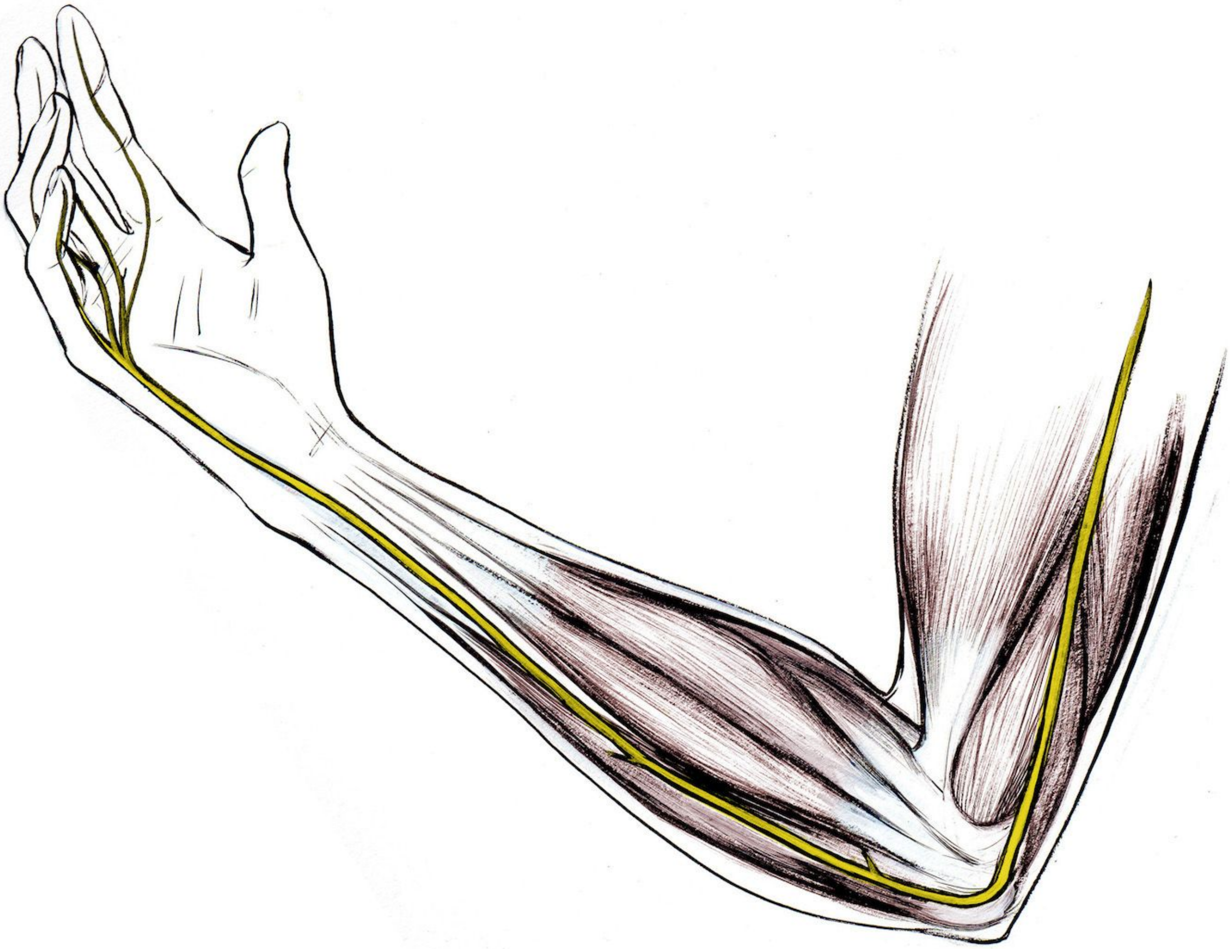
The Controlled

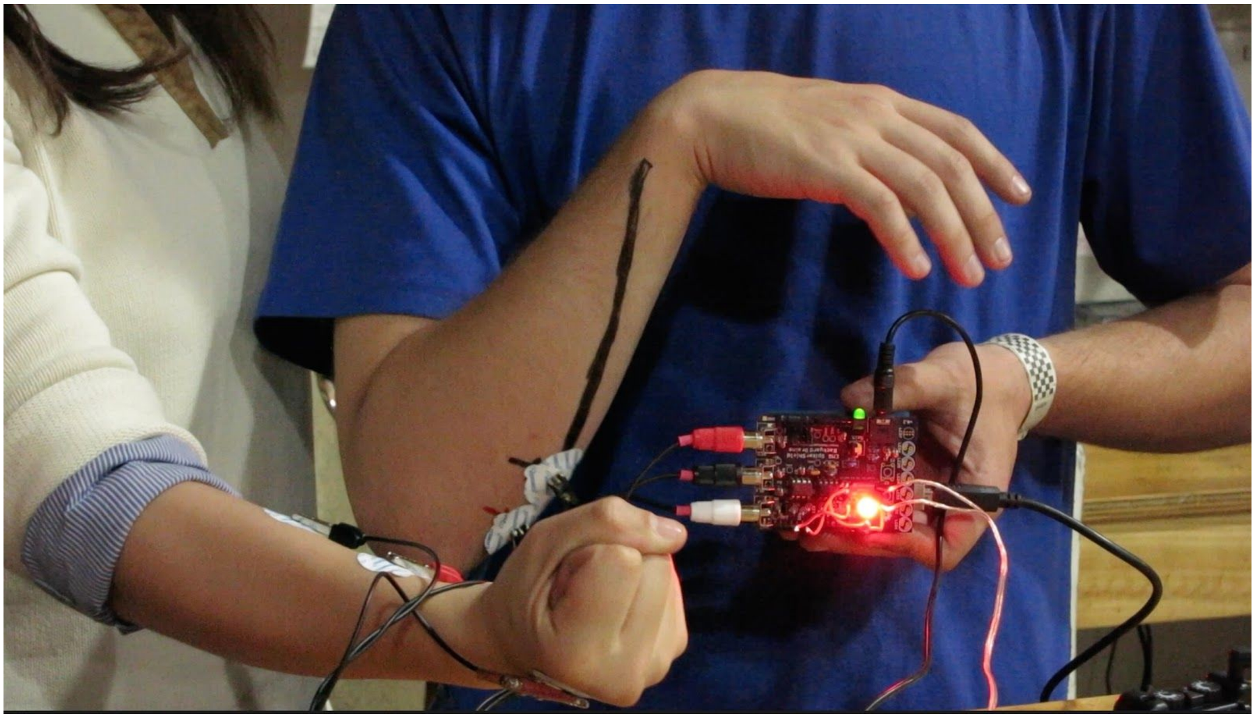
1. First, you need to set the frequency of the TENS device.



2. You can see we have the settings at 10 Hz frequency and we set a ~50-80 ms pulse duration.

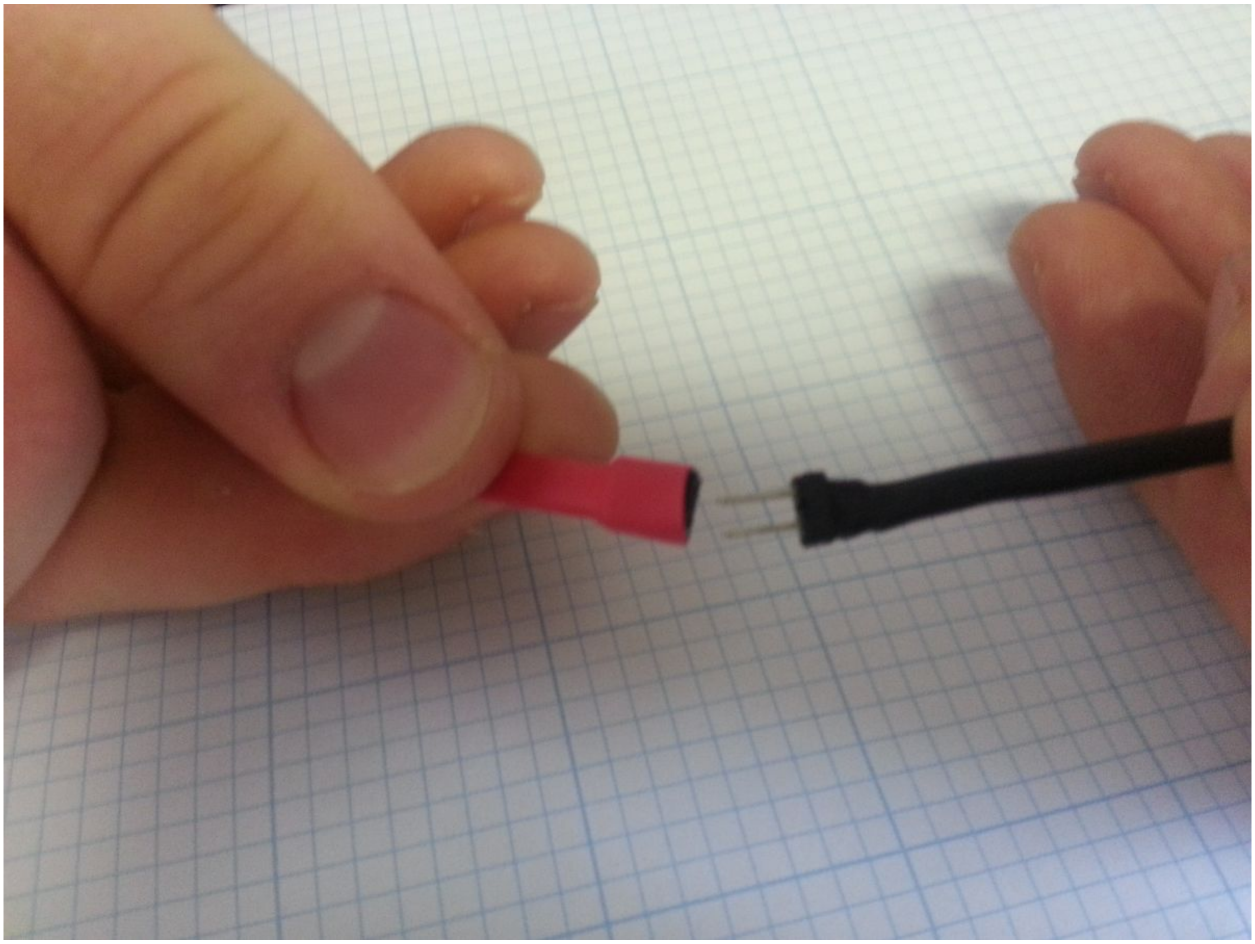
3. You will then need to find the threshold of your partner. Place two surface EMG electrodes across the ulnar nerve. This runs up the back of the forearm.



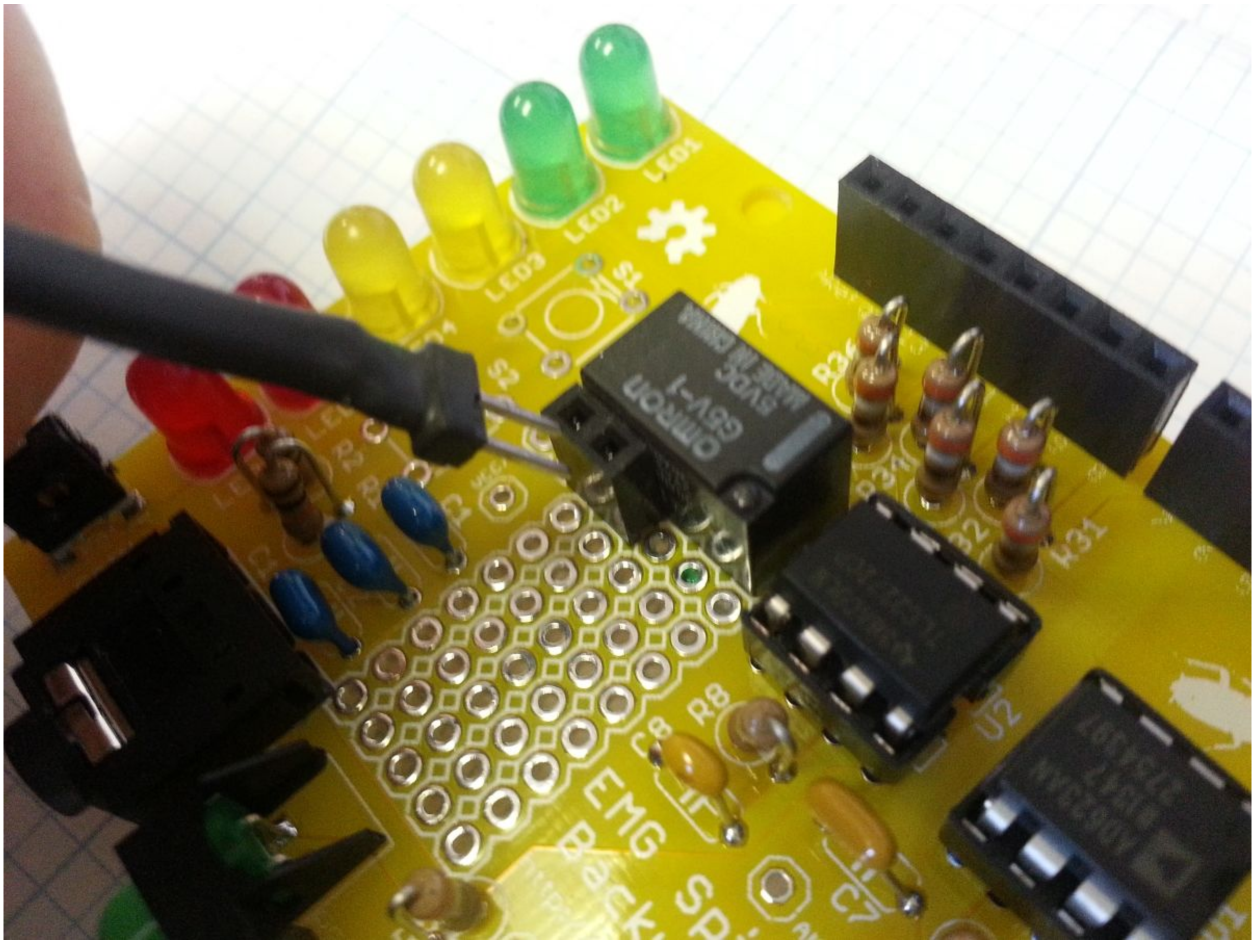


Now connect the TENS unit to the forearm with the modified TENS cable you built.

Have your partner slowly increase the amplitude of the TENS until there is a visible change of the muscle activity. You should have the male connectors connected together with the pink jumper (this uses a different kind of connector if you followed the DIY instructions). Alternatively, you could use a non-modified TENS cable for this part of the experiment. The stimulation feels like pins and needles in your muscles. It is not terribly uncomfortable but some people may not like it.



Now that the TENS' level is comfortably set, plug the Tens cables male header pins into the EMG SpikerShield Stimulator output.



And you are ready to go! Now when the "Controller" human flexes their muscles, the "Controlled" human's muscles should contract as well. Enjoy your human-human-interface!

Happy hacking! Let us know what you connect your EMG signals to! Let your creative mind flow and invent.

Troubleshooting

If your device is not working, try these solutions first!

If the leds are not lighting up at all, and/or if you only have one blinking red light, you may have accidentally reset your arduino! If this happens, reupload your code to the device. Look under the "The Code" section above for instructions.
If you are having trouble getting all the LEDs to light up with your flexing, there are two options:
1) reposition the muscle electrodes - 2) lower the thresholds in the code, again, see above in the "The Code" section for help with this.

Science Fair Project Ideas

1. This setup is only one channel, but you can stack the EMG SpikerShields to have up to three analog input channels (and our TENs unit has two output channels!). If you have two degrees of output control and three input control signals, do you think you can cause more sophisticated movements?
2. We've noticed that some people respond at lower or higher stimulation levels than others. What factors could you measure that may affect the level of someone's response to the same level of stimulation-age? Gender? How regularly they work out? If they're tired or not?
3. What other movements do you think you could induce with the Human-Human Interface? Remember to consider the earlier warnings about safe use of a TENS unit. Some suggestions-the medial nerve (another nerve in your arm), the patellar reflex (your leg moves when the center of your knee is hit), or trying to get individual fingers to move.
4. The stimulation of the muscles with the TENS, while interesting, is not nearly as elegant and smooth as the natural movements you can make with your own volitional control. Why do you think this is? How could you compare the two in an experiment? Which do you think would be stronger-your natural movements or those induced by the huge amouont of current coming from the TENS?
5. While this is a fun neuroprosthetic demo for public and classroom demonstrations, what would it take to make this a useful device for the clinic? We make tongue-in-cheek jokes about "controlling humans" here in this experiment, but what do you think it would take to fully control movements?

Acknowledgements

We heartily thank Matias Guiterrez, of the Chilean Science Company bioquimica.cl, for challenging us to build this demo for his ChileVascience outreach events we participate in.

Finalmente: Gracias to Italo Ahumada Morasky, a Chilean artist who drew these pictures for us.