Aaron Evan-Browning
Building Gadgets: Eyebrow Switch midterm project.

Background:

At the beginning of the semester, knew I wanted to work in some way with wearable devices. In the first few weeks of the semester, I became interested in the idea of making my own fabric bend sensors. I never really pulled all the necessary materials together. I still have yet to order conductive thread though I fully intend to do so. At one point, I made a canvas and conductive cloth sensor but it lacked the sensitivity I needed. I think as I work towards my final project, this will become my main focus.

Concept:

I was sitting in my room one day and decided I wanted to control things with my eyebrows. The idea is simple, put a small switch over each eyebrow and interpret their states with an Arduino. I'd seen someone else do something similar with a fabric band with conductive cloth contacts[1] but their version lacked input from the "middle eyebrow". The middle eyebrow is often used to express concern. People use it all the time without realizing it. With the addition of the middle eyebrow to the right and left eyebrows, we add a third possible digit of binary, down represents zero, up represents 1.

2 digits: 0 , 1, 11	→	3 states total.
3 digits: 000, 001, 010, 011, 100, 101, 110, 111	→	8 states total.

zero: 000 *one: 001* *two: 010* *three: 011*

four: 100 *five: 101* *six: 110* *seven: 111*

This adds a lot of possibilities to the things that can be controlled. For instance, a full scale can be played. I realize that there is a great number more actions possible by chaining numbers together but I am unclear how to do so. The software end is a good deal more flexible than the hardware end.

And so was the beginning of Eyebrow Switch.

---

1 Brauswitch: http://robertcarlsen.net/2009/02/11/thinking-physically-brauswitch-430

Materials:

When I make things, I tend to make first, think later. As a collector of many things, I already had most of my materials on-hand. I do not claim this is the best way to do things. It's just what I had on hand (with some exceptions).

- 3 3-pin switches
  - label: 0824RAA, D2MV-01L2-1C3 .1 A 125VAC .1A 30VDC
  - I salvaged these from junked printer. To work well, it must be easily actuated, the lighter the better: eyebrows aren't very strong.
- 1” Nylon webbing, ~2.5 feet.
  - This makes up the headband.
- ~3 feet 12 gauge copper wire (salvage)
  - For making switch brackets. If I had access to either a 3d printer or Shapelock plastic, I contemplated fabricating the brackets out of plastic.
- 2 1” plastic tri-glide buckles.
  - For making a sew-less connection with the webbing.
- Small pieces of neoprene
  - for holding switches.
- 5 feet (rainbow) ribbon cable, only 9 strands needed (similar to the kind used by hard disks.)
  - It simplifies the process of wiring many loose cables. I also contemplated using cat-5 cable.
- zip ties.
  - For tying everything together.
- Heat shrink tubing (variou sizes)
  - insulating and reenforcing solder joints
- 5+ feet Wire sleeving (courtesy of wire sleeving kit that came with the 3D printer)
  - It shortens after covering things.
- Strong thread (for sewing neoprene.)

Tools:

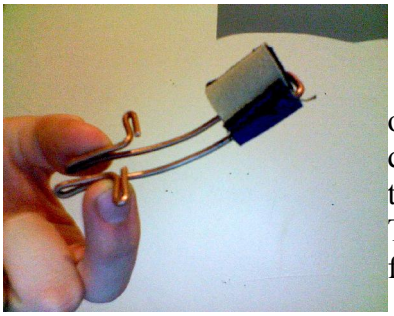
- Sewing machine (For sewing neoprene switch slings)
- soldering iron and solder.
- Scissors
- Two hefty pairs of pliers (for working with copper wire)
- Wire-strippers
- a hammer

Construction:

I didn't follow a particularly clear order of operations. I've decided the best way to show what I've made is to show the different assemblies and explain what they do:

Headband:

The headband is constructed out of nylon webbing and two plastic tri-glides. As I anticipated few other people wearing this, it takes a minute or so to adjust. The wiring goes up and over the top of the head and is fastened in back by four zip ties. It fits snugly around my head at a slightly more upward angle than a normal hat.

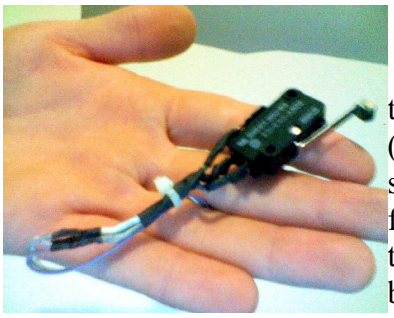


Copper bracket:

I admit this is a bit of a kludge. The switch nests between the two forks of the bracket and the neoprene sling provides enough support that it doesn't slide around too much. My sewing machine didn't like sewing through two layers of neoprene too much. There are three brackets in total. There isn't a particularly exact way of doing this. The end held in my fingers slids over the edge of the webbing.

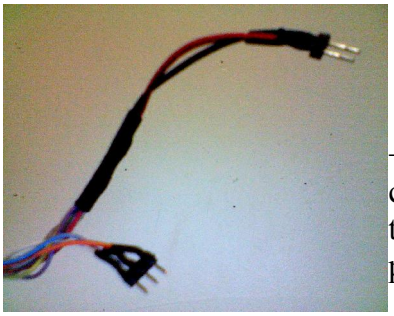
Switch-arm cover:

With the switch alone, it slides around far too much. The switch actually has a small wheel on the end which makes it slide around more. The switch-arm cover enables a much more solid connection between my eyebrow and the switch.



Switches:

Soldering and properly securing the switches took the most amount of time out of all the components. Each switch has three pins: normally open (NO), normally closed (NC) and the common pin. There are two separate soldering jobs visible. I initially attached heavier wires but afterwards found out that they were both cumbersome and ugly. To the left of the zip tie you can see my secondary soldering connections. The wire doubles back on itself to provide stability.



Wiring harness, (*Arduino end*):

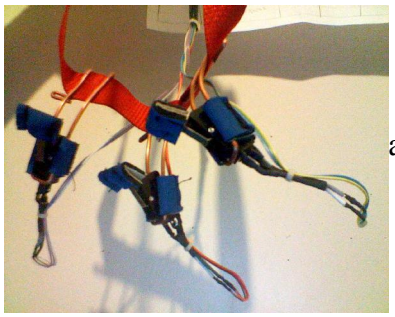
The covered wire ends in five pins: two to ground (red and black) and +5v, and three as signal pins (orange, blue and white). Each switch has a common ground and +5v pin. To simplify things, I soldered them all together. That junction is beneath the large black section of the red/black pair. Both ends terminate in header to facilitate breadboard connections.

Wiring harness (Body/end connections)

To further clean up its appearance, I sheathed the length of ribbon cable in a length of mesh wire sleeving. It is a bit of an adventure working header through the many-holed surface. I would recommend soldering header on *after* the length of wire sheathing is in place. This stuff likes to fray a whole lot. I tried using a lighter but that effort was mostly ineffective. I suppose that's what all that big heat-shrink tubing was in the package was for.

Switch/bracket-sling/switch-arm cover assembly:

It's not a very good name. I just thought this would be a useful diagram to include. I'm actually holding the switch in an upside down position. It's not a particularly elegant solution but I found it to work surprisingly well.



Full Switch/headband assembly:

The wires double back on themselves and meet together somewhere around the top of the skull.

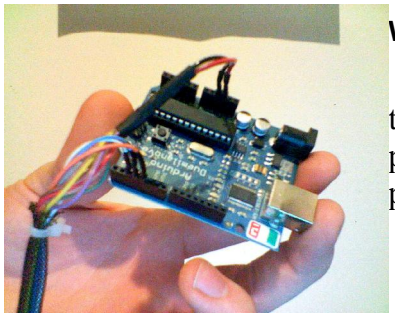
Eyebrow Switch, Fully assembled and Worn:

It's alive!

Changes between version 1 and version 2:

The initial wiring job was too heavy. In addition, the wiring on the arduino end lacked grounding, thereby picking up a hefty dose of noise. Alex helped fix this problem. I haven't had any problems with it working now that all of the connections are properly soldered and have clean inputs/outputs by using header.

Also notice how the wires are not bound together.



Wiring:

The wiring for a sketch involving the eyebrow switch is fiarly trivial. The 5v and ground pair go to +5v and GND pins. The three signal pins can be positioned anywhere that is specified by the sketch. For the purpose of several of my sketches, I used pins 2, 3 and 4.

Code:

I made several different sketches for use with my switch harness, most notably, eyebrow_servo, eyebrow_tone and eyebrow_tone2. For the purposes of saving space, I haven't pasted them inline.

I took some of my code from example sketches (http://www.arduino.cc/en/Tutorial/knob and http://www.arduino.cc/en/Reference/Tone primarily) but I think it's basic enough that I don't need to cite that I learned how to operate a servo at said website.

eyebrow_tone:

This sketch plays a note depending on which three button combination is played. When no switches are activated, it drones a middle C.

eyebrow_tone2:

This sketch plays a starting note then multiplies the frequency by a value (I used 1.1) to create a relatively smooth glissando.

eyebrow_servo:

This sketch moves a servo clockwise or counter-clockwise depending on the switch pressed:

left/right: clockwise and counter-clockwise

middle: return to center (90 degrees).

Bugs:

- As mentioned before, I initially did not ground one of the pins on each switch. This caused interference.
- If a servo goes too far in one direction it will get stuck. This is why a reset switch is required.