Final Project Report
Ishta Bhagat, Dania Azhari, Geraldine Torres
CS 362, Fall 2016
Professor Cynthia Taylor
2 December 2016

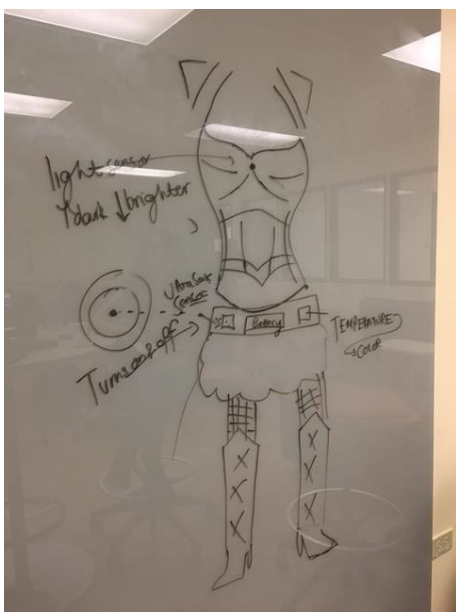
Duchess of Ether

Motivation

CS362 provides a great opportunity for students to produce a project from end to end such as working with hardware, software and producing a visual output. Arduinos are powerful and relatively easy to use hardware and can produce endless possibilities for side projects. In the past this course has been taught, other students have all made clever projects that are useful and have practicality. It was too typical CS and we wanted to be different. We decided that sometimes it is fun to make something that not efficient or practical: Just Fun. Furthermore, we were a group with three women in it so we were inspired to do something more feminine. Our initial idea was a wearable. We knew that we would something to light up and it has to be a piece of garment. We let our imagination flow. We saw pictures from the Red Carpet at the Met Gala (which had light up dresses), fantasy/sci-fi cosplay outfits and lastly, we wanted something to be fantasy but since we have wires and technology we settled on Steampunk with is a combination of Victorian Era and Industrial Fashion. Thus, our project melds these three into a wondrous wearable dress. The whimsical and technological features of this outfit is captured by lights embedded on it, which will be activated and interact with different elements of the environment, such as sound, brightness, and proximity to objects. This ambitious outfit is to show that technology can used but also can be worn.


Original Idea

Our original idea was that we would use different colors of EL wire to light up our wearable, and that the colors and pattern with which it would light up would be dependent on three environmental factors: distance (from an object), sound, and temperature. The temperature sensor will control which EL wire will light up: if the temperature sensor sensed that it is 65 degrees Fahrenheit or above, the red EL wires would be chosen to light up; otherwise, for temperatures below 65 degrees Fahrenheit, the blue EL wires will be chosen to light up. The sound sensor would have controlled how the chosen EL wire will light up: when the sound detected has a volume that goes above a certain threshold (we have not yet determined this value), then the wire will light up. Otherwise, the wire will not light up. So the EL wires will light up in harmony with the noise in the environment, producing a rhythmic effect. And the ultrasonic sensor, once close enough to an object, would have activated the entire wearable to work in the first place (if an object was far, it would be entirely off; otherwise, it would glow to the temperature color and sound in the room).



Revised Idea

Our revised idea meant changing a couple of things, once we realized what didn’t work (talked more about in the Problems section of this paper). We ended up changing our use of EL wire in the corset to an RGB LED strip. For the ultrasonic sensor, we made it glow blue when “unactivated” and an object was far from it, but made it glow red if an object was close. Also, instead of using a temperature sensor, we ended up using a photoresistor--if the room was bright, then we would make the LEDs appear more brightly, but if the room was dark, we would dim the LEDS. And rather than turn the

RGB LEDs on and off with sound, we made the strip turn rainbow when the sound was louder than a certain threshold.

RGB NeoPixel

RGB Neopixel is a AdaFruit's brand for individually-addressable RGB color pixels and strips based on the WS2812, WS2811 and SK6812 LED/drivers, using a single-wire control protocol. In other words, they are not christmas lights, but they have the spirit. They can be programmed to be some and so much cooler. They do not work on their own unless they are not programmed by a microcontroller.

Corset Construction

The corset is simply the outfit everything goes on. We are adding the three sensors, wires, and the strip to be on the corset and stay, we had to learn some skills such as sewing and not laugh while sewing. We have pricked ourselves more than someone can imagine. The wires, mini bread board, 3 Arduinos will be in the leather pack. The ultrasonic and batteries will be in the belt. The photoresistors would be in the center of the corset and away from the LED strip so it does not pick on the LED's brightness but rather the brightness of the room. Furthermore, the sound sensor will up top in the bust region so someone's voice can be picked up and adjust the LEDs based on that.

Cape with EL wire

We had a cape lying around in the WiCS Lounge and was part of Ishta's halloween costume. We could not use our EL Wires because of the voltage issue so we decided to incorporate it into this cape. Plus, we were getting good at sewing (sort of).

We sewed the lights onto the cape and used the inverter which had the control. Thus, we had another accessory to our overall outfit and presentation.

Communication of 3 arduinos

We used the I2C protocol for Arduino communication, so we had one master and two slaves. The master requested different sensor information from the two slaves, so that it could later control the RGB LED Strip that it is connected to based on that information. The master would make this request using the slave ID that each were assigned in their code and the number of bytes it wants sent back (in this case, 2 bytes for an integer, as it is stored by Arduinos. Also, because one of our Arduinos receives information from two different sensors, we have the master write an integer, 1 or 2, for that sensor, which the slave will then read and distinguish which sensor information it should send back.

The slaves remain in a constant loop and only do useful work when the master sends a command and it is received. Based on the parameters of what the master had sent, such as the slaveID and the information that the master has written, like which sensor information to get, will the appropriate slave send the correct information per that request. The communication between the master and slaves required the Wire.h library.

To set up the hardware for this, for each Arduino, we connected all of analog pins 4 to each other and all of the analog pins for 5 on a mini breadboard. We also connected all of the grounds for each Arduino to each other on the other side of that breadboard.

Sensors

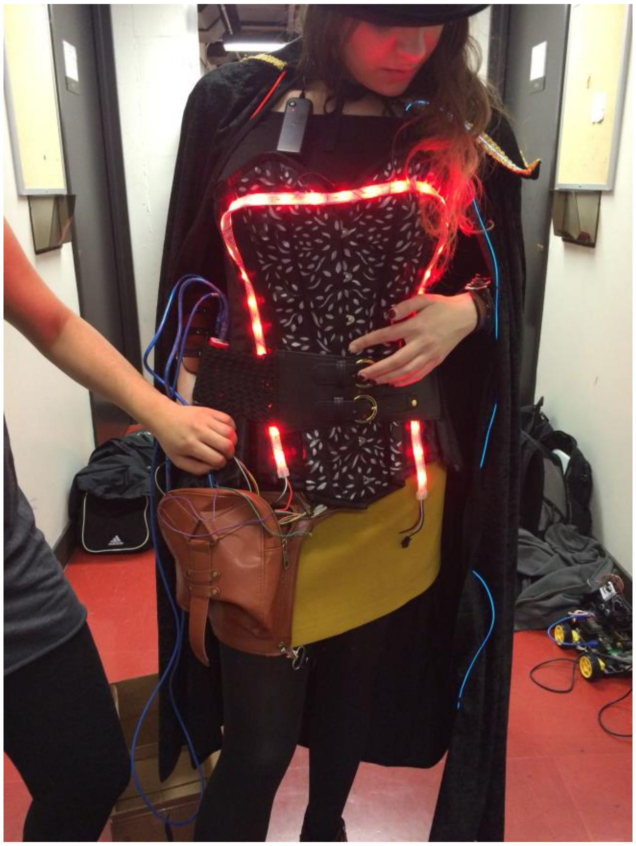
- **Ultrasonic sensor**
  The main purpose for the ultrasonic sensor was to set it as an activation for the corset. When the lights are blue, it is on off mode, and when the lights are red, it is in activation mode. The ultrasonic sensor works by calculating the distance of an object in front of it in different types of formats. For our project, we calculated the distance in centimeters. So, when sensor detects something close to it the corset will go in activation mode and turn red. Originally, we were going to make an object as a weapon, going along with the theme of our idea. We were going to use this as the object that the sensor detects to turn the corset on, but we ended up not having enough time to make it so we just used our hands to put in front of the sensor.

In this picture, we have the corset when it is in “off” mode. The sensor is placed on the black belt so that it is easily able to detect something.

Here, we have the corset going to activation mode after the sensor has detected the hand in front of it.

- **Photoresistor**
  The purpose of the photoresistor is to make the brightness on the LED strip high or low depending on how much light is in the room. When the sensor detects a high light level, the brightness goes high. And when the sensor detects a low light level, the brightness goes dim. We wanted to do something interesting with the lights other than just turning them on and off, so we thought controlling the brightness would be a cool feature. Also, the light doesn't gradually increase or decrease. It is simply either high or low.

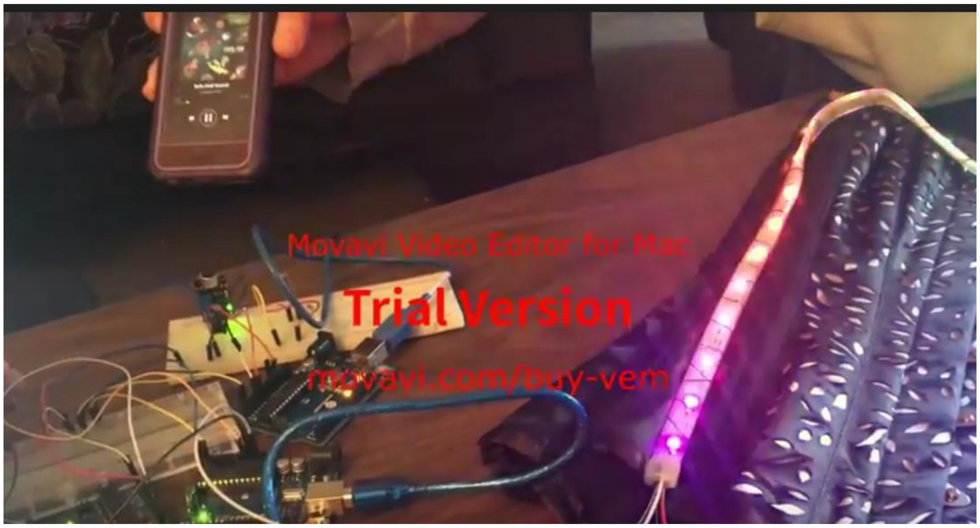
In the picture below, we have the corset when it is activated and the light level is high.



- **Sound Sensor**

The purpose of the sound sensor is to change the LED strip's lights from red to a rainbow pattern. When the sensor detects a high level of sound, the pattern goes from red to rainbow. And when the sensor detects a low level of sound, it goes back to the red lights all over the strip. We decided to go with this pattern just to add another interesting feature for changes in the lights. The lights move from one position to another causing a nice changing effect. We used music on our phones to test our sensor because people's conversations were sometimes hard to detect.

Below is the corset with the rainbow effect



The purpose of the sensors was to have them each produce a unique effect but also to work together and combine their effects to make another one. In other words, the corset has different modes depending on objects, light levels, and sounds. When the photoresistor and sound sensor both receive high values, the corset produces a bright rainbow pattern. If the photoresistor received high values and the sound sensor receives a low value, then the corset turns bright red. And so on. So there's different combinations of appearance for the corset which is what we ultimately wanted to do.

Problems

There is no journey without its problems, of which we discovered many along the way. Some of the problems we encountered are:

- Faulty temperature sensors
  So, our original plan included using a temperature sensor to choose which color we should make glow. However, we soon figured out that the temperature sensor we were using wasn't working properly: it read in values of 100 degrees Fahrenheit. And we thought that, at least if it's consistent with hot and cold values, then we might use it, but even putting it in a fridge did not affect its readings. Then, we discovered we had another temperature sensor, and alas, that one read in values of -50 degrees Fahrenheit, and it, too, showed no consistent changes between cold and warmer environments. So, we decided to use a photoresistor, which we knew works from our different projects with it, instead.
- EL Inverter and EL wire

Our original plan also included using EL inverter and EL wire. So, to connect our master Arduino to the EL wire so that it can be able to control it, we needed a relay between the EL inverter and that Arduino. However, we found out once the master sent the appropriate command and the relay had been triggered to switch, the EL inverter could turn off, like we wanted, but it could not turn back on even if the relay switched again, unless we press the button. And that voids the point of having the Arduino control anything, and since this was a problem we were having with just the way the EL inverter works, we quickly changed our idea from using EL wire to using an RGB LED strip, which we could directly control with the Arduino.

- Communicating
  Figuring out communication was a bit of a challenge, since, although we'd had to make our Arduinos communicate in a lab we had done formerly, that was done using Serial read and write, which we were familiar with. We had to learn how to communicate using three Arduinos this time, and because of what is included in the Wire.h library, it seemed easier to use the I2C bus protocol. Still, we had to learn exactly how to send the proper request to get one value for each of the multiple sensors we have, and then we had to debug things that were going wrong, such as trying to send two bytes for an integer instead of the one byte that Wire.h usually takes in, so that required some bit manipulation and masking to change the integer value we were receiving from the sensor and breaking it up into one byte each that we could then write to the master, and the master would read in each byte and, by shifting and ORing, recreate the correct 2 byte integer.

- Non-responsive sound sensor
  We had some problems with our sensor just not responding properly to sound, so figuring that out was helped by using it on its own, separate from all the other devices, and reconfiguring its sensitivity and testing out whether or not it did indeed work. Thankfully, it did, so we connected it to the slave Arduino and began testing it again, and although in loud environments it wavered a bit (sometimes when it was loud the value would still drop a bit), it still sensed all those loud noise values.

- Moving unto a smaller breadboard for less bulk and easier movement
  On Wednesday night, when we were sure that everything was working just the way we wanted, we decided that it was time to declutter our hardware so it can be more easily transportable. We decided to move everything unto one mini breadboard, and moving everything for the master and the slave connected to the ultrasonic sensor turned out to be quite easy. Then, we connected the sound sensor, and that also did not seem to cause much trouble. However, the

photoresistor proved to be quite a challenge. Because of the way the mini breadboard is, we found it difficult connecting the photoresistor because the photoresistor has two legs but requires three connections (5V, GND, and analog), and no matter how we tried to make those connections, it would not work. We even tried soldering a wire two so that one leg that would split into two (one to GND, one to analog) so that everything can be connected where it needed to be, and that didn't work. If we connected the pin connected to analog alongside all the other pins connected to GND that wouldn't work. So, what happened in the end is that we just stepped back a bit from that idea and tried to get the photoresistor to just work again on its own on another breadboard, and once that happened, we reconnected the sound sensor to that breadboard as they share the same Arduino, and we just went with that configuration.

Resources

- http://playground.arduino.cc/Learning/PhotoResistor
- https://www.sunfounder.com/learn/Sensor-Kit-v2-0-for-Arduino/lesson-35-distance-detection-by-ultrasonic-sensor-kit-v2-0-for-arduino.html
- https://www.sunfounder.com/learn/Sensor-Kit-v2-0-for-Arduino/lesson-27-sound-sensor-sensor-kit-v2-0-for-arduino.html
- http://marcoramilli.blogspot.com/2011/10/communication-between-multiple-arduino.html
- https://www.youtube.com/watch?v=yQ15Hi1E7I4