ECE 445

Spring 2018

Design Review

Conductive Fabric Gesture-Controlled Sleeve

Team 56

Mrunmayi Deshmukh, Guneev Lamba, Stephanie Wang

TA: Yamuna Phal

1 Introduction

1.1 Objective

Over the last decade, as advancements in the field of functional textiles have steadily increased, so have the number of applications available for these devices. According to leading wearable electronics researchers at The Ohio State University, these textiles are expected to play a continued and important role in areas like communication, sensing and healthcare applications (Gorder). Intelligent garments have also found a moderate degree of appeal across professional sports teams and fitness enthusiasts, often offering a combination of health monitoring and activity tracking capabilities (Holland). Few of these fitness-oriented garment devices, however, have gone beyond these use cases to offer athletes connectivity-on-the go.

Textile applications that enable music playback and integration with a smartphone benefit users who want control of their devices made far more natural. More importantly, however, they remain the first step in incorporating connectivity into everyday clothing. As such, the need for such an application forms the motivation of our senior design project. Our goal is to integrate gesture control into a fabric sleeve that can be worn by athletes and those on the move alike. This sleeve will be equipped with a capacitive touch sensor system designed on fabric using conductive thread. It will be responsible for detecting simple gestures, which in turn will be routed through an RF module to a receiver capable of performing certain actions depending on the gesture pattern.

For the purposes of this project, the external interface will be an LED array setup that will simulate the gesture performed by the user on the capacitive grid. As an added (and optional) level of complexity, we will also enable this sleeve to perform simple functions (i.e. control volume, receive calls) on a smartphone.

1.2 Background

Efforts to develop applications for smart clothing have thus far been limited. Project Jacquard, a commuter jacket designed by Google and Levi's, is perhaps the closest comparable consumer product to our design. While advertised as keeping bicyclists connected on the move, its rather large ($350) price tag has kept it from finding mainstream adoption. Our sleeve is expected to be a significantly cheaper alternative that maintains a strong degree of functionality and targets a broader set of end users.

1.3 High-Level Requirements List

- The sleeve, located on the arm, will be able to wirelessly communicate with the LED array display. Range of communication will be 0-5m.
- This product will be able to detect four different hand gestures on the sleeve: swipe up, swipe down, single tap, and double tap.
- Physical dimensions and characteristics:
  - Weight: The sleeve module must weigh less than 125 grams.
  - Size: Length of Grid - 12-16cm, Width - Less than 35 cm, variable by user

2 System Design

2.1 Block Diagram

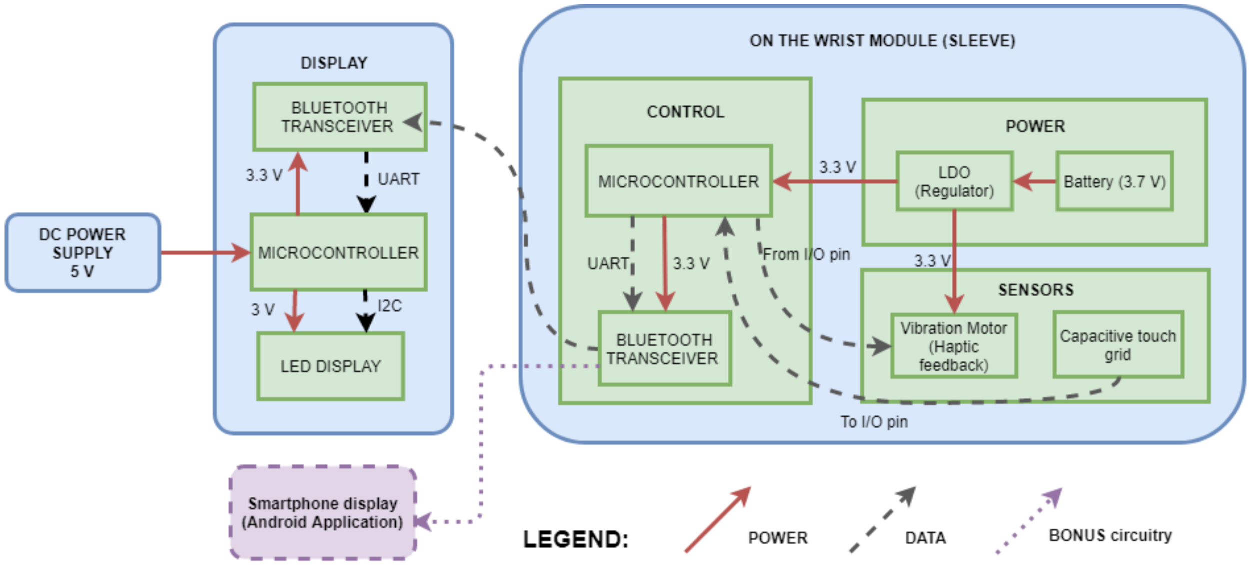

Figure 1. Block diagram overview of system.

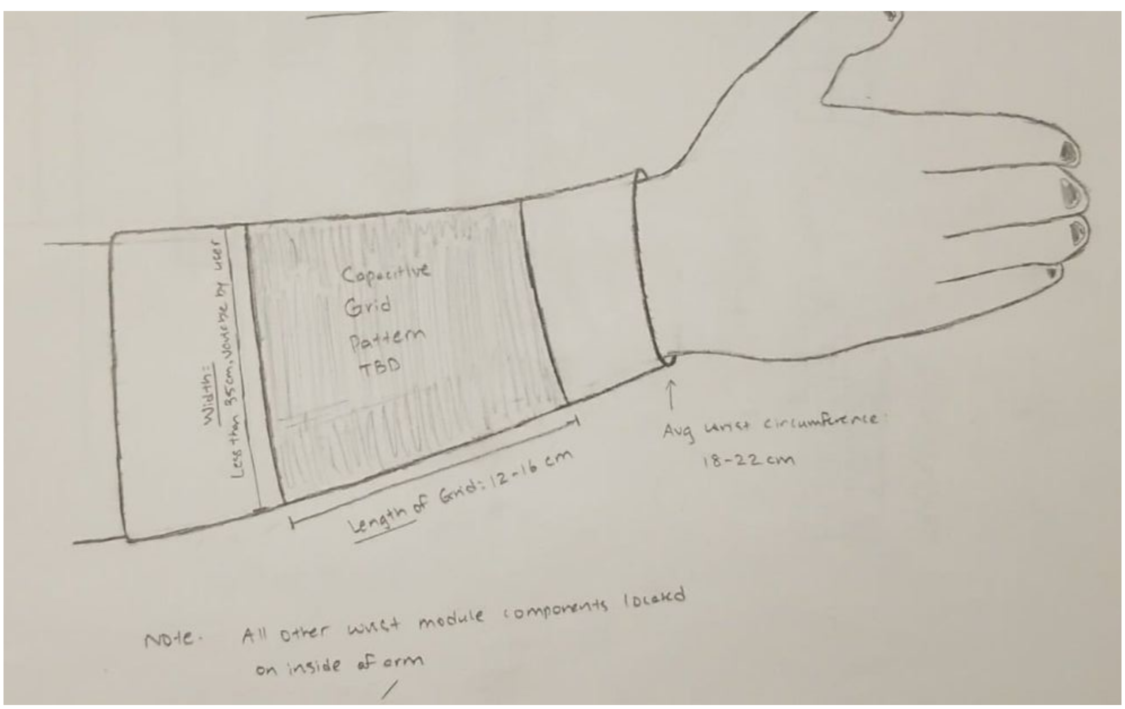

Figure 2. Proposed physical design of sleeve.

2.2 Description of Block

2.2.1 Conductive Thread Capacitive Touch Grid

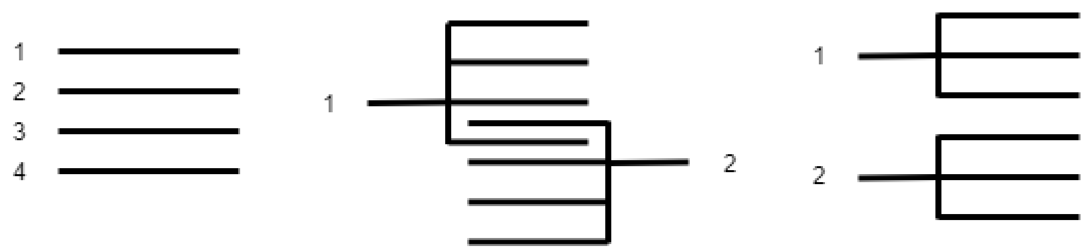

Figure 3. Potential Capacitive Grid Designs

Capacitive touch will be used to detect the gestures made by the biker on the sleeve. It will consist of a conductive thread pattern and will be powered by the microcontroller. It will be capable of detecting four gestures - swipe up, swipe down, single tap and double tap. The capacitive grid will be made of conductive thread weaved into straight lines oblique to the sleeve and will be connected directly to the microcontroller.

We plan to make use of the CapSense Arduino library to detect the capacitive touch. We will assign a single I/O pin on the microcontroller as the 'send' pin and will have multiple I/O pins for 'receive'. An internal pull-up resistor in the MCU will be activated between the send and receive pins. The value of this resistor is 20kOhms. The MCU will be programmed to read the time required to change the value of the receive pin once the send pin is set to high. This value is dependant on the RC time constant of the circuit where R is the internal pull-up resistor. When a 'receive' pin is touched by human hand, the RC time constant will change due to the introduction of a dieletric (human hand). The MCU will detect this change and indicate that the 'receive' line has been touched.

This method will be used on multiple signal lines and the MCU will be programmed to distinguish between the range of gestures (swipe up, swipe down, single tap, double tap) based on the signal lines that have been touched.

2.2.1.1 Testing data for conductive thread

The following data was measured using a Multimeter and conductive thread.

Length (cm)	Expected Resistance (Ohms)*	Experimental Resistance** (Ohms)	Difference %
5	2.625	2.3	12.38%
10	5.250	4.1	21.9%
15	7.875	5.5	30.16%
20	10.5	9.3	11.43%

*From datasheet: 0.525 ohms/cm, linear relation between internal resistance and length
**adjusted values taking into account resistance of the multimeter wires

Table 1. Test data on the resistance of the conductive thread at various lengths.

Based on the data collected, it is apparent that there are large variations in the internal resistance posed by the conductive thread, but since this resistance value is negligible compared to the pull-up resistor in the ATmega328p, we do not expect it to hamper the RC measurements.

2.2.2.2 Gesture Testing

Rigorous tests were designed and conducted to collect data in order to determine thresholds for detection between gestures. Multiple human subjects were used to provide diverse data sets. The objective was to collect data and perform time analyses on all expected interaction with the capacitive grid. The thresholds will be used to determine spacing between conductive threads and optimal RC constant.

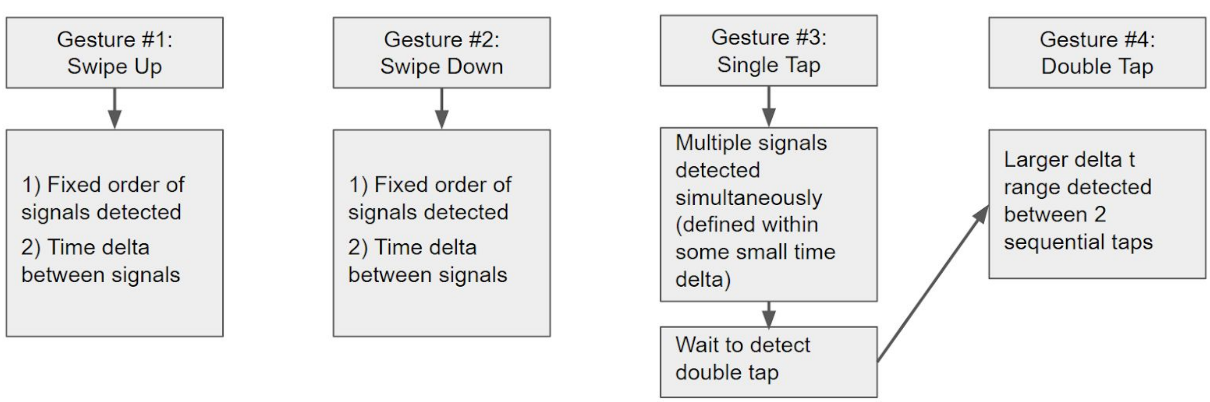

Figure 4. Gesture detection algorithm (Flowchart)

A simple timing analysis using an accelerometer tracking the length of time required to perform a hand swipe over a measured distance was performed, with the results and corresponding analysis included below (see Figure . The beginning and end times were extrapolated from data points where the acceleration values measured by the accelerometer were equal to 0, suggesting the hand had just started to accelerate (beginning of swipe) or had just come to a stop (end of swipe).

Beginning Time (s)	End Time (s)	Time Delta (End - Beginning)(s)	Length of Swipe (cm)	Average time spent per distance (s/cm)
1.578 s	2.055 s	0.472 s	15.24 cm	0.0310 s/cm
2.165 s	2.648 s	0.483 s	17.018 cm	0.0284 s/cm
2.38 s	2.868 s	0.488 s	15.24 cm	0.0320 s/cm
11.367 s	11.826 s	0.459 s	15.24 cm	0.0301 s/cm
1.042 s	1.516 s	0.474 s	15.24 cm	0.0311 s/cm

Figure 4. Timing Data for Hand Gestures (Left to Right a.k.a. Swipe Down)

Two discrete taps can be recognized as a double tap given a set threshold for the time duration between the two taps. According to literature from smartphone application development like Swift and Android, developers appear to default to 0.2 seconds between discrete taps to recognize a double tap has been performed. However, we were concerned that interacting with a phone with the touch of the thumb would be a different experience compared to making coarse taps on a wristband, especially for the target use case of people on-the-go. As a result, we performed our own tests. Based on 37 data points collected from 3 human subjects, the average duration between discrete taps for a double tap is 0.29 seconds and a range of 0.19 to 0.40 seconds.

2.2.2 Microcontrollers

For this project, we have chosen to use the low-cost Atmel Mega328P microcontroller (MCU) for its simplicity yet ability to support our desired project goals. It supports the generic driver for capacitive touch and has 23 general I/O pins that will be sufficient to communicate all input and output information from the sensors to the Bluetooth module.

There will be two MCUs located on the sleeve and LED demo submodules respectively. The MCU on the sleeve submodule will handle collecting and processing the input from the capacitive touch sensor. It will determine which gesture the user performed and what action should be performed next. Then, it will provide the output signals to the RF transmitter. A second MCU will be used to support the LED display demo. This one will collect the signal from the RF receiver. It will process it to these inputs to determine which LEDs and what color should be displayed, corresponding to the gesture made.

2.2.3 Bluetooth Module

The Bluetooth module will establish communication between the sleeve module and the LED array display module. Wireless connectivity is important for wearable technology, like this gesture-controlled sleeve, in order to communicate with other devices preserve ease of accessibility for users.

For this project, the HC-05 Bluetooth module has been chosen for several reasons. It supports both slave and master modes, which enables communication between two Arduino board for testing purposes. This Bluetooth transceiver basically acts as a generic serial COM port. The model we chose also has a breakout board that comes with some supporting circuitry including a 3.3V regulator that allows voltages between 3.6 to 6.0V for the VCC. While the board can be powered by 5V, the TX and RX pins safely operate with a 3.3V logic input. Information assigned to the separate modules using the AT command mode is included in the subsections below.

2.2.3.1 Sleeve Bluetooth Transceiver

The purpose of the sleeve Bluetooth module will be to receive data transferred serially from the sleeve MCU, with both devices embedded on the sleeve module to capture the appropriate data from the conductive thread capacitive touch grid. It will then transmit data to the demo Bluetooth module. The sleeve Bluetooth module is also responsible for collecting data transmitted by the demo Bluetooth module when the vibration motor is triggered to indicate an "incoming call." Once received by the sleeve Bluetooth, the data will be transferred serially to the sleeve MCU where it will be decoded and used to activate the current to the vibration motor. The sleeve/slave Bluetooth is marked with purple along the edge of the board.

Name	Role	Address	CMODE	Default Serial Speed
HC-05 Slave*	0 (slave)	14:3:6446e	0	38400

* Note that the sleeve Bluetooth device has been configured as the slave in order to communicate with an Android application in the case that the project is able to meet that stretch goal.

Table 2. Configuration information for the sleeve Bluetooth transceiver.

2.2.3.2 LED Demo Bluetooth Transceiver

The purpose of the LED demo Bluetooth module will be to pick up on signals sent by the corresponding transmitter module. The receiver will be connected to the demo display. For demo purposes, commands sent through to the Bluetooth receiver will be decoded by the MCU to appropriately control the LED array display.

Name	Role	Address	CMODE	Default Serial Speed
HC-05 Master	1 (master)	14:3:6434e	0	38400

Table 3. Configuration information for the master Bluetooth transceiver.

2.2.4 Alerting Vibration Motor Module

The vibration motor will be used to provide tactile feedback to alert the user that there is an incoming call for the ideal use case of this project. It will be mounted to the sleeve module and its intended location will be the underside/inside of the user's forearm so as not to be in the way of the capacitive touch grid. A coin or "pancake" motor will be used. While it may be restricted in amplitude because of its size, it will have an extremely low profile of just approximately 2.6mm.

The main limitation for driving the vibration motor with the MCU is that it requires a greater start and operating current than the output current from the MCU's pins (expected to be limited to 3.3V). As a result, it is necessary that there is a component between the MCU and the motor to safely drive the motor at its rated voltage. We will most likely be using a transistor as this component because it is simple and flexible.

2.2.5 LED Array Display

The LED block will be used to demonstrate the sensitivity of the sleeve and the state associated with the gesture pattern. A 2D array of LEDs will model the gesture performed by the user on the capacitive touch grid. For instance, swiping down on the sleeve will prompt the LED matrix to simulate colors travelling downwards, while swiping up will result in colors travelling in the opposite direction. This block will receive its inputs from the second microcontroller that is connected to the RF receiver.

The LED array we intend on using in this lab is a 1.2" Adafruit Bi-Color 8x8 Matrix, equipped with 64 Green and 64 Red LEDs. The 8x8 matrix will also be paired with a backpack (compatible PCB), that once soldered to, handles the multiplexing of the pins. Data can be transmitted to the matrix via a 2-pin I2C interface.

2.2.6 Power source

The physical requirements associated with wearable devices (thin, small and lightweight) require a sophisticated power management system with high power density batteries that can extend the device's battery life. A few ways to do this are - energy harvesting, rechargeable batteries, wireless charging and ultra-low power conversion using LDOs.

In designing the power source for the sleeve, special consideration was given to maximise the power output using a small Lithium Ion Battery (3.7 V) with an output range of 3.7 V to 4.2 V when completely charged.

Component	Part No.	Voltage (V)	Typ Current (mA)	Max Current (mA)	Power (mW)	Max Power (mW)
Bluetooth module	HC05	2.8 - 3.3	35	-	105	139.3
Vibration Motor	B1034.FL45-00-015	2.0 - 3.6	-	60	120	216
MCU	ATMega328p	3.3	1.7	2.5	5.1	7.5
LED Array	Adafruit PI:902	3		30		90
Max Voltage		3.6				
Total current				127.5		

Table 4. Power calculations for sleeve module.

In Table 2 above the following equations were used to calculate the max power consumed by each sub-module:

$$P_{max} = I_{max} * V$$

In the table, I_{typ} has been used in place of I_{max} to calculate maximum power for the Bluetooth module. Based on the current consumption, we decided to go with a 2000 mAh Li-Po battery that will provide up to 15+ hours of operation at maximum values.

$2000\ mAh\ \div\ 127.5\ mA\ =\ 15.68\ hours\ (assuming\ max\ current\ operation\ at\ all\ times)$

We will also be using an LDO by TI (TPS71933-33DRVR) to regulate the battery voltage to a steady 3.3 V. This LDO is capable of providing 200 mA output current which is more than the maximum current requirements of our chips and motor.

2.2.7 Android Interface

Currently, the Android interface module is expected to be out-of-scope of the project timeline and will be a stretch goal, as the main demo will be the LED array display. The overall interface of the Android application would be simple and minimal for the purposes of the project. It would primarily consist of a home screen with various options.

All functions would be programmed and implemented with the standard Android Studio and Android SDK and targeted for Android API Level 26 (Oreo 8.0.0).

2.3 Requirements & Verification

Component	Requirements	Verifications
Microcontrollers [8 pts]	1. Must be able to transmit/receive data from the Bluetooth module using programmable UART. **[3 pts]**	1. Verification Process for Item 1: Power cycle the Bluetooth modules and ensure that they able to transmit data wirelessly by observing the serial monitor.
	2. Must be able to operate in normal mode on 3.3 V input with +/- 15% tolerance. **[5 pts]**	
Power Supply (on sleeve) [25 pts]	1. Battery must be able to provide power for at least 15 hours. **[10 pts]**	2. Verification Process for Item 1: (a) With battery at full charge, apply a load calculated to induce current flow and measure time until current or voltage output decreases by 10%. (b) Plot the voltage over time and extrapolate to 3.7 V to obtain expected operation time.
	2. Power system size must not exceed standard sleeve size. Battery itself must not exceed 60% of total sleeve module weight (125g). **[5 pts]**	2. Verification Process for Item 2: (a) Battery dimensions will be measured using calipers and checked against standard size as defined in physical diagram (Figure 2). (b) Battery weight will be measured using a digital scale and compared against standard
	3. Voltage regulator: Must be able to provide 3.3V +/- 10% and provide rated current continuously. **[10 pts]**	3. Verification Process for Item 3: Voltage regulator output (both voltage and current) will be measured using oscilloscope for range of voltage inputs (3.7 - 4.2 V).
Bluetooth [15pt]	1. The master / slave device is properly configured. **[4pt]**	1. Verification Process for Item 1: (a) Disconnect the 5V power input to the Bluetooth module (RX pin) before powering up the Arduino board. The LED should blink at 2 second intervals to indicate it has entered AT command

		mode. (b) Confirm role, cmode, address, and bind of the modules using AT commands.
	2. Confirm that the devices are operating in the correct ISM frequency range of approximately 2.4GHz. **[4pts]**	2. Verification Process for Item 2: (a) Connect the antenna to the RF port of the network analyzer and power up the Bluetooth module. (b) Observe where the peak lies when testing the Bluetooth module on the network analyzer.
	3. Successfully pair between receiver and transmitter device and have an effective range of at least 5m. **[7pts]**	3. Verification Process for Item 3: (a) Run both Bluetooth devices through power cycle with EN (PIN34) set to low. (b) Observe that the LEDs on the devices should blink rapidly when unpaired. The devices have successfully paired when they blink rapidly twice a second.
Conductive Thread Capacitive Touch Grid [32 pts]	1. Dimension (length) of grid must be larger than standard width of four fingers. **[2 pts]**	1. Length of grid will be measured using scale and compared against standard human hand dimensions.
	2. Grid must be capable of detecting touch vs. no touch for the different signal lines with at least: 1) >50% success rate. **[20 pts]** **OR** 2) >75% success rate. **[25 pts]** **OR** 3) >85% success rate. **[30 pts]**	2. Verification Process for Item 2: (a) Using the capsense library on the Arduino, the grid lines will be individually tested. Each line will be connected to an I/O pin on the Arduino. (b) The testing will be carried out by comparing RC (time constant) values when a line is being touched vs. not being touched.
Vibration Motor [15pts]	1. Vibrations must be felt through the sleeve fabric. **[7.5pts]**	1.Verification Process for Item 1: Test the motor at different voltage levels between the operating range as

		specified on its datasheet, and record the vibration amplitude.
	2. Motor temperature does not reach 45C (where the operating temperature must be between -30-60C). **[7.5pts]**	2. Verification Process for Item 2: (a) Take a preliminary temperature reading for an untested motor. (b) Run the for at full voltage for a period of 5 minutes. (c) Take a temperature reading and ensure that this is a safe level for skin contact.
LED Array Display **[5 pts]**	1. The LED array must simulate and display each of the four gestures performed by the user on the conductive thread capacitive touch grid: swipe up, swipe down, single tap, double tap. **[5 pts]**	1. Write a basic test program that simulates all of the sketch/drawing routines in the Arduino LED library corresponding to the four gestures 2. Observe for any LED pins that do not light up or for patterns that appear to look incomplete. If this is the case, reexamine the soldered backpack for any weak connections.

Table 3. Requirements and verifications for all modules.

**Maximum possible points = 100

3 Tolerance Analysis

Critical Component: Capacitive touch grid

We expect the capacitive touch grid to be the most difficult module in this project. Based on literature and mentor advice we have determined that a robust grid design is critical to the operation of this sub-module.

Test procedure:

In order to address this, we will test out different grid designs as proposed in Figure 3. The testing will be carried out using the CapSense library on the Arduino. The testing set-up will be as described in the requirements and verifications table item 2 under Capacitive Grid.

Expected results:

We expect there to be a strict trade-off between resolution of the grid and the number of signals to be read by the microcontroller. Since we need to detect only four gestures which are relatively different from one another, we expect at least 4 unique signal lines to accurately distinguish between them. Also since capacitance values change radically based on the environment (for example, humidity - moisture content), we expect at least 50% success rate with detecting gestures.

We have also taken into consideration some possible false positives. These will be dealt with using the algorithm and for the most part do not have any effect on the physical design of the grid.

False Positive Considerations	Solution
Prolonged touch (> 1sec)	Do nothing
User brushing against lines/grid	Optimal sensitivity to be determined by next steps in testing
Grounding	Metal plate skin contact such that user acts as reference point
Environment (humidity, sweat)	**[Stretch Goal]** Implement additional sensor to collect data on environment and make capacitive sensing algorithm dynamic

Table 4. False Positives in gesture detection.

4 Cost Analysis

Part	Part Number	Unit Cost	Quantity	Total
Conductive Thread	Adafruit Conductive Thread 640	$9.90	1	$9.90
Microcontroller	Atmel ATMEGA328P-PU	$2.20	2	$4.40
Arduino Uno	-	$0 (acquired from previous personal project)	2	$0
Testing Breadboard	-	$0 (acquired from previous coursework)	2	$0

Various MST / Through-Hole Passive Components	-	$10.00	-	$10.00
Bluetooth Antennas	DSD TECH HC-05 Bluetooth Modules	$9.99	2	$19.98
LED Array	Adafruit Bicolor LED 8x8 Pixel Matrix	$21.40	1	$21.40
Battery	3.7V Li-poly batteries	$12.40	2	$24.80
Linear Regulator	TPS71933-33DRVR	$1.67	2	$3.34
Vibration Motor	Coin Type Vibration Motor B1034.FL45-00-015	$4.94	1 package / 5 motors	$5.95
Vibration Motor Driver	NPN Transistor	$0 (acquired from previous coursework)	2	$0
Total Cost				**$99.77**

Table 5. Expected component pricing information.

Labor Costs:

- 2014-2015 B.S. EE Salary Assumption: $67,000

  Hourly Wage = ($67,000 / 1 yr) * (1 yr / 52 weeks) * (1 week / 40 hours) = $32 / hr

- Implied labor costs for one group member, assuming 10 weeks of development in the semester

  ($32 / 1 hr) * (10 hr / 1 week) * (10 weeks / semester) * (2.5) = $8,000

  Total Labor Costs: $24,000

Total Development Costs = Labor Costs + Components Cost (TBD)

= $24,000 + $99.70 ~ $24,100

5 Schedule

ID	Task	Person in Charge	Date Assigned	Deadline	2/12/18	2/19/18	2/26/18	3/5/18	3/12/18
1	**Mock Design Review**		**2/12/18**	**2/19/18**					
	Sign up for MDR								
2	**Purchase/Acquire Parts**		**2/12/18**	**2/19/18**					
	Purchase conductive thread	ALL	2/12/18	2/19/18					
	Purchase insulating material								
	Purchase LED matrix	Guneev	2/12/18	2/19/18					
	Purchase RF transmitter/receiver	Guneev	2/12/18	2/19/18					
	Acquire ATmega238 MCU / Arduino	Guneev	2/12/18	2/19/18					
	Purchase vibration motor	Guneev	2/26/18	3/5/18					
	Purchase power source materials	Guneev	2/26/18	3/5/18					
	Acquire breadboard	Guneev	2/12/18	2/19/18					
3	**Capacitive Touch / Grid Design**	**ALL**							
	Dependency: conductive thread testing								
	Sewing conductive thread	Mrunmayi							
	Various pattern testing	Mrunmayi, Guneev							
	Gesture sensitivity threshold testing	Mrunmayi, Guneev							
	Insulating material								
	Waterproofing								
4	**Circuit Simulation**								
	Circuit Schematics								
5	**Component Testing**								
	Vibration motor	Steph							
	Bluetooth module	Steph							
	Capacitive grid	Mrunmayi							
	LEDs	Guneev							
6	**Module Testing**								
	LED array	Guneev							
	Power (voltage regulators)	Mrunmayi							
7	**PCB Design: 1st Round**	**Guneev**							
	Design								
	Order								

ID	Task	Person in Charge	Date Assigned	Deadline	3/26/18	4/2/18	4/9/18	4/16/18	4/23/18	4/30/18	5/7/18
8	**PCB Design: Final Round**	**Guneev**									
	Design Revisions										
	Order										
9	**Mock Demo / Final Deadlines**		**4/16/18**	**4/20/18**							
	Demonstration Sign-Ups	ALL	4/16/18	4/20/18							
	Mock Presentation Sign-Ups	ALL	4/18/18	4/24/18							
	Final Presentation Sign-Ups	ALL	4/23/18	4/27/18							
	Final Paper Due	ALL	4/23/18	5/2/18							

Table 6. Progress schedule split by team member.

6 Risk Analysis

We believe the biggest risk of our project will be the sensors block. Our design depends on the sensitivity of the capacitive touch sensor to provide accurate information to the microcontroller. Information from the touch will be input signals to the microcontroller to determine the action that should be performed next. In particular, one design challenge for the touch sensor will be to determine the adequate levels of gesture sensitivity. Different users will make swiping or tapping gestures at varying rates, so to avoid false positives, thorough testing must be performed to achieve the optimal pattern design and gesture tolerances.

As a result, our project will also be dependent on the microcontroller's clock speed. For both sensors, we will need to be mindful that they are not too sensitive, otherwise the system will perform actions when they are not needed. If they are not sensitive enough, the system will not receive enough data. Another challenge will be implementing low-level code for bluetooth module driver.

7 Ethics & Safety

As we continue to work towards finalizing the design of our sleeve, we hope to eliminate the possibility of any electric shock to a potential user. Given that the user is interfacing directly with conductive fabric and is also wearing the sleeve, we want to ensure that we incorporate some type of electrical insulating fabric to avoid direct skin to sleeve contact and afford them greater protection. One of the ways we hope to minimize this risk and also protect the sleeve from the environment is by applying a water resistant epoxy. While this is a later-stage design consideration, if pursued, we would need to consider the impact of the epoxy on the sensitivity of the conductive thread and ensure that the overall performance of the sleeve is not compromised.

8 References

"ATmega328/P Datasheet Complete." Atmel, Nov. 2016.

Capacitive Touch Sensors. Fujitsu Microelectronics Europe GmbH, 2010, *Capacitive Touch Sensors*, www.fujitsu.com/downloads/MICRO/fme/articles/fujitsu-whitepaper-capacitive-touch-sensors.pdf.

"Commuter X Jacquard." *Levi's*, www.levi.com/US/en_US/features/levi-commuter-xgoogle-jacquard/.

Gorder, Pam F. "Computers in Your Clothes? A Milestone for Wearable Electronics." *Ohio State News*, 13 Apr. 2016, news.osu.edu/news/2016/04/13/computers-in-your-clothes-a-milestone-for-wearable-electronics/.

Holland, Janet. *Wearable Technology and Mobile Innovations for next-Generation Education*. Information Science Reference, 2016.

Keim, Robert. "Circuits and Techniques for Implementing Capacitive Touch Sensing." *All About Circuits*, 30 May 2016, www.allaboutcircuits.com/technical-articles/circuits-and-techniques-for-implementing-capacitive-touch-sensing/.

Ramasamy, V. et al. "The Basics of Designing Wearable Electronics with Microcontroller." https://www.embedded.com/design/real-world-applications/4431259/The-basics-of-designing-wearable-electronics-with-microcontrollers.

Currey, Martin. "Connecting 2 Arduinos by Bluetooth Using a HC-05 and a HC-06: Easy Method Using CMODE." http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc-06-easy-method-using-cmode/.

Shuo Gao, Jackson Lai, Arokia Nathan, "Fast Readout and Low Power Consumption in Capacitive Touch Screen Panel by Downsampling", Display Technology Journal of, vol. 12, pp. 1417-1422, 2016, ISSN 1551-319X.

Humza Akhtar, Ramakrishna Kakarala, "A Methodology for Evaluating Accuracy of Capacitive Touch Sensing Grid Patterns", Journal of Display Technology, Volume: 10 Issue: 8