Conductive Fabric Gesture-Control Sleeve

ECE 445, Final Report, Spring 2018

Contributors:

Mrunmayi Deshmukh
Guneev Lamba
Stephanie Wang

TA:

Yamuna Phal

2 May 2018
Project No. 56

Abstract

This design project approached the issue of inconveniences in smartphone screen dependence by designing a wearable fabric sleeve. It wirelessly provides incoming call alert through haptic feedback and touch-sensitive gesture detection to interact with smartphones or similar devices. The touch-sensitive grid was designed using conductive thread and relied on high impedance capacitive touch detection. The control unit classified different hand gestures performed on the sensing grid by analyzing them against expected change in capacitance and timing thresholds found experimentally. The user would also receive call alerts through a vibration motor that provided haptic feedback. Gesture and incoming call data were bidirectionally communicated via Bluetooth. This system was functional in a laboratory setting and achieved >75% gesture classification accuracy. Findings from lab indicate that further work is necessary to reduce noise and interference in the capacitive touch sensing algorithm.

Table of Contents

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 ease of accessibility to athletes and everyday people who are 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

- The sleeve, located on the arm, will be able to wirelessly communicate with the phone / LED demo subsystem within 0-5m.
- This system will be able to provide incoming call alerts using haptic feedback and detect four different hand gestures on the sleeve: swipe up, swipe down, single tap, and double tap. These gestures are shown below for reference:

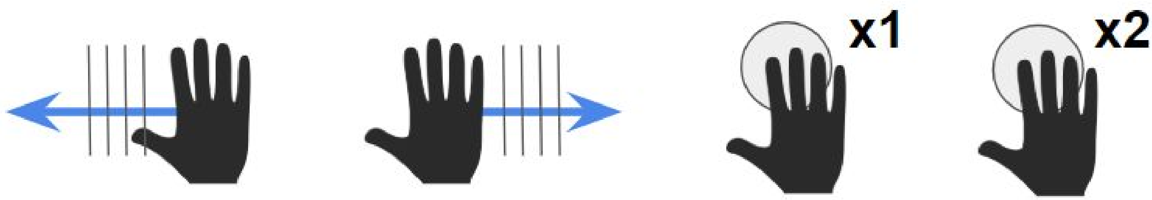

Figure 1. Types of gestures available to user.

- The sleeve must be compact and lightweight, subscribing to the following physical requirements:
    - Weight: < 125 grams.
    - Size: Length of Grid - 12-16cm; Width - Less than 35 cm, variable by user

2 System Design

2.1 Block Diagram

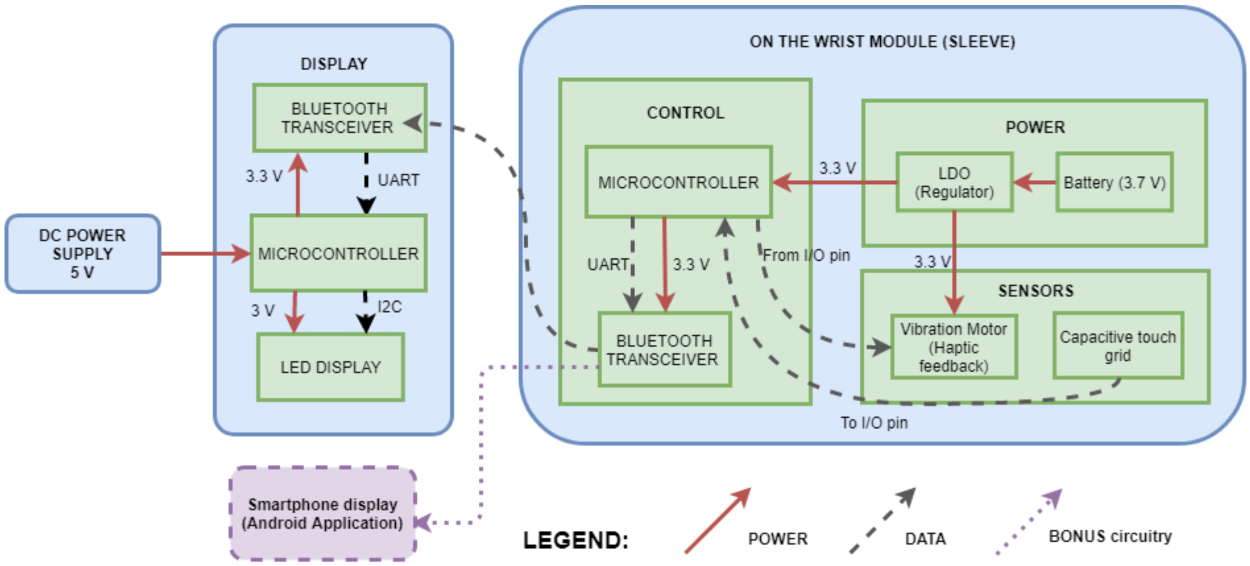

Figure 2. Block diagram overview of system.

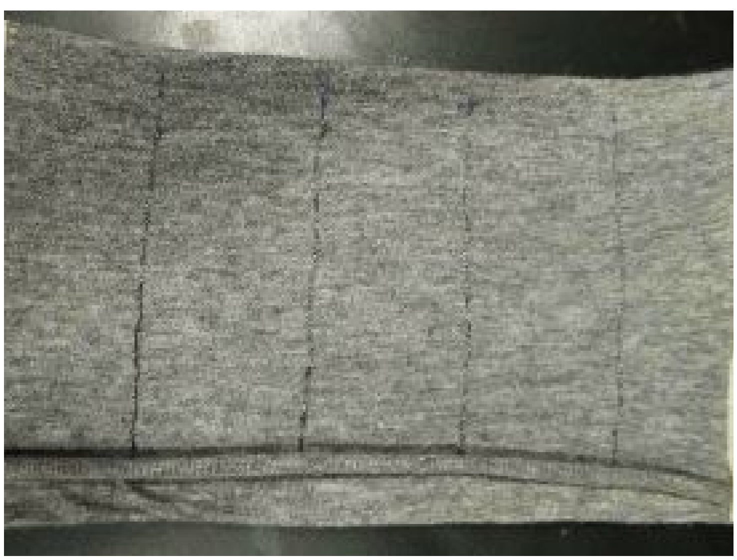

Figure 3. Physical design of sleeve.

2.2 Description of Blocks

2.2.1 Conductive Thread Capacitive Touch Grid

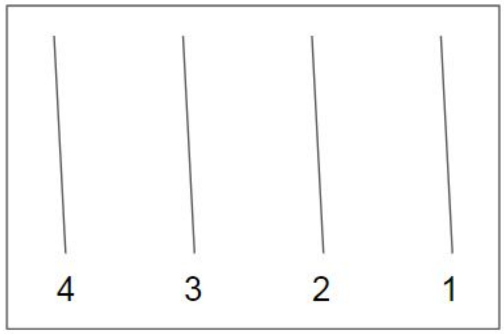

Figure 4. Mock-up of capacitive grid design.

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

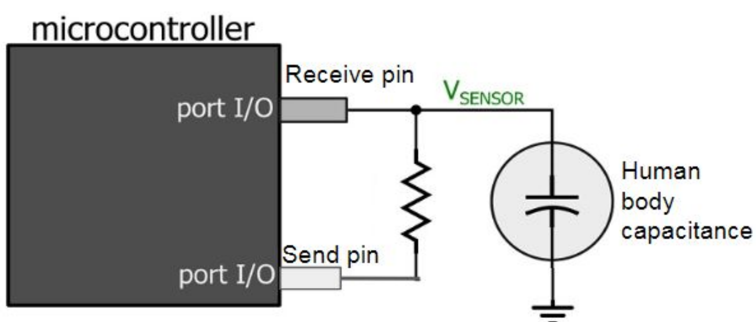

Figure 5. Capacitive touch sub-module pseudo-schematic.

The CapSense Arduino library is used to detect the capacitive touch. A single I/O pin on the microcontroller is assigned as the 'send' pin and there are multiple 'receive' pins, one for each signal line (individual line of conductive thread). External resistors are connected between the send pin and each receive pin. The value of this resistor is 8.2 MOhms.

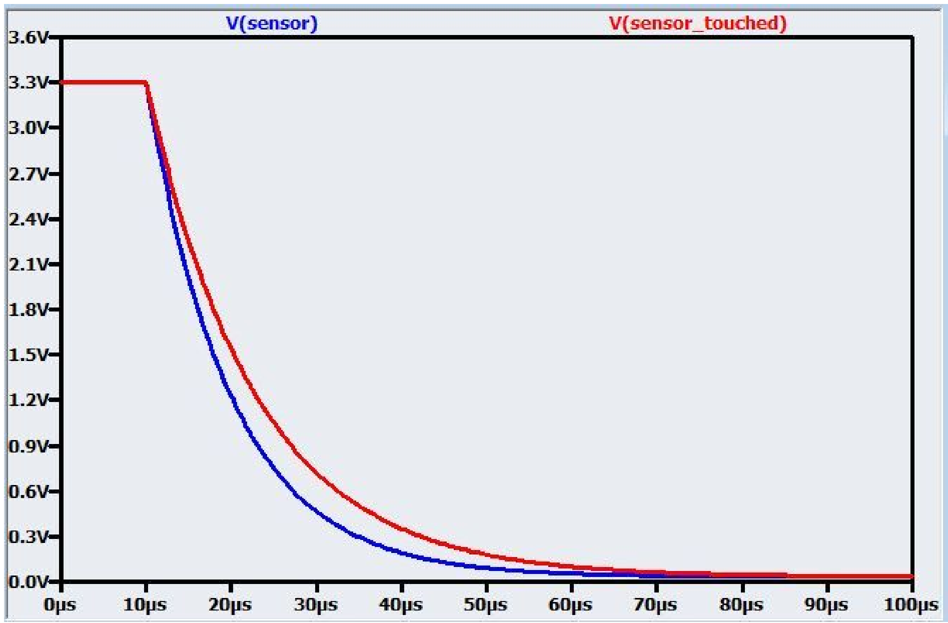

Figure 6. RC graph for receive pin showing time delta.

The MCU reads the time required to change the value of the receive pin once the send pin is set to high. This time value is dependant on the RC time constant of the circuit where R is the 8.2 MOhms 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 human body capacitance is about tens to a couple hundred picofarad. 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.

2.2.2 Microcontrollers

For this project, the low-cost Atmel Mega328P microcontroller (MCU) was chosen 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 were sufficient to communicate all input and output information from the sensors to the Bluetooth module.

There are two MCUs, located on the sleeve and LED demo subsystems respectively. The MCU on the sleeve subsystem controls the vibration motor and handles collection and processing of the input from the capacitive touch sensor. For the latter, it collects data from the conductive threads to classify which gesture has been performed and what action should be performed next. Based on the gesture detected, the MCU provides output signals to the Bluetooth device. The second MCU is used to support the LED display demo subsystem. It collects the signal from the Bluetooth receiver and determines which LED pattern should be displayed, corresponding to the gesture made.

2.2.3 Bluetooth Module

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

For this project, DCD Technologies HC-05 Bluetooth module was chosen for several reasons. It supported both slave and master modes which enabled testing between two MCUs as this project required. The HC-05 devices operate within the power requirements set for this project (3.3V) and had sufficient transmission distance (>20m, well within project requirements). This Bluetooth device basically acts as a generic serial COM port.

2.2.4 Alerting Vibration Motor Module

The vibration motor provides tactile feedback to alert the user of an incoming phone call. For our project's timeline, the vibration motor will be activated by a switch/button prompt. It is mounted to the sleeve module, located on the underside/inside of the user's forearm so as not to be in the way of the capacitive touch grid and to ensure minimal obstructions between the motor and the user's arm. Reducing the size of the vibration motor was an important consideration. A coin or "pancake" motor was chosen for this project because it had an extremely low profile of just approximately 2.6mm.

2.2.5 LED Array Display

The LED block is 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 receives 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 include: energy harvesting, rechargeable batteries, wireless charging and ultra-low power conversion using LDOs.

For the sleeve module, we used a 3.7 V, 2000mAh LiPo battery due to its high energy density and slim packaging. Along with the battery, we used a 3.3 V LDO to interface with the MCU and other sub-modules.

3 Design Procedure & Verification

3.1 Capacitive grid

3.1.1 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.

3.1.2 Gesture Timing Threshold Testing

Rigorous tests were designed and conducted to collect data in order to determine thresholds that would allow us to distinguish between different tap and swipe 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.

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).

Length	# of conductive lines	Distance b/w lines (cm)	Time b/w each line (ms)
14	2	4.7	140.014
14	3	3.5	106.015
14	**4**	**2.8**	**84.008**
14	5	2.3	70.007
14	6	2	60.006
14	7	1.8	52.505

Table 2. 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. There exists some literature from smartphone application development like Swift and Android that suggest that, on average, users take ~0.1 seconds to make contact with their phone screens and perform a single tap and ~0.2 seconds between discrete taps to perform a double tap.

We felt that we could not reliably depend on these recommended timing thresholds for taps, which, as multiple sources suggested, were also mostly up to the developers preference. The main concern was that the use case of interacting with a phone would be a different experience compared to interacting with the sleeve, especially for our target use case of people on-the-go. As a result, we performed our own tests.

The data collected on double tap time duration is summarized below:

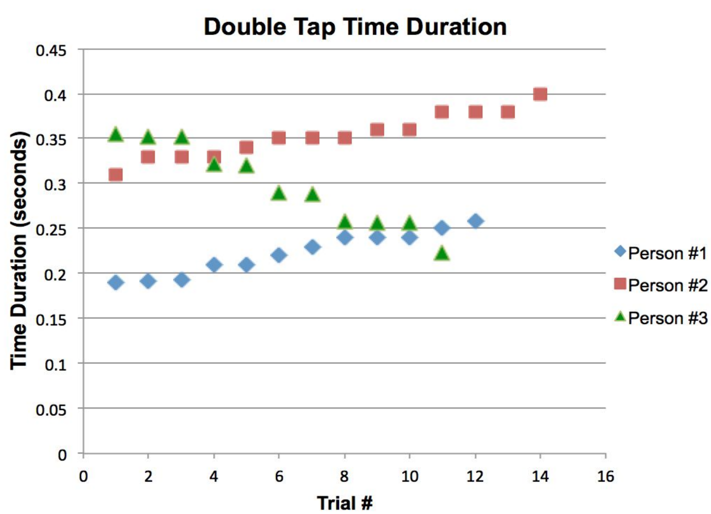

Figure 7. Data collected from tests on double tap time duration.

	Person #1	Person #2	Person #3
Average (sec)	0.223	0.354	0.298
Min time duration (sec)	0.190	0.310	0.224
Max time duration (sec)	0.259	0.400	0.355

Table 3. Average, min, and max time duration for each human subject.

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 with a tolerance of +/- 0.1 seconds. The upper bound time duration set the guidelines for the threshold to recognize a double tap gesture. As expected, the results from double tap testing are different from that recommended settings from smartphone developer documentation.

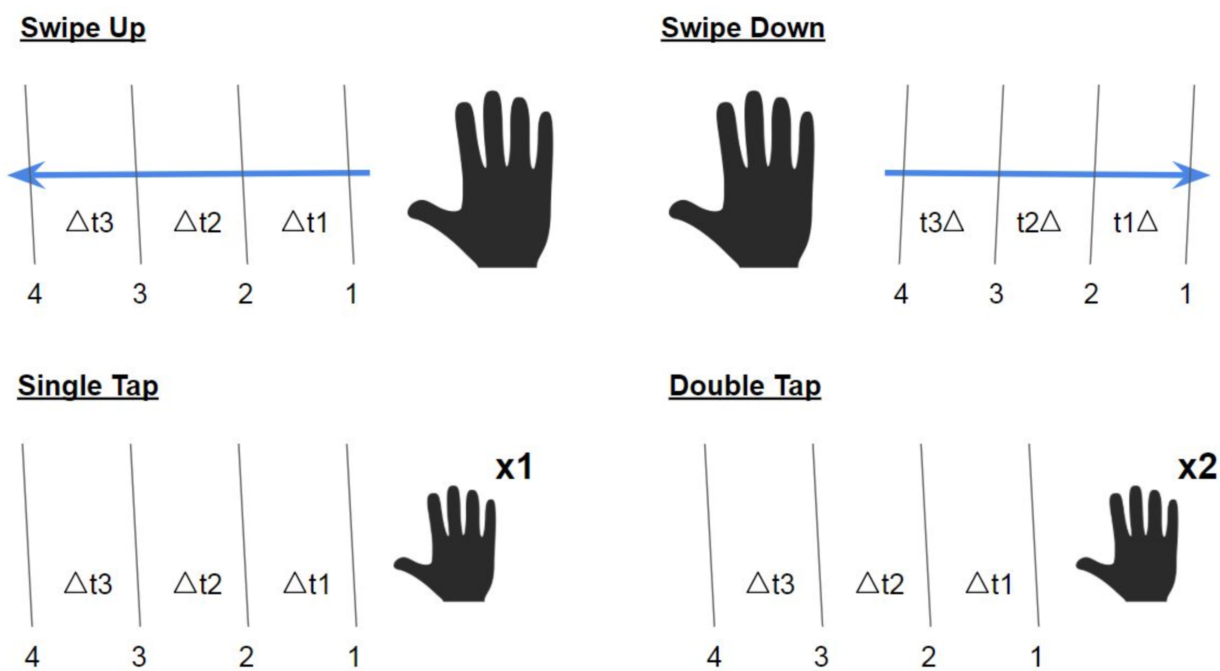

Figure 8. Time deltas used for gesture detection.

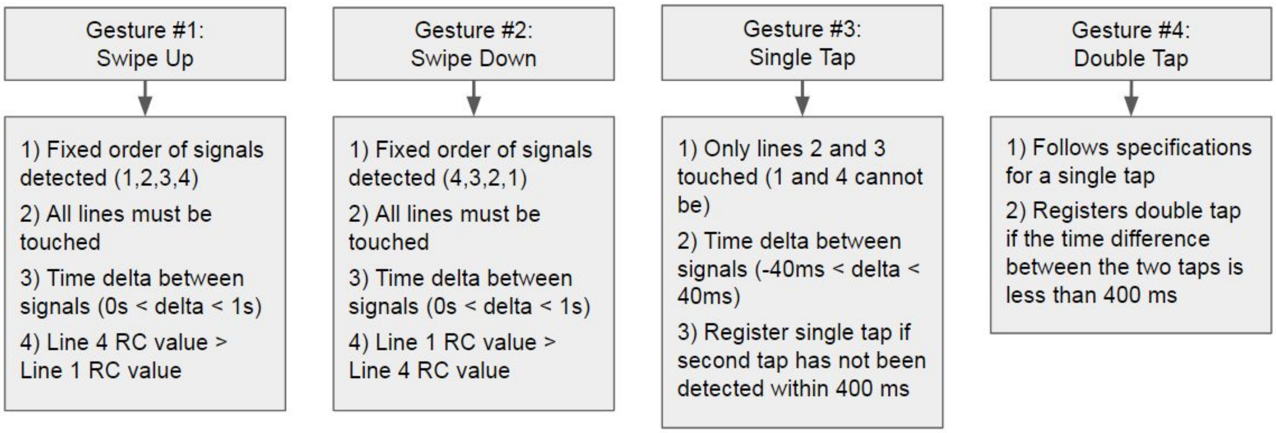

Figure 9. Gesture detection algorithm flowchart.

The difference in the charge/discharge cycles due to changing RC time constants is illustrated below using oscilloscope readings from the grid:

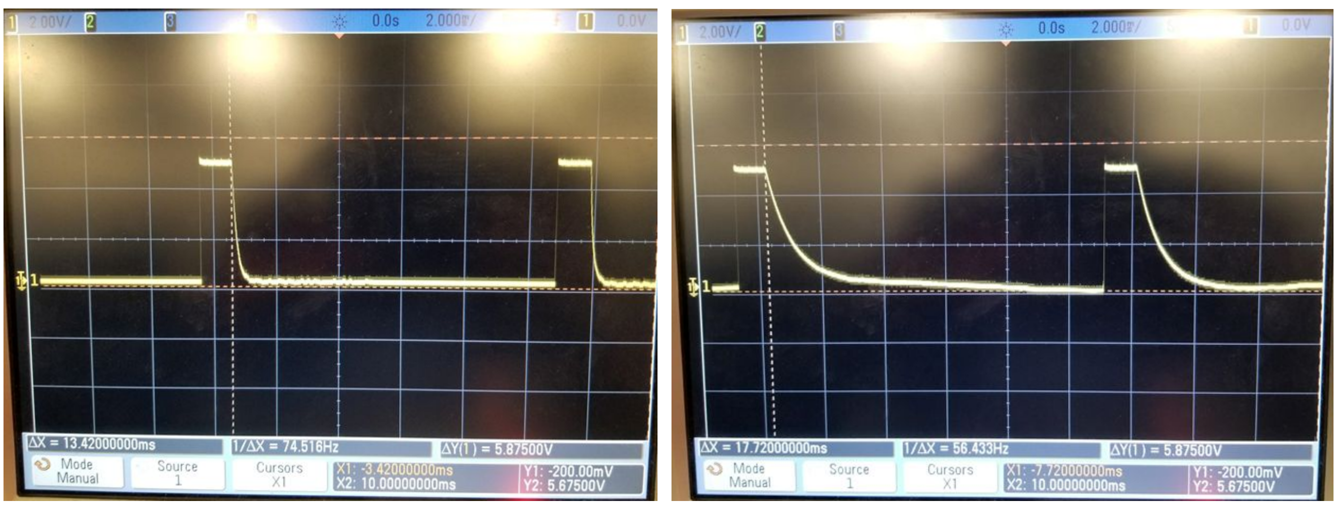

Figure 10. RC discharge curve showing time delta

In Figure 5 above, the plot on the left displays the voltage pulses on the send pin as seen at the receive pin. When the receive pin is touched, the human body capacitance increases the RC time constant of the circuit as shown in the plot on the right. The time from peak voltage to 0V discharge cycle is initially 13.42 ms. This increased to 17.72 ms once the line is touched.

We also took into consideration some possible false positives. These were dealt with using the algorithm and for the most part did 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

Table 4. False Positives in gesture detection.

3.2 Bluetooth module

For testing purposes, the Arduinos supplied 5V power. The RX pin of the HC-05 devices safely operated with a 3.3V logic input. Some documentation seemed to suggest that the RX pins could also operate at 5V, however when tested, the HC-05 device began smoking. In response, a voltage divider built from 1x 1k Ohm and 1x 2k Ohm resistors to drop 5V to 3.3V and offer the board more protection when connected to 5V power supply during testing.

The configuration information for each device is shown below:

Name	Role	Address	CMODE	Default Serial Speed
HC-05 Slave	0 (slave)	14:3:6446e	0	38400
HC-05 Master	1 (master)	14:3:6434e	0	38400

Table 5. Configuration information for the master and slave Bluetooth devices.

3.3 Vibration motor

Since this project was limited by battery size and power, the main limitation for driving the vibration motor with the MCU was that it required a high start and operating current. As a result, it was necessary to place a component between the MCU and the motor to safely drive the motor at its rated voltage. An NMOS transistor was chosen as this component because it was simple and flexible.

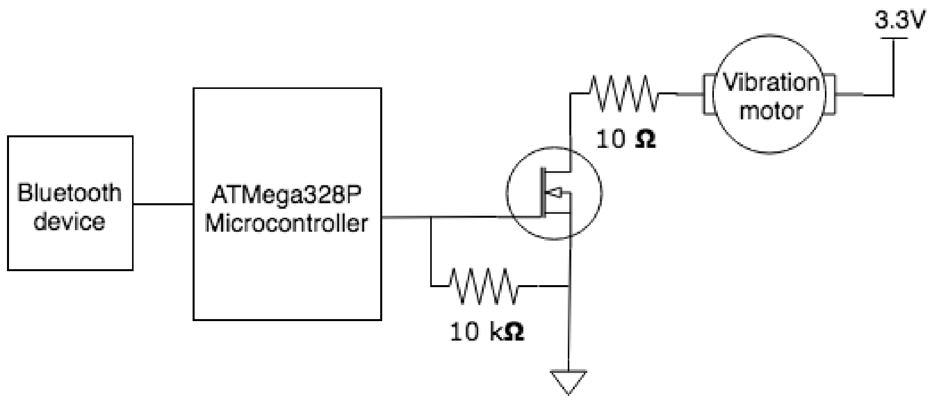

Figure 11. Motor control circuit.

3.4 Power system

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. LiPo batteries have a very low internal discharge rate, which makes them perfect for low-power consuming applications that need to run for a long time.

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
Total peak current				127.5		

Table 6. Power calculations for sleeve module.

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

$$P_{max} = I_{max} * V \qquad (1)$$

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.

$$2000\ mAh \div 127.5\ mA = 15.68\ hours\ (assuming\ max\ current\ operation\ at\ all\ times) \qquad (2)$$

We used an LDO (linear dropout regulator) by Texas Instruments (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.

Using a multimeter, we observed the voltage output of the LiPo battery to be 3.82 V. Over time this would drop to the critical voltage level of 3V when discharged.

The voltage regulator output was also tested over a range of input voltages and it consistently provided 3.28V with a tolerance of +-5%. The MCU (ATMega328p) was capable of operating normally at this voltage level.

3.5 LED Array

We wrote four routines to demonstrate the functionality of each gesture performed. To do so, we leveraged an existing Arduino library for the Max7219 LED Driver and assigned gesture values (indexed from 1-4) to each routine. To verify that each routine was working as intended, we manually sent signals corresponding to particular gestures from the BT slave device to the master.

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	2	$0
Testing Breadboard	-	$0	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	2	$0
Total Cost				**$99.77**

Table 7. Component pricing information.

Summary:

- 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 Conclusion

5.1 Accomplishments

Through the course of this project, we successfully designed and built a sleeve with a capacitive touch sensing grid and a vibration alert system for incoming calls. The sleeve module wirelessly communicated with the demo subsystem. Also the demo LED array was able to distinctly display all four gestures. With our gesture detection algorithm, we were able to achieve an accuracy of greater than 75%. We were also able to verify the PCB designs for both subsystems.

5.2 Challenges

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

There is 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 decided to use 4 unique signal lines to accurately distinguish between them. However, during testing we ran into multiple issues with grounding and noise in the PCB.

When the bluetooth module was integrated with the CapSense module, we noticed a sudden jump in the RC values of the grid. This was due to the serial protocol library that we were using for communication with the HC-05 - softwareserial. This protocol disabled interrupts for any time when a character was being sent or received. Once the bluetooth was integrated with the rest of the system, it sent garbage values to the receiver as well as added noise to the RC input data to the on-sleeve MCU.

We switched over to the AltSoftSerial library in order to resolve this issue. Even with this library we noticed heightened ambient RC values - about 100 to 200 instead of the 10 to 60 which were observed with capacitive grid operating on its own. We had to change the threshold RC value in the CapSense algorithm accordingly

5.3 Ethical considerations

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.

5.4 Future work

While we accomplished majority of our goals through the project, there are still a few things we would like to improve before this product can be sent to market. Being a wearable device for general consumers, we would like to focus on the 'usability' aspect by improving the physical design of the sleeve. In order to do this, we would improve the hard-to-soft (fabric) connections, reduce the size of the on-sleeve PCB module and reduce the size of the capacitive sensing grid itself.

Next, to improve the accuracy of the gesture detection algorithm we plan on using a dynamic model which recalculates the threshold value based on previous values. We would also want to include a re-charging circuit for the LiPo battery to make it viable for long-term consumer use. Last but not least, we would build the interface to wirelessly communicate with an Android phone and establish control of different functions on the phone using the CapSense grid.

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.

Choudhury, S. et. al. "Calibration of Sensors Using Arduino." Department of Electronics and Communication, NIT Rourkela. November 2015.

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

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/.

"Detecting Common Gestures." Android Developers Documentation. Retrieved March 11, 2018. Available at: https://developer.android.com/training/gestures/detector.html.

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/.

"Handling Tap Gestures." Apple Developer Documentation. Retrieved March 11, 2018. Available at: https://developer.apple.com/documentation/uikit/touches_presses_and_gestures/handling_uikit_gestures/handling_tap_gestures.

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

"How to Drive a Vibration Motor with Arduino and Genuino." Precision Micro Drives Tech Blog. Retrieved March 6, 2018. Available at: https://www.precisionmicrodrives.com/tech-blog/2016/05/16/how-drive-vibration-motor-arduino-and-genuino.

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

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/.

Mathavan, Hariharan. “Control an Arduino with Bluetooth.” All About Circuits Projects. Retrieved February 25, 2018. Available at: https://www.allaboutcircuits.com/projects/control-an-arduino-using-your-phone/.

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.

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.

“Transistor Motor Control.” Arduino Tutorial. Retrieved March 7, 2018. Available at: https://www.arduino.cc/en/Tutorial/TransistorMotorControl.

“UITapGestureRecognizer - Single Tap and Double Tap.” Stackoverflow. Retreived March 12, 2018. Available at: https://stackoverflow.com/questions/8876202/uitapgesturerecognizer-single-tap-and-double-tap

Wu, F., et. al. “Development of a Wearable-Sensor-Based Fall Detection System.” International Journal of Telemedicine and Applications, Vol. 2015. Hindawi Publishing Corp.

Appendix

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 8. Requirements and verifications for all modules.

**Maximum possible points = 100

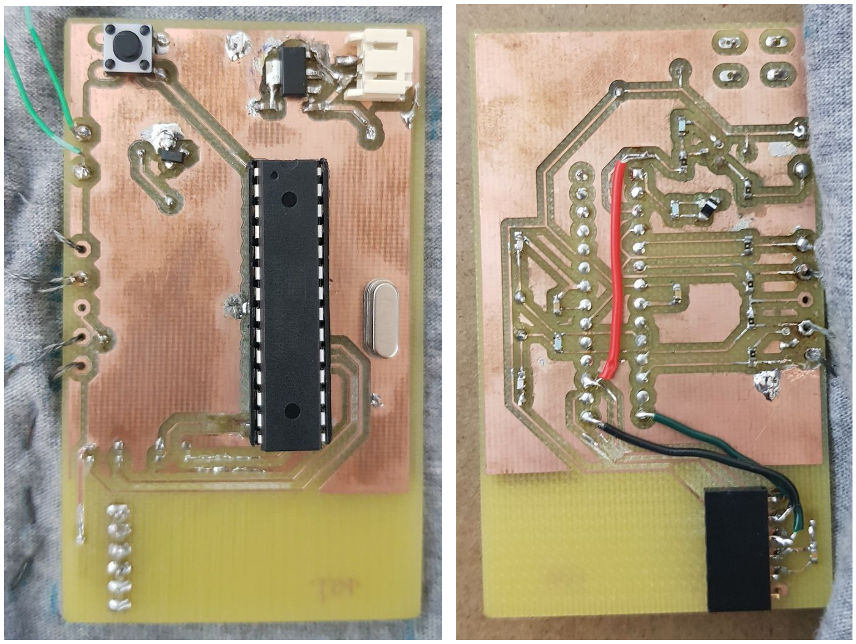

Figure 12. Sleeve PCB (from left to right: top view, bottom view)

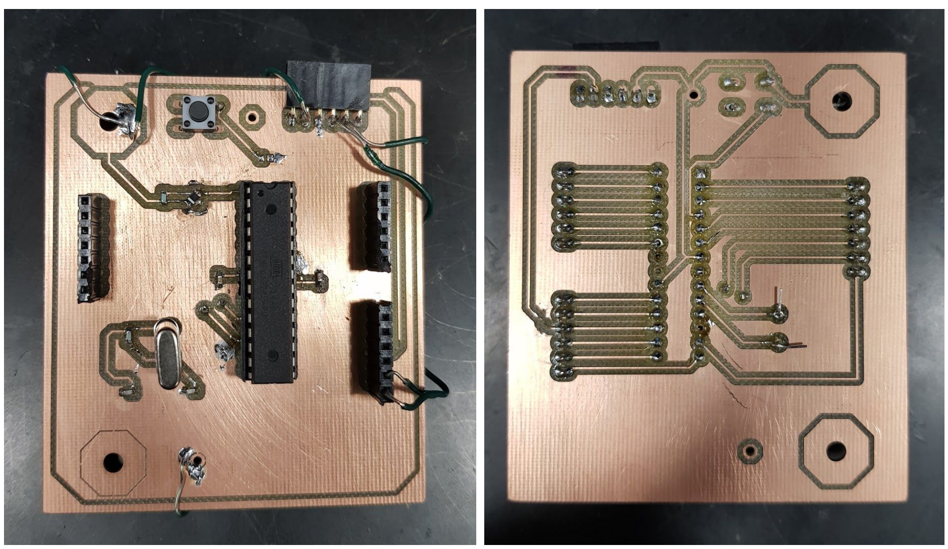

Figure 13. Demo PCB (from left to right: top view, bottom view)

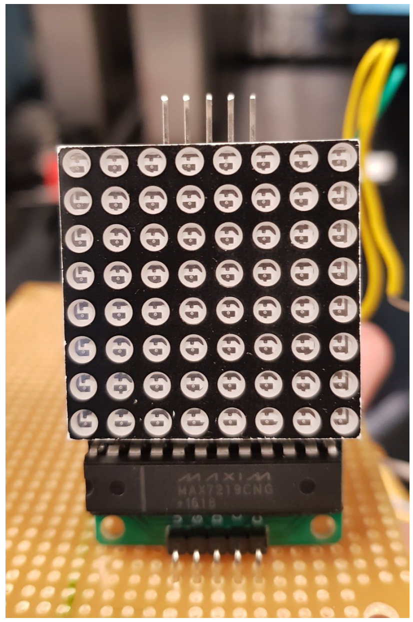

Figure 14. LED matrix with driver circuit

Figure 15. Wearable sleeve physical design (sewable conductive snaps).

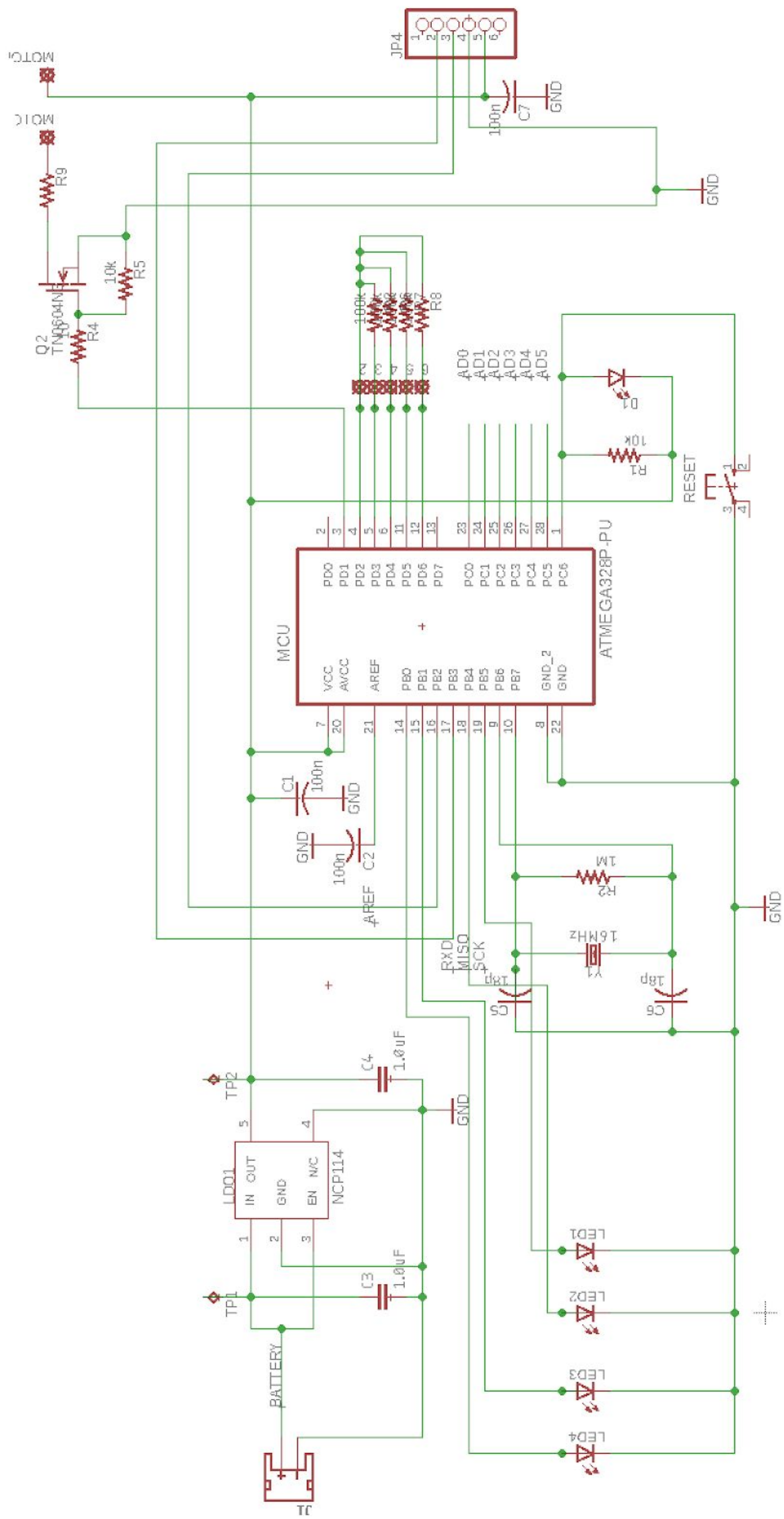

Figure 16. Full schematic of sleeve PCB.