Assistive Technology for Patients with Medical Face Blindness

A Device to Assist Social Interaction for Individuals with Prosopagnosia

Project 10 - Final Report
Timothy Chia - tchia2
Jeffrey Luan - jrluan2

ECE 445 Final Report
TA: Yuchen He
Date: May 2nd 2018

Abstract

This final report discusses our project group's attempt at developing and prototyping a wearable, minimally obtrusive, portable, system aimed at assisting social interactions for individuals suffering from prosopagnosia, or partial face blindness. Our finished system consists of three modules; a phone module, a camera module, and a wristwatch module. The user can request pictures from a Bluetooth-enabled camera at the push of the wristwatch module's button, utilize facial recognition APIs to determine matches found in a phone's picture database, alert the user upon the discovery of a match (or the lack thereof) and if a match is not found, prompt the user to add the new face into the phone's photo database.

Table of Contents

1. Introduction

1.1 Objective

Prosopagnosia [1] is a cognitive disorder which prevents afflicted individuals from recognizing familiar individuals upon viewing their face. For this reason, individuals with prosopagnosia may experience difficulty with social interaction, particularly in their relationships and careers. According to Bournemouth University, up to 1.5 million individuals in the UK alone may suffer from developmental prosopagnosia [1]; as such, we feel that designing a simple, minimally intrusive, device to help facilitate social interactions is a worthwhile expenditure.

Our proposed solution is a low cost wristband and a wearable camera that will communicate with a phone app to identify the nearest face, and display the correct name on a small wristband screen. By taking advantage of how common smartphones have become, we can minimize the cost to the end user and provide a familiar interface. At the same time, it allows us to offload expensive computation to the cloud, using cellular data, which should increase the solution's battery life.

There are currently no similar solutions on the commercial market. Previous efforts have been made to solve the problem, but typically relied on obtrusive eyewear designs which may not be suitable for users that desire more privacy. [10] Our application is unique in attempting a solution with minimal cost for patients, focusing on unobtrusiveness, and existing widespread smart phone technology.

In order to handle such a complicated application scenario, we make several assumptions. We assume that the picture is taken in daylight, with exactly one person standing significantly closer to the camera than any other people in the room, and that the person is facing the camera head on. Research has been done in finding ways to solve some of the problems that releasing those constraints would create. [11] [12] These solutions are algorithmic in nature, rather than hardware based, and as such lie outside the primary focus of our project. If the hardware we design manages to meet specifications, then it will be a pure programming problem to release these assumptions.

The following chapters will provide an overview of the hardware and software design we implemented, discuss important design decisions, unexpected difficulties, and future work to be done.

1.2 Subsystems

Our project contains three major subsystem blocks:

Our camera module block performs the task of taking subject pictures. The Bluetooth module in our camera block receives communication from the Android phone when a photo is requested from the user; in addition, it transfers any captured photos back to the Android device for further API processing.

Our wristband module block allows for the user to prompt the camera module into taking a photo, via Bluetooth communication. In addition, our wristband module alerts the user via a OLED display if a match is found.

Finally, our phone application module block provides a usable graphical user interface that allows the user to maintain an image database containing images and details of familiar faces. In addition, our phone module receives images captured from the camera device; in the special case that a new face not found in the database is detected in a camera-provided photo, the user will allow the new face to be placed in the database, if so desired.

1.3 Project Goals

Most importantly, our device must allow for the networking of a smartphone, a digital wristband, and a wearable camera. Many of the features of the individual devices in our system require the output of another device in order to proceed further; for example, our Android application requires the Bluetooth-enabled camera to take a picture before processing with a facial recognition API, and our Bluetooth-enabled camera must be prompted by our wristwatch button before it can take a picture.

Secondly, provided with a facial recognition API, our phone must be able to specifically focus on and identify the largest face in captured image. Most importantly, the employed API must be able to both recognize the presence of faces in an image, as well as identify matches between faces found in an image and faces in a user-provided database. As an additional note, in social situations where a user of our device may be surrounded by a large amount of individuals, but only need to identify a single face, it is important that our user is able to identify the correct individual.

Finally, our phone application must provide the user an intuitive UI to manage the database of known faces, as well as refining what information should be associated with each face. Examples of pertinent information include an individual’s names, classes, and mutual friends.

2. Design

2.1 Design Procedure

In designing hardware to meet our high level goal, we made several decisions between alternatives. Previous attempts to solve this problem have focused on eyeglass mounted systems to capture images and display information. We decided this form factor was too disruptive to the user for our requirements, and switched to a small camera that could be worn on a necklace. This meant that names had to be delivered on another module, which made the most sense as a wristband mounted module.

To achieve our goal of low-cost hardware, we opted to make the hardware as minimal as possible. Doing the image processing and API calls on an android phone was the result of that decision. We also considered using a higher resolution camera to increase the effective range of the design, but this raised the cost significantly, while also increasing our range beyond our range requirement.

Portability and battery life dictated much of our power circuit design. In order to use small Li-Po batteries, we had to design boost converters into the circuit to get voltages high enough for other components. An alternative design could have prioritized using low-voltage components, but we decided this would excessively restrict our component selection, and that input voltage should not be the highest priority criteria for other components such as the camera.

A final design consideration was given to the development time of the project as a whole. Without sacrificing our other considerations, we gave extra consideration to components and strategies well documented by other developers, minimizing the learning curve whenever possible, as our team was only two people with no experience with these particular technologies.

2.2 Design Phases

In accordance with the limited development time and resources, we used a modular design strategy, focusing on creating a design which could be implemented on breadboard and Arduino before PCB. The battery circuit and regulators were fully independent of the other hardware. Our software design was made general enough to be tested on a computer before Android. The Android app itself was designed to function independently of hardware modules for testing.
Then the implementation phase itself was divided as such:

1. Recognition and database capabilities on computer
2. Hardware component functionality achieved on breadboard
3. Android app functional without hardware
4. Android app communicating with hardware
5. Android app fully functional with hardware
6. Android app fully functional with PCB hardware

2.3 Hardware Design Details

The following section dives further into the details of our hardware, while reserving excessively detailed data or schematics for Appendix A.

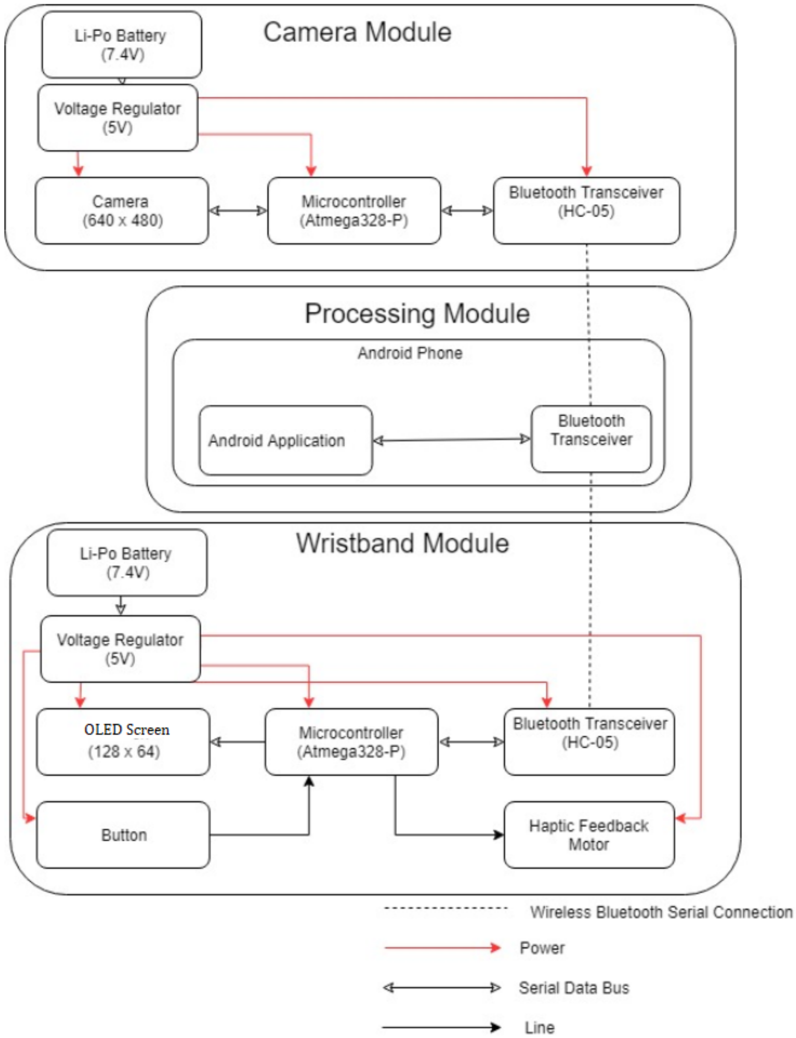

Figure 2.1: Block Diagram

The camera module is mounted on the center of a durable, semi-rigid plastic neck strap, as seen in Figure 1. Inside the camera module, the camera, PCB, and rechargeable battery are contained in a compact case.

Figure 2.2: Front and side depiction of the neck-mounted camera device. Internal breakdown of the components and their locations inside our camera housing.

The wristband is modeled after existing smartwatches on the market (ex: Apple Watch). The screen and adjacent button are located at the top of a durable aluminium case containing the Bluetooth chip, transceiver, PCB, and rechargeable battery.

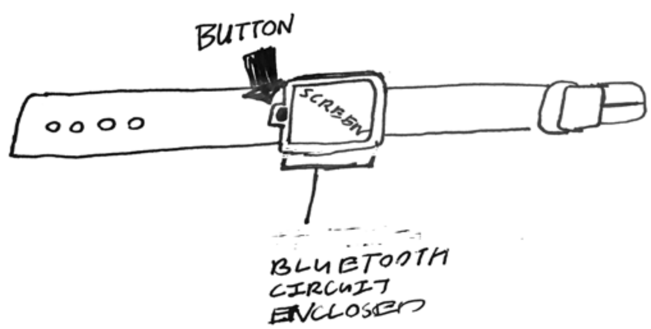

Figure 2.3: Depiction of our smart watch device

The physical designs shown above were of low priority in our design efforts, due to the time constraints. Our first round of PCBs was intentionally designed to be larger than suitable for the above designs, allowing us room to correct errors and comfortably verify the design.

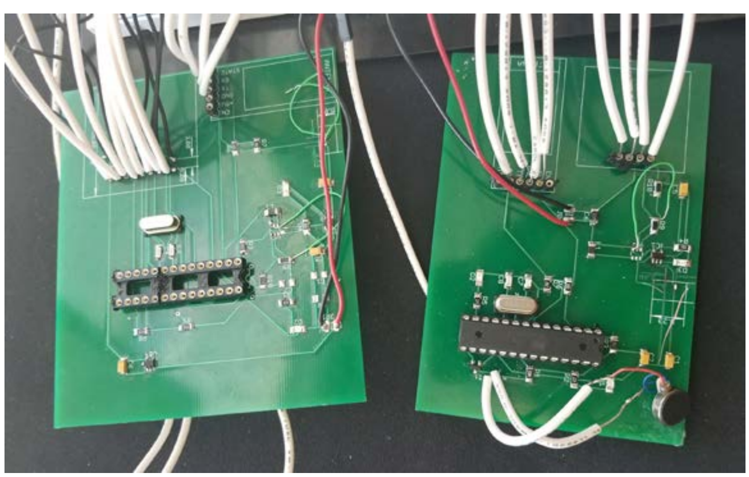

Figure 2.4: First Round PCBs for camera (left) and wristband (right) modules.

2.3.1 Camera Module

The high level requirements for this module were battery powered operation, and the ability to capture and transmit photos via bluetooth upon request by the phone module.

Bluetooth Module

The bluetooth module serves as a serial link to the phone, whose own bluetooth chip will operate as the master. We chose to use an HC-05 Bluetooth module featuring Bluetooth 2.0, as it is a widely supported Bluetooth standard. It communicates with the microcontroller via serial. In the future, we intend on exploring Bluetooth Low Energy (BLE) technology, as it has additional power-saving features that are of interest to our project.

Camera

We have chosen to use the OV7670 camera, due to its small size and sensitivity features, such as low-light capability[15]. Although the resolution is relatively low, we feel that the advantages of this camera outweigh this disadvantage. Perhaps surprisingly, in research by Tomasz Marciniak [6], facial recognition algorithms are relatively unaffected by lowered resolutions, with them concluding that a 21x21 pixel face is still recognizable. The camera outputs synchronization signals on several pins, as well as 8 simultaneous data outputs, and requires IIC communication for initialization.

Microcontroller

It’s primary purpose is to function as a logical bridge between the Processing Module and the Camera, routing wake up commands and image data. Since the requirements were fairly basic, we selected the Atmega328-P, which is an extremely common chip with excellent documentation and libraries. It also supports serial, SPI, and IIC. The additional flexibility of SPI support came in useful later, when we considered revising the design to add SPI based SRAM to the module.

Power Circuit

We decided to use a Lithium Ion Polymer (Li-Po) battery for a primary reason; improved form factor. Because Li-Po batteries can be made considerably thinner and in a greater variety of form factors than standard lithium ion (Li-Ion) batteries [7], they are more suited to our wearable devices, which are non-intrusive.

We used the Adafruit 258 3.7V battery. The two primary considerations here were given to battery capacity and physical size. By estimating current consumption by our other components over 12 hours, we determined our capacity requirement. We were then faced with a choice between several different output voltage batteries, from which we selected the physically smallest. Obtaining the needed voltages was then done with an MCP1640 boost regulator for 5V, in parallel with an TCR2EF33LM 3.3V linear regulator. Selection of the 3.3 V regulator required careful attention to the dropout voltage, which determines the minimum input voltage to the regulator. This particular regulator has a 0.2V dropout voltage, which allowed the module to operate until the battery voltage decreases below 3.5V. Finally, we added a MAX1555 IC Li-Po charging IC to allow for easy charging with a 5V USB wall adapter.

Wristband Module

We implemented this module with a second PCB with a battery subsystem. The main system consisted of a Bluetooth module, a microcontroller, a button, an LCD screen, and a motor for haptic feedback.

Bluetooth Module

We chose an identical HC-05 to the camera module bluetooth module. See above.

Button

Our button was simple surface-mount tactile switch located adjacent to our watch's LCD screen. When the user desires to obtain facial identification, the button will be pressed by the user in order to awaken the camera.

LCD Screen

We chose the HTDS-DI96 screen due to its use of OLED technology to save power and small form factor. Our design considerations for this unit were that it must display the name of the identified face, on a single line for most cases, along with using minimal microcontroller pins if possible. This screen utilized an IIC interface, which meant we could easily use other pins on the microcontroller for additional wristband peripherals such as the haptic feedback.

A width of 15 characters would be sufficient for this. A character can be legibly displayed in an 8x8 pixel square, which implies a screen width of 120 pixels. Assuming a square screen in the style of many commercial smartwatches, this would allow for 15 lines of 15 characters. This would let the user receive keywords aside from name, such as “ECE 445” or “Sarah's Friend”.

Microcontroller

Our requirements for the wristband microcontroller turned out to be almost identical to the camera microcontroller. Thus to simplify the design, we selected the ATMEGA328P here as well. The wristband Bluetooth microcontroller, upon being activated by the user via a button press, will prompt the Bluetooth chip to communicate with the camera. Our wristband microcontroller will also receive communication from the Android phone upon successful analysis between a captured photo and the phone's database of photos.

Haptic Feedback Motor

For our haptic feedback component of our wristwatch, we intend on using a single coin vibration motor; this type of haptic motor is chosen due to its compact size and lack of external moving parts. This motor will provide different series of vibrations depending on if an image match in the Android application database was found. It will buzz once if the face was identified, twice if not. We used the ROB-08449 motor, controlled by a transistor and the microcontroller.

Power Circuit (Wristband)

The power circuit for the wristband was developed to be almost identical to the camera power circuit discussed above. However the wristband did not need a 3.3V power supply, so we removed the additional linear regulator. We also used a smaller capacity battery based on our current calculations for this module. A 3.7V 350 mAh battery: Adafruit 2750.

Processing Module

Our processing module was hosted on an Android mobile phone compatible with Bluetooth 2.0 standards. This Android mobile phone will run an Android application of our own design continuously in the background. The phone we used during testing was a Samsung Galaxy S8. Given additional resources, we would have liked to test it out on several other android phones, however the platform is designed to be hardware independent, and almost all android phones meet our Bluetooth requirement.

A detailed discussion, including requirements/verification, of the software is in section 2.6.

2.4 Schematics

See Appendix B for schematics

2.5 Board Layout

See Appendix C for layouts.

2.6 Software

The software side of this design consisted of an Android based phone app. It handles image processing and facial detection/recognition using the Kairos API framework. High level goals were to recognize faces in a photo, to enroll faces and otherwise manage the database, and to facilitate all communication between the Wristband and Camera modules using bluetooth communication. The software flowchart provided below provides the most concise overview of our design.

Implementation of this software proved surprisingly challenging due to the team’s lack of prior experience with Android development. The final implementation utilizes a multithreaded approach to communicate with both modules, reserving the main thread for user input and image processing.

Figure 2.5 Breadboard Phase Phone and Camera Integration.

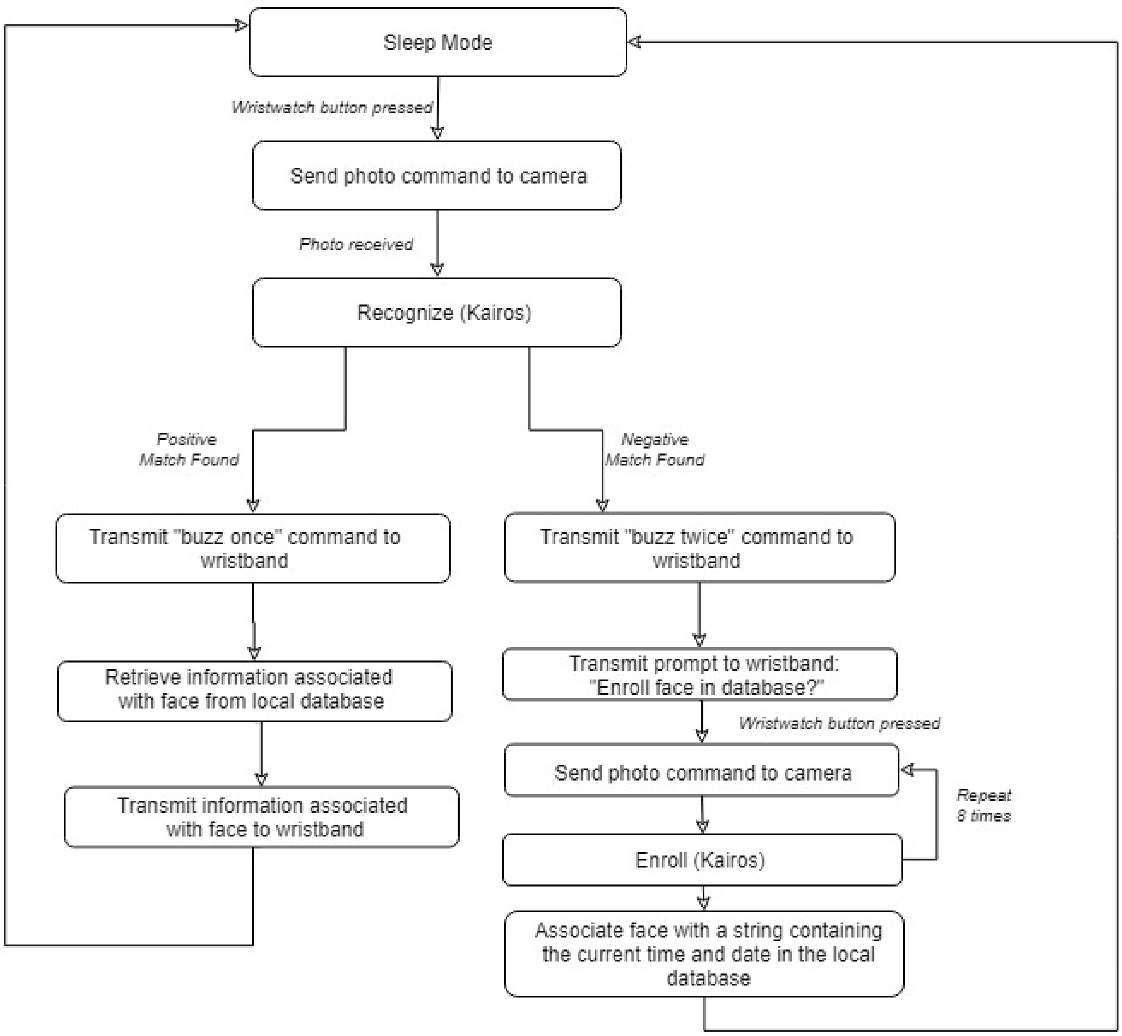

Figure 2.6 Android App Flowchart

3. Design Verification

A complete description of all verifications is provided in appendix A. This section will focus chiefly on the verifications specific to our implementation and goals, as well as those which proved unexpectedly challenging.

3.1 Facial Recognition Verification

Since the hardware in this project essentially serves to enable wireless communication between a camera, a screen, a motor, and a button, the tolerance most critical to overall project functionality is actually in software. We have chosen to discuss the false positive rate for the facial identification algorithms.

This is because it is critical for the user to trust any positive matches given by the system. If the system misidentifies the user's conversation partner, significant communication difficulties may ensue, or the user safety may even be compromised due to the user applying inappropriate precautions to the social situation. On the other hand, negative effects to the user due to false negatives are less impactful. Even for people without prosopagnosia, it is not uncommon to forget someone's name occasionally.

We needed the algorithm to perform with a false positive rate less than 10%. The tolerance for this was 5%, with a false positive rate exceeding 15% considered a failed design.
In order to understand how to minimize the false positive rate, we must give an overview of the API, and it will be integrated into our own software.

The Kairos API we have selected is a product sold by a for-profit company, which means that statistics and implementation details are not exposed to us. However their public specifications [13] indicate several requirements to achieve the accuracy they strive for. They strive for 100% accuracy in facial detection and recognition, which would be within our own tolerances if their API does achieve this. The requirements they give are a minimum of 64 pixels between the eyes, a minimum head to image width ratio of 1:8. Each user in the database should also have a minimum of 8 images enrolled, with faces closest to the center of the scene and facing the camera head on. Maximum distance is dependent on the eye gap pixel requirement, which is thus a function of the image sensor and the lens focal length. The Kairos API will return a confidence value between 0 and 1, when a face is submitted for identification. Our own app will then determine the threshold at which we classify the face as recognized.

As per our goal of bringing false positive percentages under 10%, we experimentally determined an appropriate threshold using testing and training sets of images captured through our camera module. This testing and verification method will simulate the actual use of the entire hardware combination, which is a necessity as accuracy percentages on the testing sets will be heavily influenced by the quality of the training set. Though it may be beneficial to overall performance to allow the user to upload higher quality photos as a training set, it is better optimal to set the threshold based on the average use case.

As per the Kairos Best Use guide [13] each unique face will have 8 photos in the training set. We will store 20 unique faces in the database, then create a testing set of 40 unique faces that include the 20 known faces. By running the confidence values through hard linear thresholds also in the range [0,1], we can determine the false positive and false negative percentages at each threshold. We will then select the highest possible threshold, as arbitrarily decreasing the threshold will cause false negatives to increase as well. In the extreme case of a 0 threshold, the false negative rate would be as high as 50%, even though the false positive rate would be 0%. Where else with a threshold of 1, the false positive rate would 50% and the false negative rate would be 0%. And in both cases, the overall accuracy would be 50%. Thus our threshold value will be selected to minimize the false positive rate to as close to 10% as possible, assuming our data does indicate an overall accuracy drop associated with lower false positive rates. Testing set faces are constrained to be from a well lit room, from head on, and at typical conversational distances.

We wrote a set of python scripts that execute the method described in section 2. Due to time constraints, the graphs shown below were generated using a training set of 7 individuals with 14 images each. Photos were taken using the OV7670 connected to an Arduino with frames captured using a Java program using a method implemented by [3]. Photos were taken at a distance of 1.25 meters following Kairos best practice guidelines [4]. Within the 14 photos, several contained significant visual artifacts that would cause them to be rejected by the API upon an enrollment attempt. Manually removing low quality training images by visual inspection proved to have no effect on the results, so we chose to leave them in, as we are currently unable to guarantee our full implementation will be free of motion blur or over exposure. The testing set was 13 individuals with 1 image each, including all individuals also belonging to the training set. Data points for the graph were then generated by storing the false positive, false negative, true positive, true negative, proportions in a CSV file as we varied the threshold value from [0,1.1]. Statistics are a proportion of the testing set, thus ranging from [0,1].

By graphing all four curves against the threshold value, we calculated the receiver operating characteristic curve below. The blue line is the false positive value we prioritized minimizing. It is clear that minimization occurs at all thresholds greater than 0.6. However the false negative curve increases if the threshold is increased arbitrarily. This tradeoff yields an optimal threshold range of 0.60 to 0.65, during which total accuracy is 1.0. With an increased training and testing set, the threshold optimal range should shrink, eventually converging on a specific threshold value.

Photos of the training sets and examples of acceptable photo quality are given in Appendix A.

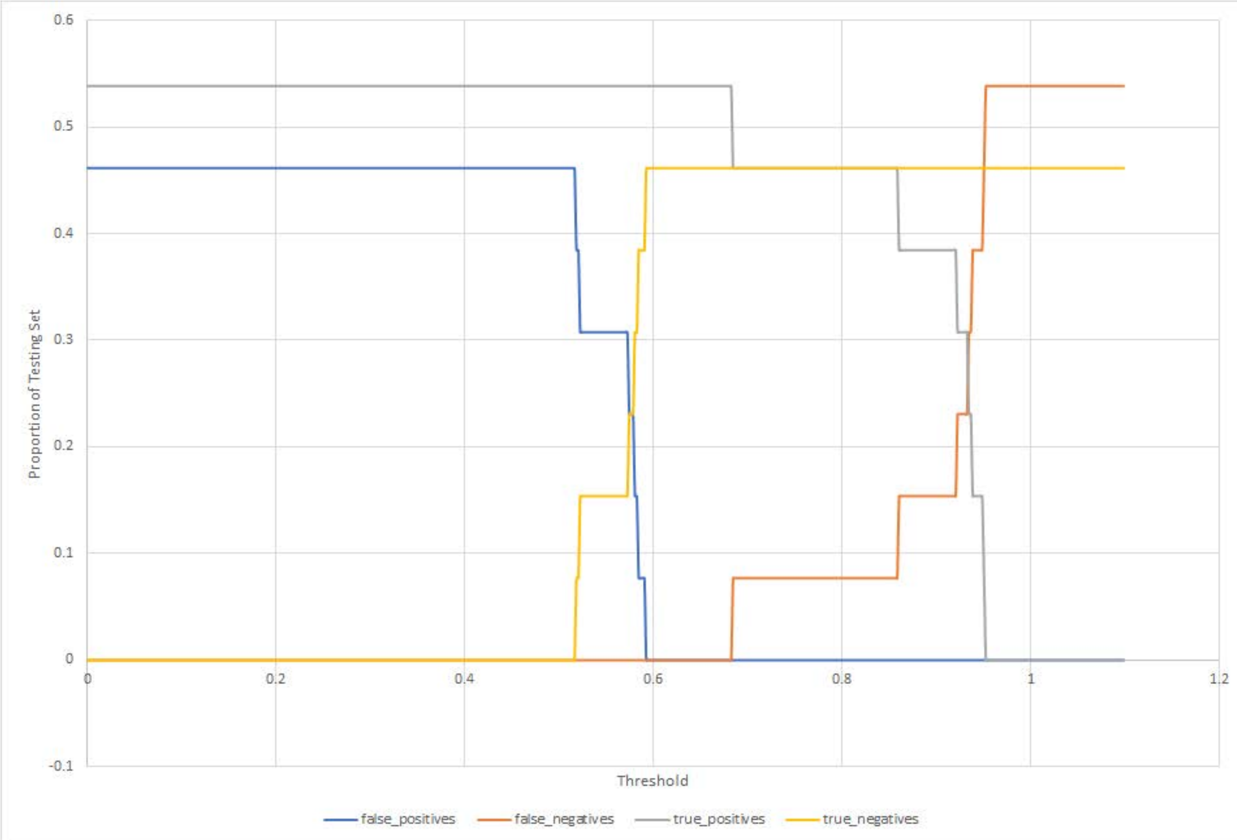

Figure 3.1: Receiver Operating Characteristic Curve

3.2 Voltage issues

After we received our PCBs and assembled them, we found out that our voltage regulators on both our wristband and camera sections were not functioning as intended; our wristband regulator only delivered around ~4V, while our camera regulator only delivered ~2V. This was further complicated by an additional misoriented capacitor on our camera PCB which caused a short.

After verifying that our traces surrounding the regulator worked as functioned, we attributed these issues to damaged voltage regulator components. As we had little time remaining and could not acquire spare regulators, we had to resort to bypassing the voltage regulator circuit, thereby preventing us from using our battery charging capability.

3.3 Bluetooth Issues

The verification of our bluetooth related requirements is worth discussing because they ultimately caused the largest unexpected slowdown in our development time. Our high level requirement was the ability to transfer 320 by 240 images from the camera module to the android phone, via bluetooth.

Upon receiving our components and testing them, we discovered our bluetooth module, the HC-05 was not able to meet its datasheet specifications. It rejected commands to initialize it to a 1,000,000 baud rate, forcing us to operate at 115,200 baud.

Additionally, while testing bluetooth communication between our android app and an Arduino using the HC-05, we found that the bluetooth connection would consistently crash if over 10,000 bytes had been transmitted without a pause. We attempted extensive troubleshooting, using a variety of different software techniques on the android side. These included reading the input stream byte by byte, using buffered readers, implementing our own buffers, and wrapping various types of readers around the stream. Due to time constraints, we were eventually forced to give up on transferring the entire image in one chunk. The workaround was to use chunks of 10,000 bytes, pausing transmission at each chunk to wait for the phone to send back an acknowledgement.

The result of the phone and the HC-05 both behaving out of manufacturer specification meant that significant workarounds had to be made to the design, since we were unable to secure replacement components in time. Without any memory designed into our camera module, there was no way to store 76,800 bytes per frame to be transmitted. We chose to downscale our resolution by a factor of 1/16, bringing it to 4800 bytes per frame. 4800 still exceeded the memory of the microcontroller, but further splitting that into 80 byte chunks per row of pixels would let us solve that as well. This solved the issue caused by the phone.

With that solved, many additional hours were spent rewriting our camera driver to properly synchronize with the pixel by pixel output of the OV7670. The low baud rate we operated at meant that it took longer to transmit a byte than it did for the OV7670 to capture a pixel. Using timing diagrams and several iterations on our code, we eventually managed to use an approach where we synchronously (with respect to the OV7670) captured 80 pixels from one row, then asynchronously transmitted 80 bytes during the next three rows of pixels. Using additional signals from the horizontal sync output of the OV7670, we then resynchronized with the OV7670 in time for the next row to be captured.

With all of this completed, we achieved a workaround to our failed bluetooth verification, successfully transmitting 80 by 60 images from the camera module to the phone. By selectively enabling and disabling interrupts, we also managed to receive commands transmitted by the phone to the camera module. Unfortunately, the hardware failure and software workaround did result in a resolution far below our original requirements. We found facial recognition to work extremely inconsistently at this resolution. With additional time to procure components, we ideally would have met the hardware requirement rather than use this software workaround.

See Appendix A for related timing diagrams and photos.

4. Costs

4.1 Parts Cost

Part/Part #	Price Per Unit	Quantity	Total Cost
0.96” in OLED LCD (HTDS-DI96)	$4.50	1	$4.50
3.3V Coin Haptic Motor (ROB-08449)	$4.95	1	$4.95
0.3MP Camera Module (OV7670)	$3.96	1	$3.96
HC-05 Bluetooth Module (LearningTech)	$9.95	2	$19.90
Surface Mount Tactile Switch (KSC422J 70SH LFS)	$0.43	1	$0.43
Li-Po Rechargeable Battery (3.7V 1200mAh) (Adafruit 258) (Camera)	$9.95	1	$9.95
Li-Po Rechargable Battery (3.7V 350mAh) (Adafruit 2750) (Smart Watch)	$6.95	1	$6.95
Arduino Uno R3	$24.95	2	$49.90
OTHER PCB PARTS			
Li-Po Battery Charging IC (MAX1555)	$2.03	2	$4.06
Boost Regulator (MCP1640)	$0.59	2	$1.08
Micro USB-B Header	$0.88	2	$1.76
Voltage Regulator IC (TCR2EF33LM)	$0.39	1	$0.39
PCB Capacitors (1206	N/A	14	$4.77

Size) (Details in Appendix A)			
PCB Resistors (1206 Size) (Details in Appendix A)	N/A	17	$7.64
Other Parts			
4.7 uF Inductor (Murata)	$0.40	2	$0.80
1206 Size Yellow LED (Kingbright)	$0.45	5	$2.25
16 MHz Clock Crystal (Abracon)	$0.25	2	$0.50
GRAND TOTAL	-	-	$188.55

4.2 Labor Cost

With the bulk of the work done over approximately 60 days, with an estimated 2 hours per day.

Salary ($/hr)	Hours	Total Labor Cost
40	120	$4800

5. Conclusion

5.1 Accomplishments

In planning this project, many phases of implementation were planned in order to minimize risk of total failure. Both authors began with almost no experience doing the types of development that were demanded by our design. The result of this planning meant that we successfully reached our late stage prototype phase.

We managed to achieve the full recognition cycle with components laid out on breadboards. Wireless communication was operational between modules, with images and data being processed successfully by our microcontrollers, the phone, and the API. The touchscreen user interface was working as well, allowing for database management and recognition independent of the hardware. The design was implemented with partial success on PCBs as well, although component damage and a mismatch between PCB and breadboard schematics prevented us from testing the camera module in PCB.

Even without final PCB implementation, our results indicate the success of the design. Our system is indeed able to recognize faces and communicate their names to the user, with a physical modularity and simplicity that minimizes user inconvenience.

5.2 Future Work

Continuing on from the previous section, our future work is to finalize hardware implementation of the design. This requires some corrections to the PCB schematics (in Appendix B) to bring them in line with the breadboard designs, as well as an overall shrinking of the PCB to achieve portability. Specifically, the primary issue with the PCBs is a mismatch on the camera to microcontroller connections with the schematic we based it on, as well as the missing button traces. [16]

The android app itself also has a lot of room for further features and a refined UI. Some changes will need to be made to the code to support background operation as well, currently the app only functions when it has the top priority of the android host. In the event of a negative match, the software should display a prompt on the wristwatch. If the user presses the button again, this would trigger the software to begin adding this person to the database. Though we would like for the software to also handle automatically associating information with the face, ethical concerns about recording audio make it likely that the user will have to enter the information manually using the phone touchscreen UI. As a placeholder, the system will associate this new face with the current time and date. See Appendix A for an example of an additional feature to be fully implemented in the future: multiple face recognition and reporting.

Additional power saving features can also be implemented, with the usage of the Bluetooth Low Energy standard.

5.3 Ethical Considerations

We recognize the importance of the IEEE Code of Ethics [2]. In particular, because this is an device that is designed to improve the lives of those suffering a particular medical condition, tenet 5, "...the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies…" [2] is most important to us. We would seek to make our device and its capabilities known to the greatest amount of prosopagnosia sufferers, as well as their friends and family, in an attempt to ultimately bring positive change in their lives.

While our device can help alleviate the social hurdles that individuals with prosopagnosia undergo, we must recognize the possible ways that our technology can be used nefariously or compromised due to its power. For this reason, tenet 1 of the IEEE Code, protecting the safety of the general public, is also of concern. Because a camera is used for a portion of our project, we understand that there may be security-related issues related to the use of the camera. In particular, we understand that any camera could be illegally accessed by hackers or other unscrupulous individuals. To prevent this, we will enable users to secure their cameras with a password and/or other measures.

An additional concern is the privacy issues associated with a camera capable of recording the general public. This issue continues to be debated as digital cameras continue to become commonplace. Possible ways of addressing these issues include adding an LED that brightly indicates the camera is in use, involving a mandatory disclaimer or user agreement before providing our device, and making changes to the way we store facial data in our database. For instance, it would be of practical use to the user to have unprocessed photos of new individuals in the database. However we are committed to tenet 1 of the IEEE code, and are prepared to reduce functionality if necessary, perhaps by only storing the key mathematical facial features needed to identify that face, rather than any type of reconstructable representation.

Finally, in our research and development of our device, it is important to recognize and credit any previous works and research that may have proved useful. Not doing so, aside from showing disrespect, would be clearly unethical. For this reason, we must recognize tenet 7 of the IEEE Code.

References

[1] S. Bate, “Information About Prosopagnosia,” *Prosopagnosia Research - About Prosopagnosia*. [Online]. Available: https://prosopagnosiaresearch.org/index/information.

[2] “IEEE Code of Ethics,” *IEEE - IEEE Code of Ethics*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

[3] B. Ray, “Bluetooth Vs. Bluetooth Low Energy: What's The Difference?,” *Link Labs: Low Power Wide Area Networks (LPWAN) and LTE-M for IoT*. [Online]. Available: https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy.

[4] A. Peterson, “Why those Samsung batteries exploded,” *The Washington Post*, 12-Sep-2016. [Online]. Available: https://www.washingtonpost.com/news/the-switch/wp/2016/09/12/why-those-samsung-batteries-exploded/?utm_term=.2a48bae64171.

[5] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, “Burst photography for high dynamic range and low-light imaging on mobile cameras,” *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–12, Nov. 2016.

[6] T. Marciniak, A. Chmielewska, R. Weychan, M. Parzych, and A. Dabrowski, “Influence of low resolution of images on reliability of face detection and recognition,” *SpringerLink*, 03-Jul-2013. [Online]. Available: https://link.springer.com/article/10.1007/s11042-013-1568-8.

[7] “Lithium Polymer vs Lithium-Ion batteries: What's the deal?,” *Scottie's Tech.Info*, 21-Jun-2015. [Online]. Available: https://scottiestech.info/2015/06/21/lithium-polymer-vs-lithium-ion-batteries-whats-the-deal/.

[8] Adafruit Industries, “Lithium Ion Polymer Battery - 3.7v 1200mAh,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/258.

[9] Adafruit Industries, “Lithium Ion Polymer Battery - 3.7v 350mAh,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/2750.

[10] W. Shi, X. Wang, X. Zhao, V. Prakash, and O. Gnawali, “Computerized-Eyewear Based Face Recognition System for Improving Social Lives of Prosopagnosics,” *Proceedings of the ICTs for improving Patients Rehabilitation Research Techniques*, 2013. Available: https://dl.acm.org/citation.cfm?id=2534520

[11] B. Mandal, S.-C. Chia, L. Li, V. Chandrasekhar, C. Tan, and J.-H. Lim, “A Wearable Face Recognition System on Google Glass for Assisting Social Interactions,” *Computer Vision - ACCV 2014 Workshops Lecture Notes in Computer Science*, pp. 419–433, 2015. Available: https://link.springer.com/chapter/10.1007/978-3-319-16634-6_31

[12] B. A. Singletary and T. E. Starner, “Symbiotic Interfaces For Wearable Face Recognition.” [Online]. Available: https://www.cc.gatech.edu/~thad/p/032_10_face%26handwriting/symbiotic-interfaces-for-wearable-face-recognition.pdf.

[13] “Best Practices for Developing with Kairos,” *Kairos*. [Online]. Available: https://www.kairos.com/docs/api/best-practices.

[14] B. Virdee-Chapman, “Face Recognition API,” *Kairos*. [Online]. Available: https://www.kairos.com/face-recognition-api. [Accessed: 22-Feb-2018].

[15] “OV7670/OV7171 CMOS VGA (640x480) CameraChip with OmniPixel Technology Datasheet.” [Online]. Available: https://www.voti.nl/docs/OV7670.pdf.

[16] OV7670 ARDUINO CAMERA SENSOR MODULE FRAMECAPTURE TUTORIAL, web page. Available at:
http://www.instructables.com/id/OV7670-Arduino-Camera-Sensor-Module-Framecapture-T/ [Accessed: 25-Feb-2018].

Appendix A: Additional Photos and Tables

Figure A.1: Photos used in training set for False Positive verification.

PCB Capacitors (1206 Size)			
1uF Tantalum (AVX)	$0.35	5	$1.75
22pF Film (Kemet)	$0.20	4	$0.80
10uF Tantalum (Kemet)	$0.45	2	$0.90
4.7uF Tantalum (AVX)	$0.40	2	$0.80
0.1uF Tantalum (AVX)	$0.52	1	$0.52
PCB Resistors (1206 Size)			
330 ohm (Panasonic)	$0.66	2	$1.32
10k ohm (Yageo)	$0.55	2	$1.10
220 ohm (Yageo)	$0.10	2	$0.20

1k ohm (Panasonic)	$0.66	2	$1.32
2k ohm (Yageo)	$0.10	2	$0.20
4.7k ohm (Yageo)	$0.10	2	$0.20
15 ohm (Stackpole)	$0.66	1	$0.66
976k ohm (Panasonic)	$0.66	2	$1.32
309k ohm (Panasonic)	$0.66	2	$1.32

Figure A.2: Complete Resistor and Capacitor List

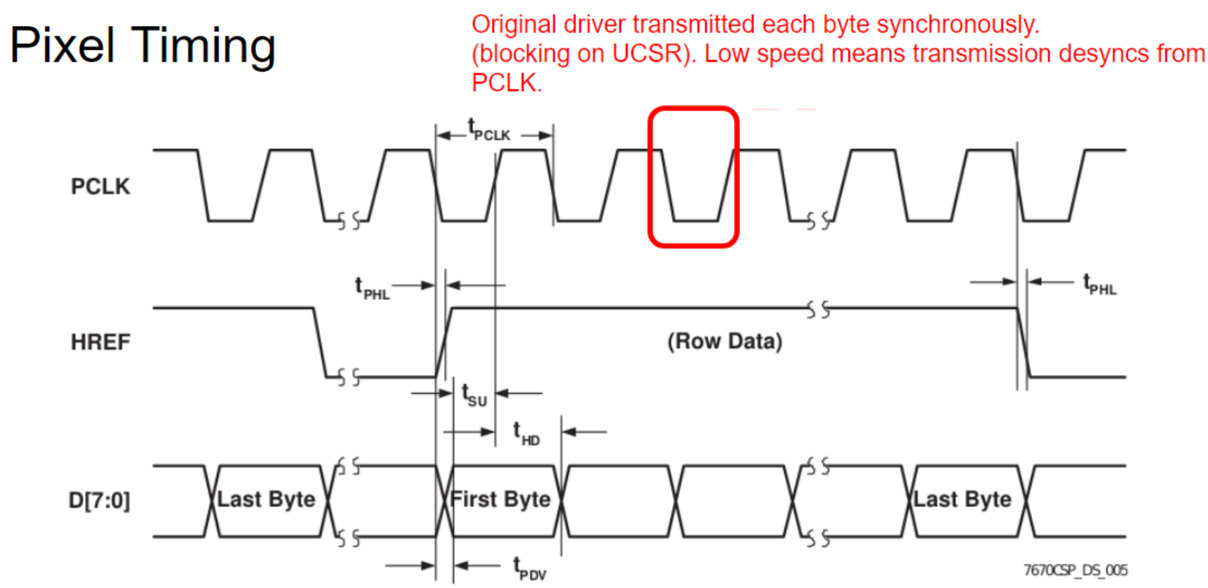

Figure A.3: Pixel timing diagram used to debug bluetooth and camera issues

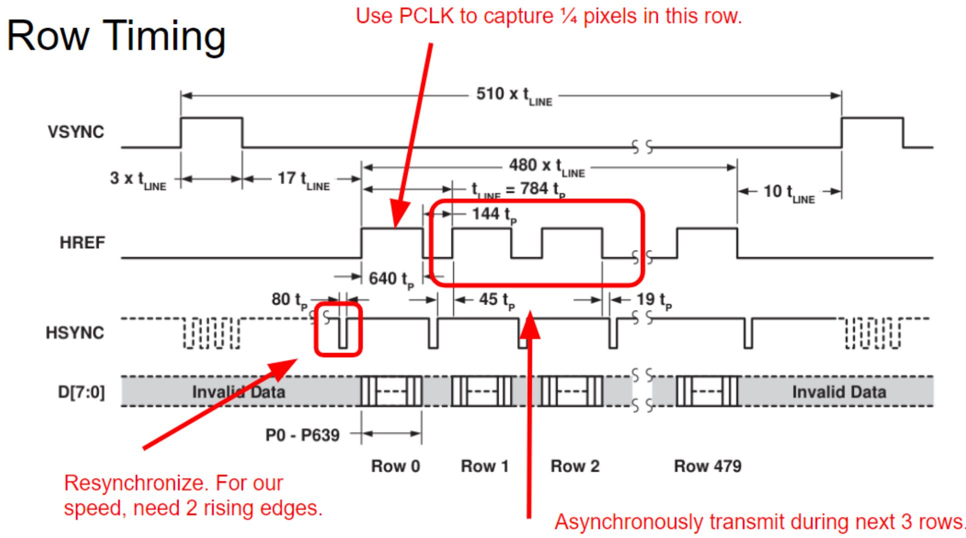

Figure A.4: Row timing diagram used to debug bluetooth and camera issues

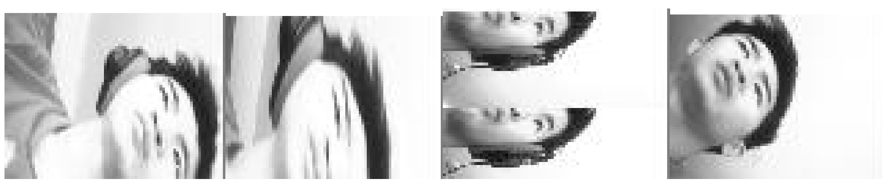

Figure A.5: Various Issues caused by bluetooth hardware performing out of spec. Rightmost image is with finalized software workaround.

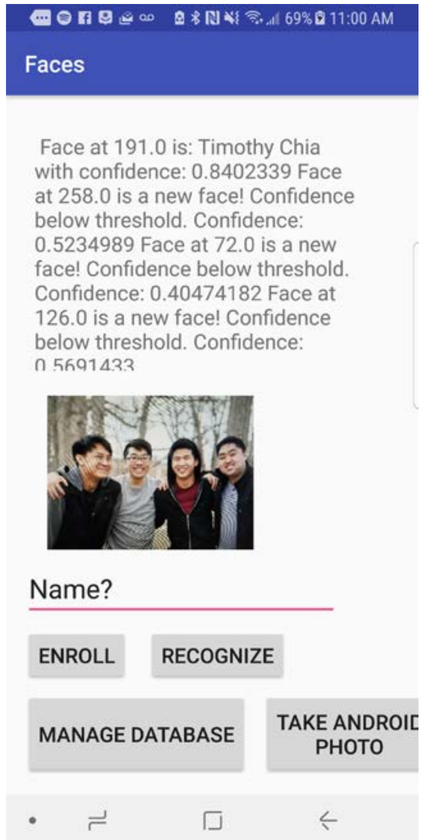

Figure A.6: Prototype for continued work. Multiple face recognition.

Appendix B: Schematics

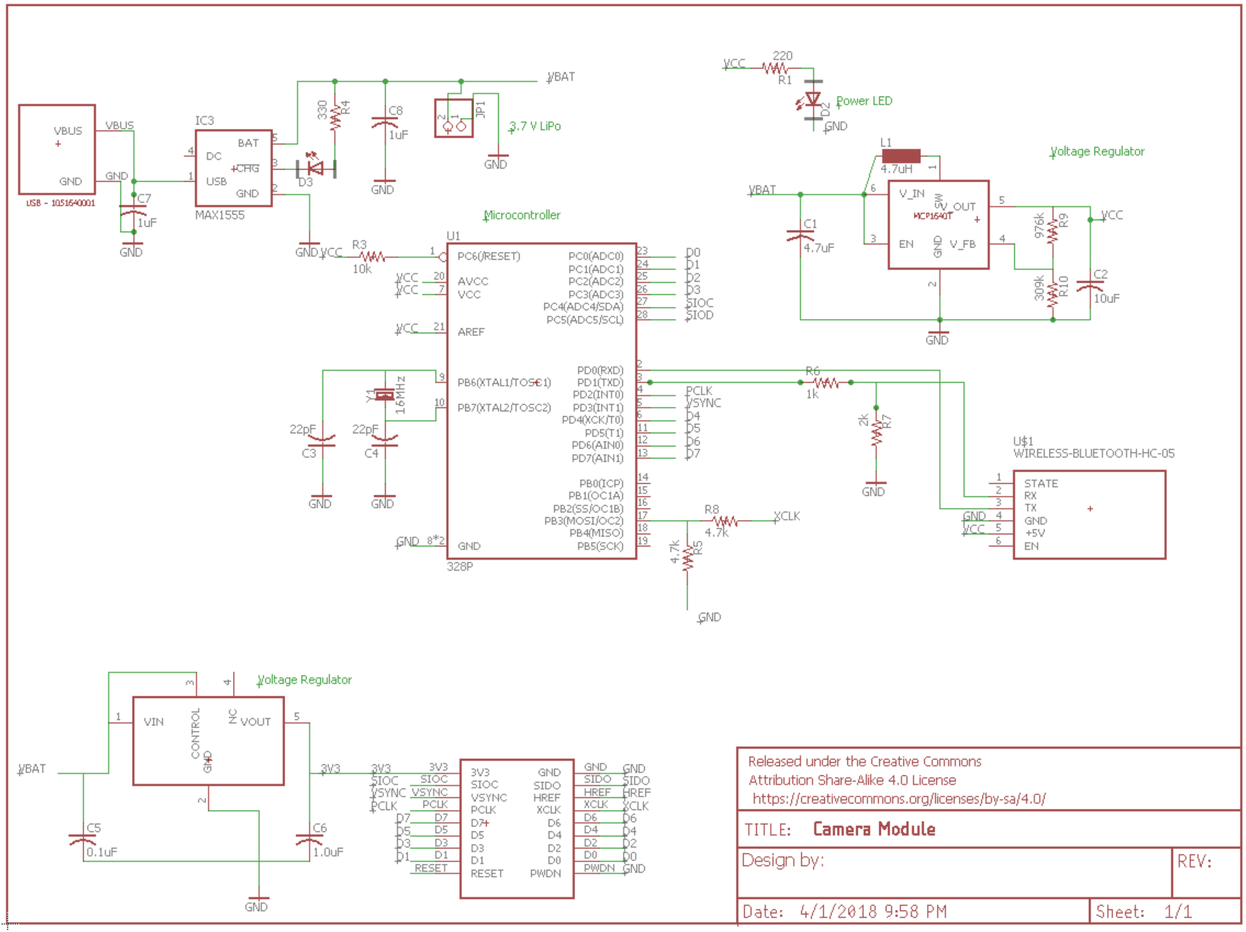

Figure B.1: Camera Module Schematic

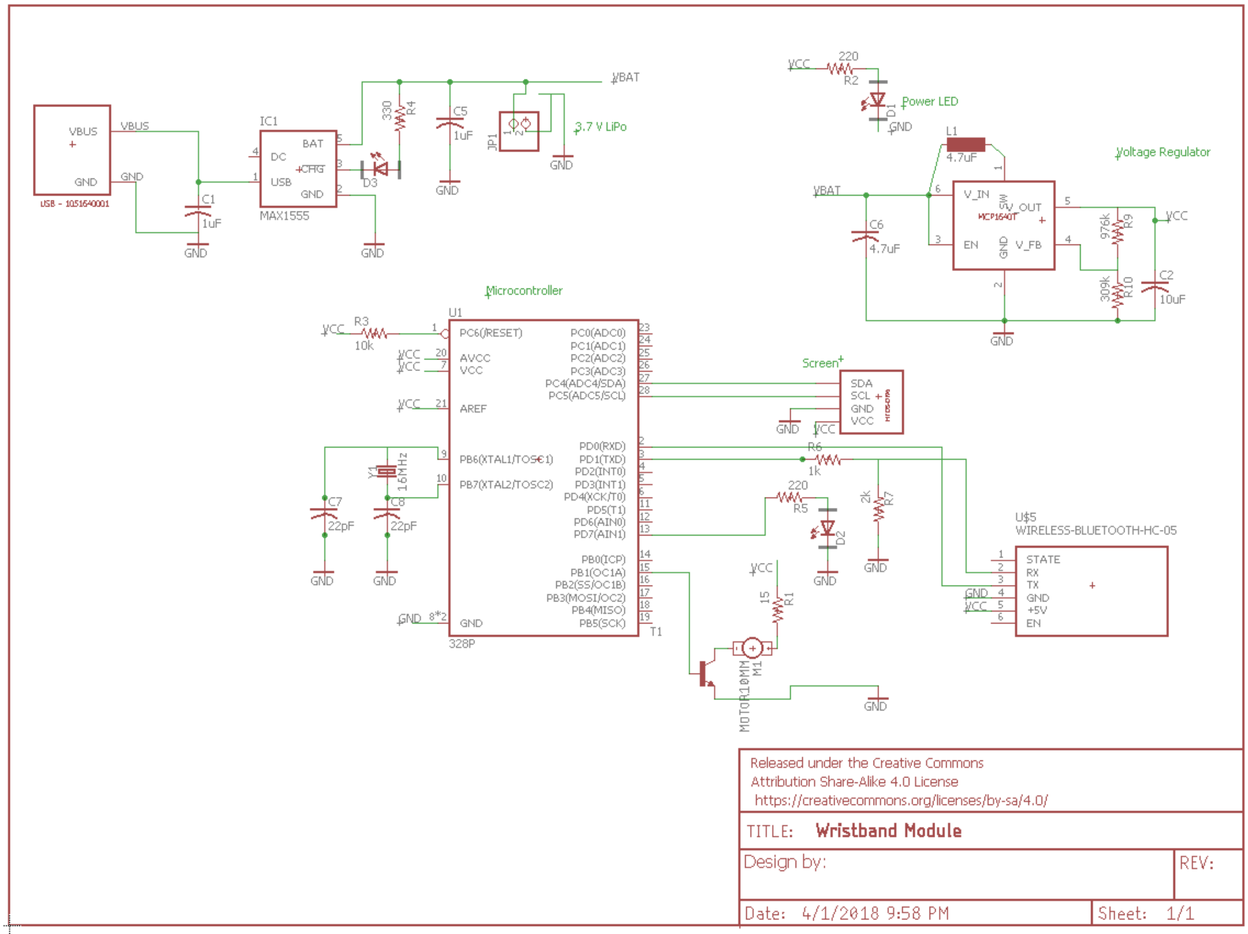

Figure B2: Wristband Module Schematic

Appendix C: Board Diagrams

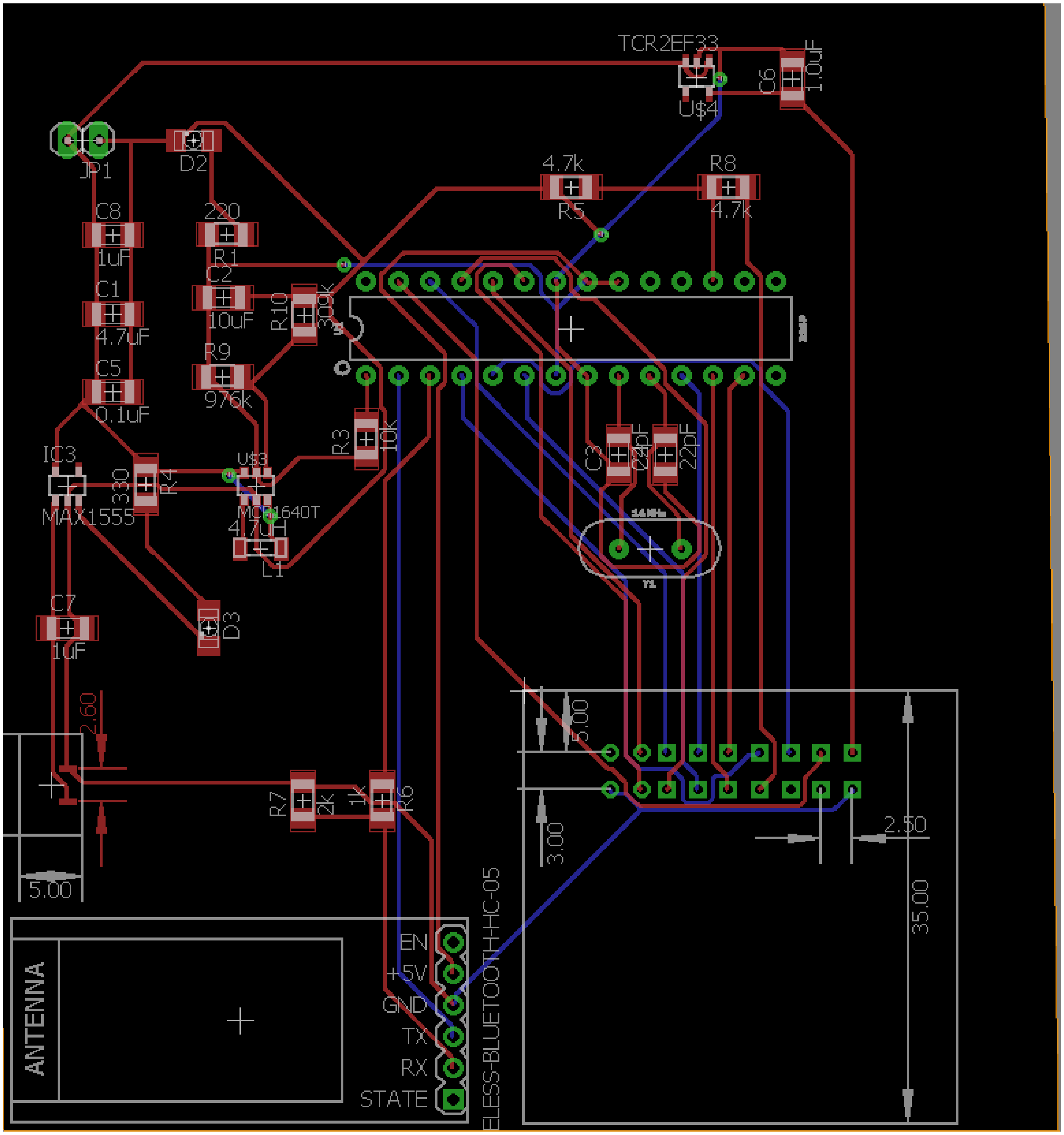

Figure C.1: Camera Module Board Diagram

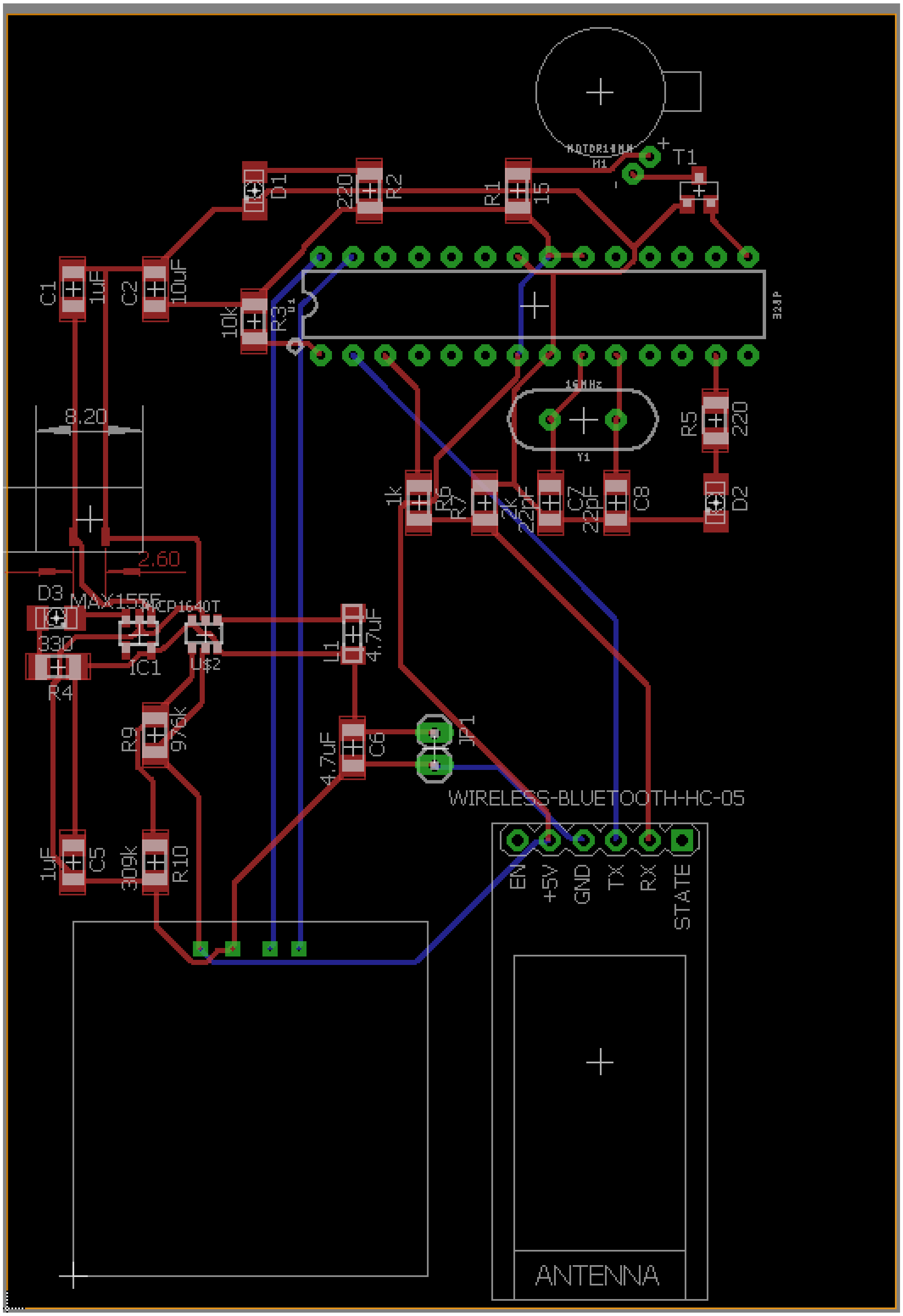

Figure C.2: Wristwatch Module Board Diagram

Appendix D: R/V Tables

Requirement	Verification
Transmission Speed >= 2.05 Mbps*	We will first transmit a string “*RDY*” from the arduino via HC-05 to the android. Immediately after, we will transmit a byte array of length 30,000. Taking a timestamp when the string is received, we will take a second timestamp when the next readcall returns from attempting to read 30,000 bytes.
Effective Range: 0 - 5m	Use a bluetooth terminal app on Android to communicate with the HC-05, separate by a distance of 5 meters. Confirm data is transmitted without error.
Current Draw < 40 mA, < 1mA inactive	Using test resistance and known voltage in, verify that current is within range.

Table D.1: Camera HC-05

Camera

Requirement	Verification
Resolution: 320x240	Obtain image from camera and transfer it to a computer. Check for resolution and visual artifacts.
Voltage In: 3.3V+/- 10%	Apply a voltage of 5V or less.
Current Draw: < 30mA (active) < 1mA (inactive)	Measure current drain both while taking a photo and while in inactive (soft sleep) mode, providing test resistance/voltage when needed.
Face > 21 x 21 pixel	Obtain and transfer series of captured images of a single subject (2ft, 4ft, 6ft, and 8ft distances) w/match in our image database. Check facial dimensions on computer.

Table D.2: Camera OV7670

Microcontroller

Requirements	Verification
Input Voltage: V_in = 5V ± 5%	Execute test programs on an input voltage of 5V

Serial I/O: At least 1 channel.	Use small test programs (on Arduino) to verify that needed I/O pins provide correct output.
Pins: 1 channel each of IIC and SPI	
GPIO: At least 14.	

Table D.3: Camera Microcontroller

Requirement	Verification	
Battery must be able to store enough charge for 2200mAH. To provide at least at 3.6V for 14 hours, or ~157 mA per hour.	With battery at full charge, apply a load calculated to induce current flow of 130mA. Measure time until current or voltage output decreases by 10%.	**1**
Voltage Out: 3.6V (+/- 10%)	With battery at full charge, measure voltage.	1

Calculation:

Device	Current Draw
Arducam OV7670	<30 mA (working) <1 mA (sleeping)
ATMega238-P Microcontroller	30 mA
HC-05 Bluetooth Module	40 mA (working) <1 mA (sleeping)

Table D.4: Camera Battery

Requirements	Verification
Transmission Speed >= 1 kbps*	We will first transmit a string “*RDY*” from the arduino via HC-05 to the android. Immediately after, we will transmit a byte array of length 30,000. Taking a timestamp when the string is received, we will take a second timestamp when the next readcall returns from attempting to read 30,000 bytes.
Current Draw < 40 mA, < 1mA inactive	Using test resistance and known voltage in, verify that current is within range.

Table D.5: Wristband HC-05

Requirements	Verification	Points
Input Voltage = 5V ± 5%	Providing 5V, verify that LCD is able to display text properly.	**1**
Characters Displayable Per Row > 15	When linked to Arduino, can display more than 15 characters on a row.	**1**
Must operate reliably with no display/pixel generation errors.	Enable a test message on screen; observe displayed text for any inaccuracies or visual glitches.	**1**
OLED display text viewable without backlighting.	Enable a test message on screen and have subjects with normal vision attempt to read from 1ft away.	**1**
Current Draw: < 10 mA	With the screen displaying the string "Display Initialized!" and measure the current on VCC.	**2**

Table D.6: Wristband Screen

Requirements	Verification
Input Voltage: V_in = 5V ± 5%	Execute test programs on an input voltage of 5V.
Serial I/O: At least 1 channel.	Use small test programs (on Arduino) to verify that needed I/O pins provide correct output.
Display Pins: IIC (LCD)	
GPIO: At least 2.(motor and button)	

Table D.7: Wristband Microcontroller

Requirements	Verification	Points
Voltage In: 2.0-3.6V Haptic usage: ~3.0V	Apply a voltage under 3.6V and check for regular function.	**1**
Current Draw: 100mA (motor in use) <1mA (motor inactive)	Measure current draw when motor is triggered and when motor is inactive, using multimeter.	**2**
Pulse of 1sec, 100% duty cycle		

Table D.8: Wristband Motor

Requirement	Verification	
Battery must be able to store enough charge to provide at least	With battery at full charge, apply a load calculated to induce current flow of 170mA. Measure time	1

140mA at 3.6V for 14 hours.	until current or voltage output decreases by 10%. (Maybe consider simply looking at 1-2 first hours only and determine what battery life would look like for whole time period)	
Voltage Out: 3.6V	With battery at full charge, measure voltage.	**1**

Calculation:

Device	Current Draw
Haptic Feedback Motor	5 mA = 100mA * 5% + 0* 95%
ATMega328-P Microcontroller	30 mA
LCD Screen	10 mA
HC-05 Bluetooth Module	60 mA
Total	105 mA

2200mAH/105mA = 15.7 Hours

Table D.9: Wristband Battery

Requirement	Verification	**points**
Facial Detection > 95%	Discussed in detail in the algorithm section of DD.	2
Match Accuracy* > 90%		3
False Positive < 10%		3
Runtime from photo received to associated information transmitted: < 5 seconds		2

*Accuracy rate is assuming a sample pool of 10-15 database subjects, with the recommended 8-10 images per user as recommended in the Best Use Guide [13]

Table D.10: Software Accuracy

Requirement	Verification	Points

Connectivity: Can transmit and receive calls to the Kairos API	Send a correctly formatted gallery_list_all POST json_request and print the JSON object received as a reply to the debug log.	**2**
UI: Can accept user input from the touchscreen via buttons and text input fields	Let the user type into a text field. Pressing a button should cause this text to be read into the app and displayed elsewhere or printed to the console.	**2**
Recognize: Can take a photo and return the name of the face in it. Or that the face is not present in the database.	Register a person in the database using the python library already verified to work. Use the phone camera to take a photo of the same person, attach it as a base64 string, and POST the request to the API. Log the response to the debug console, and verify no errors occurred and the correct subject_id was returned.	**2**
List: Can display a list of names of all unique faces in the database.	Post a request to the API for a list of all the subjects in the database. Print it to the phone screen and the debug log. Verify the list appears correctly on the phone screen.	**1**
Remove: Can remove a name and associated data from the database, using a user inputted name.	Allow the user to type into a text field, then press a button to call a remove function. After posting a remove request to the API, use a list request to check if the name was successfully removed.	**1**
Enroll: Can enroll a new face in the database under a user inputted name.	Allow the user to type into a text field, then press a button to call an enroll function. This function should trigger the phone camera, then post the image in an enroll request to the API. Then use a list request to check if the name now appears in the database.	**2**

Table D.11: Android App

Appendix E: R/V Data

	Start Times (ns)	Finish Times (ns)
Transmission Speed	481,181,570,122,471.00	481,184,045,322,783.00
	481,191,076,294,499.00	481,193,577,274,446.00
	481,206,197,081,993.00	481,208,624,296,888.00
	481,215,681,689,333.00	481,218,018,404,801.00
	481,226,558,835,006.00	481,229,080,845,995.00
	481,234,687,131,722.00	481,237,215,195,888.00
	481,245,953,636,666.00	481,248,429,181,508.00
	481,253,808,473,433.00	481,256,391,405,307.00
	481,261,684,617,701.00	481,264,240,420,981.00
	481,288,581,564,462.00	481,291,117,113,367.00
	481,297,563,598,417.00	481,300,045,173,989.00

Table E.1: Camera Bluetooth Times (Start and Stop)

Average Speed: 12042.55102 bytes/second

	Time for 30000 bytes (ns)	Speed (bytes/second)
Transmission Speed	2475200312	12120.23118
	2500979947	11995.2981
	2427214895	12359.84505
	2336715468	12838.53358
	2522010989	11895.26934
	2528064166	11866.7874
	2475544842	12118.54437
	2582931874	11614.70819

2555803280	11737.9926
2535548905	11831.75759
2481575572	12089.09386

Table E.2: Camera Bluetooth Times (30000 Bytes and Speed)

Current (mA)	Voltage (V)	Power (W)
32.60	4.94	0.161
34.70	5	0.174
33.10	5.01	0.166
33.20	4.97	0.165
33.7	4.89	0.165

Table E.3: Camera Bluetooth Readings (Current/Voltage/Power)

	Voltage In (V)
Trial #1	3.29
Trial #2	3.28
Trial #3	3.30
Trial #4	3.26
Trial #5	3.30

Table E.4: Camera (Voltage In)

Current Draw (mA)	ACTIVE mA	INACTIVE mA
Trial #1	16.75	0.976
Trial #2	16.74	0.979
Trial #3	16.73	0.98
Trial #4	16.72	0.979
Trial #5	16.73	0.977

Table E.5: Camera (Current Draw)

Face	Distance between face and camera (ft)	Pixel Dimensions
Trial #1	5.00	82 x 107
Trial #2	10.00	49 x 55
Trial #3	15.00	35 x 43
Trial #4	20.00	22 x 29
Trial #5	25.00	17 x 25

Table E.6: Camera (Subject Distance and Pixel Dimensions)

	Voltage In (V)
Trial #1	4.98 V
Trial #2	4.97 V
Trial #3	4.99 V
Trial #4	5.00 V
Trial #5	4.93 V

Table E.7: Camera Microcontroller (Voltage In)

Duration of Time Passed	Current (mA)	Voltage (V)
0 hr	126.40	3.92
0.5 hr	124.50	3.92
1.0 hr	122.70	3.9
1.5 hr	120.90	3.86
2.0 hr	118.80	3.83

Table E.8: Camera Battery (Current/Voltage Output vs. Time)

	Start Times (ns)	Finish Times (ns)
Transmission Speed	481,181,570,122,471.00	481,184,045,322,783.00
	481,191,076,294,499.00	481,193,577,274,446.00
	481,206,197,081,993.00	481,208,624,296,888.00
	481,215,681,689,333.00	481,218,018,404,801.00
	481,226,558,835,006.00	481,229,080,845,995.00
	481,234,687,131,722.00	481,237,215,195,888.00
	481,245,953,636,666.00	481,248,429,181,508.00
	481,253,808,473,433.00	481,256,391,405,307.00
	481,261,684,617,701.00	481,264,240,420,981.00
	481,288,581,564,462.00	481,291,117,113,367.00
	481,297,563,598,417.00	481,300,045,173,989.00

Table E.9: Wristband Bluetooth Times (Start and Stop)

Average Speed: 12042.55102 bytes/second

	Time for 30000 bytes (ns)	Speed (bytes/second)
Transmission Speed	2475200312	12120.23118
	2500979947	11995.2981
	2427214895	12359.84505
	2336715468	12838.53358
	2522010989	11895.26934
	2528064166	11866.7874
	2475544842	12118.54437
	2582931874	11614.70819
	2555803280	11737.9926
	2535548905	11831.75759
	2481575572	12089.09386

Table E.10: Wristband Bluetooth Times (30000 Bytes and Speed)

	Voltage In (V)	Current (mA)
Trial #1	4.74	2.43
Trial #2	4.68	2.44
Trial #3	4.7	2.43
Trial #4	4.68	2.42
Trial #5	4.73	2.42

Table E.11: Wristband Screen (Voltage/Current Draw)

Characters per row	Characters displayed
	25

Table E.12: Wristband Screen (Characters Displayed)

	Voltage In (V)
Trial #1	4.98
Trial #2	4.97
Trial #3	4.99
Trial #4	5.00
Trial #5	4.93

Table E.13: Wristband Microcontroller (Voltage Draw)

	Voltage In (active) (V)	Voltage In (Inactive) (V)
Trial #1	3.83	0.003
Trial #2	3.81	0
Trial #3	3.79	0.001
Trial #4	3.87	0.002
Trial #5	3.82	0.003

Table E.14: Haptic Feedback Motor (Voltage In)

	Current Draw (active) (mA)	Current Draw (Inactive) (mA)
Trial #1	45.1	0
Trial #2	47.4	0
Trial #3	49.3	0
Trial #4	47.2	0
Trial #5	48.1	0

Table E.15: Haptic Feedback Motor (Current Draw)

	Voltage Out (V)
Trial #1	3.89
Trial #2	3.88
Trial #3	3.88
Trial #4	3.89
Trial #5	3.89

Table E.16: Wristband Battery (Voltage Out)

Time Passed	Current (mA)	Voltage (V)
0 hr	122.5	3.76

0.5 hr	122.2	3.74
1.0 hr	121.8	3.72
1.5 hr	121.2	3.69
2.0 hr	120.6	3.61

Table E.17: Wristband Battery (Current/Voltage Out vs. Time Elapsed)