Self-Adjusting Monitor Stand

ECE 445 Final Report

Team 14: Anna Miller (annam4), Jake Nickel (jnicke7), Iris Xu (iris2)

Professor: Victoria Shao

TA: Jamie Xu

Spring 2022

Abstract

The goal of this project was to construct a reliable, easily adjusted monitor stand that can center itself on the user's face upon command. Face detection was utilized to determine the current position of the user relative to the center of the monitor and to calculate how far the motors in the stand must rotate to be centered on the user. Additionally, manual vertical adjustment options were implemented for each user to set the height to his or her preference. The design of this project consisted of three electrical and software-based subsystems, a mechanical system, and a wired "remote" module. The finished product was capable of adjusting the height according to the request of the user, accurately detecting the user's face in most conditions, converting the offset of the face from the center to a number of encoder steps the motor must turn, and centering itself on the user upon command all in a reasonable amount of time.

Table of Contents

1. Introduction			1		
	1.1.	Purpose	1		
	1.2.	Visual Aid	1		
	1.3.	High-Level Requirements	. 1		
	1.4.	Block Diagram	2		
	1.5.	Subsystems Overview	3		
2.	Desig	gn	4		
	2.1.	Power Subsystem	4		
	2.2.	Location Detection Subsystem	4		
	2.3.	Processing and Motor Control	5		
	2.4.	Mechanical Components and Wired "Remote" Module	8		
3.	Verifi	ication	10		
	3.1.	High-Level Functionality	10		
	3.2.	Power Subsystem	10		
	3.3.	Location Detection Subsystem	10		
	3.4.	Processing and Motor Control	11		
	3.5.	Mechanical Components and Wired "Remote" Module	13		
4.	Cost.		14		
	4.1.	Cost Analysis	14		
	4.2.	Schedule	14		
5.	Conc	lusions	15		
	5.1.	Successes and Challenges	15		
	5.2.	Future Work	15		
	5.3.	Ethics and Safety	16		
	5.4.	Broader Impacts	16		
6.	References				
7.	Appendix A: PCB Layout and Schematic				
8.	Appendix B: Requirements and Verification Tables				
9.	Appendix C: Parts Cost Table and Schedule				
10.	Appendix D: Face Detection Code and Flowchart				

1. Introduction

1.1. Purpose

Many monitors used today have fairly tight viewing angles, and viewing a computer screen from more than 30-45 degrees off the normal can introduce visual artifacts that make it difficult to read [1]. However, most consumer monitor stands are not easily adjustable, and it is time-consuming to constantly tweak the position to match every viewing angle. Additionally, many workplaces require the use of privacy screens that are designed to limit the field of view, which exacerbates this problem. To provide both screen privacy and viewer ease in a user-friendly package, we implemented a monitor stand as shown in Figure 1 that automatically adjusts the monitor to the user's position. This monitor stand has both automatic and manual components. The automatic function uses the user's position as detected by a camera and adjusts the angle of the monitor to be centered on the user. The height of the monitor is manually adjusted to the user's preference by a linear actuator.

1.2. Visual Aid



Figure 1. Front and side view of the finished monitor stand.

1.3. High-Level Requirements

• Adjustment to User Input: If the "adjust" button is pressed, the monitor stand should automatically pan and tilt the monitor to be centered on the user within 10 degrees both vertically and horizontally. See Figure 2. When the "up" or "down" button is pressed, the system should raise or lower the monitor at a minimum rate of 2 cm/s until the button is released, or the linear actuator reaches the maximum or minimum height.



Figure 2. Depiction of desired accuracy in the vertical direction.

- **Face Detection:** The camera must be able to detect its surroundings with a minimum of 15 FPS so that the user can adjust the monitor without pausing for the camera to process an accurate image of their current location. The detection system must also identify a face within the frame using a Haar cascade classifier.
- **Reasonable Response Speed:** The system must adjust the angle of the monitor to be centered on the user (according to the first listed requirement) within 8 seconds. This time was chosen to ensure the system is capable of properly adjusting the monitor more quickly than the average time it would take a user to stand up, adjust the monitor by hand, check that it was adjusted to the desired angle, and readjust as needed.



1.4. Block Diagram

Figure 3. High-level block diagram of the monitor stand and remote module.

1.5. Subsystems Overview

The primary subsystems in this product can be seen in Figure 3. The monitor stand is powered using a wall adapter, which invoked the need for this subsystem. The power subsystem converts the input from 120 VAC to DC 12 V and 5 V to accommodate the different voltage requirements of each component using an AC/DC converter and a buck converter.

A core aspect of this project is the ability to detect the user and use that location to adjust the monitor. The location detection subsystem consists of a USB camera and a computer vision (CV) processor. We used a Raspberry Pi 3 Model B to serve as the processor in this project. The camera connects to the processor by USB, and the processor sends data to the microcontroller in the processing and motor control subsystem using the serial peripheral interface (SPI) protocol. One change that was made to this subsystem from the initial design was the method used to power it. Originally, we planned to use an ODROID XU4 as the CV processor, which would be powered using the 5 V rail at 4 A. However, a switch was made to the Raspberry Pi due to certain microcontroller voltage level requirements. This introduced different powering requirements compared to the ODROID, so instead we powered the CV processor directly from the wall.

The processing and motor control subsystem uses data from the user and the CV processor to control the motors. The main components in this subsystem are the microcontroller and MOSFET motor drivers. The microcontroller is powered using a low-dropout regulator (LDO), which will step-down 5V from the power subsystem to 3.3V. If the microcontroller receives a signal to move the monitor up or down, it will communicate that information to the vertical motor driver. If the signal to adjust the angle of the monitor is received, the motor drivers will move the motors based on the data from the CV processor, while receiving data from the encoders on the motors to ensure the monitor is adjusted appropriately.

The mechanical subsystem is the main body of the product and consists of three motors. A linear actuator is used to adjust the height of the monitor in the vertical direction. Two identical gearmotors are used for the pan and tilt motors, which rotate the monitor about a vertical axis and horizontal axis respectively. The remote module serves as the user interface of the project and simply sends a signal to the microcontroller in the motor control subsystem based on which button the user pressed. There are three buttons: "up", "down", and "adjust". The "adjust" button communicates to the microcontroller that it should use the data from the location detection subsystem to adjust the monitor angle, while the "up" and "down" button will cause the microcontroller to drive the linear actuator accordingly.

2. Design

2.1. Power Subsystem

Design Procedure

In the primary design of the monitor stand, it was decided that the best method for powering the system was by wall-plug. Any internal powering system such as a battery would have a limited lifetime and would introduce the inconvenience of replacing or recharging the system. To use a wall-plug, the project required the ability to convert an AC voltage to a DC voltage. For the purposes of this project, the voltage and current ratings of an AC/DC converter and buck converter found from a retail store were sufficient.

Design Details

Since the project used off-the-shelf components for the power subsystem, the only values that were considered were the desired voltage and current ratings. The monitor stand requires a 12 V rail to run the motor drivers and H-bridges and power the motors. The two gearmotors and the linear actuator are all rated for a stall current of 1.8 A. In the initial design, a 5 V rail was used to power the CV processor, the camera, and the microcontroller. The ODROID XU4 is rated to require 4 A, and the camera and microcontroller together are rated to draw less than 1 A. To create two separate rails with these current ratings, we chose to convert the 120 VAC from the wall to a DC 12 V source. Then, a buck converter was trimmed to step-down the 12 V to create a 5 V source. Based on the current ratings of each component, the 12 V rail must be capable of supporting 10 A, and the 5 V rail must be capable of supporting 5 A. These current ratings informed the selection of the components in the power subsystem.

2.2. Location Detection

Design Procedure

We wrote a Python script to run for facial detection and angle calculations. Python was an appropriate choice because it is a high-level programming language, and the speed and memory consumption for this project were not major concerns. Python also has very thorough documentation and prebuilt packages on computer vision and SPI communication using the Raspberry Pi. We chose to use a base code that detected the general location of a face and circle it, because we did not need extreme specificity or recognition of individual monitor users. For this project, the lab inventory webcam was an affordable and sufficient option.

Design Details

The algorithm from OpenCV uses Haar features to distinguish backgrounds from faces. These features are better for recognizing edges, as long as the user's face is

not obscured. This allows us to create a classifier with a smaller dataset and decreases the time the program needs to detect the face with the camera. The Haar Cascade uses integral images, which allows us to calculate the angle in O(1) time rather than O(n) time, where n is the number of pixels in the window. Finally, the algorithm also uses Adaboost, an ensemble learning method that repeatedly changes the weights of the Haar features until a minimum error rate is met. There is also a final classifier of the weighted sum of the weak Haar Cascade classifiers. As a result, we can use as few as 200 features with a 95% accuracy [7]. After the face is detected, we use a function that takes in the aperture size of the camera and average width of a face to convert the number of pixels the user is offset from the center of the frame to a corresponding angle. The script then multiplies the angle by a factor of 434.22 to convert the angle to a number of encoder steps the motor should move and sends a 16-bit signed integer equivalent to the microcontroller. This value was measured by commanding the pan motor assembly to move exactly 10,000 encoder ticks in the clockwise direction, then determining the angle traveled during this move using calibrated pictures taken before and after. For 10,000 ticks, 23.03 degrees were traveled, which resulted in an angle-encoder conversion factor of 434.22 ticks per degree.

A flowchart for the software, as well as the code used in the final product, can be found in Appendix D.

2.3. Processing and Motor Control Design Procedure

The type of microcontroller was the primary design decision made during the development of the processing and motor control subsystem. Popular choices are often from the Atmel ATMega line, due to the ease of programming with Arduino utilities. However, it was determined that greater processing speeds and more memory would be required for this project, so rather than an Atmel chip, an STM32G0 series microcontroller was selected. This microcontroller was also selected because of its included hardware timers, which make high precision timing of PWM signals and the reading of encoders simpler.

Other design decisions for this subsystem revolved around the H-bridge motor drivers. It was decided that a symmetrical N-channel MOSFET design would be the most suited for this application. The alternative would be a P-channel MOSFET to drive the high side of each half bridge and an N-channel MOSFET driving the low side. This is a simpler design to drive, as the source terminal of each MOSFET would be held at a constant reference voltage. However, P-channel MOSFETs have slower response times and are higher on resistance than equivalent N-channel devices. Thus, a symmetrical design was chosen. However, this means that the high side MOSFET's source is not held at a constant voltage, as it is on the output of the half bridge. Thus, in order to meet the V_{gs} threshold voltage, the gate voltage of this MOSFET must be higher than the supply voltage of 12 V. As a greater supply voltage is not readily available to us, the easiest solution is to use a bootstrap circuit, as described in more detail in the following section.

Design Details

The circuit schematics for one half-bridge motor driver can be seen in Figure 4. This consists of the motor driver and the bootstrap circuitry.



Figure 4. Circuit schematic of one half-bridge motor driver for the H-bridge configuration.

The basic function of the bootstrap circuit shown is to provide a voltage boost to the gate of Q1 during the operation of the high side of the half bridge. As shown in Figure 5, during the low side on time, C_{boot} is charged to the supply level of 12 V. In order to activate the high side (Q1), the bootstrap IC routes the C_{boot} voltage to Q1's gate terminal. However, since C_{boot} is biased relative to the source terminal of Q1, a constant V_{gs} is maintained throughout the process of switching Q1 on, ensuring it stays active. The value of C_{boot} must be chosen such that it is not discharged before the low side is activated again to restore its voltage. Because of this requirement, this circuit cannot stay in the high side activation state for too long, thus limiting the maximum duty cycle of the half bridge to approximately 85%.



Figure 5. Charging and discharging path of the bootstrap circuit. Source: Adapted from [4].

The most critical component value in the half bridge circuit is that of the bootstrap capacitor, C_{boot} . (C7 in Figure 4) Calculation of this component value depends on the gate capacitance of the high side MOSFET, which can be calculated by the following equations from [4].

$$C_g = \frac{Q_g}{V_{Q1g}} \tag{1}$$

$$C_{boot} \ge 10 \times C_g$$
 (2)

where Q_g is the gate charge of the MOSFETs used and V_{Q1g} is the difference between the V_{DD} and the forward voltage across the boot diode. Applying this equation gives a gate capacitance of 0.994 nF. This yields an optimal value of 10nF for C_{boot}.

2.4. Mechanical Components and Wired "Remote" Module Design Procedure

The mechanical subsystem was designed over a couple of conversations with the ECE Machine Shop. The goal of this subsystem was to have the monitor capable of vertical adjustment and the adjustment of the angle of the monitor. For the vertical adjustment, we decided that the best motor to use would be a linear actuator. For the angular adjustment of the monitor, gearmotors were selected as the most practical option. In the conversations with the Machine Shop, it was decided that the linear actuator would push the monitor up and down along a physical rail and that the gearmotors would be used in conjunction with a worm gear each to rotate the monitor mount about a horizontal and a vertical axis.

The remote was designed to send a signal to the microcontroller when a button is pressed. The remote consists of two buttons for vertical adjustment and one for the automatic adjustment. See Figure 6. When initially designing the project, we considered different ways to implement the remote, including wireless methods. Ultimately, it was decided that a wired remote would be the best method as it would ensure the remote would not be separated and lost. A wired remote also simplifies the implementation and reduces the chances of additional errors and bugs. We used a phone cable because it is safer than using a USB cable.



Figure 6. The remote module of the monitor stand.

Design Details

When choosing the motors, we calculated the torque that was required to hold up a typical monitor so the monitor does not weigh the motors down when they are not running. Our original monitor weighs approximately 10 pounds, or 44.4 Newtons. Assuming a reasonable moment arm length of 0.1m, the gravitational torque on the motor will be \sim 4.5 Nm.

The tilt motor is also driven using a worm drive system, so as to prevent back-driving the motor with gravity. This system introduces an approximate 4:1 reduction in gearing, so our motor should be able to output \sim 1.2 Nm.

For a good safety factor, and to account for frictional losses, our motor's stall

torque should ideally be roughly 4 times this value. This gives us an approximate motor torque requirement of 5 Nm. The gearmotor we selected also fits this requirement within the current limit of 10 A continuous draw.

3. Verification

3.1. High-Level Functionality

Ultimately, the monitor stand satisfied all of the high-level requirements. It appropriately adjusted itself in response to user input, it was able to detect faces with reasonable accuracy, and it was able to finish repositioning itself in under 8 seconds from the time a command was sent. The requirements and verifications for each of the subsystems can be found in Appendix B.

3.2. Power Subsystem

The measurements and verifications for this subsystem were taken by probing the two power rails with a multimeter. Table 1 displays the output voltage and the current drawn from the power subsystem. The measured values were within the stated tolerances of our requirements, and the subsystem was able to effectively power our project.

	DC Output Voltage (V)	Current Drawn (A)
12 V Rail	12.001	1.58
5 V Rail	4.998	0.11

Table 1.	Recorded	multimeter	readings	from	the power	subsystem.
			0		1	2

3.3. Location Detection

The location detection subsystem was capable of analyzing a frame at 15 FPS and calculating the degree offset of the user relative to the monitor from the target position. In most situations, the camera was able to accurately identify faces in the frame. Figure 7a, 7b, and 7c demonstrate this. These images were obtained by annotating the faces detected in the frame in the Python script. Figure 7a shows a face properly identified in front of a background with other objects. Figure 7b shows the user's face selected and identified when other faces are present. Figure 7c shows a face with glasses properly identified. However, occasionally, the processor recognized non-faces as shown in Figure 7d.



Figure 7. Annotations in Python to demonstrate face detection of (a) a single face, (b) the largest face in the frame, (c) a single face with glasses, and (d) a false positive face.

After completing the software aspect of the project, we attempted to quantify the

number of false positives the processor identified. To do this, we ran the code and camera for 1 minute in a room with a clean background and good lighting as shown in Figure 7c. Then, we ran the code for 2 minutes in the same room with poor lighting that made it more difficult to see the contrast in a person's face. For both of these conditions, a face was present in the frame for the first minute and the blank wall was shown alone for the second minute. In good lighting, we found that 50 detections were processed by the code for the minute where the face was present, and two detections were processed when the face was removed. After analyzing the images, we concluded that only one of the first 50 detections were incorrect in addition to the two false positives. In dim lighting, 38 detections were made in the first minute, while two detections were made in the second minute. From the images, we found four of the first 38 detections were inaccurate in addition to the false positives. Finally, we ran the code for one minute with frames of objects that were not faces such as the sweater shown in Figure 7d in the same lighting conditions as the first test. In that first minute, there were 14 detections processed, which were all false positives. Based on this data, we concluded that the environmental factors such as lighting and background objects played a heavy role in inaccurate detections. Overall, however, the facial detection was accurate often enough for our purposes.

3.4. Processing and Motor Control

Verification of the PWM motor control signal requirement was carried out using an oscilloscope. The motor driver software was activated at a target duty cycle of 20%. The high side and low side control signals were probed simultaneously to ensure proper synchronization, PWM frequency, duty cycle, and dead time. Results are shown below.



Figure 8. Motor driver PWM control signals.



Figure 9. Motor driver 12 V output.

Figure 8 clearly shows a measured PWM frequency of 19.994 kHz, which meets our frequency requirement. Figure 9 shows the 12 V output of the half bridge being driven with the above control signals. The same 20% duty cycle is present in the full voltage output.



Figure 10. Full duty cycle operation of the half bridge.

Figure 10 shows the half bridge operating at maximum duty cycle. In our design, this maximum duty cycle was 85%. This meets our requirement of 80%.

In order to meet the encoder requirement for this subsystem, some minor design changes had to be made. In our original PCB layout, encoder 1's "A" channel

followed a path to the microcontroller that ran over the 12 V copper plane used for powering the motor drivers. When these drivers were running, high switching currents injected noise into this encoder line, which would occasionally trigger the rising edge interrupt monitoring that encoder. In order to solve this issue, this noisy trace was severed, and a jumper wire soldered in its place. This ensured that the noisy copper plane was bypassed, and noise reduced greatly. After this design change was made, we observed no missed encoder steps at any operational speed tested.

In order to meet our response time requirement, data must be transferred quickly from the Location Detection subsystem to the Processing and Motor Control subsystem. In our original design, the communication protocol selected was I2C, which was selected for its small number of required pins and relatively high baudrate. However, after extensive testing of the STM32's integrated I2C peripheral, proper operation was not able to be achieved. The STM32 appeared to be constantly driving the I2C clock line low, as if it was attempting to perform "clock stretching" on the bus. Clock stretching is when the peripheral device is not ready to accept data from the controller, and pulls the clock line low to signal this. However, even after disabling this feature on the STM32, the symptoms persisted. Rather than spend excessive time attempting to solve this issue, it was decided to switch to the Serial Peripheral Interface (SPI). This was a simple change, as only the clock and MOSI lines were required for our project. Simply soldering jumper wires from the SPI microcontroller pins to the board's external communication port was enough to solve this issue.

3.5. Mechanical Components and Wired "Remote" Module

Before installation of the linear actuator with the monitor mounting box, we verified the total linear motion of the actuator to be 6 inches. However, the mounting box reduces this movement to a total of 4 inches instead.

As for the pan and tilt motors, due to the availability of parts, the worm gears cause the motors to stall if the monitor is too heavy. By slowly increasing the weight on the monitor mount, we found it currently supports the weight of 2 lbs before the motor is unable to move and starts stalling.

4. Costs

4.1. Cost Analysis

Labor

In this project, the cost of labor can be attributed to that of the team and the Machine Shop. The average annual salary for an electrical engineering graduate of the University of Illinois at Urbana-Champaign (UIUC) was \$79,714 as of the 2018-19 academic year [8]. Assuming a 40 hour work week for 52 weeks in a year, this salary can be converted to an hourly rate of around \$38.32/hr. This project will take approximately 100 hours to complete.

Total Team Labor = $3 \times (\$38.32 \times 2.5 \times 100 \text{ hrs}) = \$22,992$

Labor and material rates for engineering machine shops at UIUC range from \$35/hr to \$60/hr depending on the department [9, 10, 11]. Therefore, a reasonable rate assumption of \$50/hr can be made. This project will take approximately 18 hours for the Machine Shop.

Total Machine Shop Cost = $50/hr \times 18 hrs = 900$

Parts

The total cost of all parts is \$265.37. The calculation of the cost is performed in the Parts Cost Table in Appendix C.

Total Cost

Sum of Costs = Total Labor Cost + Machine Shop Cost + Cost of Parts = \$22,992 + \$900 + \$265.37 = \$24,157.37.

4.2. Schedule

See Appendix C.

5. Conclusions

5.1. Successes and Challenges

Successes

The project was able to satisfy the major requirements and high-level goals. This monitor detects faces with reasonable accuracy and converts it into an angle that is within our $\pm 10^{\circ}$ tolerance. It also has all three types of motion: automatic pan and tilt and user-input vertical adjustment. Finally, it is fully integrated, as the script is able to communicate with the microcontroller and does not require any reprogramming to adjust the viewing angle.

Challenges

Our current camera used for facial detection has an approximate field of view of 30 degrees. This is acceptable for our proof of concept, but an actual product would require a much greater field of view. A wide angle camera would serve this purpose well. Integration of such a camera into our system would be trivial, as only the focal length would need to be updated for proper camera frame to world frame conversion.

The current facial detection algorithm exhibits occasional false detections, which cause incorrect movement of the monitor. This could be iterated upon by selecting a higher resolution camera, or by using more computationally heavy face detection algorithms.

5.2. Future Work

Given more time to work on our project, there are a few changes that would increase its overall effectiveness and stability. As it currently stands, our mechanical subsystem does not meet the 15 lb weight requirement, and as such cannot support the full weight of a true monitor attached. Improving the weight handling capacity of this subsystem would allow for a real monitor to be attached, thus completing our project.

Regarding the current motor control method, there are improvements that could be made here as well. The current microcontroller software utilizes rapid polling of a hardware timer to ensure outgoing signals are switched at the proper time. This means that when the software polling rate slows down, the PWM control signals are not switched in time, resulting in incorrect operation of the motors. In order to rectify this, we would like to switch to an interrupt-based approach, which would operate regardless of the computational load placed on the microcontroller by other functions. This would improve the overall stability of our motor control solution. In addition, our facial detection algorithm could be iterated upon to yield slightly more accurate results. As it stands, the number of Type 1 and Type 2 detection errors are acceptable, but could be reduced further with the use of a higher resolution camera. Use of a more powerful CV processor than a Raspberry Pi 3B would also allow for more computationally heavy face detection methods to be

used, which may yield more robust results.

Finally, a more robust angle-encoder tick conversion factor would lead to higher accuracy in the target position. Our current measuring method relied on visual cues to determine the traveled angle for a known number of encoder ticks. This method may not be the most accurate, as camera distortion and other confounding factors could lead to inaccuracies. A more robust method would rely on the use of a gyroscope or accelerometer to determine this angle.

5.3. Ethics and Safety

One of the major ethical concerns in our system is the potential for privacy invasion. The monitor stand uses a camera that is constantly running to identify where the user is sitting and processes the image data to determine how the monitor should be moved. Section I.1 of the IEEE Code states that we must "protect the privacy of others" [12] and Section 1.6 of the ACM Code of Ethics states that "an essential aim...is to minimize negative consequences of computing, including threats to health, safety, personal security, and privacy" [13]. To ensure the privacy of the user is protected, the camera image is only accessed internally by the processor and deleted once the calculation is complete. There is no long-term storage of any image data, and the images are not used for any purpose other than calculating the necessary monitor adjustments.

Another ethical consideration related to the detection of the user. A common issue in detection and facial recognition is the disparity of detection between races and skin tones. It is our responsibility to treat everyone fairly and avoid engaging in any kind of discrimination based on color or race [12, 13]. We shall do our best to address this problem by selecting training datasets that contain faces of various races and skin tones.

Additionally, there are safety considerations that must be made regarding the system. The IEEE Code of Ethics states in Section I.1 that we must "hold paramount the safety, health, and welfare of the public" [12]. Section II.9 also states that we must consider how our system could injure others or their property. In our design, we are limiting the speed at which the motors move the monitor to prevent damage, and we are addressing user safety concerns by requiring that the surface temperature of the entire device remain below 115°F and by controlling the current through the motor. The linear actuator also includes electromechanical endstops to prevent motor stalls when the end of travel is reached.

5.4. Broader Impacts

This project demonstrated an application that can be used at home or in the workplace to help the user create a more efficient, convenient, and ergonomic workstation. However, the technology can be adapted to further applications. One way the self-adjusting nature of the device can be used is to improve accessibility. By circumventing the need for manual adjustment, it becomes easier for people with disabilities or limited mobility to use the monitor. The self-adjusting technology can also be useful for face scanners that require a person to be well-aligned with the scanner such as the thermal scanners recently deployed in some hospitals to screen people for COVID-19 symptoms [14].

Acknowledgements

Our project would not have been possible without the support of numerous people. We would like to thank our TA, Jamie Xu, and David Null for their assistance throughout the project, as well as Professor Victoria Shao for her support. We would also like to thank Gregg Bennett, David Switzer, and the ECE Machine Shop for helping us to design the mechanical components of our project and building a prototype for us.

References

- [1] Kaelee Nelson, "Why Do I Need a Computer Privacy Screen?" *HP Development Company*, *L.P.*, 11-Jan-2019. [Online]. Available: https://www.hp.com/us-en/shop/tech-takes/why-do-i-need-computer-privacy-screen. [Accessed: 5-Feb-2022].
- [2] "Basic rules for a successful facial biometrics project," *RecFaces*, 11-Aug-2021. [Online]. Available: https://recfaces.com/articles/basic-rules-for-a-successful-facial-biometrics-project. [Accessed: 10-Feb-2022].
- [3] "Driving a high current DC motor using an H-bridge," Driving a high current DC Motor using an H-bridge - Northwestern Mechatronics Wiki, 03-Nov-2009. [Online]. Available: http://hades.mech.northwestern.edu/index.php/Driving_a_high_current_DC_Motor_using _an_H-bridge. [Accessed: 22-Feb-2022].
- [4] Mamadou Diallo, "Bootstrap Circuitry Selection for Half-Bridge Configurations," *Texas Instruments Incorporated*, 2018. [Online]. Available: https://www.ti.com/lit/an/slua887/slua887.pdf?ts=1645368266361. [Accessed: 21-Feb-2022].
- [5] A. Rowe, "Safe Workplace Touching Temperatures," Safety Action, 5-Jul-2018. [Online]. Available: https://www.safetyaction.com.au/blog/safe-workplace-touching-temperatures. [Accessed: 8-Feb-2022].
- [6] Gupta, Rohan. "Breaking Down Facial Recognition: The Viola-Jones Algorithm." Medium, Towards Data Science, 12-Feb-2020. [Online]. Available: https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algori thm-29d9106b69999. [Accessed: 10-Feb-2022]
- [7] "Using AdaBoost to Minimize Training Error." *MIT Press*. [Online]. Available: https://mitpress.mit.edu/sites/default/files/titles/content/boosting_foundations_algorithms/ chapter003.html. [Accessed: 10-Feb-2022].
- [8] "Salary Averages," *Ece.illinois.edu*, 2022. [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages .[Accessed: 22-Feb-2022]
- [9] "Machine Shop," Scs.illinois.edu, 2022. [Online] Available: https://scs.illinois.edu/resources/cores-scs-service-facilities/machine-shop [Accessed: 22-Feb-2022].
- [10] "MechSE Master Maker Shop," Machineshop.mechanical.illinois.edu, 2022.[Online] Available: https://machineshop.mechanical.illinois.edu/home. [Accessed: 22-Feb-2022].

- [11] "Machine Shop," *Physics.illinois.edu*, 2022. [Online]. Available: https://physics.illinois.edu/research/facilities/machine-shop. [Accessed: 22- Feb-2022]
- [12] "IEEE Code of Ethics," IEEE, Jun-2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 1-Feb-2022].
- [13] "ACM Code of Ethics and Professional Conduct," *ACM*, 22-Jun-2018. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 1-Feb-2022].
- [14] R. Ali, R. Blaszkiewicz, T. Schuessler, and J. Stuteville, "Thermal scanners Provide No-touch COVID Screening at Franciscan Health Hospital Entrances," *Franciscan Health*, 6-Aug-2020. [Online]. Available: https://www.franciscanhealth.org/about/news-and-media/thermal-scanners-provide-no-tou ch-covid-screening-at-franciscan-health-hospital-entrances. [Accessed: 03-May-2022].



Appendix A: PCB Schematic and Layout

A.2. PCB layout.



Appendix B: Requirements and Verification Tables

Requirements	Verification	Result?
 Must convert 120 VAC to 12 V±1 V and 5 V±1 V power rails. 	 a) Connect the subsystem to a 120 VAC voltage source from the wall. b) Probe the 5 V rail using a digital multimeter (DMM) and ensure the measurement remains between 4 V and 6 V. c) Probe the 12 V rail using a DMM and ensure the measurement remains between 11 V and 13 V. 	Passed.
2. The 5 V rail should have a 5 A capacity, and the 12 V rail should have a 10 A capacity.	 2. a) Use electronic load testing equipment to verify current requirements. b) Load should be placed on each rail and run for 10 minutes to ensure stability. While the load test is occurring, a DMM should be used to measure the voltages of both the 12V and 5V rails. These rails should not deviate from their designed voltages by more than the design specification (±1V). c) The maximum voltage deviation from nominal should be recorded in the notebook. 	Passed, but ultimately these current maximums were never drawn by our project.

B.1. Power subsystem requirements and verifications.

B.2. Location detection subsystem requirements and verifications.

Requirements	Verification	Result?
1. Image processing must occur at 15 FPS to ensure accurate face detection without excessively heavy computation [2].	 a) The image processing algorithm should be developed with an integrated FPS tracker. b) The system should be run in various lighting and facial conditions, while recording the framerate. Lighting conditions: Normal lighting measure in a lit workspace. Low light in the same workspace, but with the light turned off. High light in the workspace with an additional bright lamp directed at the face. Facial Conditions: Neutral expression looking into the camera Neutral expression turned slightly away from the camera. Contorted facial expressions (e.g. angry) looking into the camera 	Acceptable. The face detection is reliable a majority of the time as shown in Section 3.3.

	- Faces with glasses or accessories c) The minimum framerate should be recorded in the notebook.	
2. The calculated ray from the camera to the detected facial position should be within 10 degrees of the actual ray from camera to user.	 a) The image processing algorithm should include reporting on the target position (vector coordinates). b) Define the true camera-face vector by measuring the distance from the camera to the user's face (x₁) and measuring the distance that the face is horizontally from the center of the camera (x₂). The angle between the camera-face vector and a ray normal to the monitor can be calculated as θ = sin⁻¹(x₂/x₁). c) The system should be run in a variety of lighting conditions and environments, and the true camera-face vector should be compared against the calculated. d) The maximum deviation of the calculated position from the actual position should be recorded in the notebook. 	Passed.

B.3. Processing and motor control requirements and verifications.

Requirements	Verification	Result?
1. Must convert the voltage from the 5 V rail to $3.3 V \pm 0.2 V$ to be used by the microcontroller.	 a) Connect the input of the LDO to a 5 V source. b) Probe the output of the LDO using a DMM and ensure the measurement remains between 3.1 V and 3.5 V. c) The maximum deviation from 3.3 V should be recorded in the notebook. 	Passed.
2. The microcontroller must not miss encoder steps at normal operational speeds between 2 RPM and 10 RPM.	 a) Connect the motor/encoder Clk to an Arduino board. b) Program the board to output a counter that increases when the encoder is rotated CW, and decreases CCW. Manually rotate the encoder assembly, then attempt to return it to the original position as closely as possible. Measure the encoder's reported new position. c) The average position drift should be recorded in the notebook. 	Passed.
3. The motor drivers must supply 3 A without exceeding 140°C.	 3. a) Use an electronic load to simulate the motors and input a square wave (V_{pp} = 10 V) using a function generator. b) Using a DMM, probe the output of the half-bridge driver configuration to measure the 	Passed.

	 current. c) While the motor is operating in Test 1, measure the temperature of the assembly using an infrared thermometer. d) Ensure the temperature never rises above 140°C. 	
4. The motor control unit must output a 12 V PWM with a minimum of a 80% duty cycle at 20 KHz.	 4. a) Power the PCB containing the motor control circuits using 12 V and 5 V from a power supply. b) Using an oscilloscope, probe the output of the half-bridge driver configuration to measure the output PWM. 	Passed.

B.4. Wired remote control module requirements and verifications.

Requirements	Verification	Result?
1. The remote must send a signal to the microcontroller in the processing unit via a minimum 3-ft cable.	 a) A DMM should be used to verify the signal at the PCB is detectable by the microcontroller. b) The voltage received by the microcontroller should be above the minimum logic 1 level (0.7 * V_{DD} or 2.31 V). This will be measured with the DMM under various operating conditions. c) The average voltage at the microcontroller should be measured and recorded in the notebook. 	Passed.

B.5. Vertical mechanics requirements and verifications.

Requirements	Verification	Result?
1. The linear actuator assembly must have a minimum of 6 inches of travel while supporting a load of 15 lbs.	 a) Place a 15 lb load to simulate a monitor on the stand. A ruler should be used to measure travel length. b) A fixed point on the linear actuator mechanism should be chosen, and the system driven to its lower and upper end stops. The difference between the height of this point at these lower and upper stops should be calculated. c) This travel distance should be recorded in the notebook. 	Partially passed, the stand was unable to support the weight of a monitor due to the limited availability of mechanical components.
2. The surface temperature of the assembly must not exceed 115° [5].	 a) While the motor is operating in Test 1, measure the temperature of the assembly using an infrared thermometer. b) Ensure the temperature never rises above 115°F. 	Passed.

Requirements	Verification	Result?
1. The pan motor assembly must be capable of traveling 45° in either direction when measured from the center position.	 a) Using a ruler, draw a straight line out from the front of the motor assembly base. b) Align a protractor with the 90° marking on top of the line drawn in Step a). Draw rays originating from the same point as the initial line to mark out 45° to the left and right. c) With the monitor mount and pan motor in the center position, mark a position of the rotating assembly that is in line with the center line. d) Drive the motor to rotate to the left. Check that the mark made in Step c) is now aligned with or past the leftmost 45° line. Return the assembly to the center position. e) Repeat Step d), this time rotating the motor to the right. 	Passed.
2. The surface temperature of the assembly must not exceed 115°F [5].	 a) While the motor is operating in Test 1, measure the temperature of the assembly using an infrared thermometer. b) Ensure the temperature never rises above 115°F. 	Passed.

B.6. Pan mechanics requirements and verifications.

B.7. Tilt mechanics requirements and verifications.

Requirements	Verification	Result?
1. The tilt motor assembly must have 15° of travel upward and downward when measured from the center position.	 a) Start the monitor perpendicular to the mounting box. b) Set up a stable vertical clamp with a protractor whose flat edge aligns with the edge of the monitor. c) Have the tilt motor go to maximum downwards position and measure the new angle with the stable protractor. d) Repeat for maximum upwards position. 	Passed, but the assembly was unable to support a monitor's weight.
2. The surface temperature of the assembly must not exceed 115°F [5].	 a) While the motor is operating in Test 1, measure the temperature of the assembly using an infrared thermometer. b) Ensure the temperature never rises above 115°F. 	Passed.

Appendix C: Parts Cost Table and Schedule

C.1. Parts Cost Table.

Part Number	Description	Manufacturer	Quantity	Price/ Unit	Retail Cost	Paid Cost
LM200-10B12-C	AC/DC Converter	Mornsun America, LLC	1	\$24.06	\$24.06	\$24.06
200217	5A Voltage Regulator	DROK	1	\$9.99	\$9.99	\$9.99
VA-139-52	3A Voltage Regulator	Valefod	6	\$1.83	\$10.99	\$10.99
926-LP2986AIM X33NOPB	LDO Voltage Regulators	Texas Instruments	1	\$2.28	\$2.28	\$2.28
STM32GO61C8 T6	Mainstream Arm Cortex-M0+ 32-bit MCU	STMicroelectronics	1	\$4.05	\$4.05	\$4.05
490-TB007-508- 02BE	Fixed Terminal Blocks 2	CUI Devices	5	\$0.85	\$4.25	\$4.25
NCP51530BDR2 G	Half Bridge Driver	onsemi	6	\$2.03	\$12.18	\$12.18
IRFZ24NPBF	N-Channel MOSFET	Infineon Technologies	12	\$0.65	\$7.84	\$7.84
1655-1354-1-ND	Diode Schottky 45V 15A	SMC Diode Solutions	12	\$0.80	\$9.60	\$9.60
4869	227:1 Gearmotor 25Dx71L with encoder	Pololu	2	\$34.95	\$69.90	\$69.90
L11TGF1000NB 150HW-T-1	8-inch Linear Actuator Motor	ECO LLC	1	\$41.99	\$41.99	\$41.99
RASPBERRY PI 3	Raspberry Pi 3 Model B BCM2837	Raspberry Pi	1	\$41.25	\$41.25	\$0.00
Tecknet C016 720P Web Camera	USB Camera	iNassen	1	\$26.99	\$26.99	\$0.00
			Total		\$265.37	\$197.13

C.2. Schedule.

Week	Anna	Jake	Iris
2/28	Finish PCB design	Finish PCB design	Finish PCB design
3/7	Research face-detection algorithms	Research and code half-bridge control	Research face-detection algorithms
3/14	BREAK	BREAK	BREAK
3/21	Continue research and come up with test cases for face detection and start soldering	Work on communication between camera and CV processor and start soldering	Continue research and begin coding face detection and start soldering
3/28	Finish soldering and start debugging the power subsystem	Finish soldering and start debugging the power subsystem	Finish soldering and start debugging the power subsystem
4/4	Test and debug the remote module	Debug the motor control	Debug the location detection subsystem
4/11	Finish any remaining debugging	Finish any remaining debugging	Finish any remaining debugging
4/18	Mock Demo	Mock Demo	Mock Demo
4/25	Address any issues and finalize project for Demo	Address any issues and finalize project for Demo	Address any issues and finalize project for Demo
5/2	Work on Final Report	Work on Final Report	Work on Final Report

Appendix D: Face Detection Code and Flowchart

D.1. Software flowchart.



D.2. Face detection code.



```
def detectFace(frame):
   frame gray = cv.cvtColor(frame, cv.COLOR BGR2GRAY)
    frame gray = cv.equalizeHist(frame gray)
   faces = face cascade.detectMultiScale(frame gray)
   maxRect = 10000
   if len(faces) > 0:
        img name = "opencv frame.png"
        cv.imwrite(img name, frame)
        face = 1
   return face
def analyzeFrame(imagePath):
   midpoint = [0,0] # midpoint between eyes
   eye_cnt = 0 # keep track of how many eyes
   frame = cv.imread(imagePath)
   frame gray = cv.cvtColor(frame, cv.COLOR BGR2GRAY)
   frame_gray = cv.equalizeHist(frame_gray)
#-- Detect faces
   faces = face cascade.detectMultiScale(frame gray)
   maxRect = 0
   for i in range(len(faces)):
        (x, y, w, h) = faces[i]
        area = np.pi*(faces[i][2]*faces[i][3])/4
        if area > maxRect:
            maxRect = area
            faceROI = frame gray[y:y+h,x:x+w]
            eyes = eyes cascade.detectMultiScale(faceROI)
            for j in range(len(eyes)):
                (x2, y2, w2, h2) = eyes[j]
            target = (x + w//2, y + h//2 - h//4)
        d, pixeld = pixToDim(w)
   offset = [310-target[0], 240-target[1]]
   phys offset = [offset[0]*pixeld, offset[1]*pixeld] #mm
   pan = np.arctan(phys offset[0]/d) #radians
   tilt = np.arctan(phys offset[1]/d) #radians
   return pan, tilt
```

```
def pixToDim(width pix):
    focalLength = \overline{3.85} #mm
    avg width = 142.5 \# mm
    F = 1080 \# (width pix*600) /avg width
    d = (avg width*F)/width pix #mm
    pixeld = avg width/width pix #mm
    return d, pixeld
def write targets(t1, t2):
   msb1 = (t1 >> 8) \& 0xFF
   lsb1 = t1 \& 0xFF
   msb2 = (t2 >> 8) \& 0xFF
    lsb2 = t2 \& 0xFF
    spi.writebytes([msb1, lsb1, msb2, lsb2])
parser = argparse.ArgumentParser(description='Code for Cascade Classifier
tutorial.')
parser.add argument('--face cascade', help='Path to face cascade.',
default='data/haarcascades/haarcascade frontalface alt.xml')
parser.add argument('--eyes cascade', help='Path to eyes cascade.',
default='data/haarcascades/haarcascade eye tree eyeglasses.xml')
parser.add argument('--camera', help='Camera divide number.', type=int,
default=0)
args = parser.parse args()
face cascade name = args.face cascade
eyes cascade name = args.eyes cascade
face cascade = cv.CascadeClassifier()
eyes cascade = cv.CascadeClassifier()
if not face cascade.load(face cascade name):
    print('--(!)Error loading face cascade')
    exit(0)
if not eyes cascade.load(eyes cascade name):
    print('--(!)Error loading eyes cascade')
    exit(0)
spi = spidev.SpiDev()
spi.open(0, 0)
spi.max speed hz = 50000
spi.mode = 0
#-- 2. Read the video stream
cap = cv.VideoCapture(-1)
while(1):
    face = 0
    offset = (100, 100)
    if not cap.isOpened:
            print('--(!)Error opening video capture')
            exit(0)
   while face == 0:
```

```
ret, frame = cap.read()
if frame is None:
    print('--(!) No captured frame -- Break!')
    break
face = detectFace(frame)
#-- 3. Analyze the saved image
path = r'opencv_frame.png'
# Determine the difference between face and the target position
pan, tilt = analyzeFrame(path)
panenc = int(math.degrees(pan)*434.22)
tiltenc = int(math.degrees(tilt)*434.22)
# Send data to the MCU
write_targets(panenc, tiltenc)
# Delete image when finished
os.remove(path)
```