Smart Desk

Daniel Mathieu, EE/CSE, Aidan Fitzpatrick, EE, Tristan Koopman, EE, and John Melloni, CSE

***Abstract*—This document will go over the current design for the senior design project titled Smart Desk. The introduction will go in depth about motivation and need for this project. The overview section will then give the reader a high-level understanding of what the system is, and how its functionality will be achieved. Following the overview, the different blocks of the system will be decomposed in detail. Lastly, the conclusion will summarize the current state of the project and go over the planned future work. *Abstract*—The theory, design, and implementation of a desk that offers conventional functionality in addition to touch capabilities is presented. The Smart Desk benefits people ranging from students to professionals and even casual users by providing a functioning computer within a desktop. The main technical exploration is obtaining touch capabilities through IR light rather than through capacitive materials or other traditional implementations. This alternative would offer both a cheap and durable solution to touch screens that allow for a versatile Smart Desk.**

I. Introduction

Have you ever tried to get work done at your desk, but it was too cluttered? Most users find it difficult to be productive in an environment where they must constantly be searching through stacks of papers and books. Our solution to this problem is eliminating the need for physical resources, from papers and books to even calculators, and replacing them with a digital alternative. Our system offers the conventional functionality of a desk, a surface where one can place items on and do work, while offering an embedded touch screen. This touchscreen will interface with current computer and offer all the functionality current computer offers. Instead of searching by hand for the gain equation of a common collector amplifier through a 500-page electronics textbook, one can simply search for it in the book's digital copy in seconds. This allows users to work more efficiently and reduce the space required to store these physical resources. The idea of a touch screen desk is nothing new, as architects and several other professionals already use this technology every day. What separates this solution from the others is cost. For most people, it is not feasible to spend 4-16 thousand dollars on a desk that is not critical to their ability to do their jobs. Another question one may ask is, "Why do I need a touch screen desk, why don't I just buy a touchscreen monitor?" This is a fair question, one could simply buy a large touchscreen monitor, and lay it flat on a desk surface to achieve the same goal. Once again, this solution costs one to five thousand dollars based on the size and resolution of the screen. Our solution will cost under $500 to deliver a durable and cost-effective touchscreen. Current solutions [1] can achieve a higher screen resolution due to the fact they are able to place the LCD screens directly behind a capacitive touch screen, but this is costlier. The price of the proposed solution is possible due to the availability of inexpensive LCD screens and replacing the capacitive screen with an infrared (IR) solution. The reason for substituting IR is the fact the IR surface scales better. Capacitive screens not only cost significantly more to produce, but when scaling the screen size, powering the surface and achieving the same accuracy becomes an issue. This scaling issue is not for large IR screens. The only issue is powering the number of IR LEDS necessary to fill the surface with light. The specifications for our design are listed in Table 1. The requirements analysis for the design was motivated by trying to allow the desk to maintain its functionality of a standard desk while integrating a touch screen in the surface of the desk. Regarding the touch screen, the requirements revolve around the system's ability to behave like a traditional touch screen, such as sufficient resolution, real-time responses, and high accuracy up to a fingertip. For the desk, it became clear that the main requirement would be a space limitation on the systems necessary to achieve the touch screen in order preserve a person's ability to comfortably fit their legs under the desk. At the end of the design, a person will be able to attach their current computer to our system and see their screen displayed in the surface of the desk. From there, they will be able to do any form of work they would like, from writing out calculations for a math problem to searching through articles for a research paper to simply playing games, all through the touchscreen.

TABLE I
Specifications

Requirement	Specification
Provide accurate touch inputs	Accuracy up to a fingertip
Provide increased versatility	Maintains function of conventional desk, while offering touchscreen
Responses in real time	<100ms
Provide sufficient leg room	Area under desk will be at least 2.5ft high.
Interfaces with current computer	Interfaces with standard computer inputs
Robustness	Scratch and spill resistant
Power	Use the power of an average TV

A. Overview

Our solution involves taking apart a 32" HD (720P) LCD TV [2], and embedding it into a wooden desk. A piece of EndLighten acrylic is placed on top of the screen and is lined with IR LEDs along all four sides. When a user places his or her finger on the acrylic, the IR light is reflected downward, through the screen, and captured via two cameras below, as is shown in Figure 2. The cameras are connected to a Raspberry Pi, which is used to process the images, and convert them into a mouse click at the desired location.

Prior to deciding on using an LCD screen combined with IR LEDs, we had considered several alternatives. A projector would have fulfilled the same role as the LCD screen, however it would have eliminated all legroom underneath the desk. Instead of using IR LEDs, we considered using IR lasers. We chose not to use lasers, as the touch accuracy would not have been as high, since the lasers are shot out in straight lines while the LEDs diffuse in a fan-like pattern. Furthermore, the cost of the lasers and receivers would have used our entire budget, if not more.

Additionally, we considered using capacitive touch instead of IR reflection. Capacitive touch primarily comes in the form of surface capacitive touch or projected capacitive (PCAP) touch [3]. Surface capacitive touch was not a possible solution, as it does not allow for multi-touch. PCAP does allow for multi-touch. However, PCAP suffers from two flaws. PCAP does not scale well. This is a result of Indium Tin Oxide (ITO) being the most commonly used transparent conductive material. ITO has a relatively high resistivity. As the size of the touch sensor increases, the electrode length increases as well, resulting in an even higher resistance. This increase in resistance yields an increase in the RC time constant, concluding in a slower response time and poor sensor performance [3]. The second flaw for PCAP is the price of a sensor. The largest PCAP touch screen development kit that we could find was Reach Technology's 12.1" kit, which costs over $660 [4].

For those reasons, we have chosen to implement our design using the LCD Screen with IR LEDs. Figure 1 shows the block diagram of our proposed solution.

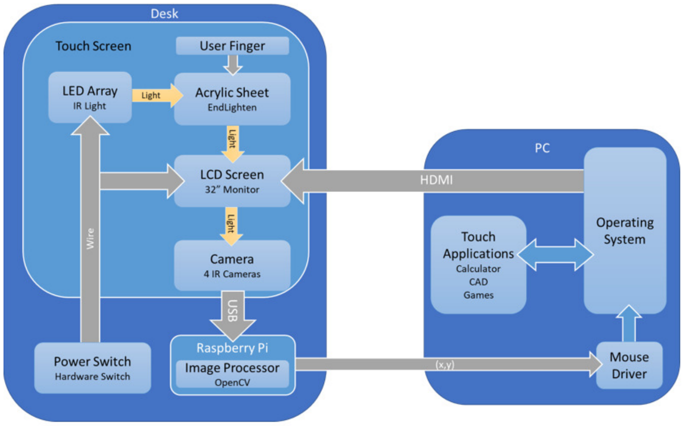

Figure 1: Block Diagram

B. *Block 1: IR LEDs & EndLighten Acrylic*

We reduce the cost of the touch screen by using IR light rather than a capacitive surface. IR light is used because it is invisible to the human eye [5]. Therefore, we can illuminate the entire screen of the desk without distracting the user. When the user touches their finger down on the surface of the desk, the IR light is reflected down through the screen where it is captured by cameras below.

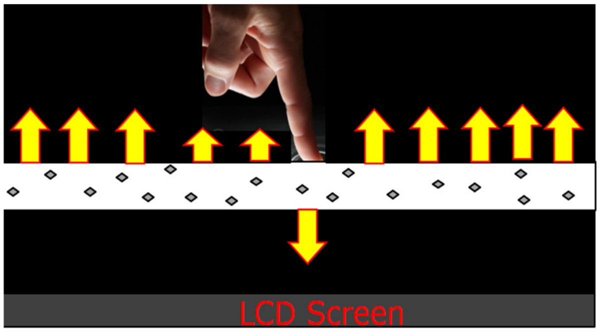

Figure 2: IR light reflected down by user's finger

To build this system we took advantage of ACRYLITE's product EndLighten acrylic [6]. This acrylic contains clear light-diffusing particles embedded inside the material which evenly emit light out of the top surface. We inject IR light into the sides of the material by using a thin IR LED strip. The injected light is evenly reflected out of the top surface of the acrylic. This process is shown in Figure 3. Also, EndLighten is a completely transparent and colorless material, making it ideal for our project.

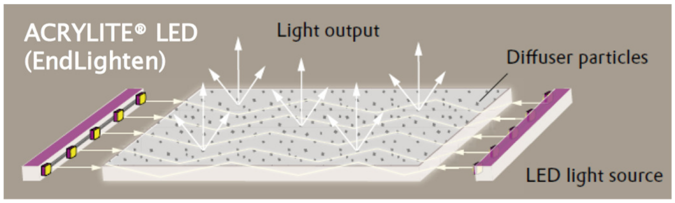

Figure 3: Light diffusion in EndLighten acrylic[6]

The IR light must reach all parts of the acrylic material to ensure the entire screen is illuminated. Otherwise, the touch screen would lose functionality in poorly illuminated areas. Looking at the EndLighten specifications [6], we chose to use the XL size sheet which is fully illuminated up to 48 inches when LED light is injected on every side of the sheet. We are using a 29" x 17" sheet of EndLighten to completely cover the LCD screen. Surrounding the acrylic on all sides with the IR LED strip also has the advantage of maximizing the screen illumination which will increase the SNR (signal-to-noise ratio) when capturing images of the IR light with our cameras.

This subsystem was implemented successfully as part of our MDR deliverables. Using an IR camera, we were able to confirm that the IR light was evenly distributed across the EndLighten material. When a finger was touched to the surface of the acrylic, IR light was reflected down towards the camera. This process was repeated for every part of the screen ensuring that the entire surface will have the ability to be converted to a touch screen.

C. *Block 2: LCD Screen*

The EndLighten acrylic sheet is placed on top of a modestly priced LCD screen to build the touch screen. The acrylic is transparent, so the user clearly sees the images displayed on the LCD screen. To embed the touch screen in the surface of the desk, the LCD screen must have its hardware components relocated.

We purchased the LCD screen [2] from Walmart to use in our design. The screen was housed in a plastic encasing that acts as a layer of protection and a stand. Inside this casing, there are three main hardware components: the screen controlling PCBs, LED backlighting, and a diffuser that will evenly disperse the light. These components are critical to screen functionality. The PCBs control the screen including the crystals that display the RGB colors and the volume and channel controls. The backlighting illuminates the screen, so that it can be viewed, and the brightness can be adjusted. The light diffusers evenly distribute light across the LCD screen, so the picture is undistorted.

To install the LCD screen inside the desk, we disassembled the plastic cover over the screen and carefully took out the hardware components. The PCBs controlling the screen and TV functions were attached to the screen by a ribbon cable. Due to the short length of this cable, the PCBs were moved from underneath the screen to just behind it at the back of the desk. We installed a ledge with a ground plane to house the PCBs. The PCBs are installed close enough to the LCD screen so that the ribbon cables are not stressed. The backlighting for the screen was relocated to farther underneath the screen along with the light diffuser layers. The wires connecting these components to the PCB were lengthened using wire connectors. The complete relocation of the screen components is shown in Figure 4.

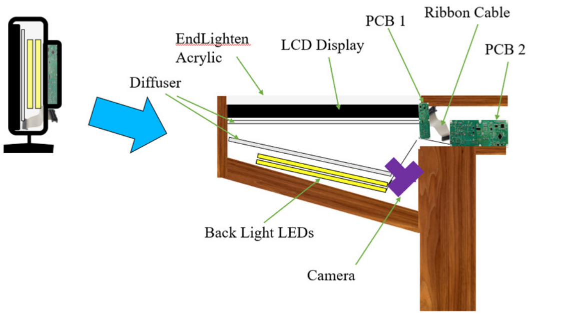

Figure 4: LCD screen hardware component relocation

This subsystem was successfully completed for our MDR deliverables. The screen was disassembled, and its parts relocated as shown in Figure 4. The screen maintained its functionality. We were able to test this by using the TV remote, sound system, and connecting the screen to a computer with an HDMI cable.

D. Block 3: Desk Structure

The technical block to be discussed in this section is the structure of the desk. Although this does not appear to be a "technical" block, it is extremely important in meeting our specifications. The primary goal of the desk is to ensure that the user will have sufficient leg room. To maintain functionality of a conventional desk, the user must be able to sit comfortably at the desk. With our proposed design, cameras must be placed below the LCD screen to capture IR-lit touch points from above. The cameras need sufficient field-of-view to cover the entirety of the screen. Using multiple cameras and concatenating their individual images prior to image processing is an option; however, concatenation is a time-costly operation which will be discussed in further detail in Section F. Consequently, our secondary goal with the design of the desk's structure is to minimize the number of cameras needed to lower the processing overhead and obtain the fastest possible response time. Our design options to be considered for the desk's structure are to have either a flat-bottomed or sloped enclosure for the cameras. Visualizations of these options can be seen in Figure 5.

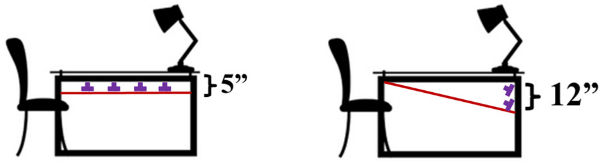

Figure 5: Flat Enclosure (Left) & Sloped Enclosure (Right)

For the flat-bottomed enclosure, we measured that it can be no more than 5 inches below the screen to maintain comfortable leg room. With this specification, we can calculate using simple trigonometry how much of the screen each camera will cover. The field-of-view of our selected camera, a PS EYE [8] with a 2.1mm wide angle lens, is 104°. Assuming the camera is about 1 inch in height, the lens of the camera is then 5" – 1" = 4" from the LCD screen. Figure 6 below clarifies the setup being considered for this calculation along with labeling specific quantities to be considered.

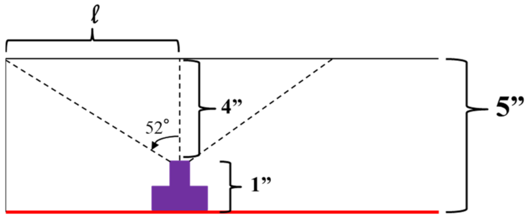

Figure 6: Diagram to determine field-of-view of one camera

The quantity labeled ℓ in Figure 6 is 5.12 inches; thus, the camera covers 10.24 inches in one direction. For simplicity of calculation, we will consider the camera having uniform coverage of 10.24" in all directions. The dimensions of the LCD screen are 28.25" length by 16.5" height; therefore, we would need 3 cameras to cover the length of the screen and 2 cameras to cover the height of the screen or a total of 6 cameras (3x2 array) for the flat-bottomed enclosure.

Next, we consider the sloped enclosure design as depicted in Figure 7. Through measurement and experimentation, we determined that the back-end of the slope can be no more than 12 inches down to maintain sufficient leg room. Knowing that the screen is 16.5" in height, we can calculate the interior angles of the slope. The setup for the subsequent calculations can be seen below in Figure 7.

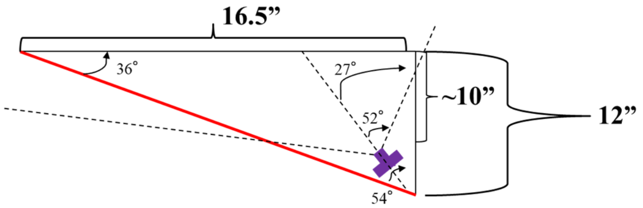

Figure 7: Diagram to determine field-of-view of one camera

If the camera sits in the slope’s angle of 54° with its normal at half of the angle, its 104° view angle will cover the entire height of the LCD screen with only one camera as shown in Figure #. To determine how much of the length of the LCD screen that one camera will cover, we will perform a similar calculation to that completed for the flat-bottomed design. The camera now sits 10” down from the screen. The camera now covers 25.6” of the screen’s length. Since the screen is 28.25” in length, 2 cameras will be needed (2x1 array) for the sloped-enclosure. From these calculations, it appears to be an easy design decision to choose the sloped enclosure over the flat-bottomed enclosure to save both the cost of 4 cameras as well as the computational cost of extra concatenations; however, there is one more thing to be considered. With a sloped enclosure, the resolution distribution of the cameras becomes much less uniform. This will be discussed in the next section to conclude whether the sloped enclosure will function to meet our accuracy specifications.

E. Block 4: Camera Resolution

To meet the specification of fingertip accuracy, the resolution of the IR cameras is critical. To reduce the number of cameras required to capture the entire space, the resolution of the cameras was affected. After researching camera resolution, it was determined that at least 2 pixels per centimeter would be required for a computer to detect and classify an object. To verify that the current design and camera configuration would meet the accuracy specification, a simulation in python was run to determine resolution distribution across the screen. The simulation was based on the set up seen in Figure 8 below, where the camera is at a distance viewing an object A.

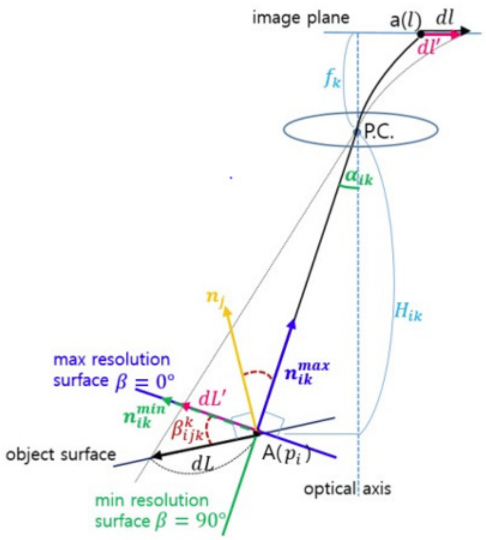

Figure 8: Configuration for Python Simulation [12]

The resolution is the ratio between the actual length of an object being viewed, dL, versus its projected length on the camera sensor plane, dl. The equation below was used in this simulation to calculate the resolution of an object being viewed at every position on the screen.

$$r = \frac{dl}{dL} = \frac{f cos\left(\frac{\alpha}{2}\right) d\alpha cos(\beta)}{dL\prime} = \frac{f cos\left(\frac{\alpha}{2}\right) cos(\beta)\, cos(\alpha)}{H} \quad (1)$$

Equation (1) was derived using the arc length equation and written in terms of α, the angle between the optical axis and the object, β, the angle of the objects orientation from normal, and H, the distance of the object from the camera lens. Since the current solution uses a wide angled lens, a model for a wide camera was used, as seen in (2) where f is the focal length.

$$dl = f cos\left(\frac{\alpha}{2}\right) d\alpha \quad (2)$$

Using Equation (1) and Equation (2), a python simulation was run to calculate the resolution at all points of the screen. After running the simulation, the areas closest to the camera have the highest resolution while the areas furthest from the camera has the lowest, as expected. The resolution distribution can be seen in Figure 9, where the two cameras are positioned at the back of the desk. The yellow areas represent the areas with the highest resolution, while the dark purple areas represent the areas with the lowest resolution.

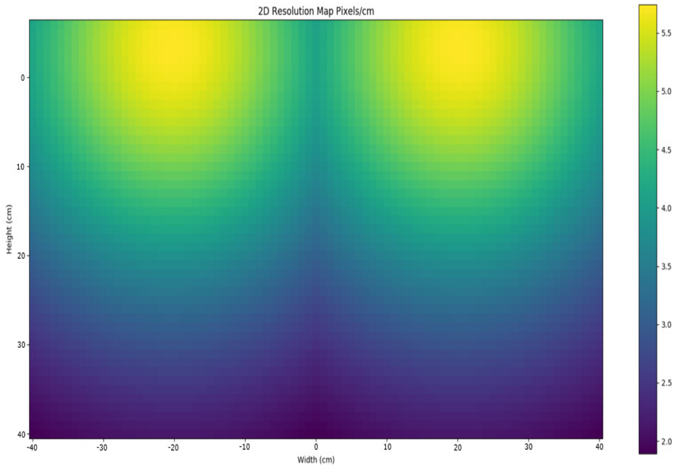

Figure 9: 2D Resolution Map

One consequence of the current configuration is the areas furthest from the user have the best resolution, whereas the areas directly in front of the user have the poorest. Since the majority of the activity, such as writing, will be done directly in front of the user, this could negatively affect the user experience. One can see, however, that this will not be an issue due to the fact that even the poorest resolution still meets the requirement of at least 2 pixels per centimeter. To enhance the user experience more, different camera configurations are currently being researched, such as employing mirrors in addition to the sloped enclosure in order to bring the areas of highest resolution closer to the user.

F. Block 5: Raspberry Pi - Image Processing

The image processing will be handled on a Raspberry Pi 3 Model B [8]. This device will be stored underneath the desk. The choice was made to offload the image processing to a Raspberry Pi instead of the user's computer so that more processing power would be available to run applications. The Raspberry Pi will be running OpenCV 3.3.1 [9], a computer vision library, and most scripts will be written in Python 3.5.3.

When motion is detected, at time t=0, the two cameras will each capture a frame. The two frames will then be concatenated into a single image using OpenCV. That image will be processed using the SimpleBlobDetector library, which allows us to transform each point of contact on the screen into a pair of (x, y) coordinates [10]. The output of the image processing is a series of those (x, y) coordinates. They are fed into a mouse driver, such as CL Eye, which converts them into a series of clicks on the computer. Figure 10 shows a detailed breakdown of this process.

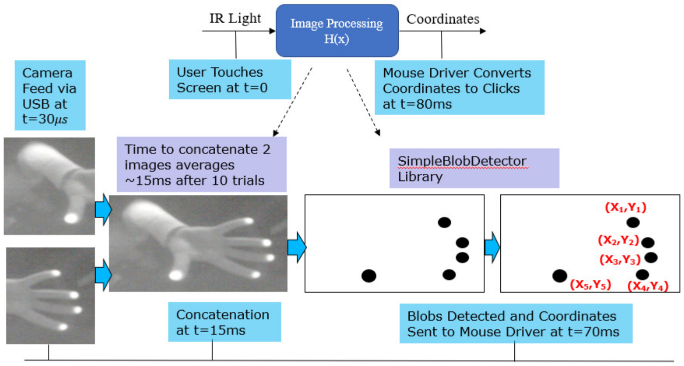
IR Light
Image Processing H(x)
Coordinates
Camera Feed via USB at t=30μs
User Touches Screen at t=0
Mouse Driver Converts Coordinates to Clicks at t=80ms
Time to concatenate 2 images averages ~15ms after 10 trials
SimpleBlobDetector Library
(X_1,Y_1)
(X_2,Y_2)
(X_3,Y_3)
(X_5,Y_5)
(X_4,Y_4)
Concatenation at t=15ms
Blobs Detected and Coordinates Sent to Mouse Driver at t=70ms

Figure 10: Breakdown of Image Processing on Raspberry Pi

Analysis of this processing technique is done via the timing methods inherent in Python. We have already implemented the image concatenation, which averaged out over ten trials to be ~15 ms for the two cameras. To further analyze the data, we must first implement the SimpleBlobDetector library, and then ultimately the mouse driver. Both tasks require further research, but ample documentation is available.

Our specification for responsiveness is less than 100ms. This benchmark was determined by considering the average human reaction time, which is displayed in Figure 11 below, which was created using over 55 million data samples [11].

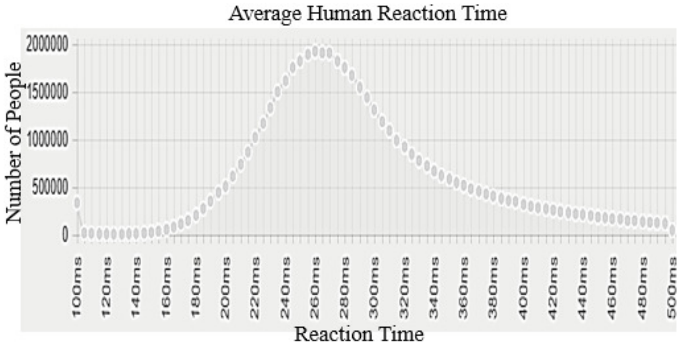

Figure 11: Average Human Reaction Time [11]

II. Project Management

TABLE I
MDR DELIVERABLES

Goal	Status/Description
Acrylic and IR LEDs installed	Accomplished: IR light fills Acrylic
LCD screen mounted with successful PCB relocation	Accomplished: LCD's PCB relocated to a non-issue location
LCD screen functioning with proper backlighting	Semi-Accomplished: LCD has proper backlighting
Camera and Raspberry Pi enclosed without prohibiting user leg room	Accomplished: sloped enclosure provides sufficient leg room
Camera captures IR-lit touch points	Accomplished: IR-lit touch points seen by camera and displayed by Pi

Table I contains our proposed MDR goals and the status of each at our MDR presentation. Our over-arching goal for MDR was to have a functioning test bed. Our definition of a working test bed is a semi-permanent structure with all of the subsystems installed and functioning as expected so that image processing and computer-desk communications can now be completed. This includes having all of the goals listed in Table I completed. We have completed all of the goals with a slight caveat on a functioning LCD screen. An accident with the fragile screen caused it to crack and no longer be functional. Despite the screen no longer displaying the proper image, it is clear that some image is displayed and that commands with the remote control change the image. This ensures us that this subsystem would be functioning properly had the screen not broken. Since we do not depend on a screen display for implementing the next few steps, we decided that there is no rush to install the new screen and risk a future accident. In Figure 12 below, a Gantt chart is used to display the major remaining tasks to be completed. Within the blue boxes on the chart are our team member's initials displaying who will be leading each portion of the project (TK – Tristan Koopman, JM – John Melloni, AF – Aidan Fitzpatrick, DM – Daniel Mathieu). Due to the nature of the project, parallelization of tasks is difficult and thus much of the work being completed will be a team effort. Those listed as leaders of each portion are simply because of their expertise or experience pertaining to that aspect of the project.

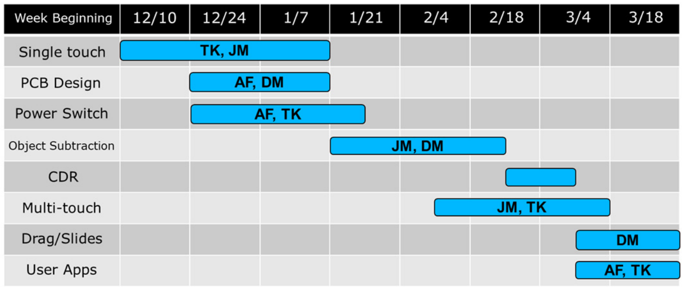

Figure 12: Gantt Chart displaying plan of future

The next component to be implemented is mapping a single touch to a coordinate on the screen with unnoticeable delay. On the hardware side, a PCB will be designed as a power supply for the IR LEDs (ideas are still being discussed for how else we could benefit from a PCB). In addition, a hardware switch will be designed to control whether the touch component of the screen is on. This switch is essential in meeting our specification of maintaining functionality of a conventional desk. Object subtraction, multi-touch, and drag touches will be handled in software to output the proper response to be completed by the mouse driver. Finally, user applications will be created to provide for a demo-friendly final product.

Team Smart Desk is extremely passionate about this project and is motivated to produce a final product that will excite both the faculty and students on Demo Day. Because of this passion, the team often constructively debates even the finest of details to enhance our project. We are confident as a team that our work ethic and communication skills will allow us to complete an impressive Smart Desk that offers a viable product that could compete in the current market.

III. Conclusion

The current state of our project can best be described as a critical turning point. We demonstrated the proof of concept for our proposed solution at our MDR presentation. This entailed embedding the LCD screen into the desk, lining the acrylic with IR LEDs, building an encasement for the cameras and backlighting, and showing that a touch on the screen produced an output that was visible through a single camera connected to a Raspberry Pi.

Much of the progress up until now has been hardware-oriented, such as physically constructing our Smart Desk. From here on out, the primary focus will shift to be software-oriented. The SimpleBlobDetector library must be leveraged to extract coordinates from the points of contact [5]. Those coordinates must be sent to the mouse driver and converted into a click on the desktop. Following successful execution of a single click, we must experiment with processing multiple blobs to allow for multi-touch capabilities. Additionally, we intend to implement drag/swipe gestures into our touch screen, such as being able to "swipe-up" to close an application. This involves performing tests with various timing thresholds to emulate longer versus shorter clicks. Once all of those features are enabled, we will then begin developing applications to display them. These applications will not only show how our Smart Desk can improve a user's organization and efficiency, but also lay the foundation for a successful Demo Day.

As far as hardware goes, there are still several advancements that must be made. The physical construction of the desk has not yet been deemed as our final prototype. Additional backlighting must be added to improve the screen's display. Alternative camera positions will be tested to determine the optimal resolution layout. Furthermore, we must finalize whether our PCB will primarily be a power supply for the LCDs, or if it will be utilized differently. Lastly, we must construct a hardware switch that will allow for the touch screen capabilities to be turned off.

We are aware that unexpected difficulties will arise, but we believe that our timeline will allow us to overcome them prior to the deadline.

Acknowledgment

We thank Professor Siqueira, Professor Holcomb, and Professor Vouvakis for research advice and general technical support.

References

[1] "Ideum Pro Touch Table". *Ideum.com. 2018.* [Online]. Available: http://ideum.com/products/touchtables?gclid=EAIaIQobChMIk6e1geeF2QIVVQeGCh1pTgDxEAAYASAAEgJWGvD_BwE [Accessed: 01-Feb-2018].

[2] "X322BV-SR LED HDTV", *Sceptre.com*, 2018. [Online]. Available: http://www.sceptre.com/store/TV/LED-HDTV/X322BV-SR-LED-HDTV-product973category1category21.html. [Accessed: 01- Feb- 2018].

[3] The 5 Types of Touch Screen Technology. Which One is Best for You?. TRU-Vu Monitors, 2017, pp. 2-3.

[4] Reach Technology, 2017. [Online]. Available: https://www.reachtech.com/. [Accessed: 18- Dec- 2017].

[5] Science, L. and Earth, P. (2017). *What Is Infrared?*. [online] Live Science. Available at: https://www.livescience.com/50260-infrared-radiation.html [Accessed 19 Dec. 2017].

[6] acrylite-shop.com, '*ACRYLITE - Shop'*. 2017, Available: https://www.acrylite-shop.com/US/us/acrylite-led/light-guiding-edge-lit-ga7iwmq7gnt.html. [Accessed: 015- Dec- 2017].

[7] "Playstation 3 Eye Camera". *Amazon.com*. 2018. [Online]. Available: https://www.amazon.com/PlayStation-Eye-3/dp/B000VTQ3LU [Accessed: 01-Feb-2018].

[8] "Raspberry Pi 3 Model B - Raspberry Pi", *Raspberry Pi*, 2018. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 01- Feb- 2018].

[9] "OpenCV library", *Opencv.org*, 2018. [Online]. Available: https://opencv.org/. [Accessed: 01- Feb- 2018].

[10] S. Mallick, "Blob Detection Using OpenCV (Python, C++) | Learn OpenCV", Learnopencv.com, 2017. [Online]. Available: https://www.learnopencv.com/blob-detection-using-opencv-python-c/. [Accessed: 18- Dec- 2017].

[11] "Human Benchmark - Reaction Time Statistics", Humanbenchmark.com, 2017. [Online]. Available: https://www.humanbenchmark.com/tests/reactiontime/statistics. [Accessed: 18- Dec- 2017].

[12] "CCTV Coverage Index Based pm Surveillance Resolution and Its Evaluation Using 3d Spatial Analysis", ncbi.nlm.nih.gov, 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4610599/ [Accessed: 8- Feb- 2018].