Gesture-Controlled LED Matrix Display

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Adam Dirting

Spring, 2024

Technical Project Team Members

Mitchell Taylor

Spencer Hernandez

Zakaria Belkhayat

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Adam Barnes, Department of Electrical and Computer Engineering

Final Capstone Project Report: Gesture-Controlled LED Matrix Display

The Pack, **ECE4440/4991** - December 5, 2023

CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. STATEMENT OF WORK

1) Adam Dirting
Adam wrote a program to utilize the time-of-flight sensors in the final light emitting diode (LED) matrix design. This program was responsible for translating the gesture and hand data given by the time-of-flight sensor and its libraries into commands for the Arduino to know which LEDs to light up on the LED matrix. This program also utilized this data to implement various functions such as multicolored drawing, screen clearing,

brush size, and more. Furthermore, he assisted in establishing communication between Microcontrollers. Adam also managed the project's budget.

2) Mitchell Taylor
Mitchell designed the buck converter and linear regulator circuits using the datasheets and recommendations from the manufacturer. He also designed the LED driver circuitry and the control circuitry using multiplexers and transistors. Near the end of the project, he designed and tested the light emitting diode matrix circuit board and the main control circuit board. This hardware served as the foundation for the project.
3) Spencer Hernández
Spencer was primarily responsible for controlling the light emitting diodes through the light emitting diode driver with the Arduino. This involved debugging the LED Driver breakout boards Mitchell designed to ensure that they could properly interface with several LEDs. Once the Arduino was able to control several LEDs, scanning was implemented to control a 3 by 3 grid of LEDs on a breadboard. After this, he helped solder several hundred LEDs onto the circuit boards and rigorously tested them once fully populated. UART was then established between the Arduino and the STM32 with Adam to ensure reliable and efficient serial communication. Since the LED boards were ultimately unusable, he assisted Mitchell in soldering new LED strips onto the final housing board that was used.
4) Zakaria Belkhayat
Zakaria helped adapt the gesture tracking code provided by STM to send output to the secondary Arduino microcontroller. He explored the provided gesture library and demo software to determine what functionality could be implemented in the final product and how. He wrote the code to convert the gesture algorithm's x, y, and z outputs to grid coordinates for the LEDs. After designing breakout boards which were unfortunately unusable, he also helped Adam test the time-of-flight sensors using breakout boards purchased from STMicroelectronics. Zak also designed the educational brochure to accompany the device. He found an attractive template layout and a host of beginner-friendly guides for each component to use as an educational aid.

II. Abstract

In this project, we offer an approach to introducing rudimentary electrical and computer engineering (ECE) concepts to young students in high school through the use of an interactive matrix of Light Emitting Diodes (LEDs). This system will utilize time-of-flight (ToF) sensors in order to determine the X, Y, and Z position of the user's hand. This data collected by the ToF sensor will be processed, analyzed, and transmitted between two microcontrollers (MCUs), allowing the user to draw and manipulate the LED display — as if drawing on a canvas — using natural gestures. Our hope is that by offering an easy to approach and engaging example of an ECE project, students will be more excited about the field.

III. Background

There are many examples of hobbyists designing their own LED matrices [1] for personal use. This can be attributed to the fact that there are a wide variety of easy to use MCUs such as the Raspberry Pi and Arduino which have extensive libraries supporting LED drivers and addressable LED strips. LEDs with high luminosity are also relatively cheap, especially on a small scale; specifically, LEDs that can light up in Red, Green, and Blue (RGB) colors. Most hobbyist's projects utilize store-bought LED displays or addressable strips due to the fact that they are easier to control than a custom matrix. However, there are other dedicated hobbyists that have built their own displays using a combination of individual LEDs, drivers, MCUs, and Printed Circuit Boards (PCBs) [2][3]. The aforementioned displays are interesting art pieces and proofs-of-concept, but ultimately lack many real-world applications. Our group believes that there are several potential applications for these devices, namely as educational tools.

An example of the use of these matrices for pedagogical purposes is shown in [4] as a fun way to engage with children, and get them excited about learning; specifically, in remote areas with minimal technology. Rather than watching their teachers draw on blackboards, students can now engage with a brightly colored display, making learning in the classroom more fun. One of the main drawbacks of current LED matrix designs is that while they may be appealing to look at, there is a lack of interaction with the students.

Additionally, using hand tracking to draw on a computer is a well-documented concept. In [5], a "Virtual Drawing Board" is shown that uses a laptop camera to sense the user's hand movements, which is then processed in a Python program. The program then draws the corresponding sketch on the drawing board. This project was not used to drive a specially designed LED matrix, but it does demonstrate the concept of using real-time motion-detection to generate a drawing.

One of the most notable and complex issues facing the ECE industry is the difficulty in preparing young students for the workforce and producing good academic outcomes in a rapidly changing technological landscape [6]. Our group believes that designing devices like this LED matrix and using them as artifacts for instruction in electronics can help improve education in this field. Where this LED matrix design differs from the aforementioned ones commonly found today is the motivation to help spread an excitement for the field of ECE to a younger generation. The basis of this design is to blend art and interaction with technology to get students engaged, and then utilize this engagement to instruct them on this field. Additionally, an educational guide has been created to accompany the device, allowing students to learn alongside their interaction. This can be seen in Section XIII-D.

Our project will allow users to use their hands to draw on an eye-catching and interesting LED matrix, creating a stimulating way for them to learn about electronics, while also giving instructors a new way to communicate and connect with students, combining concepts from [4] and [5].

A. Relevant Coursework

This project utilizes a variety of sophisticated concepts our group has learned throughout our time as ECE students. Table I shows a comprehensive list of the different courses our project is sourcing knowledge from, as well as the specific material that will be incorporated in the final deliverable.

Introduction to Embedded Computer Systems	The microcontroller processing the sensor data will be an STM32 embedded system.
Fundamentals of ECE I, II, and III	Circuit analysis and design used to construct LED matrix layout and PCB design.
Analog Integrated Circuits	Current sources and transistors to power LEDs.
Power	Designing circuitry to provide sufficient voltage and current to various circuit components.

TABLE I
LIST OF COURSEWORK UTILIZED FOR THIS PROJECT.

IV. CONSTRAINTS

A. Societal Impact Constraints

As a responsible and ethical engineer, it is important to provide a thorough examination of societal impacts and constraints in any engineering project. This project is no different, and it is imperative to analyze the unintended consequences of our project to minimize potential harmful side effects. These considerations may include cultural, economic, environmental, global, public health, safety, social, and welfare factors. Analyzing these constraints is not only a moral imperative, but also serves as a strategic safeguard against potential negligence and liability.

As previously mentioned, one of the main applications of this project is to be used in K-12 classrooms to educate students on Electrical and Computer Engineering. While we certainly believe this would be an engaging way to teach students, the project could potentially lead to worsening the digital divide, the idea that there is a growing gap in outcomes between those who do and those who do not have access to technology [7]. It has been proven that digital learning does provide better learning outcomes than traditional learning, which means that if nothing is done, students in impoverished communities will fall behind their counterparts in more urban and developed areas [8].

There are also economical and environmental factors that should be taken into account with our project. One of the initial goals of this project was to make this low-cost and low-power. Unfortunately, this project can no longer be considered low-cost due to the LEDs being through-hole and therefore more expensive. The environmental impact of our project requires a commitment to energy efficiency, which was a large consideration throughout the design process. If the display were to simultaneously illuminate 500 LEDs, the power supply would need to provide roughly 30 Amperes of current to it. Through the use of scanning, we were able to reduce the current consumption down to roughly 1.2 A, a reduction by a factor of 25.

B. Physical Constraints

Along with societal constraints, there were several points throughout the course of the project where financial limitations and manufacturing challenges influenced the project. Having to change plans and pivot is a key part of the engineering design process, and this project was no exception.

As mentioned, cost constraints were a large consideration during this project. While the budget of $500 allowed for some mistakes, it forced our group to be more conscious of our design choices so we could determine what was feasible without having to purchase parts and test them. At the beginning of the semester, we decided that the matrix would need at least 500 LEDs, which would cost around $300. Since this was a large portion of the budget, we decided to wait until just before Thanksgiving break to finally order them. This ensured that we would be able to use the other $200 to find appropriate parts that would be able to power the LEDs as well as read data from the ToF sensors. More on how the budget was utilized can be found in Section IX.

Perhaps the second biggest "physical" constraint that was taken into consideration was the timeline of this project. While we were able to complete our project within the given time, this was not without careful planning and budgeting. Throughout the semester, there would be occasions where our group would be testing a part that we ordered, only to find that there are other existing parts which would perhaps be better suited for our specific use case. In these moments, we were forced to discuss as a group if it would be more beneficial to wait a week to buy a new part, or to instead continue to use the components that we have. There were times when we decided that there were better uses of our time in the project than to try and make a certain component work, in which case we would buy a new part that we were certain would work. An example of this would be purchasing the ToF sensor breakout boards instead of debugging the breakout boards we designed. Other times, we would have to continue to use what we have because we couldn't afford to wait for a better part. Both of these cases taught us valuable lessons about the engineering design process and how crucial it is to research and plan ahead as best as possible when working on a tight timeline.

To properly develop embedded code and PCBs to house the LEDs and other components, a variety of different software was used to ensure that each one best fit the needs of the developer working on it. Our group utilized the Arduino IDE, STMCubeIDE, Microsoft VSCode, and GitHub for software development as well as Altium Designer, TinkerCAD, and KiCad for circuit design, and 3D modeling.

For PCB design, Altium Designer and KiCad were used to design initial break-out boards for surface-mount Integrated Chips (ICs), as well as the final matrix and power supply boards. Our group had to review both Altium and KiCad in order to both design functional PCBs and export them for manufacturing. To transfer files between the two pieces of software and across devices, GitHub was used to share files and convert file types.

Since the project's inception, our group decided that it would be most beneficial to divide the project using two

MCUs, and for two main reasons. Firstly, it allows for more parallel work between group members: one group member could work with the STM32 MCU to read and process data from the ToF sensors, while another could work with the Arduino Mega MCU and turn on the LEDs. This design choice not only allowed for more parallel work, but it also made the project easier to debug, since each system could be tested individually from one another and then integrated together. Another big reason for this is the ToF sensors; the ToF sensors are designed by ST Microelectronics (STM), so it made most sense to use a readily-compatible MCU that could communicate with them. One of our concerns was that with having to not only read and process sensor data, but also toggle LEDs with minimal latency, that the STM32 may not have the computational power to handle both tasks at high speeds. Our group discussed the idea of utilizing parallel processing on the STM to use the most processing power as possible, but ultimately ended up siding with using another MCU for a more modular and debug-friendly design. This decision ended up being extremely beneficial, as the Arduino MCU has a free library that can be used to control the LED drivers with minimal overhead. While we initially intended on building our own library, we ultimately had to use the preexisting library due to allow for more time to debug and test the design of the LED display.

V. External Standards

For this project, we plug our matrix into a standard North American wall outlet. This means we have to abide by National Electrical Manufacturers Association (NEMA) 5-15 standards, which means we must have the proper 3-pin plug. Following this standard, we are limited to a 15A current draw at 120V for our power supply. In addition, we abide by IPC-2221 standards in the design of our PCB, following standard trace widths and through hole sizes which allow for simple verification of our design and increased final quality. To ensure, safe and ideal operation of our design, we abide by National Fire Protection Association (NFPA) code 79, which provides the standards for wire gauge ampacities.

VI. Intellectual Property Issues

There are not many patents matching our project to a high degree, but there are some that overlap or detail very similar systems. Below is a breakdown: H

1) US11175741B2 [9]: This patent details "frameworks, devices and methods configured for enabling display for facility information and content, in some cases via touch/gesture controlled interfaces." It consists of 19 claims, 3 of which are independent. Claim 1 (independent) is in relation to a method for displaying and traversing information about a building on a display, and claims 2 through 13 (dependent on claim 1) are related to implementing multiple displays or gesture/touch control with hands or objects. Claim 14 (independent) addresses a specific computer system with two displays and interactivity, while claims 15 through 18 (dependent on claim 14) address variations and extensions on this claim. Claim 19 (independent) addresses the same methods as Claim 1, except including a superimposed image on the display. The patent is essentially for displaying information in a format that can be traversed by a user to show them things like directions or quick info. This patent is broad but includes gesture controls. However, it does not mention any form of drawing or specifically using an LED matrix.
2) US10831281B2 [10]: This patent is for methods of interacting with user interfaces using gestures. It consists of 8 claims, 3 of which are independent. Claim 1 (independent) addresses a method of differentiating gestures in a 3D space. Claims 2 and 3 (dependent on claim 1) detail using individual fingers for gestures, as well as a scaling function for translating gestures. Claim 4 (independent) is a claim to a method for translating these gestures in augmented reality. Claims 5 through 7 (dependent on claim 4) address specific gestures in relation to spatial planes. Claim 8 (independent) addresses a method for rejecting malformed or incorrect gestures. This is more of an "overlapping" patent, most closely mirroring the gesture control portion of the project. This patent does not cover the interfaces or devices being controlled, but the method of gesture control itself.
3) US11353962B2 [11]: This patent is similar to US10831281B2, concerned with methods of gesture control using simulated virtual objects. It consists of 21 Claims, 3 of which are independent. Claim 1 (independent) is the bulk of the patent, addressing user interface control using a specific control object in free space. Claims 2 through 19 (dependent on claim 1) are specific methods for gesture control using the control object. Claim 20 (independent) addresses a system including the gesture control method. Claim 21 (independent) details a storage medium for logging and transferring gesture control data. Again, this patent does not detail specific display or interface devices, which our project does explicitly include.

The most relevant patent to the project is US20210117011A1 [12], which is the patent for STM's STGesture system, the gesture recognition and hand-tracking algorithm used in this project. This patent is a claim to their system of using their ToF sensors in conjunction with an algorithm and software for a turnkey gesture control solution. It comprises 20 claims, 4 of which are independent and 16 of which are dependent. The independent claims address the algorithm and the ToF sensor's processing capability, memory, and compatibility in a larger system. The dependent claims address methods which are variations on using these different aspects of their algorithm and ToF sensor.

According to 35 USC §103, a patent cannot be granted if it is "obvious" prior to patent filing. Considering STM has provided their own patented gesture recognition for the express purpose of enabling projects like ours, it is easily argued that our specific device is too "obvious" to patent. In any case, the intention of the project is to produce a device for educational purposes by exposing all the work done to create it, therefore patenting the device would be contradictory to our goals.

VII. Project Description

A. Performance Objectives and Specifications

The device is designed to work in real time, with minimal latency between user interaction and feedback on the LED matrix. There is also precise tracking of the user's hand, to allow for precise designs when needed. There is robust gesture control for interacting with the device naturally, i.e. behaving similar to real drawing or writing. Additionally, a pamphlet reviewing the various electronics and ECE concepts utilized in this deliverable, and their application, is included to assist instructors in educating students on these concepts (Section XIII-D). Previously, this pamphlet was not to be included in the final deliverables; however, we decided to add it to as a deliverable after deciding to gear our project towards ECE pedagogy.

B. System and Component Designs

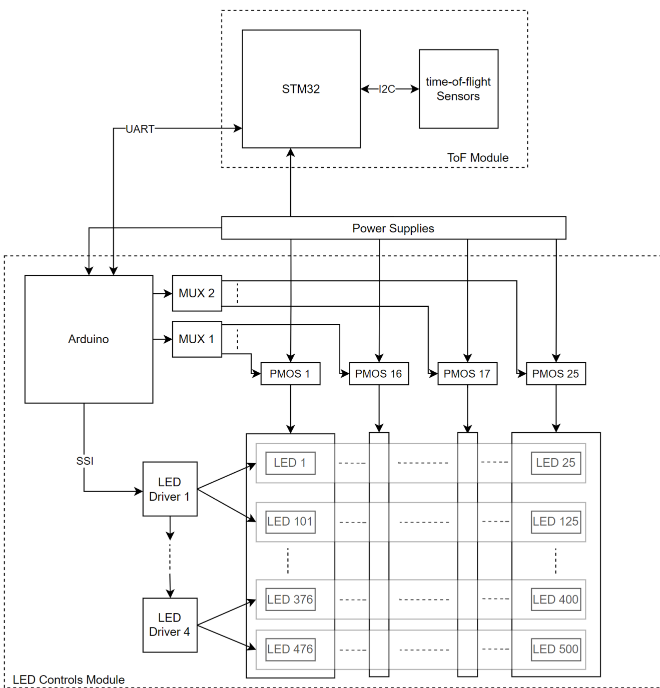

Fig. 1. System level diagram for the LED Matrix Project

As shown above in Figure 1, our system works as two separate, modular parts: the ToF Module (Section VII-B1) and the LED Controls Module (Section VII-B2). These modules each have their own software and hardware architectures.

1) Time of Flight Module: This module is responsible for reading the data from the ToF sensors, interpreting this data, and then packaging and sending it to the LED Controls Module over Universal Asynchronous Receiver/Transmitter (UART). Figure 5 above shows an overview of how the software written for the STM32 MCU accomplishes this. After all peripherals are initialized, the software then continuously reads data from the four ToF sensors on the board, one at a time, and processes the data using various software packages provided by ST. After doing so, the program then interprets this data and determines how to alter the LED Matrix (i.e. which LEDs to turn on, and which to turn off) based on the user's hand position, and gesture. Throughout this process, the STM32 MCU keeps track of the state of each LED in the matrix; doing so, it recognizes when there is a necessary change to make on the matrix. When this occurs, a command is sent to the Arduino MCU over UART, this command tells the Arduino: which LED to turn on/off, and which RGB values to supply it in order to make it a certain color.

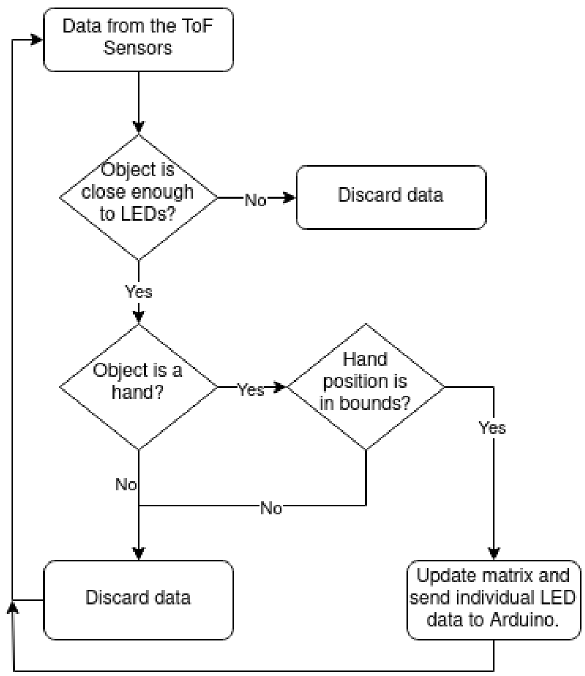

Fig. 2. Software process diagram for the Time of Flight Module, after initialization.

For the communications with the Arduino MCU in the LED Controls Module, the UART protocol was chosen. For this particular communication protocol, only two wires are required between the modules: Transmit (Tx), and Receive (Rx). The Tx of one MCU is connected to the Rx of the other, and vice versa. Furthermore, the UART protocol utilizes "frames" which consist of 10 bits — 8 bits for data, 2 bits for start/stop acknowledgement. A diagram for one UART "frame" can be seen below in Figure 3.

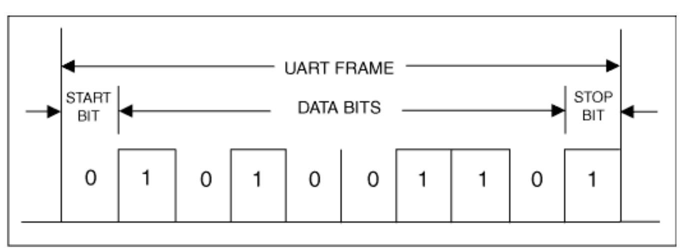

Fig. 3. Composition of one UART frame, obtained from [13].

In terms of how we will utilize this protocol, multiple frames will be used to construct a "packet". A diagram for this packet architecture is shown below in Figure 4. Frame one and two consist of the row and column number, respectively, for the LED in question — the range for possible row frame values is 0 to 19, and 0 to 24 for column values. Lastly, frames three, four, and five will be used for the red, green, and blue codes, respectively — these values each range from 0 to 255. In total, one of these packets is a total of 50 bits.

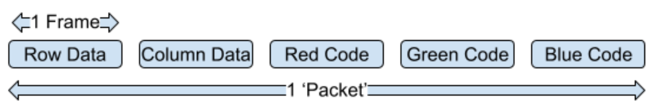

Fig. 4. Composition of the packets sent over UART for LED control.

2) LED Controls Module: The LED module is responsible for toggling the individual lights in the matrix of LEDs, and reading in data from the time-of-flight module. This module is controlled by the Arduino Mega MCU, which uses SSI to communicate with four LED drivers. Each LED driver has 16 channels, meaning each driver can control 5 4-pin RGB LEDs for a total of 20 LEDs. Given that the actual matrix has 500 LEDs (20 rows by 25 columns), the Arduino refreshes each column one by one at a high enough refresh rate that is indistinguishable to the human eye. As it's powering each row, the corresponding LED data for that respective row is being sent to the LED driver. In this configuration, the matrix consumes significantly less power and only 4 LED drivers need to be used.

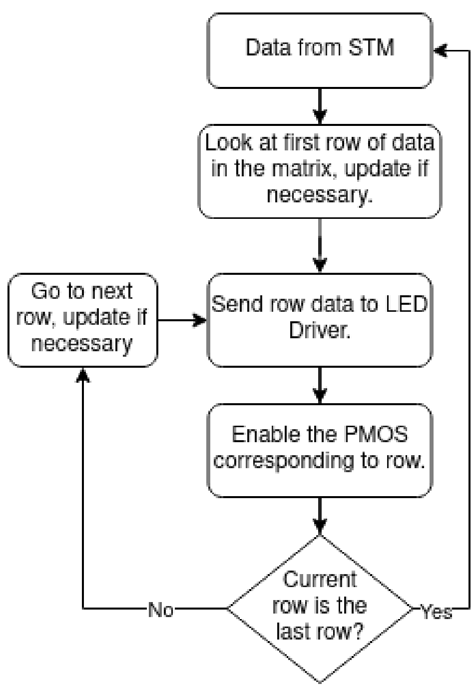

Fig. 5. Software process diagram for the Time of Flight Module, after initialization.

3) Power Module: Given the quantity of LEDs being used in this project, the matrix is supplied power from the wall at 120VAC. That voltage is stepped down to 12VDC using a prefabricated switching regulator and fed into a buck converter for a 5V Voltage Common Collector (VCC). To power the Time of Flight sensors, the 5V was fed into a 3.3V linear regulator.

C. Technical Details

When engineering this project, we decided to split the design into two separate modules, the main control PCB and the matrix PCB. The main PCB houses the power supplies, the STM32, the Arduino connections, and the LED control circuitry. The matrix PCB houses the LEDs and the LED drivers, as well as the necessary bypass capacitors and pull-up resistors.

Originally, we planned to use a Raspberry Pi Pico as our secondary micro-controller designated for controlling the PCB, but we ran into multiple issues involving interfacing with the drivers from the Pico. The Pico cannot supply high enough General Purpose Input/Output (GPIO) voltage to latch the LED drivers and the timer interrupts on the Pico were found to be difficult to configure. We fixed this problem by replacing the Pico with an Arduino Mega. There were a few tradeoffs with this switch, mainly cost and space. The Arduino is much larger than the Pico and cannot fit onto our PCB which was already designed for manufacturing at the time of the switch. As such, we did waste a portion of our budget on the unused Pico, but were thankfully able to acquire the Arduino for free. We also wasted a portion of our budget purchasing custom breakout boards for individual ToF sensors. We were unable to properly solder the components due to their size and surface mount package, so we resorted to purchasing pre-made breakout boards from STMicroelectronics.

1) Power Supplies: Estimates made with conservative values found in the datasheets for the Arduino MEGA [14], STM32 [15], Time of Flight Sensor [16], the LED drivers [17], and the LEDs [18]:

Component	Current (mA)	Voltage (V)	Power (mW)	
Arduino MEGA	70	5	350	
STM32	100	5	500	
ToF Sensor (x4)	24	3.3	320	[3ex]
RGB LED (x60)	40	5	12000	
LED Driver (x4)	25	5	500	
Mux (x2)	1.8	5	9	
Total	2.77A		13.68W	

TABLE II
POWER CONSUMPTION BY COMPONENT

The current calculated power draw of the LED matrix and control circuitry has been estimated to be roughly 13.68 Watts (W. This value falls well within the 1800W limit of a wall receptacle. To supply sufficient current and voltage to the components on the PCB, a 12VDC, 150W switching regulator is connected to the wall outlet supplying 120VAC [19]. The 12V output is connected to a DC jack on the Arduino and a 5V buck converter on the main PCB rated for up to 5A [20]. The 5V is used to power the STM32, multiplexers (muxes), P-channel metal-oxide-semiconductor (PMOS) transistors, LED drivers, and the LEDs. The 5 Volt (V) supply is also connected to a 3.3V linear regulator to power the ToF sensors, which only draw 24mA when ranging, meaning they draw significantly less than the maximum 1.5A output current of the regulator [21].

2) LED Control Circuitry: The LEDs are to be driven using two multiplexers connected to PMOS drivers for column control and LED drivers to control rows. The matrix is designed in a row-cathode, column-anode topology, and it will scan to reduce LED ghosting as well as power consumption [22].

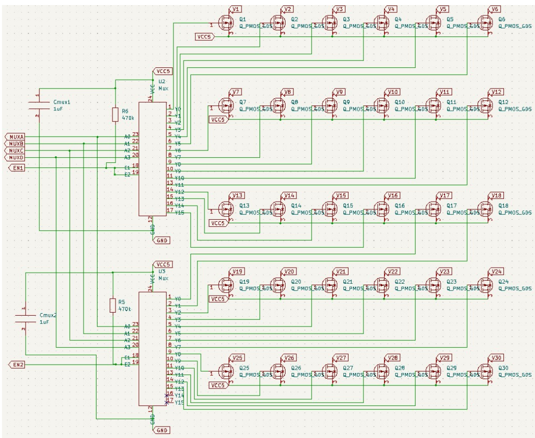

Fig. 6. Multiplexer Schematic

The multiplexer circuitry is depicted above in Figure 6. The 4:16 mux is driven by the Arduino using an active-low enable pin and four active-high inputs to be decoded to 16 different 5V outputs [23]. The IC takes a 5V VCC, but it latches high at a minimum of 2V and low at a maximum of 0.8. Controlling the muxes with the Arduino Mega, the input voltages to the mux are 5V. The outputs are driven low when selected and driving the gate of a p-type metal–oxide–semiconductor field effect transistor (PMOSFET) to act as a switch between 5V VCC and the common anode of the LED. The PMOS transistors are connected to inputs to the LED matrix are designed for power applications, and they can handle up to 60V and 14.5A, more than sufficient for a maximum column current of 2.4A [24].

3) Arduino MEGA: To interface with the matrix, a microcontroller is needed to control the LEDs and drive the outputs to the desired voltage and current levels. To achieve this, an Arduino MEGA has been implemented in conjunction with the STM32 microcontroller, communicating through UART protocol. The Arduino MEGA is used to drive the correct LEDs to the desired color.

4) STM32: The Nucleo-F401RE board comes equipped with a host of functions, including Inter-Integrated circuit (I2C) and UART capabilities [15]. Using I2C, the STM can read the data outputs from the ToF sensors. The STM can then compute which LEDs should be on and communicate that information to the Arduino MEGA via UART. The MEGA also controls the columns of the matrix through the use of MUXes and PMOS transistors. There are two MUXes connected to the MEGA that drives outputs low when they are written to using the four input "bit" channels (as long as the enable input is driven low). When a channel is driven low, the PMOS transistor connected to the output will supply power from the power supplies to the desired column. The MEGA iterates through the channels of the MUXes so rapidly that, although only one row is on at any given time, it appears that all of them are on.

5) Control Board: The control board contains the PMOS transistors, MUXes, buck converter, and power supply connections for the entire matrix. It also routes all necessary MCU pins and power traces for any external connections. The control board is shown below in Figure 7.

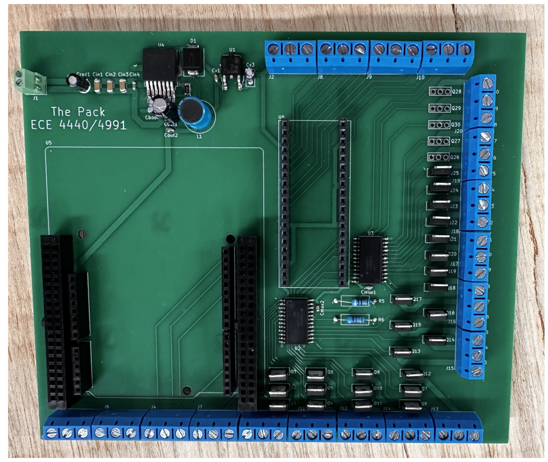

Fig. 7. Control Board for the LED Matrix

6) Time of Flight Sensors: Time of Flight sensors are used to track hand motion and recognize gestures using a gesture recognition algorithm provided by STM. Time of flight sensors function using the time of flight principle, where an array of LEDs creates infrared light and the reflected light is compared to the original signal. A simplified diagram of the process is shown below in Figure 8 [25]. Using an array of LEDs divided into sections, a single time of flight sensor can track the position of an object in front of it as well as recognize shapes and specific gestures[16]. The ToF sensors used in this project (STM VL53L5CX) have a maximum ranging distance of 400 cm, and a minimum of 40 centimeters (cm) in ambient light ranging a poorly reflective target. By utilizing 4 array-based time of flight sensors, we can cover a grid of 16x16 sections, or 256 total zones mapped to our 500 LEDs. This provides sufficient granularity and precision when logging the x, y, and z hand coordinates using STM's tracking algorithm.

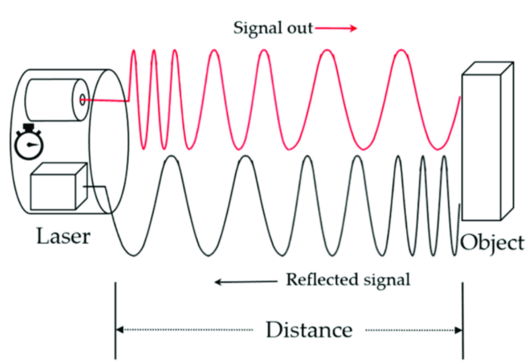

Fig. 8. Physical Diagram of Time of Flight Sensor Mechanism

7) Light Emitting Diodes: The LEDs used are through-hole, with 3 different colored LEDs in each unit, lighting up red, green, and blue and sharing a common anode. Their maximum current is 120 milliamps (mA), which is also the limit of the LED Drivers we used. Their minimum brightness is 1500mcd on the blue LED, which is plenty for this purpose [18].

8) LED Drivers: The TLC5940 LED Drivers are used to supply current to the LEDs in the matrix. These drivers are able to communicate with the MEGA through the Serial Peripheral Interface (SPI) protocol, which makes for straightforward communication. Each driver has 16 channels, each with a maximum output current rating of 120 mA, however the LEDs only require 20 mA nominally to function properly [17]. Thankfully, the driver includes an output to attach a current limiting resistor to limit the max current output of each channel, and is given by $I_{max} = \frac{V_{REF}}{R_{IREF}} * 31.5$. Since only 20 mA are required for the LEDs, the max output is doubled as a conservative estimate of necessary power. Therefore, to force $I_{max} = 40mA$, R_{IREF} must equal 910Ω since $V_{IREF} = 1.24V$ is given. The drivers have two different programming modes: dot correction and grayscale. Dot correction is used to individually adjust the percentage of max current output for each channel, and is given by $I_{OUTn} = I_{max} * \frac{DCn}{63}$. Since this is generally only needed for calibration purposes, it was not used in the final design. The grayscale mode is used to adjust the brightness of each of the 16 channels using Pulse-width modulation (PWM). The brightness of a channel is given by $\%Brightness = \frac{GS_n}{4095} * 100$, and is toggled by sending a 12 bit number for each channel. Since each driver can only control 16 LEDs, one channel was attached to individual R, G, and B pins of the LEDs so that all three values can be toggled. As such, we needed 4 drivers to drive 20 LEDs.

9) LED Matrix: The matrix is designed in a row-cathode, column-anode topology, and it will scan to reduce LED ghosting as well as power consumption [22]. An example matrix in row-cathode, column-anode topology is found below in Figure 9.

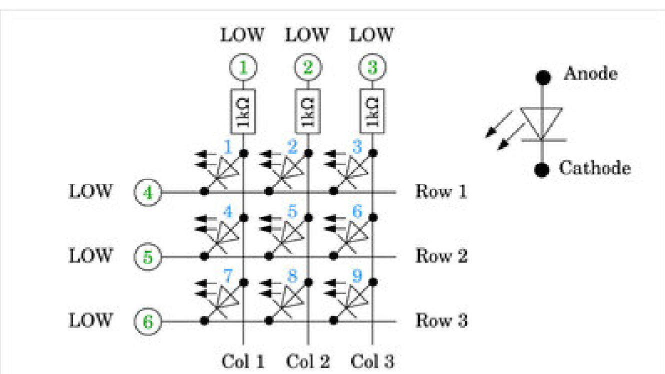

Fig. 9. LED Matrix Configuration

The boards our group created were specifically designed in sections that could be linked together in individual squares or "quadrants". We used 18 American Wire Gauge (AWG) wire for powering the anode columns due to the high power passing through these traces. We used 22 AWG wire for the cathode rows. The LED Drivers are attached to the boards and are linked to other quadrants similarly to the LEDs themselves. Figure 10 below shows the final PCB populated.

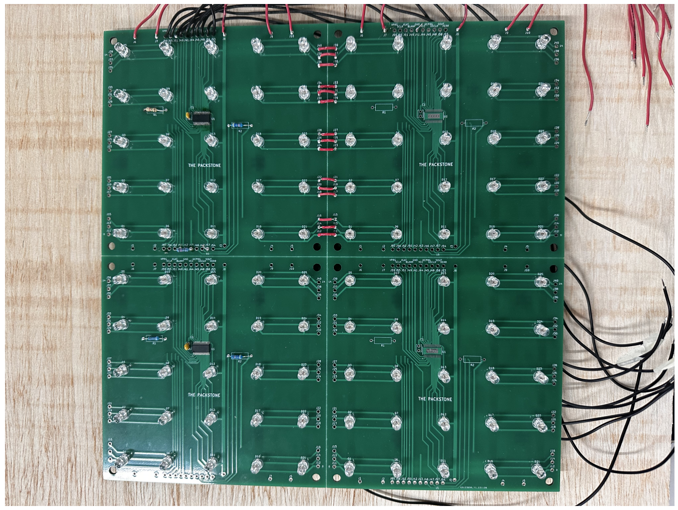

Fig. 10. Final LED Matrix PCB

D. Test Plans

In order to conclude that this system functions at a level our group is satisfied with, the project needed to be tested at individual steps, and then tested again once these steps are integrated with one another. This layer-by-layer testing plan began with a proof of concept 3x3 LED matrix, and finished with a fully fleshed-out 25x20 LED design. Ideally, the tests performed on one layer of the project could be reused on higher layers with minimal alterations to test plans (i.e. first figuring out how to control 1 LED and then scaling up to a 3x3 grid, then 10x10, etc.).

The modules are as follows: LED Matrix, ToF Sensors, MCU Communication, and the Power Supplies. Each module needed to be tested separately, and then together. Test plans are, roughly, as follows:

- LED Matrix
    1) Test signals were sent to every LED in the matrix, in order to display correct performance of all combinations of red, green, and blue (RGB) light (i.e. Rxx, RGx, xGx, etc.). Tests were verified by visually inspecting the board for each signal, and ensuring that the entire matrix is displaying the same color. Additionally, this test procedure was run on each individual row in order to ensure that they were correctly operating independent of one another (i.e. the rows aren't just mimicking the values of adjacent rows).
    2) Next, in order to test the interface between the matrix and the MCU, a variety of test programs were designed for the MCU and run on the matrix. These varied between programs that will change the color of the entire screen at a specific frequency, to lighting up specific shapes and having them move around the screen. Once again, these results were confirmed visually.

3) We unfortunately encountered many issues during this test phase with our LED Matrix PCB. These problems and test failures caused us to switch to a set of addressable LEDs, which did achieve test success.

- ToF Sensors
  1) Initially, each sensor was tested with a variety of very rudimentary tests in order to confirm basic operation. These tests simply lit up an LED if they detected motion, and objects within certain ranges. This initial test was used to ensure the sensor is operating properly before moving onward to more rigorous testing.
  2) Once the sensors were confirmed to be functioning properly, an evaluation software pack made by ST for the ToF sensors was used in order to ensure they are correctly reading data from each of the 64 zones. These results were confirmed visually by analyzing the output data of the software kit. Additionally, this software kit was used to ensure that the gesture recognition functions properly.
  3) Previously, this system was going to be tested based off of an old sensing architecture which consisted of both ToF sensors and Hall Effect sensors. Since Hall Effect sensors are no longer a part of this project, changes were necessary for this particular test plan. We also had to switch our STM platform to ensure we had enough flash memory for the gesture algorithms, but testing continued smoothly after this.
- MCU Communication
  1) In order to ensure that the MCUs are correctly relaying the input data from the positioning sensors to the LED Matrix, a basic test program was written on the STM32. This program simply lit up an LED wherever the ToF sensors signified that the user's hand was on the XY plane. This output was confirmed visually, and matched with the actual position of the user's hand to ensure it is correctly communicating the XYZ location of the hand, and lighting up the necessary LEDs.
  2) Part of the original communication tests was designing a cursor which changed size, but this was removed to prioritize the drawing function of the matrix.
- Power Supplies
  1) To begin testing, the output of the power supply circuitry was tested to verify functionality of the expected 5V and 3.3V outputs.
  2) To safely ensure sufficient power was delivered to the LEDs, positioning sensors, and MCUs, power draw for each block was calculated and then tested experimentally using the National Instruments VirtualBench configured with low current limitations to prevent accident over-current conditions.
  3) Once the safe operation of each subsystem was tested, the MCUs were connected to the power supplies and tested.

VIII. Timeline

Figures 14, 15, and 16 in Appendix XIII-A shows the initial timeline for our group's project, in the form of a Gantt chart, made at the beginning of the semester. The division of labor was decided in a way that allows for group members to work in parallel with each other, preventing bottlenecks from developing and progress being hindered. With every engineering project comes setbacks and changes that must be made, meaning that the Gantt chart was also changed quite a bit throughout the course of the semester. Some of these setbacks included needing to change from using the Raspberry Pi Pico to the Arduino MEGA due to the limited capabilities of the Pico. We also needed to buy a different STM MCU than was provided to use because there wasn't enough flash memories for our purposes. This ended up costing our group a week of developing time because there was no further work that could be done at the time with the time-of-flight sensors. The final and updated Gantt chart can be seen below in Figures 17, 18, and 20 in Appendix XIII-B.

IX. Costs

For this project, a total of $500 was given from the ECE department. In this section, how this funding was utilized and what additional funding was required will be analyzed. First and foremost, out of the $500 offered by the department a total of $499.43 was used; in other words, we were able to utilize 99.9% of our budget. Unfortunately, this was not enough funding for the project, and we needed to spend $240.97 of our own funds. This funding breakdown can be seen below in Figure 11. After all costs incurred, the total price for this project was $740.40. More details on individual items purchased can be found in Section XIII-C

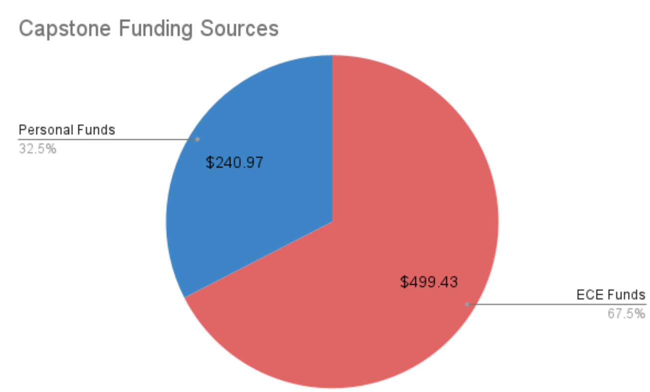

Fig. 11. Breakdown of Funding Sources for the LED Matrix Project

Out of the total amount spent, only a certain portion was used in the final deliverable. The rest was used throughout our Research and Development Process (R&D). R&D typically included: testing software on sensors, testing out different hardware components, and designing breakout PCBs. In the end, the funds spent on R&D totaled $138.72, leaving $601.68

used in the final deliverable. Percentages on this breakdown can be seen below in Figure 12.

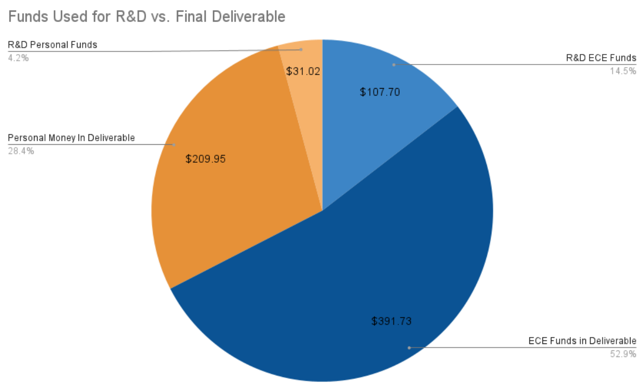

Fig. 12. Breakdown of Money used for Research and Development, or in the Final Deliverable.

As mentioned previously in Section VII-B, this entire project can be separated into three separate modules: the ToF Module, the LED Controls Module, and the Power Module. As such, the effective costs of each module can be interpreted. Figure 13 below shows the cost of each module. The costs in the module exclude only one item grouped under "other": a light diffusing sheet that will be used to conceal the LED PCBs, and make the LEDs easier to view. That being said, the total of all three modules was $729.02.

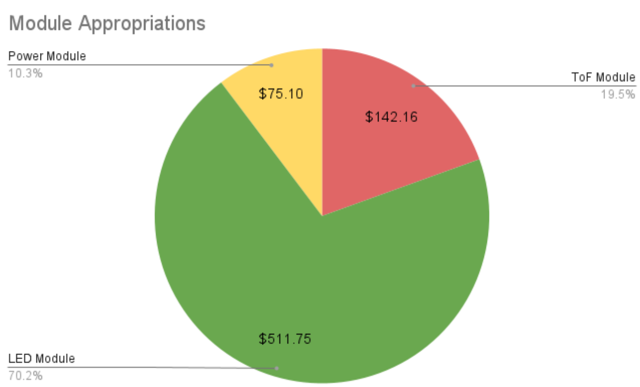

Fig. 13. Breakdown of Funds Used for each Module

By and large, when looking at the prices of most components used in this project on DigiKey and Mouser, their costs greatly decrease when ordered in quantities of 1,000 or more (by at most a factor of 4.7) — for the purposes of the estimations that follow, a factor of 3 will be used, to account for differences in this factor. Additionally, given the fact that the final deliverable only cost $601.68, it can be estimated that the parts required for the final deliverable will cost roughly $\frac{\$601.68}{3}$, or, $200.56 — this price is likely even lower, as most components used were on breakout boards, which significantly increases cost. Since this estimation only includes parts, realistically the cost to produce one LED matrix, en masse, will cost at most $50, bringing the total cost for one matrix to $250. In reality, a mass-produced LED matrix would utilize surface-mount components as well, which could reduce this price to $200, or as low as $150.

X. Final Results

Our group was able to design hand tracking and communication code for the STM32 and Arduino MEGA to allow for hand positioning data and gestures to be sent to the Arduino to light up specific LEDs. We also designed a brochure detailing each aspect of the original project with links to good learning resources for students, seen in Section XIII-D. Due to unexpected parasitics present in the full PCB for the LED matrix, we were unable to get the proposed design for the 25x20 panel working. We also had to switch from UART to I2C communication between the STM32 and Arduino MEGA, because UART is not sufficient for high speeds. We decided in the end to switch to addressable LED strips, since they wouldn't present the same issues with parasitics and needing to interface with individual LED drivers.

In terms of our rubric, we believe we met enough criteria for an A- grade, despite the many challenges and setbacks we faced. We only have one robust mode as of this report, putting us in the C+ range for that specific aspect, but we have high-precision, minimal latency tracking far surpassing our expectations, within the A+ range from our self-made rubric. There are also only some small bugs with the current code, firmly in the B+ range. Overall, this puts us in the neighborhood of a low A- by our own rubric.

XI. Future Work

There were a few key mistakes our group made during this project. One major mistake was the design of custom breakout boards for our ToF sensors. We believed it would be more cost-effective to purchase the sensors and separately buy PCBs to hand-solder them, but it turned out to be near-impossible given their surface-mount design. We instead discovered that STM sells 2-packs of sensor breakout boards for an equivalent price, when we initially thought they were single packs. We also encountered issues using a NUCLEO-G071RB, which does not have enough flash memory for STM's gesture tracking algorithms. We had to switch to an F401RE model, which solved most of the issues we had with gesture tracking and made implementation much more straightforward. The Raspberry Pi PICO microcontroller we set out to use for controlling the LED drivers also presented issues in terms of GPIO and interrupt timers, but switching to an Arduino MEGA with extensive library support proved to be effective in resolving these problems. We also realized that UART is insufficient for an application like this due to its asynchronous nature, and opted to use I2C to communicate with our sensors and the MEGA simultaneously. Our biggest stumbling block was probably the parasitics on our matrix PCB. We couldn't effectively measure the issue, and instead resolved to reduce the scale of our matrix in order to avoid the problem altogether. In the end, we switched to addressable LEDs to ensure a product that was more appealing and representative of the work we did. An extension on the project or enhancement should take into account the issues scaling can cause in terms of effective power delivery and accessing individual LEDs without parasitic capacitance. The design of a case for the display was also a challenge due to its sheer size. Most 3D printers don't have a large enough print bed for something like our matrix, and filament costs would be exorbitant. Instead, we opted for a simple case made of plywood, which is more cost-effective and straightforward.

The biggest lesson our group learned was that it is almost always more beneficial to use pre-supported combinations when creating a large design project, especially on a short timescale. For example, in our project, the time of flight breakout boards and the PICO vs the MEGA. These changes could have saved us a lot more time if they had been made earlier in the process. We also floundered in the beginning since our initial idea wasn't very motivating, so we ended up having to pivot several times. Future groups should be sure to start with an idea they like and stick to it.

XII. References

[1] *Led matrix wave moves with hand*. en. URL: https://www.hackster.io/Mostafa_Abcreno/led-matrix-wave-moves-with-hand-183166.

[2] B Babu et al. "Wireless Based LED Dot Matrix Message Display". In: *IJRAR* 6.1 (2019). URL: https://www.ijrar.org/papers/IJRAR1ACP037.pdf.

[3] H. Bouazza et al. "New and low cost product solution for LED matrix display". In: *2016 International Conference on Engineering & MIS (ICEMIS)*. Agadir, Morocco: IEEE, Sept. 2016, pp. 1–5. ISBN: 9781509055791. DOI: 10.1109/ICEMIS.2016.7745301. URL: http://ieeexplore.ieee.org/document/7745301/.

[4] M.A. Kader et al. "LED matrix based digital learning display for children with wireless control". In: *2014 17th International Conference on Computer and Information Technology (ICCIT)*. Dec. 2014, pp. 397–400. DOI: 10.1109/ICCITechn.2014.7073134. URL: https://ieeexplore.ieee.org/document/7073134/.

[5] Prof. Supriya Telsang et al. "Virtual drawing board(Drawing and writing using hand gestures)". In: *International Journal for Research in Applied Science and Engineering Technology* 10.12 (Dec. 2022), pp. 359–361. ISSN: 23219653. DOI: 10.22214/ijraset.2022.47873. URL: https://www.ijraset.com/best-journal/virtual-drawing-board-drawing-and-writing-using-hand-gestures.

[6] David Valiente et al. "Analysing students' achievement in the learning of electronics supported by ict resources". en. In: *Electronics* 8.3 (Mar. 2019), p. 264. ISSN: 2079-9292. DOI: 10.3390/electronics8030264. URL: https://www.mdpi.com/2079-9292/8/3/264.

[7] Rowena Cullen. "Addressing the digital divide". In: *Online Information Review* 25.5 (Jan. 1, 2001). Publisher: MCB UP Ltd, pp. 311–320. ISSN: 1468-4527. DOI: 10.1108/14684520110410517. URL: https://doi.org/10.1108/14684520110410517 (visited on 11/26/2023).

[8] Bulent Tarman. *THE DIGITAL DIVIDE IN EDUCATION 1*. 2003. URL: https://files.eric.ed.gov/fulltext/ED508213.pdf.

[9] John D. Morrison and Graeme Laycock. *Frameworks, devices and methods configured for enabling gesture-based interaction between a touch/gesture controlled display and other networked devices*.

[10] Hua Yang et al. *Systems and methods of free-space gestural interaction*.

[11] Raffi Bedikian et al. *Free-space user interface and control using virtual constructs*.

[12] Carlo Crippa and Rossella Bassoli. *Frameworks, devices and methods configured for enabling gesture-based interaction between a touch/gesture controlled display and other networked devices*.

[13] *Figure 3.12-UART frame, 8 bits, 1 stop bit*. ResearchGate. URL: https://www.researchgate.net/figure/UART-frame-8-bits-1-stop-bit_fig26_324792444 (visited on 11/28/2023).

[14] *Arduino Mega 2560 Rev3 Product Reference Manual*. Jan. 2023. URL: https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf.

[15] *STM32F401xD STM32F401xE Datasheet$_2$015*. Jan. 2015. URL: https://www.st.com/resource/en/datasheet/stm32f401re.pdf.

[16] *VL53L5CX Datasheet*. Mar. 2023. URL: https://www.st.com/resource/en/datasheet/vl53l5cx.pdf.

[17] Texas Instruments. *TLC5940 16-Channel LED Driver With DOT Correction and Grayscale PWM Control*. Texas Instruments, Nov. 2015. URL: https://www.ti.com/lit/ds/symlink/tlc5940.pdf?ts=1701803675049.

[18] *NTE30153 thru NTE30156 Discrete RGB LED Indicators*. URL: https://www.nteinc.com/specs/30100to30199/pdf/nte30153_56.pdf.

[19] Mean Well. *150W Single Output Switching Power Supply*. Mouser, Oct. 2020. URL: https://www.mouser.com/datasheet/2/260/mwec_s_a0011714497_1-2274579.pdf.

[20] Texas Instruments. *TPS56624x 3-V to 16-V Input Voltage, 6-A Synchronous Buck Converter in a SOT-563 Package*. Texas Instruments, Apr. 2022. URL: https://www.ti.com/lit/ds/symlink/tps566247.pdf?ts=1701834877210&ref_url=https%253A%252F%252Fwww.google.com%252F.

[21] ST. *Ultra low drop BiCMOS voltage regulator*. ST, Aug. 2017. URL: https://www.st.com/en/power-management/ld39150.html.

[22] Wan-Fu Huang. "Designing a display unit to drive the 8×8 LED dot-matrix displays". In: *2013 IEEE 5th International Nanoelectronics Conference (INEC)*. 2013, pp. 385–388. DOI: 10.1109/INEC.2013.6466054.

[23] Texas Instruments. *High-Speed CMOS Logic 4- to 16-Line Decoder/Demultiplexer*. Texas Instruments, June 2004. URL: https://www.ti.com/lit/ds/symlink/cd54hc4514.pdf?ts=1701798484090&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCD54HC4514.

[24] Taiwan Semiconductor. *P-Channel Power MOSFET*. Taiwan Semiconductor. URL: https://www.taiwansemi.

com/assets/uploads/datasheet/TSM680P06CH_A2204.pdf.

[25] Ling Zheng et al. "Lane-Level Road Network Generation Techniques for Lane-Level Maps of Autonomous Vehicles: A Survey". en. In: *Sustainability* 11.16 (Aug. 2019), p. 4511. ISSN: 2071-1050. DOI: 10.3390/su11164511. URL: https://www.mdpi.com/2071-1050/11/16/4511.

XIII. Appendix

A. *Appendix A - Initial Gantt Chart*

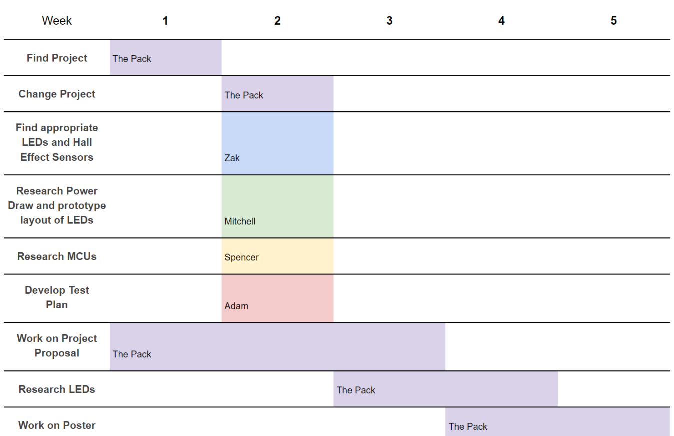

Fig. 14. Initial Gantt Chart Weeks 1 through 5

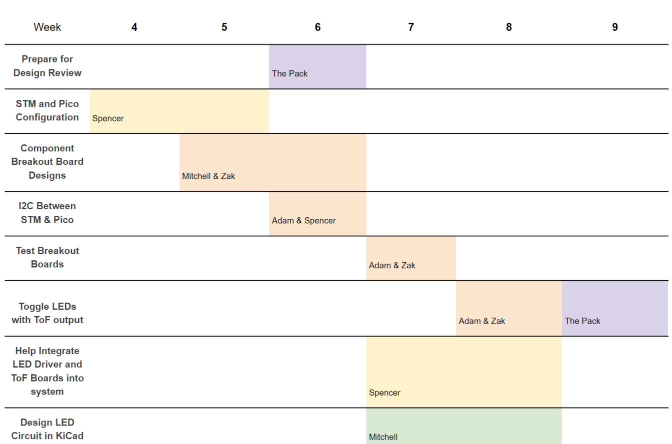

Fig. 15. Initial Gantt Chart Weeks 4 through 9

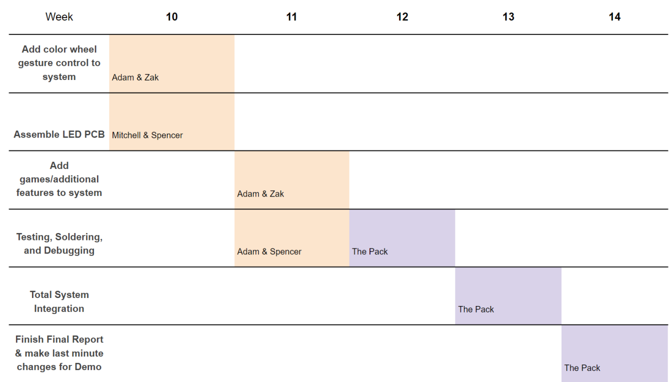

Fig. 16. Initial Gantt Chart Weeks 10 through 14

B. *Appendix B - Final Gantt Chart*

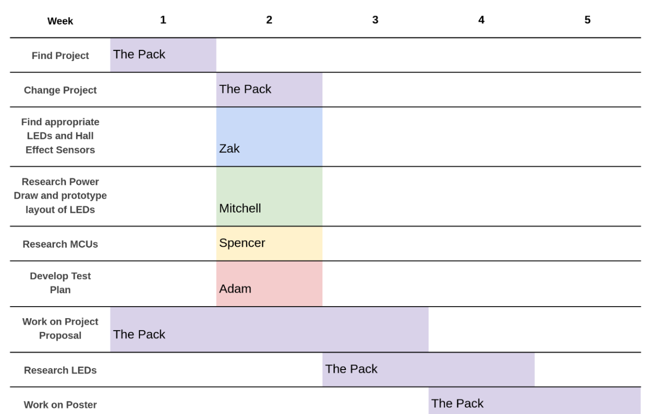

Fig. 17. Final Gantt Chart Weeks 1 through 5

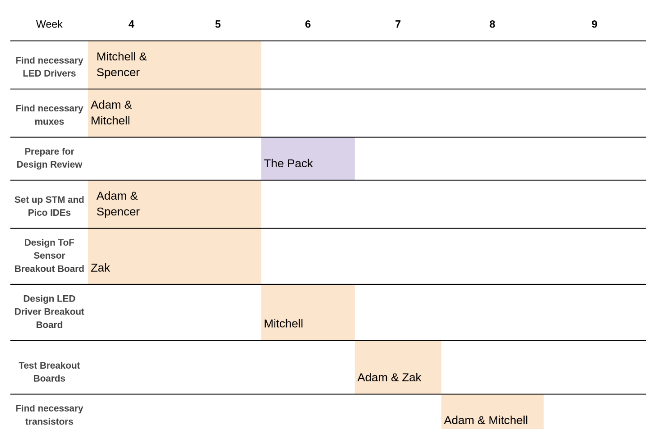

Fig. 18. Final Gantt Chart Weeks 4 through 9

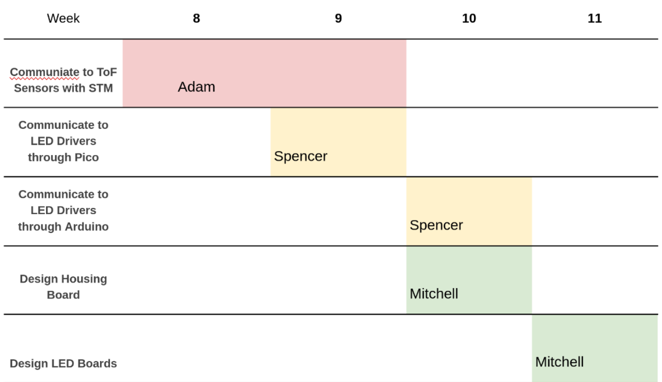

Fig. 19. Final Gantt Chart Weeks 8 through 11

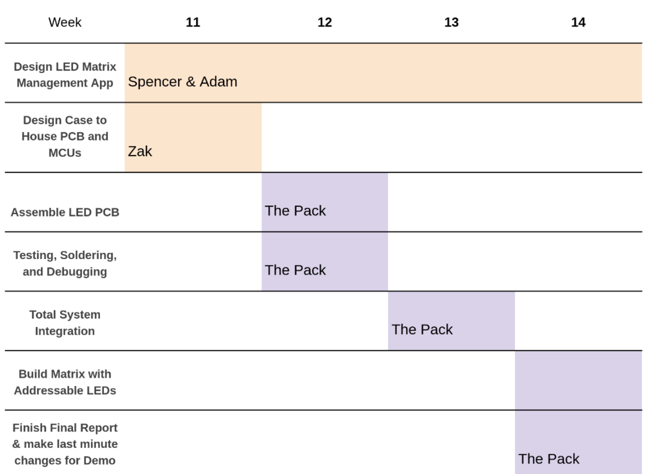

Fig. 20. Final Gantt Chart Weeks 11 through 14

C. Appendix C - Finances

Department Funds		Total	$499.43	
Part #	Item Description	Item Use	Used in Final Design?	Price
HV-5RGB25	Through Hole LEDs	LED Module	Yes	$20.19
DRV5053OAQLPGQ1	Hall Effect Sensors	ToF Module	No	$10.05
SC0915	Raspberry Pi	LED Module	No	$4.00
XZCMEDGCBD110W	Surface Mt LEDs	LED Module	No	$5.65
VL53L1X	Old ToF Sensors	ToF Module	No	$14.32
VL53L5CX	ToF Sensors	ToF Module	No	$34.60
LP5891ZXLR	wrong LED Drivers	LED Module	No	$20.10
TLC5940PWP	LED Drivers - got broken	LED Module	No	$15.04
CD74HC154M	wrong Muxies	LED Module	No	$1.80
TSM680P06CH X0G	PMOS	LED Module	Yes	$20.37
TPS566247	buck	Power Supplies	No	$0.49
SBC3-2R2-602	inductor	Power Supplies	No	$1.15
CF14JT30K0	resistors	Power Supplies	No	$0.50
CD74HCT154M	correct muxes	LED Module	Yes	$2.78
LD39150DT33-R	voltage regulator	Power Supplies	Yes	$1.53
PPTC201LFBN-RC	header pins	Power Supplies	Yes	$2.48
SFH11-PBPC-D20-ST-BK	header pins	Power Supplies	Yes	$3.92
PPTC061LFBN-RC	header pins	Power Supplies	Yes	$0.52
PPTC081LFBN-RC	header pins	Power Supplies	Yes	$1.32
PPTC101LFBN-RC	header pins	Power Supplies	Yes	$0.65
EBBA-03-C-SS-BU	terminal block	Power Supplies	Yes	$5.81
SSQ-120-03-T-S	header pins	Power Supplies	Yes	$6.40
NTE30153	Through hole LEDs	LED Module	Yes	$310.99
TLC5940PWP	More LED Drivers	LED Module	Yes	$14.76

Fig. 21. List of all Purchases Made with Department Funds

Personal Funds		Total	$240.97	
Part #	Item Description	Item Use	Used in Final Design?	Price
PCB	LED PCBs	LED Module	Yes	$76.11
PCB	CPU PCB	Power Supplies	Yes	$28.73
PCB	LED Breakouts	LED Module	No	$19.96
LM22678TJE-5.0/NOPBCT-ND	Buck convertor	Power Supplies	Yes	$6.84
490-12330-1-ND	caps	Power Supplies	Yes	$3.15
490-14384-1-ND	more caps	Power Supplies	Yes	$1.44
ED10561-ND	term block	Power Supplies	Yes	$5.05
399-18927-ND	inductor	Power Supplies	Yes	$0.96
112-VS-30BQ015HM3/9ATCT-ND	schkotty diode	Power Supplies	Yes	$1.16
4713-GMC21CG103F25NTCT-ND	more caps	Power Supplies	Yes	$1.18
RG32P470KBCT-ND	resistors	Power Supplies	Yes	$1.16
1189-2913-ND	more caps	Power Supplies	Yes	$0.66
VL53L5CX-SATEL	ToF breakout boards from ST	ToF Module	Yes	$49.43
NUCLEO-F401RE	new stm	ToF Module	Yes	$22.70
amazon	diffuse sheeting	Other	Yes	$11.38
PCB	tof breakout boards	ToF Module	No	$11.06

Fig. 22. List of all Purchases Made with Personal Funds

D. Appendix D - Educational Guide

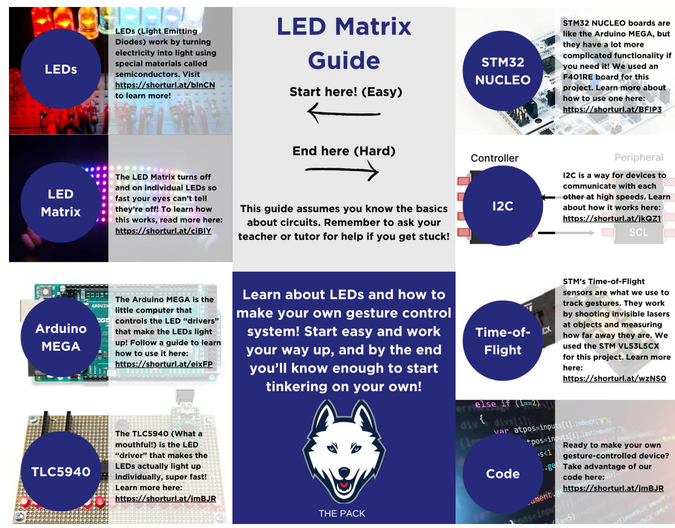

Fig. 23. Guide to the LED Matrix and Gesture Module

Links to images used for educational guide:

1) https://www.cleanpng.com/png-university-of-connecticut-connecticut-huskies-wome-6348867/download-png.html
2) https://www.st.com/bin/ecommerce/api/image.PF271455.en.feature-description-include-personalized-no-cpn-medium.jpg
3) https://soldered.com/productdata/2023/03/i2c-controller-peripheral-graphic.png
4) https://i.ebayimg.com/images/g/mtUAAOSwuYth81Q6/s-l1200.webp
5) https://www.pjrc.com/teensy/td_libs_Tlc5940_1.jpg
6) https://store.arduino.cc/cdn/shop/products/A000067_00.front_934x700.jpg?v=1637830343
7) https://cdn-learn.adafruit.com/assets/assets/000/003/888/medium800/leds_LEDrainbow.jpg?1396803411
8) https://www.smart-prototyping.com/image/cache/data/2_components/Light/101801%20WS2812%20flexible%20LED%20matrix%208x32/flexible-rgb-led-matrix-8x32-ws2812b-44263-750x750.jpg