Eye Tracking Headset Using Infrared Emitters and Detectors

Timothy Wells, Michael Fillinger, James Curtis, Patrick Gonzalez

Dept. Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* - The premise of this paper is to conceive and implement a system that can interpret the user's eye position and operate an application dependent on the user's line of sight. It will do this through an array of IR emitters and receivers inside the headset. The components chosen for the design, testing details of prototypes, and software details are also discussed and explained in this paper.**

***Index Terms* - Eye Tracking, Touch Free, Headset, IR, Raspberry Pi, Senior Design**

I. INTRODUCTION

Eye tracking technology has seen a huge jump in development over the past 20 years. This can largely be accredited to the development of neural networks and machine learning algorithms. However most eye tracking technologies that are available are very expensive and the average consumer is not able to purchase these technologies because of the exorbitant price.

Typical eye tracking technologies contain a camera, projector, and use machine learning algorithms in order to determine the user's gaze. These machine learning algorithms must engage in image processing which requires a powerful CPU or GPU. The cameras used in eye tracking must be able to take high resolution images of the subject. If the resolution of the images taken from these cameras are below a certain level of quality then the machine learning algorithm will not be able to function properly. Projectors of near-infrared light are used to create a pattern in the image that is captured by the camera. The requirements of the specs for the CPU or GPU used and the high resolution images that must be captured by the camera results in the eye tracking technologies being expensive.

We aim to develop an eye tracking technology that is affordable to the average consumer. This project seeks to eliminate the use for a camera and powerful CPU/GPU for eye tracking. The project uses infrared emitters and detectors in order to determine the gaze of the user. Depending on the gaze of the user, the detectors in the appropriate positions will react with a drop in the voltage. Whichever detector has this voltage drop will tell us the gaze of the user.

We will be using the analog to digital converter from the MSP430FR6989 in order to convert the signal that is read from the infrared detectors from an analog signal to a digital signal. A selector chip is used in order to choose which infrared detector we want to read from at any given time. We are using a Raspberry Pi 3 Model B+ in order to run our own machine learning algorithm along with the application we'll be using to demonstrate the eye tracking capabilities.

II. SAFETY REQUIREMENTS

The safety requirements of this project are of utmost importance. In this project we are exposing the human eye to infrared light and with enough intensity, infrared light can be damaging to the human eye. This damage can cause a higher likelihood of developing cataracts at later stages in life.

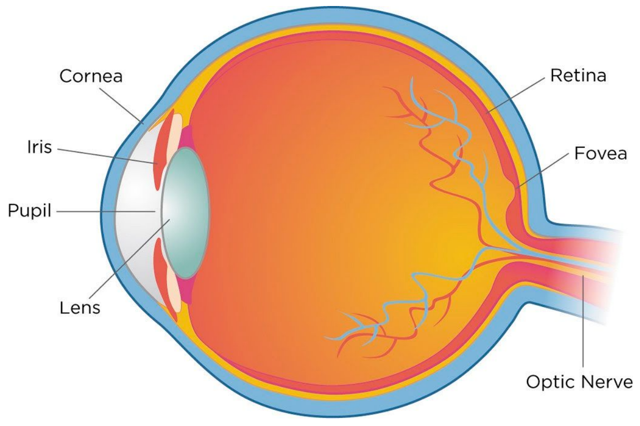


Fig. 1 Components of the Human Eye

Any of the components of the human eye shown in Fig. 1 can be damaged with sufficient exposure to infrared light radiation. Thus far, Light-emitting Diodes have not been found to have caused damage to the human eye. [1] However, LEDs have become more and more powerful over time and may pose a risk. We have utilized the equations that are defined in the IEC-62471 standard in order to ensure that these Infrared LEDs are not hazardous to us with prolonged exposure. [2]

$$E_{IR} = \sum_{\lambda=780}^{\lambda=3000} E_{\lambda} * \Delta\lambda \leq 18000 * t^{-0.75} (W * m^{-2}) \qquad (1)$$

Equation 1 is used to determine whether our Infrared LEDs will pose a risk to the cornea of our eyes. E_{λ} is spectral irradiance and $\Delta\lambda$ is the spectral bandwidth of our Infrared LED. E_{IR} must be less than the right side of equation 1 in order to be considered safe.

$$L_{IR} = \sum_{\lambda=780}^{\lambda=1400} L_{\lambda} * \Re\lambda * \Delta\lambda \leq \frac{6000}{\alpha} (W * m^{-2} * sr^{-1}) \qquad (2)$$

Equation 2 is used to determine whether our Infrared LEDs will pose a retinal thermal hazard. L_{λ} is spectral radiance, $\Re\lambda$ is the burn hazard weighting function, and α is the angular subtense on the retina. For when exposure time is greater than 10 seconds, α = 0.011 radians. L_{IR} must be less than or equal to the right side of the equation 2 in order to be considered safe.

We want to use the eye tracking goggles for a maximum of about 10 minutes or 600 seconds. So we will be using 600 seconds for the exposure time in Equations 1 and 2.

TABLE I
COMPARISON BETWEEN CORNEAL EXPOSURE LIMIT AND IRED CORNEAL HAZARD VALUE

E_{IR} (W / m^2)	Exposure Limit (W / m^2)	Safety Factor ($\frac{\text{Exposure Limit}}{E_{IR}}$)
21.65	148.48	6.86

The values in Table I were calculated using equation 1. The E_{IR} value represents the value behind only one Infrared LED. We are using 5 Infrared LEDs on each eye so we must multiply E_{IR} by 5 in order to get the true value that is affecting each eye. The value affecting each eye is 108.25 W/m^2 which is still solidly below the exposure limit.

TABLE II
RETINAL THERMAL HAZARD COMPARISON BETWEEN BURN HAZARD WEIGHTED LIMIT AND IRED BURN HAZARD VALUE

L_{IR} ($W/m^2/Sr$)	Burn Hazard Weighted Limit ($W/m^2/Sr$)	Safety Factor ($\frac{\text{Limit}}{L_{IR}}$)
65.17	918416.14	14092.05

The values in Table II were calculated using equation 2. Once again we must note that we are using 5 Infrared LEDs on each eye so the value calculated for the Infrared LED must be multiplied by 5. This gives us a value of 325.85 W/m^2 which is drastically lower than the Burn Hazard Weighted Limit.

These calculations show that the Infrared LEDs are safe for the exposure times that we wish to utilize them for.

III. DEVICE OPERATION

In this section the concept of how the device's system operation will be broken down for one's understanding. The basic principle of the system operation is that the system has an array of sensors that are placed around the operator's eye, for the system to collect, process, and translate the eye state data to an action towards the application of this project. The project's system is broken down to three main components that make up the operation, they are Goggles, the converter/PCB, and the processing unit.

Below is the block diagram that gives the reader a visual perspective of how the eye tracking headset works.

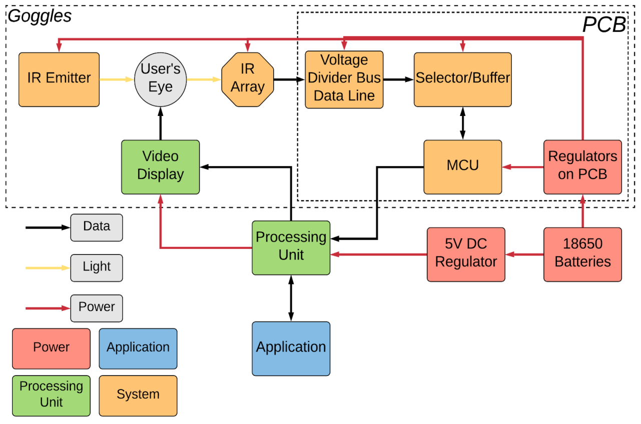


Fig. 2 Block diagram of how the operations of the system will work

A. Goggles

The first main section of this project is the apparatus used for collecting the data while also being a part of the application for the operator of the system. This section is made from a pair of goggles that contains two subcomponents: the sensor array and display screen. The idea of how these two components work together in the goggles is that the screen is a stemulie to the users eye and the sensor array captures the changes or states of the users eyes in a constant loop

B. Converter/ PCB

The second subsection of the project is the intermediate component for processing the data from the real raw data from the sensors to the processing unit. The principle behind the need for this was beside converting analog to digital data value for the sensor array, but also being the first processing of the data for the system, freeing up the main processor unit for the main algorithm and application to run unhindered. This section of the system is composed of three main components and a few power regulators to power those components. The first section is the voltage divider data bus line, which is the interface for the sensor array to the converter board for it to generate the analog data for the system. The second component is the selector unit, which is a buffer for the system due to the number of sensors from the array vs the MCU of the board. This is because most MCUs can only process one set of analog data at a time and you can't let the analog data lines directly cross since they are voltage levels they would mix and corrupt each other. The third component is the MCU which fulfills the multi part rolls of the converter of analog to digital conversion process, a sub communication point of the system between the sensors, selector unit and the main processing unit.

C. Outside Components

These are the rest of the system we don't directly make but use to run our system like the main processing unit. These components consist of the components, processing unit, power source and regulator for it. The processing unit holds the system main algorithm and application program. This requires the processing unit to be a multi core system to be able to run both programs simultaneously.

IV. DEVICE COMPONENTS

In this section the individual components of this touch free eye tracking device will be discussed. The component that was used will be spoken of and the reason why it was chosen will be discussed. There will be some components that were almost picked to be used for the device but the reasons they were not will be made.

A. Sensor Array

The first component to be used is the sensor. The signal starts by the IR emitter emitting light onto the eye and then the IR sensors receiving it. The important thing the sensor needed to be able to do was be able to detect light at an efficient enough range. The first sensor that was going to be used was Radioshack's Infrared LED Emitter and Detector. This one ended up not working because it could not receive the signal correctly for the project. It could not keep the receiver and emitter at an angle of 180 degrees.

Therefore, this sensor did not work for the touch free eye tracking headset. The sensor that was used for the project was the QRE1113. This sensor was able to receive and emit the infrared light and keep the components at an 180 degree angle always, thus working for the project.

Below is a picture of the sensor that was selected for this senior design project of touch free eye tracking headset.

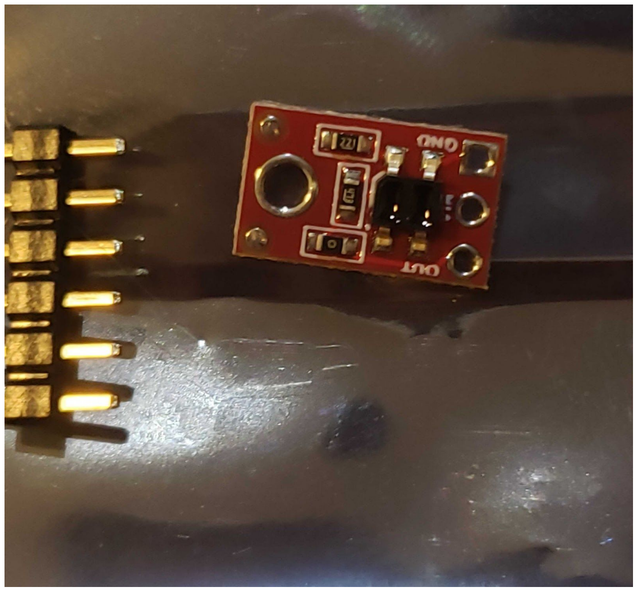


Fig. 3 Picture of the sensor being used for the touch free eye tracking headset

B. Selector Chip

The selector chip is supposed to select the Sensor from the array and gather the data it is detecting. Once it gathers the data from one sensor it will move on to the next sensor. Since there are ten sensors it needs to gather data, switch to another sensor, gather data and keep repeating quickly. For this reason we chose to use the CD74HC4067M. The CD74HC4067M has a delay of 51 nanoseconds to 49 nanoseconds. For this senior design project that is quick enough for the purpose that it is needed for.

There were some other considerations for this component such as CD4067B, MPC506AU/1K, or CD74HCT4067M but the chip that was chosen was the best option because it had the fastest time for selecting between the sensors in the array and price, which is why it was picked over the others.

C. Processing Unit

The primary processing unit used in our design is the Raspberry Pi 3 Model B+. The primary function of this device is to receive the data created on the microcontroller device, and use that data to determine the relative position of the eyes.

To accomplish this, the processing unit first reads the data from the serial line as it is transmitted. It then organizes the data into a format in order to be further processed. Once the data is properly formatted, it is then processed through the algorithm for determining the eye position.

From the algorithm, an output is produced that will represent the relative position of the eye. This output is then used by the application software to display a graphical representation of the eye position. This can be easily modified to accomplish various other tasks, however a simple graphical display was chosen to easily demonstrate and test the system.

The decision to use the Raspberry Pi 3 Model B+ was based primarily on the small size of the unit, and the low price per unit. This results in the least cumbersome and least expensive choice for the system.

D. MCU

The MSP430FR6989 was the microcontroller unit selected to perform the majority of the logical functions on the printed circuit board for the system. The primary function that the microcontroller performs is the analog-to-digital conversion of the signal received from the IR sensor array. The microcontroller also organizes the data and transmits the data to the processing unit via a serial communication line.

The secondary function of the microcontroller is to control the selector chip. This allows for the microcontroller to be able to select which lines are required to read, and in whatever order may be necessary. This allows the programmer to easily reconfigure the data lines that are read by the device. This means that the system can possibly use between one and sixteen data lines.

The microcontroller also allows us to choose the reference voltages that need to be measured against for the analog to digital conversion. A positive reference voltage of 5 volts was selected to measure the signal coming from the IR sensors. Unfortunately, due to the inability to procure a new PCB because of the recent COVID-19 pandemic, a low voltage reference cannot be used because the pin 1.1 is reserved on the MSP430FR6989 development board.

This device was selected for multiple reasons. It is able to independently accomplish many of the tasks required for the system on the circuit board level. It also provides a relatively high resolution for analog to digital conversion, which is useful because the changes in the IR sensor readings can be small between two close eye positions. Another reason that this device was selected was due to the large amount of resources available for free through Texas Instruments, which were very useful during the development process.

E. Headset

The headset is a key piece in the product because it is the one the user will be most aware of immediately. The type of headset determines the comfortability of the product for the user. Certain headsets can cost as much as the MCU or Microcontroller. The cheaper the product the more likely people will use it, for people never want to spend more money than they have to. The product also needs to be adjustable. Adjustable in this sense means it needs to be able to fit all average heads. If the headset doesn’t fit on the head then it won’t work. The headset also needs to be able to be changed so the senior design team can make it compatible with the requirements.

The Utopia 360 was the best option because it was comfortable, cheap, adjustable, and easily changeable. Other options were the google cardboard or a self built one. The google cardboard’s material was as dependable as the Utopia 360 and a self built headset would have taken too much time. Therefore, the Utopia 360 meets all the requirements for the design and is the best option for the device. Below is a picture of the Utopia 360.

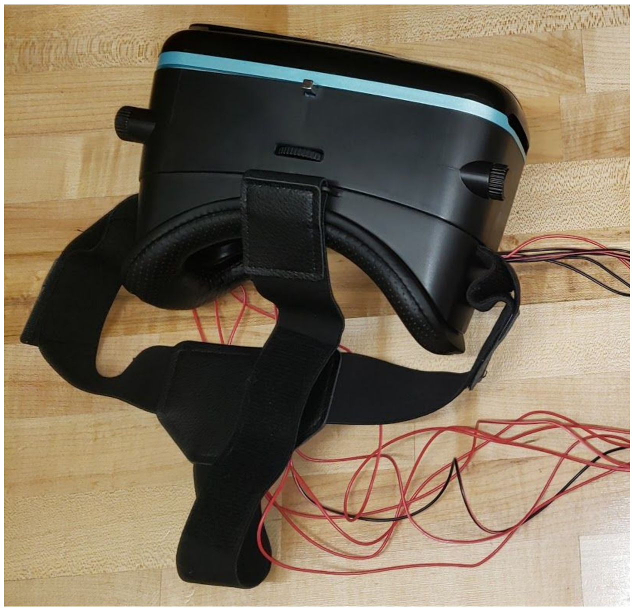


Fig. 4 Goggle set for the touch free eye tracking headset

F. Screen

The screen was originally going to be a phone because it would fit into the Utopia 360 and since all team members owned a phone it would not have been an expense to record on the budget. However, due to complications with connecting the phone to the MCU, a new screen was searched for. The new screen was the LCD Display screen that came with the Raspberry Pi Piper kit. The advantage of this screen was that it was completely compatible with the Raspberry Pi. To connect the screen to the Raspberry Pi the only thing needed is to connect the two with a HDMI cable.

The problem with this new LCD Display is that it did not fit into the Utopia 360. To solve this problem the team took off the front of the Utopia 360 and hot glued a wooden frame to it. The attached wooden frame allowed the new screen to be put on and taken off of the headset, which solved the problem. Therefore, the screen used for the project is shown below.

Fig. 5 Front side of the LCD Display Screen used for the touch free eye tracking headset

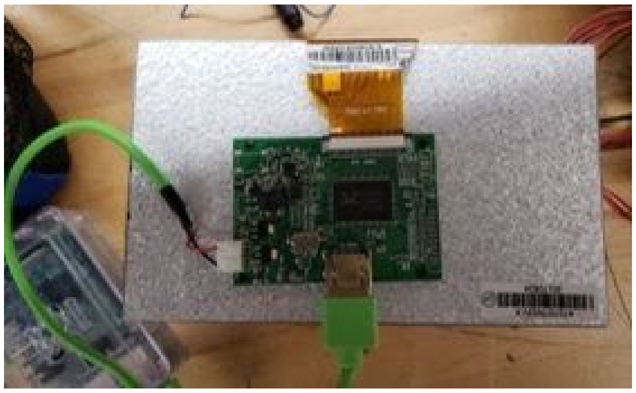


Fig. 6 Back side of the LCD Display used for the touch free eye tracking headset

The front of the screen is shown in the first image and the back with the connection chip and HDMI input is shown in the second image.

V. TESTING

In this section we will go over the results from testing components for the system. The individual tests conducted on the system were that of the sensors, the communication between MSP430 and the Raspberry Pi, the converter system of the MSP430 for the sensors, the selector chip unit in between the MSP430 and sensor array. Due to a global outbreak, components were limited to break out components.

A. Sensor Test

For this section of the system's test we first set up a test rig with the use of a converter chip MCP3008, which is a 10 bit converter, the setup was at first a test if the concept even was variable in the sense of sensitivity. If it was a range of values to an all or nothing like sensor like a push button. The system showed it was variable but with close margins thresholds between colors, distance, and more importantly the resistance values used in for the sensors. The resistance was found to cause a somewhat expenecal curve to what the sensitivity vs the resistance of the circuit. The setup of how the sensor reads for this and the rest of the test was that it read low for low state, or no light at the receiver and it lowers the value depending on the amount of IR light shining onto it. Example is that if the lead resistance is low like ten thousand ohms the range in values is low while but the overall range of the system can still use most of the voltage of the sensors, meaning the highest value the sensor can reach is higher. However for when the resistance got to be above hundred thousand ohms the range the sensor could get went up but the highest point the sensor can reach. When we got to one mega ohm the high value was a little more than half the full range but the sensor could read about seventy to eighty percent of that range.

B. Communication for the Pi and MSP430

The test for the Infrared LED sensors allowed us to also test the communication between the Raspberry Pi and the MSP430. The microcontroller and the Raspberry Pi communicate along a serial line. The microcontroller transmits data at 9600 baud. The communication was tested by first sending test characters to the Raspberry Pi and ensuring that the data could be read without any loss or corruption of the data. Once it was clear that the data could be read to the Raspberry pi properly, the code was then configured to send the digitally converted signal in a more organized form such that it can be read by an application. The data is then stored into a comma separated variable file. This was to allow for the data to be easily used for training the machine learning model.

C. MSP430 Converter Test

When we moved over to the proper converter the main difference was the difference in the resolution of the two converters. Where the first test setup system, MCP3008, was a ten bit converter while for the MSP430 it is a twelve bit converter. The difference in converters change the sensor sensitive range due to the increase in resolution. However due to the setup of the launch board of the MSP430 the converter negative or ground reference point for the converter is tied to something else on the board so the values of the sensors are slightly skewed a little on the high side. On top of that there was added noise to the system sensitive due to the negative reference

D. Goggles Sensor Array Test

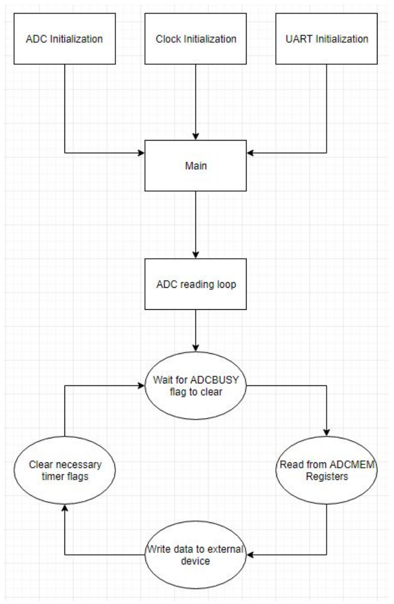
Moving on to the main testing phase for the system was the first test of the whole setup together. The setup is the sensor feeding into the selector unit, then the selector into the MSP430, then MSP430 to the Raspberry Pi. During this phase we went through a few setups of the sensor arrays for the emitters vs the receivers. Mostly the emitters were moved around due to the receivers being in fixed point that we were interested in measuring the reflectance of the users eye. The placements were first one emitter per eye array, then two per array then four all on one eye. During all points of these tests we came across errors like some sensors not reading at all or sensors reading when they shouldn't. One error was strange that happened multiple times when we got any good progress was the sensors some of the time, inverted the principle of operations when not being tough, we could only think that they were shorting out at times. These were happening with the old IR diodes which led us to move to the QRE-1113 IR sensors. These new sensors are more arrcote by requiring to be much closer, and more emitters. This caused more changes to the placement of the sensor in the goggle set due to eye safety and requirements of the new sensors. The only problem from the new sensor was that they were sensitive to the voltage spikes from the system due to mechanical problems of the setup, so ever so often the sensor might spike for one time interval up or down.

VI. SOFTWARE DETAILS

In this section we will be delving into the details of the software used for our project. We will be using a machine learning algorithm in order to help determine the gaze of the user. Also we will be detailing the application that we will be utilizing to showcase the ability of our device to track the user's gaze.

A. System Firmware

The System Firmware is contained within the MSP430FR6989. We enabled communication between the Raspberry Pi 3 Model B+ and the MSP430FR6989 using the UART communication protocol. We are reading in 10 different channels of analog signals as input from the QTR-1A reflective sensor shown in Figure X. These analog signals are converted to digital signals using the ADC chip found on the MSP430. The values of the digital output created by the MSP430 ADC chip are printed onto the terminal screen as shown in Fig X. The UART code was developed on the Integrated Development Environment Code Composer Studio.



ADC Initialization
Clock Initialization
UART Initialization
Main
ADC reading loop
Wait for ADCBUSY flag to clear
Read from ADCMEM Registers
Write data to external device
Clear necessary timer flags

Fig. 7 Microcontroller code diagram

B. *Machine Learning Algorithm*

The method for which the user's eye position is determined is through a machine learning model. The model has an input dimension size of 10 for each of the IR devices in the array. The model was built using the Keras API, as it is very easy to adjust and train the model quickly for the purposes of testing and refining the model to have a higher accuracy. The backend being used for the project is Tensorflow, and this combination allows us to easily convert the data transmitted by the microcontroller into a set of training data to be used by the model. The model is then trained using a set of several thousand data points to create a set of weights to predict the user's eye position.

The initial model used is not highly complex due to the relatively simple input data being created from the MSP430. The first layer of the model after the input contains 20 nodes and uses ReLU. The initial model uses a class of 4 different outputs, which correlate to the user looking up, down, left, and right.

The reason machine learning is being used to determine eye position is that the combination of noise and the difference in eyes between individuals can make it difficult for a human to detect a pattern in the data, and that pattern may even change from one person to another. A machine learning model may be able to detect a pattern in the data more efficiently.

C. *Application*

Our application is a graphic user interface built using TKinter with Python.

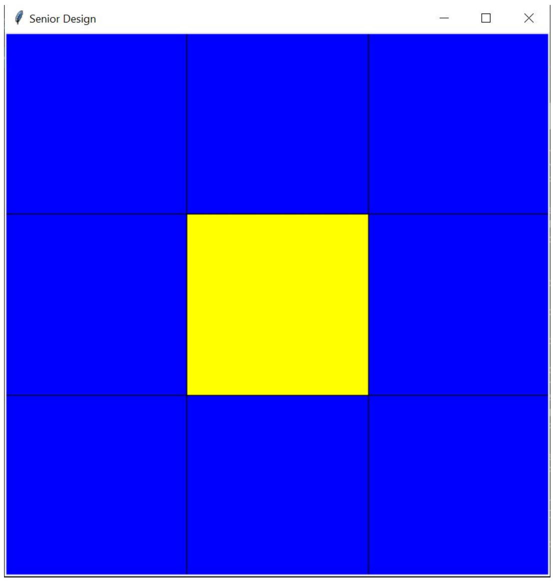


Fig. 8 Application that indicates the user's gaze

Our application contains 9 squares in a 3 x 3 configuration. There are always 8 blue squares and one yellow square at any given time when the application is running. The yellow square indicates where the user is looking.

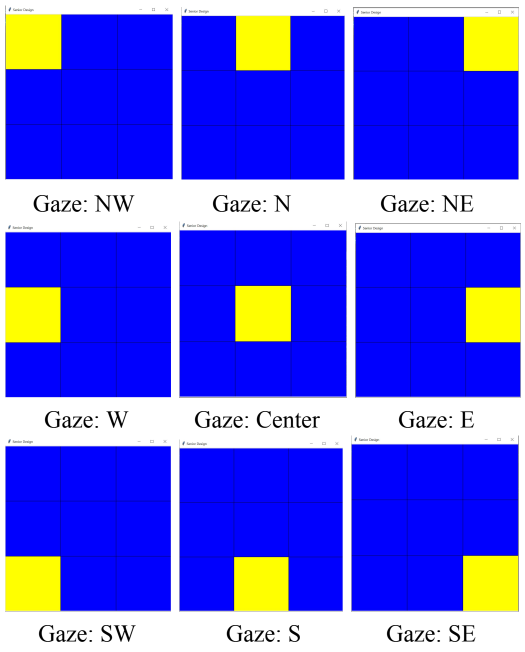


Fig. 9 All of the possible displays of the graphic user interface application

If the user's gaze is centralized in their perspective then the yellow square will be in the center of the 9 squares. If the user's gaze is to their right then the square in the second row and the third column will become yellow. If the user's gaze is to their left then the square in the second row and first column will become yellow. This application tracks the cardinal directions of the user's gaze from the user's perspective and the directions in between Northwest, Northeast, Southwest, and Southeast.

D. Serial Communication

Using serial communication between the MSP430 and Raspberry Pi, we were able to take the digital values from the MSP430 and store them in a text file. The serial communication was developed using Python.

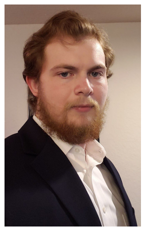
CONCLUSION

Our experience in developing this project has been a huge learning experience for all of the members of our group. This Senior design team has learned how to conduct meetings in an effective manner along with navigating through everyone's busy schedule in order to set time aside to work on a project within the team. Learning all the intricacies of writing a technical document on a project has also been a valuable learning experience.

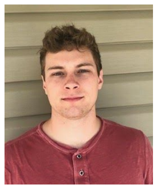
We have also learned

BIOGRAPHY

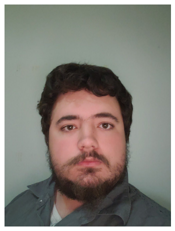
Patrick Gonzalez will graduate in May of 2020 and receive his Bachelor's Degree in Computer Engineering from the University of Central Florida. He is currently pursuing work in the field of Computer Engineering.



Michael David Fillinger is currently a senior at the University of Central Florida. He is pursuing a Bachelor's Degree in Computer Engineering. Michael will be graduating in May 2020



James Curtis will be graduating in May 2020 and receive his bachelor's degree in electrical engineering at the University of Central Florida. After graduation he will be starting his professional career at Lockheed Martin's Space Department in the state of Colorado.



Timothy Wells will graduate in May of 2020 with a Bachelor's Degree in Electrical Engineering with a minor in robotics at the University of Central Florida.

REFERENCES

[1] Allen M. Earman, "Eye Safety for Proximity Sensing Using Infrared Light-emitting Diodes" Renesas Electronics Corporation, AN1737 Rev 3.00 April 28th 2016

[2] IEC-62471 Photobiological safety of lamps and lamp systems. s.I.: IEC, 2006. Clause 4.X CIE/IEC-62471:2006