Holographic LED Display

ECE 445 Design Document

Rishab Sriramoju (rishabs3), Sriram Rameshbabu (sriramr2), Amar Patel (apate377)
Group 23
TA: Yichi Zhang
9/25/20

1 Introduction

1.1 Objective

People have always discussed what direction technology is going to steer towards in the future, whether that's transparent displays or holographic images. Movies always fantasize about the future, including works such as Back to The Future and Interstellar. There have been products on the market to emulate such technologies, like semi transparent televisions or fan “holographic” led displays. Our group would like to take this concept to the next level. 2-D hologram replicants are in existence already. They are mostly sold as a novelty, with little practical use.

Our objective is to create a display of LEDs and sensors that are rotating in some fashion such that it can appear holographic and to be able to display certain features. We want our hologram to be reasonably priced as well, because we would like it to be more of a home decoration piece, rather than a museum artifact. Some features that we would like to include would be the current time, a preset visual animation, or a text-to-display message from a phone. Essentially, our goal is to be able to program letters, numbers, or animations and view it on our 3-D display.

1.2 Background

Currently, LED lights are extremely useful in homes and buildings and give us more diversity in terms of aesthetic purposes. There are many creative ways they have been implemented: audio visualizers, LED strips, and 2-D holograms. The current form of home 2-D holographics are good in nature, but they can easily be topped. 3-D Holograms exist on the market, but they are often very tech focused and largely expensive. We can say for certain that it is not worth having a hologram display in your home if it is way out of a working class’ price range.

For holographic displays, there is current research occurring around the world to make this possible [8]. For the scope of our project, we are trying to mimic the holographic effect which has advantages such as increasing realism, as well as to keep the price range reasonable. Using the LEDs in a synced fashion for a 3-D display makes it possible for an interesting effect to come into play. The refresh rate of these LED lights is also really important in terms of generating our display; by making our refresh rate faster, we are able to obtain a clearer image because there is less downtime between a flicker of light.

1.3 High-Level Requirements

- Sync web app to local host using microcontroller with DHCP protocols and be able to toggle between settings of display functions such as preset animations, time, or specified texts
- Effectively rotate propellor between 400-500 rpm with the center of mass in the middle of the chassis to ensure proper alignment
- Calculate rpm from the hall sensor to ensure consistency of the hologram and avoid the presence of inaccurate information displaying on LEDs, which should update *under* .00075 seconds.

2 Design

2.1 Block Diagram

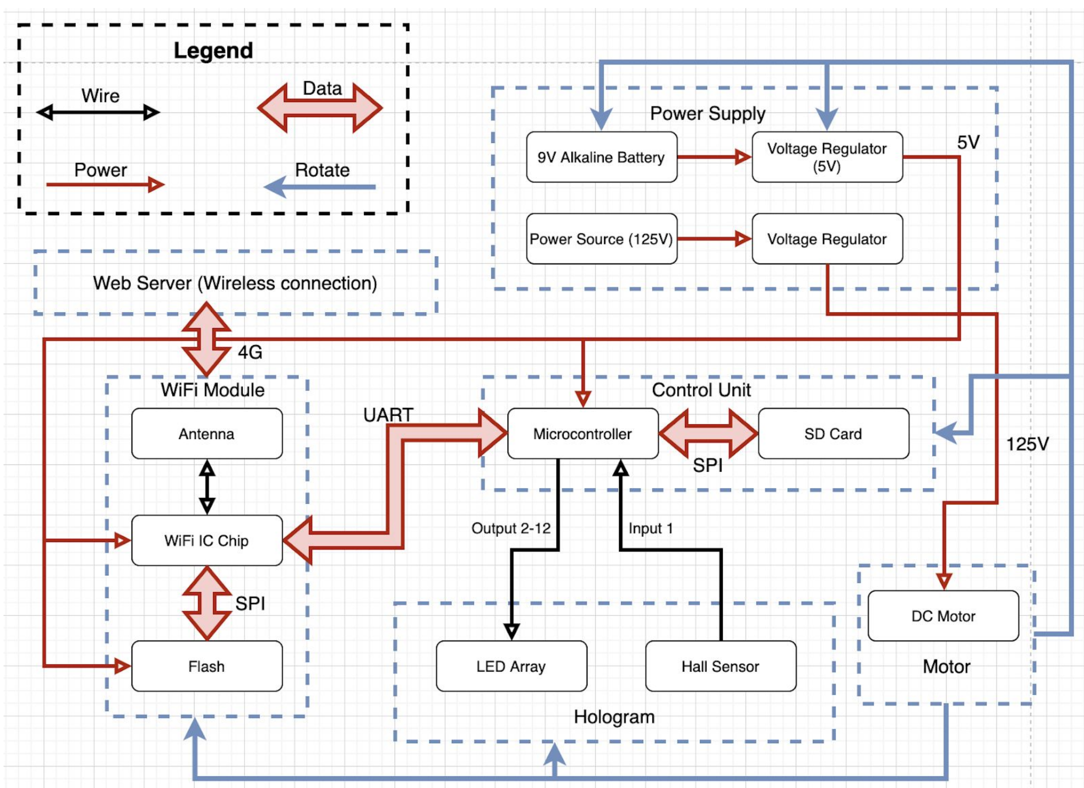

Figure 2.1. There are several types of connections such as wire, power, and data. A motor is sourced from a power source of preferably 125V that will help power the DC motor. A 9 Volt Alkaline battery will serve as the main power source for the rest of the system. Since the

microcontroller, Wifi module, and LEDs do not take up a lot of power, this will be sufficient enough. The UART protocol will be used for providing the connection between the microcontroller and the WiFi IC chip, and the SPI protocol will be used for providing the linkage between the WiFi chip and flash. The microcontroller will serve as the crux of our control unit and will dictate what is shown on our LED array and to keep the hall sensor in coordination with our system. There will also be a wireless connection to the web server via the antenna. No physical connection is necessary and is thus not shown on the block diagram.

2.2 Physical Design

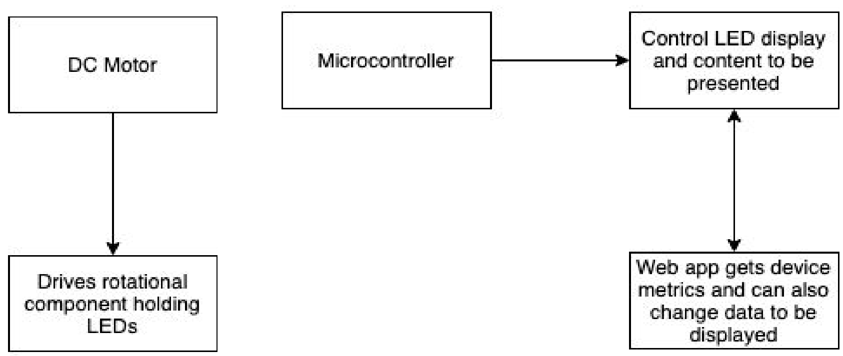

Figure 2.2. This figure shows the design features of the main modules to reach the objectives for this project. The main gist of the project is that there is a control unit that has presets for each setting already built in it. The Wifi module is connected to the web server in which you can toggle between settings. The microcontroller hosts the code in the SD card or alternative storage. The outputs of the control unit are connected to the LED array, and one input for the hall sensor. The hall sensor works in conjunction with a magnet in order to calculate the rotations per minute of the motor. The motor mechanism will thus be completely disconnected from the system.

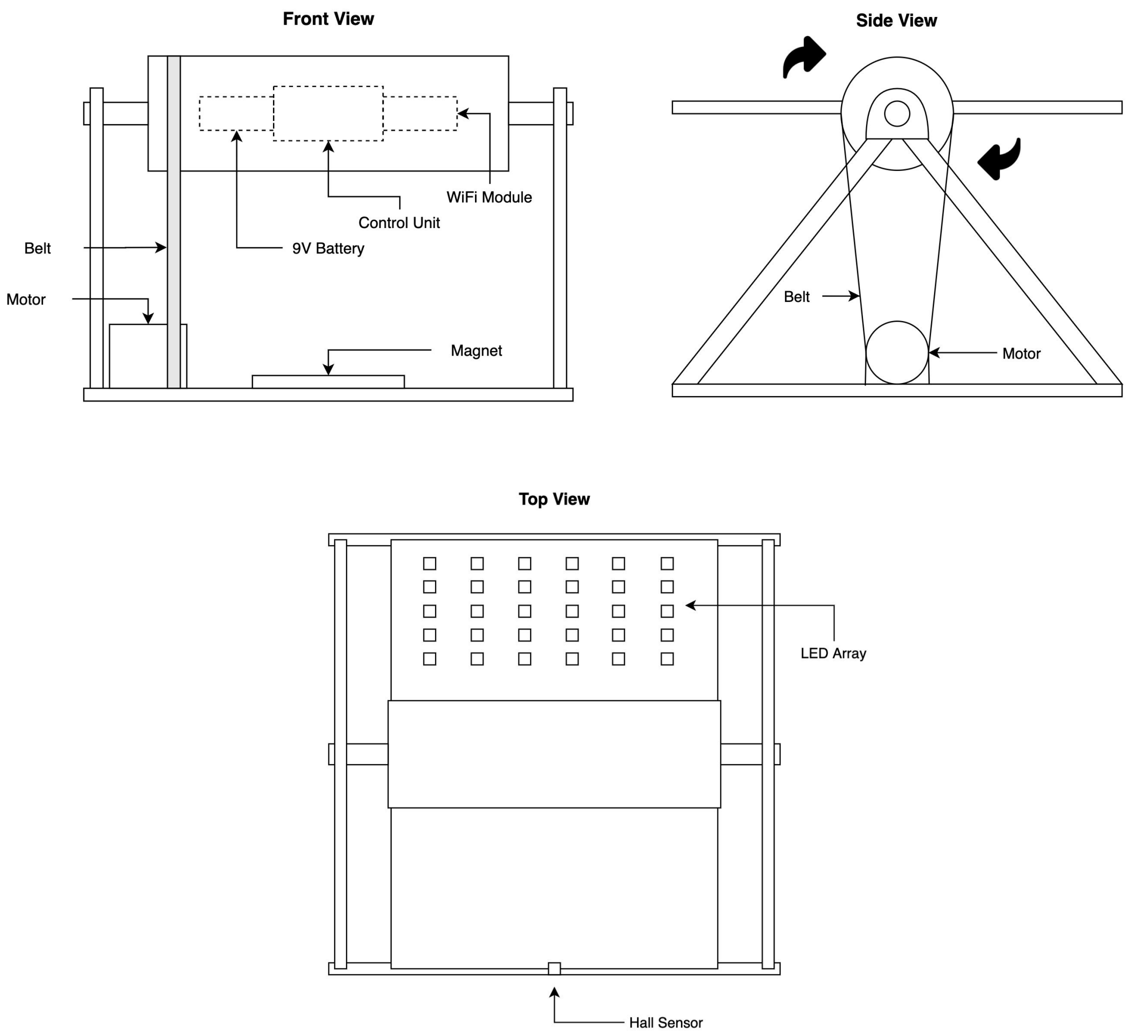

Figure 2.3. This figure shows what the overall product will look like in terms of a final solution. The motor will be connected to the 3-D printed base and use a belt system to rotate the propeller. The propeller will be attached to the main unit via ball bearings, for the purpose of rotating with as little friction as possible. All the electronic components besides the motor will be held inside the propellor. None of the electronic components are visible besides the LED array and the hall sensor. The LED array is displayed on one side of the propellor, while the hall sensor is on the other side, keeping track of the rotations of the motor. The two system design makes it very easy for the propeller to move independently from the entire system.

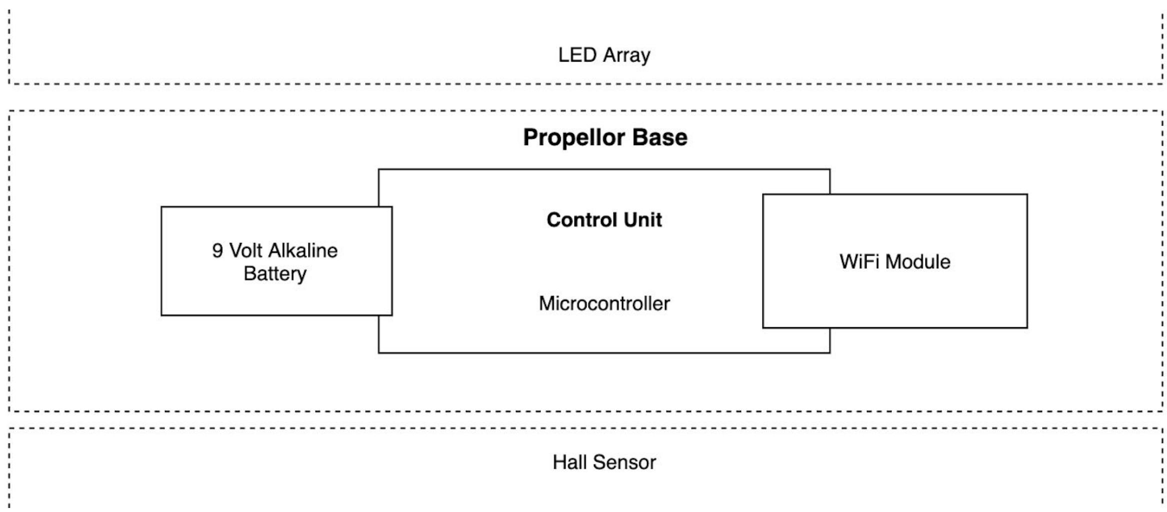

Figure 2.4. This figure represents what is stored inside the propeller base, something that is not shown clearly in the previous figure. This is housed inside the rotating middle piece, with two branches coming from the control unit going up to the LED array, and down towards the hall sensor. There is a battery connected to the PCB and the WiFi module connected to the other side. The 3-d printed base will have to be designed in such a way to allow the center of gravity to be in the middle of the base. If it is not the spinning mechanism will not rotate correctly, thus leaving the project in jeopardy.

2.3 Software Design

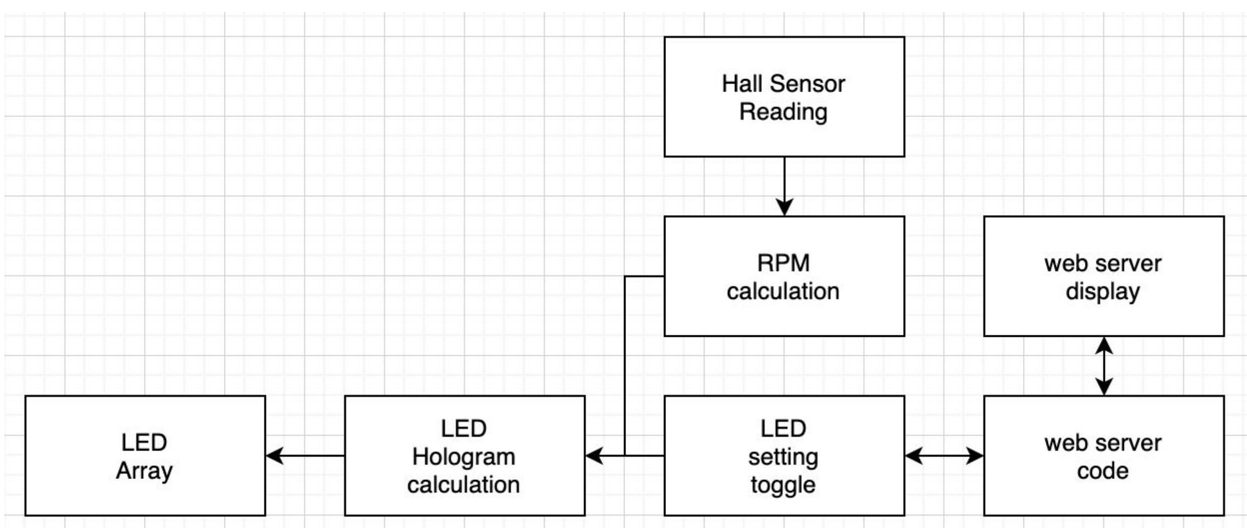

Figure 2.5. This is our software flowchart of how the code in the microcontroller is going to be run. The input from the hall sensor is going to be tracked via a function in order to calculate the rpm. A web server is going to be in contact with the web server code that is also housed in the microcontroller. When a button is pressed on the web server, the code will instantly know and send the information to the LED setting toggle, which will then be sent to the hologram LED calculation part of the code. This is the stage where both the data from the rpm and the LED setting toggle is taken into account in order to calculate when the LEDs should be on at any given moment in time. The result is displayed on our LED array.

2.4 Requirements and Verifications

2.3.1 Power Supply

The power will consist of a 9 Volt alkaline battery for the electronic components in the propellor. Most of the components require less that 9 volts to operate at their given capacity so this requires us to use a voltage regulator in order to reduce the voltage to 5 volts. The microcontroller, the WiFi module, and the LED array are all able to operate at 5 volts.

Requirement	Verification
1. Make sure the voltage output is 5 Volts ±5%	1. Use a multimeter and measures voltages across microcontroller, WiFi module and LED array a. Threshold voltage should be (4.75, 5.25) V.

2.3.2 WiFi Module

The WiFi module will be the main connecting point between our control unit and the web server to toggle between settings. The IC chip that we will be using is the ESP8266 [6], which will be the transceiver that will send the data to the microcontroller.

Antenna

The antenna in our WiFi module is used to send/receive signals from the WiFi chip itself and the web app.

Requirement	Verification
1. The antenna should be able to send and receive signals between 0-15 feet (0 meters - 4.572 meters).	1. Stand at intervals of 5 feet away from the set up (5, 10, 15) ft. and ensure a strong signal connection from each of these areas. a. This connection will be deemed as a strong signal by executing code to switch on an LED from the set intervals of 5 feet, 10 feet, and 15 feet. b. We can quantitatively assess a strong signal by ensuring there is no loss of packets during data transmission and all packets are transmitted within 2 seconds.

WiFi IC Chip

The WiFi IC Chip is used to be the logic for our WiFi module. It acts as the middleware between communicating what our objective is from the web server to the microcontroller.

Requirement	Verification
1. Able to connect to web server 2. Convey proper information from the code to a smartphone	1. Verify DHCP protocol IP address from any computer and log in to that address in a web browser. a. The content of the code should display there. 2. When a button is pressed, a temporary LED should turn on and off in order to determine functionality of the chip

Flash

The flash is connected to the WiFi IC to store its program memory.

Requirement	Verification

1. The flash is able to operate at least 80 MHz for full capacity.
2. The flash must have at least 1 MB of program storage.
	1. The flash should be able to run a simple program to switch on an LED at 80 MHz.
	    a. Make the flash at 80 MHz and run the simple program and confirm that it works.
	2. Have a simple program to turn on LED lights that is at least 1 MB of storage.
	    a. Read LED data back and forth using the UART terminal and make sure that there is consistency between the data.

2.3.3 Control Unit (Microcontroller)

The microcontroller that we are using is called the ATmega328 chip, which is the same chip used in a lot of low level computational devices and will also suffice for our project. It is commonly used in Arduino [5], which is perfect for the scope of our project. The chip operates at 5 Volts, and has a CPU speed of 16 MHz, which will also be sufficient for our project. The calculations done in our tolerance analysis shows how fast the microcontroller has to send outputs to the LEDs. The rest is up to the speed of our code. For the software component for our microcontroller, this is explained in 2.3.4.

Requirement	**Verification**

1. Memory is able to store code without reuploading the code to the microcontroller
2. Microcontroller is able to update the values of the LEDs under 0.00075 seconds.
	1. Turn on and off the microcontroller and see if everything is still functioning properly.
	    a. Execute code that turns on an LED light.
	    b. Unplug the power and then plug it back into the microcontroller and see if it is possible that there is proper functioning of executing the original code of turning on and off an LED light.
	2. Create a program which is able to run executed code of switching LEDs on and off for within 0.00075 seconds.
	    a. This can be done using a timer library in terms of code and to create code that is able to refresh the LEDs on and off and make sure that the time is satisfied.

2.3.4 Software

The web server will be connected to the microcontroller via the WiFi module. The code for the microcontroller will consist of several components. The first component will be the calculations for the hall sensor. The hall sensor will be used to calculate the time that has elapsed between the rotations of the propellor. Since our motor isn't attached to the microcontroller, this is the easiest way to simply calculate the output. The next component would be assigning the LEDs that need to turn on based off of the calculated rpm. The last component would be the web server. Using DHCP protocol, we are going to make HTML code to host a local domain so that the settings can be toggled through any device with internet connection.

Requirement	Verification

1. Code should be able to turn on and off all LEDs within our threshold of **.00075 seconds**.
2. Be able to gauge the rpm of the motor using the hall sensor on web server client
3. Toggling different settings should change display to desired outcome
	1. Unit test each input in series configuration to see if each LED maintains high or low as specified by voltage source
	    a. When voltage is high, each LED should light up at full brightness
	    b. When voltage is low
	2. Measure timestamps between repetitive passes by Hall sensor.
	    a. Measure time duration between each low input of the hall sensor.
	    b. Calculate rotational speed; this speed should match the rpm of the motor at any given time.
	3. Input a message into the web server client and check that the new message text is appropriately displayed on the LED display.
	    a. If previous message is "x", inputting "y" should instantaneously display "y" on LED display

2.3.5 Hologram

LED array

The LED array is used for the actual holographic display. The LEDs have to update fast enough for our display, which is more dependent on the microcontroller than the actual LEDs. Nonetheless, there should be a baseline requirement for the LEDs. The LEDs that we bought are standard diodes of a singular color.

Requirement	**Verification**
1. Refresh rate of LEDs are faster than .00075 seconds, which is fast enough to update with the code	1. Ensure that the refresh rate is below .00075 seconds a. Rotate the propellor at the rated 500 rpm and get the code to update an LED every .00075 seconds b. Granted that there is less than .00075 latency for the LED, the hologram should show 20 points to create a circle based off of our calculations in tolerance

Hall Sensor

A hall sensor is going to be used to calculate the rpm of the motor. Every time the hall sensor comes in contact with the neodymium magnet, the output voltage will be 0 Volts, and when it is not, it will be 5 Volts. We can track this data through the microcontroller. The hall sensor will be used to determine position

Requirement	**Verification**
1. Hall sensor should send 0 Volts when next to the magnet and 5 Volts elsewhere 2. The Hologram should not waver when displaying	1. Ensure output of the hall sensor is Active LOW a. No voltage present when placed next to magnet and +5V at multiple places 2. Test 45 degree lines with the LEDs and ensure that they do not waver every rotation of the propellor

2.3.6 Brushless DC Motor

The motor that we are using for this project is a brushless DC motor. Since we are not worried about the sudden stopping and starting of the motor, a brushless DC motor is the best motor specification for us. The motor is connected to the propellor via a belt system. The motor will be connected to a wall outlet rated at 120 volts. The voltage regulator is in place to ensure that the voltage will step down just enough to spin the motor between 400-500 rpm.

Requirement	**Verification**

1. The propellor is spinning between 400-500 rpm
2. Voltage spikes do not affect the motor speed for safety
3. Hardware body is stable
	1. Calculate gear ratio between spinning motor and propellor (attached to belt) to ensure 400-500 rpm of the propellor.
	    a. Use voltage regulator to adjust accordingly if rpm is too high or low
	    b. Voltage and RPM are directly proportional
	2. Create a simulated voltage spike at +12V, and measure brushless DC motor's speed.
	    a. Motor speed should remain unimpacted
	3. Run propeller at overdriven speed (1.5x ~ 600 - 750 RPM) and ensure the hardware body has minimal movement.
	    a. Body stability with overdriven speed implies stability with motor being run at regular speeds

2.5 Board Layout

Control Unit:

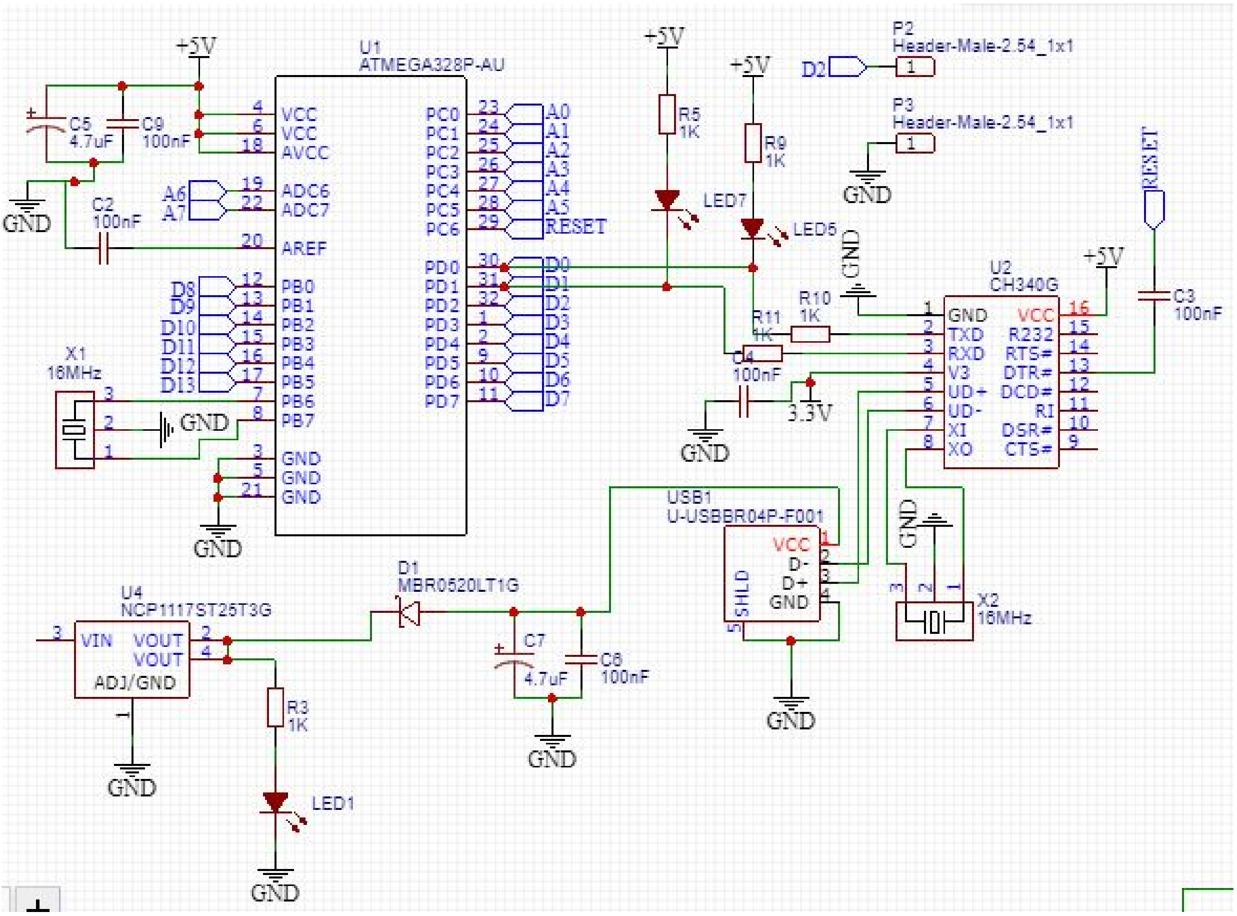

Figure 2.6. This figure represents a circuit schematic of the ATMEGA328P control unit.

Wifi Module:

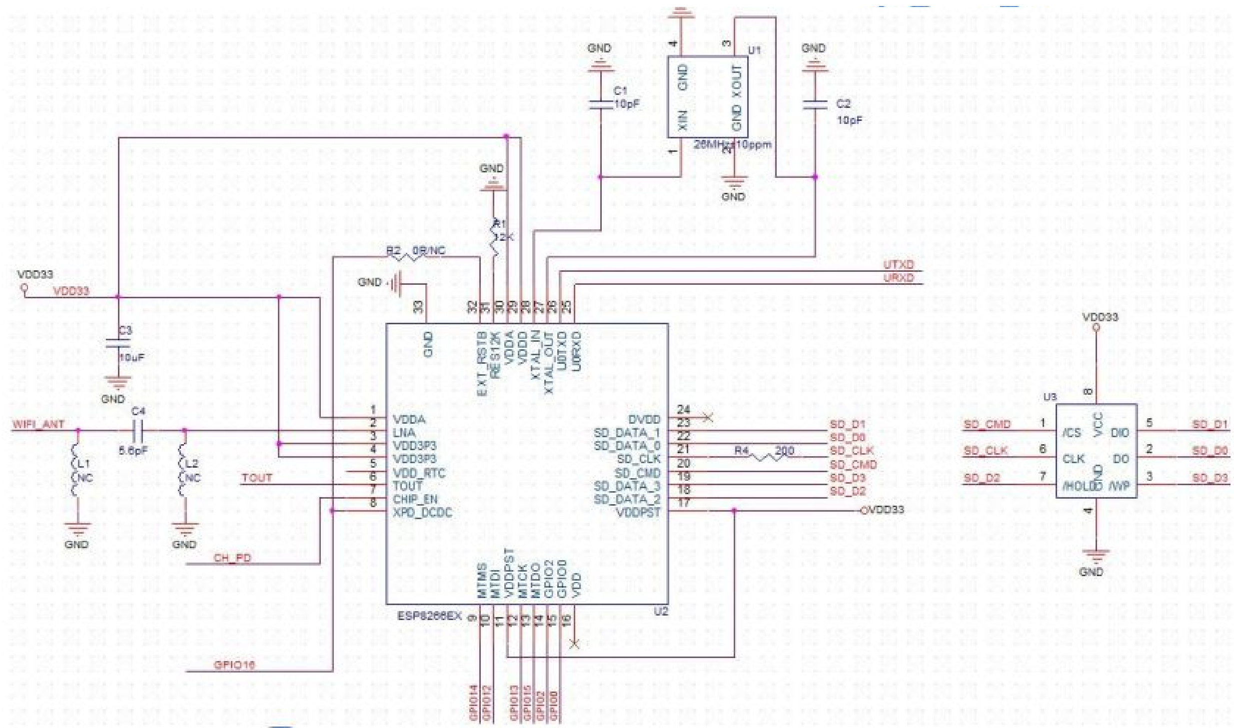

Figure 2.7. This figure represents the Wifi module. It shows the proper connections and parts such as the Wifi IC Chip, flash memory, and the antenna.

LED Array:

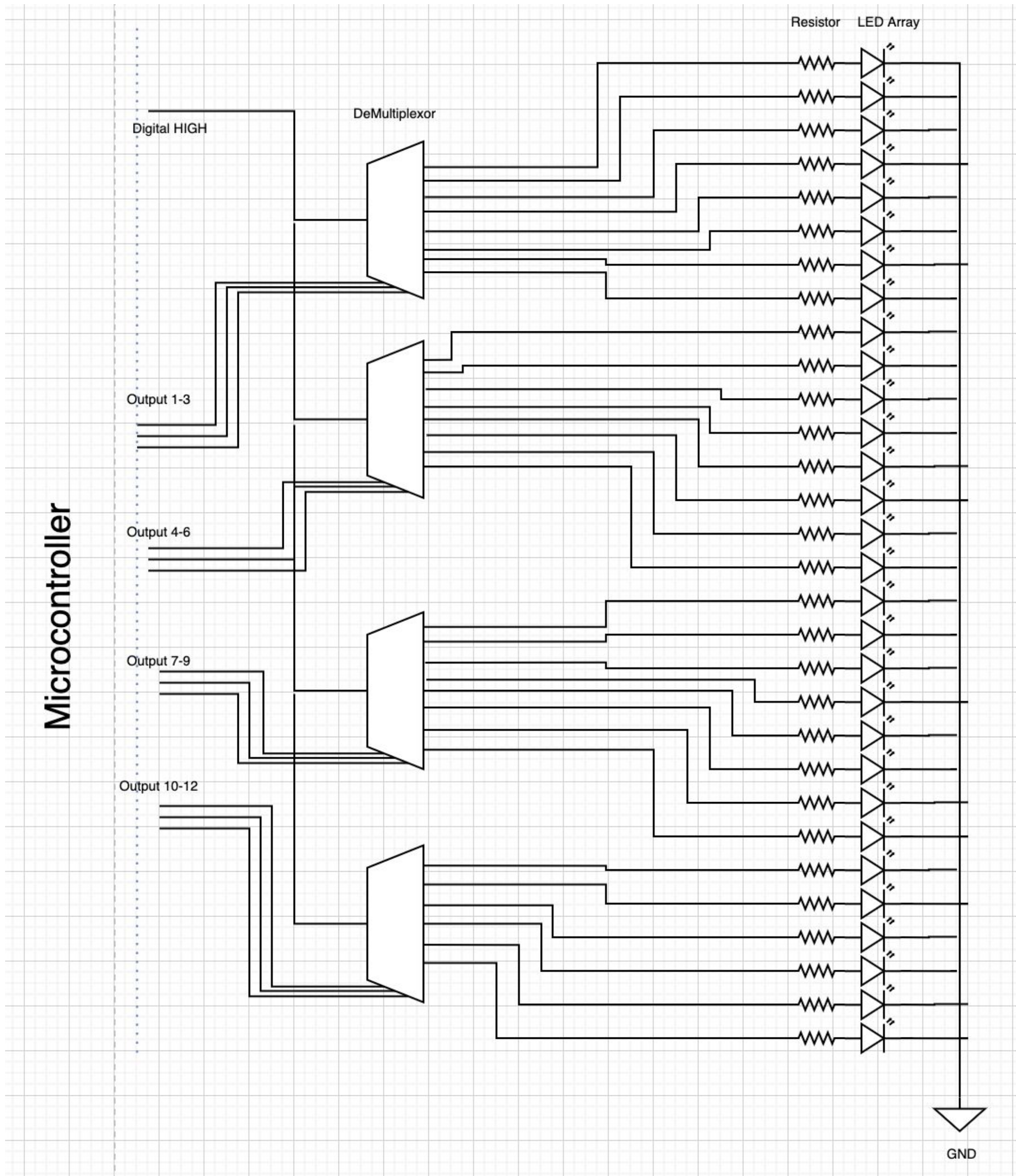

Figure 2.8. This figure shows the connections between the LED array and the microcontroller itself. There will be the use of a demultiplexer which is able to convert logic from the microcontroller, allowing the proper LEDs to be toggled.

2.6 Tolerance Analysis

There are two critical components of our design which will be the most important. The first component is the refresh rate of the LED array. Our output speed based on the calculations the microcontroller is performing has to be faster than our required refresh rate of our LEDs. Essentially the LEDs need to be updating at such a rate to match the speed of the rpm of the motor. The complication also comes in because of the fact that we have 13 available input/output pins for our given microcontroller. 1 of the outputs goes towards the hall sensor. This gives us 12 available slots. Because of this, we now need to have an even faster refresh speed. The calculation is done below. In order to create this "grid pattern" in 3 dimensional space, we need to get each LED to update 20 times per second. This is a number that the group came up with is a sufficient number to get the display we are looking for. To add another layer on top of this, we only have a limited number of outputs. Because of this, we have to use a demultiplexer. This in effect, requires a higher refresh speed from the microcontroller:

2.6.1 LED Refresh Rate

Rotations per minute of *motor* (max) - 500 rpm

Rotations per second - 500 rpm/ 60 mins/sec = 8.33 rps → .12 seconds per rotation

.12 seconds per rotation → .006 second refresh rate per LED

4 sets of 3 outputs → 12 outputs

For each set of 3 outputs

3 outputs are effecting 2^3 = 8 LEDs

.006/8 = .00075 seconds

The microcontroller has to update each output every .00075 seconds for the given rpm. If the microcontroller fails to do so, the system will fail and create a not so desirable output. Since a refresh time implies the time to change the display and hold the output, we need to ideally hold our LEDs to below this refresh rate. Anything less than .75 ms is appropriate; we may have unintended consequences with a greater refresh rate.

LED refresh rate < .00075 seconds (.75 ms)

2.6.2 Resistor Power Throughput

The second component is the power supply driving our LED array. We need to ensure that our LEDs are appropriately lit but will not burn out from receiving too much current. Since we are using a demultiplexer, our 12 outputs are split into 30. To keep power throughput low across resistors, each LED will have a dedicated resistor to drop voltage across. The resistors we plan on using are rated at ¼ W = 250 mW.

$P resistor = V resistor * I resistor$

$P resistor, min = 5\ V * 10\ mA = .05\ W = 50\ mW < 250\ mW$

$P resistor, max = 5\ V * 20\ mA = .1\ W = 100\ mW < 250\ mW$

Thus, we verify that connecting our resistors in series will stay within the resistors' power rating.

2.6.3 Resistor Resistance Values

For each resistor, we will also need to ensure the resistance is appropriate to meet the threshold for current through each LED. Current threshold for LEDs is between 10 and 20 mA. Resistance and current through the resistor and inversely proportional. Thus, resistance will be maximized when current is minimized and vice versa.

$R resistor = V resistor / I resistor$

$R resistor, min = 5\ V / 20\ mA = 250\ \Omega$

$R resistor, max = 5\ V / 10\ mA = 500\ \Omega$

This indicates to us that resistor values should be **250 and 500 Ω.**

3. Cost and Schedule

3.1 Cost Analysis:

Based on Glassdoor [4], the average graduate salary for a software/hardware engineer in Illinois is roughly $78,182. Assuming everyone is working an average of 40 hours a week, we can calculate the per hour average salary.

$$\frac{78{,}182\ dollars}{year} * \frac{1\ year}{52\ weeks} * \frac{1\ week}{40\ hours} = 37.59\ dollars/hour$$

This averages out to $37.59 an hour. The TA said there would be weeks that we end up working on our projects more than other weeks, but the average amount of time would end up looking at around 10 hours per week. Given that the class is 16 weeks long, and the group consists of 3 people, we can calculate the total amount of labor that it will cost to develop the product.

$$\frac{37.59\ dollars}{hour} * \frac{10\ hours}{week} * 16\ week * 3\ people = \$18,043.20\ dollars\ (\$6,014.40\ dollars/person)$$

Besides the cost of labor, there is also the cost of our prototype as well as the hypothetical cost of production. There are many things that would change from the prototype level to the manufacturing level, including developing plastic molds of the design for quick, easy, and cheap duplicating of the project.

Part	Quantity	Cost (prototype)	Cost (bulk)
LED- 5 mm light emitting diode round	30	$0.79	$0.04
Hall Sensor- A3144 Pin 3 Terminal Sensitive	1	$0.556	$0.15
Motor- brushless DC motor	1	$19.56	$7.83
Microcontroller- ATMega328p IC Chip	1	$5.327	$1.60
WiFi module (ESP826P)	1	$3.245	$0.95
Motor Belt	1	$2.245	$1.80
9 Volt Alkaline Battery	1	$1.373	$1.20
3-D printed parts	2	~$20	
Injection molds	2		$3.20
Ball bearings	2	$1.38	$1.02
Magnet- Super strong neodymium disc magnets	1	$1.498	$0.019
PCB print	1	~$20	~$3.50
Total cost		**$75.97**	**$21.27**

Total cost including wage would be

$18,043.20 + $75.97 = **$18,119.20**

If we were to mass produce the product, the price per unit would significantly decrease.

However, the overall price including labor would not vary significantly.

$18,043.20 + $21.27 = **$18,064.50**

3.2 Schedule:

Week	Rishab	Amar	Sriram
9/27/20	Work on tolerance analysis and requirements and verification for design document	Work on schematic block diagrams and tolerance analysis for design document	Work on requirements and verification and cost analysis for design document
10/4/20	Work in Eagle on PCB design iteration 1	Work in Eagle on PCB design iteration 1	Work in Eagle on PCB design iteration 1
10/11/20	Continue iterating on PCB	Continue iterating on PCB	Look into purchasing other hardware required for project
10/18/20	Start constructing LED array layout	Start constructing hall sensor component and connections	Start constructing the motorized housing for the LED array layout
10/25/20	Finalized PCB iteration for final order	Finalized PCB iteration for final order	Finalized PCB iteration for final order
11/1/20	Prepare for mock demo	Prepare for mock demo	Prepare for mock demo
11/8/20	Perform unit testing for each of the components	Perform integration testing for each of components	Document issues with testing and suggest changes
11/15/20	Work on additional hardware design + extra parts	Implement feedback from demo and TA	Implement feedback from individual group’s testing
11/22/20	**Thanksgiving Break**	**Thanksgiving Break**	**Thanksgiving Break**
11/29/20	Work on final paper	Work on final paper	Work on final paper

3.3 Contingency Plan:

In the case that the semester goes completely online, we have an alternative simplified design that does not require extensive use of lab equipment, including our 3-D printed design. Our alternative consists of creating a similar chassis made of electrical pipes, metal rods, and PVC pipe. Thus, our completed hardware would still have a similar form, without lack of design functionality. The size may be forced to increase as well, due to the lack of precision that we cannot emulate without a 3-D printer. In the situation we are unable to receive or create printed PCBs, we will pivot to one of two designs. The first would be to use a compact breadboard design, which would end up taking a lot more space than we originally intended. The second would be to use the Arduino Nano, which our PCB design layout is based off of. We will most likely be going with this second design pivot. This would result in functionality and logistical constraints such as size very similar to our prior expected result. Overall, despite the given pivots, we still intend to produce a slightly modified deliverable.

4 Discussion of Ethics and Safety:

There are a variety of ethical guidelines and safety measures we will have to take into account for the design and functionality of this project. As with any electromechanical hardware, there are electrical and mechanical hazards to take into consideration.

With regards to safety, we will need to acknowledge two main attributes of our physical design that we will need to mitigate.

Since our hardware body housing the LED arrangement will be rotating at a very high RPM, it will be important to disclose to users that they should not touch the holographic display while it is in operation. Any sort of interaction that involves interacting with the physical display while it is in rotation poses two issues. The first issue is one of functionality in that safe functionality is unable to be guaranteed after a user mangles with the device initially. Although they may not get injured the first time, improper handling of the device can deform and mutilate parts from their intended purpose in our device. Thus, unstable parts could pose a hazard for future operation of the device (i.e. parts are loose or electronic components are improperly grounded).

We may need to implement certain safety measures as well to ensure that younger children and less attentive audiences will not injure themselves or others. Children pose a significantly greater risk to the use of our device since they are inherently less careful than adolescents or adults [7]. Therefore, we intend to restrict the usage of our device to adults for the time being.

Additionally, the flashing LED lights at full brightness may have the potential to trigger symptoms in people with epilepsy or seizure risk. Epilepsy can be caused by a variety of hosts but only 3% of epileptic persons are photosensitive (react to flashing lights) [3]. Regardless, we will need to disclose the appropriate warnings for this portion of the population as our LED display may feature flashing LEDs when we are rapidly changing from one message to another.

Although one LED's brightness is relatively low, the combined luminance from our LED array will be rather bright. Those with sensitive eyes or other vision conditions should take caution when operating the device, and should possibly generally refrain from doing so. Our warning will also indicate that those with pre-existing vision conditions should take caution in operation.

Overally, we will need to disclose to avoid or closely monitor usage in people that are part of the aforementioned groups or mentioned health risks.

Our project also needs to uphold engineering ethical standards and address the safety, health and welfare of the public as mentioned in [1] and [2]. This would include designing the rotational component in a manner that upholds safety of all users. We would also like to improve the understanding and design of our device. Obviously, as technology improves, we would like for our device to have that goal in mind, and utilize whatever safety cues are most appropriate.

From an ethical standpoint, we will need to ensure privacy of information for our device. Since we essentially aim to display information from a web client on our display, there will potentially be private and/or sensitive information present on the holographic display at any times. We also need to ensure that the display is not able to have malicious software that pulls other information from the user’s web client. The only information going into the display should be what the user enters on the web client.

We will need to take the appropriate measures to mitigate any security risks that seem inherent especially since there will be a WiFi module through which data may be communicated to the web. We plan on ensuring the privacy of our users by making sure that the WiFi module is secured with a WPA2 type connection. Although there are some amount of vulnerabilities [9], this protocol is generally the most safe out of popularly accepted Wifi protocols.

5 Citations

[1] IEEE Code of Ethics. (n.d.). Retrieved September 15, 2020, from https://www.ieee.org/about/corporate/governance/p7-8.html

[2] ACM Code of Ethics and Professional Conduct. (n.d.). Retrieved September 15, 2020, from https://www.acm.org/code-of-ethics

[3] The Curious Case of Epileptic Seizures: What Triggers a Seizure? – Penn Medicine. (n.d.). Retrieved September 29, 2020, from https://www.pennmedicine.org/updates/blogs/neuroscience-blog/2019/august/surprising-epilepsy-seizure-triggers

[4] Salary: Software Developer in Chicago, IL. (n.d.). Retrieved September 16, 2020, from https://www.glassdoor.com/Salaries/chicago-software-developer-salary-SRCH_IL.0,7_IM167_KO8,26.htm

[5] Arduino Nano. (2018, March 12). Retrieved September 16, 2020, from https://components101.com/microcontrollers/arduino-nano

[6] ESPRESSIF SMART CONNECTIVITY PLATFORM: ESP8266. (2013, October 12). Retrieved Spring, 2020, from https://www.electroschematics.com/wp-content/uploads/2015/02/esp8266-datasheet.pdf

[7] Carroll, J., Cole, M., Cronin, L., Dawson, J., & Paravattil, B. (n.d.). Toys-R-Dangerous? What You Need to Know about Children's Products and Recalls. Retrieved September 30, 2020, from https://www.center4research.org/toys-r-dangerous-need-know-childrens-products-recalls/

[8] Takashi N., Yamamoto, Y. (2018, November 28), Retrieved September 18, 2020, from https://www.sciencedaily.com/releases/2018/11/181128082711.htm

[9] Kumkar, V., Tiwari, A., Tiwari, P., Gupta, A., & Shrawne, S. (2012, April). Vulnerabilities of Wireless Security protocols (WEP and WPA2). Retrieved September 29, 2020, from https://dl.irstu.com/wp-content/uploads/Download/Education/Book/Network/Network%20Security/WEP-WPA-Article/Vulnerabilities%20of%20Wireless%20Security%20protocols.pdf