5G Repeater Project

Final Report

Sophie Canepa, Matthew Huebner, Ray Kman, and Megan Riley

1 Introduction

1.1 The Problem

Low-emissivity glass used in modern homes and businesses can impede delivering 5G connectivity to users indoors. Low-e glass is designed to insulate homes by minimizing the amount of infrared and UV light that comes through the glass. Unfortunately, high frequency 5G mmWave signals can be reflected from the Low-E coating on the glass, resulting in a weak in-building signal. Therefore, there is a need for a 5G repeater mounted on the inside of a window to amplify services from a 5G base station into a user's home inexpensively.

5G networks intend to incorporate 5G base stations to provide localized wideband internet access to residential areas. The demand for high-speed data transfer motivates the development of highly directive antenna platforms with low power consumption to implement beamforming capabilities. Current technologies use beamforming to focus the wireless signal in a chosen direction, normally towards a specific receiving device. Typically beamforming is performed using phased array antennae systems, which requires a large array of antennas treated as active elements with phase control to change the direction of the beam. As a result, building phased array systems at high frequencies with many antenna elements is subsequently very expensive. Our solution offers an inexpensive alternative.

1.2 Overview of Solution

Using Dr. Garcia and Dr. Chisum's novel lens antenna prototype, a simple, linear feed antenna can be used in series with gradient index (GRIN) lenses to beamform inexpensively and passively without sacrificing performance. GRIN lenses achieve wave collimation by means of spatially varying index of refraction to collimate a plane wave to a single point. Dielectric lens antennas offer another approach to beamforming applications since they have high directivity with minimal power consumption.

This demonstration will prove that the GRIN lens antenna can be used to reliably receive mmWaves from a base station and effectively amplify the signal in an indoor environment. This lens technology allows for a signal to be detected through modern low-emissivity window glass. The overall system will include a window-mounted 5G repeater that relies on a stepper motor to scan feed antennae positions to optimize the direction of the beam formed by the lenses.

1.3 Reflection

Overall, our design met our system requirements we made for the project. The lens antennae accurately scanned the entire length of the PPWG, stored the location at the peak signal, and returned to this location with success. The LEDs displayed the strength of the signal as new values were found, and displayed how the strength was maintained after the lens antennae stopped at their desired location. The system was accurate in finding the position of the collimated beam.

There were some aspects of the design that could have been improved in a future iteration. The method of mounting the PPWG and linear rail to the window with zip ties and suction cups worked, but sometimes the lens antennae shifted the PPWG out of position because the PPWG was not secured tight enough to the window. This would return suboptimal results because the correct maximum value recorded would change after the PPWG shifted. In a future iteration, a connection between the linear rail and the PPWG would help to mitigate this shifting problem. While the signal strength was displayed accurately by the LEDs, it could have been displayed in more detail if the range of values defined in the code was better suited to our application (as explained in more detail at the end of section 3.3.1).

2 Detailed System Requirements

The Feed Translation Mechanism required our group to integrate (1) a power sensor that takes input from the GRIN lens technology, (2) a functional microcontroller to control the feed translation, and (3) a linear rail stepper or threaded rod/servo motor. We decided to go with the stepper motor to achieve a discretized step size that could take very small measurements along the linear rail. The purpose of this system was to determine the optimal position of the lens in order to provide the optimal signal for receiving data and move the lens antenna accordingly. The position would be the same for the transmitted data. This was achieved successfully with our feed translation solution.

In order for our prototype to be completed, it needed to be able to be mounted to a window. This required the design of a mounting system including hardware for the electronic components, linear feed mechanism, and PPWG. This system required 3D printing and CAD work, as well as suction cups and zip ties for window attachment.

The microcontroller and related functions required coding in an ESP32 environment and debugging any issues that arose along the way. In addition to this, the control of our feed translation mechanism and LED display were very important components. The microcontroller also needed to connect to the stepper motor driver and be able to parse input from the power sensor using an on-board ADC.

The board design needed to accommodate analog-to-digital communication lines between the power reading device and our microcontroller. In order to be able to properly detect the position at which the mmWave signal is collimated using the PPWG and GRIN lenses, the power along the PPWG must be able to be sampled accurately. After being sampled, this value must then be converted into a digital value that can be read by the microcontroller and allow the linear actuator to find the position where the power is maximized along the PPWG. Additionally, the board needed to connect the microcontroller to the feed translation mechanism and provide LED status updates to the user. The LED output was the only external communication about the signal strength, so this was necessary for proper communication and use.

In order to direct our desired signals between GRIN Lenses we needed to design a low-cost simple fabrication Parallel Plate Waveguide. This was our 3D printed waveguide with metal layerings between the plates in order to form the necessary waveguide. We needed appropriate connectors to be able to clamp these waveguides together tight enough to prevent loss, but not so tight that a wave could not propagate through the plates.

An indoor antenna to transmit and receive in indoor space was needed for our project. We needed a wideband antenna for the best possible signal generation. Our omnidirectional antenna achieved this in an optimal way, as we could clearly see the output from the Parallel Plate Waveguide on a spectrum analyzer elsewhere in the room when we were testing our product.

3 Detailed project description

3.1 System theory of operation (how the whole thing works)

The 5G repeater device will be mounted to a window to receive mmWave signals from the outside as well as transmit mmWave signals back through the window. The mounting was accomplished using suction cups and zip ties to the hooks 3D-printed to the parallel plate waveguides (PPWG). Once secured and flush to the window, the millimeter waves enter the PPWG and travel through the GRIN lenses to collimate the beam. The mmWave is fully collimated after passing through two of the GRIN lenses and will focus at a singular point along a 1-Dimensional axis that holds the lens antenna. From here, the focused signal will be sent through two 25 dB gain amplifiers for a total of 50 dB of gain before having 20 dB siphoned off by a directional coupler.

From here, the signal will follow two different paths. The first path sends the rest of the RF signal to an omnidirectional antenna that propagates the wave into the indoor environment. The second path is the 20 dB siphoned off with the coupler which will be subsequently fed into the LTC5596 where it is converted from the RF logarithmic input into a linear, analog DC signal from 0 to 1.2 V. This analog signal will be sent to an unity gain operational amplifier before reaching its final destination, the ADC pin on the ESP32 microcontroller. The benefit of a unity gain operational amplifier is that this configuration has infinite input impedance, which ensures that the current into the ESP32 ADC pin is below the max current draw for that pin and that the voltage read on that pin is accurate. Here, the signal is digitized and displayed on the serial port as well as the LED indicator light.

This value is compared against the previous maximum RF value stored during each iteration of the stepper motor's movement. If less, it is ignored, but if greater, the x value of the position of the lens antenna, along with the max value will be stored in the software. During each power on, the stepper motor will drive a linear actuator that moves the lens antenna along the length of the 1-D axis of the PPWG. The motor will always move the antenna towards the end with the limit switch during power on, in order to calibrate its position. Once calibrated, the motor will rotate the opposite direction and scan the entire length of the PPWG. The scan will determine the collimated focal point along the axis and move the antenna there. Once fixed in the correct position, the antenna is ready to receive the signal.

On the transmit side (Tx), the signal is picked up by a parallel transmit chain that has an identical omnidirectional antenna to the receive side. The mmWave signal from the indoor environment will be amplified through two 25 dB gain amplifiers before being sent into the transmit side of the PPWG. The GRIN lenses work in the opposite direction as well, allowing us to beamform the transmit signal correctly to the base station outside.

3.2 System Block diagram

A system block diagram of all the major components in our system is included below, detailing the separate transmit and receive components of the system.

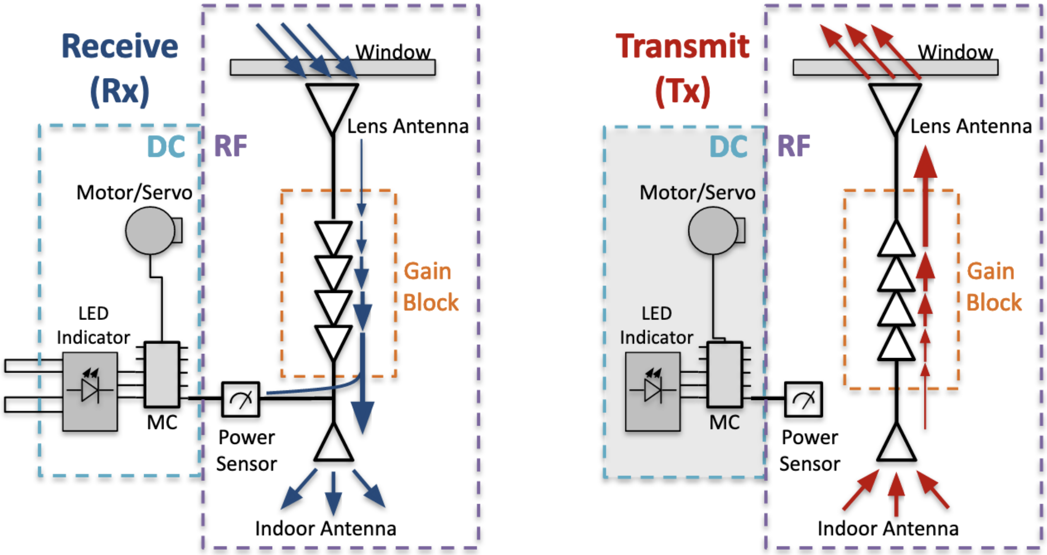


Figure 1: System Block Diagram

3.3 Detailed design/operation of Feed Translation Mechanism

3.3.1 Software

The feed translation mechanism's central role is to move the lens antenna across the 1-D axis of the PPWG. This is accomplished with the Nema-17 stepper motor, A4988 stepper motor driver, limit switch, linear actuator rail and the 3D printed antenna holder. The Nema-17 motor is mounted to one end of the aluminum linear rail and is held taut with a band that has teeth which can grip the rotor of the stepper motor. The band runs the length of the rail and wraps around a wheel that holds the band at the other end. The 3D printed component is held in series with the band allowing it to move with the band. This will be the way the lens antennas move.

The motor interfaces directly with our custom PCB board via four header pins that connect it with our A4988 stepper motor driver. 12 V and 1.5 A are provided to the motor to adequately power it. The routine described using the flow chart below was used to control the motor and accurately determine the best place to position the antennae for the highest signal amplification. The Ardunio IDE was used to compile and upload our C code to the ESP32. From a software perspective, the ESP32 made sense for our project due to its ease of communication with individually addressable WS2812 LEDs and the easily accessible on board ADC pins.

The software for this project follows the logic diagram shown in Figure 2. Once the board is powered, the desired pins and variables are declared, and the serial port is set up. The reset and sleep pins on the motor driver are held in the HIGH position using output from the MCU throughout the routine to allow for the motor to run properly. To achieve the desired step size, MS1, MS2, and MS3 are held in the LOW position. These parameters control the microstep resolution, and in our software we used a full step resolution. After setup, the motor scanning code begins to execute. This motor routine will run once to set up the position of the antennae where the mmWave signal coming from the other side of the window is strongest.

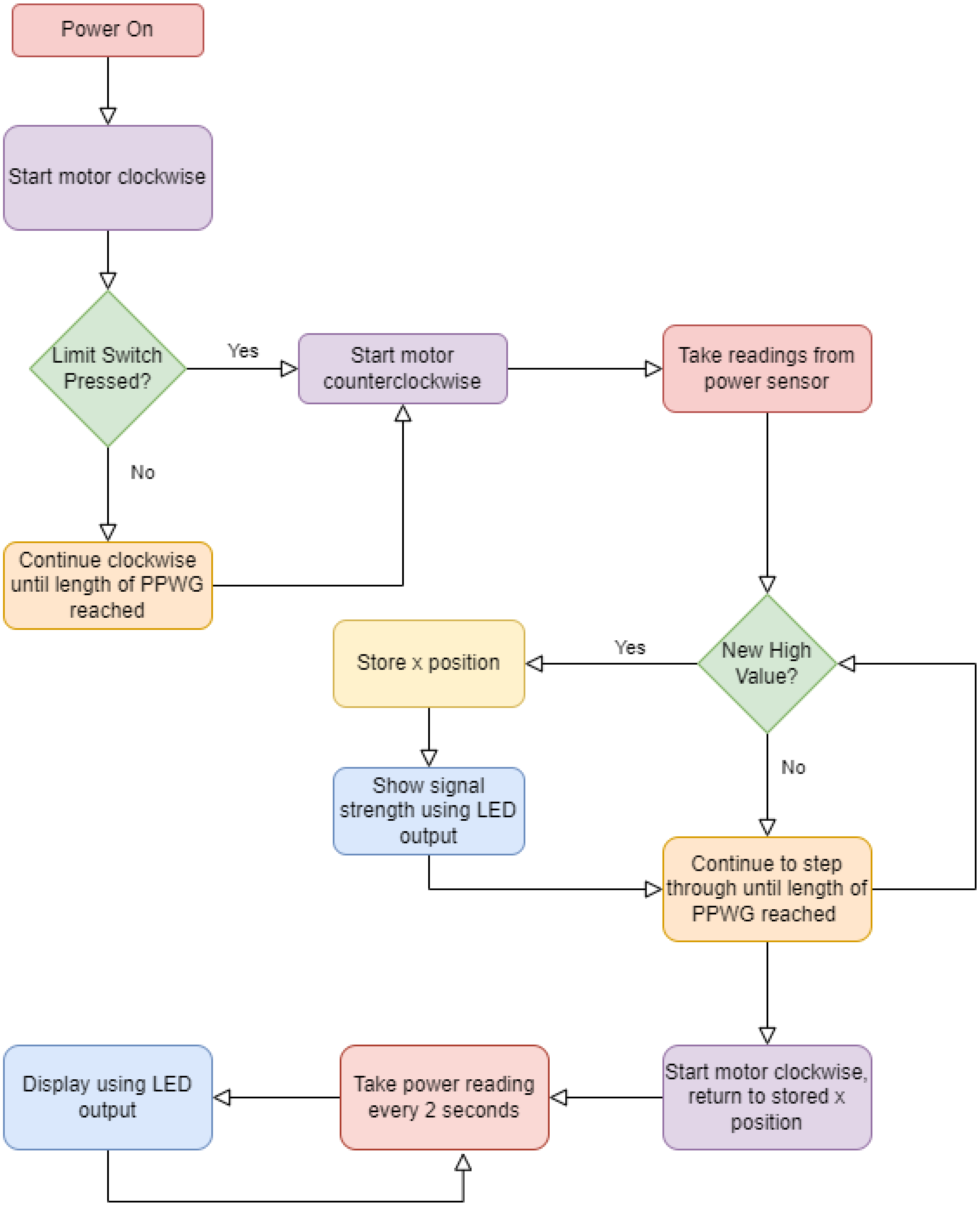
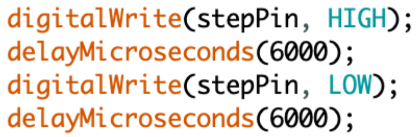

Power On
Start motor clockwise
Limit Switch Pressed?
Yes
No
Start motor counterclockwise
Take readings from power sensor
Continue clockwise until length of PPWG reached
Store x position
Yes
New High Value?
No
Show signal strength using LED output
Continue to step through until length of PPWG reached
Display using LED output
Take power reading every 2 seconds
Start motor clockwise, return to stored x position

Figure 2: Software Logic Flow Diagram

The routine to move the stepper motor involves sending alternating high and low pulses to the STEP pin of the motor driver. A 6,000 microsecond delay is used for the switching speed and one step is accomplished as shown below.



```
digitalWrite(stepPin, HIGH);
delayMicroseconds(6000);
digitalWrite(stepPin, LOW);
delayMicroseconds(6000);
```

Figure 3: Step Example Code for Nema-17 Stepper Motor

This one step shown in Figure 3 is repeated until the end of the PPWG is reached. For our 150 mm distance, 755 steps were used in a for loop to traverse to the other side of the PPWG. This allows for very fine adjustments on the order of tenths of a millimeter in step sensitivity.

At the start of the routine, the motor is enabled from its off position and is engaged. The direction pin is set HIGH to turn the rotor in a clockwise direction and pull the band it is connected to, using the teeth on the band to gain traction. It first pulls the band towards the limit switch, which is placed at the end of the rail opposite the stepper motor. Once the connector piece, shown below, activates the limit switch, the motor switches directions by adjusting the direction pin and begins to take power measurements the entire length of the PPWG.

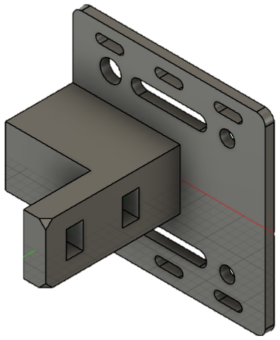
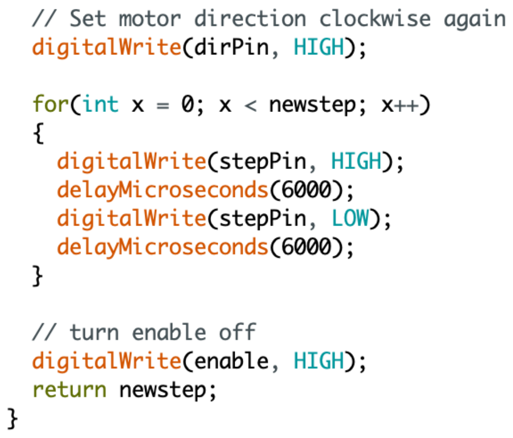


Figure 4: Lens Antenna Connector

The RF values are converted by the power sensor and fed into the MCU. Incrementing as described above, when a high value is detected, the x position where this value occurs is stored. Once the scan is complete, the below routine moves the lens antenna to the stored position corresponding with the highest RF signal input.



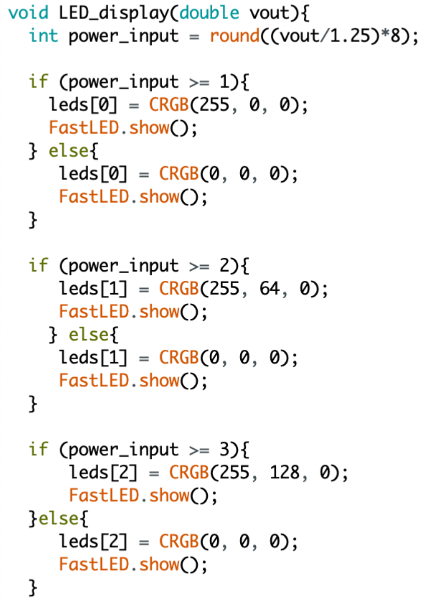
```
  // Set motor direction clockwise again
  digitalWrite(dirPin, HIGH);

  for(int x = 0; x < newstep; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(6000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(6000);
  }

  // turn enable off
  digitalWrite(enable, HIGH);
  return newstep;
}
```

Figure 5: Nema-17 Stepper Motor Returning to X Position

Here, the *newstep* variable stores the x position of the max value which is the step within the iteration that the antenna encountered with the highest RF signal value. This removes any need for time dependencies, allowing the motor to provide consistent and accurate scans based on the motor step count. The motor is then disengaged by setting the enable bit to HIGH on the motor driver in order to avoid excess power consumption.

In order to rerun the scan on the current prototype, a power cycle is required. This can quickly be accomplished by replugging in the barrel plug. In a future iteration, a rescanning button will be incorporated to easily re-run the routine without power cycling. After the motor setup ran, the power level continued to be updated and displayed by the LEDs in a continuous loop until the system was power cycled.

The external LED code routine that displayed the power intensity of the RF input employed the use of a few if statements to determine up to what threshold the input signal exceeded. The vout variable stored the analog RF signal value scaled from 0 to 1.2 V. Thus, the 8 LEDs could be partitioned into 8 different thresholds as seen below.



```
void LED_display(double vout){
  int power_input = round((vout/1.25)*8);

  if (power_input >= 1){
    leds[0] = CRGB(255, 0, 0);
    FastLED.show();
  } else{
     leds[0] = CRGB(0, 0, 0);
     FastLED.show();
  }

  if (power_input >= 2){
     leds[1] = CRGB(255, 64, 0);
     FastLED.show();
    } else{
     leds[1] = CRGB(0, 0, 0);
     FastLED.show();
  }

  if (power_input >= 3){
      leds[2] = CRGB(255, 128, 0);
      FastLED.show();
  }else{
     leds[2] = CRGB(0, 0, 0);
     FastLED.show();
  }
```

Figure 6: A portion of the LED Display Code

As seen in Figure 6, the input voltage from the power sensor is scaled to be a number between 0 and 8. A linear scale was implemented because the power sensor already converted the signal strength on a exponential scale to a linear voltage scale. When the voltage increases, the LEDs are set to colors that go from red to green to display the signal strength. Figure 6 shows the first 3 LEDs and their corresponding colors. If the signal is above the threshold for the specific LED, the color defined in its statement will be shown. If the signal is not at this threshold, black (0,0,0 on the RGB scale) will be shown on the LED, and there will be no output on that LED.

Upon implementing this system, it became clear that the power sensor was only reading values up to around 0.3V when the millimeter wave signal was at a maximum. This was because only a a portion of the signal collimated by the GRIN lenses was actually read by the power sensor, and the rest was sent to the omnidirectional antenna to be amplified in the room. Scaling the LED output from 0 to 0.3V instead of 0 to 1.25V would show a better resolution of the signal strength using the LEDs.

3.3.2 Hardware

On the hardware side, the A4988 Stepper motor driver and MCU were the two main components to be laid out using the EAGLE CAD software. The MCU pins (left) connected to the A4988 (right) included:

- GPIO4 → Direction
- GPIO16 → Step
- GPIO17→ Sleep
- GPIO5 → Reset

- GPIO21/19/18 → MS1/2/3
- GPIO23 → Enable

These GPIO pins allow for the digital control of the motor driver from the MCU with very little pins to worry about. The original decision to select the A4988 was two fold. First, the driver could support/drive the Nema-17 stepper motor selected for the project, and two, there was an evaluation board available that allowed for quick prototyping in order to work out the kinks with the software side of things. The ESP32 was chosen as it was a great universal MCU that provided included libraries in the software interface.

Laying out the MCU was straightforward and is hashed out further in the section below. The A4988 proved to be much more difficult, as there were numerous capacitors and resistors to correctly lay out. An image of the PCB layout is shown below, showing just how complex laying out the stepper motor was.

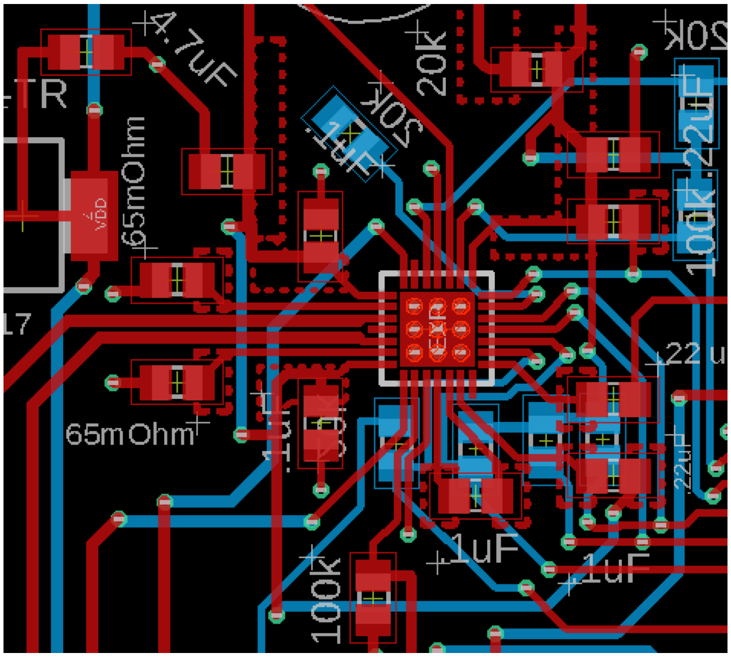
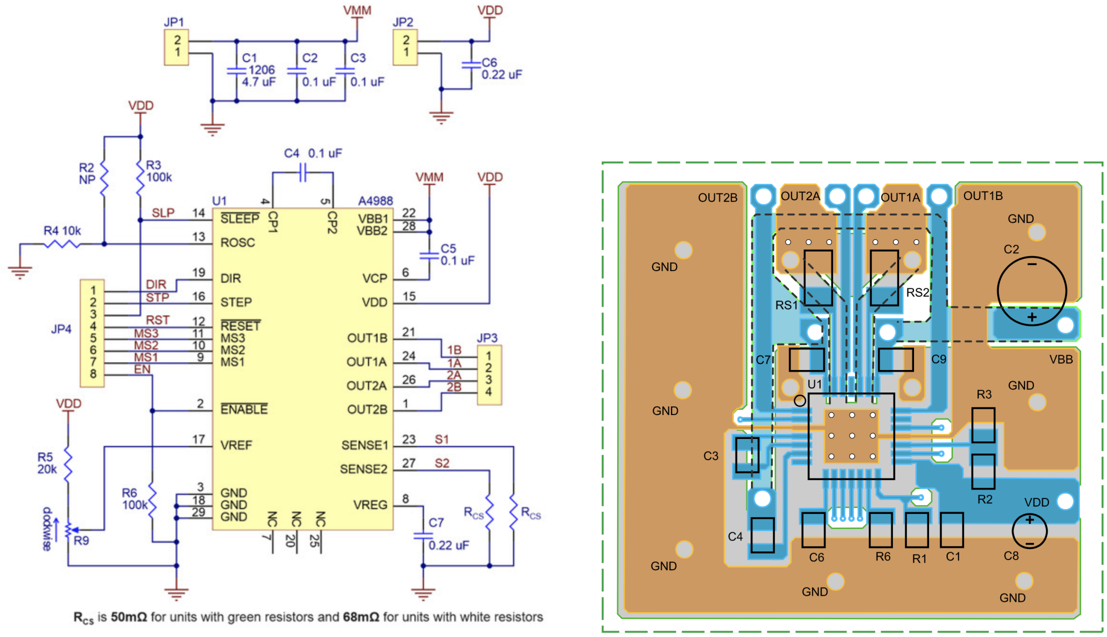


Figure 7: Stepper Motor Driver Layout

One of the first design considerations was to decide if the board had to be four-layers, as the datasheet recommended, or if a two-layer design would suffice. The four-layer design was originally created to combat the thermal heat that would be generated from the motor driver. Given the use case for the motor was to only run it once on the initial scan, it was decided the motor would never get to a point of overheating, allowing the two layer board design to move ahead. As a precaution, star grounds underneath the QFN package were implemented as thermal vias to dissipate any heat that may be generated from the motor running.

The rest of the capacitors and resistors were laid out by cross-referencing various application diagrams available for the A4988. The datasheet along with the Pololu website were extremely helpful in understanding and determining the appropriate values.



R_{CS} is 50mΩ for units with green resistors and 68mΩ for units with white resistors

Figure 8: Schematic diagram (left) and PCB layout (right) of the A4988 stepper motor driver

Between these two diagrams in Figure 8, the values and positions of the components were laid out accordingly. The large capacitor seen on the right hand diagram labeled C2, was the electrolytic capacitor that protected the motor driver from voltage spikes created during the operation of the motor. Based on the recommendation of the datasheet, a large 100 uF capacitor was implemented for this.

Another design consideration was the ROSC pin on the motor driver which set the internal PWM current control circuitry's on/off time. The decision to run a 20k Ohm resistor allowed t_{off} to be in the range of 24 us and be in automatic Mixed decay for all step modes except full step. The potentiometer on the VREF pin also was involved in the current limiting configuration. The value of this potentiometer, however, was copied directly from the potentiometer on the evaluation board and did not lead to any issues.

When the custom PCB board was initially being tested, the speed at which the motor moved the lens antennas was extremely slow and was not covering the entire length of the rail. The first thought that crossed the team was that the motor driver was wired incorrectly. This was quickly debunked, though, as it was a matter of setting the Microstep resolution pins explicitly to 0 (MS1/2/3) to achieve full step resolution. By default, leaving the pins floating allowed for some of the MS pins to drift high and make the microstep resolution much smaller, hence the slower speed and inability of the motor to cover the entire length of the rail. By making the step resolution smaller, more steps can be made over the same distance, leaving our fixed step value from being the entire length of the rail, to just 1/16 of it. In future iterations, if finer steps are needed, this motor driver is more than capable of providing the functionality given that it was operated at full step resolution with no sensitivity issues.

3.4 Detailed operation of Microcontroller and Associated Functions

The microcontroller is the crux of the custom PCB board. Without it, none of the motor driving or max RF signal value algorithms would run. On the hardware side, the board is laid out at the edge of the PCB so that the antenna can hang off the edge to reduce interference. Initially, this was not done and forced the team to redesign the board to account for the interference issue. The MCU is powered with a 3.3 V input from our voltage regulator that steps down the voltage from 12 V.

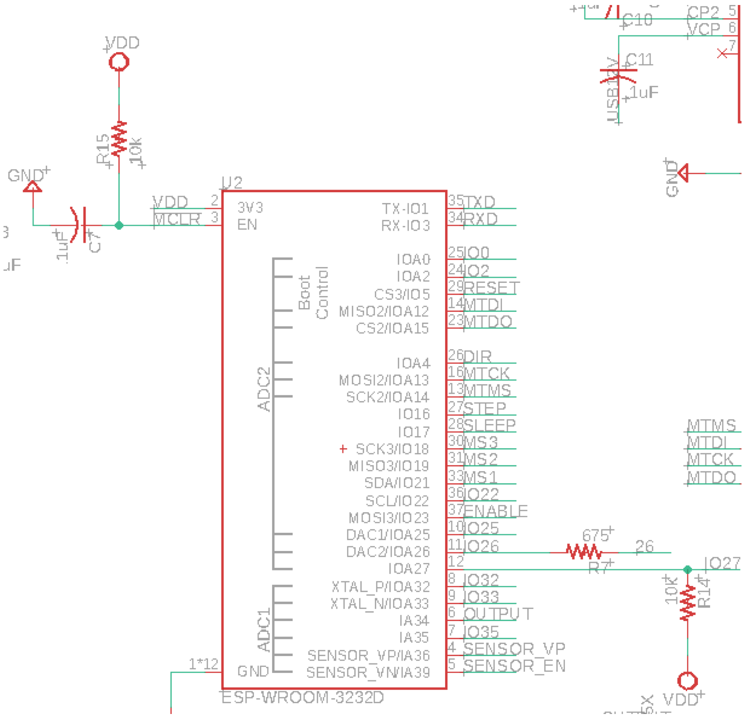


Figure 9: ESP32 Schematic

The EN pin is the reset pin and is relabled !MCLR to distinguish it from the Enable pin on the motor driver. To clarify, the ENABLE pin or GPIO23 is the pin that controls the reset on the motor driver. The unity gain operational amplifier is attached to GPIO34 or OUTPUT in the bottom right of the schematic. This is the pin that will act as the ADC and convert the analog DC input of the RF signal into a digital value that can be interpreted by the MCU's other algorithms. GPIO26 is the LED output pin that controls the strength and color of the LED strip. This pin is regulated with a 675 Ohm resistor as a precautionary current limiter. Outside of the motor driver pins described in the above section, a few key programming pins are left to be mentioned. In order to program the ESP32, a FTDI UART-to-USB programmer is used to communicate from the Arduino IDE to the MCU. Specific header pins were set aside to do this programming as seen below.

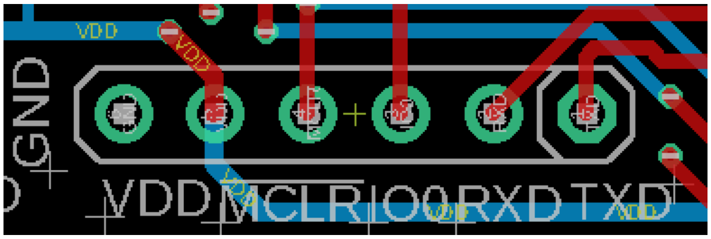


Figure 10: Programming Header Pins

The necessary pins include VDD, Ground, Reset, GPIO0, RX and TX. By establishing this header pin configuration at the edge of the board, the team was able to easily program the on-board ESP32 with little trouble. In fact, to test initial functionality of the board past the on board LED lighting up, a simple random LED cycle routine was uploaded to test the ESP32 and external LED on GPIO26. This routine showed that not only could the board be programmed but also output high and low values to the GPIO pins. Luckily, the MCU did not have any significant issues driving the motor or powering the LED.

However, the board did have issues with the ADC and unity gain operational amplifier. The primary issue here was that the operational amplifier designed into the board, despite having the same footprint, differed in its leads. The two op-amps are compared side-by-side below.

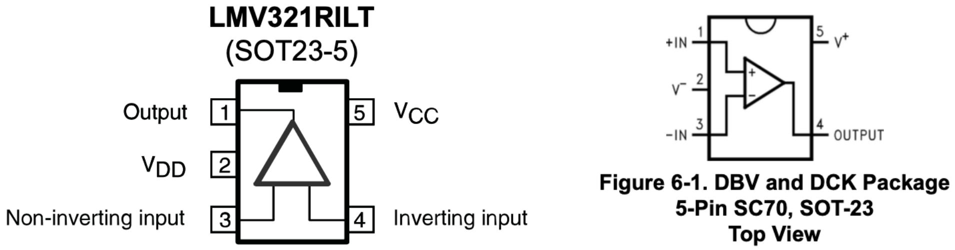


Figure 11: Comparison of Unity Gain Op-Amps

The LMV321M5X (right) was designed into the board, but the LMV321RILT was ordered. Thus, as can be seen, the pin configurations are quite different and led to the inability of the analog input value to be read correctly by the MCU. This error was not noticed until after the first board that the team constructed was burnt out. The MCU was torched when the team attempted to apply a voltage to the op amp input pin. When we initially tested our board, everything worked as expected and we were able to program to the ESP32. However, as soon as we tried to apply a voltage to the unity gain buffer, we ran into an issue where the ESP32 would continually reset itself for no apparent reason. Upon looking closer at the issue, we realized our operational amplifier was getting very hot to the touch. After this issue occurred, our code did not work as expected until finally nothing would run. The MCU was deemed inoperable as the MCU fell into a continuous reset and brownout loop whenever powered on. We built another board using the extra components we thankfully ordered, and the MCU compiled as expected and we no longer had the issue. Given that the VDD was also incorrectly wired, this most definitely led to the op amp being scorched as soon as the voltage was applied, snowballing into the MCU subsequently after. In order to get around this issue, a separate through hole LMV358 was breadboarded into a periphery circuit and proved to be a quick fix. In the future, either the part on the board should be rerouted or have the correct part ordered. In the revised EAGLE file, the correct part (LMV321M5X) is recommended to be ordered.

3.5 Detailed operation of Parallel Plate Waveguide

The PPWG's role was to collect the incoming mmWave plane wave and focus the signal using the GRIN lenses for the lens antenna in an inexpensive form factor. In the transmit version, the PPWG was required to beamform the signal to the base on the outside. The main body of the prototype was provided by Dr. Garcia, who fabricated a metal CNC version for his research. Thus, this CAD design was incredibly easy to port over into an inexpensive, 3D printed structure.

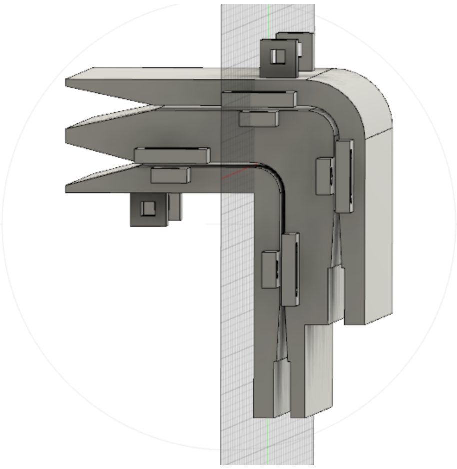


Figure 12: CAD Design of PPWG

The propagation surface was able to be fabricated using super glue and 5 mil copper FR4. During the physical construction of the PPWG, the copper proved to be difficult to work with and had to be redone a few times. The main issue was that the copper would pop off at the 90 degree bend of the PPWG and lead to the narrow passageway being pinched and unable to carry mmWave. This was resolved by applying significant quantities of superglue to hold the copper in place. A second issue dealt with the clearance of the lens antenna within the slit of the PPWG where it was supposed to sit. The opening unfortunately had to be tight to prevent the mmWave from getting around the lens antenna but also led to the antenna being pinched, adding unnecessary strain to the motor to scan the length of the PPWG. In the future, an additional .1mm clearance would help enormously, without significantly affecting the performance of the repeater.

3.6 Detailed operation of RF Circuitry

The role of the RF circuitry is to amplify the incoming RF signal, manipulate it for the power sensor, assign a strength value to the signal and propagate the signal indoors and outdoors.

On the receive side, the signal follows two chains as described in Section 3.1, power sensing and indoor propagation. Both of these routes receive the collimated signal from the lens antenna which is amplified 50 dB in the process. This amplification is done by the HMC1040LP3CE, a GaAs pHEMT MMIC low noise amplifier (24 - 43.5 GHz) which each do 25 dB in gain. By putting two of these in series, the team is able to achieve 50 dB in gain, which is then passed through a PE2CP1126-20 directional coupler. This device samples -20dB from the mmWave signal and allows the rest of the signal to pass through to the waveguide phalanges which are used as an omnidirectional antenna on the inside of the window.

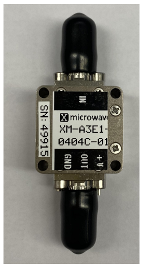


Figure 13: LTC 5596 RF Amplifier

The power sensor path receives the -20 dB from the directional coupler and puts it through the LTC5596 which is an ultra wideband (100 MHz to 40 GHz) RMS power detector with the below characteristics.

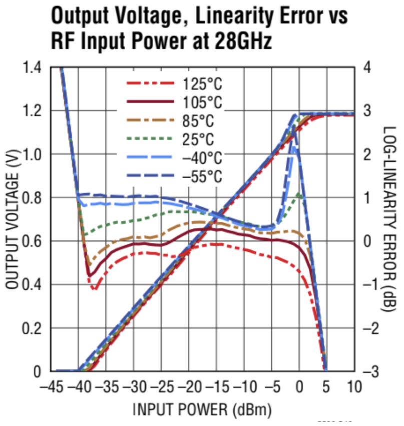


Figure 14: Voltage Characteristics of LTC5596

This graph shows that the sensor can take an input power of -40 to 0 dB and output it linearly to a DC voltage scale of 0 to 1.2 V at 28 GHz, the target frequency. The DC output of this device is connected to the microcontroller using a female SMA adapter with solid core wire soldered onto the ground and input conductors.

In order to power all of the RF components multiple solutions were approached. An original idea was to use a 9v battery to power an LDO that would drop the 9v down to the 3.3v required by all of the active RF components. This however was avoided due to the lack of current limiting protection in this solution so the agreed upon solution was to use a DC power supply with current limiting capabilities.

A finding that the team had after interacting with the power sensor was that input power would almost never reach the 0 dB level as much of the signal was attenuated by the window despite the after the fact amplification. Furthermore, the window proved to be a significant attenuator at around 40 dB of loss, solidifying the need for window repeaters to recover the mmWave signal. Thus, the entire DC output voltage range could never be fully utilized given the strong attenuation from the window. In the team’s demonstration, a 14.5 dB RF signal was the max that could be produced at 28 GHz, limiting the power sensor range to about half of what it was capable of. For future iterations, serious consideration should be given to the input RF signal strength and if a different power sensor should be used.

3.7 Detailed operation of Mounting Hardware

The mounting hardware is to serve one purpose, to inexpensively and securely attach the 5G repeater system to the window. The design the team decided to go forward with was a hook and zip tie system that would provide flexibility and robustness to the overall structure. Small cube-like hooks were attached to the top of the PPWG and on the inside as well.

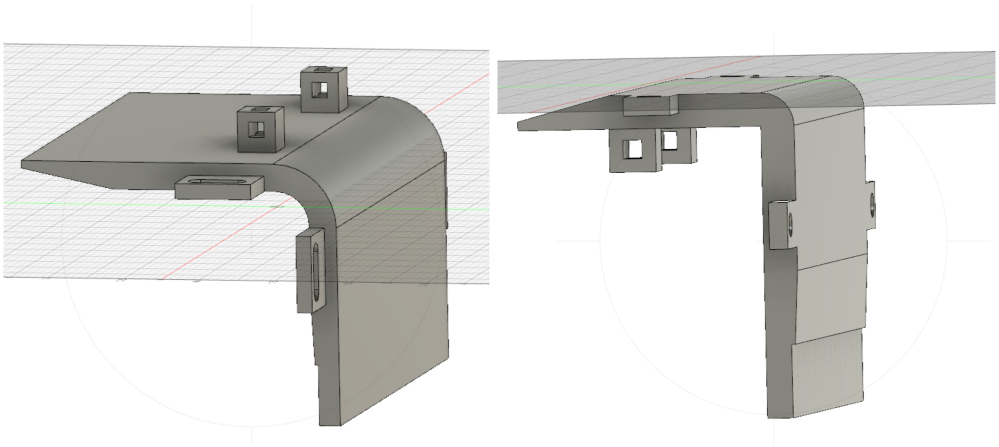


Figure 15: CAD of PPWG

The top hooks would zip tie to the suction cups that were mounted ahead of time on the window. The inner cube hooks zip tied to a custom linear rail holder piece that was 3D printed in the EIH Lab. This provided support to both sides of the linear rail to prevent any slanting.

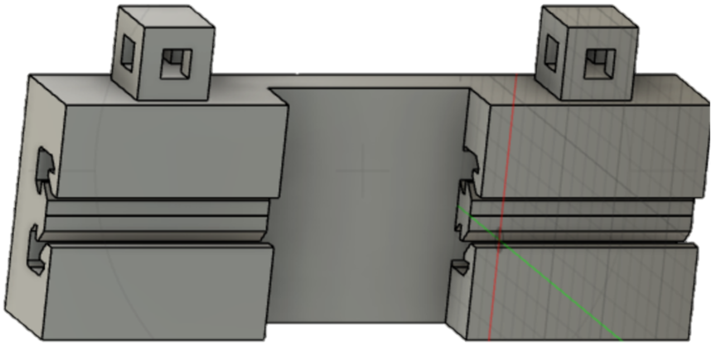


Figure 16: CAD of Linear Rail Connector

On the backside, the RF daisy chains were held in place by the hooks seen in the image below.

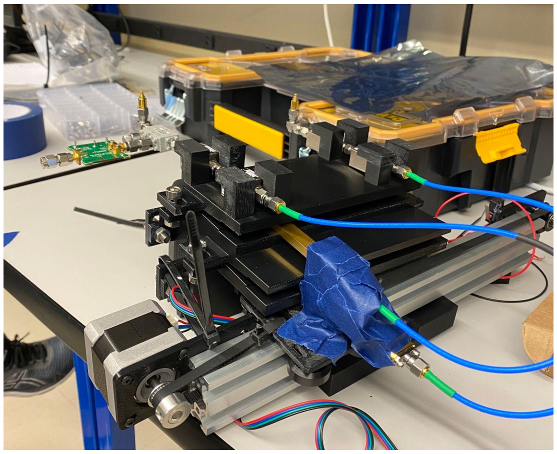


Figure 17: Physical Implementation of RF Daisy Chain Hooks

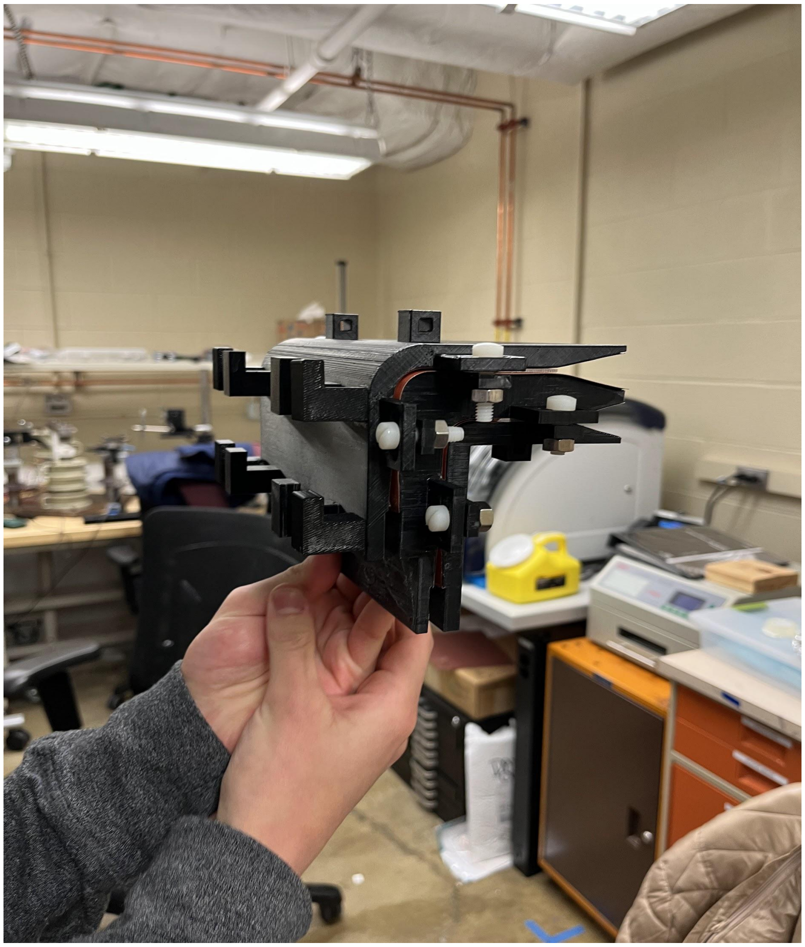


Figure 18: Assembled 5G Repeater Prototype

These were fabricated with the depth of the amplifiers in mind, in order to make sure the components were securely held in with the help of gravity. The idea of these hooks was to support the RF chains at the amplifier components where most of the weight resided. Luckily, the distance between the amplifiers in the chain was less than the height of the PPWG, allowing for two pairs of the hooks to be used for each chain. These hooks were quickly superglued and deployed for use in the system. Much of the troubles the team faced with the mounting aspect of the project was the lack of consistency or ability to place the repeater on the window without some sort of slant from the PPWG or the linear rail. With the rail and PPWG not in line, the lens antenna would often have difficulty scanning the length of the PPWG as the antenna would get cockeyed in the slot, preventing any further movement. In a future iteration, the team discussed printing a component that would stretch from inside of the PPWG to the linear rail in order to ensure the two would always be in line. By having these two parts lined up from the start, it would save enormous amounts of time in set up and allow for consistent, replicable operation of the scan.

3.8 Interfaces

In order to interface the RF power readings with the DC microcontroller, a power sensor that provided an analog DC output was sampled using the built in esp32 ADC. This solution provided a link between the mmWave signal on the receive side and allowed for a beamscan using a linear actuator to find the appropriate position of the collimated beam. For more information on the power sensor see section 3.6.

4 System Integration Testing

4.1 How the integrated set of subsystems was tested.

Our initial project testing was completed by testing each subsystem individually, both using an Arduino and PIC32 to control each subsystem. After a brief stint testing the components with a PIC32 microcontroller, we decided the individually addressable LEDs and integration with the ADC pin would be much more straightforward on the ESP32. Once we ensured each of these components worked, we integrated them together using the ESP32 development module. The A4988 stepper motor driver breakout board was implemented on a breadboard and the corresponding pins were connected to the ESP32. We also used a LMV358 unity gain buffer on the breadboard to test voltage inputs and how they displayed on the LEDs. The LEDs were connected using 3.3V output from the board, ground, and a data wire to send the display protocol.

Once all of the subsystems were individually tested, all the subsystems were tested using an analog voltage source from a power supply to see if the device would be able to return to a correct position after seeing a peak voltage. This allowed for verification that the DC side of the device was working properly.

After the board was completed and device assembled, the device was tested using the mmWave lab’s equipment. This setup involved using a signal generator set to the target frequency of 28 GHz at a power level of 14.5 dBm. This signal was incident upon the window where the 5G repeater was mounted. The signal from the repeater was then measured using a spectrum analyzer set to look at 28 GHz. Testing was then done to verify that the device could appropriately beamscan by changing the angle at which the signal source was incident upon the outside of the window.

4.2 How the testing demonstrates that the overall system meets the design requirements

After testing was concluded, the device was able to appropriately beamscan to find the correct position of the collimated signal along the PPWG using mmWave test equipment.

5 Users Manual/Installation manual

5.1 How to install your product

Installation of the 5G repeater is quite straightforward. Securing the suction cups to the window is the first step. Place two suction cups on the top part of the window near the frame about 150 mm apart. Next, secure two more suction cups equal with the PPWG and about 50 mm away from each edge of the PPWG. At this point, zip tie the linear rail connector to the hooks on the inside of the PPWG. This will make the rail and PPWG into one object. Next, begin running the zip ties from the suction cups to the hooks on the top side and to the side mounts. The PPWG and linear rail should now be securely mounted to the window. Adjust appropriately to ensure that the lens antennas run smoothly across the length of the PPWG.

5.2 How to set up your product

The setup of the system requires uploading our scanning routine attached in the appendix at the end to the MCU. This can be done by connecting the FTDI programmer to the header pins to the left of the ESP32 on the custom PCB board. Ensure that you are only using the VDD from this header or 12 V barrel connector. DO NOT connect both at the same time.

5.3 How the user can tell if the product is working

Once programmed, switch VDD sources if powering with the header pin. Once the 12 V is applied, the motor scanning routine should begin and start recording values. The lens antenna should settle at the maximum RF signal input value which can be confirmed either with a multimeter placed in parallel before the op-amp or by checking the serial monitor that provides the live value that the lens antenna is receiving.

5.4 How the user can troubleshoot the product

A few common issues include:

- The lens antenna is settling in the wrong spot
- The routine is not running and the scan never starts
- Two peak DC output voltages at different locations

For the first issue, the solution is often simple as moving the lens antenna .5 mm in either direction. If the track of the lens antenna is not perfectly smooth, the little bumps and additional friction can lead to a slight error in the scanning routine and subsequent incorrect position. This can quickly be solved by making sure that the track is running smoothly and that the lens antenna is not moving the PPWG in a way that can cause this discrepancy.

For the second issue, ensure that the programming pins are plugged in correctly to the header pins on the board. Remember that the Tx on the header pin should be connected to the Rx of the FTDI programmer and vice versa. If the more complicated routine does not upload, try uploading the simpler LED routine in the appendix to make sure the ESP32 is functioning correctly. Another common solution is to ensure that the board is being powered with the 12 V and not the 5V from the programmer. This will prevent the motor driver from engaging and running.

For the third issue, this is often due to the collimated RF signal not being fully focused at the lens antenna. Oftentimes, the rectangular waveguide antenna has fallen slightly out of the holder and leaves a gap between its end and the edge of the copper-lined waveguide. This gap between the rectangular waveguide and the PPWG causes multiple symmetric peaks in power along the beamscan as the rectangular waveguide is not oriented as designed. The cause of the multiple peaks is that the device is reading power below the focal point that the GRIN lenses designed for. Thus, it is critical to make sure that the lens antenna is pushed in all the way.

6 To-Market Design Changes

The primary design change that we would like to improve is the form factor of the 5G repeater. Our proposed solution was to keep the product compact and light so the system would not hinder the user's view. However, one of the main challenges we faced during the project was mounting the linear rail to the PPWG. Our initial design used zip ties to fasten the two subsystems together, but it was difficult to work to keep the system flush against the wall while remaining level. As a result, we had to use several zip ties and make small adjustments to each point of contact to keep the entire system stable while the feed antennas shifted along the PPWG. Due to extended lead times on 3D printing, we were unable to make modifications to our final 3D prints. In a future iteration, we suggest a more compact and rigorous design for attaching the linear rail to the PPWG.

Another associated problem with the project is the cost of the components needed to produce this project. While the senior design budget covered the linear rail, board design, and 3D printed parts, the RF components were sourced from Notre Dame's mmWave lab. In order to take this project to market, the RF components could be designed into a PCB that would remove the casing and shields but were beyond the scope of this course. Adding the RF components to a circuit board would remove the hanging wires and reduce the cost of the system.

Interfacing the RF components with the DC circuit boards could also be improved. The esp32 ADC has noted issues with its linearity and ability to detect voltages below .1 volts. There exist I2C addressable ADCs that would allow for much more accurate power sensing from the power sensor that is chosen for the next iteration of the project.

In the future, the PCB and CAD designs will be updated to improve the aesthetics of the 5G repeater. During our demo, we had to power the RF components from the power supply due to the current limitation on our board. To resolve this issue, we recommend changing the power supply to a USB-C connection and correcting the op-amp configuration by ordering a part with the same pin connections. The circuit board to house the RF components and linear actuator circuitry could be placed inside a 3D printed case to provide a modular system.

7 Future Work

Currently, this product can only be used for detecting plane waves incident upon a window and can only collimate waves in a vertical direction. Creating a full circular lens would allow for more directions where waves can be scanned from. Additionally, there is work being done to find an inexpensive solution for producing GRIN lenses. Finally, this product could eventually be used to allow 5G signals to enter homes from low earth orbit satellites. Additional work is needed to resolve the coupling issues between the receive and transmit antennas on the indoor side. Ideas have been floated to either take an amplifier out of one of the chains or build a physical signal isolator between the two chains.

8 Conclusions

Our device successfully demonstrates proof of concept of using GRIN lenses to extend the range of mmWave through Low-E glass to amplify services into indoor spaces. The team successfully met the requirements with key subsystems interacting to accomplish a 5G repeater. Although we faced challenges with the initial design, we were able to produce a second prototype board that addresses the ADC and unity gain operational amplifier issues that could be used in a future iteration. With the addition of a more compact mounting solution, a future version could improve the potential product. Overall, the system showcased our understanding of advanced electrical engineering topics to create a 5G repeater.

9 Appendices

9.1 Datasheets:

Directional Coupler:
https://www.pasternack.com/directional-20-db-2.92mm-coupler-40-ghz-pe2cp1126-20-p.aspx

Amplifiers:
https://www.micro-semiconductor.com/products/ADI(Analog-Devices,Inc.)/HMC1040LP3CE

Power Sensor:
https://www.analog.com/en/products/ltc5596.html

Motor Driver:
https://www.pololu.com/product/1182/resources

Motor:
https://amzn.to/3rYqCpo

ESP32:
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Individually Addressable LEDs
LED strip:
https://www.adafruit.com/product/1426?gclid=CjwKCAjwuYWSBhByEiwAKd_n_tGvcTTSvYP3CFECqlDVmo7-EKkKovWqzDdZs1iba5vQ1bopA_BWEBoCDmMQAvD_BwE
datasheet:
https://cdn-shop.adafruit.com/datasheets/WS2812.pdf

9.2 *Software Listings*

Full Scanning Routine

```
#include <FastLED.h>

#define LED_PIN     26
#define LED_TYPE    WS2812
#define NUM_LEDS    8
#define BRIGHTNESS  50

#define adcPin      32

#define SW_CLOSED   27

#define dirPin      4
#define stepPin     16
#define enable      23
#define reset       5
#define sleep       17
#define ms1         21
#define ms2         19
#define ms3         18
```

```
CRGB leds[NUM_LEDS];

double vout = 1;
const int stepsPerRevolution = 786;

void setup() {
  Serial.begin(115200);
  pinMode(adcPin, INPUT);
  analogSetAttenuation(ADC_2_5db); //reads value from 100
mV to 1250 mV


  //feed translation pin declaration
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(enable, OUTPUT);
  pinMode(reset, OUTPUT);
  pinMode(sleep, OUTPUT);
  pinMode(ms1, OUTPUT);
  pinMode(ms2, OUTPUT);
  pinMode(ms3, OUTPUT);

  //LED Declaration
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
  FastLED.clear(true);
  FastLED.setBrightness(BRIGHTNESS);

  // Limit Switch Declaration
  pinMode(SW_CLOSED,INPUT_PULLUP);

  test_motor();
}

void loop() {
  vout = analogRead(adcPin)*1.25/(4095);
  Serial.println(vout);
  LED_display(vout);
  delay(2000);
}

//FEED TRANSLATION CODE
```

```
int test_motor()
{
  float newADC = 0;
  float lastADC = 0;
  float highADC = 0;
  int newstep;

  //reset pin must be high
  digitalWrite(reset, HIGH);
  digitalWrite(sleep, HIGH);
  digitalWrite(ms1, LOW);
  digitalWrite(ms2, LOW);
  digitalWrite(ms3, LOW);

  // turn enable on
  digitalWrite(enable, LOW);

  // Set motor direction clockwise
  digitalWrite(dirPin, HIGH);

  for(int x = 0; x < stepsPerRevolution; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(6000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(6000);

    //get new power reading
    newADC = analogRead(adcPin)*1.25/4095;
    //Serial.println(newADC);
    if (newADC > highADC && (newADC - lastADC) < 0.3){
      highADC = newADC;
      LED_display(newADC);
      newstep = x;
      Serial.print("New High Value = ");
      Serial.println(highADC);
    }
    lastADC = newADC;
    if (digitalRead(SW_CLOSED) == 0){ //switch is closed
      Serial.println("SWITCH CLOSED");
      Serial.println(x);
      break;
    }
  }
  delay(1000); // Wait a second

  // Set motor direction counterclockwise
  digitalWrite(dirPin, LOW);

  for(int x = 0; x < stepsPerRevolution; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(6000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(6000);

    //get new power reading
    newADC = analogRead(adcPin)*1.25/4095;
    //Serial.println(newADC);
    if (newADC > highADC && (newADC - lastADC) < 0.3){
      highADC = newADC;
      LED_display(newADC);
      newstep = x;
      Serial.print("New High Value = ");
      Serial.println(highADC);
    }
  }
  delay(1000); // Wait a second

  // Set motor direction clockwise again
  digitalWrite(dirPin, HIGH);

  for(int x = 0; x < newstep; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(6000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(6000);
  }

  // turn enable off
  digitalWrite(enable, HIGH);
```

```
  return newstep;
}

void LED_display(double vout){
  int power_input = round((vout/1.25)*8);

  if (power_input >= 1){
    leds[0] = CRGB(255, 0, 0);
    FastLED.show();
  } else{
     leds[0] = CRGB(0, 0, 0);
     FastLED.show();
  }

  if (power_input >= 2){
     leds[1] = CRGB(255, 64, 0);
     FastLED.show();
    } else{
     leds[1] = CRGB(0, 0, 0);
     FastLED.show();
  }

  if (power_input >= 3){
      leds[2] = CRGB(255, 128, 0);
      FastLED.show();
  }else{
     leds[2] = CRGB(0, 0, 0);
     FastLED.show();
  }

  if (power_input >= 4){
     leds[3] = CRGB(255, 192, 0);
     FastLED.show();
  }else{
     leds[3] = CRGB(0, 0, 0);
     FastLED.show();
  }

  if (power_input >= 5){
     leds[4] = CRGB(255, 255, 0);
     FastLED.show();
```

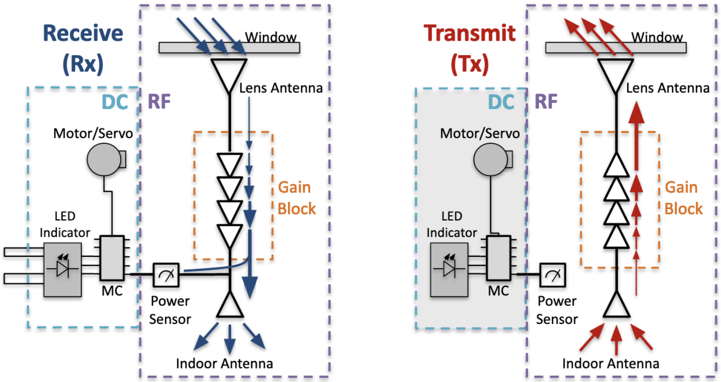
```
        }else{
           leds[4] = CRGB(0, 0, 0);
           FastLED.show();
        }

        if (power_input >= 6){
           leds[5] = CRGB(170, 255, 0);
           FastLED.show();
        }else{
           leds[5] = CRGB(0, 0, 0);
           FastLED.show();
        }

        if (power_input >= 7){
           leds[6] = CRGB(85, 255, 0);
           FastLED.show();
        }else{
           leds[6] = CRGB(0, 0, 0);
           FastLED.show();
        }

        if (power_input >= 8){
           leds[7] = CRGB(0, 255, 0);
           FastLED.show();
        }else{
           leds[7] = CRGB(0, 0, 0);
           FastLED.show();
        }
      }
```

LED Only Debugging Code

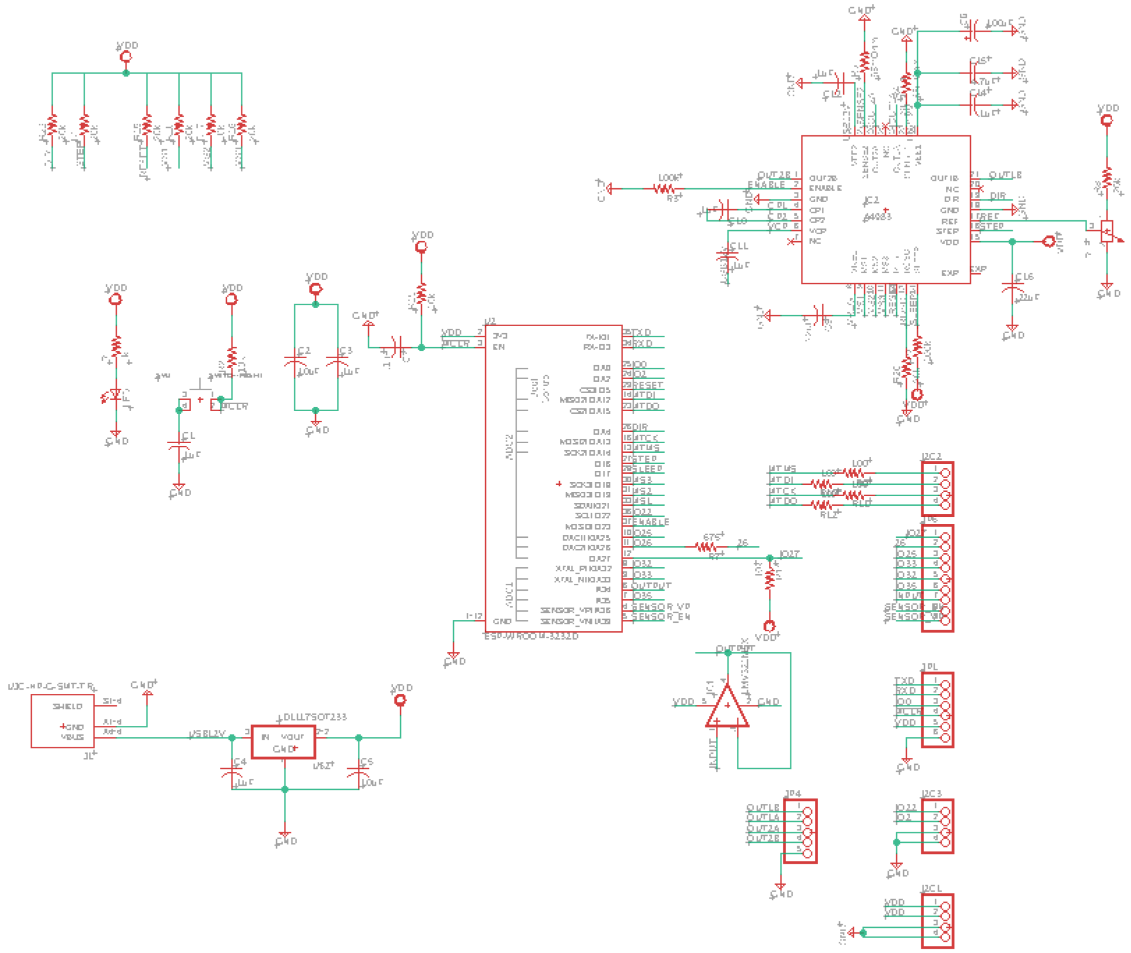
```
const int adcPin = 26;
int voltage = 0;
double vout = 1;
//ADC_ATTEN_DB_2_5

void setup() {
  Serial.begin(115200);
  analogSetAttenuation(ADC_2_5db);
  delay(1000);
}

void loop() {
  //voltage = voltage + analogRead(adcPin);
  vout = analogRead(adcPin)*1.25/(4095);
  Serial.println(vout);
  delay(2000);
}
```

9.3 *Hardware Schematics*

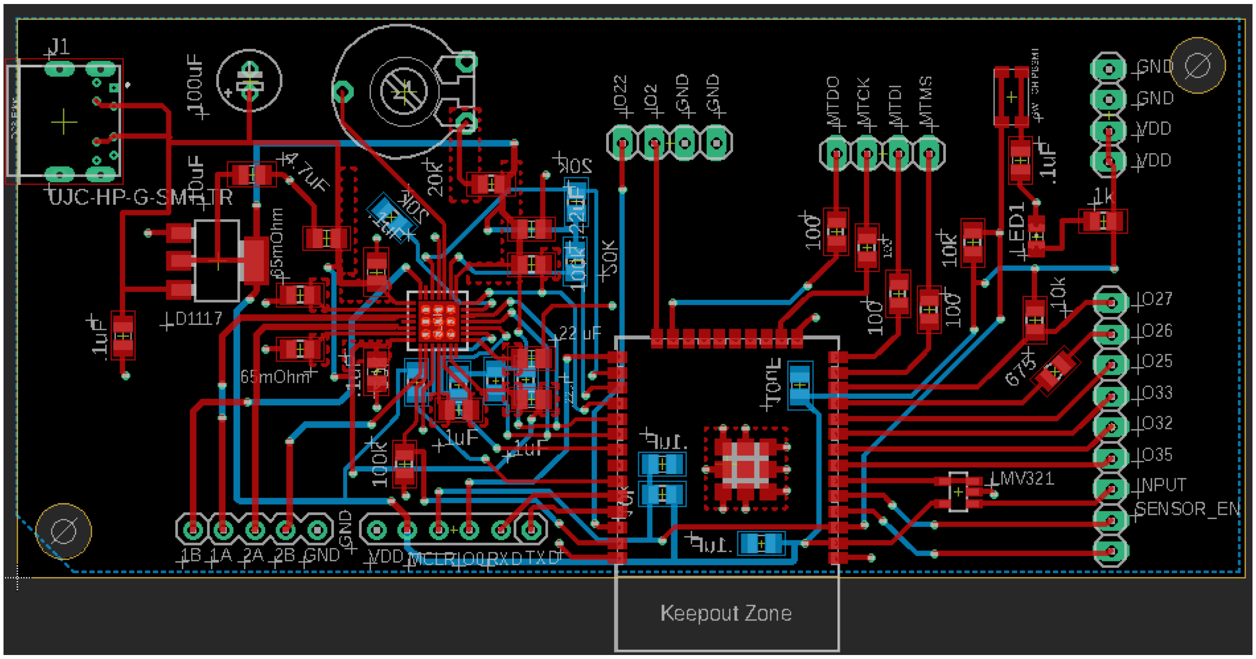
RF components with link to DC board.



DC board
Revised Board Schematic



Revised Board Layout


J1
100uF
UJC-HP-G-SMT-TR
10uF
4.7uF
65mOhm
LD1117
1uF
65mOhm
20K
30K
20K
22uF
100k
30K
22 uF
1uF
100k
1uF
1uF
10uF
.1uF
.1uF
10K
IO22
IO2
GND
GND
MTDO
MTCK
MTDI
MTMS
100
100
100
100
10k
LED1
.1uF
1k
10k
675
GND
GND
VDD
VDD
IO27
IO26
IO25
IO33
IO32
IO35
INPUT
SENSOR_EN
LMV321
1B
1A
2A
2B
GND
GND
VDD
MCLR
IO0
RXD
TXD
Keepout Zone