ECE:4880 Lab 2 - Electric Eye
Four Engineers Walk into A Bar
Tony Longo, Emma Taylor, Jacob Sindt, Ryan Griffin

Lab Report 2

1. Design Documentation

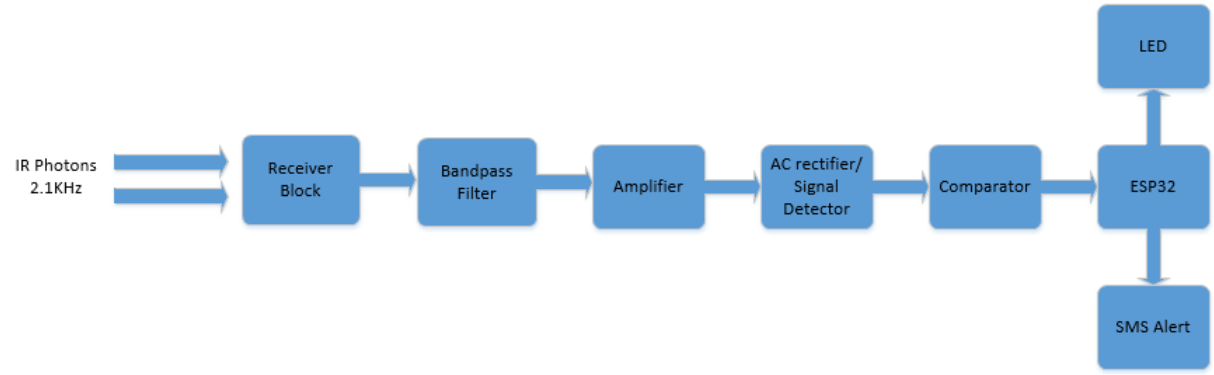


Figure 1: Flow chart of the receiver system showing how the signal is processed and how the SMS messages and LED are controlled.

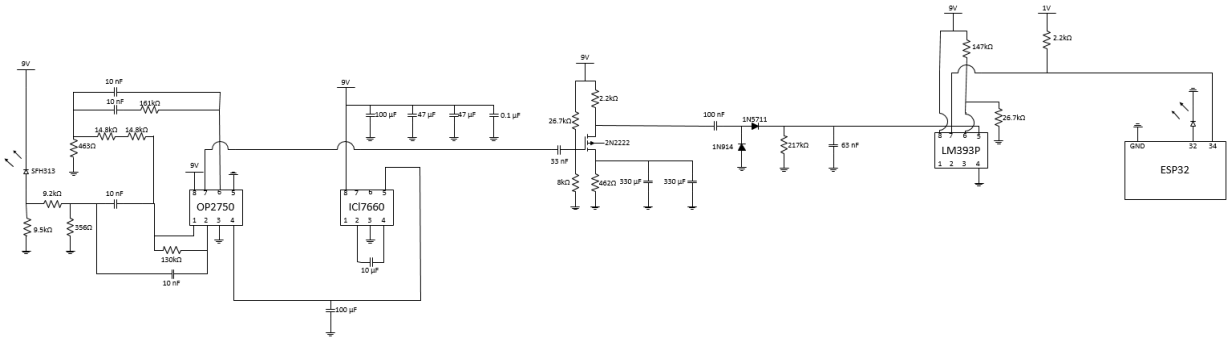


Figure 2: Circuit schematic of the receiver system and the overall signal processing.

The system block diagram in figure 1 and the circuit schematic in figure 2 work as follows: The receiver block consisting of the phototransistor detects the 2.1KHz IR signal being transmitted from the transmitter and creates a

corresponding voltage waveform. This waveform is filtered through an active bandpass filter constructed from OP-amps and discrete components that passes only the detected signal and removes any unwanted low and high frequency noise. This filtered signal is then amplified by a BJT amplifier and is rectified into a constant DC waveform through a diode signal detector. Once this DC signal is created it is sent through a voltage comparator that outputs either a high voltage from a DC signal input or a low voltage from a near zero DC signal input. These outputs are then fed into the ESP32 for analysis and an SMS text message along with an LED turning on is performed accordingly.

ITEM NO	QTY	DESCRIPTION
1	1	Phototransistor used to receive the IR radiation
2	1	9.5 kOhm resistor
3	1	463 Ohm resistor
4	2	14.8 kOhm resistor
5	1	161 kOhm resistor
6	1	9.2 kOhm resistor
7	1	356 Ohm resistor
8	1	130 kOhm resistor
9	1	8 kOhm resistor
10	2	26.7 kOhm resistor
11	2	2.2 kOhm resistor
12	1	462 Ohm resistor
13	1	217 kOhm resistor
14	1	147 kOhm resistor
15	3	10 nF ceramic capacitor
16	2	100 uF electrolytic capacitor
17	1	10 uF electrolytic capacitor
18	2	47 uF electrolytic capacitor
19	2	100 nF ceramic capacitor

20	1	33 nF ceramic capacitor
21	1	63 nF ceramic capacitor
22	2	330 uF electrolytic capacitor
23	1	1N914 Rectifying diode
24	1	1N5711 Schottky diode
25	1	2N2222 BJT Transistor
26	1	LED any color
27	1	OP2750 8-DIP operational amplifier IC
28	1	ICL7660 switching capacitor voltage inverter 8-DIP IC
29	1	ESP32 microcontroller
30	1	9V battery
31	2	8x5.5 cm Breadboard
32	AR	26 AWG breadboard wire
33	1	1V power supply
34	1	LM292P Voltage comparator 8-DIP IC

Figure 3: Bill of materials (BOM) for the receiver system. Note: AR stands for “as required”.

1.1. Filter Simulation and Design

When designing our filter, we first had to pick which kind of filter we wanted. This decision was based on what frequencies we would like to pass through and which frequencies we would want to attenuate. Our prototype is expected to be able to perform well next to a lamp as well as outside. The noise created from the lamp, as well as any light source in the lab, is at a frequency of 120 Hz. The noise created from light outside is at all frequencies. Since the prototype is expected to perform well in both conditions, we finalized our filter to be a bandpass filter that is centered around our transmitting frequency.

After deciding on which type of filter we wanted, we identified the frequency of our transmitted signal. The schematic for the transmitter was given, but due to tolerances for component values, we decided on measuring the frequency of the transmitted signal

instead of building and simulating the transmitter schematic. For this measurement, we hooked up the transmitter built in the lab to a voltage source and measured the output of the transmitter using an oscilloscope. Using this setup, we measured a transmitting signal of 2.1 kHz. A picture of this measurement is pictured below in figure 4. This measured frequency was chosen to be the center frequency of our designed bandpass filter.

Figure 4: Oscilloscope reading for the transmitter signal. Focusing on the frequency of the signal, we see that it is roughly 2.1 kHz.

After finalizing the type and center frequency of our filter, we began to design the topology for our filter. Our decision for the topology of our filter we used the “Analog Filter Wizard” resource given to us. The filter designer website is a great resource that finds R and C component values for a filter when given the desired parameters for the filter. This website also provides a plot for gain magnitude vs. frequency for each filter, which helped provide insight on what order filter was needed and what our passband bandwidth should be.

For our bandpass filter, we finalized our design to operate using the OP275 op-amp, have a passband bandwidth of 800 Hz, and that the filter would be a 4^{th} order 3-dB Chebyshev. We decided on using the OP275 op-amp, because it was both able to be

simulated on the “Analog Filter Wizard” resource and that it was an available IC in the engineering electronics shop. The OP275 IC, is an 8-pin integrated circuit that has two op-amps built into it. Since there are two op-amps built into the OP275, we were able to make a 4^{th} order filter while also saving space by utilizing all the op-amps in our IC. Another good reason to use an op-amp to filter is because it provides very good isolation between the photo transistor and the rest of the circuitry. This is good because we don’t have to design a separate buffer system.

For finding the passband bandwidth, and the order of the frequency, we used the gain magnitude vs frequency plot provided on the “Analog Filter Wizard” resource. The passband bandwidth is an input parameter for the simulation tool, so because of this we swept through values for the passband bandwidth. While sweeping through values, we focused on how these values affected our filters passband and stopband behavior. For our design, we want our passband to be narrow and for our stopband to have high attenuation over a large variety of frequencies. We also want to have a significant amount of attenuation at 120 Hz, since the prototype is supposed to be able to perform with a strong light source nearby. Since we were unaware of the intensity of the light source noise, we first designed our filter to have at least 50 dB of attenuation at 120 Hz. This value of 50 dB was completely arbitrary. We didn’t know if that was too little or too much attenuation, but it was a good start. After building the filter, tests were conducted to verify the performance of the filter. Documentation for the tests performed are described in a later section. After selecting the cutoff frequency and the amount of attenuation we wanted to design for, we ran the simulation on the “Analog Filter Wizard" resource. With the filter parameter values we chose we found that our design could be attained by a 4^{th} order filter.

After finding the passband bandwidth and the order of the filter, we needed to design the bandpass filter type. When designing the bandpass filter type, we wanted to mainly focus on the goal of our filter. Our filter is supposed to attenuate all unwanted frequencies and to pass the transmitter signal that is at 2.1 kHz. Our filter is not necessarily required to have a clean complete copy of the signal on the output of the filter. We are only required to detect that a signal has passed through. Using this information, we decided on using a 3dB ripple Chebyshev filter. Chebyshev filters have very steep attenuation close to the passband providing for more attenuation at closer frequencies.

Using all the findings above, we simulated our design on the filter wizard resource to find the RC component values we need to build our designed filter. The simulation provided us with ideal component values as well as E-12 and E-24 valued equivalent schematics. For our design we were constrained to E-24 values for resistors, so we focused on the E-24 equivalent filter, but pictures of our ideal schematic and gain

magnitude vs. frequency plot are pictured below in figures 5 and 6 respectively.. Since the values for each resistor and capacitor depend on each other, we decided to pick all our capacitor values first since we had less variety for capacitors than we do with resistors. The filter resource also provided E-24 equivalent gain magnitude plots which were the driving factor in our decision for our capacitor values. We picked our capacitor values so that we could have the lowest amount of attenuation at our operating frequency of 2.1 kHz. After sweeping through capacitor values, we found that our filter operated best when all the capacitors were set to 10 nF. With these values, the filter resource proposed that we would even have gain at our operating frequency of about 1.6 dB. Using the filter designer resource, we set all the capacitor values to 10 nF and read off each of the recommended E-24 resistor values to finalize our circuit. The final schematic for our filter on the “Analog Filter Wizard" resource can be pictured below in figure 7. The gain magnitude vs. frequency plot using E24 values is pictured below in figure 8. This plot also accounts for the tolerances of each of our components.

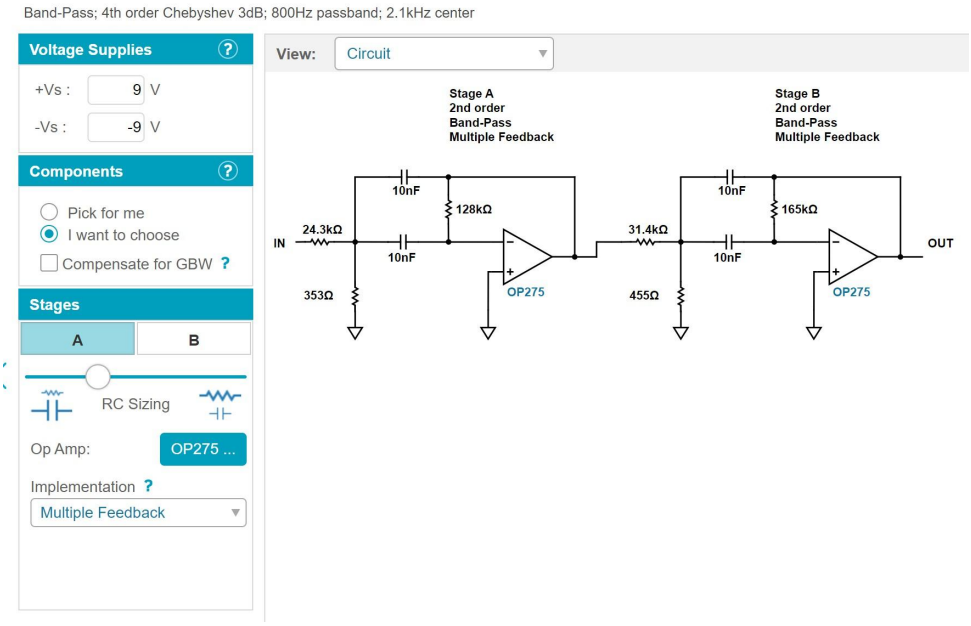


Figure 5: Schematic for the filter we designed using “Analog Filter Wizard”. This schematic is the final design using ideal component values.

Figure 6: Gain magnitude vs frequency plot for our final designed filter. The marker shows the magnitude of the attenuation at the light frequency of 120 Hz. This plot only represents the behavior of the filter using ideal values

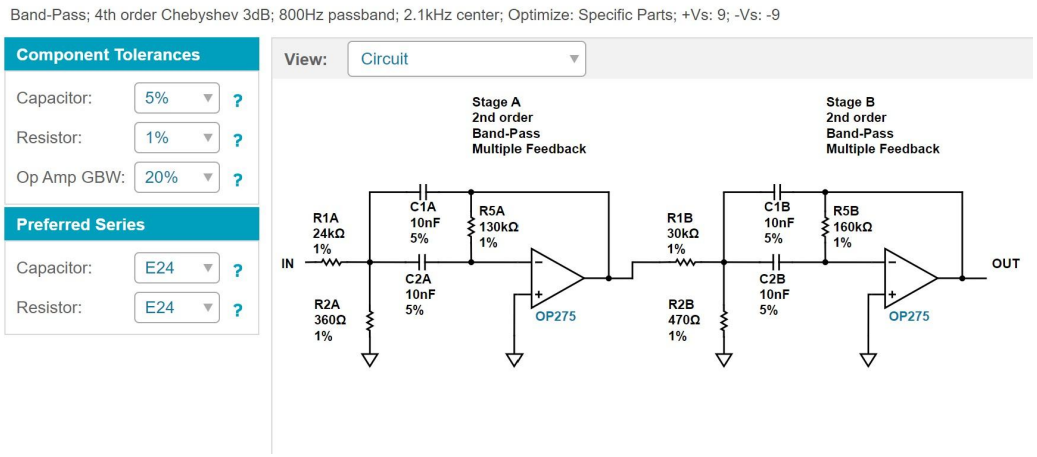


Figure 7: Schematic for the filter we designed using “Analog Filter Wizard”. This schematic is the final design using E24 values.

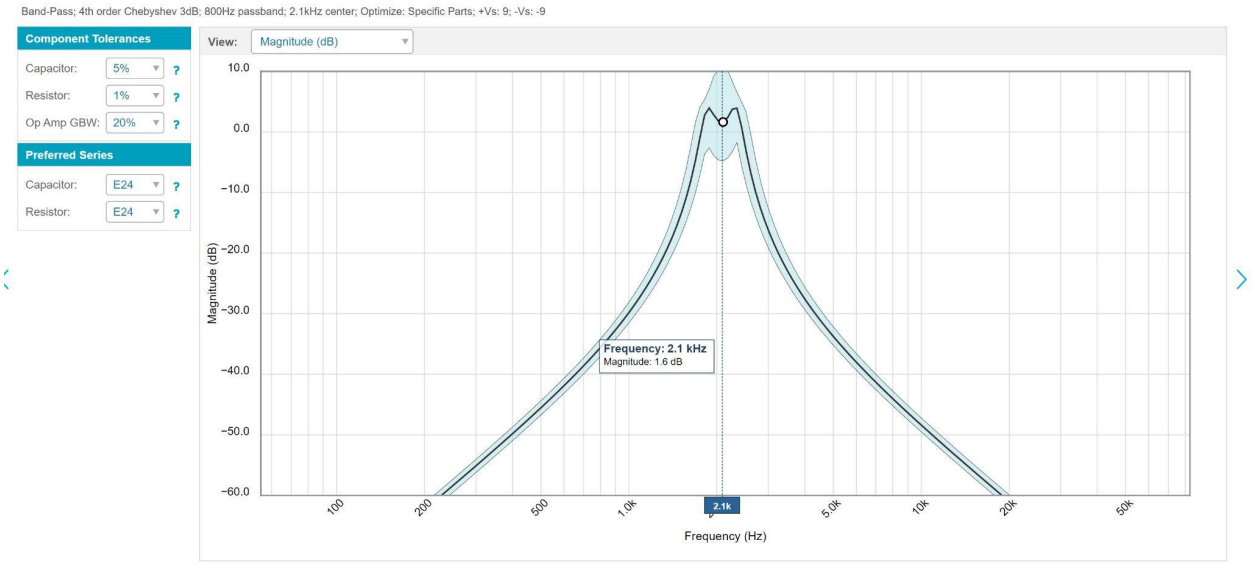


Figure 8: Gain magnitude vs frequency plot for our final designed filter. The marker shows the magnitude of the gain at the operating frequency of 2.1 kHz. This plot accounts for E24 values and the tolerances of each of our components.

1.2. Amplifier Design

After filtering out the unwanted noise, we wanted a way to validate if a signal was being detected or not. For this, we decided that it would be best if we were to amplify our desired signal to a higher voltage. The current created from our phototransistor is in the same scale as a few micro amps. The resistor in series with the phototransistor converts that current to a voltage signal in the same scale as a few mV's. A few mV's is not capable of being detected by our microcontrollers digital I/O pins. Signals with that low of magnitude are also hard to drive circuitry. Making our signal have a greater peak to peak voltage makes it easier to detect vs if the peak to peak is very small. So this prompted us to design an amplifier block that should make detecting our signal much easier.

To start our design, we first picked our amplifier stage to be centered around the P2N2222A BJT. This choice was decided based on the ease of finding the component, and that we had lots of experience biasing this exact BJT in the past.

Following that, we wanted to pick our supply voltage. The supply voltage is important for our amplifier stage since it sets the maximum peak to peak voltage we can have without clipping the supply. Clipping is when the gain becomes so high that the signal wants to

reach voltage levels that are higher or lower than our supply rails. We do not want any clipping because we still want our signal to be at our operating frequency of 2.1 kHz. Clipping will distort our signal, which is not good since we designed our detector to work well with a signal at 2.1 kHz. We want to design the amplifier such that the signal can have a high peak to peak voltage so in turn we want to design our supply voltage to be as high as we can. The absolute maximum supply voltage for the ICL7660, the switched capacitor inverter that powers up our op-amps in the filter block, is +10.5V. This prompted us to choose a supply voltage of 9V. This is a high value that still operates at a safe margin away from the maximum voltage rating. Using the value of 9V, ideally we want our signal to be centered at 4.5V and to have a peak to peak voltage of 9V in our amplifier block. This however might be an overcompensation so we initially designed our amplifier to be centered near 4.5V and to have a relatively high value of gain that doesn't clip our supply rail.

After picking our supply voltage, we wanted to design the lumped elements for our amplifier. For this we wanted to first design the key parameters for the amplifier. Again, we want the voltage at the collector of the BJT to be roughly half of the supply voltage, so in our case that would be Vc = 4.5 V. We also wanted to pick a reasonable value for our collector current. Looking at the BJT data sheet, attached below in section 5.4, figure 3 shows the DC current gain vs collector current. We decided on picking an initial collector current value of 3 mA. This would result in a current gain magnitude slightly less than 200. After this, we also wanted to identify the desired emitter voltage. We could ground the emitter of the BJT so that we can maximize our gain, but having an emitter resistor provides negative feedback that lowers the variability of the functionality of the BJT. We decided that we want the emitter resistor to be roughly 20 percent of the supply voltage. Using Ohm's law we were able to quickly find the emitter and collector resistor values. For both lumped elements, we assumed that the current going through them was both 3 mA. For our model we found collector and emitter resistors to be 1.8 kΩ and 470Ω respectively.

After finding the collector and emitter resistors, it was time to pick out our dc biasing resistors that set the voltage at the base. Assuming the current gain is about 200, this can be assumed using figure 3 on the P2N2222A datasheet linked below, we are quickly able to calculate the base current for our design. We also wanted to design the current going through the resistor between the base and Vcc to be roughly 5 times that of the base current, we also want the voltage at the base to be 20 percent of the supply voltage plus 0.7 V. This is because we design the emitter to have a DC voltage equal to 20 percent of the supply voltage and we assume there is a Vbe of 0.7 V. Knowing the supply voltage, the base voltage, and the current through this resistor, we were able to use ohm's law to calculate a resistance of about 23 kΩ. Finally, we found the resistor

between the base and ground nodes to be 8.2 kΩ. We did this using ohm's law since we knew that the current through this resistor is roughly 4 times that of the base current and that we knew what the base DC voltage was.

After finalizing our lumped elements we had one more key component to add to the amplifier. To increase the signal to noise ratio, we wanted to focus on amplifying only the signal at our operating frequency. The noise source we are most concerned with in this lab is the light source at 120 Hz. For this we decided to add a capacitor in parallel to the emitter resistor. This is because having an emitter resistor decreases the gain of the amplifier. We wanted to make the emitter impedance equal to the resistance at DC, so that the gain at lower frequencies is small, and then we wanted the emitter impedance to be very small, so that the gain at higher frequencies is very large. When designing this capacitor, we wanted the equivalent impedance at our design frequency to be essentially a short. We picked a value of about 0.1Ω. Using the equivalent impedance equation for the capacitor at our design frequency, we found that the ideal capacitance should be roughly 758 uF. For our actual design, we decided to put two 330 uF capacitors in parallel.

1.3. Signal Detector Design

When designing the signal detector we started by listing out our criteria. We have an AC voltage signal riding on a DC voltage of 4.5 Volts on the input, and we want a DC voltage proportional to the AC voltage amplitude on the output. For our actual design we wouldn’t get a perfect DC voltage at the output of the detector, but we would get a slight ripple voltage with the DC value. For this, we decided on using a simple peak detector. At the input we placed a 100 nF capacitor to block out the DC component of the signal, following that point, we added a diode pointing from ground to the node after the DC blocking capacitor. This diode's purpose is to help clamp the voltage to be positive in value. Next from the DC blocking capacitor, we wanted to place a schottky diode, 1n57711, facing towards the output so that we can eliminate the negative voltages from the signal. Finally, we placed a resistor and capacitor in parallel with the output node so that we can fine tune our ripple voltage at the output. These two components create a low pass filter so that we can ideally get just a DC signal at the output. The values for these components were found through experimentation. We placed an oscilloscope probe at the output of the detector and changed values for our components until our ripple was lower than 50 mV. We finalized our design to have R and C values of 220 kΩ and 63 nF respectively.

1.4. Comparator Design

For determining a way to reliably send a high or low voltage to the ESP32 without having ripple voltage from the detector circuit, a comparator circuit was implemented. The comparator used in this system was the LM393 which was designed as follows: the input to the non-inverting terminal of the comparator was the AC rectified signal from the output of the signal detector circuit, a voltage divider of around 0.9V was created for the inverting terminal, and the output of the comparator was pulled to a 1V VCC with a 47Ω resistor along with an input to the ESP32 attached.

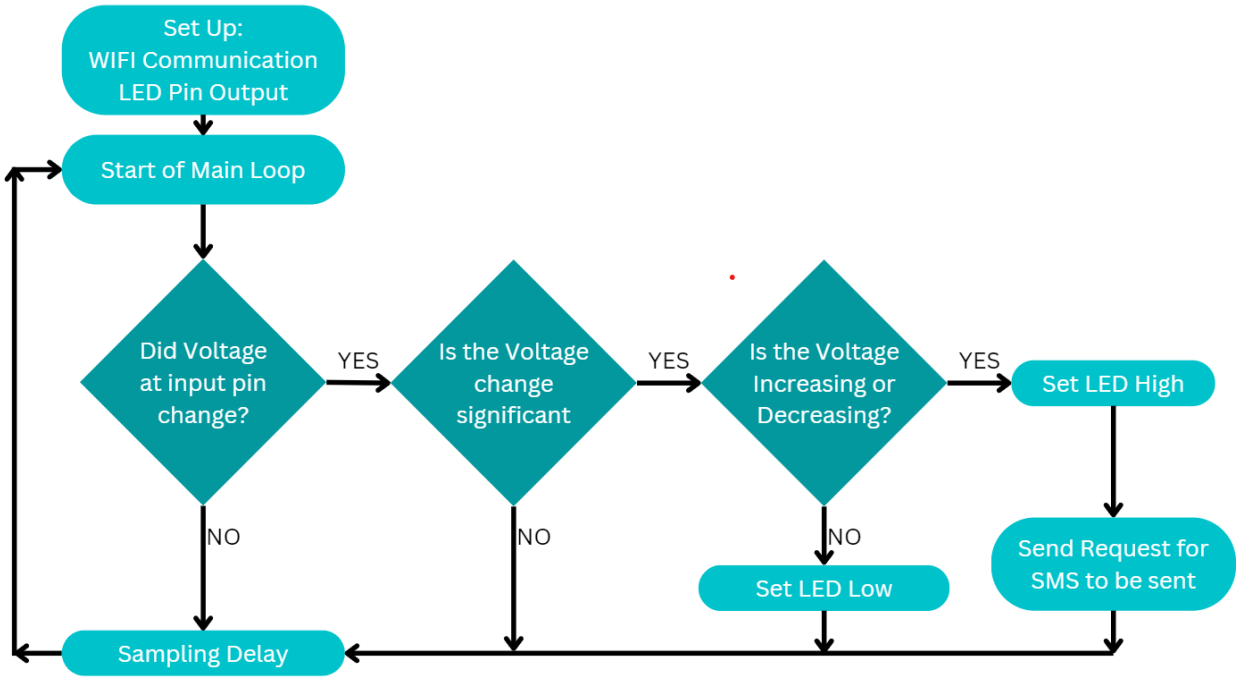
This circuit allowed the use of sending a digital high voltage signal to the ESP32 when the detector circuit was rectifying and a low digital voltage signal when the detector circuit was not rectifying.

1.5. Microcontroller Design

Throughout the development of this lab, we used an ESP32-WROOM-D2 as our microprocessor. This was due to the onboard WiFi capabilities, which we utilized to send SMS. Output GPIO pins were utilized for LED signaling. The LED was simply placed between an I/O pin and to ground. We did not need a current limiting resistor since the output voltage of the ESP is not high enough to burn the LED. The code for the ESP32 was written in Arduino IDE.

1.6. Software Design

1.6.1. Flow Chart



1.6.2. Software Design Description

The software component of this lab was run on a single ESP32-WROOM-DA which was running a single .uno file. This uno file on startup initiated a connection to a group members mobile hotspot to enable the microcontroller to have WiFi connectivity. Later this is used to send text message alerts. Our device ran a loop that checked the voltage value at one of the ESP's ADC ports every 500 milliseconds. If a significant voltage increase was detected the software initiates a request for a text message to be sent from the remote IFTTT server and turns on the LED. If a significant voltage decrease was detected the LED is turned off.

2. Experimentation

2.1. Filter testing

For verifying the filters functionality, we simply wanted to send two signals in the stopband, one at a frequency lower than the passband and one at a frequency higher than the passband, and send one signal at our operating frequency. The test setup consisted of connecting a signal generator to the input of the filter and to put a load resistor on the output of the filter. The signal generator is very convenient as we can change frequencies of our signal with ease. We then measured the input signal from the

signal generator and the output signal using an oscilloscope probe. The load resistor we used was a standard 1 k resistor. The value of the resistor did not necessarily matter, we just needed the signal to dissipate the power we were sending through the input side.

When looking at the magnitude vs frequency plot on the “Analog Filter Wizard” resource, we should expect to have 70 dB of attenuation at a frequency of 120 Hz and 34 dB of attenuation at a frequency of 5 kHz. We also should expect to see 1.6 dB of gain at our operating frequency of 2.1 kHz. We didn’t necessarily care about the exact values we were reading off, we just wanted to verify that at 120 Hz and 5 kHz we were not seeing any of the signal at our output. We also wanted to verify that we were getting a copy of our signal at the output of our filter.

Pictured below, in figures 9 through 11, are the three tests we conducted. Going left to right, the first image shows the signal generator. The display on the signal generator shows the frequency of the signal that is being sent through the filter. The second image shows the measurement at the input using an oscilloscope. The third image shows the measurement at the output using an oscilloscope.

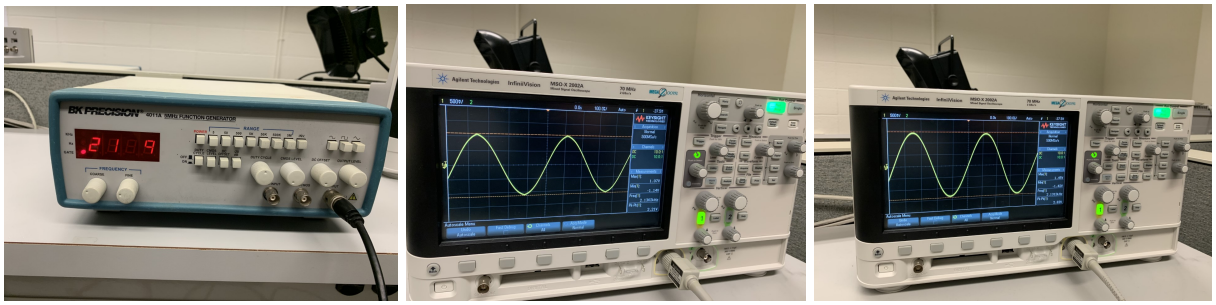


Figure 9: Experiment pictures for when we used a signal at a frequency of 2.1 kHz

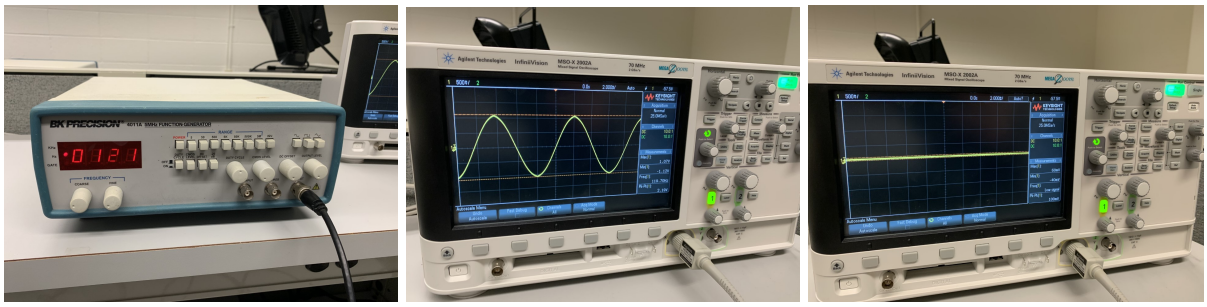


Figure 10: Experiment pictures for when we used a signal at a frequency of 120 Hz

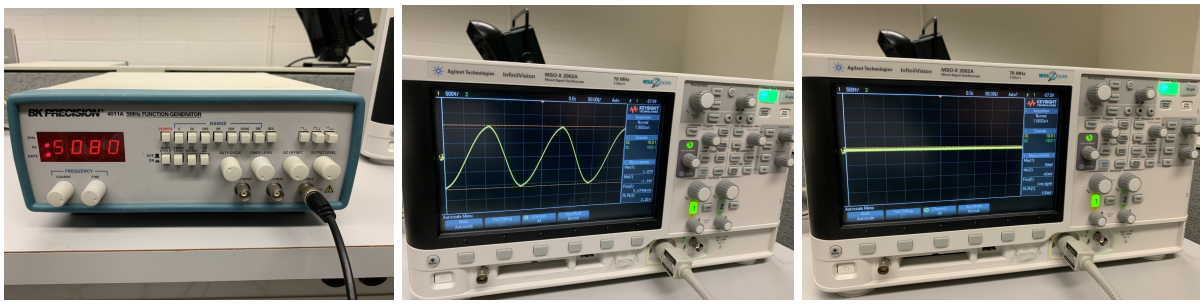


Figure 11: Experiment pictures for when we used a signal at a frequency of 5 kHz

The results from our tests verified the correctness of our design. Looking at the two stop band signals, tests from figure 10 and figure 11, we can see that the signals are properly attenuated. The output signals in both cases are successfully zeroed out at the output of our filter. This is exactly the behavior we had designed our filter for. Looking at the test where we used a signal at our operating frequency, pictured in figure 9, we can see that not only do we properly get a signal at our operating frequency at the output, but we also see that we were able to get some gain. Our input signal had a peak to peak voltage of 2.21 V and our output signal had a peak to peak voltage of 2.83 V. That is roughly 2.15 dB of gain, higher than the simulated gain of about 1.6 dB. Even though this value is relatively small, it gave us more wiggle room when designing the amplifier since we didn't need as much gain as we would've needed if we didn't have this gain or even worse, if we had attenuation at our operating frequency.

After testing our initial filter, we were very pleased with our results. We decided that this filter would likely be used as the filter for our final prototype. We were still uncertain at this point since we had not integrated everything together, but after putting all of the systems together we decided that our final prototype would have our initial designed filter.

2.2. Amplifier Testing

When testing the amplifier we connected the input to the filter and the detector and the output to a resistor that will act like a load. Since we knew that both the filter and phototransistor blocks worked well, we decided to test the amplifier by hooking up all 3. We hooked up the example transmitter in the lab and pointed it towards our receiver. We then measured the input of the gain block and then the output of the gain block. Our goal was to have a high peak to peak at the output of our signal and we measured a peak to peak voltage to be about 8 Volts. This is sufficient for our design and we decided to keep this as our final designed amplifier. It is worth noting that the choice of using a BJT amplifier was compared to other methods of amplification such as OP-Amps. We ended up choosing the BJT amplifier due to the small form factor

involved along with the simplicity in constructing one as opposed to using a larger OP-Amp IC.

3. Test Report

Test	Test Procedure	Expected Outcome	Pass/ Fail	Measured
Receiver must work as well as prototype receiver	Compare receivers false positives and false negatives over a distance with the prototypes false signals at the same distance	Receivers should receive less false signals that the prototype at the same distance	**Pass**	Receiver functions as expected three feet from transmitter
Receiver should work reliably in the presence of a 100 W incandescent light bulb	Compare receivers false positives and false negatives with the presence of a light a varying distance from receiver, also compared the maximum receiving distances for both receivers with a light present	Receiver should function as expected with the presence of light	**Pass**	Receiver function as expected with light 8 inches from the receiver
Sent a SMS every time the transmitted beam is interrupted	Intercept the transmitters signal and record the results	Test message should be sent on interrupt and received in a timely manner	**Pass**	Text was delivered in the correct format momentarily after the signal was interrupted
Demonstrate receiver functionality with prototype transmitter	Use receiver with the prototype transmitter to confirm true positives and true negatives signals	Receiver should filter for true transmitter signals	**Pass**	The receiver filtered signals successfully three feet away from the transmitter
Receiver size should be convenient for testing	Move Receiver between lab benches and reposition it	Receiver should be easy to carry and position	**Pass**	Receiver was handheld and easy to move

4. Project Retrospective

4.1. Project Outcome and Summary

Our project was an overall success. We were able to pass all of the requirements set out for us to begin our lab. Although we may have had a slight delay in our LED turning off once the signal was sensed, we felt that we succeeded in building a high quality IR receiver prototype. We found that our design choices for each section of the receiver system that we made as described above allowed for easy integration with one another. Along with this, each section of the receiver worked successfully to complete its task, allowing the following section of the receiver to in turn work effectively.

4.2. Failures

One improvement we could have made to our design was to use software interrupts to shorten the delay in changes of state for the warning LED. Our LED could stay on for a short period if the text messages were taking longer to send. This could have been shortened by adding an immediate interrupt to that task and completing it after the LED was powered off.

This same issue may have also be improved if we were to build a hardware

4.3. Team Member Roles

Tony Longo

Worked on the software to ensure our ESP32 would recognize when a change in voltage was detected due to the signal being blocked. Coded additional blocks to ensure false positives were excluded. Communicated ESP32 wiring for voltage detection.

Emma Taylor

Wrote software for ESP32 to connect to IFTTT server and transmit text messages when voltage change is recognized. Formatted desired text message format with IFTTT web interface. Wired and programmed LED functionality with ESP32.

Jacob Sindt

Worked on the hardware required to implement the analog filter, amplifier, detector, and comparator circuits. Found solutions to issues in the design and optimization of the comparator circuit. Optimized the integration between the ESP32, comparator circuit and LED.

Ryan Griffin
Worked on designing and verifying the hardware for this lab. Focused mostly on designing the filter using the “Analog Filter Wizard” resource and helping test the filter using both the signal generator and oscilloscope. Designed the switched capacitor inverter to help minimize the cables necessary to power up our circuit.

4.4. Workload Distribution

For this lab, Ryan and Jacob contributed the most towards putting together our prototype. This was because out of the three labs, this was the primarily hardware design project and Ryan and Jacob are both focus areas are on hardware. They have the most experience when dealing with hardware so they tackled designing the hardware for the project. Tony and Emma focused on contributing to the text message as well as the code behind the warning LED. This was because this was the software side of the project and both of them have software focus areas.

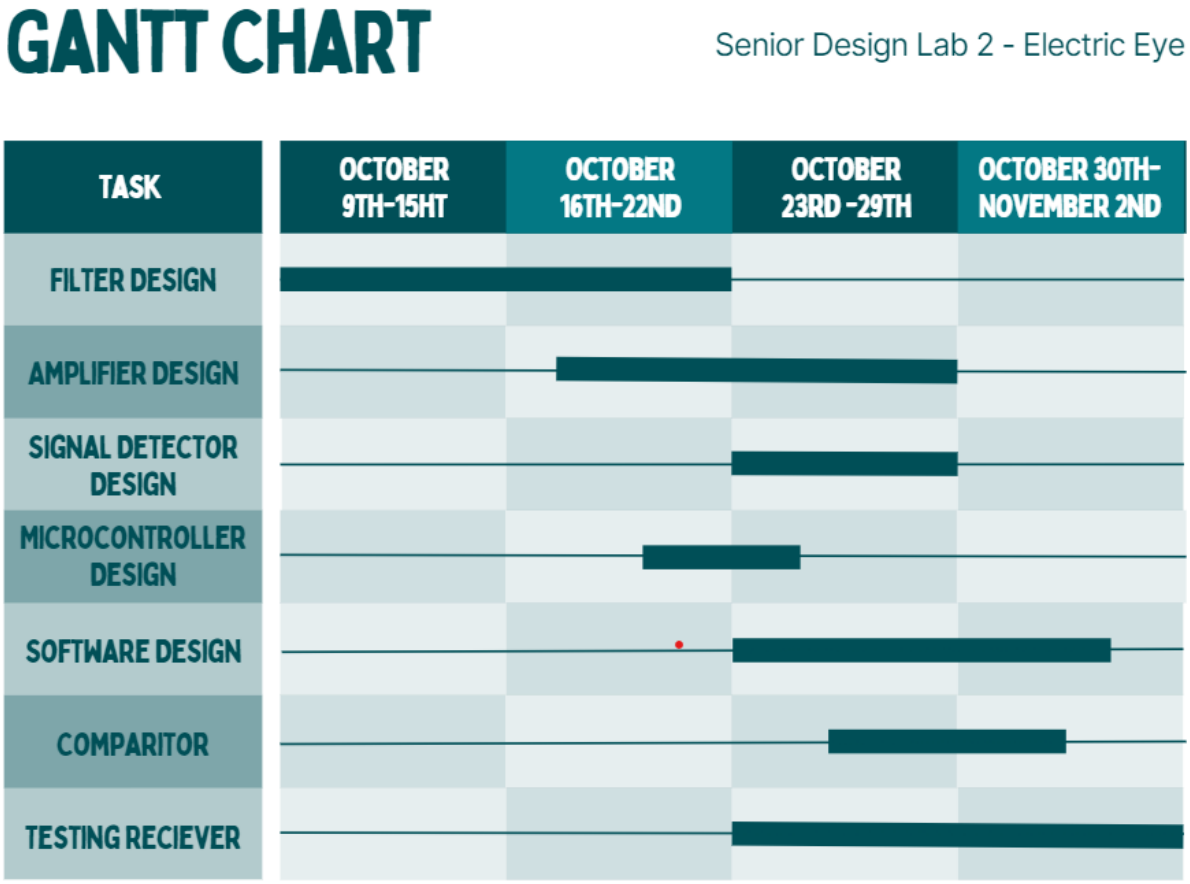
4.5. Project Management Process

Our project involved both waterfall and agile, using waterfall for the hardware design and agile for the software. This helped us quickly accomplish the goals of our lab and maximize the effectiveness of our team members.

Waterfall (Hardware Development):
A waterfall approach takes in all the requirements of the project and designs a roadmap with set goals and plans from the beginning of the project. During this time we make design choices about the type of hardware we are going to use and how we would test it. This lends itself well to the hardware development because we were able to create compartmentalized sections of the circuit to build, test and then combine into a single system.

Agile (Software development):
Our software was developed with an agile approach to allow us to be more flexible to fit the needs of the project. Since our project was largely focused on the hardware components that are slower to change we decided to design the software around its needs. For example, at a point in the project we were unsure if we were going to be able to use the ESP32’s onboard ADcs to read in the voltage values. To solve this issue we moved the software to a focus on a digital I/O pin. We were able to do this because we modularized the parts of the software so that changes to the flow would not be hard to implement.

4.6. Gantt Chart



5. Appendix and References

5.1. SMS Services

IFTTT:

IFTTT is a site that creates applets that allow one action to trigger another. We used this to trigger an SMS text message by sending it a JSON object using POST.

5.2. Libraries

WiFi.h

WiFi is a library contained in the arduino-esp32 library. It is used to connect to local Wi-Fi networks using the standard name and password connection

HTTPClient.h

HTTPClient is a sub-library of arduino-esp32 which can be used to control client behavior that is hosted through an ESP32.

HTTP_Method.h

HTTP_method is a library that helps to identify key terms in the arduino-esp32 library so that its libraries can communicate more easily.

WebServer.h

WebServer is a library contained in the arduino-esp32 library. Used to use functions like server.begin() and initiate HTTP requests

ESPDateTime.h

ESPDateTime was used to determine the timestamp of the signal interruption

Time.h

Time was used to track the interrupts for when the signal was blocked

ESPAsyncWebServer.h

EspAsyncWebServer is a comprehensive library that contains most of the packages needed to host a web server from an ESP32.

Uri.h

Uri is a sublibrary of the ESP package that is used to read requests

SPI.h

SPI allows the user to communicate with SPI devices

AsyncTCP.h

AsyncTCP is a base library for EspAsyncWebServer to allow it to have an easier time connecting to an ESP32's MCUs for a network environment

5.3. Hardware design resources

Analog Filter Wizard

Analog Filter Wizard is a tool to design analog filters given a set of input parameters

5.4. Hardware datasheets

P2N2222A is the NPN BJT used for the amplification of the signal
OP275 is the op-amp used for the analog filter block
ICL7660 is the switched capacitor inverter used to help bias the op-amps used in the filtering block.

LM393P is the comparator used for detecting the signal and down converting the signal to one that the microcontroller could detect.
ESP32 is the microcontroller we used to program the texting and LED.