

ENGPHYS 3BA4
Final Design Project (Week 1-5):

The Ultrasonic Range Finder

Muqtadir Hussain

400250481

Submission Date: 26th November, 2021



Week 1

Introduction:

The objective of this lab is to use an ultrasonic transceiver module to construct a range finder which provides a digital reading of the distance an object is away from the device, when directly pointed at it. This lab makes use of the propagation of high frequency sound waves which are emitted from the transceiver, reflected off a surface and then returned to the receiver. The working range of this device is 10.99 ± 1 cm.

Theory:

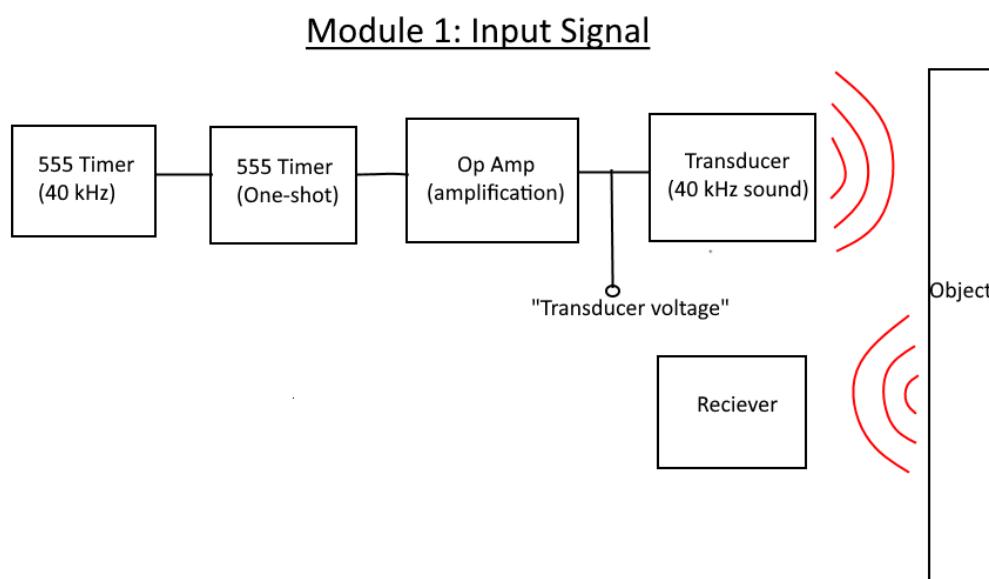
Generally, knowing the frequency of the signal and hence the velocity, one would use the time taken for the wave to return to the sensor to find the distance of the object. In our scenario we want the device to constantly be providing an updated distance displayed on two seven-segment displays. Since we don't know the wavelength of our signal, a different approach was decided for this build. Knowing the speed of sound in air (340 m/s) and by varying our signals frequency to perhaps 34 kHz, we can obtain a wavelength of approximately 1 cm. Knowing the period of this wave and the time taken for the signal to return, the number of full cycles can be determined. Knowing that each cycle has a width of 1 cm, the distance of the sensor from the object can then easily be calculated while keeping in mind that the wave travels twice the actual distance since it hits the object and then has to travel back. In order to build this fully functional circuit especially without a microcontroller, we need multiple different components. The first task would be to produce a 34 kHz signal from the ultrasonic transmitter. In order to do so, a CMOS 555 timer will be used to convert the DC signal into an AC square wave. Using the CMOS 555 timer in astable mode and varying the resistance and capacitance, a 34kHz wave with a 50% duty cycle can be produced. In order to detect the returning signal, we can use a D type flip flop that triggers on the first rising edge of the received signal. The flip flop would output a 1 if the receiver signal and clock is high and a 0 if the receiver signal is low but the clock is high. However, when the clock is low, it would just output memory or in other words the previous value. An integrated counter would then store the number of times the D type flip flop was triggered. Furthermore, to remove interference of input and output signal while also obtaining periodic measurements of distance, another CMOS 555 timer can be implemented to add a delay in our cycle. The ratio can be such that ON period is much shorter than the OFF period and therefore the output pulses from the

transmitter would not interfere with our returning signal. Another important thing to keep in mind is the attenuation of the returning signal along with digital noise in the environment which can make it difficult for signal detection upon return. In order to overcome this, we can use a high gain low noise amplifier such as a ‘two stage inverted band-pass amplifier’ to boost the returning signal. Finally, we want to display the periodically measured distances in cm on the actual seven-segment displays. Two displays will be used with one representing the 1’s position and the other representing the 10’s position. Using the counter previously mentioned, the display representing the ones position will constantly be updating and eventually resetting upon reaching 10 cycles. However since the signal has to bounce from the object and then return to the receiver, this may need to be changed to 20 cycles since the actual distance travelled by the signal is doubled. Every time the 1’s display reaches 10 cm it will reset to 0 and the 10’s display will be incremented up. IC Decoder chips will be needed with the seven-segment displays.

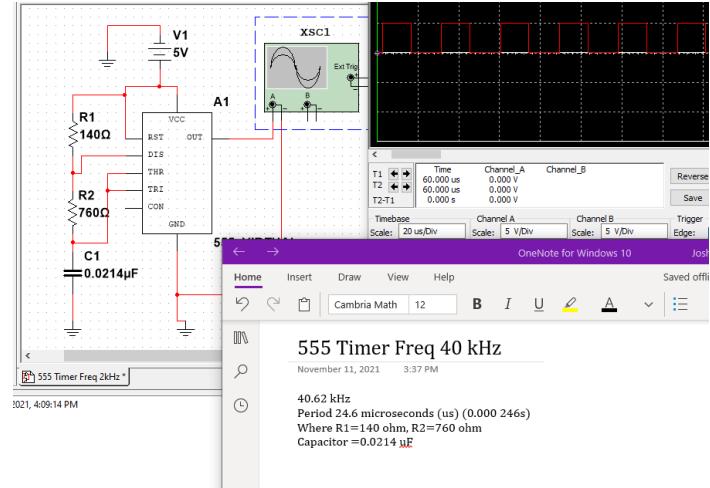
Week 2

The design for the ultrasonic range finder was split up into four modules for easier comprehension and building. Along with making the initial building process more efficient by splitting up the workload, modularity allows for better detection of the error source if the final outcome isn’t as expected as the outputs of each module can be individually observed to find out where the error is arising from. The design is as follows:

Module 1:



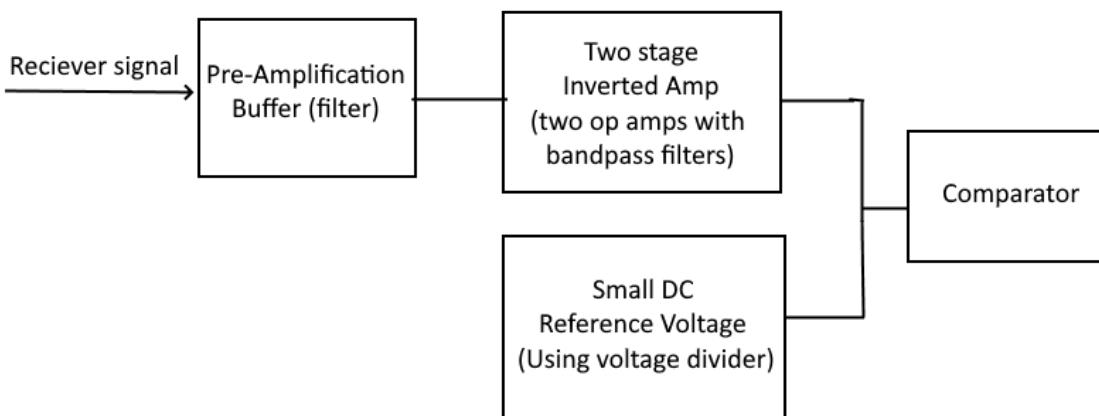
The first module consists of two 555 timers, an operational amplifier and the actual transceiver. The first 555 timer produces the required 40 kHz signal for the ultrasonic wave.



This continuous square wave then passes through another 555 timer which is built to be a monostable multivibrator (one shot). This creates periodic pulses of the 40Khz signal instead of a continuous output to avoid interference between outgoing and incoming waves after reflection. Before sending this wave to the input of the ultrasonic transducer, it is passed through an op amp to amplify the voltage of the signal. The resulting signal is then passed through the transducer to produce periodic bursts of acoustic waves (sin waves) with a frequency of 40Khz.

Module 2:

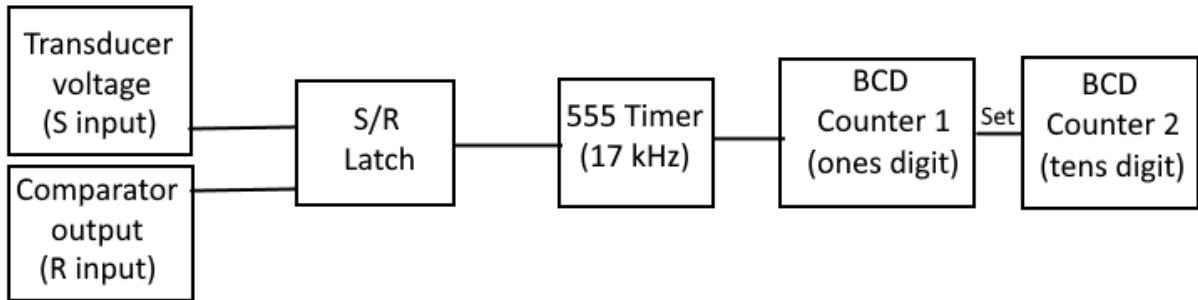
Module 2: Signal Amplification



The second module has to do with the amplification of the wave reflected off the object. The incoming wave which is read by the receiver is then passed through a buffer before it can be further amplified for processing. This is done to remove digital noise from the signal and ensure a cleaner signal. Once filtered, this signal is passed through a two stage inverted amp with bandpass filters to only allow a narrow range of required frequencies to pass through. This final amplified signal is then sent to one of the inputs of a comparator with a small DC reference voltage at the other input. The comparator compares both of its inputs and outputs a digital signal indicating which one is larger. The main purpose of the comparator is to convert the analog signal (acoustic wave) back into a digital signal (square wave).

Module 3:

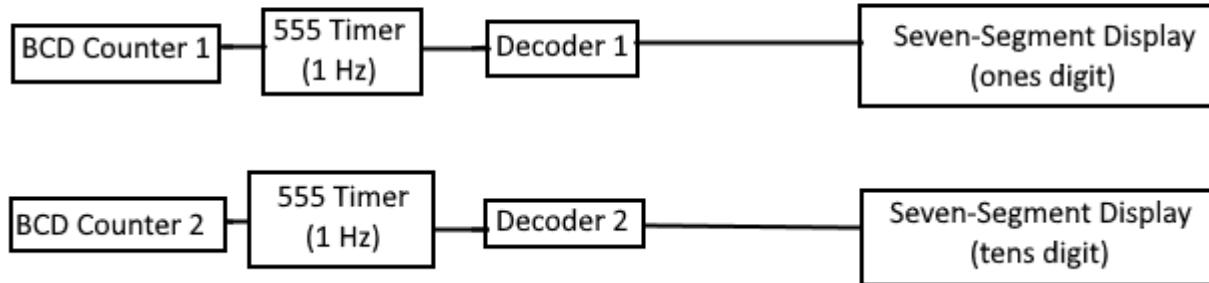
Module 3: Signal Processing



The digital signal outputted from the comparator and the initial post amplification input signal for the transducer are then inputted into an S/R latch. Both of these signals are delayed by some Δt . The purpose of the S/R latch is to detect when the receiver signal is high and in turn count the number of cycles which is stored in two BCD counters, one for the 1's column and one for the 10's column. The output is passed through another 555 timer with a 17Khz frequency to basically convert the digital signal into a distance value. Using the speed of sound and the frequency of the timer, we can determine the wavelength of each cycle and given the number of cycles, we can calculate the distance travelled by the wave while keeping in mind that the wave has to travel to and from the object (distance divided by two).

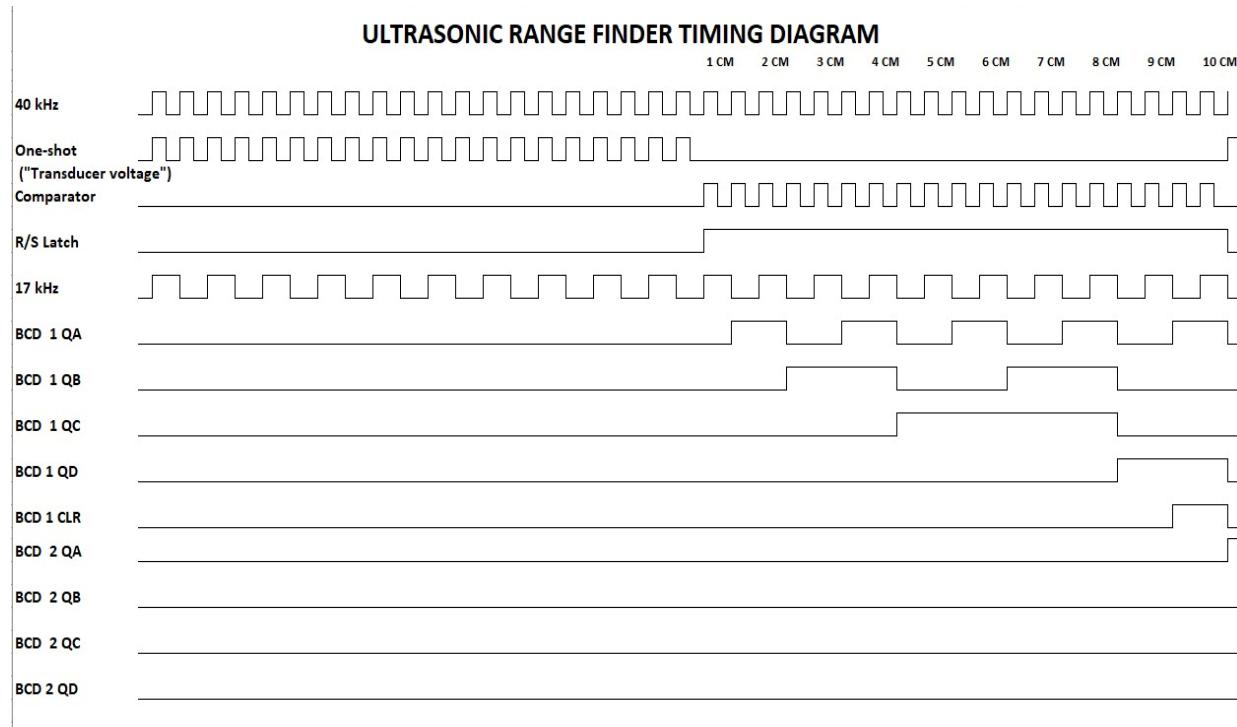
Module 4:

Module 4: Displays



The final module has to do with converting the distance to a binary output for the seven-segment display. The two seven-segment displays increase by one for each peak counted by its respective BCD counter. In order for the display to actually read the signal it must be connected through a decoder chip. Furthermore, the counter signal is passed through another 555 timer with a frequency of 1 Hz to give the display a constantly updating value (refresh rate of the display).

Timing Diagrams:



Week 3

Once the design of all four modules is complete, experimental work is required to test if all parts of the circuit work accordingly. In order to test the working of each module, they were built on either a breadboard or in multisim (or both) and their outputs were observed to determine if any changes needed to be made.

Module 1: Input Signal

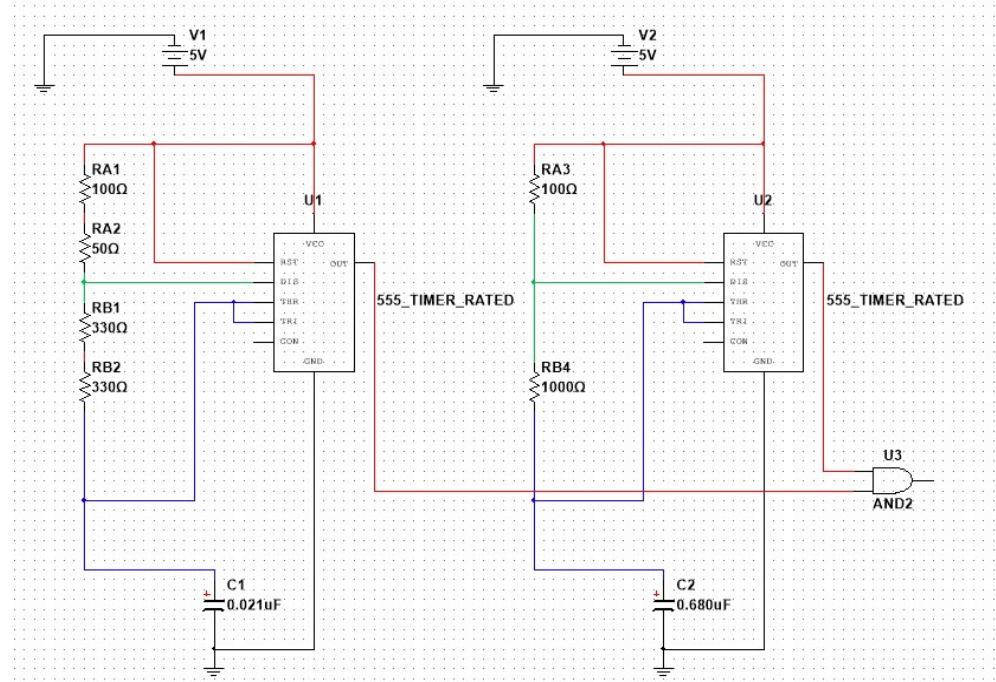


Figure 1: Module 1 (Input Signal)

Module 1 pertained to obtaining an optimal input signal for the transducer. This module consisted of two 555 timers with their outputs passing through an AND gate. The first 555 timer is intended to provide an output signal of approximately 40 kHz. In order to obtain this, the astable timer was set up with $RA = 150 \Omega$, $RB = 660 \Omega$ and $C = 0.021 \mu F$. This would theoretically operate with an output frequency of 46.74 kHz and a 55.1% duty cycle. The second 555 timer which was initially meant to be a monostable multivibrator was replaced with an astable timer with an intended output of 1 Hz. This was set up with $RA = 100 \Omega$, $RB = 1000 \Omega$ and $C = 680 \mu F$ to theoretically provide an output signal with a frequency of 1 Hz and 52.38% duty cycle.

When simulated on multisim, the output of the first 555 timer was measured to be 41.0 kHz which falls within our required range and is therefore an accepted value.

However, the frequency of the second 555 timer could not be discerned in the simulation because multisim is unable to compute simulations in real time. Therefore for demonstrative purposes, the value for C in the second timer's build was lowered to 0.680 uF and the output frequency was reported as 725 Hz. The purpose of the second timer is to create periodic pulses of the initial 40 kHz input and therefore as long as the value is significantly smaller than the initial input and the period of the wave is discernible, the build is still theoretically viable.

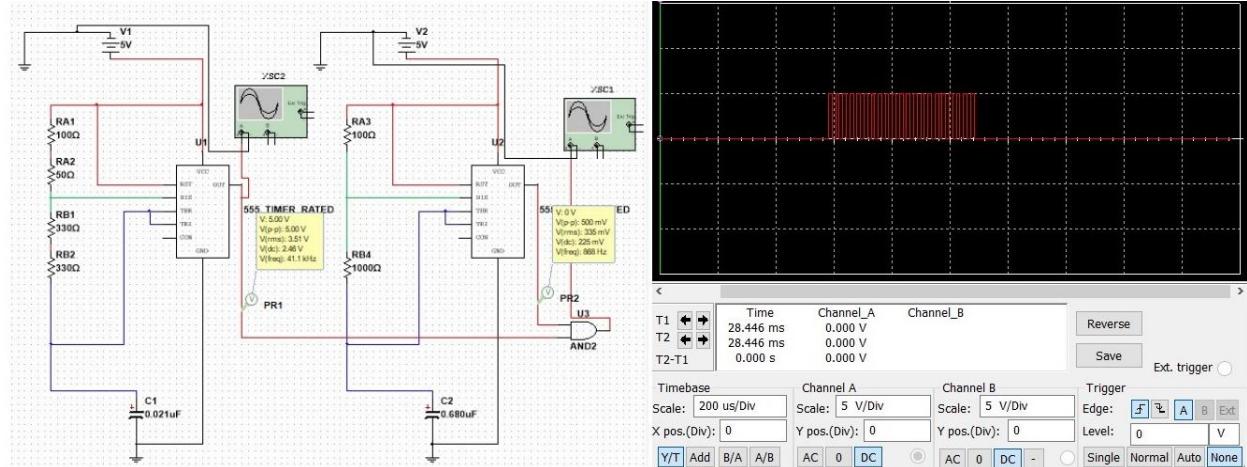


Figure 2: Module 1 Simulation

When experimentally built on a breadboard the output of the first timer was measured to be around 36 kHz. In order to raise this value to an acceptable value between 38-42 kHz, RB was lowered to 500 Ω. The second timer when measured with the oscilloscope on the actual build provided an output of 1.1 Hz which is an acceptable value.

Module 2: Signal Amplification

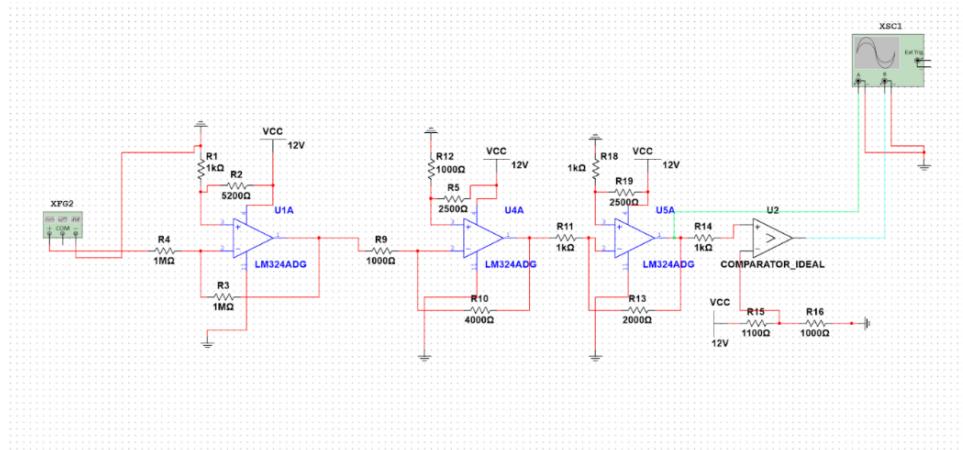


Figure 3: Receiver Signal Amplification

The second module pertains to the amplification of the reflected receiver signal (sine wave) for further processing. For testing and simulation purposes, arbitrary values were used for the signal that was being processed. The input signal which was to be amplified was chosen to have a frequency of 5 Hz so the output is visible on the oscilloscope and the input voltage was chosen to be 150 mV peak. In this section, the signal passes through three op-amps before being sent to a comparator.

The first op-amp is the ‘pre-amplification stage’. This amplifier acts as a buffer or filter of sorts because of the two $1 \text{ M}\Omega$ resistors. The resulting output is just the inverted input wave (phase shift of π):

Purple: Input wave
 Green: Op-amp 1 output

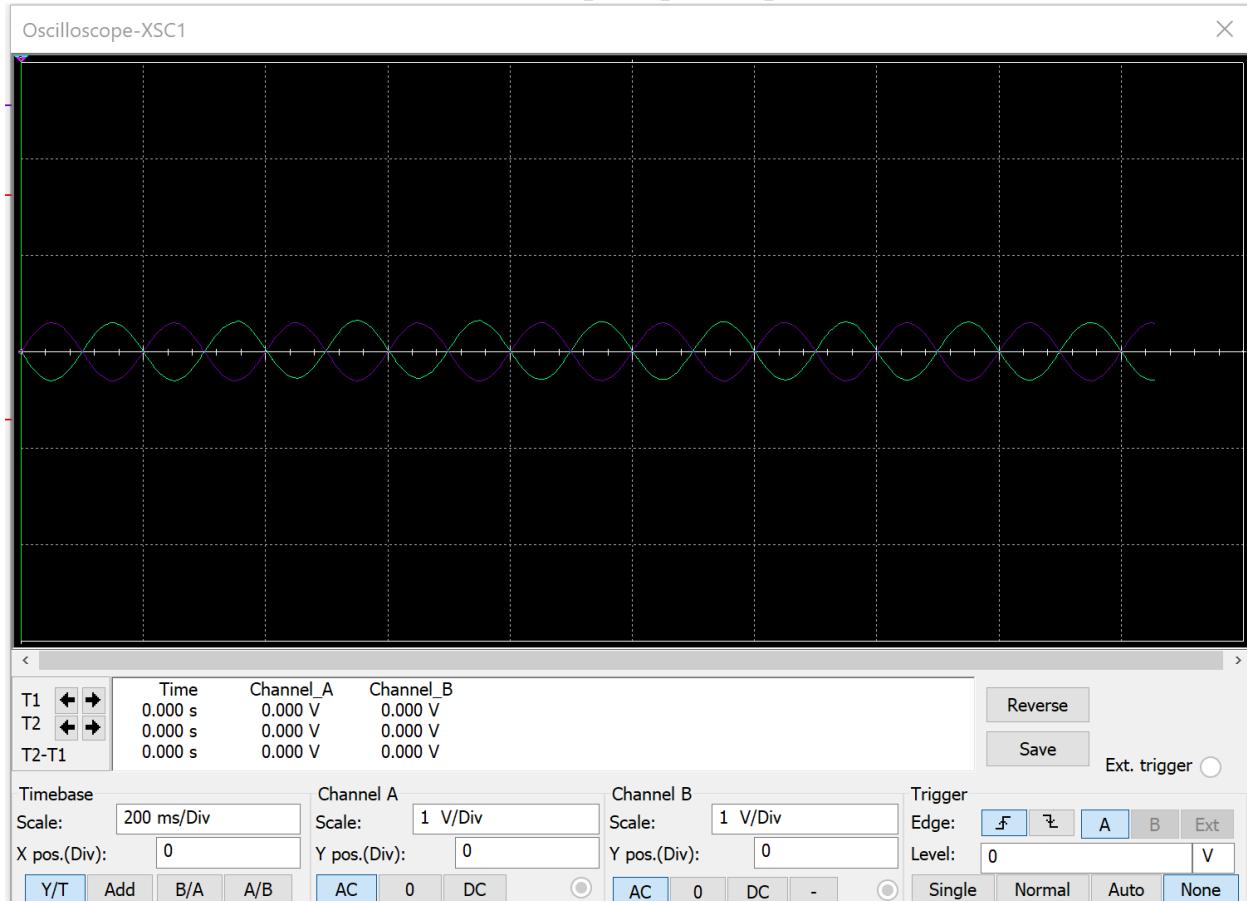


Figure 4: Pre-amp Output (Op-amp 1)

The second op-amp amplifies the signal as seen:

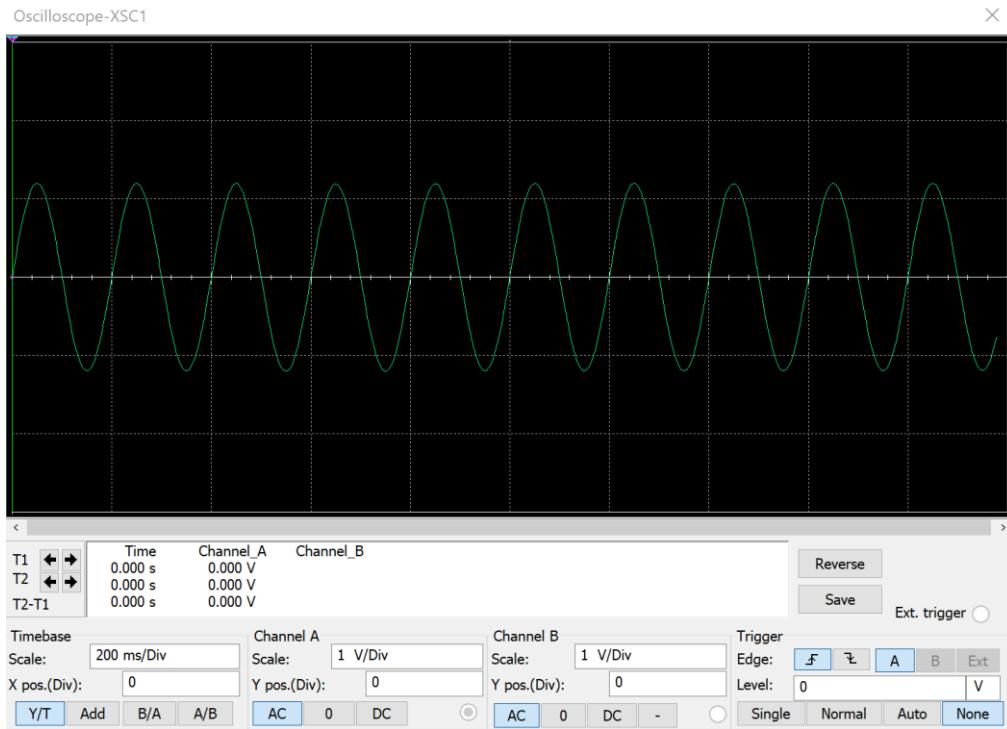


Figure 5: Op-amp 2 Output

The third op-amp amplifies the signal as seen:

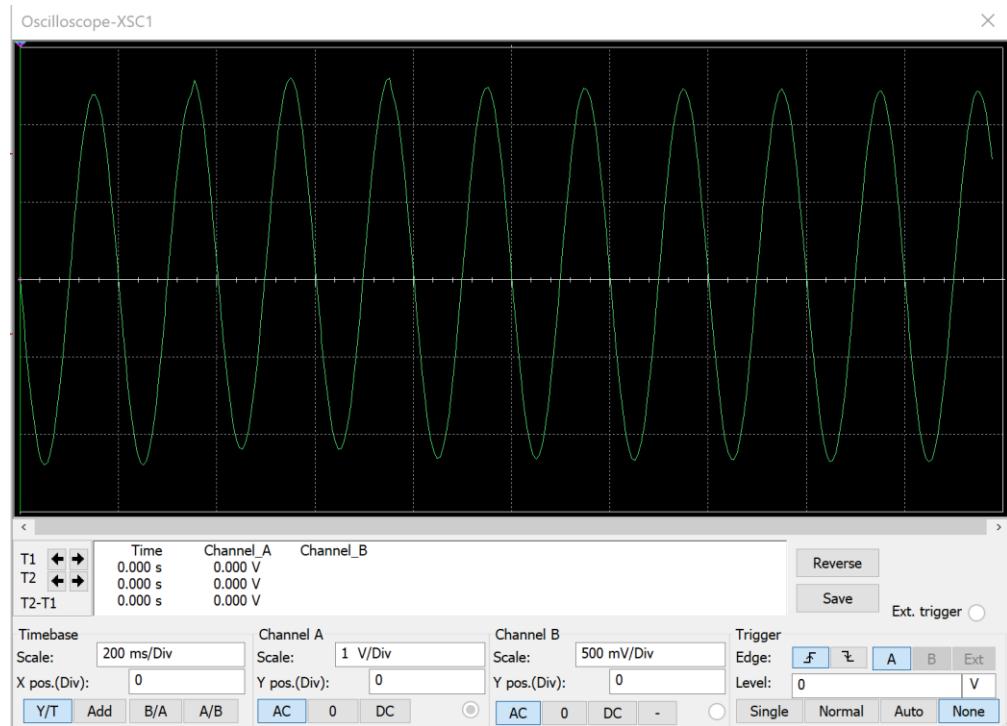


Figure 6: Op-amp 3 Output

After the two stage inverted bandpass amplification, the signal is passed through a comparator to essentially convert back this sin wave to a square wave for further processing. The comparator does this by comparing this input signal to an arbitrary small reference voltage and outputting the maximum voltage in square wave form. The resulting output of the comparator is a square wave with the same voltage as the sine wave output of the third op-amp:

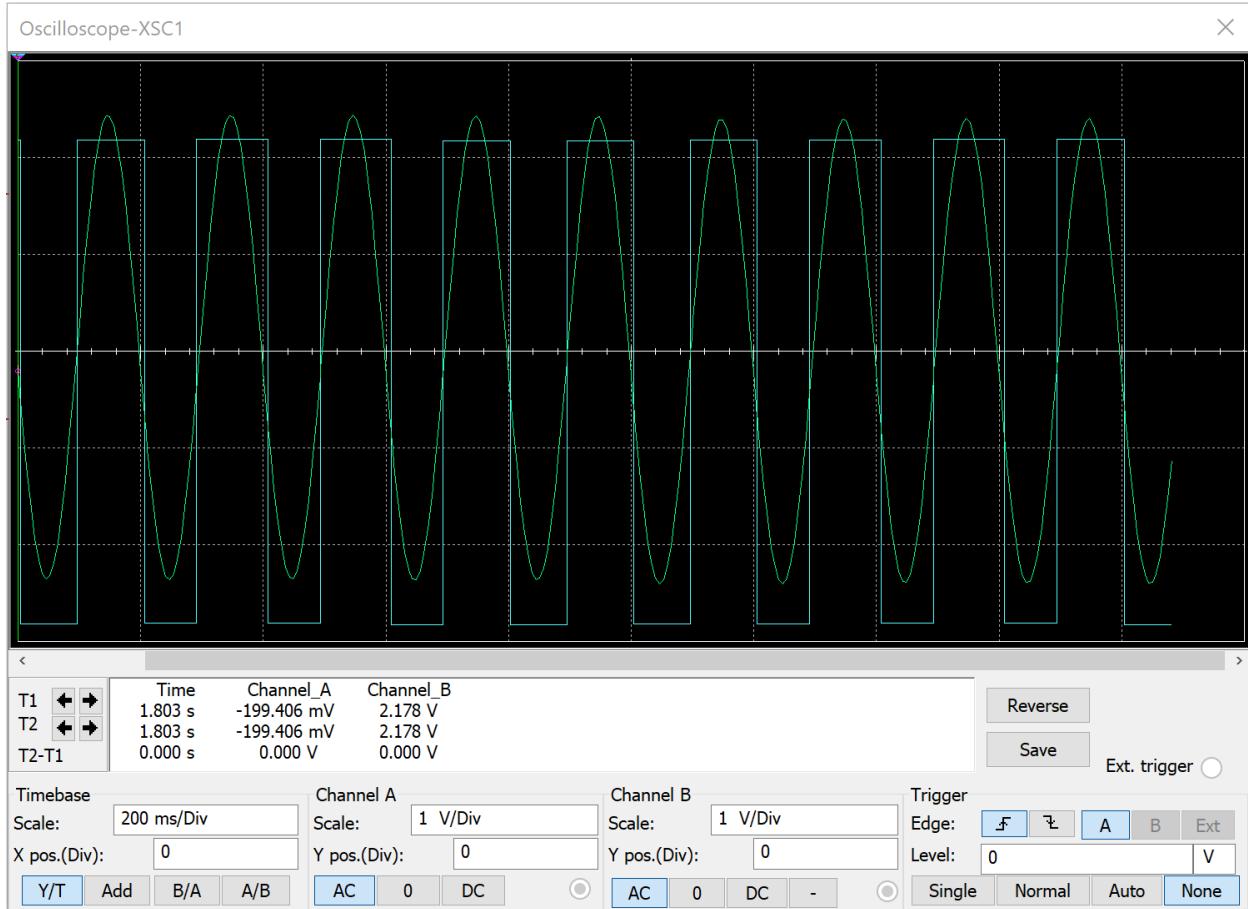


Figure 7: Comparator Output

Module 3: Signal Processing

Since the output of the comparator is now a square wave version of the amplified receiver signal, this signal can simultaneously be observed with the input signal from module 1 and considering that both signals have some delay Δt , the number of cycles at the receiving end can be counted. This is done with the help of an S/R latch which takes in both signals as one of its inputs and detects when the receiver signal is high. The output of the S/R latch is then passed through another 555 timer with an output frequency of 17 kHz (approximately half of initial input signal) to convert the signal into an actual distance value in cm. This value is stored on

two BCD counter, one for the (1's) digit and the other for the (10's) digit. The first counter resets to 0 when the number goes above 9 and the second counter starts iterating from 1.

The entire build until Module 3 can be seen below(excluding transducer):

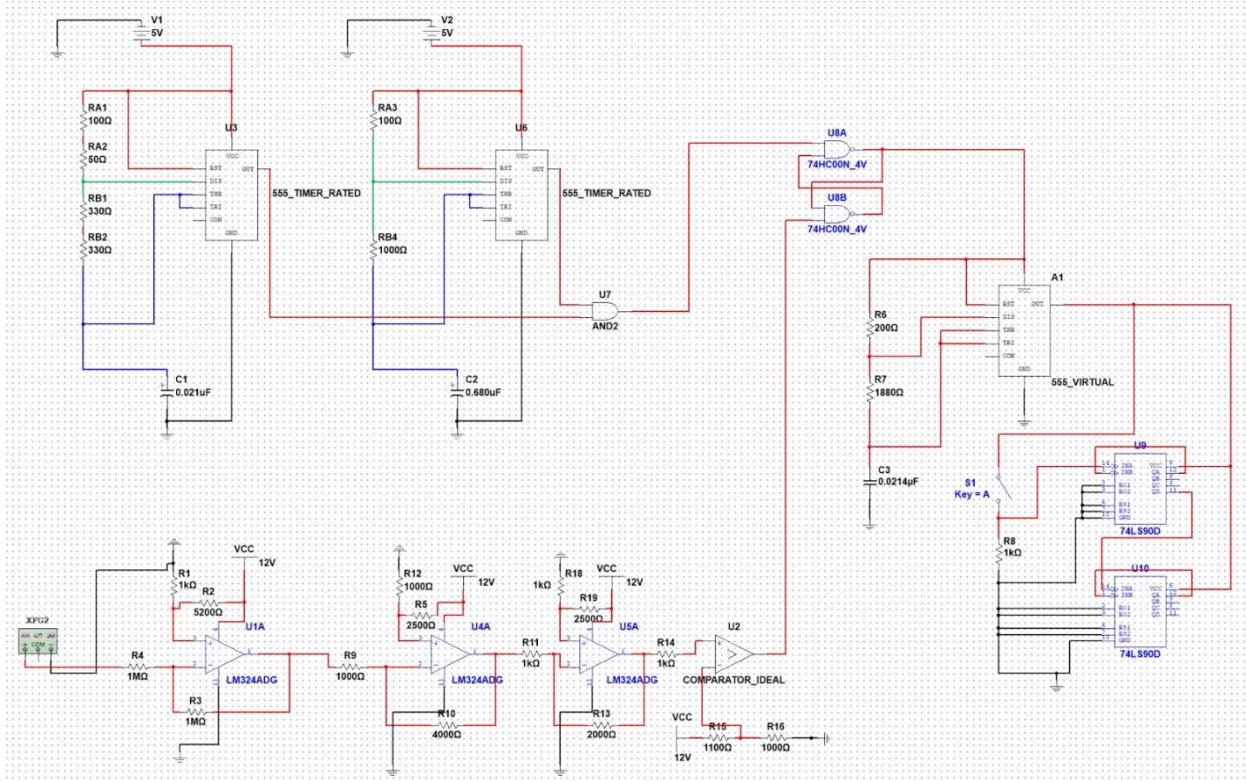


Figure 8: Signal Processing

Module 4: Displays

Once the signal is processed after module 3 and the distance stored on the counters, module 4 pertains to displaying that distance on our two 7 segment displays (one for each counter). The output of the counters is first passed through another 555 timer with an output of 1 Hz. This basically acts as a refresh rate for our display so the value is constantly being updated as the distance between the transducer and object changes. This signal is then hooked up appropriately to a 7 segment decoder before being connected to the 7 segment display as shown below (for simulation purposes the 1 Hz refresh timers are omitted):

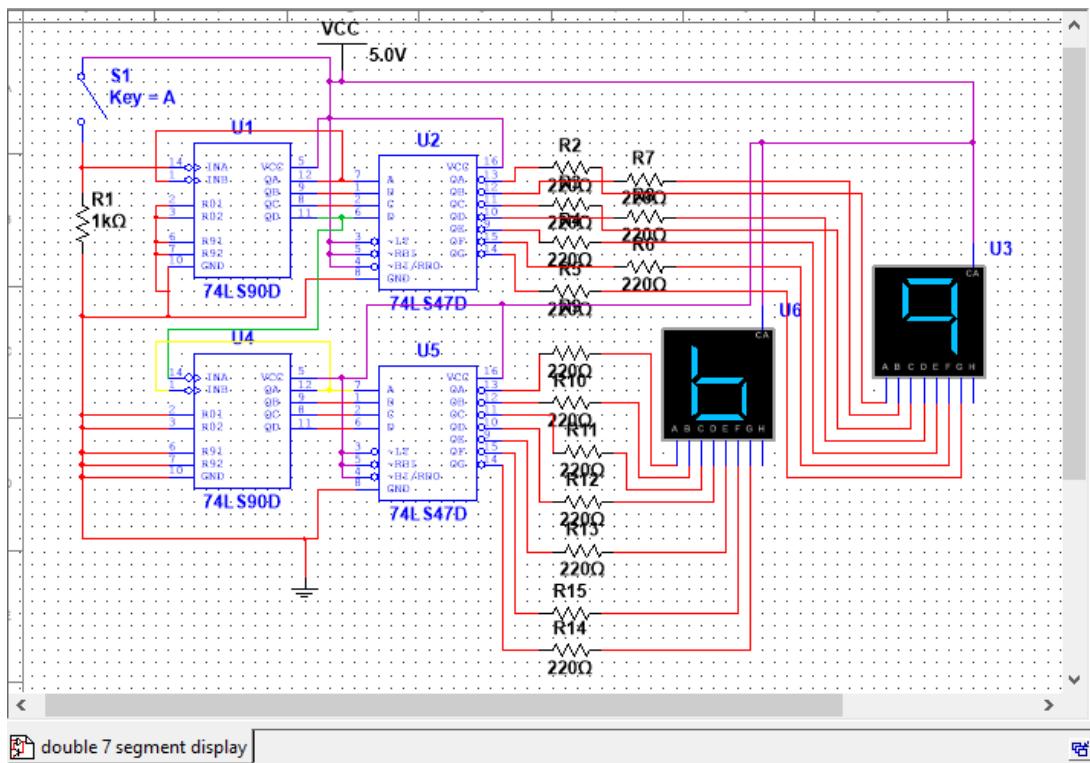
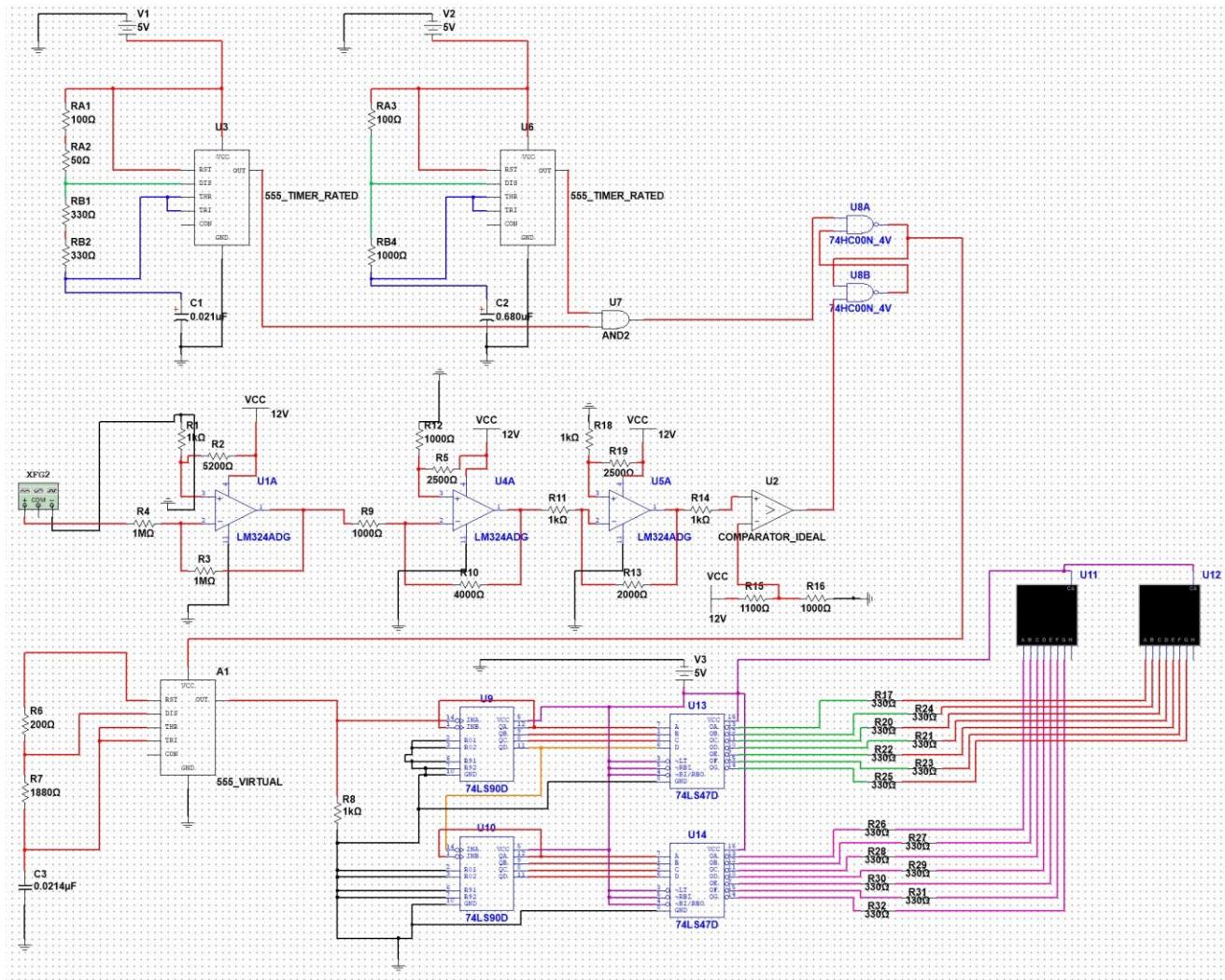


Figure 9: Display Module

The 7 segment displays then display an output up to 99 cm with one of them displaying the ones digit and the other one displaying the tens digit.

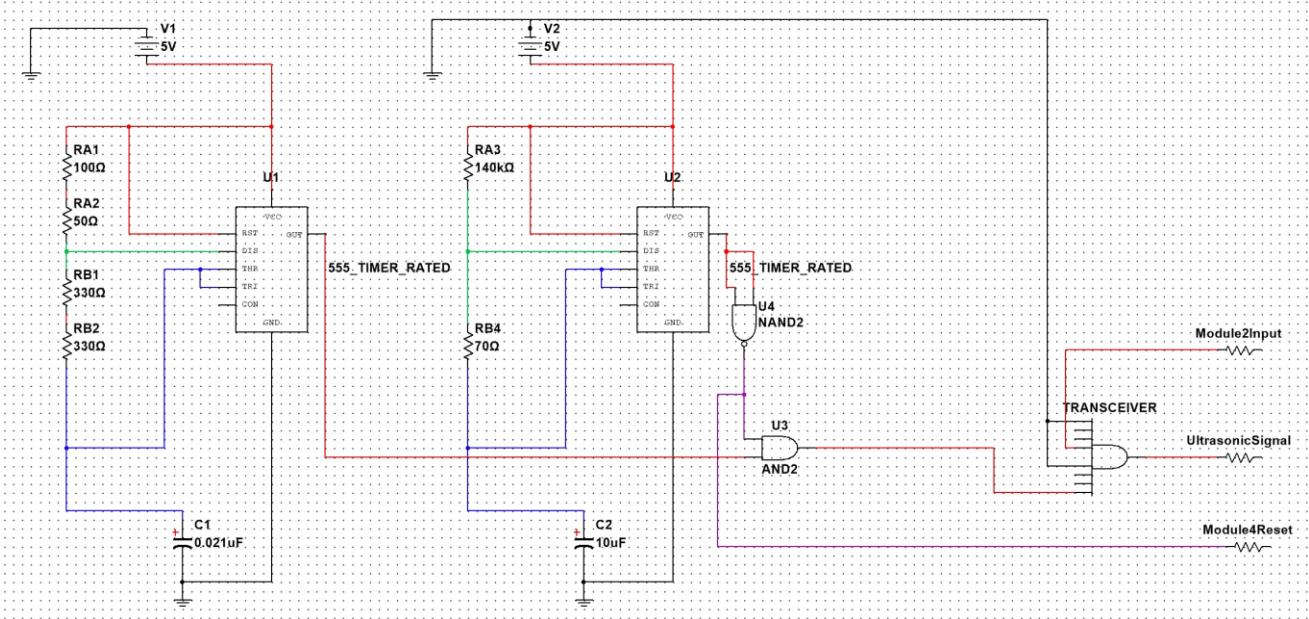
Week 4

After individually testing each module and ensuring they all provide the intended output, the final build was put together by hooking up all four modules appropriately. The complete circuit diagram of the final build is as shown below:

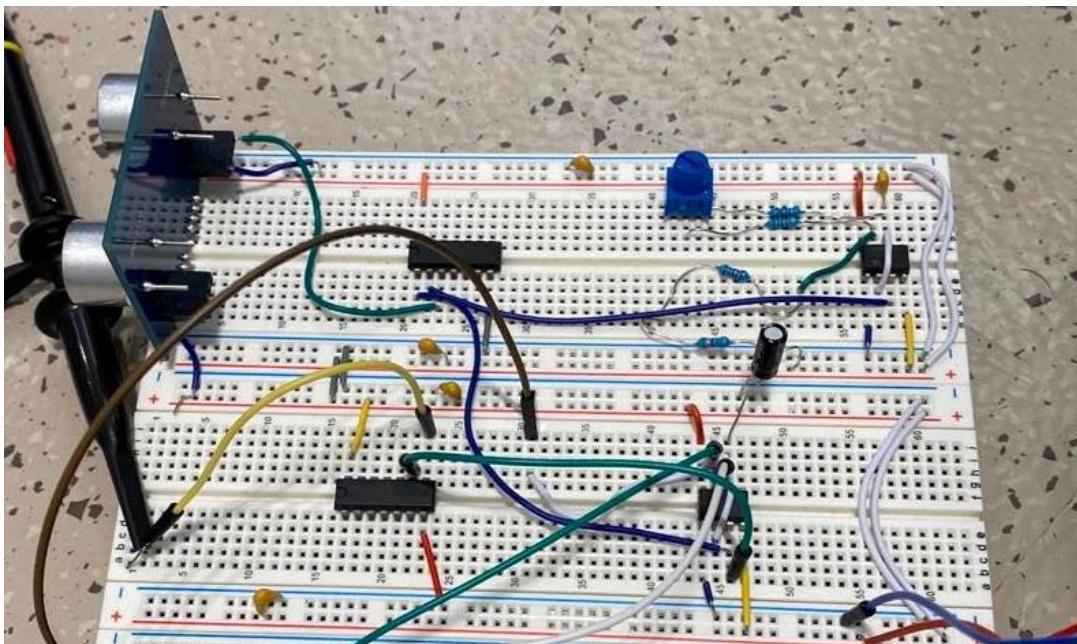


Considering the sheer size and complexity of the build, it was divided into four modules which were all built separately.

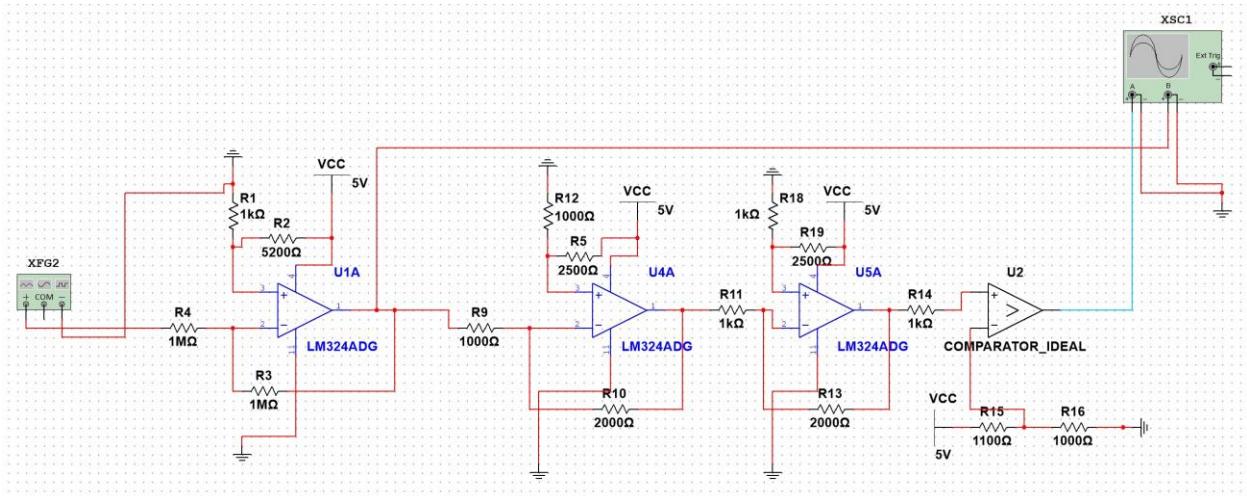
Module 1: Input Signal



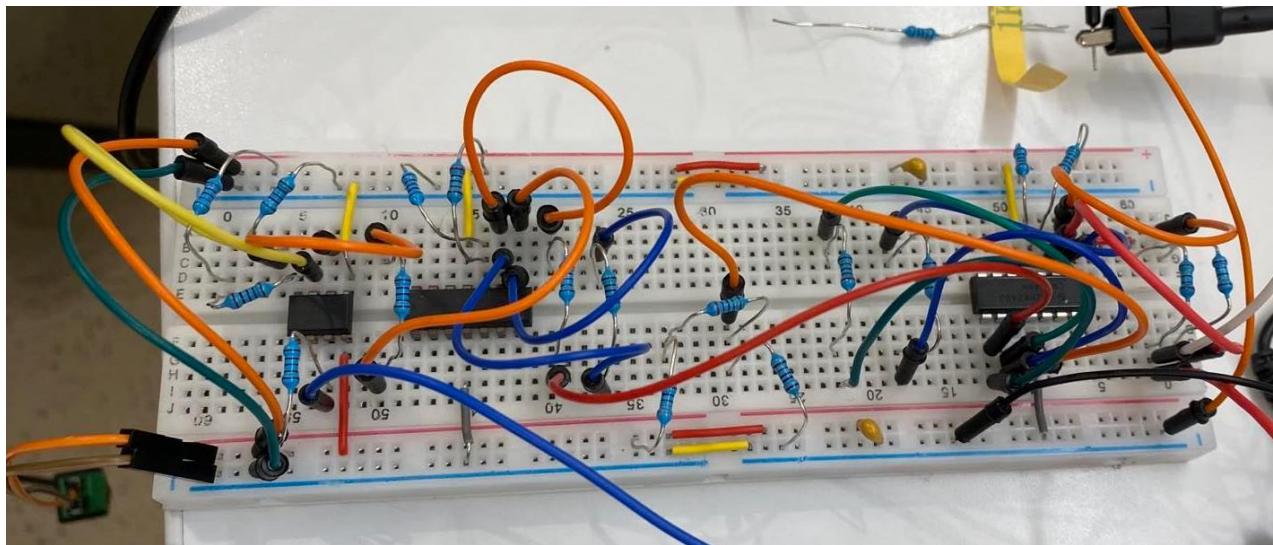
The input signal module was built on two breadboards. This module consisted of two 555 timers, one set at 40Khz ($R_1 = 150 \Omega$, $R_2 = 660 \Omega$, $C = 0.021 \mu\text{F}$) and the other at 1 Hz ($R_1 = 140\text{k}\Omega$, $R_2 = 70 \Omega$, $C = 10 \mu\text{F}$). The output of the second timer is first inverted to flip the duty cycle so that the signal is only HIGH for a very short period and LOW for the rest (99% duty cycle to 1%). Both of these timer's outputs are then sent through an AND gate before being sent to the transducer to produce the acoustic burst. The actual build is as shown below:



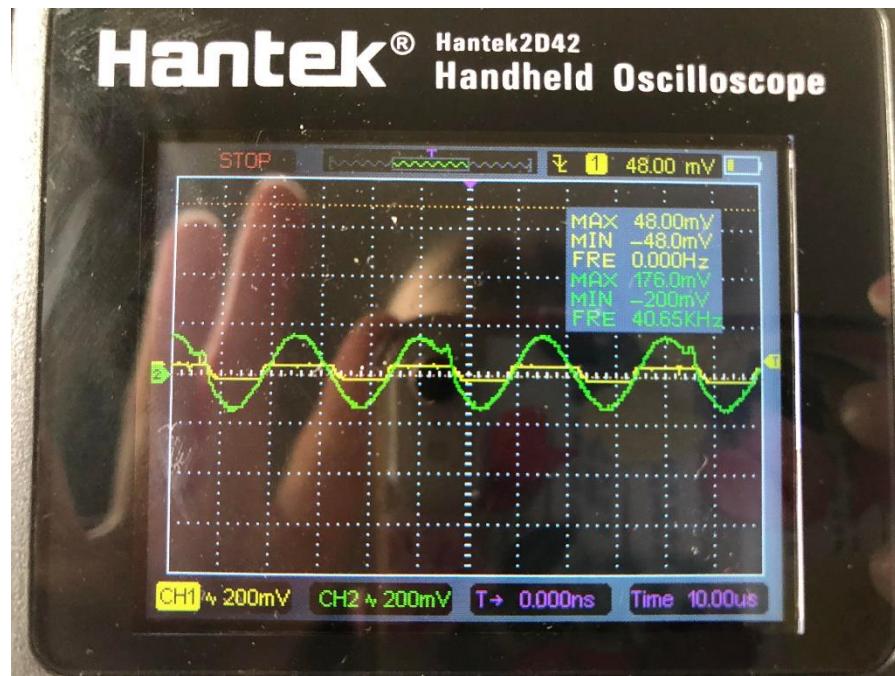
Module 2: Signal Amplification



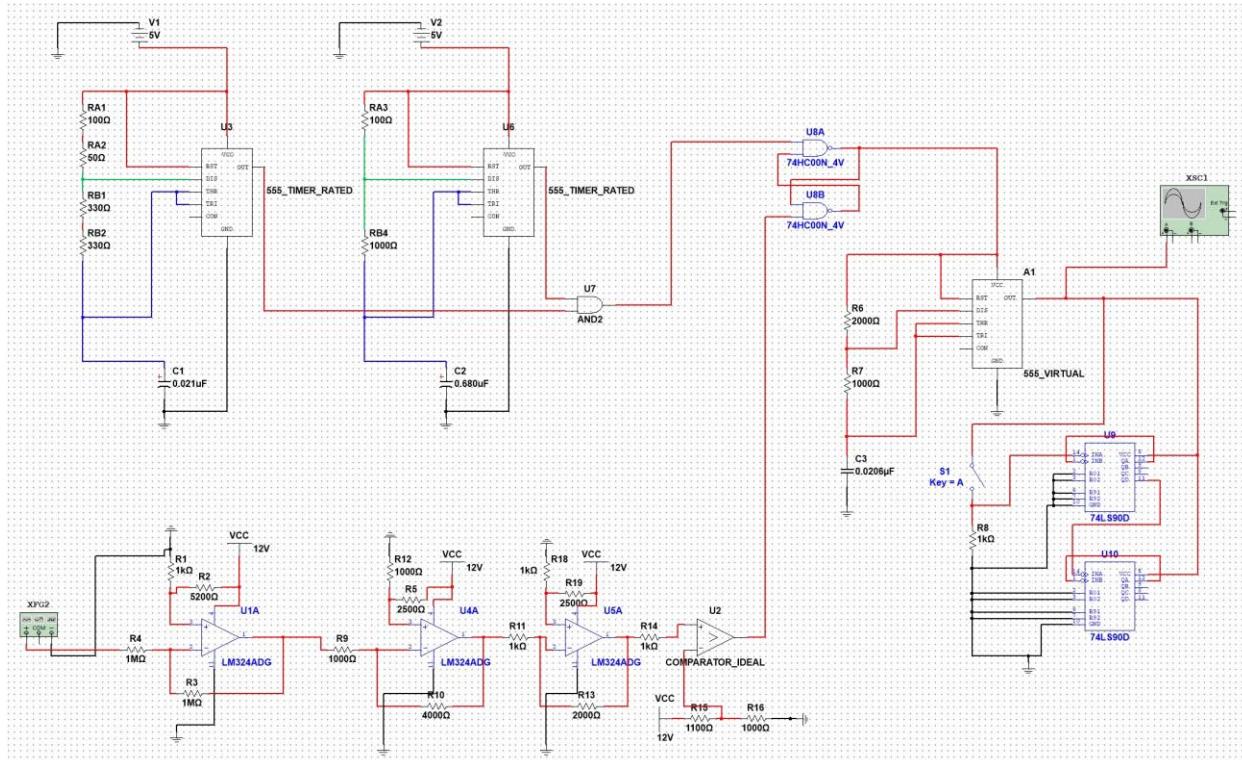
The second module focused on the amplification of the receiver signal for further processing. Since it was built in a modular fashion, the input signal is just an arbitrary sin wave from a function generator. The signal first passes through an op-amp meant for filtering out the noise (buffer). After cleaning the signal, it is passed through two more op amps (two stage inverted amp) to further amplify the signal before it is passed through a comparator to convert it into a square wave. The gain from all three op-amps is 1.29, 2.13 and 1.59 respectively. The physical build for this module is shown below:



The output of this module is a square wave which can then be sent forward to module 3 for further processing:

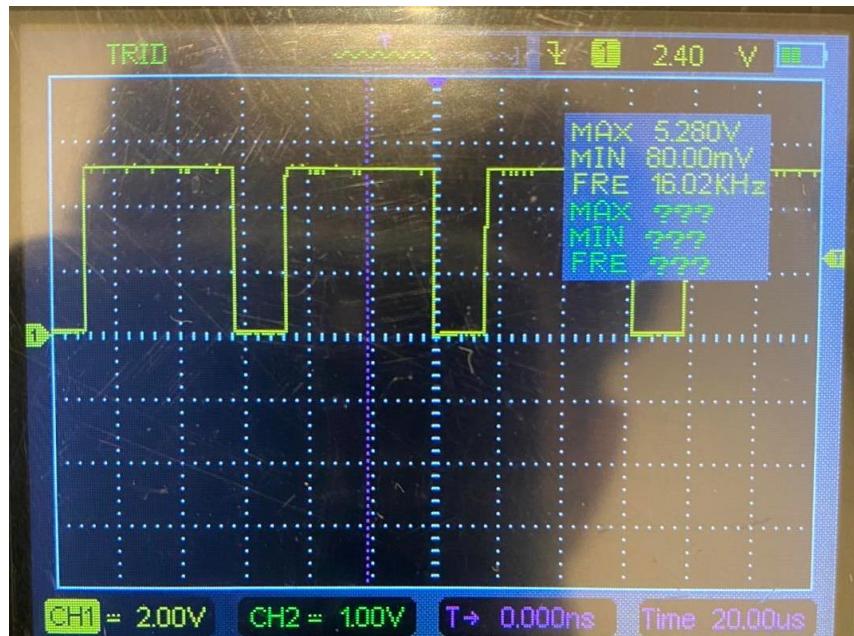


Module 3: Signal Processing

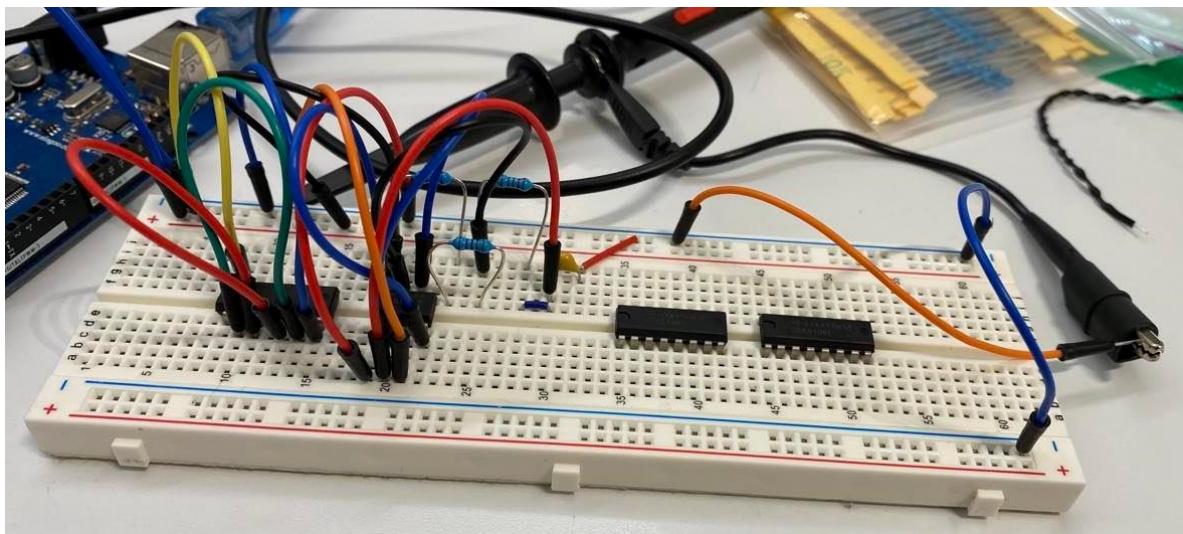


The third module takes the inputs from both module 1 (before sending to transducer) and module 2 (comparator output) and passes these through an SR latch. Since both of these signals have a delay, the SR latch is able to detect and

store every rising edge and in turn produce a corresponding signal which is passed through a 17Khz timer to essentially convert the signal into a distance value. The 17Khz signal is produced with another 555 timer ($R_1 = 2000 \Omega$, $R_2 = 1000 \Omega$, $C = 0.0206 \mu F$). In the actual build the output was measured to be approximately 16Khz which is an accepted value.

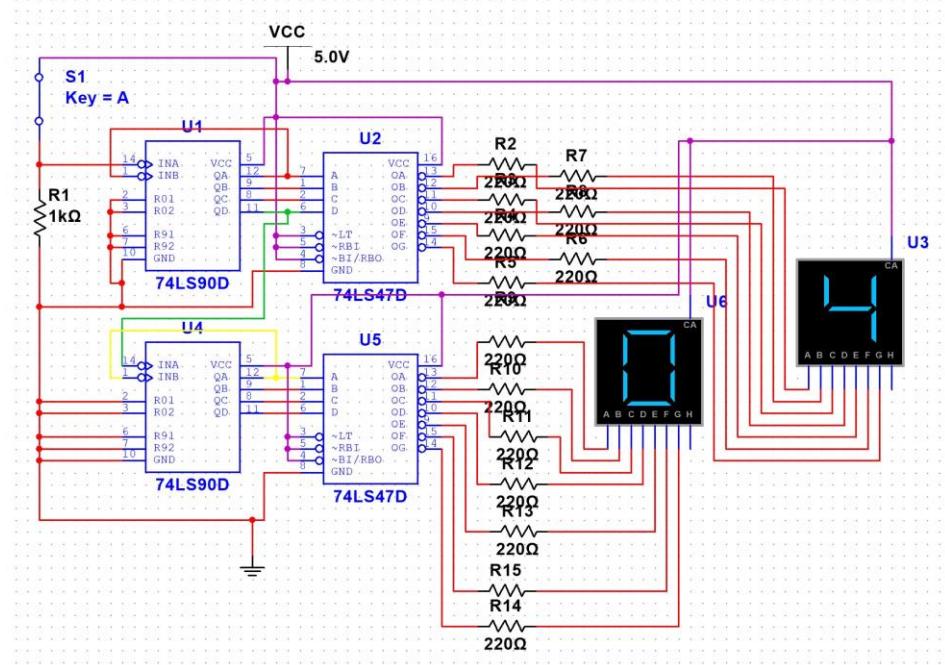


The actual build was hosted on one breadboard and is as shown below:

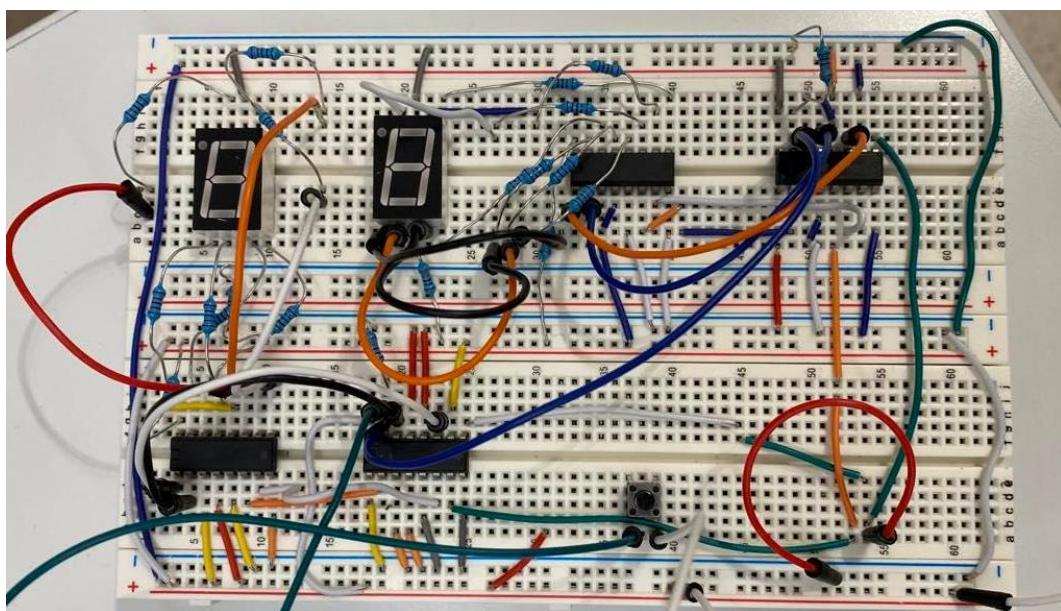


The two BCD counters were omitted in this build as they were included in the following module.

Module 4: Displays

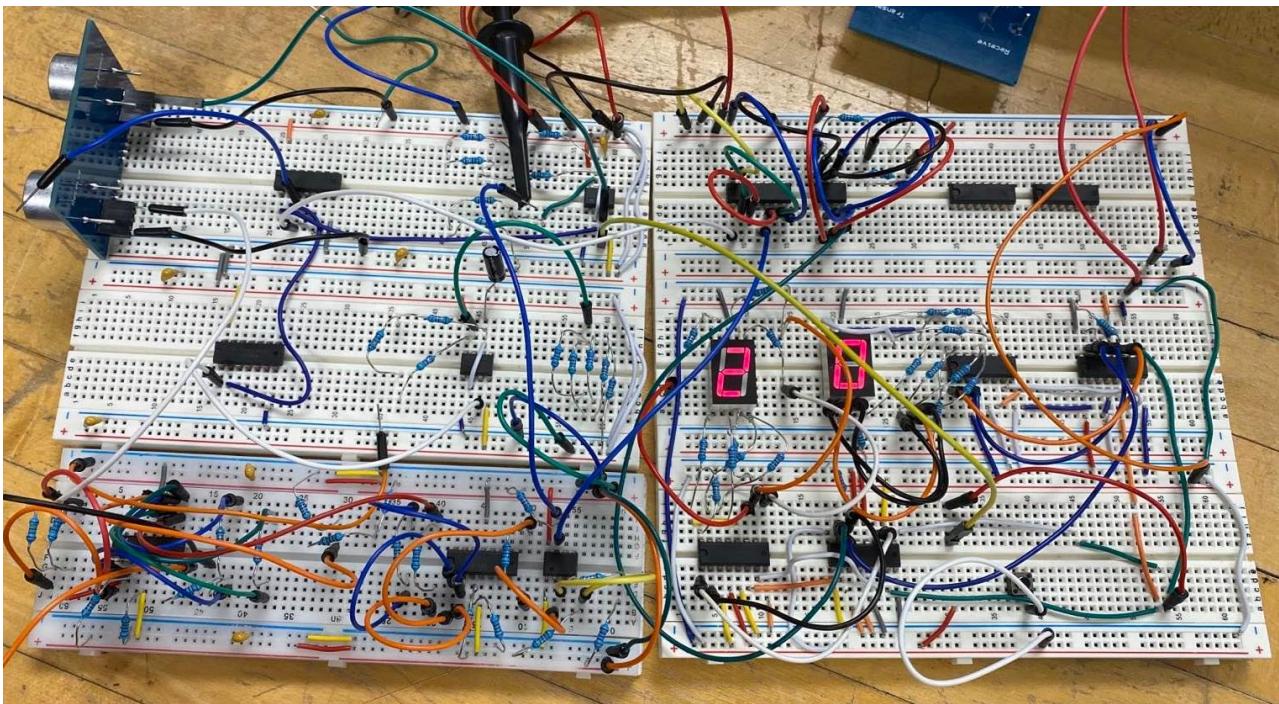


The fourth module works directly with the output signal of module 3 (output from 17Khz timer). After hooking up the signal to the respective BCD counters, 7 segment decoders are added before finally hooking up the 7 segment displays for the outputs. One component missing from the circuit diagram above is the 1Hz signal for the display refresh rate. This signal is added right before the decoders and is obtained directly from the 1Hz 555 timer in module 1. The build of module 3 is as shown below:



Final Build:

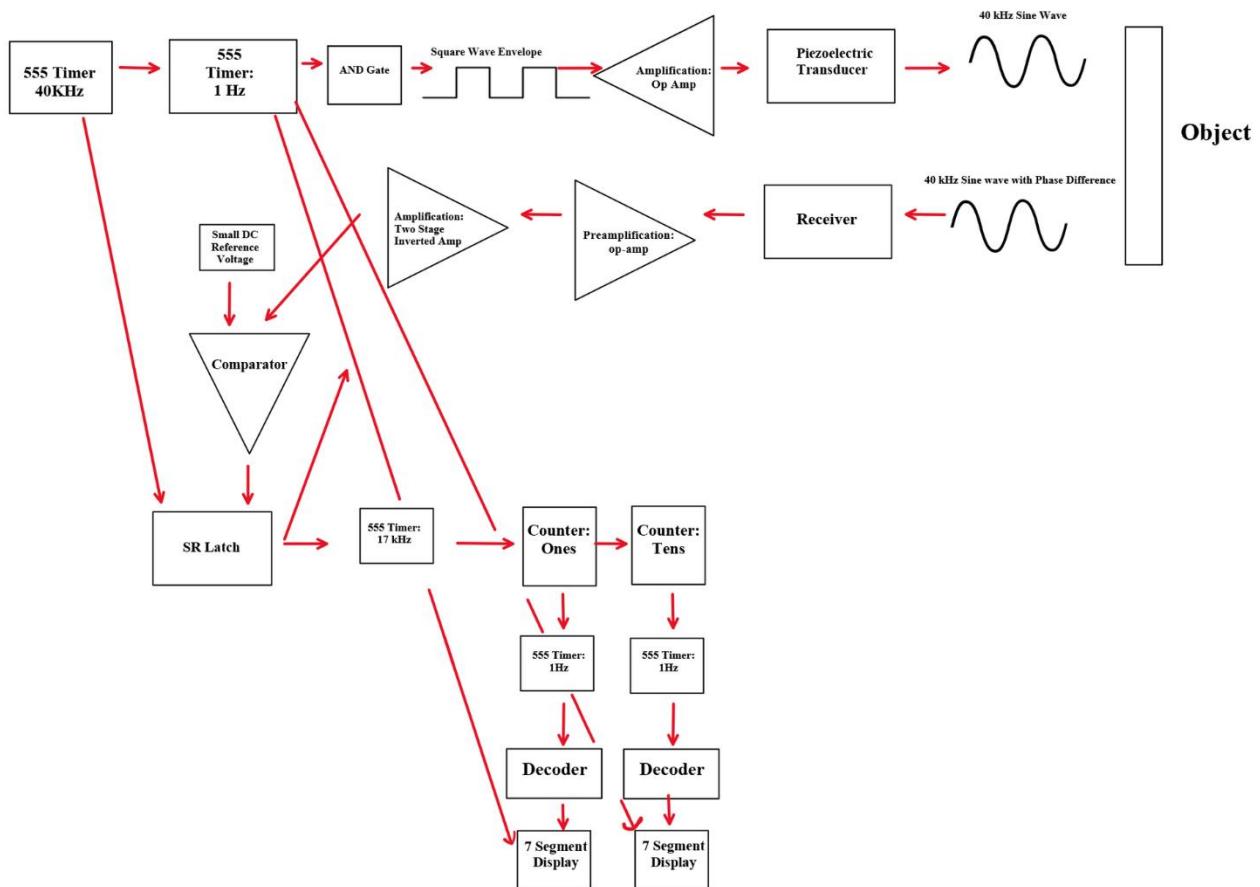
A total of six breadboards were used to host the final build as shown:



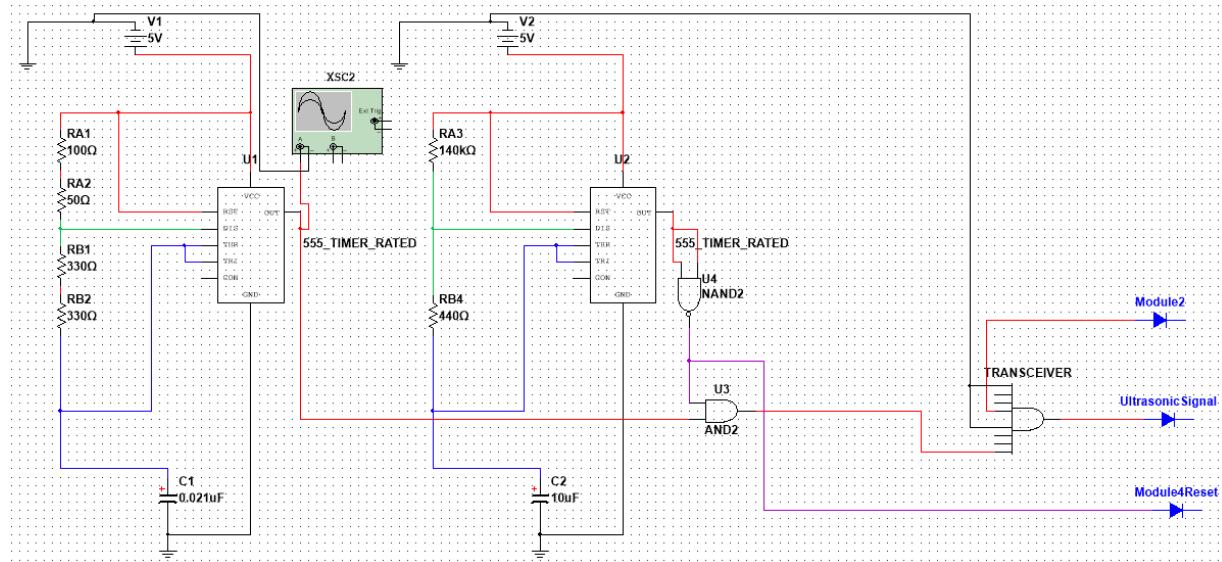
Unfortunately, the outputs weren't as expected and the device didn't work as intended. The 7 segment displays would not update and would hold still at a random number even when the signal is reset. One possible reason could be too much noise in the receiver signal despite the filter. Due to this the waveform starts clipping and the comparator output isn't a clean discernible square wave. Furthermore, the 1Hz timer for refreshing the display may not be the most appropriate measure and maybe replacing that with a monostable multivibrator (one shot) might be a better alternative. Doing so would only refresh the display once a trigger is received and a refresh is needed. These concerns will be addressed in our final build in week 5 and an attempt will be made to ensure complete functionality of our device.

Week 5

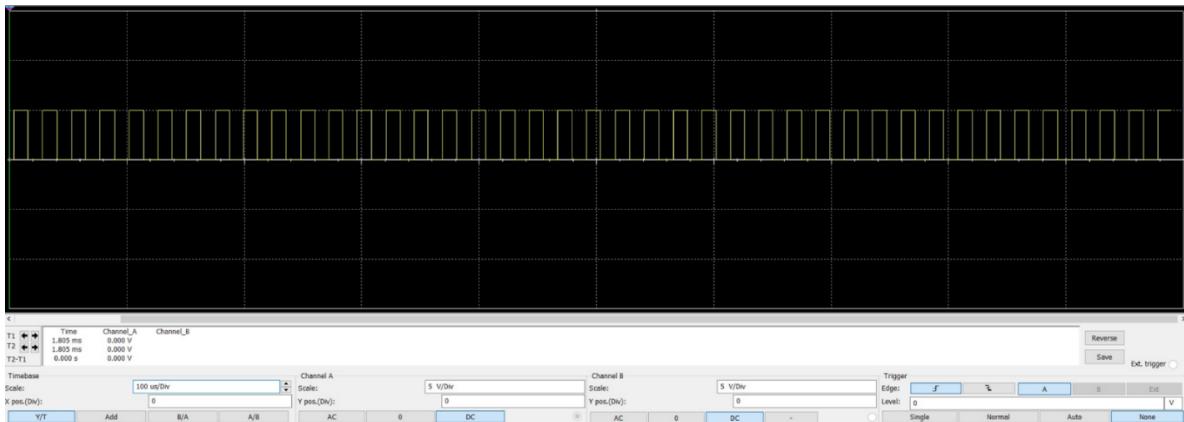
In an attempt to fix the physical build and achieve proper functionality of the device, some minor changes were made to the circuit and after testing on Multisim, it was rebuilt. The final block diagram of the build is as follows:



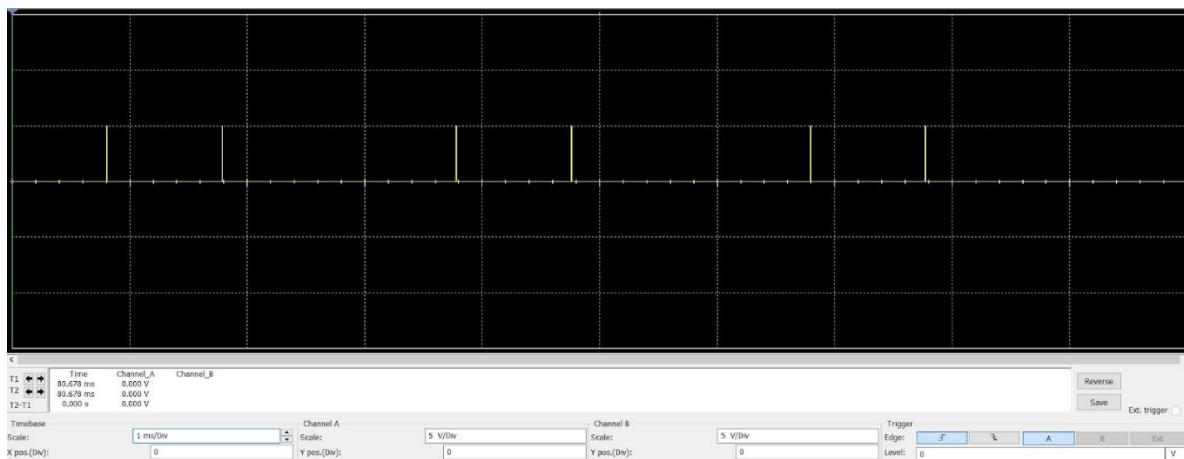
In the first module, the duty cycle of the 1 Hz timer was further reduced so that the signal is only high for 5 ms. This was done by using $R_1 = 440 \Omega$, $R_2 = 140 \text{ k}\Omega$ and $C = 10 \mu\text{F}$ and then inverting the output so that the signal is inactive for most of the cycle and only high for a small fraction of a second to create the required wave pulses. Doing so allowed us to obtain the desired signal of 1 Hz with a duty cycle of 0.31%. Shown below is the final circuit diagram for module 1:



Since multisim is unable to depict such a low frequency, the non-inverted output is measured instead as is shown below:



When the waveform is inverted the output is as shown below. For proper depiction, C1 is reduced to 0.1 uF making the period 9.76 ms instead of 0.976 ms:

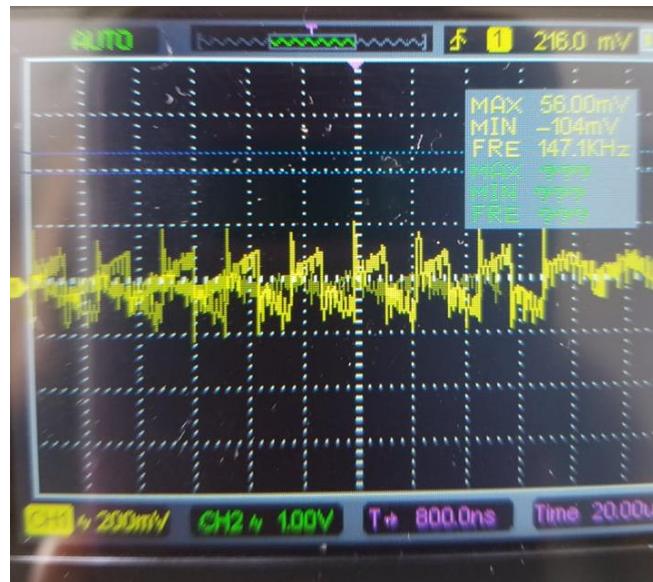


As seen above, approximately every 10 ms a packet is produced. The inconsistency in the gaps is supposedly due simulation errors in multisim as the simulation would crash and have to be reset multiple times.

After physically constructing module 1 on a breadboard and measuring the output (input to the transceiver) the following output is seen which is deemed appropriate:

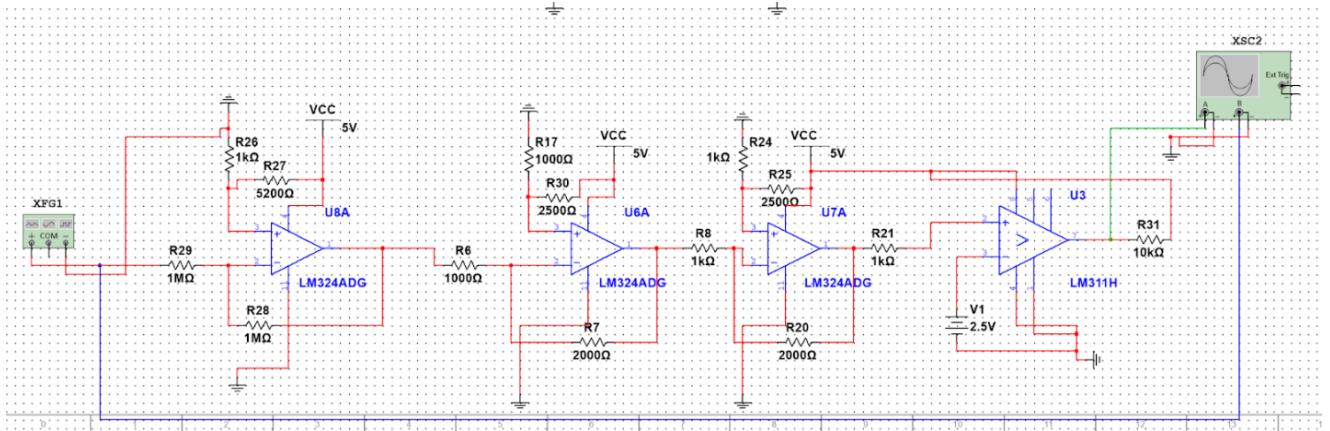


As we can see a proper square wave is detected as required with some slight noise. When inputted to the transceiver, the output from the receiver end is measured as follows:

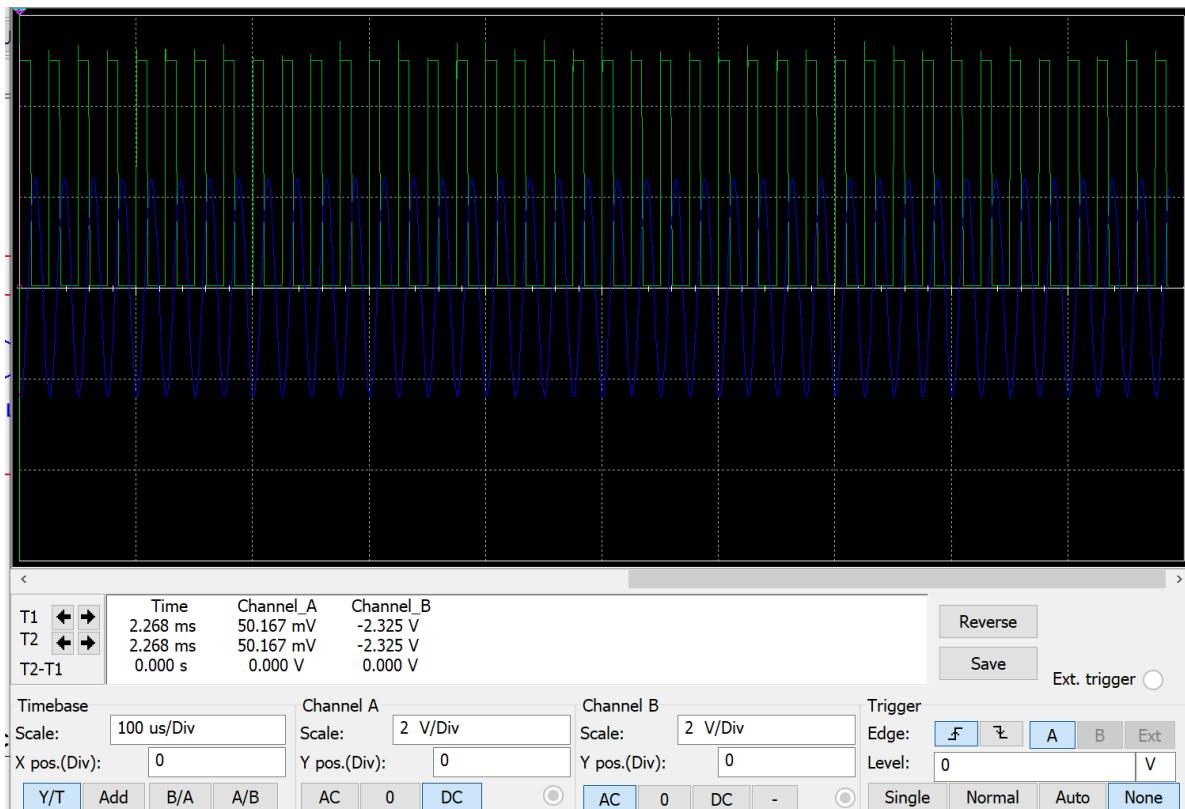


The pulses at the receiver end correspond correctly to the input pulses except with lots of noise as expected.

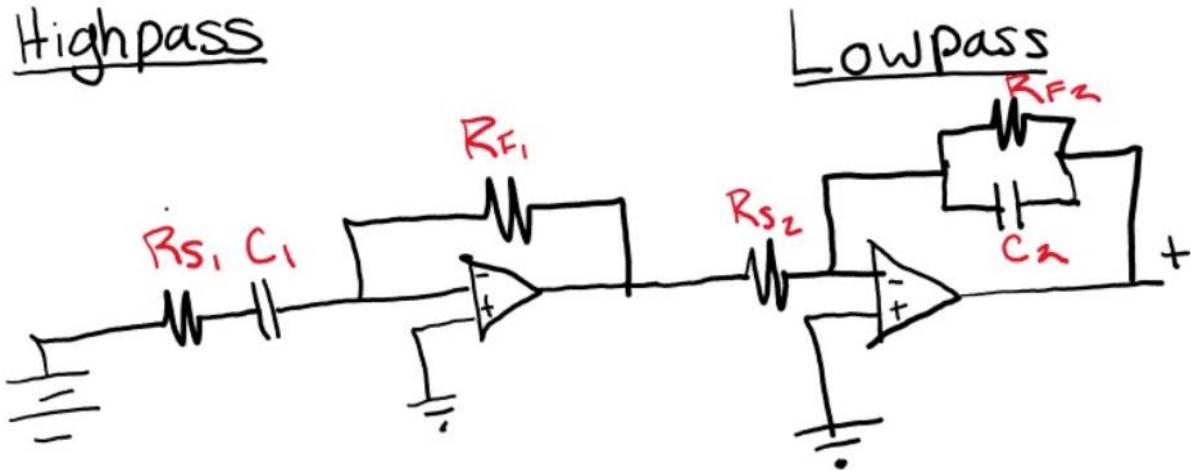
The receiver signal is then sent to module 2 to clean up the noise, amplify it and then convert back to a square wave using a comparator for further processing. Again, the module was first built in multisim for testing:



The input used for the simulation was an arbitrary 1.2 V sine wav. The first op amp with two 1 MΩ resistors behaves as a buffer (filter) and gives a gain of 1. The next two op-amps give a gain of 2. The comparator uses a reference voltage of 2.5 V so whenever the input signal is greater than 2.5 V a high signal is achieved and a low signal when it's less. With this switching between 0 and 5 V, the desired square wave is obtained:



In order to make the active bandpass filter, a high pass and low pass filter are used together. Our desired frequency range was chosen to be 30 kHz to 50 kHz to counter any discrepancies in our original 40 kHz wave.



The calculation for the filters is as follows:

High Pass filter:

$$1/(R_s C) = 30 \text{ kHz}$$

$$1/(2\pi 1000 C) = 30 \text{ kHz}$$

$$C = 5.305 \text{ nF}$$

Low pass filter:

$$1/(R_f C) = 50 \text{ kHz}$$

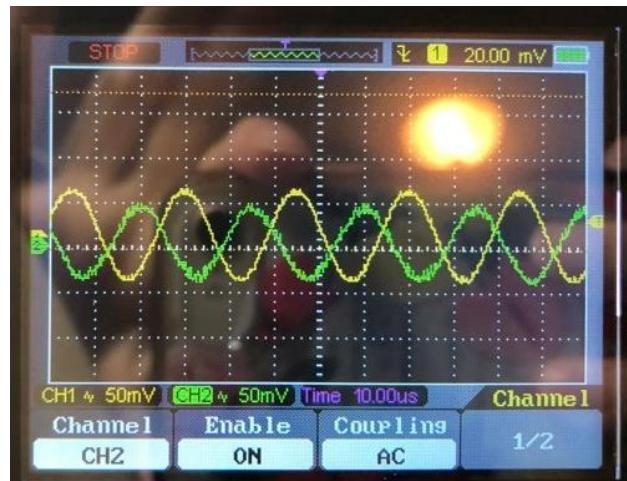
$$1/(2\pi 2000 C) = 50 \text{ kHz}$$

$$C = 1.59 \text{ nF}$$

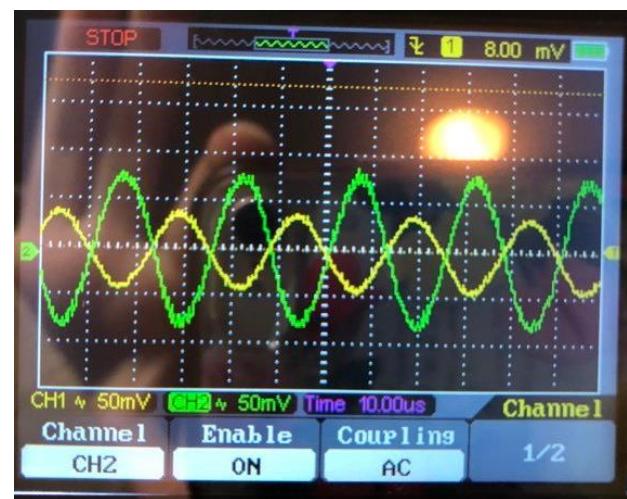
After adding the active bandpass filter however, we noticed signal clipping and therefore decided to remove it.

After a sufficient amount of multisim testing, the second module is built using the output from module 1 as the input signal. The output of all three amplifiers are shown below:

Op-Amp 1:



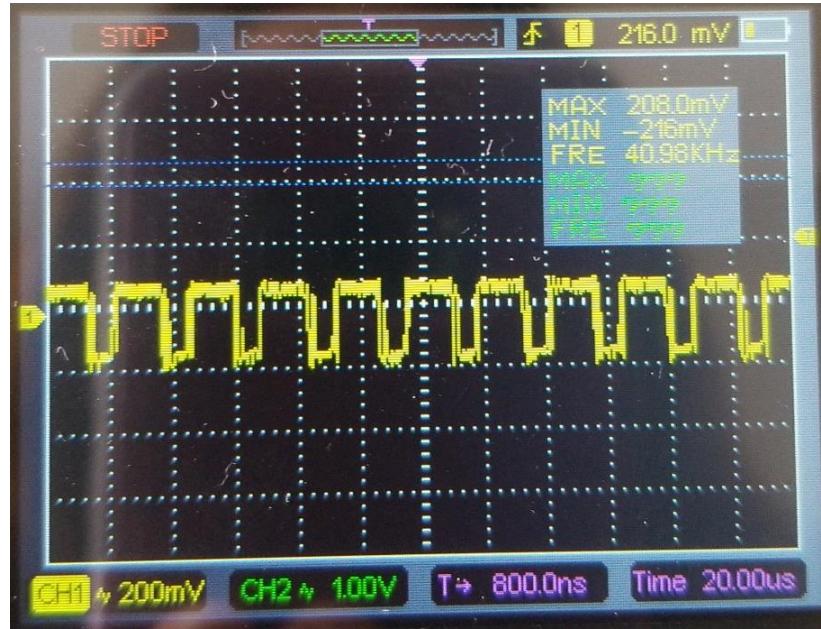
Op-Amp 2:



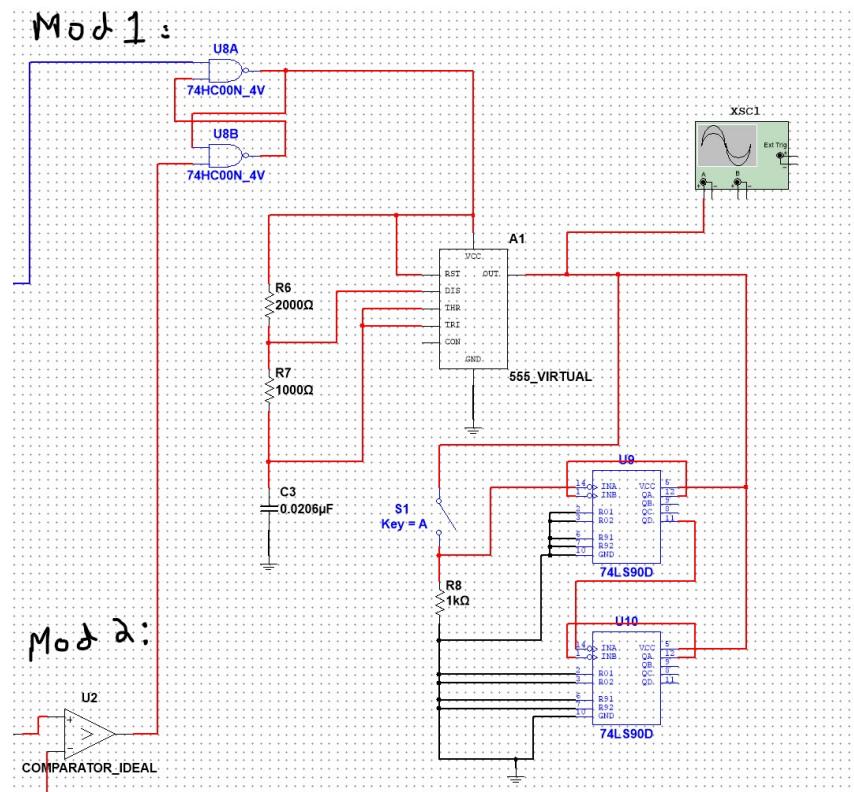
Op-Amp 3:



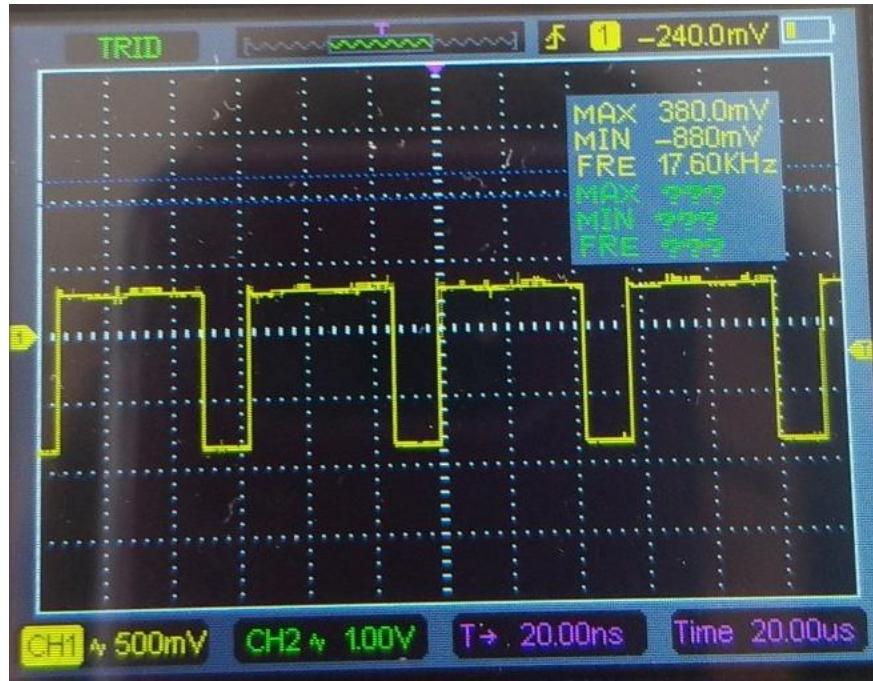
The final comparator output is a square wave as expected but with some considerable noise which was difficult to remove entirely:



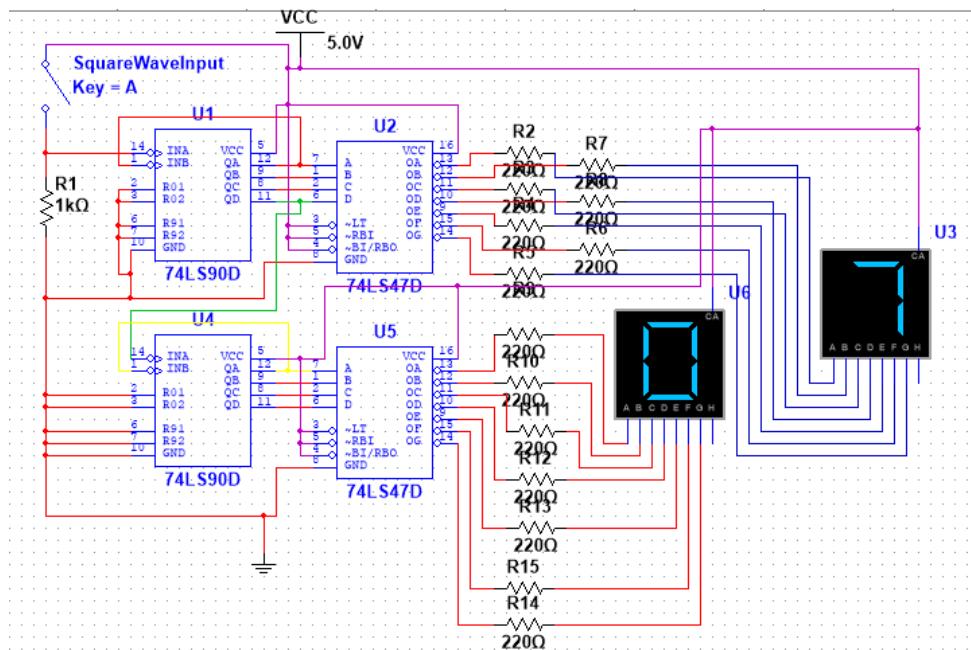
Once obtained, we proceed to module 3 which analyzes the initial input from module 1 along with the comparator output from module 2 and taking advantage of the phase difference between both, uses an SR latch to detect and count the peaks.



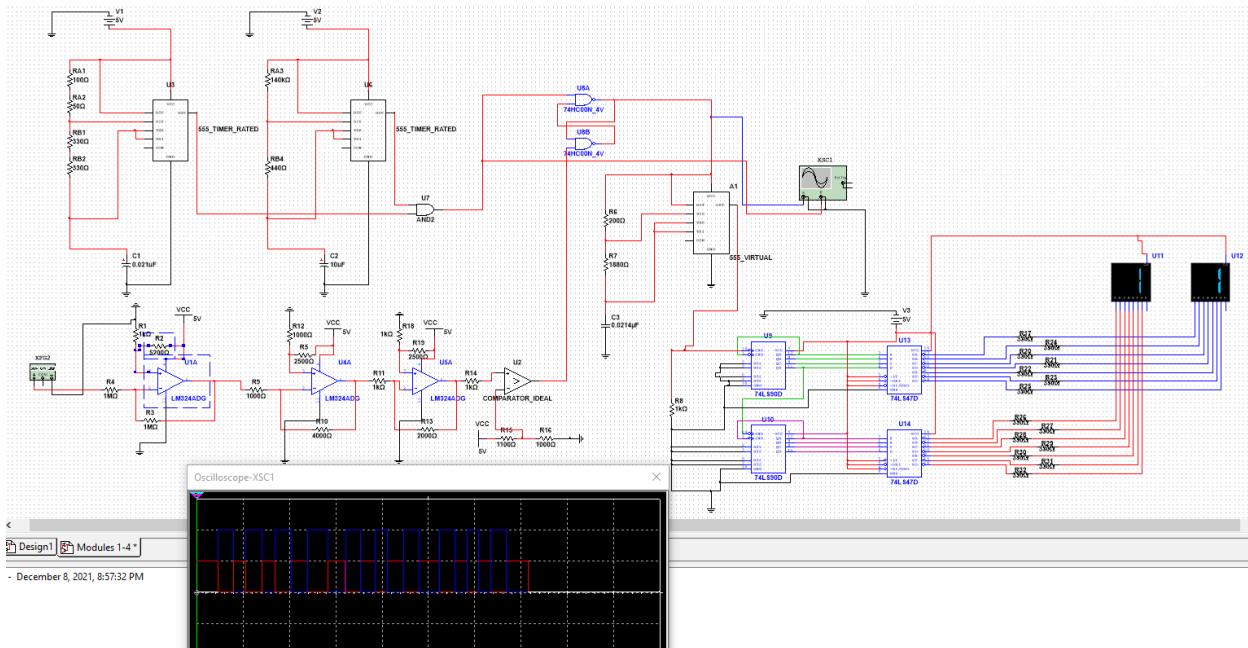
Once counted this signal needs to be converted to an actual distance value which is why the SR latch output goes through a 17 kHz 555 timer. The 555 timer was constructed with $R_1 = 2000 \Omega$, $R_2 = 1000 \Omega$ and $C = 0.0206 \mu\text{F}$. The following output is provided which is deemed acceptable:



The output from module 3 is then finally sent over to the two BCD counters in module 4 which are connected to two 7SD through two individual 7 segment decoders:



The BCD counters used in multisim differed from the ones we owned. Instead of the 74LS90D, we used the CD4510BE where we connected the clock to both the ‘ones’ and ‘tens’ BCD with the carry out of the ‘ones’ fixed into the carry in of the ‘tens’. In the multisim design however, the presets, ground and reset were all set to ground. In order to make the counters reset and keep it constantly updating, a cycling reset was added by attaching the 1 Hz timer from module 1 through a 0.7 V forward biased diode to prevent noise from passing through. The end of the diode and reset pins were grounded keeping it from resetting from anything else besides the pulse from our 1 Hz input which is higher than the threshold 0.7 V. This fulfills the high and periodically resets the BCD’s. In our multisim simulation and a basic build with an arbitrary square wave as the input, the counter would successfully count upwards and display incrementing numbers on the displays until the reset:

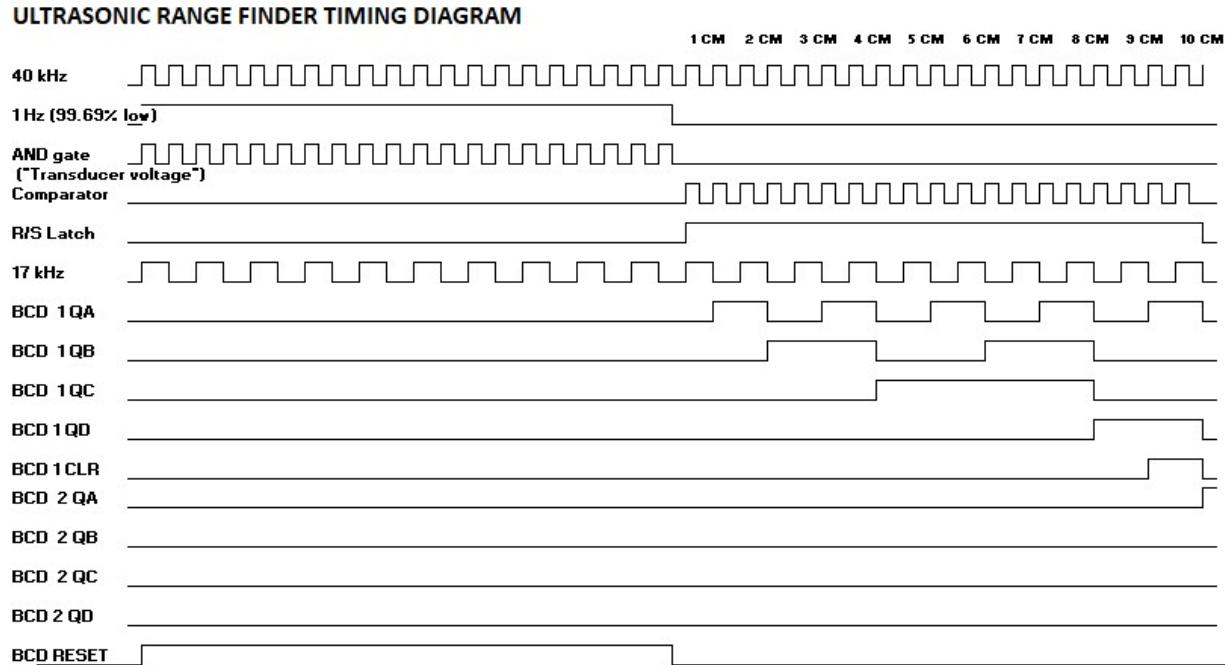


Unfortunately however, when put together with all the other modules, there was just too much noise in the signals for the BCD counter to read successfully although the forward biased diode did reduce noise to prevent unnecessary resetting. Resistors were also placed in front of the BCD VCC input and 7SD VCC input to prevent overheating of the chips and LED burnout.

Our BCD output to 7 Segment decoder input are the following

Q1	D0	000X
Q2	D1	00X0
Q3	D2	0X00
Q4	D3	X000

The updated theoretical timing diagram is as follows:



According to the timing diagram above, the device should work accordingly but unfortunately the biggest obstacle we faced was excessive noise and even after multiple modifications we were unable to reduce it to insignificant levels.

Conclusion and Takeaways

The Ultrasonic Range Finder project demonstrated both the promise and the challenges of constructing a fully discrete, microcontroller-free distance-sensing system. By dividing the design into four distinct modules; (1) generating a clean 40 kHz transmit burst using cascaded 555-timer stages, (2) amplifying and band-pass filtering the weak echo with multi-stage op-amps and a comparator, (3) counting received cycles via an S/R latch and a secondary 555 timer, and (4) driving dual seven-segment displays with BCD counters, we successfully verified each block in isolation. Module testing showed that individual tasks (signal generation, amplification, digital conversion, and display decoding) functioned as intended on breadboards and in Multisim. However, when all four modules were integrated, persistent electrical noise in the receiver and processing stages prevented the seven-segment displays from reliably updating. In practice, the amplified echo often contained clipping and spurious switching, causing the BCD counters to latch at incorrect values or reset unpredictably. Despite careful component selection (forward-biased diodes on reset lines, resistive V_{CC} limiting on ICs, and phased 1 Hz reset pulses), the accumulated noise floor proved too great to overcome within the discrete-logic framework.

From this experience, several key lessons emerged. First, even minor noise or impedance mismatches in analog front-end stages can cascade into complete system failure when no microcontroller or digital filtering is available to correct spurious triggers. Although the dual-stage inverted amplifier and comparator did convert the echo to a digital waveform, the wideband nature of real-world reflections and environmental interference overwhelmed the latch-and-count logic. Second, implementing a monostable (one-shot) approach rather than a fixed 1 Hz refresh might more effectively isolate valid echo pulses and prevent needless resets. Finally, for future iterations, incorporating a small microcontroller (e.g., an Arduino Nano or PIC) to handle cycle counting and digital filtering could drastically simplify the signal-processing chain and mitigate the noise issues intrinsic to pure discrete logic. In summary, while the project did not yield a fully operational range display, it provided invaluable insight into analog-to-digital conversion, timing considerations, and the importance of signal integrity. With further improvements, particularly tighter analog filtering, controlled impedance layouts, or modest microcontroller assistance, a robust ultrasonic range finder can be realized.