Analog Heart Rate Monitor

Gavin S. Burns

March 2021

1 Overview

In this project we had the challenge of designing and building a functioning heart rate monitor using analog and digital circuitry. Taking what we've learned throughout the term we needed to come up with a design that could fulfill all of the following: amplify a faint heartbeat from a microphone, generate a single signal from a conjoined pair of heartbeats, generate a longer signal for the initial heartbeat, ensure each signal can run a small speaker, and have the entire system work reliably from a range of 60-100 beats per minute.

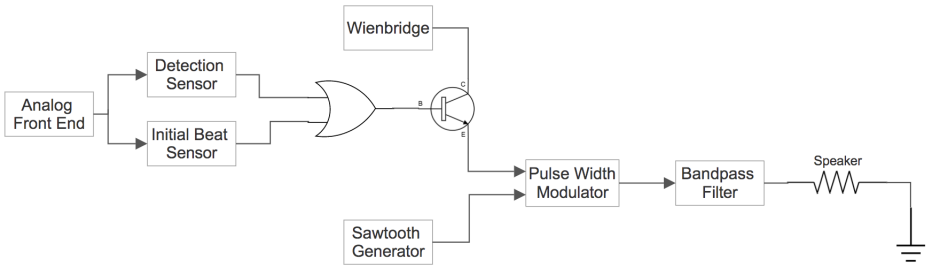


Figure 1: Block Diagram of the System

The figure above represent our solution for the design challenge; each block representing a major aspect of the circuit's design. Beginning with the Analog Front End, this is where we condition the heartbeat signal taken from the microphone. Through the front end we introduce the heartbeat with a microphone biasing circuit, amplify the signal to our desired amplitude using a preamplifier, and filter our any excess noise with a bandpass filter. After the analog front end, we separate the signal to generate our two distinct heartbeat sensors; one to detect an initial heartbeat and to output a long signal, and the second to detect every pair of heartbeats after and to output a short signal. We then rejoin these signals using an OR gate to ensure that the signals do not interfere with each other. Both the Wienbridge and Sawtooth Generator are used to create a sine wave and sawtooth signal respectively, with each of these signals being

used to drive the speaker at the end of the circuit. To keep the speaker from running continuously, we have a NPN transistor placed after the Wienbridge to act as a switch; when one of our sensors is tripped, the switch will close and the sine wave can pass freely to the Pule Width Modulator. Our Pulse Width Modulator is what combines our sine and sawtooth signals into a legible signal for our speaker to read and emit. Lastly, we filter out any unpleasant frequencies using another bandpass filter before delivering our final signal to the speaker.

2 Heartbeat Signal Conditioning

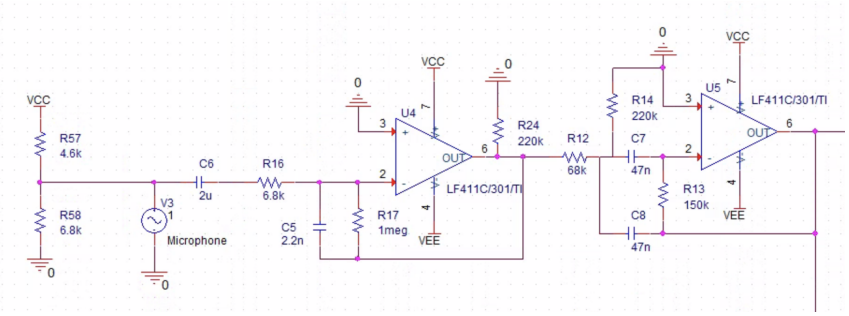


Figure 2: Analog Front End

2.0.1 Microphone Bias

Before shaping our heartbeat signal, we first prepare to receive it with a Microphone Biasing circuit. This DC bias is what drives the microphone and applies a DC offset to the signal. The microphone we used requires a 2.7 volt DC bias (fig. 3), but without a way to supply that DC directly we needed to create an equivalent circuit using our $\pm 4.5V$ rails.

To accomplish this we used the Thévenin Theorem in reverse to find an equivalent circuit. Because Thévenin's Theorem states that any linear, two terminal DC network can be expressed as a single voltage source and a resistor, the reverse must also apply.

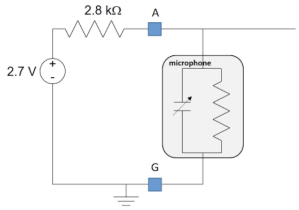


Figure 3: Microphone Bias Circuit

$$R_{th} = \frac{R_x * R_y}{R_x + R_y} \tag{1}$$

$$V_{th} = V_{cc}\frac{R_y}{R_x + R_y} \tag{2}$$

Taking the equations for Thévenin equivalent resistance (R_{th}) and voltage (V_{th}) we can solve for the Thévenin equivalent current which gives us the equation

$$\frac{V_{th}}{R_{th}} = \frac{V_{cc}}{R_x} \tag{3}$$

and with $R_{th} = 2.8k\Omega$, $V_{th} = 2.7V$, and $V_{cc} = 4.5V$ we get that $R_x = 4.\bar{6}k\Omega$. And plugging that back into the first equation we get that $R_y = 7k\Omega$. Sure enough when we apply these values into an equivalent circuit we get a nice operating microphone with steady heartbeat signals. Figures 4 and 5 below depict the output signals of the microphone for the fast and slow heartbeats using the implemented microphone bias circuit.

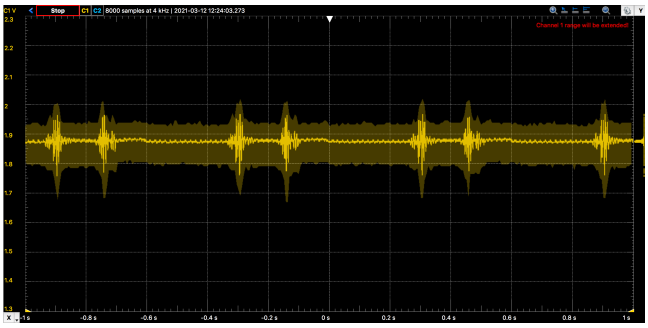


Figure 4: Fast Heartbeat

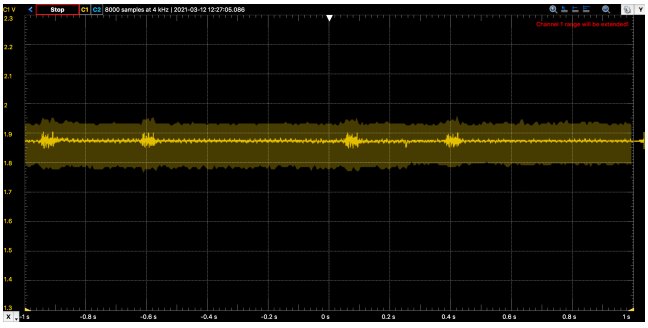


Figure 5: Slow Heartbeat

as we can see these initial signal are both very small and noisy, having them run/signal any other part of the system would be bad idea.

2.0.2 Preamplifier

To help correct the amplitude problem, we fed the signal through a preamp to increase the amplitude of the peaks.

We used a traditional RC active filter (fig. 6) as the chassis for our amplifier to easily identify the poles of our system: $p_1 = \frac{1}{R_1 C_1}$ and $p_2 = \frac{1}{R_2 C_2}$. The heartbeats we receive will have a frequency between 62-440$\frac{rad}{sec}$ so we can set each to their respective pole. We also know that the gain of the amplifier at the center frequency will need to be at least 40dB and will be equal to $20log(\frac{R_2}{R_1})$.

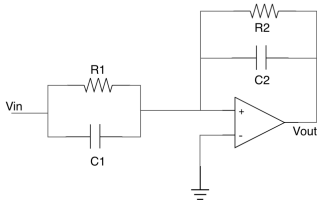


Figure 6: Preamplifier

To begin solving for our values, we developed an R_1 and R_2 that gave us our desired gain at the center frequency ($R_1 = 6.8k\Omega$ and $R_2 = 1.0M\Omega$) we then used the equations of our poles to find our necessary values for C_1 and C_2 ($C_1 = 2\mu f$ and $C_2 = 2.2nf$). Our finalized design had a gain of above 30dB between the frequencies of 10-100Hz.

Below is shown the active effect of the preamp, with the yellow waveform representing the unmodified fast heartbeat and the blue showing the new amplified signal. Because of the series capacitor in our preamp circuit we get the additional benefit of getting rid of the DC offset voltage provided by the Microphone Bias.

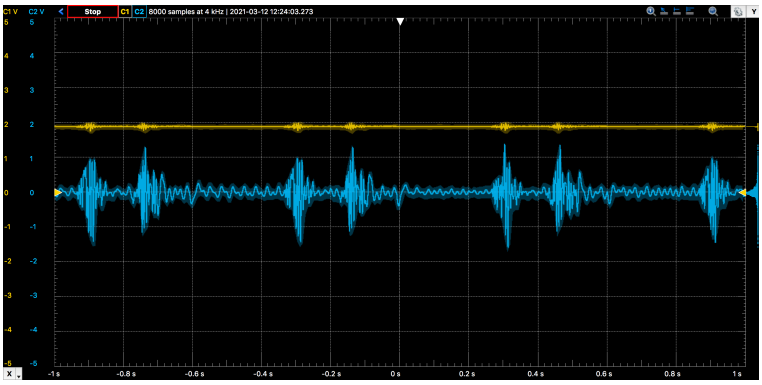


Figure 7: Amplified Fast Heartbeat

2.0.3 Bandpass

The amplified signal is looking nice, but we needed to clean up the noise as much as we can to reduce any possible error down the line. We used an online filter calculator to design a butterworth bandpass filter that could deliver on a smooth passband region. In order to generate a suitable filter we first needed to solve for our filter's center frequency, passband bandwidth, and stop gain.

the equation for center frequency is

$$f_c = \sqrt{f_1 * f_2} \tag{4}$$

where f_1 and f_2 are the poles of our region. With our poles being 10 and 70Hz we get that the center frequency is 26.46Hz.

The passband bandwidth is simply the difference between our poles which gives us $f_{bw} = 60Hz$ and our stop gain is the expected gain at some region outside of our band pass. So the choice we went with is -20dB at 700Hz. We know this is true because we are designing a first order filter and outside the passband region there is a slope of -20dB per decade; so if we start with 0dB at our pole of 70Hz then one decade late, at 700Hz we should expect -20dB.

Plugging in these values nets us the bandpass filter

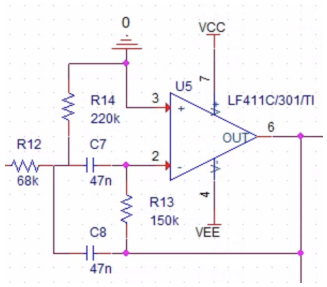


Figure 8: Bandpass Filter Schematic

In implementation we got the final bode plot shown below, this is the magnitude in dB for the combination of both the preamp and the bandpass filter. First iterations of testing with the heartbeat signals showed that our original amplification (roughly 20dB at the center frequency) wouldn't be enough to get a solid signal; we quickly increased the gain from the preamp until we got a near perfect amplitude that would suit the rest of the circuit.

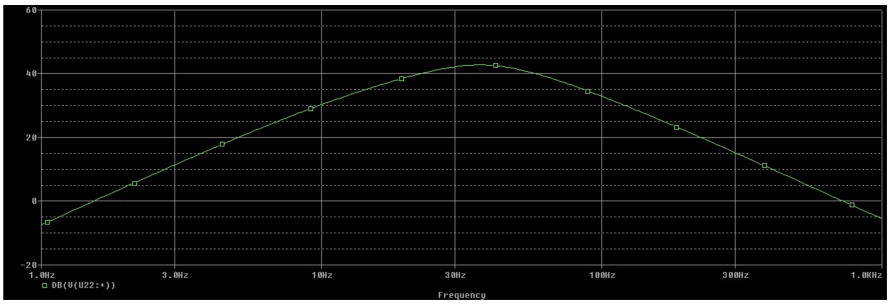


Figure 9: Bode Plot

In all, for all the parts used in this block include:

- LF412 opamp x1 ($1.34)
- 1μf capacitor x2 ($0.43)
- 47nf capacitor x2 ($0.23)
- 2.2nf capacitor x1 ($0.25)
- Assorted Resistors x8

This gives us a total price for building this block equal to $2.25

3 Detection Sensor

With the heartbeat signal at a desirable amplitude and excess wavelengths attenuated we can now focus on the actual detection of the heartbeats. In this project there are two specific instances of triggers that we are looking to detect; the initial heartbeat to confirm that the monitor is operational, and every subsequent heartbeat after that. To differentiate these sensors, we have differing tone lengths to represent the initial beat (long) and the repeated beats (short). But in order to have separate tones for separate triggers we'll need to build separate detection mechanism.

Both of these circuits are very similar in their design with only minor differences in component values. Both systems begin with a peak detector which extends the peak duration of the heartbeat. We then have a comparator to translate the extended signal into a high/low square wave which prepares it for

the digital logic. These square waves are then fed through some digital gates and output is a single edge trigger symbol that will detect only the crucial heartbeats. This last signal is finally sent to a Monostable 555 Timer which, when triggered, will release a high signal for a specified amount of time. When the 555 Timer is high, it signals to the rest of the heart monitor system to play an audible tone form our speaker. So having a steady and consistent detection sensor is paramount for the accuracy of the rest of the project.

3.1 Repeated Signal

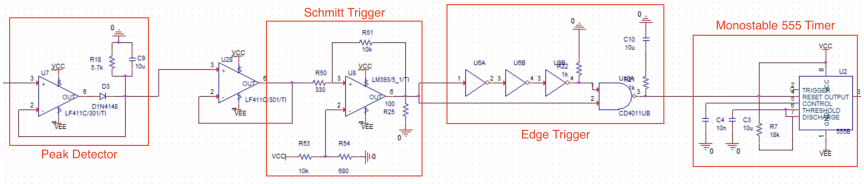


Figure 10: Heartbeat Sensor

3.1.1 Peak Detector

To start with the repeated signal, the peak detector is the most important part in detecting the heartbeats themselves. When a beat is sent through the diode becomes forward biased and the voltage flows through freely while also charging local capacitor ($V_{in} = V_{out}$). But when the beat goes away the detector doesn't immediately discharge; because $V_{in} < V_{out}$ now the diode becomes reverse biased and the capacitor now discharges at a rate of decay equal to

$$V_{out} = V_p e^{-(\frac{t}{RC})} \tag{5}$$

Where V_p is the peak voltage of the signal before it begins to decrease.

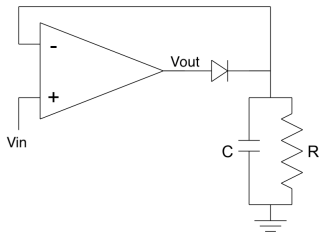


Figure 11: Peak Detector Template

Now the heartbeats we are working with are comprised of two "lub-dub" signals, with the peaks of the lub and dub being separated by about 160ms and the separation between each lub-dub being at least 450ms.

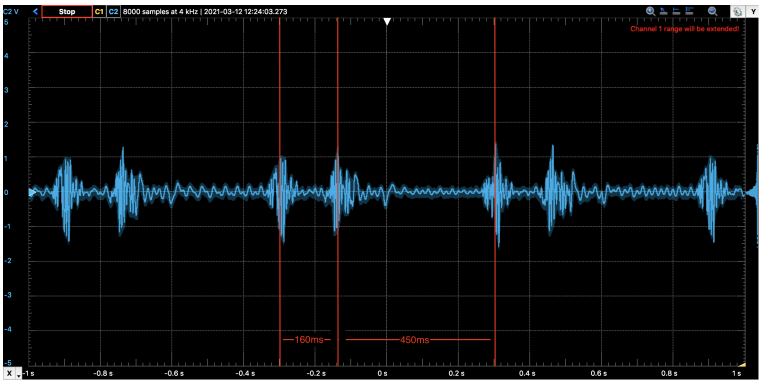


Figure 12: Fast Heartbeat Measurements

For this peak detector we are looking for a quick discharge so that we can easily separate each part of the signal while soaking up all the less important noise. Specifically, we decided that we wanted the output of the peak detector to be at $\frac{1}{2}V_p$ after only 40ms. Using these numbers we can plug these values into the above equation and get that

$$RC = .0577 \tag{6}$$

and with the resistor and capacitor values we had on hand, we chose $R = 5.7k\Omega$ (comprised of 1k and 4.7k resistors) and $C = 10\mu f$.

Sure enough, with the values we derived we get the waveform below that exhibits a quick discharge after reach the peaks.

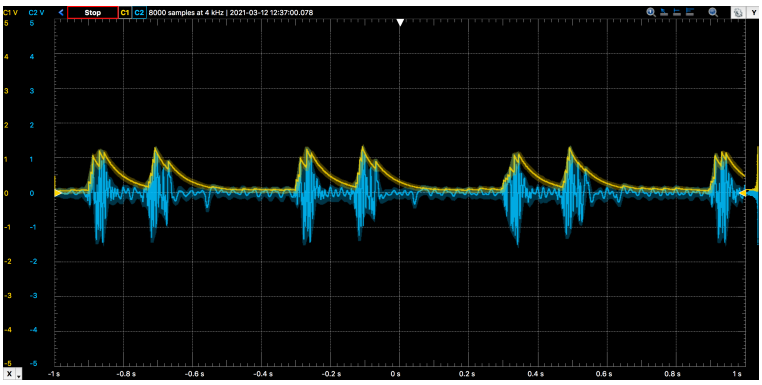


Figure 13: Repeated Peak Detector Signal

3.1.2 Schmitt Trigger

Using the output of the peak detector we can now turn our "lub-dubs" into proper square waves for the logic gates. To do this, we utilized a comparator

deliver a high voltage whenever the peak detector is above a certain threshold.

But before we integrate the comparator we need to put a unity buffer between the it and the peak detector. If we were to connect them directly together, then both components would have resistors in parallel with each other. This would cause problems with both the discharge times of our peak detector and the threshold values for our Schmitt Trigger. By placing a negative feedback opamp after the peak detector (with the signal feeding into the $V+$ terminal) we separate the compnents from interfering with each other while preserving the signal.

Our original design called for just a comparator that went high while the peak detector was above 300mV but the fast rate of discharge left some short glitches on the edges that took away from its reliability. To remedy this problem we adapted the comparator into a Schmitt Trigger that would have an upper threshold of 450mV and a lower threshold of 150mV. This way the comparator would only shoot high when the peak detector was above 450mV and would only shoot low when it fell past 150mV.

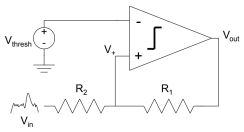


Figure 14: Schmitt Trigger Template

To determine the values for R1 and R2 we can use the Schmitt Trigger equations

$$0.45V = V_{thresh}\frac{R_1 + R_2}{R_1} + 4.5\frac{R_2}{R_1} \tag{7}$$

$$0.15V = V_{thresh}\frac{R_1 + R_2}{R_1} - 4.5\frac{R_2}{R_1} \tag{8}$$

and subtract them from each other to get the ratio for R_2 and R_1

$$0.3V = \frac{R_2}{R_1} \tag{9}$$

Finally we decided on the values of $R_1 = 10k\Omega$ and $R_2 = 330\Omega$ to act as the resistors for our Schmitt Trigger.

The last part of the Schmitt trigger is determining the threshold hold voltage; this can be done with a simple voltage divider. We decided the threshold should be 680mV so using the voltage divider rule we got that our resistor values should be $10k\Omega$ and 680Ω (With the output voltage flowing across the 680Ω resistor.

To get the schmitt trigger fully operational we'll also need to place a pull up resistor between the comparator output and V_{cc}, this is to ensure that the comparator has enough current to pull up the output voltage to the 4.5 power

rail. All together the Schmitt Trigger works beautifully. As we can see below when a heart beat comes through the sensor the Schmitt Trigger goes high and after the peak detector dips below 150mV it drops low again.

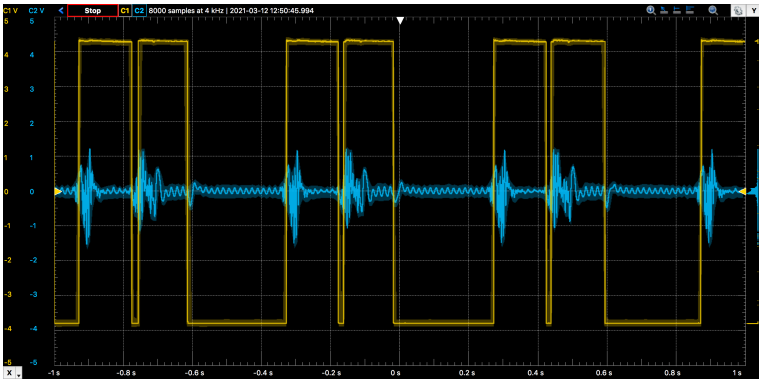


Figure 15: Schmitt Trigger Output

3.1.3 Edge Detector

With the Schmitt Trigger working we are still having the problem of detecting both the lub and dub pulses. In order to correct this we implemented a D Flip-Flop. This chip can be set up to only trigger when it detects the rising edge of an input. So when the Schmitt Trigger rises from low to high on the "lub" pulse, then the D flip-Flop will begin outputting high. When the Schmitt trigger drops from high to low the D Flip-Flop won't react. But when the Schmitt Trigger rises again on the "dub" pulse the D Flip-Flop will fall back down to low; thus turning our double "lub dub" pulses into a single pulses that can represent both.

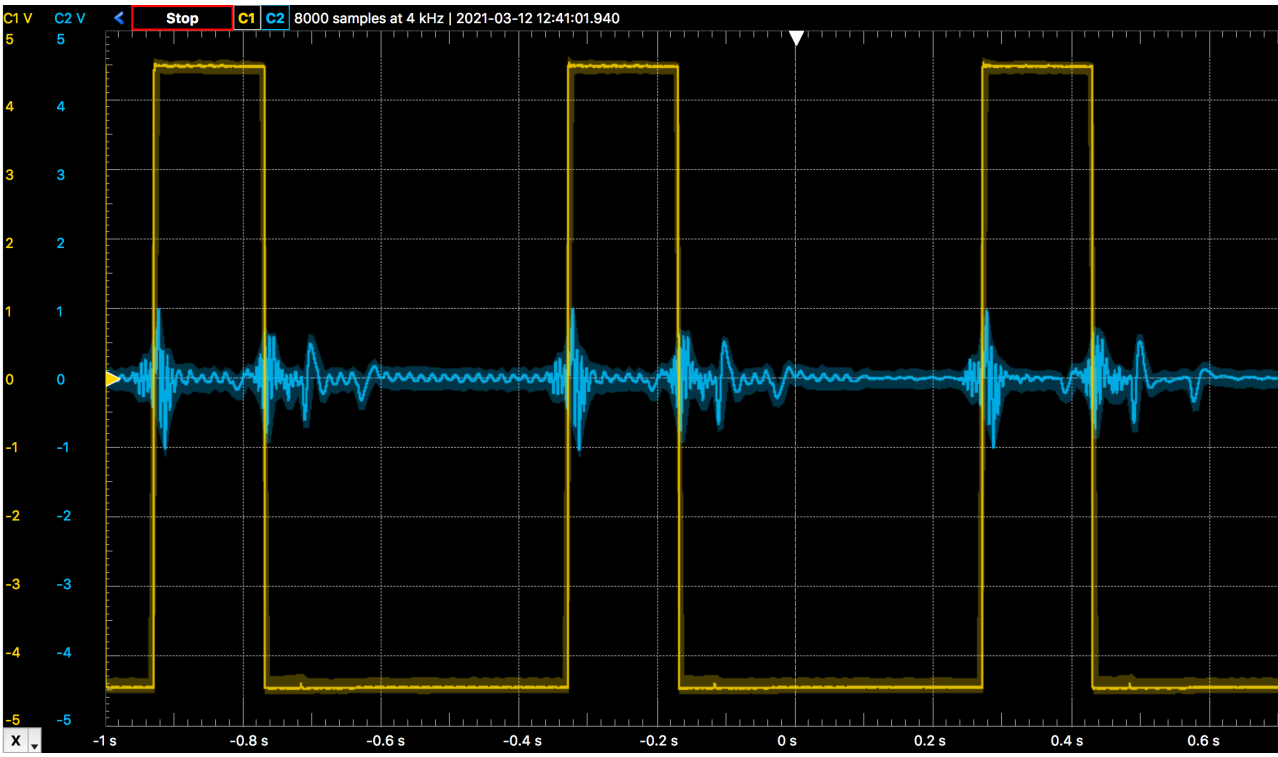


Figure 16: D Flip Flop signal

In order to set up the D Flip Flop, we first need to feed the output of the Schmitt Trigger into the clock input of the Flip Flop (bottom left triangle). This is what the D Flip Flop reads and uses to change output values. We'll also need to hook up the input terminal D to $\bar{Q}$ pin. This provides a feedback loop for the signal to remember its previous state. As mentioned above, altogether the D Flip Flop is set up to cut the frequency of the Schmitt trigger square wave in half. Since it will only shift from low to high/high to low on the positive edge of the Schmitt trigger signal. the output wave from node Q will be the sole square wave depicted in figure 16

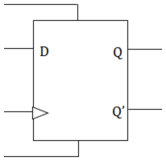
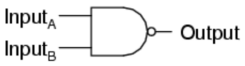


Figure 17: D Flip Flop Template

Now that we have a single signal representing a single heartbeat there is

still one more problem to attend to before sending it into the 555 Timer. The problem is with the duration of the high state of the D Flip-Flop; we mentioned previously that the separation between the lub and dub peaks is about 160ms at the shortest, but its also 300ms at the longest. This means that the duration of the D Flip-flop signal is equal to 160-300ms. Now this is a serious problem because our monostable 555 Timer is only suppose to be on for 200ms, If we feed it a signal that lasts longer than its on time then we run into problems with its operation. So our last task before the 555 Timer is to detect only the rising edge of the D Flip-Flop.

I order to accomplish this we abused a side effect of logic gates known as propagation delay. For example when you feed a signal through an inverter it actually lags by a fraction of a second. Now if we separate a signal into two exact copies and lag and invert one of them then both signals will be always opposite except for at the rising and falling edges. If we then send both signals through a NAND gate the the output will only be negative when both signal are positive as show in the truth table below; this occurs only at the rising edge of the delayed signal



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

Figure 18: NAND Gate Truthtable

In testing we found that the propagation delay of putting one of the signals through three inverters wasn't enough to get a well defined edge signal. So in order to increase the delay we added a $10\mu f$ capacitor. This capacitor increases internal RC decay of the inverter which is the reason propagation delay occurs. As we can see in the below figure, we not have beautiful edge trigger which outputs low for only a brief peroid of time at the rising edge of our D Flip-Flop

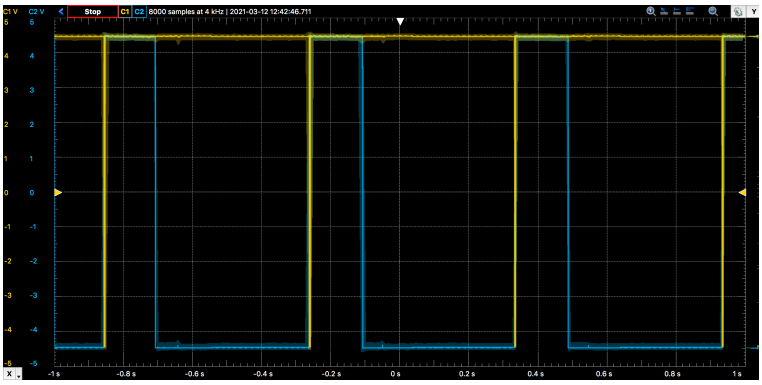


Figure 19: D Flip-Flop

3.1.4 555 Timer

The last part of our Repeated Detection Sensor is the 555 Timer which should turn on whenever a pulse is detected. We have the 555 Timer set up in the monostable configuration so that when triggered it release a single 200ms high pulse. This pulse is generated by scharge of a capacitor over a time until it rises above the threshold of the 555 Timer ($\frac{2}{3}V_{cc}$). To model the time it takes to charge up the capacitor we can use the equation

$$V_{out} = \frac{1}{3}e^{(\frac{t}{RC})} \tag{10}$$

and solve for R and C to get

$$RC * ln(3) = 200ms \tag{11}$$

With this equation we can solve for R and C values that we have at our disposal; we decided on $R = 18k\Omega$ and $C = 10\mu f$ which should give us a duration of roughly 198ms. Putting this all together, the output of the Monostable 555 Timer shown below begins at the first "lub" signal (when the edge detector triggers) and lasts for just about the 200ms we expect.

This final output of the 555 Timer is the culmination of the detection sensor for the repeated beeps. When a heartbeat is sent through it is detected, turned into a square wave, conditioned, and output as a monostable signal which will later trigger a switch to turn on our speaker.

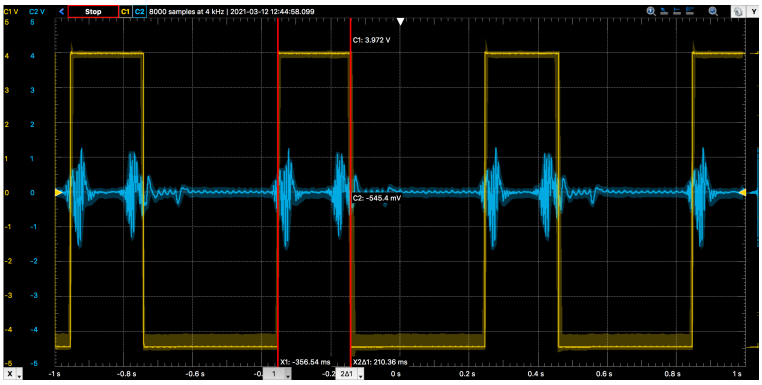


Figure 20: 555 Timer Output

In all, for all the parts used in this block include:

- LF411 opamp x2 ($1.41)
- LM393 Comparator x1 ($0.42)
- CD40174B D Flip-Flop x1 ($0.49)
- CD4069UB Inverter x1 ($0.43)
- CD4011UB NAND x1 ($0.43)
- LM555 Timer x1 ($1.06)
- 1N4148 Diode x1 ($0.11)
- 10μf capacitor x3 ($0.84)
- 10nf capacitor x1 ($0.23)
- Assorted Resistors x7

This gives us a total cost for this block equal to $6.83

3.2 Initial Signal

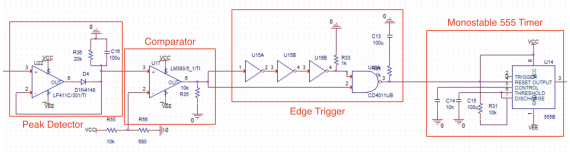


Figure 21: Initial Heartbeat Sensor

Having the speaker turn on for every heartbeat is wonderful and one of the most important parts of the entire project, but we cant forget about the second trigger for our speaker. When trying to come up with a solution for an initial signal we thought it best to model it after our already function repeated signal detector; the base function is fundamentally the same, we would only need to alter some values around with the peak detector and Monostable 555 Timer.

3.2.1 Peak Detector

With this initial signal peak detector, instead of having a quickly discharging peak we want to extend the duration of the signal around the peak region; each instance of heartbeat should keep the peak detector signal "afloat". This means that the only time the signal will rise from zero to a peak is either at the initial heartbeat or when heartbeats stop for an extended duration and the start up again.

To achieve this we can induce an extremely long discharge time using the same equation for the previous peak detector. This time we shot for a time around 1.4 seconds, giving us the equation

$$RC = 2.02 \tag{12}$$

In turn, we decided on the value of $R = 20k\Omega$ and $C = 100\mu f$, giving us a decay rate of 1.38 seconds to reach $\frac{1}{2}V_p$.

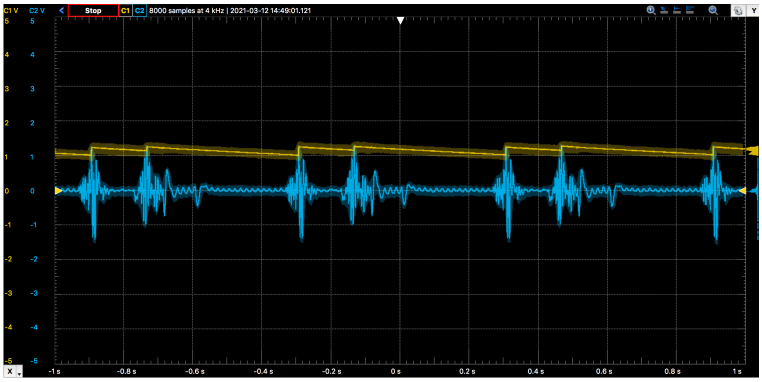


Figure 22: Initial Peak Detector Signal

3.2.2 Comparator

Because of the slow discharge time of the peak detector we have no need to implement a Schmitt Trigger into the initial signal sensor. Instead, we can keep the original comparator to turn our peak signal into a square wave. For our threshold voltage we used the same 0.3V as the Schmitt Trigger. Using the voltage divider formula we get

$$0.3V = 4.5\frac{R_2}{R_1 + R_2} \tag{13}$$

or $R_1 = 10k\Omega$ and $R_2 = 680\Omega$. Just as we should expect when we see the same threshold resistor values for the Schmitt Trigger. Just as with the previous comparator, we'll have to add a pullup resistor at the output of this one. This it ensure that the comparator has enough current to pull the signal high for as long as the peak detector remain high.

3.2.3 Edge Detector

The edge detector was left exactly the same as the repeated sensor and since the long discharge time forgoes the separate "lub dub" signal we actually had no need for a D Flip-Flop in this sensor.

As we can see depicted in the figure below, the edge trigger only falls low when the first heartbeat is detected, it then remains high in the face of all the other heartbeats that come after it.

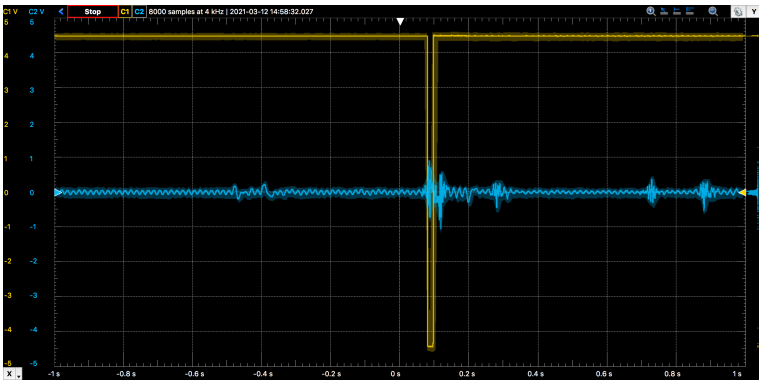


Figure 23: Initial Edge Detector

3.2.4 555 Timer

To differentiate this signal from the typical heartbeat signal we will have this signal played out for a much longer time when triggered. This means that we'll be using the same Monostable 555 Timer configuration as before with a much larger charge up time.

Using the previous Monostable 555 Timer equation and a desired high time of roughly 1 second we get the equation

$$RC * ln(3) = 1s \tag{14}$$

Which gives us suitable R and C values of $10k\Omega$ and $100\mu f$ respectivley.

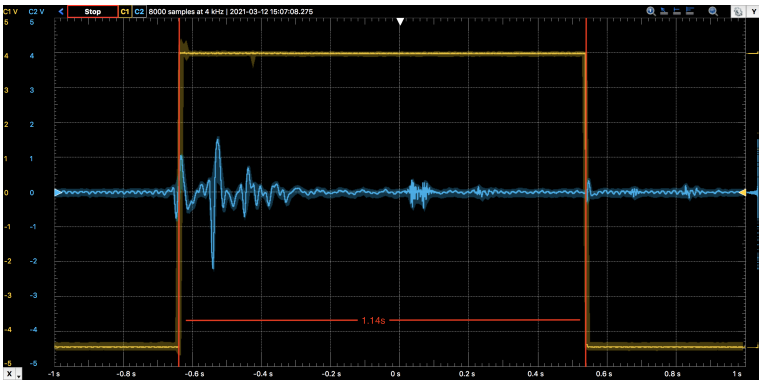
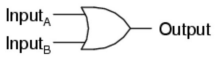


Figure 24: Initial 555 Timer Output

The final output of the initial detector is shown above and follows closely with the repeated detector's output signal. What should be noted is the extended run time of about 1.1 seconds and the single pulse that begins with the introduction of the initial heartbeat and does not trigger for any other beats.

3.3 Combining Sensors

With each of our detector working we now need a way to combine both signals with one interfering with another. Simply enough this is done with an OR gate; according to the truthtable, if either input is high then the output will also be high.



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Figure 25: OR Gate Truthtable

In all, for all the parts used in this block include:

- LF411 opamp x1 ($1.41)
- LM393 Comparator x1 ($0.84)
- CD4001B NOR x1 ($0.43)
- CD4069UB Inverter x1 ($0.43)
- CD4011UB NAND x1 ($0.43)

- LM555 Timer x1 ($1.06)
- 1N4148 Diode x1 ($0.11)
- 100μf capacitor x3 ($0.85)
- 10nf capacitor x3 ($0.22)
- Assorted Resistors x 4

This gives us a final price for this block equal to $5.78

4 Wienbridge Oscillator

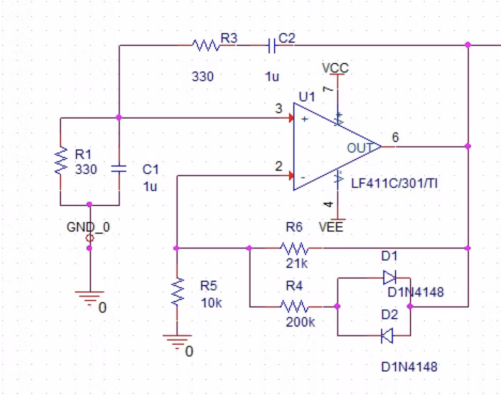


Figure 26: Wienbridge Oscillator Circuit

The goal of our Wienbridge oscillator is to generate a 450Hz sine wave that can be used in conjunction with our Sawtooth Generator to develop a pulse width modulating signal that, in turn, can be fed into our speaker. Traditionally, a simple function generator could be used to make any necessary sin or sawtooth waves we'll need, but that can be costly and doesn't suit the portable nature of our heart rate monitor. Instead we used a Wienbridge Oscillator to produce our 450Hz sine wave.

The Wienbridge is comprised of two main parts, an oscillating RC circuit and a nonlinear amplifier. The RC circuit is what generates a sinusoidal wave with a frequency of $\frac{1}{RC}$, but overtime this wave will die out do to natural dampening. To counteract this, we have the nonlinear amplifier which will actively adjust the amplitude of the sine wave when it goes too low or high.

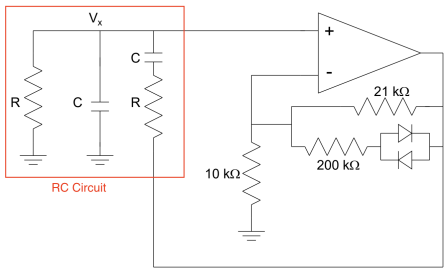


Figure 27: Weinbridge Diagram

As we can see, when the amplitude of the sine wave increase the diodes become forward biased and the gain of the circuit is equal to

$$\frac{V_{out}}{V_{in}} = \frac{(200k\Omega||21k\Omega) + 10k\Omega}{10k\Omega} = 2.9 \tag{15}$$

This keeps the dampening term positive and the amplitude of the sine wave slowly reduces. Conversely, when the amplitude gets too small and the diodes become reversed biased the gain of the circuit becomes

$$\frac{V_{out}}{V_{in}} = \frac{(21k\Omega) + 10k\Omega}{10k\Omega} = 3.1 \tag{16}$$

which, in turn, pumps up the amplitude. This slight push and pull from the nonlinear amplifier keeps the peak to peak amplitude of the wave nearly constant at $\frac{3}{2}V_{th}$ where V_{th} is the voltage drop across the diode. For the diodes we are using $V_{th} = 1$ so it is expected that we have a $V_{pp} = 1.5V$.

For choosing our values for R and C we simply convert our desired frequency of 450Hz to $2827\frac{rad}{sec}$,set it equal to $\frac{1}{RC}$ and find closely appropriate values. For our circuit we chose $R = 330\Omega$ and $C = 1\mu f$ this should net us a wavelength of roughly 482Hz; and looking at the output wave of our Wienbridge Oscillator below. we see that it has a period of 2.16ms or a frequency of 463Hz.

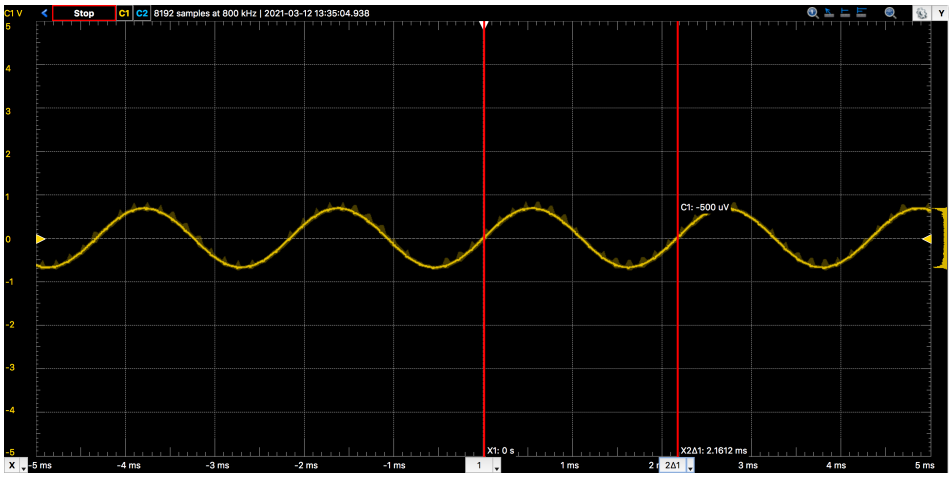


Figure 28: Output Sine Wave

Now this sine wave is running continuously for as long as voltage is being supplied to the power supplies of the amplifier. But we only want it to interact with the rest of the circuit when either the initial or repeated signal is active. For this, we'll use an NPN Transistor as a switch which will only be closed when the output of Detection Sensor is high and the transistor itself is in saturation.

To show this, lets assume that sensors are detecting a pulse and are outputting 4.5V, this is being lead through a $10k\Omega$ resistor into the base of the transistor to ensure that it is supplying safe amounts of current. For the transistor we're using the minimum voltage needed to close the switch is 0.65V (which is easily surpassed by the 4.5V). This also means that the current running through to the base is equal to

$$I_b = \frac{4.5V - 0.65V}{10k\Omega} = 0.385mA \tag{17}$$

With a β value of 100 that puts $I_c = 38.5mA$. This means that the voltage drop across the $1k\Omega$ resistor on the collector side is 38.5V, but since the max voltage supplied by the Wienbridge is .75V then the entire transistor must be in saturation mode when the sensors are high. And when the sensors are low, the voltage is -4.5V and there isn't enough voltage to close the switch.

The figures below depict the effect of having the NPN initially off and then immediately turned on. Each yellow "block" in the first figure is the time at which the detection sensor is high and the NPN switch is closed; the bottom figure provides a zoomed in copy of each yellow block as the NPN becomes closed.

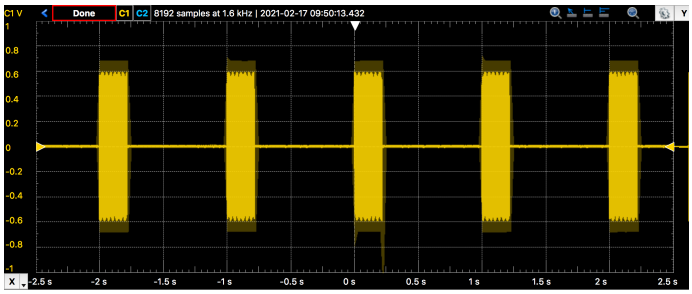


Figure 29: Triggering Sine Wave In Tine With 555 Output

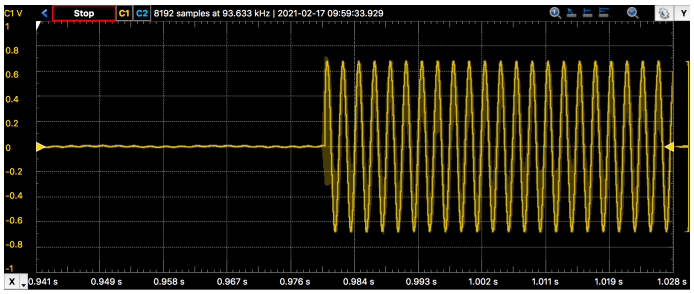


Figure 30: Triggered Sine Wave Close Up

In all, for all the parts used in this block include:

- LF411 opamp x1 ($1.41)
- 1N4148 Diode x2 ($0.11)
- 2N3904 NPN Transistor x1 ($0.18)
- 1μf capacitor x2 ($0.43)

This gives us a total cost for this block equal to $2.24

5 Sawtooth Generator

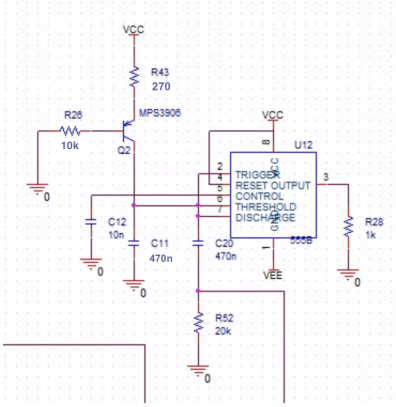


Figure 31: Sawtooth Generator Circuit

In the design, our sawtooth generator should work by having a 555 time charge up a capacitor and then immediately discharge it when it reaches a certain threshold. To accomplish this we have a PNP transistor working as a current source which feeds into a 555 timer and capacitor, over time the constant current flow will charge up the capacitor until reaches the threshold of the 555 timer at which point the discharge switch will close and the capacitor will discharge down to the lower threshold of the 555 timer.

Starting with the PNP transistor, we have this transistor solidified in the forward active region to keep it as a constant current source for our 555 timer.

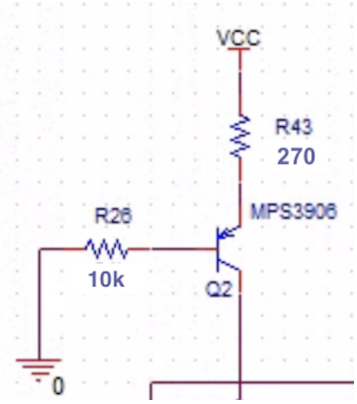


Figure 32: PNP Transistor

Looking at our PNP Transistor datasheet (2N3906), we see that our transistor has a base to emitter saturation voltage (V_{be}) of 0.65V and a $\beta = 100$ when

I_c is at least 10mA. Using MoHat analysis we can say that

$$V_{control} = \frac{V_{cc} - V_{be} - I_b * (\beta + 1) * 270\Omega}{10k\Omega} \tag{18}$$

substituting in our values for β, V_{be}, V_{cc}, and $V_{control}$ we get

$$0V = \frac{4.5V - 0.65V - I_b * 101 * 270\Omega}{10k\Omega} \tag{19}$$

which means $I_b = 141\mu A$ and $I_e = 101 * I_b = 14.2mA$ (exactly what we expect). This also means that the transistor is in the forward active region and is supplying a constant 14.1mA to the sawtooth generator.

Now the frequency at which our Sawtooth wave oscillates is determined by the size of our capacitors. According to the equation

$$C\frac{dV}{dt} = I \tag{20}$$

we know that the current running across a capacitor is equal to its capacitance times the change of voltage over it, but by rearranging the equation we can also see that the Voltage rate of change is equal to

$$\frac{dV}{dt} = \frac{I}{C} \tag{21}$$

For our 555 Timer, the capacitor will charge up until it reach $\frac{2}{3}V_{cc}$ and then discharge down to $\frac{1}{3}V_{cc}$. With a $V_{cc} = 4.5V$, this means that our wave should have an amplitude of 1.5V which we can use substitute to for the change in Voltage (dV). In order to find our frequency all we need to know is the capacitance of our capacitor (the current being supplied is the even 14.1mA calculated earlier). If you look at the schematic we implemented in our project you can see that there are actually two separate $470nf$ capacitors in parallel together heading to ground. So that means that their combined capacitance is equal to $940nf$. In all we have a $940nf$ capacitor, 1.5V change in voltage, and 14.1mA of current; plugging these numbers into our equation and we see that

$$dt = \frac{C}{I} * dV = 100\mu s \tag{22}$$

which is right around our measured period for our sawtooth waves we can see in the figure below. Another aspect of the sawtooth that should be kept in mind is its frequency, ideally we want it to be many times greater than the frequency of the sine wave to make sure that it is easier to filter out later down the line. For our Sawtooth, with a period of $146.16\mu s$ we get a frequency of 6842Hz; which, when compared to the 450Hz sine wave, is about 15 times larger.

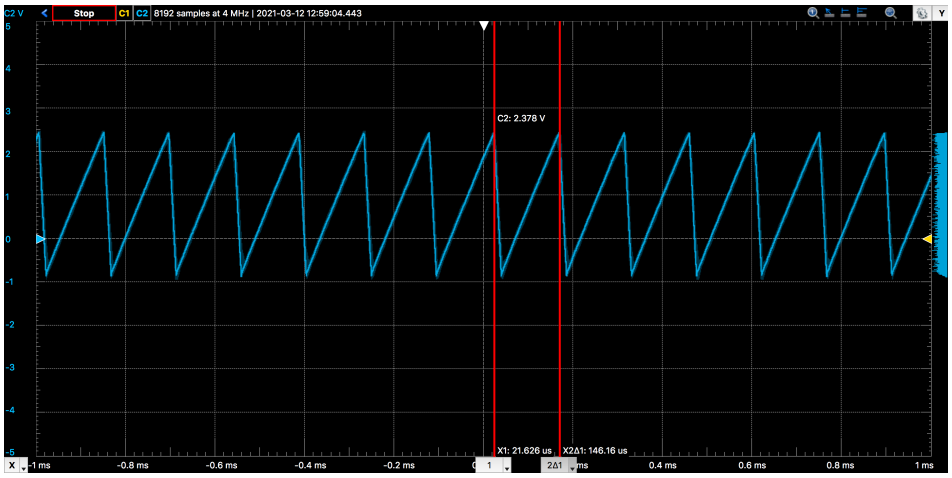


Figure 33: Sawtooth Wave

In all, for all the parts used in this block include:

- 2N3906 PNP Transistor x1 ($0.18)
- LM555 Timer x1 ($1.06)
- 10nf capacitor x1 ($0.23)
- 470nf capacitor x2 ($0.32)
- Assorted resistors x4

This gives us a total cost of this block equal to $1.79

6 Pulse Width Modulator

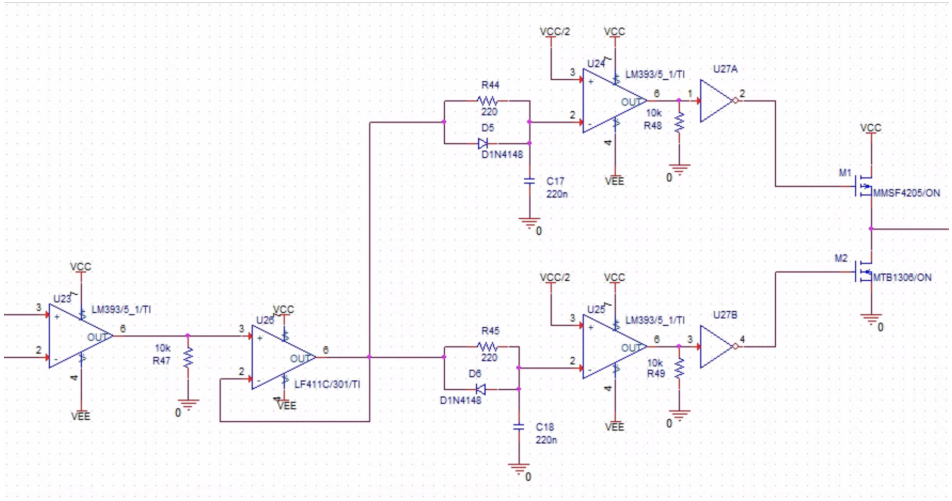


Figure 34: sawtooth Wave

6.1 Pulse Width Signal

With our Sawtooth and Sine waves generated, we can now create our pulse width modulator signal using a comparator and break before make circuit. We first feed the Sawtooth wave into the positive terminal of the comparator and the sine wave into the negative terminal. With the input signals overlaid in the figure below we can see how we expect them to interact and what out put signal we expect.

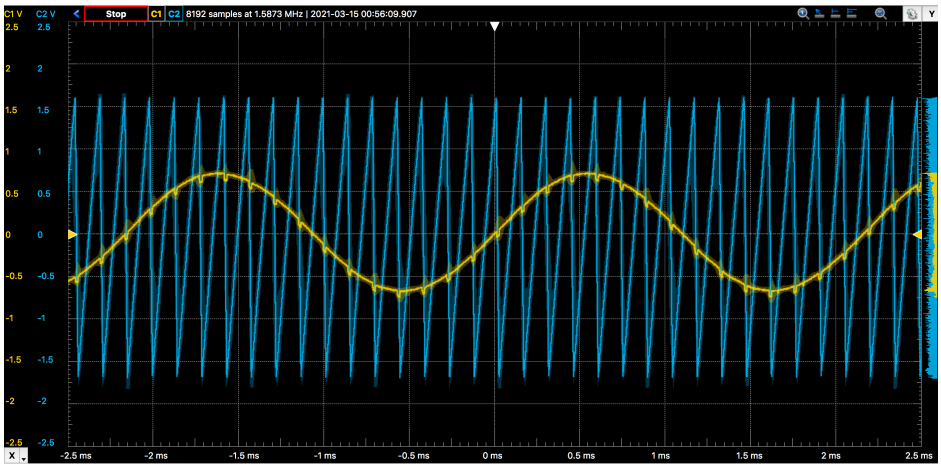


Figure 35: Sawtooth and Sine Wave

Whenever the sawtooth wave is greater than the sine wave we should expect to see a high output from the comparator; conversely, when the sine wave is greater than the sawtooth the there should be a low out put. Furthermore as the Sine wave rises, the amount of time that the sawtooth is greater than the sine wave decrease. And this time increases as the sine wave falls. Overall we should expect that the out should be a square wave with a variable pulse width depending on if the sine wave is high or low, and sure enough, looking at the output waveform below that is exactly what we get.

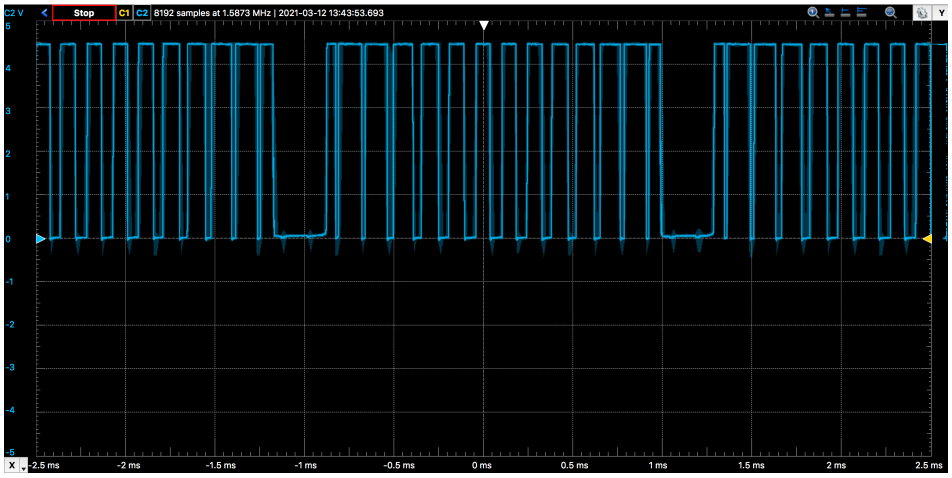


Figure 36: Pulse Width Signal

Notice that the areas of long low time correspond to times when the sine wave is larger than the sawtooth wave for a larger instance of time, and as the

sine wave cycles, so too does the pulse width of the modulating signal.

6.2 Break Before Make

We'll now need to feed this waveform through a Break-Before-Make (BBM) Circuit to prevent our Class D Amplifier from blowing up. What the BBM circuit does is cycle the on states of the PMOS and NMOS components of the D Class Amplifier to ensure that both aren't on at the same time.

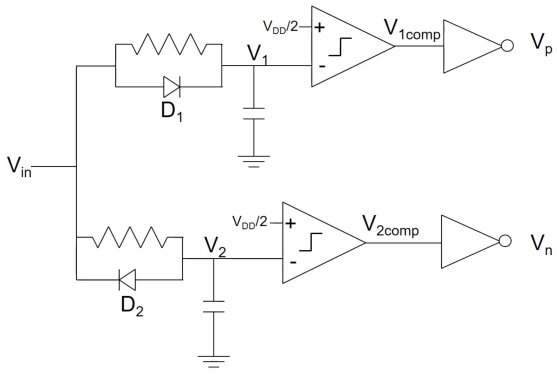


Figure 37: Break Before Make Template

In the figure above we can see that the circuit is comprised of two comparators each connected to an inverter, diode, and RC delay. If we imagine that V_{in} is positive then D_1 is forward biased and D_2 is reversed biased; this effectively lets V_1 bypass the RC Delay freely while V_2 is slightly buffered because of it. Now when we consider V_{in} as negative we get the exact opposite effect, V_1 becomes buffered by the delay and V_2 can follow V_in.

Putting it altogether, lets assume that V_{in} is a simple square wave. At the rising edge of V_{in}, V_1 follows and V_2 is buffered. At the falling edge of V_{in}, V_1 is buffered and V_2 follows. The figure below highlights this pattern flawlessly as we can now see that square wave at the V_p node is wider than that of the V_n node.

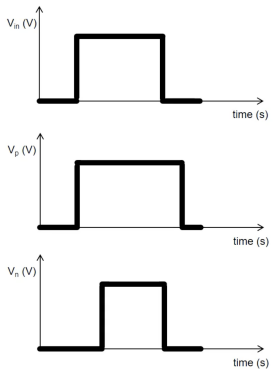


Figure 38: Break Before Make Square Waves

In order to get signal comparable to the figure above it is important to pick the right R and C values to get a sizable delay between signals. The equation for this delay is roughly

$$t = -RC * ln(\frac{V_i}{V_f}) \tag{23}$$

where V_i and V_f are the initial and final voltages across the capacitor. If we can assume that V_i is a small, but non-zero value (lets say .1mV and V_f is the upper power supply then it is clear that we need a large capacitor and resistor to increase the delay to a noticeable point. For our values of $R = 220\Omega$ and $C = 220nf$ we get the very nice and clean signal show below. The yellow signal corresponds to V_p output and blue shows off the V_n output.

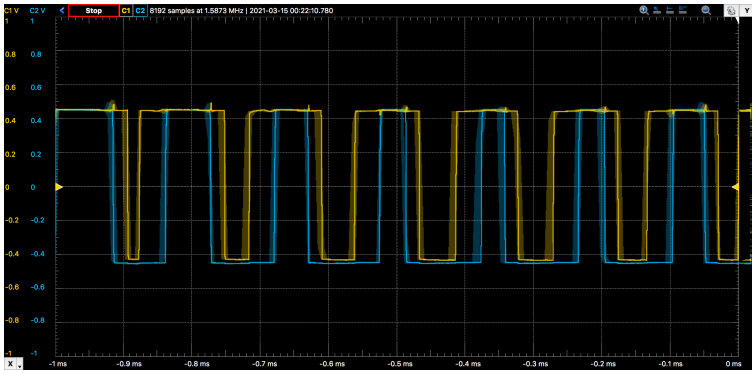


Figure 39: BBM Output Signal

Just as with what we expected, the V_p signal is the first to rise and the last to fall when the actual square wave is passed through the break before make.

This gives ample cushioning room for V_n to rise and fall without the worry that square wave will be transitioning from high to low at the same time.

As a reminder, this is ideal because we want the V_p and V_n signals to open and close the MosFet switches of the Class D Amplifier. And if Either gates are open at the same time, then there will be a dangerously large current flowing across both of them that could easily damage the chis and the speaker.

6.3 Class D Amplifier

Now that the Pulse Width Modulator signal is safe and wont cause any explosions we can send it carefully to the actual Class D Amplifier. The Class D Amplifier is comprised of a PMOS and NMOS gate which take in the the pulse width modulated signal and drive it with large amounts of current into the speaker.

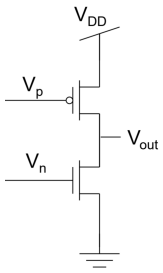


Figure 40: Class D Amplifier

Going back to our V_p and V_n signals, when V_p is low our PMOS is on and whenV_n is low the NMOS will be off. As soon as the first rising edge of V_p comes in, the PMOS begins to open the switch and turn off. Thanks to the delay, when V_n finally reaches high, the PMOS will already be completely shut off and we won't have to worry about sending a large current through the chips and potentially damaging them. Once both V_p and V_n are high, the PMOS is shut off and the NMOS is on, connecting the output to ground. On the falling edges, V_n goes low first which turns of the NMOS chip. Then V_p goes low and PMOS turns back on again, closing the switch and supplying 4.5 volts to the output.

This cycle continues for as long as the pulse width signal is active and what the PMOS and NMOS do is preserve the signal of the pulse width while supplying a large current to it which will help feed it to the speaker. The figure shown below is the final output of the Class D Amplifier; we cna see that the pulse width signal is still being translated by the system but there is a lot of noise surrounding it. By sending it through a band pass filter we should be able to deliver a cleaner signal to the speaker.

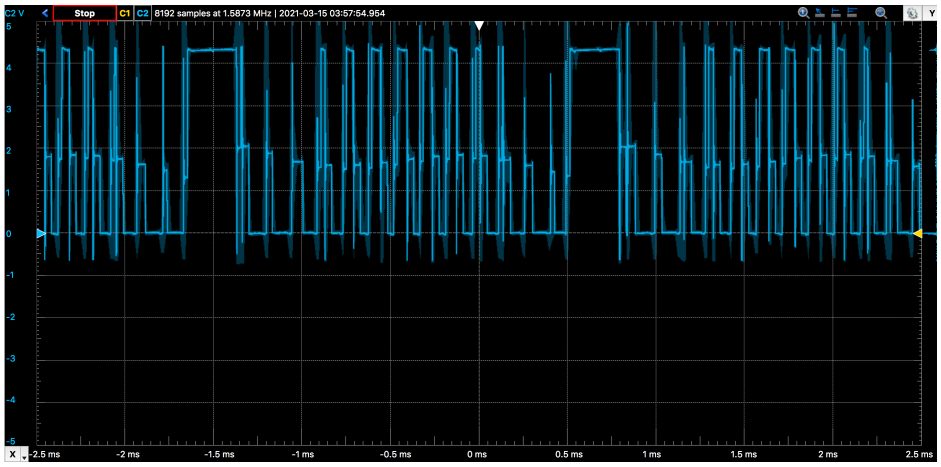


Figure 41: Class D Amplifier Output

- IRLD024 NMOS x1 ($1.22)
- IRFD9020 PMOS x1 ($1.57)
- LM393 Comparator x2 ($0.42)
- LF411 opamp x1 ($1.41)
- CD4069 Inverter x1 ($0.43)
- 1N4148 Diode x2 ($0.11)
- 220nf Capacitor x2 ($0.26)
- Assorted resistors x7

This gives us a total cost for the block equal to $5.84

7 Bandpass Filter

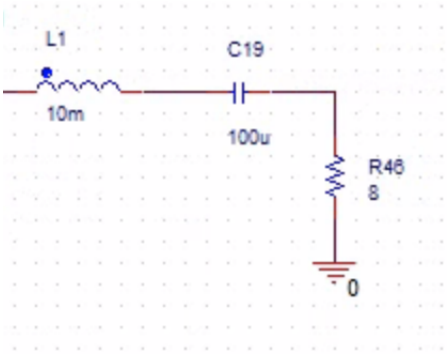


Figure 42: sawtooth Wave

The final part of the system before sending the signal to the speaker is a bandpass filter. This filter is comprised of just an inductor to filter out the high frequencies and a capacitor to filter out the low frequencies. We would typically use an active RC filter with an opamp to build a bandpass filter (check section 2.0.3) but to preserve the current produced by the Class D Amplifier that is necessary to run the speaker we are limited to using a passive RLC filter.

The goal of this filter is to remove the added sawtooth wave, leaving a pulse width modulated sine wave to be delivered to the speaker. This means that the only frequency we are trying to pass is the 450Hz of the original sine wave. Given that the equation for the center frequency of an RCL circuit is

$$f_c = \frac{1}{\sqrt{LC}} \tag{24}$$

We can find appropriate values for L and C. The values that we've happen to chose are $L = 1.2mH$ adn $C = 100\mu f$. Since we didn't have a 1mH inductor we decided to go with the 10mH inductor to use in the filter.

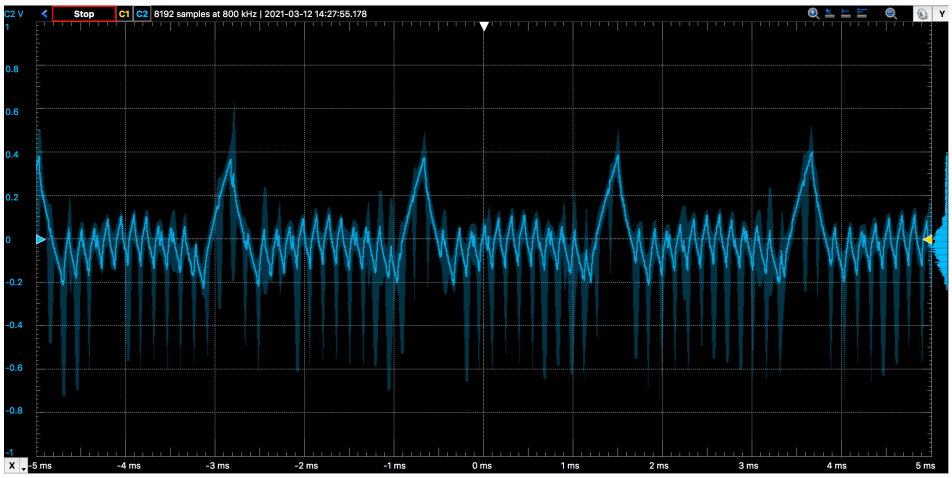


Figure 43: Speaker Input

- 10mH Inductor x1 ($0.69)
- 100μf Capacitor x1 ($0.85)

This gives us a total price for this block equal to $1.54

8 Financial Costs

Tallying up the individual cost for each of the circuit blocks in the project and we get that the total cost for the entire project is $27.97. It should be noted that this was calculated by adding up all of the individual cost of the blocks but with a few exceptions. The cost for the capacitors is how much they're worth to buy in bulk so no matter how many of a single capacitor value I used, I only counted the bulk purchase once. I also had all of the resistors cost $3.00 for one bulk, 600pc resistor pack that included every value I used in the project.

9 Experimental Results

With the project complete we conducted a couple of tests to see how the system functioned as a whole.

The very first thing we did was turned on the power rails and connected the Wienbridge Oscillator directly to the Pulse Width Modulator (bypassing the transistor) to make sure that the speaker was operational. It thankfully was, and we moved on to the next test.

We went from component to get an idea of how the signal evolved throughout the system and to make sure everything was working as it should.

One important measurements we took included the the operation of the initial detector with the initial heartbeat as well as the first heartbeat it would

detect after going silent for awhile. After letting the heartbeat wave play out for a couple of tones we paused it and let the peak detector in the initial detector circuit settle down. To really solidify that it works we began playing the heartbeat again and it perfectly play the long signal as soon as it the first beat went through.

To test the operation of the repeated sensor we played both a fast and slow heartbeat to test the range at which the system was effective. The system worked flawlessly for the fast heartbeat, but we did run into a couple situations when the system would trigger on the second "dub" pulse of the "lub dub" signal. This is probably do to the detection sensor detecting a single phantom signal that shifts that high/low time of the D Flip-Flop by one pulse signal.

Overall, every signal we measured followed closely, if not exactly, to what we expected the waveform to look like at that given part of the circuit. It was also shown that the correct tone and duration of an audible signal were played at the correct times during the casual test of the heartbeat monitor with both the fast and slow beats. There was some area for improvement, but having the total circuit work reliably across multiple heartbeat tests definitely verified the functionality of the product.

10 Conclusion

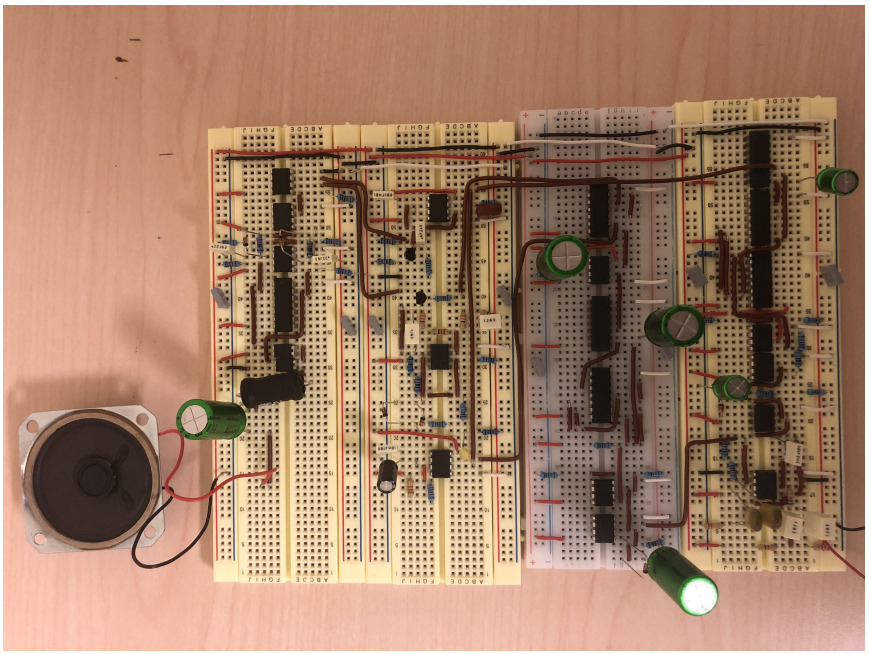


Figure 44: Circuit Board Layout

This project was great test to the things we learned over the term, it really gave me some freedom to implement all sorts of designs and problem solving ideas. Building the Break-Before-Make circuit greatly helped me in understanding how it worked and the same goes for the Sawtooth generator. Specifically, I feel that understood the effects of the resistors and capacitors within the break before make circuit, but while I was build and testing it it made the importance of RC decay much clearer and helped me get the values I needed.

I think one of the greatest things I learned from this project is how to troubleshoot effectively. There were so many instances of me assembling component blocks together and hooking them just to find that nothing is working. It was incredibly humiliating and frustrating to see something not work after already proving it should on PSpice and with calculations; but thankfully, the problems were usually a misplaced wire or misused component. Taking the time to go block by block and checking each component's inputs and outputs trying to find where the broken signal was honestly a great lesson. Doing everything one step at a time and checking the status of the signal also gave a lot of insight into what each part of the circuit does and how it interacts with everything else. I think overall, debugging really made me think critically about my circuit and the importance of every individual part.

Going back there are definitely a couple of things I'd like to improve. The first thing I'd like to tweak is the final bandpass filter, there were some wavelength that could've been attenuated more and the output signal was not as clear as I'd like it to be. I think I could've cleaned it up a bit more and had the speaker come out with a much more pleasurable noise. The problem may also have lied within the output of the Class D Amplifier, the signal was a lot messier than I anticipated but I'm not quite sure how it was expected to look. I also had an issue where any external bumps that the mic picked up could cause the D-flip flop trigger on the "dub" pulse instead of the "lub". I feel like there is some way to fix that up to make it an even rarer occurrence. If I could, I'd also try to implement the 9V battery as the power source to the monitor. I had a brief test with a voltage divider and a virtual ground, but it was overall unsuccessful. With more time, I definitely could've worked out some solution to get it up and running on a 9V battery; finally get it "portable".