

# **Final Report**

## **Third Eye for Blind People**

ECE-198 JS1

04/28/2024

### **Table of Contents:**

Topic Name	Page Number
1. Proposal Statement	2
2. Timeline & Milestones	2-3
3. Analysis of Sub-Circuits	3-16
4. Final Project Assembly	17-18
5. Individual Contributions	18
6. Conclusion and Future Directions	18

Satyam Raj - ECE110 (satyamr2)  
Apoorva Sharma - ECE110 (as204)

- **Proposal Statement:**

“Third Eye for Blind People” is a device that uses an ultrasonic sensor to provide feedback to the person wearing it in two different ways. Firstly, it provides haptics-based feedback wherein the vibration intensity increases as the person gets closer to any object that could be a hurdle while they are walking. Secondly, it offers sound feedback as beeps. The intensity of the beep increases as well when the person goes closer to any object. In order to build this circuit, we will be utilizing materials studied in both ECE110 and ECE120. We will implement several circuits in order to build it, some of which will be digital circuits and some of which will be analog circuits.

The output from the ultrasonic sensor will go into a digital circuit, which we will call the digital integrator. The digital integrator will count the length of the Echo signal coming from the Ultrasonic sensor. The counted value can be read as digital bits from a four-bit binary counter. The four-bit binary counter’s value will be sent to another digital circuit containing JK Flip-Flops, which will store the counted value till the time a new value is generated. The stored digital values in the JK Flip-Flops will be converted to analog and fed into a tone generator and motor control circuit. The tone generator circuit will drive the speaker whereas the motor control circuit will drive the motor.

- **Timeline**

Week of February-5	Project Proposal Revision & Changes	Completed
Week of February-12	Schmitt-Trigger Oscillator	Completed
Week of February-19	<b>Schmitt-Trigger Oscillator (Milestone-1)</b>	Completed
Week of February-26	<b>Final Functioning of Flash ADC (Milestone-2)</b>	Completed
Week of March-4	Tone generator circuit	Completed
Week of March-11	Spring Break	Completed
Week of March-18	<b>Final Functioning of Tone Generator circuit (Milestone-3)</b>	Completed
Week of March-25	Combinational logic design of the digital integrator	Completed
Week of April-1	<b>Digital Integrator Circuit &amp; DAC (Milestone-4)</b>	Completed

Week of April-8	<b>JK Flip-Flop (Milestone-5)</b>	Completed
Week of April-15	<b>Motor Sub-Circuit (Milestone-6)</b>	Completed
Week of April-22	<b>Final project functioning (Final Milestone)</b>	Completed
Week of April-29	Project Demonstration	Scheduled for 30th April.

## ● **Milestones**

In order to divide the work fairly and plan the project in the most efficient way we have divided our project into 7 milestones.

1. Schmitt Trigger Oscillator
2. Flash Analog to Digital Converter
3. Tone Generator Circuit
4. Digital Integrator Circuit
5. JK Flip-Flop Circuit
6. Motor Sub-Circuit
7. Final Project Assembly

## ● **Analysis of Subcircuits:**

### **1. Schmitt Trigger Oscillator**

Schmitt Trigger is our first sub-circuit in the project. We are using it to provide the trig signal of the ultrasonic sensor. We have used a typical square wave oscillator, which we learnt in our ECE-110 class. Since we want to calculate the distance almost every second, we have used a capacitor with a value of 30 microfarad and a resistor with a value of 30 kilohms.

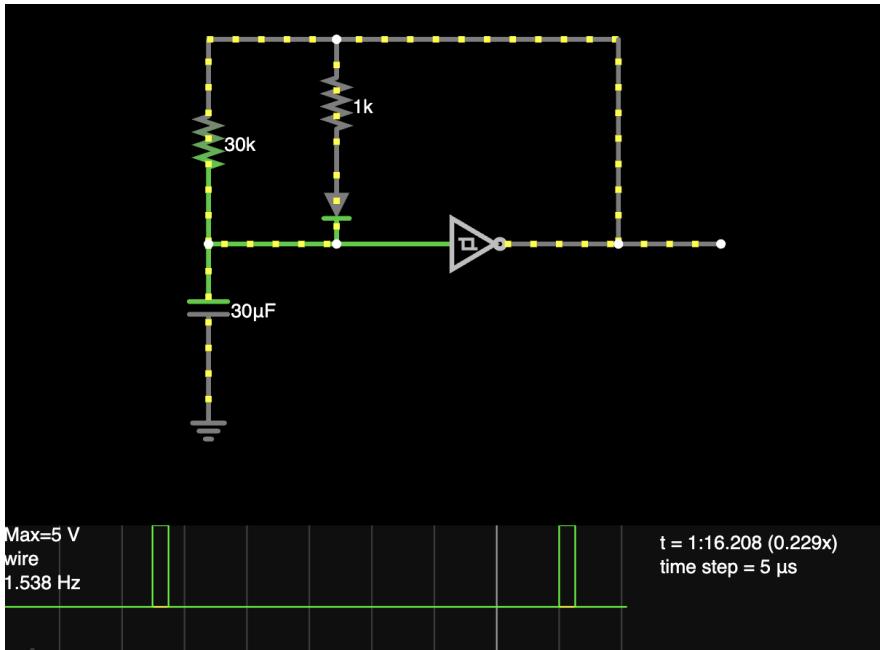


Fig: The trig signal generator circuit for the ultrasonic sensor

As can be seen, we have used a diode so that the charging time of the capacitor is less and the discharging time is high. Doing this ensures that we are giving enough time to the ultrasonic sensor to send and receive its signal and the digital integrator to count the length of that signal.

## 2. Flash Analog to Digital Converter

It is very evident from the name that our Flash ADC takes an analog voltage as an input and outputs a digital signal. The Flash ADC that we have built takes a voltage ranging from 0 to 7 volts as input and outputs a three-bit corresponding digital signal.

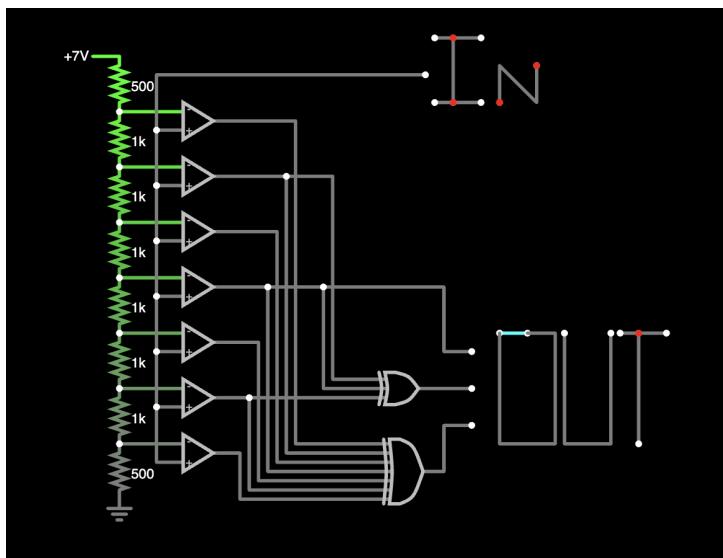


Fig: Flash ADC built on Falstad

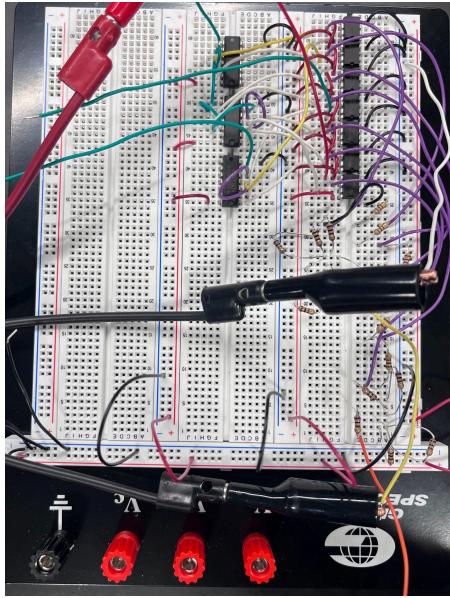


Fig: Flash ADC built on the breadboard.

Our Flash ADC works on the principle of comparing a voltage that is being passed through a network of resistors. In our circuit, the resistor network consists of a 500-ohm resistor followed by 6 1-kilo-ohm resistors in series, and finally, a 500-ohm resistor connected to the ground. At every resistor junction, we compare it with the previous junction using an op-amp. Furthermore, using XOR gates, the signal is finally converted into digital.

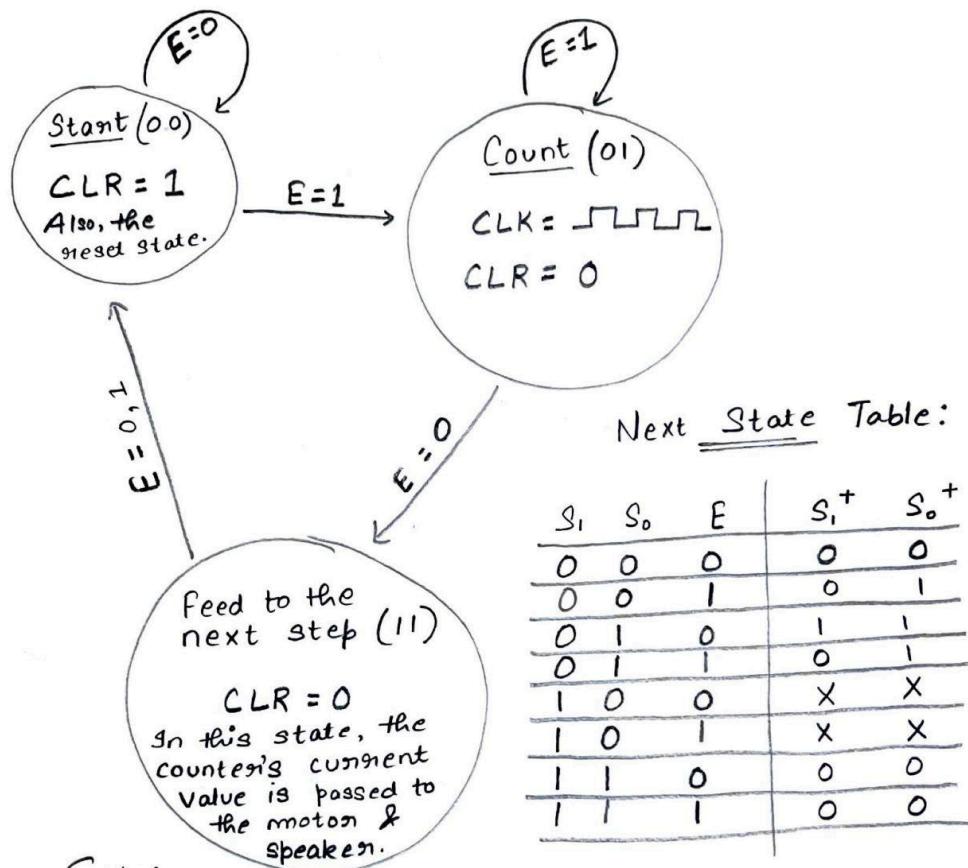
### 3. Digital Integrator Circuit

We developed a combinational logic that would count the length of the Echo signal of the ultrasonic sensor and then give the corresponding counted value in binary.

This circuit contains two parts: the FSM and the Clock cycle of the binary counter.

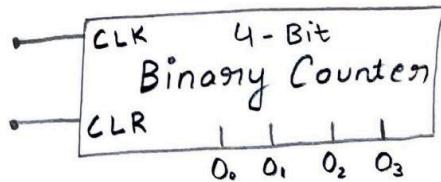
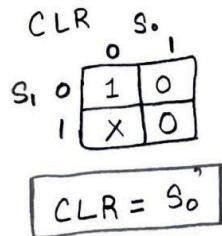
FSM of the Digital Integrator:

\* State diagram for the digital integrator:



CLR Signal Table:

$S_1$	$S_0$	CLR
0	0	1
0	1	0
1	0	X
1	1	0



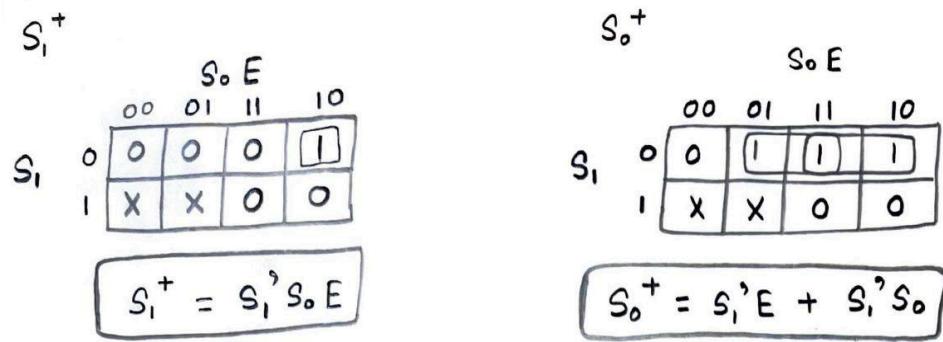


Fig: Digital Integrator FSM state diagram and K-map.

As seen in the above picture, our FSM consists of 3 states: Start/Reset (00), Count (01), and Feed (11). The input of our FSM is the Echo signal of the ultrasonic sensor. If the Echo signal is low ( $E=0$ ), then the FSM will remain in the Start (00) state and the binary counter will store the value 0 (CLR=1).

Once the Echo signal becomes high ( $E=1$ ), then the initial Start state will be transitioned to the Count (01) state, and the value of the binary counter will increase by 1 bit. For as long as the Echo signal is high ( $E=1$ ), the FSM will remain in the Count state and the value of the counter will consequently keep increasing by 1 bit.

When the Echo bit becomes low, the current Count state transitions to the Feed (11) state. In this state, the current value of the counter will be given to the next part of our circuit, and then regardless of the Echo signal's value ( $E=x$ ), the Feed state will transition to the Start state and the counter will be reset.

Now, the maximum value of the Echo signal that our binary counter can count will be 8 bits. We then measured the longest value of the Echo signal that the ultrasonic sensor can measure and divided it into 8 parts because we wanted to provide a voltage between 1 to 8 Volts to our speaker to generate the required beeping sound. The divided value of the echo signal will be the pulse length of the clock cycle for the FSM.

Clock Cycle of the Binary Counter:

Fog, the clock cycle:  
We will use MOSFETs:

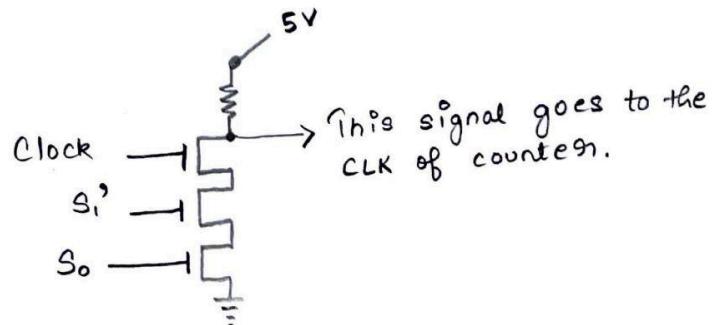


Fig: Clock cycle of binary counter schematic diagram

We used AND logic built using 3 MOSFETS to generate the clock signal of the binary counter since we only want to count the value in the state 01.

**To test the functionality of this circuit,** we provided the Echo signal as the input to our FSM and then used the oscilloscope to measure if our binary counter was correctly counting the length of the Echo signal. Below, we have provided a picture of the output of our digital integrator on the oscilloscope.



Fig: Output of the digital integrator circuit as shown on the oscilloscope

The blue graph is the Echo signal of the ultrasonic sensor. The remaining three graphs: red, yellow, and green, are the first, second, and third bits of the binary counter, respectively. We can

see that while the ECHO signal is high, in the first clock cycle the first bit (red signal) becomes high and the remaining two bits (yellow and green) remain low, meaning that the value of the binary counter currently is 001 bit. Then at the next clock cycle, the values of the first and the third bit (red and green signals) become low and the value of the second bit (yellow signal) becomes high stating that now the binary counter has the value 010 bits which is 2, meaning that the length of the Echo signal is 2. Thus, the digital integrator counts the length of the Echo signal, and at the last clock cycle, it can be observed that the length of the Echo signal as counted by the digital integrator is 7 (111 bits) for that particular instance.

#### 4. JK Flip-Flop Circuit

The Digital integrator counts the echo signal and stores it in a four-bit binary counter. In order to build a circuit that stores the counted value of the binary counter till the time a new circuit is built, we will use JK Flip-Flops.

Firstly, in order to understand how a JK Flip-Flop works we will analyze the truth table of a JK Flip-Flop circuit:

**Symbol**
**Truth Table**

<b>CLK</b>	<b>J</b>	<b>K</b>	<b>Q<sub>n+1</sub></b>
↑	0	0	Q <sub>n</sub>
↑	0	1	0
↑	1	0	1
↑	1	1	Q <sub>n'</sub>

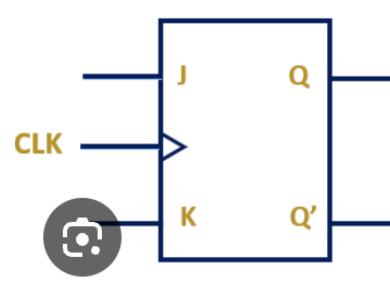
**Symbol**


Fig: JK flip-flop truth table

As can be seen, the JK Flip-Flop does not store a direct input like a D Flip-Flop but rather operates in a different way. In order for the JK flip-flop to store a 0, the JK inputs to the flip-flop have to be 01 respectively, and in order for it to store ,1 the JK inputs to the flip-flop have to be 10. The JK flip-flop also stores the value from the previous cycle if the JK input is 0,0, respectively. We utilized this behavior of the JK flip-flop and designed a separate FSM that will drive these JK Flip-flops.

#### FSM of the JK Flip-Flop:

$$D = S_1 \cdot S_0$$

$B_i$  = One of the Binary input to this FSM , coming from the binary counter.

Truth Table

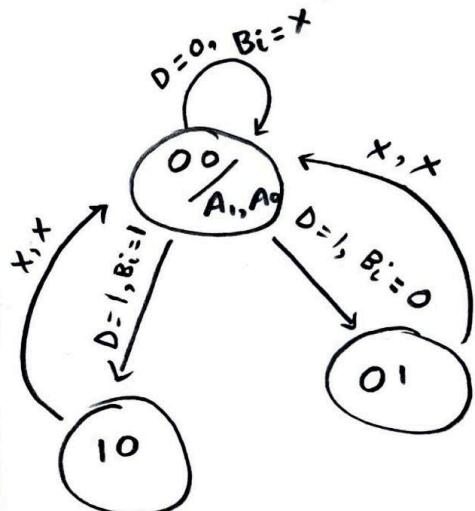
$A_1$	$A_0$	$D$	$B_i$	$A_1^+$	$A_0^+$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	X	X
<hr/>				0	0
0	0	1	-	0	0
0	1	0	-	0	0
1	0	0	-	0	0
1	1	0	-	X	X
<hr/>				0	1
0	0	1	0	0	•1
0	1	1	0	0	0
1	0	1	0	0	0
1	1	1	0	X	X
<hr/>				1	0
0	0	1	1	1	0
0	1	1	1	0	0
1	0	1	1	0	0
1	1	1	1	X	X
<hr/>					
				$A_1^+$	$A_0^+$

$DB_i$	$00$	$01$	$11$	$10$
$00$	0	0	X	0
$01$	0	0	X	0
$11$	1	0	X	0
$10$	0	0	X	0

$$A_1^+ = DB_i A_1' A_0'$$

State Diagram:

$A_1 - J$   
 $A_0 - K$   
 $(S_1, S_0)$  is from the digital integrator FSM.



$DB_i$	$00$	$01$	$11$	$10$
$00$	0	0	X	0
$01$	0	0	X	0
$11$	0	0	X	0
$10$	1	0	X	0

$$A_0^+ = DB_i' A_1' A_0'$$

Fig: FSM of JK flip flop and its K-map.

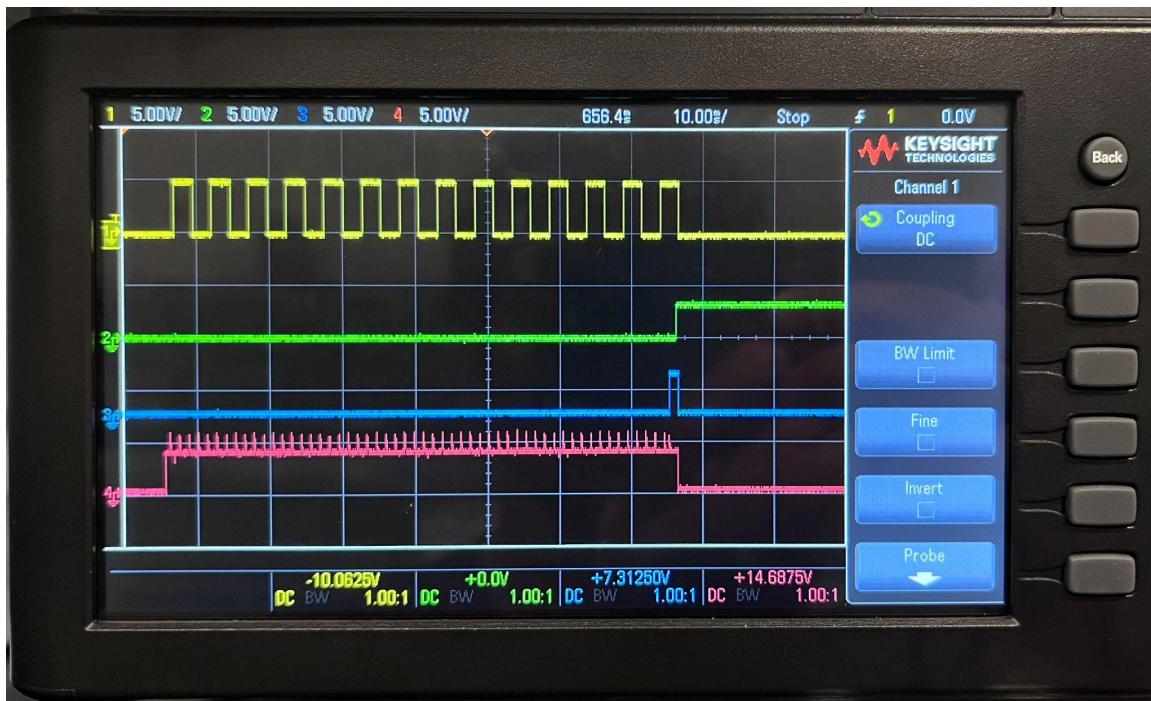
We know from the Digital Integrator that only in state 11 we have to convert the binary counter's stored value and feed it into the next circuit. Thus, we have defined a new signal called D which

is logical AND of S1 and S0. The other input to the FSM will be Bi, which is the ith bit of the binary counter. Since we need to store four bits we will need four similar circuits, one for each bit. The binary counter has three states, one in which it starts/resets while our D (S1 AND S0) is low. However, in case S1 and S0 become 11 the FSM goes to two different states based on the Bi, if the Bi is one it goes to 10 and when it is 0 it goes to 01. That has been chosen so that we can easily feed the state bits of this FSM to the J and K of the JK flip-flops.

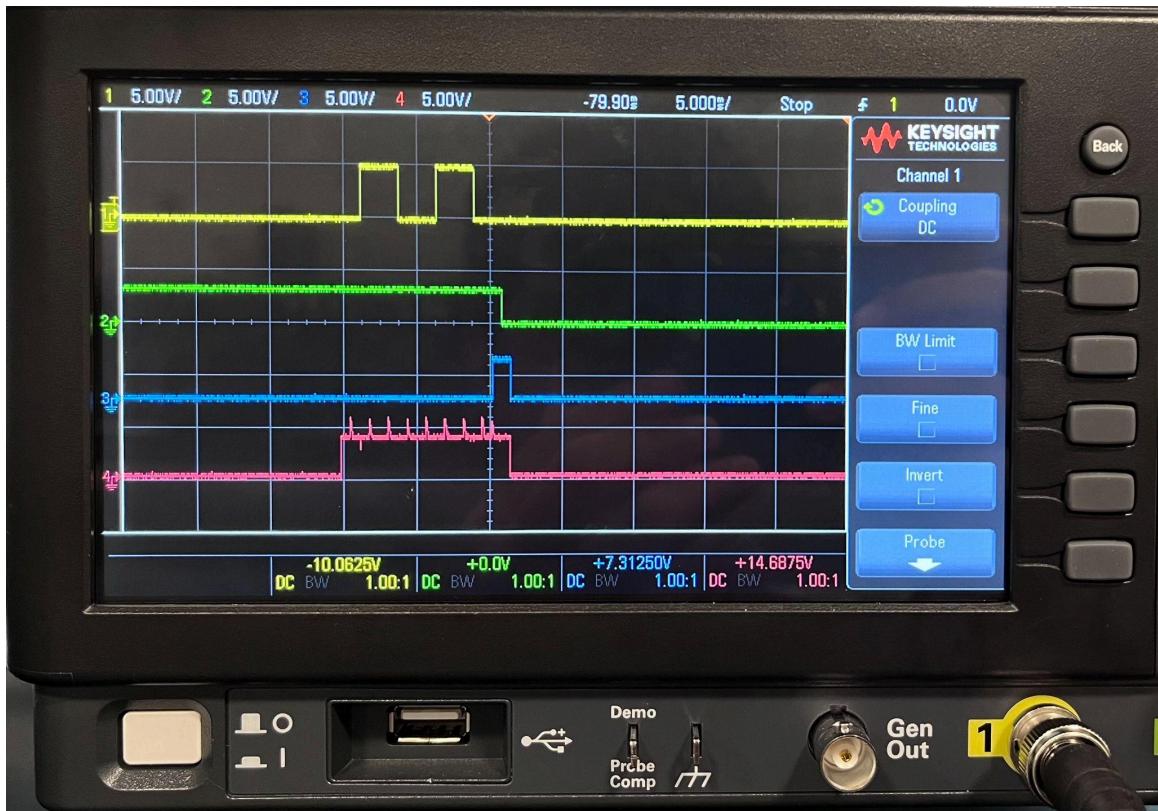
### Oscilloscope image verifying JK Flip-Flop Circuit Functionality:

Below we have provided two oscilloscope images to verify that our JK Flip-Flop circuit works as expected. In both the images the yellow signal is connected to the first bit of the binary counter and the JK flip flop is connected to the first bit of the JK flip-flop circuit. The blue and the red bit are connected to the state bits of the integrator.

As it has been explained above that the way the JK flip-flop circuit is expected to work is it should copy the binary counter's bit value to the JK flip-flop but only and only at the instance when both the state bits of the integrator are high.



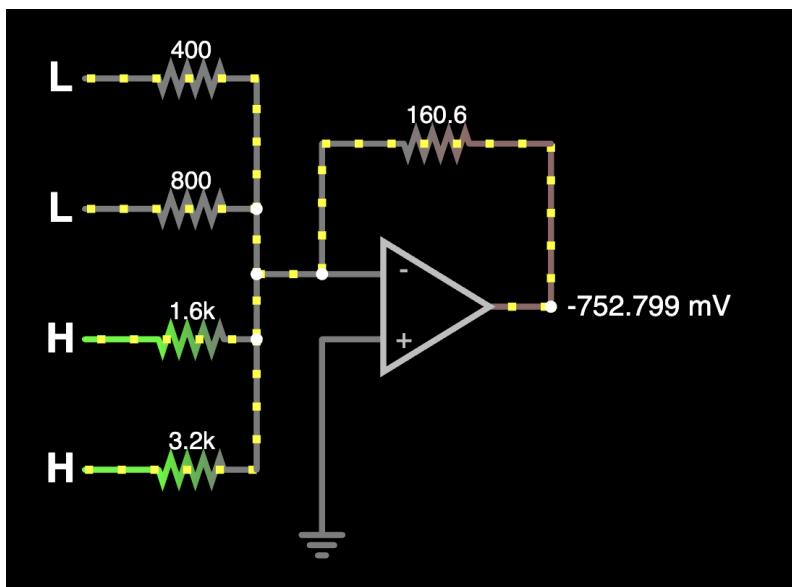
It can be seen in this above figure that when both the state bits (red and blue) are high, the binary counter's value (yellow) is copied to the JK flip-flop value(green). Initially the green was low but since at the instant when both the integrator state bits went high the yellow was high which resulted in the green to be high.



It can be seen in the above figure now that when both the state bits (red and blue) are high, the binary counter's value (yellow) is copied to the JK flip-flop value(green). Initially the green was high but since at the instant when both the integrator state bits went high the yellow was low which resulted in the green to be low after that.

## 5. Digital to Analog Converter

Circuit Diagram of the DAC:



Our four-bit digital-to-analog converter circuit was taken from Falstad. However, it was not working properly and our assumption based on testing was because of few resistors reaching its power limit thus we multiplied all the resistors by a factor of 4 and our DAC was completely functional.

## 6. Tone Generator Circuit

The tone generator circuit takes an analog signal as input and converts it into beeps of different frequencies. In the case of lower voltage, it beeps more rapidly, and in the case of higher voltage, it beeps less rapidly. To build this circuit we used the concepts learnt in ECE110. We used a logical AND gate with two inputs. One of the inputs is a variable frequency between 1 to 8 Hz, which is voltage-dependent. Another input to the AND gate is a constant high-frequency signal of about 700 Hz made of a Schmitt trigger inverter. Since our voltage-controlled oscillator was not only changing the frequency but the duty cycle as well of the signal we used a D-type flip-flop to overcome that challenge.

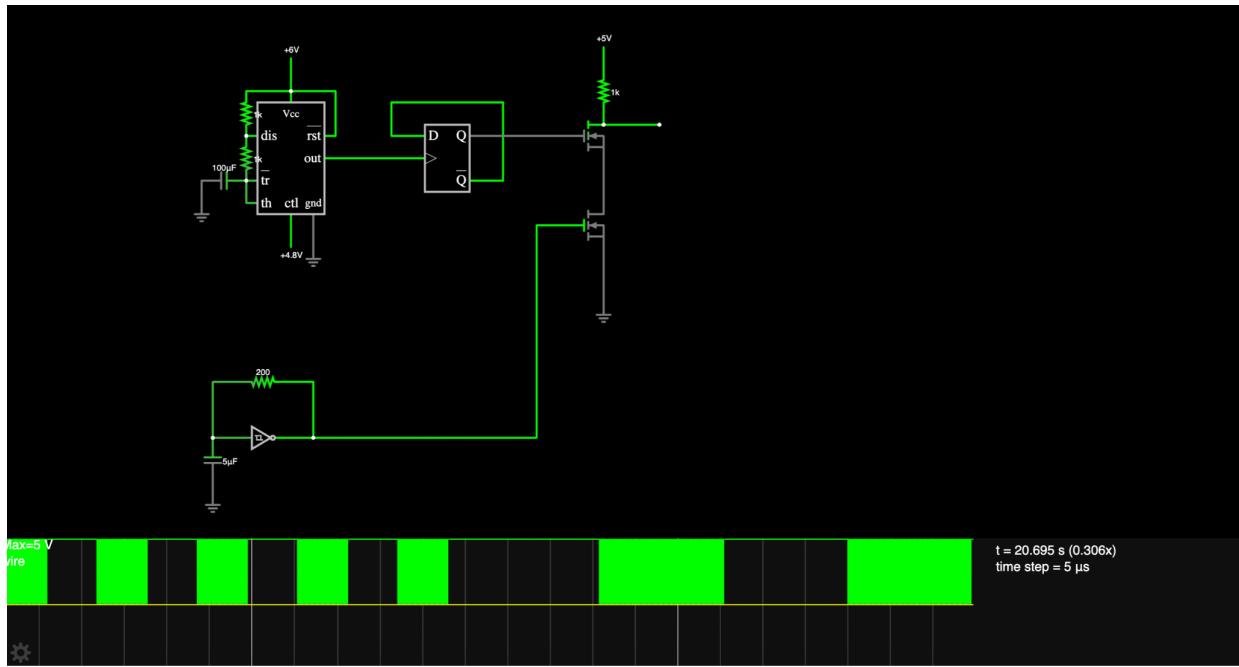


Fig: Circuit schematic of the tone-generator circuit.

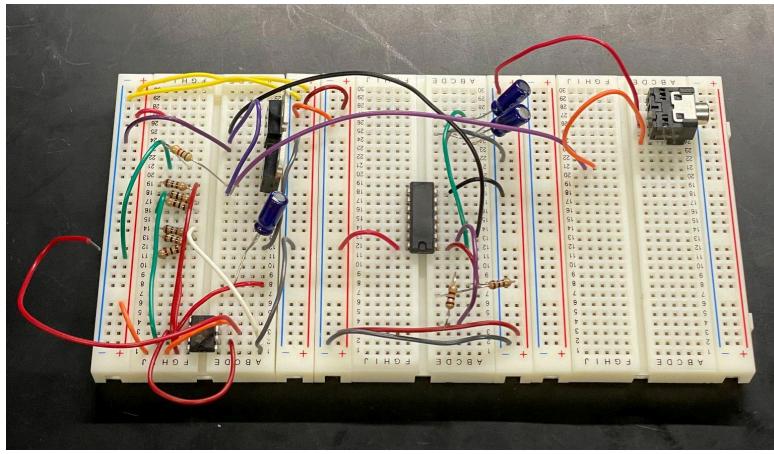


Fig: The tone generator circuit on the breadboard.

Since the output of the tone generator is sound, we have included detailed testing of it in the video submission of the Final report. In order to provide the control voltage of the 555 timer we have used the benchtop voltage supply.

## 7. Motor Sub-Circuit

We started working on the Motor sub-circuit by checking the supply voltage range for the motor. We found that the motor could operate within the voltage range of 1.5-4.2V. However, the output voltage from the DAC was in the range of 0-8 Volts. So, we built a circuit that would take the voltage range from 0-8 Volts as input from the DAC and convert it into the required operating voltage range of the motor from 0-4.2 V. This circuit converts the input voltage in such a way that if the input voltage will be 0V then the output voltage will also be 0V and the motor will be off, but if the input voltage will exceed 0V, then the output voltage would also increase subsequently at every higher value of the input voltage until 8V. At 8V the output of the circuit would be approximately 4.2V and the vibration of the motor would also be at its maximum vibration which is what we want. Below, we have provided a picture of the circuit on Falstad. We can notice that as the input voltage is 8V, the output voltage is 4.159V.

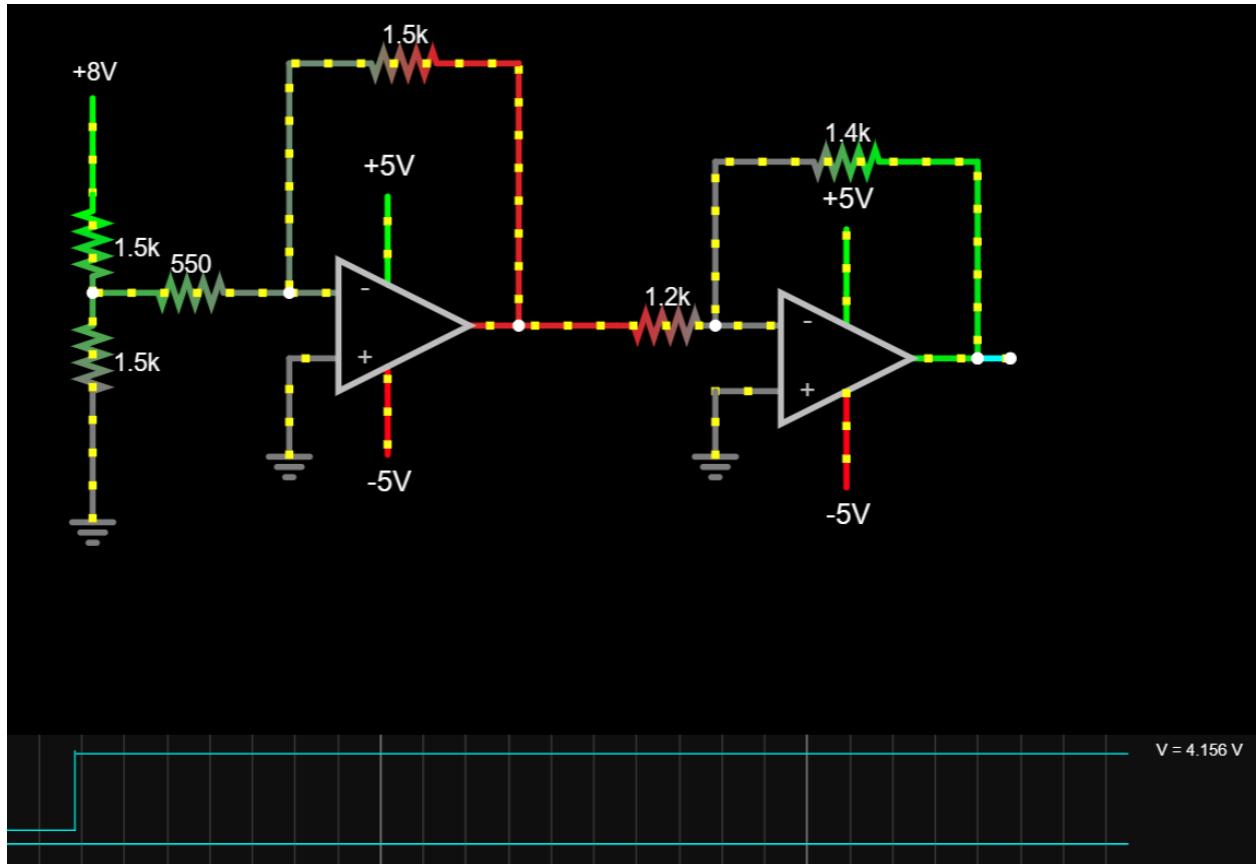


Fig: Motor sub-circuit on Falstad

We built the above circuit using 2 LM358 op-amps. The first op-amp works as an amplifier and amplifies the incoming voltage. Since the output of the first op-amp is negative, we use the second op-amp to convert the negative voltage to positive.

For the first amplifier, we obtained the required resistor values by using the formula  $V_o = -V_{in}(R_2/R_1)$ .

Below, we've provided a picture of the circuit built on the breadboard:

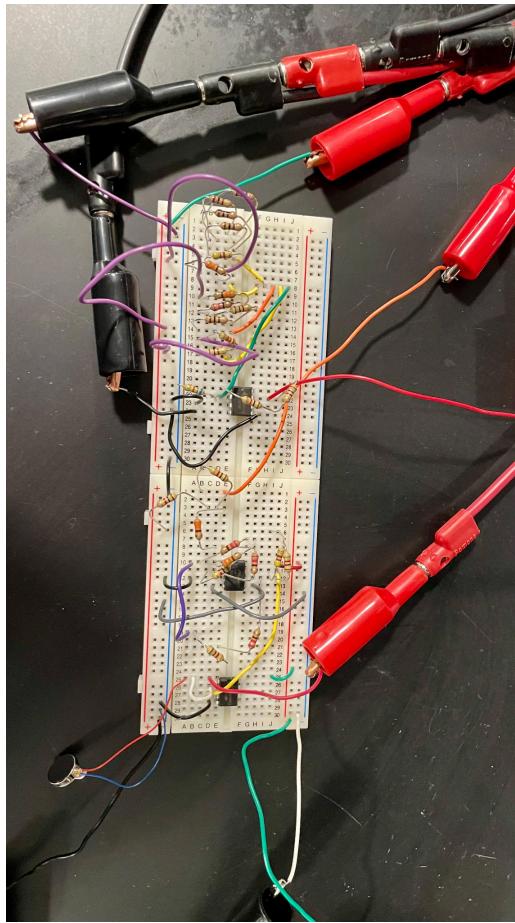
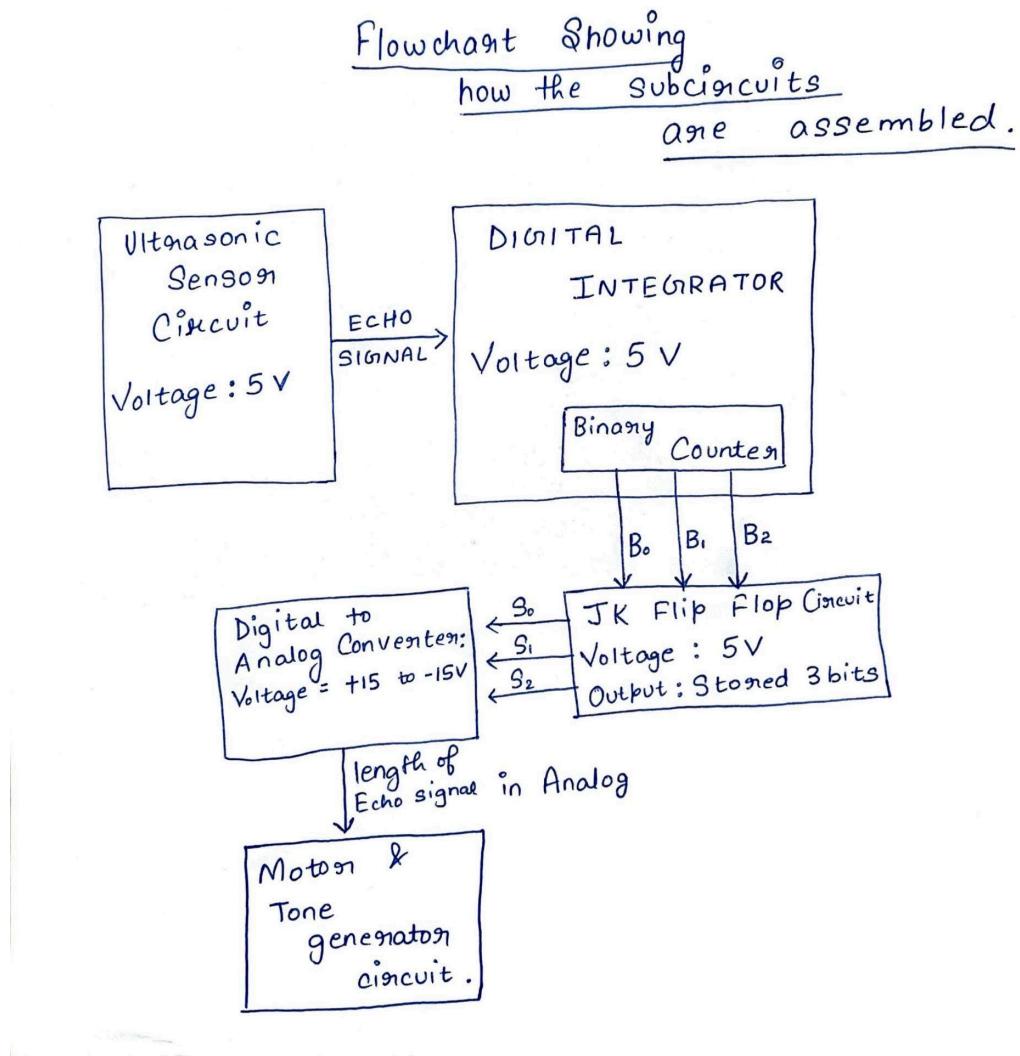


Fig: Motor sub-circuit on breadboard

**NOTE:** Before building the motor sub-circuit, we took the output of the JK flip flop and fed it into a hex inverter, and then gave this output to the DAC because at lower voltage we wanted our motor to produce high vibrations and vice versa as lower voltage from the JK flip flop corresponds to smaller distance. So, if we obtained a low voltage from the JK flip-flop, the hex inverter would convert it into high voltage, and then this voltage would be converted to its specific analog value by the DAC which would further be given to operate the motor and thus the motor would produce high vibrations.

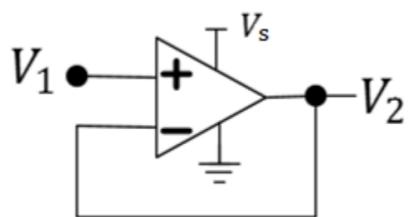
It should be noted that our motor circuit works completely differently from the Tone-generator circuit. The Tone-generator circuit is already designed to produce low beeps per second when given a high input voltage.

- Final Project Assembly



A very important thing to keep in mind while assembling the subcircuits is to ensure that the connection of subcircuits is not altering the behavior of any circuit individually. For our case we noticed that connecting the output of the Digital to Analog converter to the tone generator circuit was modifying the converted value and to make sure it does not happen we put an op-amp buffer in between the 555 timer of the tone generator circuit and the DAC.

Below is a circuit diagram of the Op-Amp buffer we used:



Since the final output of our circuit is sound and vibrations we have demonstrated that in our video submission.

- **Individual Contributions:**

**Apoorva:** Apoorva took the lead while we were building the tone generator circuit, the motor driving circuit, the digital-to-analog converter circuit, and the ultrasonic sensor circuit. Apoorva also wrote several sections of the document and weekly journals which corresponded to these above-mentioned circuits.

**Satyam:** Satyam took the lead while we were working on the digital integrator circuit and the JK flip-flop circuit. He also designed the FSMs for both circuits. Satyam also took the lead while building the analog to digital converter. He wrote the sections corresponding to these sub-circuits and the conclusion for the final report.

- **Conclusions and Future Directions:**

Our project, Third Eye, includes various concepts learned in ECE-110 and ECE-120 such as the Schmitt trigger oscillator, 555-timer, finite state machines, MOSFETs, etc. Thus, it is a very useful project which strengthens our understanding of the material. It uses an ultrasonic sensor as an input and gives the output through a speaker and a motor. The processing part of this project includes three steps and each of them have been explained in detail for anyone to understand and build it. For the successful completion of our project, we had divided it into 7 sub-circuits. The detailed timeline has been presented in the document above.

There are a few shortcomings in the project that need to be considered while building the circuit. During our build, we noticed that the circuit only works when we are connecting the oscilloscope probe to the integrator bits which we are passing from the digital integrator and sending to JK flip-flop. We were not able to figure out the exact reason why it was happening and it could have been an impedance mismatch. To fix this as per Prof. Schmitz's suggestion we inserted a one megohm resistor and a 10 microfarad capacitor in place of the oscilloscope probes. While this was not a viable solution it worked as we were able to replicate the Thevenin equivalent circuit of the inside of the oscilloscope. While building this circuit, it would be a good idea to figure out why that happened.

A possible expansion of our project would be to try and make the circuits as small as possible so that they can actually fit in a device that is easily accessible by blind people. In our project, we did not work on the mechanical design of the device. Another possible expansion would be to use infrared sensors to measure the distance instead of ultrasonic sensors.