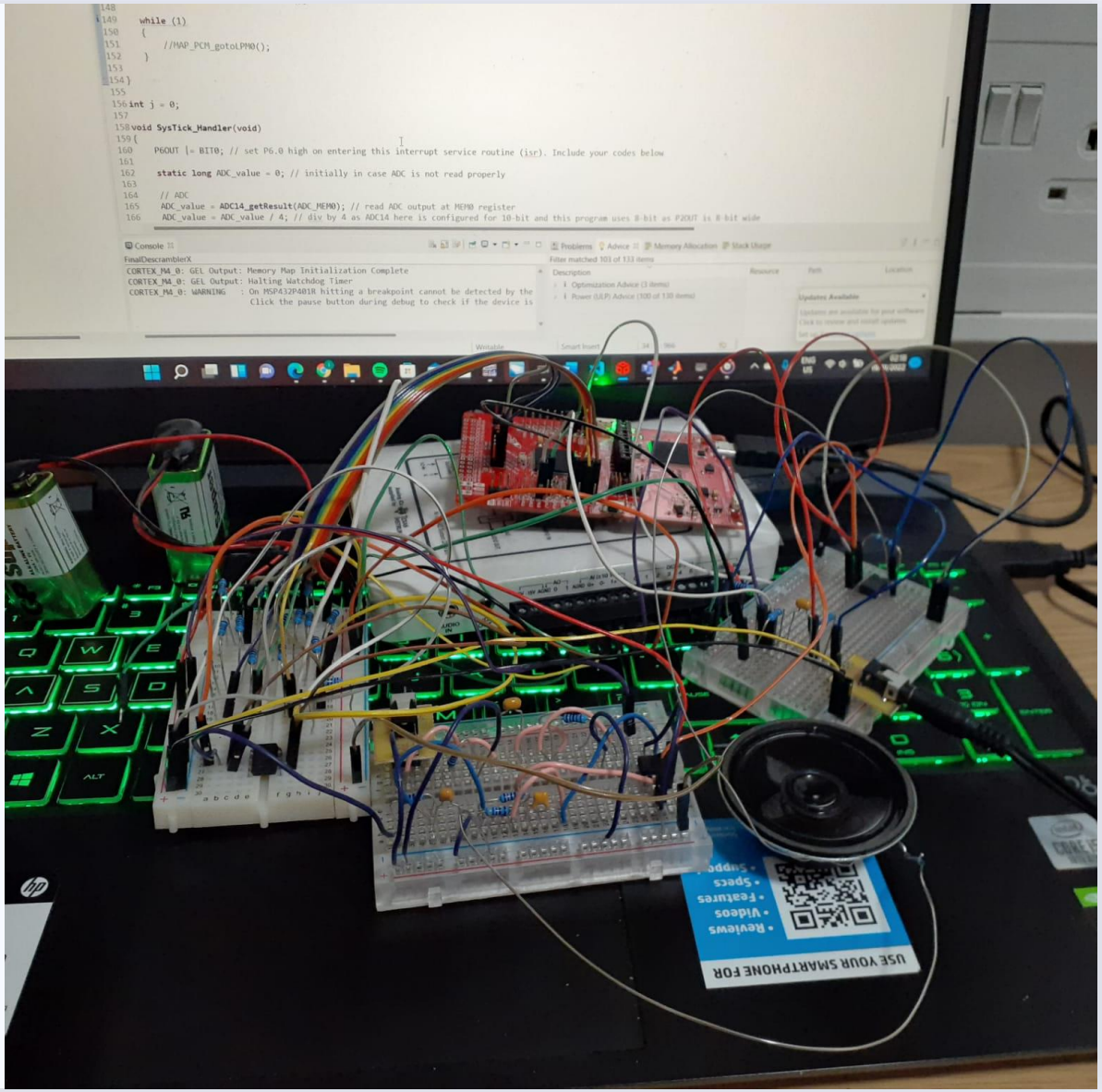
SCENARIO X

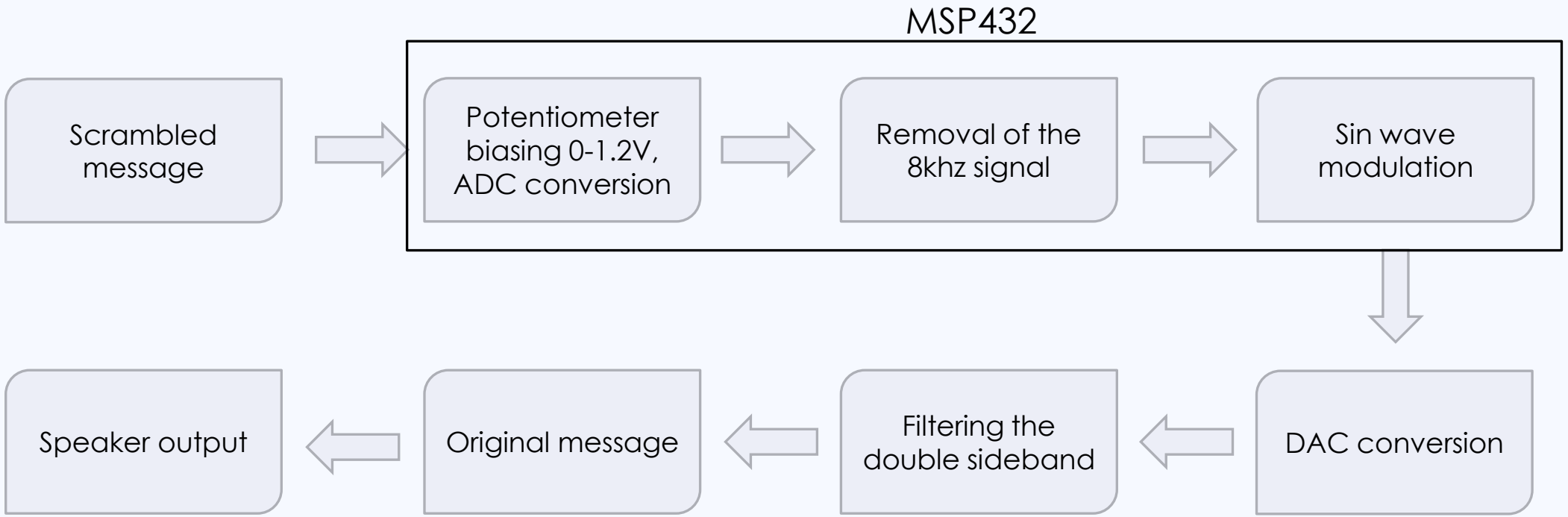
Team: Iason Chaimalas, Arlend Osmani, Joshua Tan, Mingyu Jia



MAIN OBJECTIVES

1. Work out how the scrambling is done
2. Design and build a digital real-time audio descrambler to work out what the secret message is.

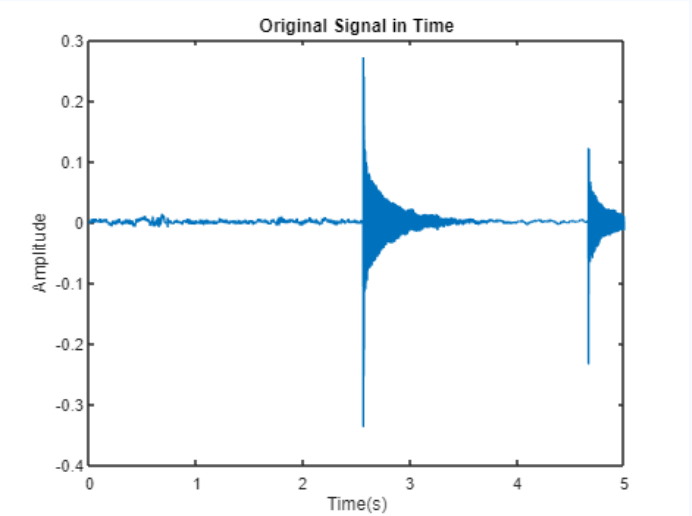
GENERAL BLOCK DIAGRAM TO DESCRAMBLE MESSAGE

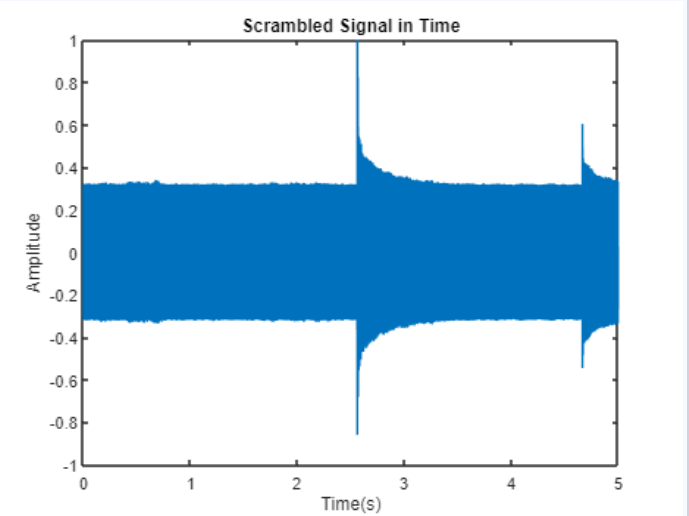


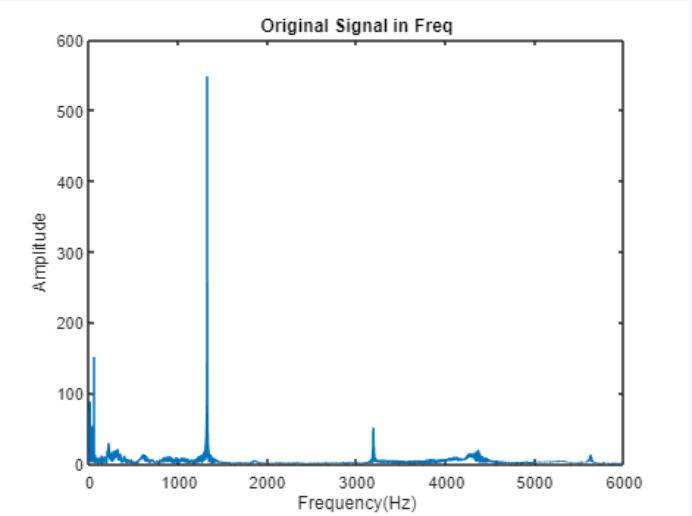
How the Message was Scrambled

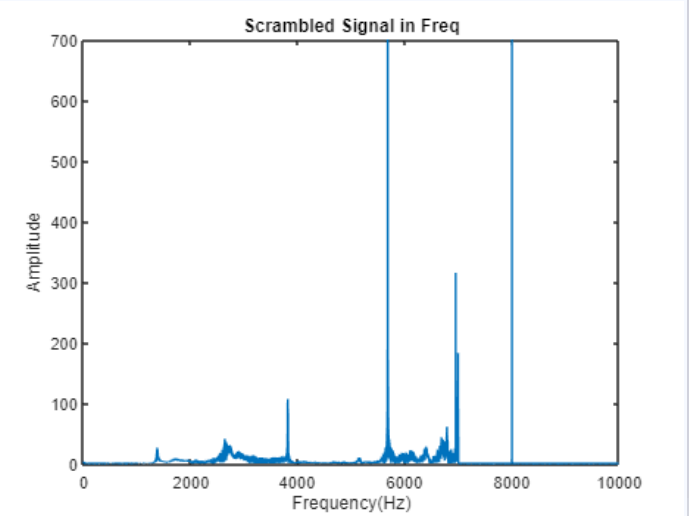
The message was flipped in the frequency spectrum about the point 3.5kHz.

Furthermore, an 8kHz sine wave was added to the signal.

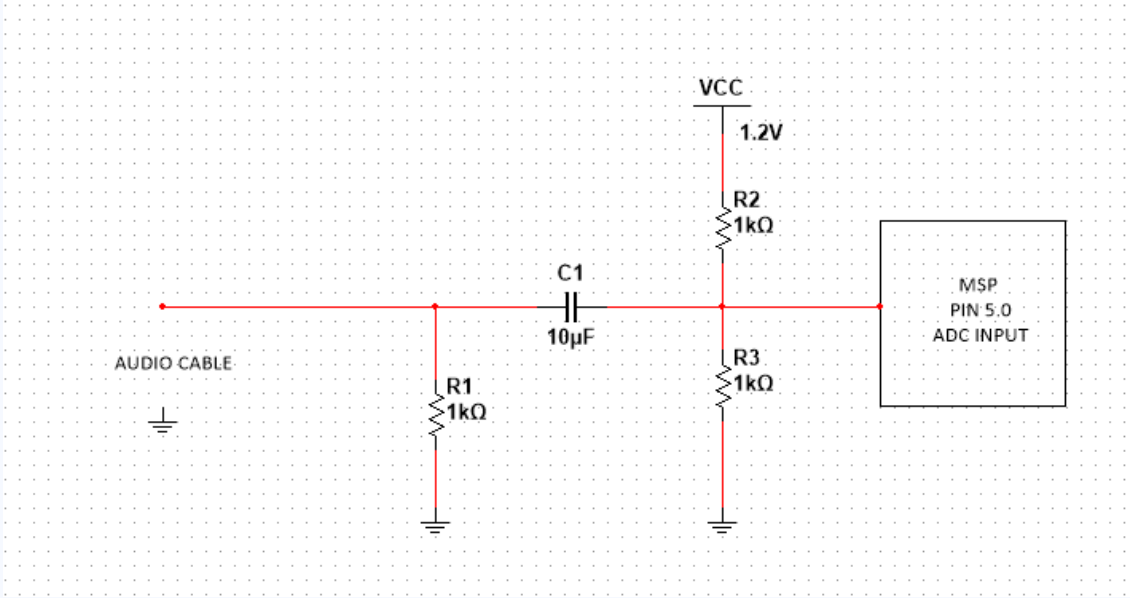






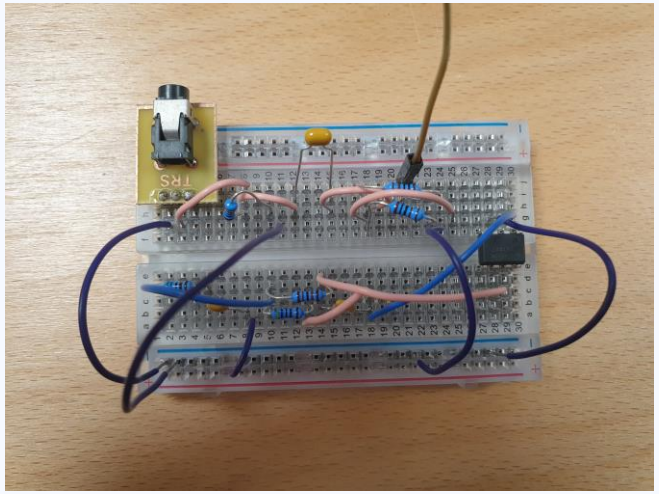


Potentiometer Biasing Circuit



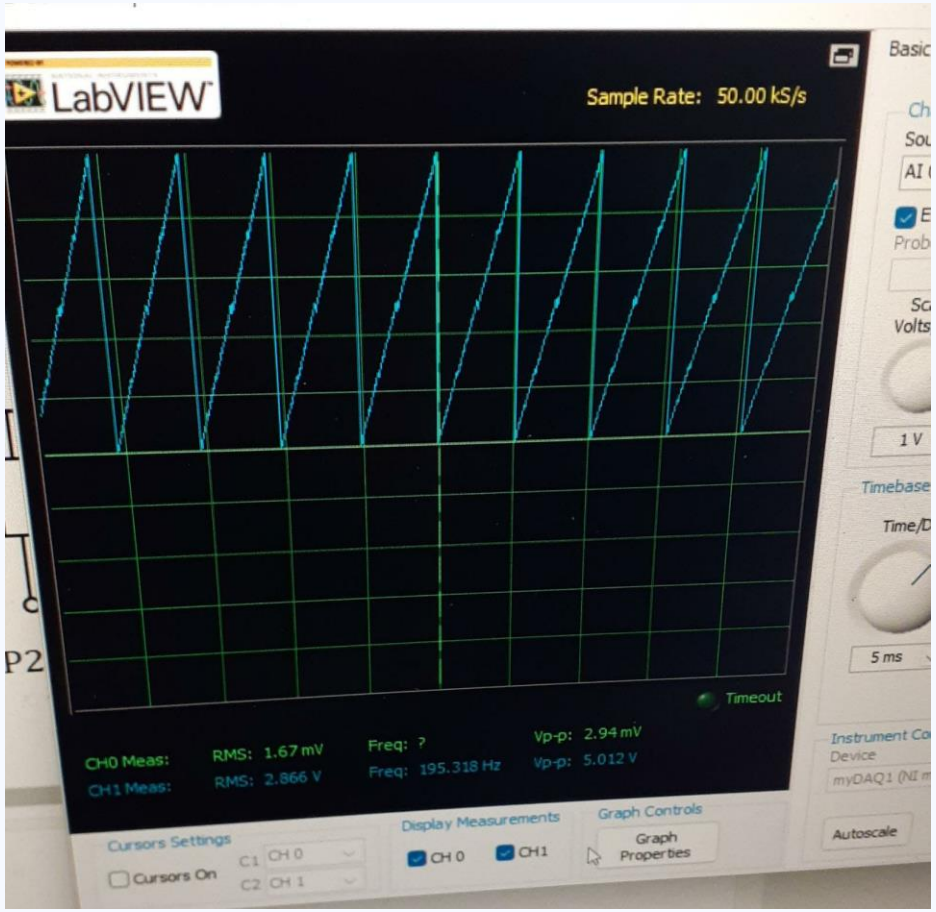
How it works:

- Removes any previous DC offset with C1 capacitor
- It is a bias circuit that is centered at **0.6V mean value** (added bias)
- **Scales audio input to 0 – 1.2V peak-to-peak**
- The MSP input pins can only read positive voltages; therefore, this circuit is used to shift the signal 0.6V upwards in time domain



ADC

- PIN 5.0 of the MSP Board is the ADC
- The audio input comes from the audio jack and goes through the potentiometer biasing circuit and into the ADC.
- The ADC has a set reference voltage of 1.2V. This means that the ADC can read voltages from the input pins between GND and 1.2V.
- For accuracy, the ADC is 10-bit 0 -1023
- This is divided by 4 to become 8-bits 0-255
- This is due to pins 2.0 - 2.7 only being 8-bits wide, and the signal is outputted at these pins.

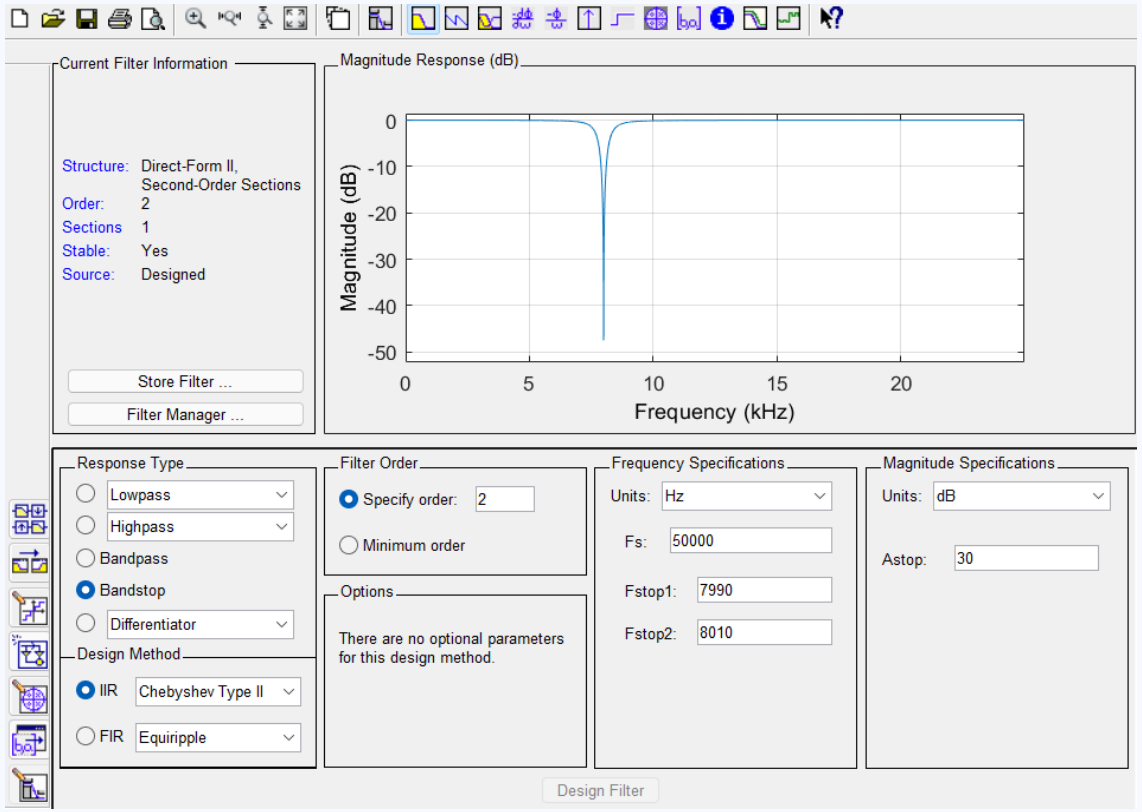


Ramp output testing ADC + DAC

Bandstop Filter

remove the 8 kHz tone.

Infinite Impulse Response (IIR) Type 2 2nd-Order Chebyshev Digital Filter




```
Section #1
--------------------------
Numerator:
 1
-1.071654436102308816103345634473953396082
 1
Denominator:
 1
-1.030715945021520196789310830354224890471
 0.923597589481017777757188014220446348l9
Gain:
0.961798794740508888878594007110223174095
--------------------------
Output Gain:
1
```

$y_n - 1.03y_{n-1} + 0.92y_{n-2} = 0.96(x_n - 1.07x_{n-1} + x_{n-2})$ -> $\mathbf{y_n = 1.03y_{n-1} - 0.92y_{n-2} + 0.96(x_n - 1.07x_{n-1} + x_{n-2})}$

Sine Wave

WHY Modulate by Sine?

$$\sin(x)\sin(y) = \tfrac{1}{2}\left[\cos(x-y) - \cos(x+y)\right]$$ -- Half-power sidebands about y frequency

- Scrambled Audio has flipped the original in 0Hz – 7kHz
- Multiply by sin(2π7000) to flip scrambled audio about 7kHz with flipped sideband in 0Hz-7kHz and high-frequency sideband.
- 0Hz-7kHz is double-flipped so back to original audio.
- High-frequency sideband filtered by anti-aliasing filter.

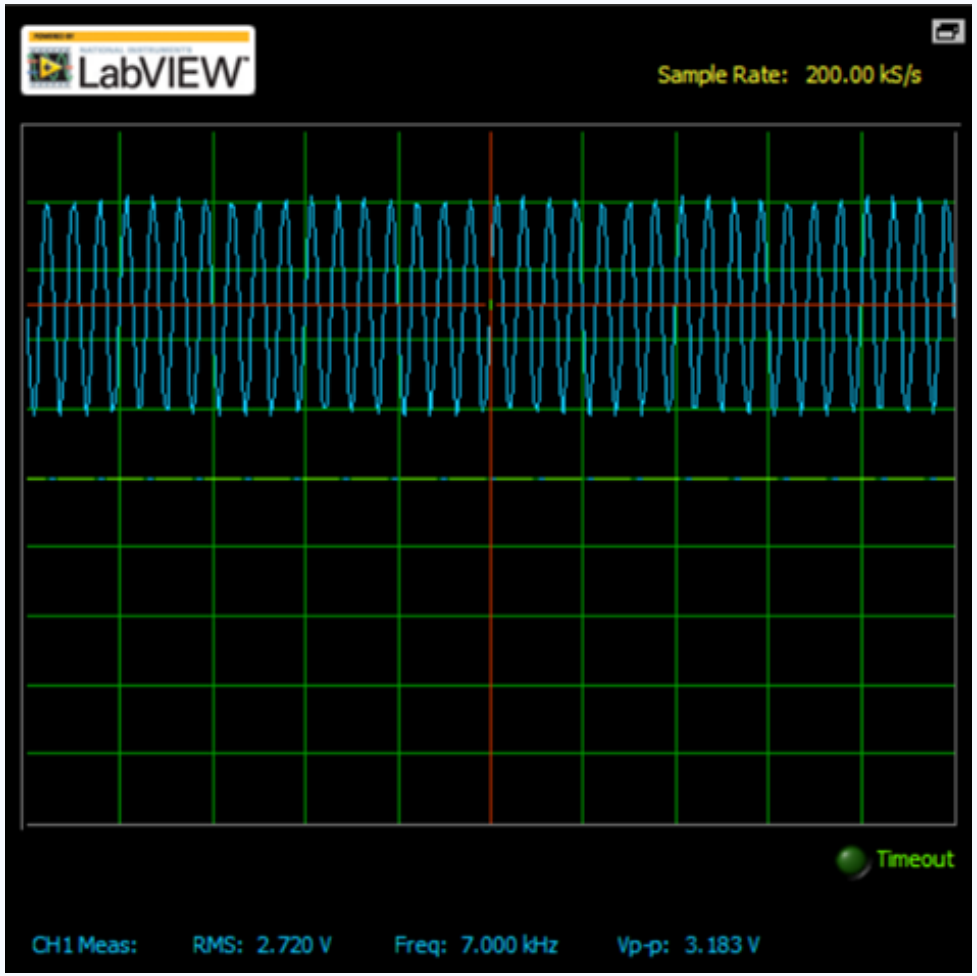
Sine Wave

HOW to make smooth Sine

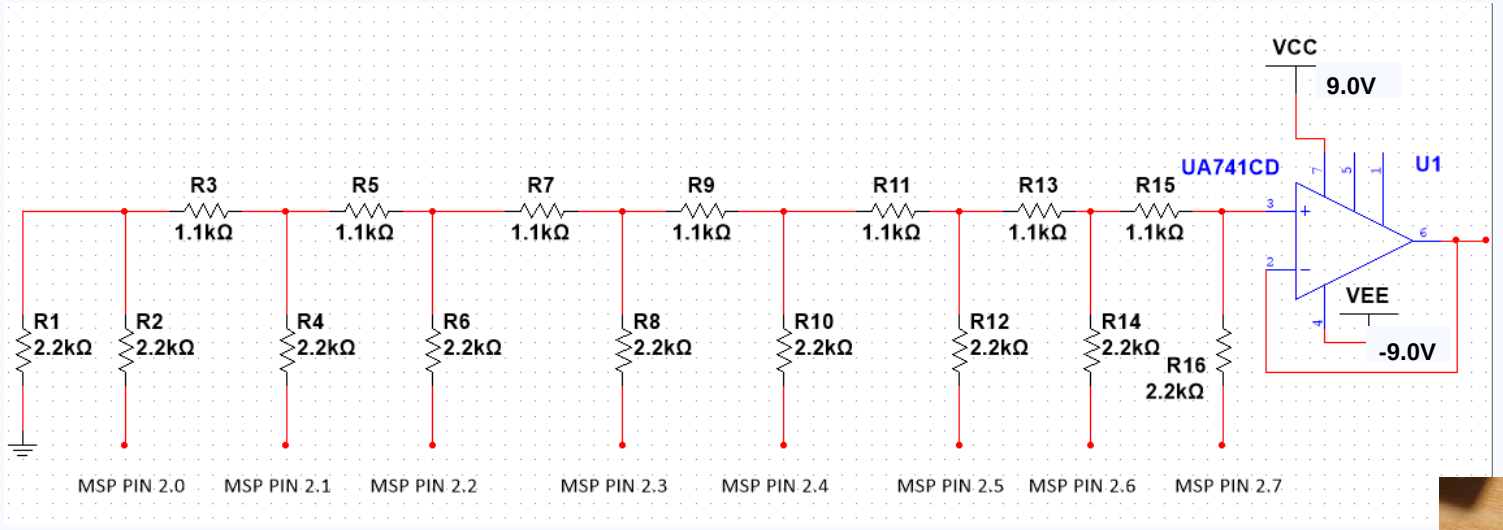
Master Clock (MCLK) = 48kHz and SysTick ISR runs every 960 CLK cycles
--> **SysTick runs at 50kHz sample rate**

Sine multiplies the Bandstop output in SysTick so the **sine must be sampled at 50kHz**

Problem: 7kHz sine frequency does not divide into 50kHz sample frequency
Solution: Using sine and cosine at A=2kHz and B=5kHz as A and B divide into 50kHz!
sin(A+B) = sin(7kHz) = sin(A)cos(B) + cos(A)sin(B)

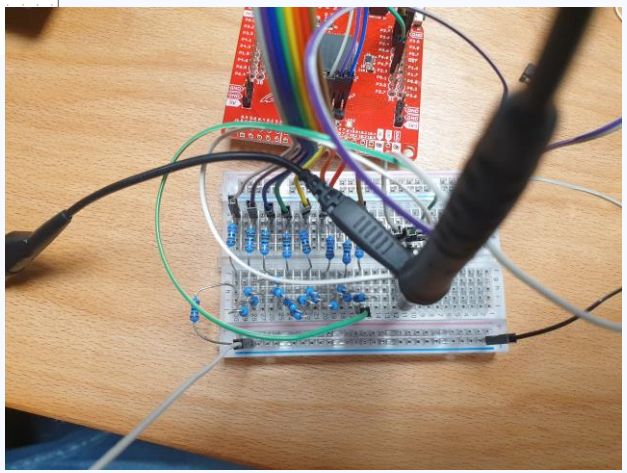


Digital-to-Analogue Converter (DAC) ⟶ R – 2R Ladder:



How it works:

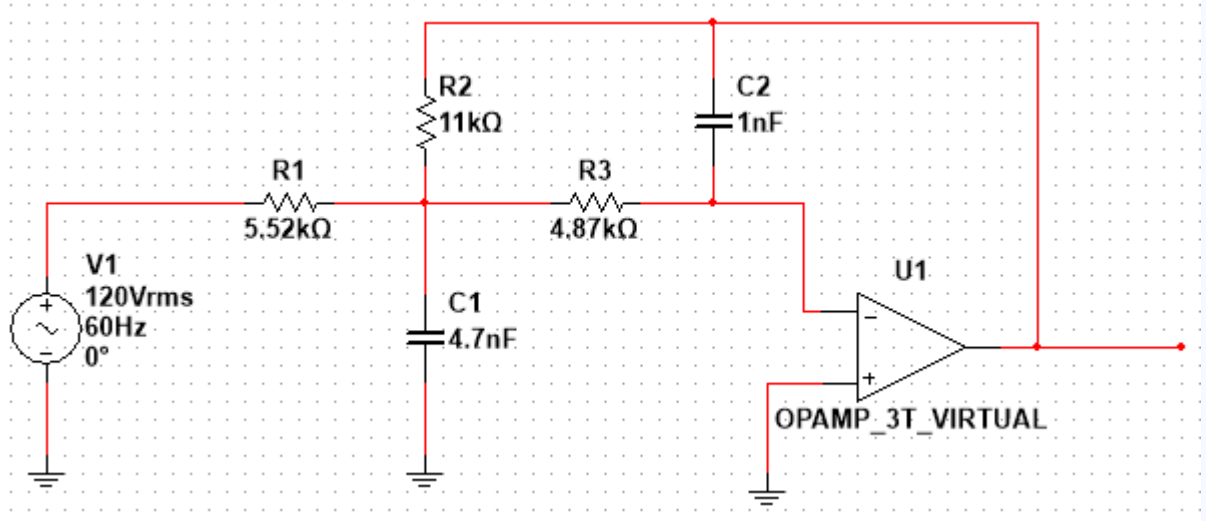
- The 8-bit digital signal from the MSP output is converted into an analogue signal
- Input voltages are applied to the R – 2R Ladder at various points, the MSP Pins.
- More input points results in better resolution
- All the input voltages combine at the end of the R – 2R Ladder
- This drives the amplifier input



ALIASING

Aliasing is also commonly called folding back or mirroring of the input signal.

The anti-aliasing filter is essentially a low pass filter.



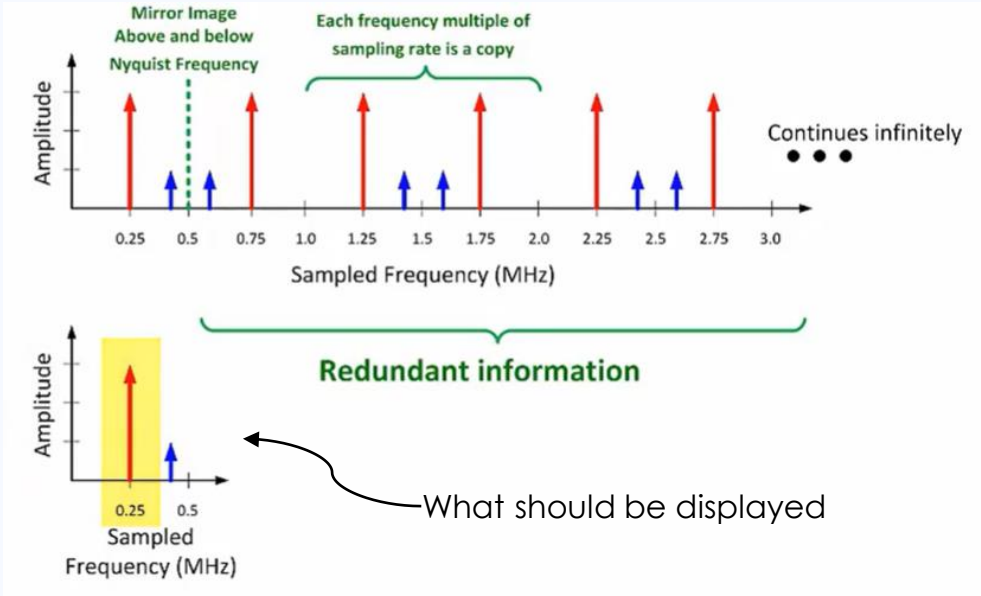


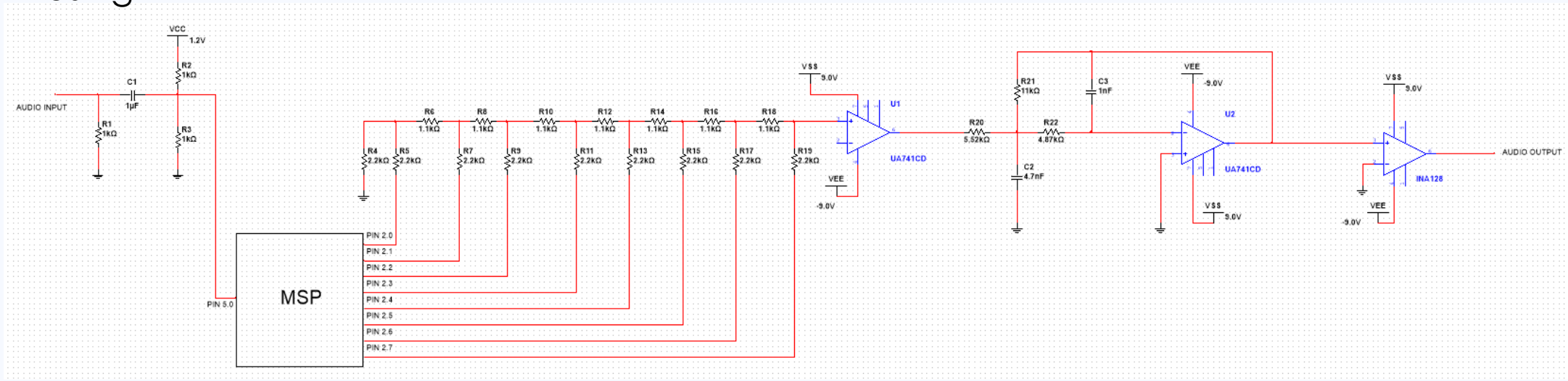
Figure a.

Full Schematic

Potentiometer Biasing

DAC

Anti-Aliasing Filter

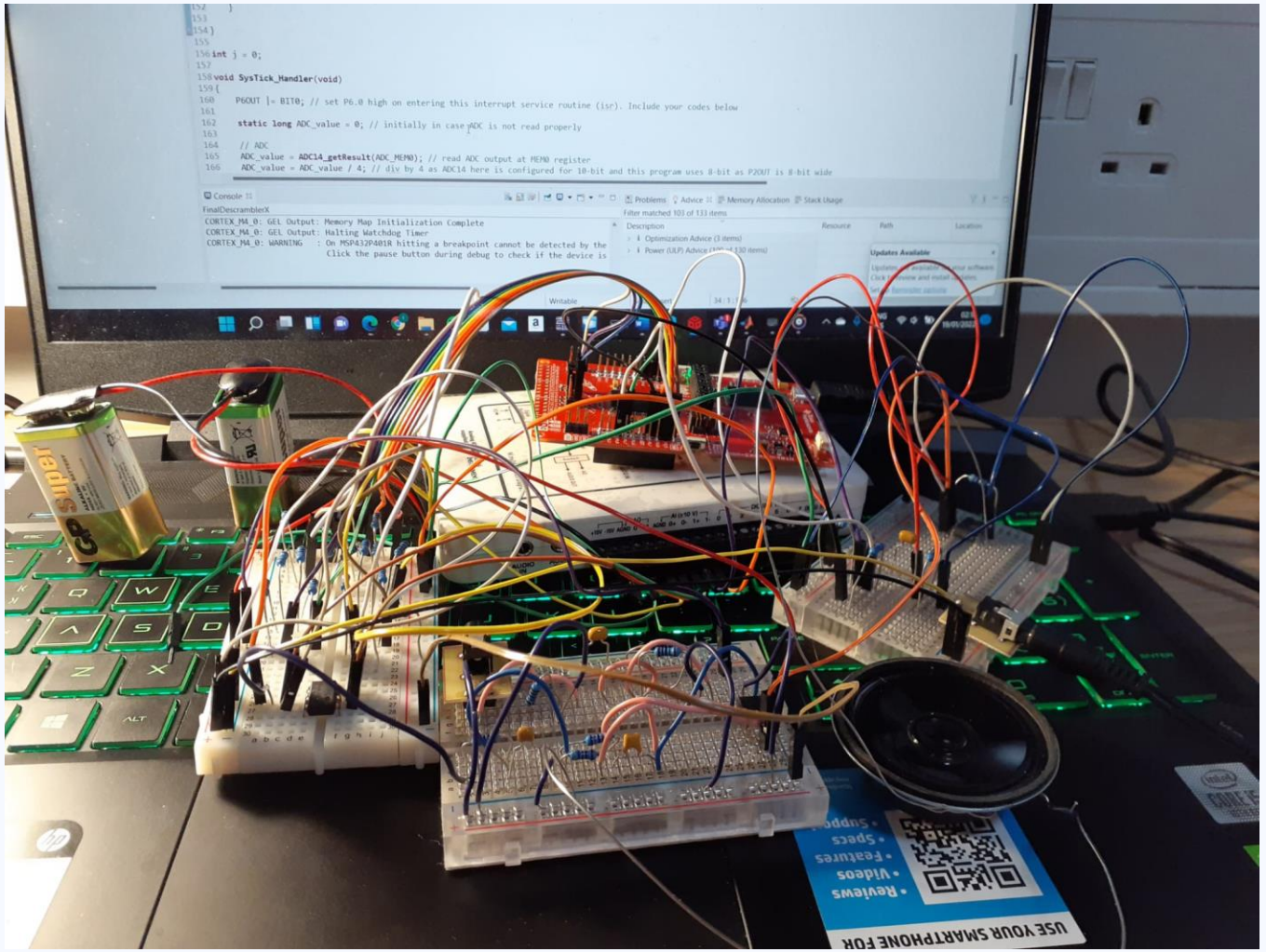


MSP:
- ADC
- Band-stop Filter
- Multiply by Sinewave

Buffer

INA128 Amplifier

Full Built Circuit



Demonstration

References

[1] Figure a: Texas Instruments. (2018, February 20). TI Preciyoutubesion Labs - ADCs: Aliasing and Anti-aliasing Filters [Video]. YouTube. https://www..com/watch?v=T4M6xN-LMWw