CG1111 Engineering Principles and Practice I

Sensors and Applications

(Week 11, Studio 1)

Time	Duration (mins)	Activity
0:00	20	Briefing
0:20	60	Activity #1: Analyzing an Audio Signal and Designing a Passive First Order Low Pass Filter
1:20	60	Activity #2: Designing an Amplifier with an Envelope Detector for Processing the Audio Signal in Arduino
2:20	15	Final discussions and wrap-up

<u>Introduction</u>:

- In this studio, we will be identifying signal and noise in an audio signal, filter out the noise signal and use the audio signal to control real-world application circuits.
- A passive low pass filter is designed with a resistor and capacitor as follows:

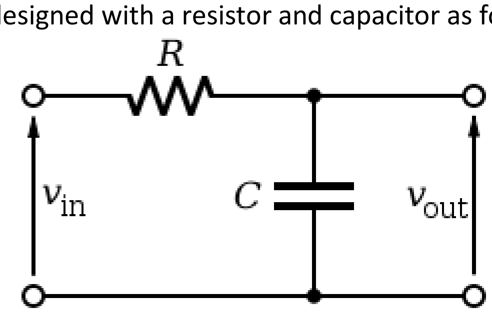


Figure 1: First Order Passive Low-Pass Filter

- In Activity 1, we will be analyzing an audio signal to identify and differentiate the noise with respect to the signal. Subsequently, we will be designing a passive first order low pass filter to filter out the noise.

- In Activity 2, we will be designing an amplifier to amplify the audio signal along with an envelope detector to input to the Arduino's analog to digital convertor. We will be coding the Arduino to perform logical operation based on the audio signal's amplitude.

- An envelope detector is an electronic circuit that takes a high-frequency signal as input (**sound**) and provides an output which is the envelope of the original signal.

- The capacitor in the circuit stores up charge on the rising edge and releases it slowly through the resistor when the signal falls.

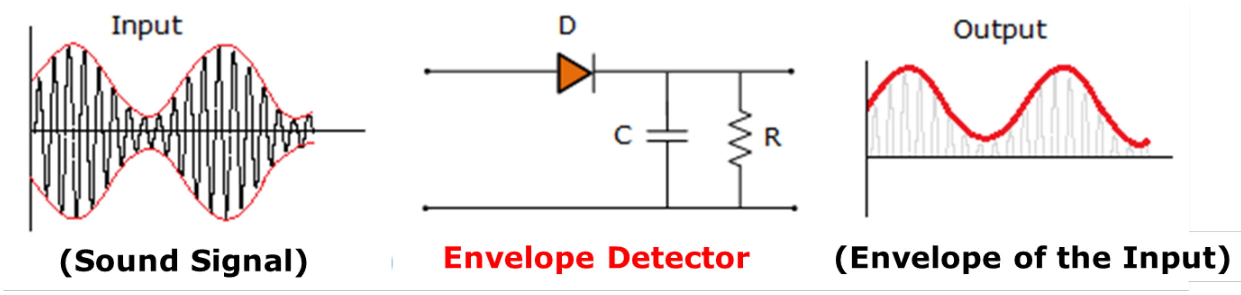


Figure 2: Envelope Detector

Objectives:

- To differentiate signal and noise in an audio signal
- To design a filter to suppress the noise by at-least 10 dB
- To design amplifying circuits to utilize the full voltage-range of the Arduino

Materials:

- BitScope
- Breadboard
- Arduino Uno
- USB Breakout Cable
- 1 X LM358
- 1 X IN4001 diode
- LEDs:
  - 1 x Red
  - 2 x Yellow
  - 1 x Green
- Resistors
  - 1x 470 Ω
  - 4 X 560 Ω
  - 2 X 1 kΩ
  - 1 x 390 kΩ
  - Variable resistor 1 X 100 kΩ
- Capacitors
  - 1 x 100 pF
  - 1 x 2.2 µF (Electrolytic Capacitor)

Setups:

A. Bitscope DSO Setup:

a) Ensure that the "GND" pins of the Bitscope are connected to the common ground terminal of the circuit.

b) In the "Channel Controls (7)" panel, change the voltage per division of "CHA" according to the setup requirements.

c) In the “Capture Control (8)” panel, click on the light brown box at the top and select “MIXED DOMAIN SCOPE”. You should see the main display split into two with the bottom display showing the signal in time domain and the top display showing the spectrum analyzer with the frequency plot of the signal.

d) In the “Timebase Control (6)” panel, change the Timebase to “2ms/Div” and set the Zoom to 1:2.

e) Activate the waveform generator as described in **Setups: B. Waveform Generator Setup** to verify the functioning of the MIXED DOMAIN SCOPE

f) Change the vertical position for both “CHA” to “Mean”.

g) You should be able to see the “RAMP” waveform in the bottom panel and its frequency plot in the top panel.

h) Measure the values of different frequencies (in Hz) and their amplitudes (in decibels) as described in **Setups C. Measurements using Bitscope.**

i) Disconnect the “AWG” pin from “CHA”

B. Waveform Generator Setup:

a) Turn on the waveform generator by clicking on “WAVE” in the “Scope Selectors (2)” panel (see Appendix for its position).

b) In the “FUNCTION GENERATOR” panel (top left), select the type of waveform by right clicking it and selecting “RAMP” (for **triangular** waveform).

c) Adjust the frequency to 1 kHz, peak-to-peak voltage to 400 mV and DC offset voltage to 0V.

d) Connect your “AWG” pin to “CHA” and observe the waveform. The “AWG” pin is the output of your waveform generator.

C. Measurements using Bitscope:

a) Ensure that “MIXED DOMAIN SCOPE” is selected in the Bitscope DSP Setup.

b) Turn on “CURSOR” in the “Scope Selectors (2)” panel. You should have the following four terms under the frequency plot in the Display of the DSO.

- FP: Frequency Point
- MP: Magnitude Point
- FM: Frequency Mark
- MM: Magnitude Mark

c) Ensure that FM is ‘0 Hz’ and MM is ‘0dB’.

d) To note down any specific frequency component, drag the vertical cursor to the frequency that you want to observe and note down the value in Frequency Point (FP).

e) To measure the Magnitude of the corresponding frequency component, drag the horizontal cursor to the peak of the frequency component and note down the value in Magnitude Point (MP).

Activity #1: Analyzing an Audio Signal and Designing a Passive First Order Low Pass Filter (60 mins)

We will be using frequency analysis to identify the noise in the given audio signal. FFT stands for Fast Fourier Transform and is used to identify the frequencies present in any given signal.

1. Follow the steps in Setups: A. Bitscope DSO Setup to setup the spectrum analyzer and observe the 1kHz “RAMP” (triangle) waveform. Explain what you observe in the frequency plot of the waveform.

   Note: You can also observe the “STEP” (square) and “TONE” (sinusoid) waveforms to have a comparison of the different frequency plots.

2. Download the audio signal from the Learning Flow (under Studio Handouts). Listen to the audio signal and check if there is any noise signal present in the audio recording.
3. Connect the circuit as shown in Figure 3 and connect ‘Vout1’ to ‘CHA’.

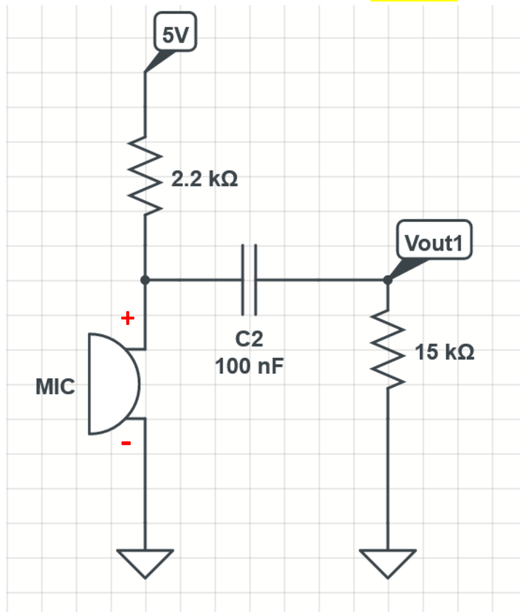


Figure 3: Microphone with the High Pass Filter

4. Play the audio signal on your mobile and hold it over the microphone. Observe the frequency plot in the DSO Spectrum Analyzer. Identify the noise, its frequency and amplitude and save the frequency plot in your learning journal. Also note down the frequency and amplitude of the noise as described in **Setups: Section C. Measurements using Bitscope**:

   Frequency of Noise: ______________________________________
   Amplitude of Noise: ______________________________________

5. We need to suppress this noise by at least **10 dB**. Design a passive low pass filter by calculating the RC value of the filter. **Analyze the passive filter in Figure 1 and show the workings on how you get the RC value in your learning journal**. (Hint: Similar to your Week 9 Tutorial question)
6. Calculate the cut-off frequency of the filter using the RC value and show the workings

7. You can choose either the R or the C value and find the other. For simple calculations it is advised to choose C of 100pF and calculate the value of R correspondingly.

C =______________________,   **R(Calculated)** = ______________________

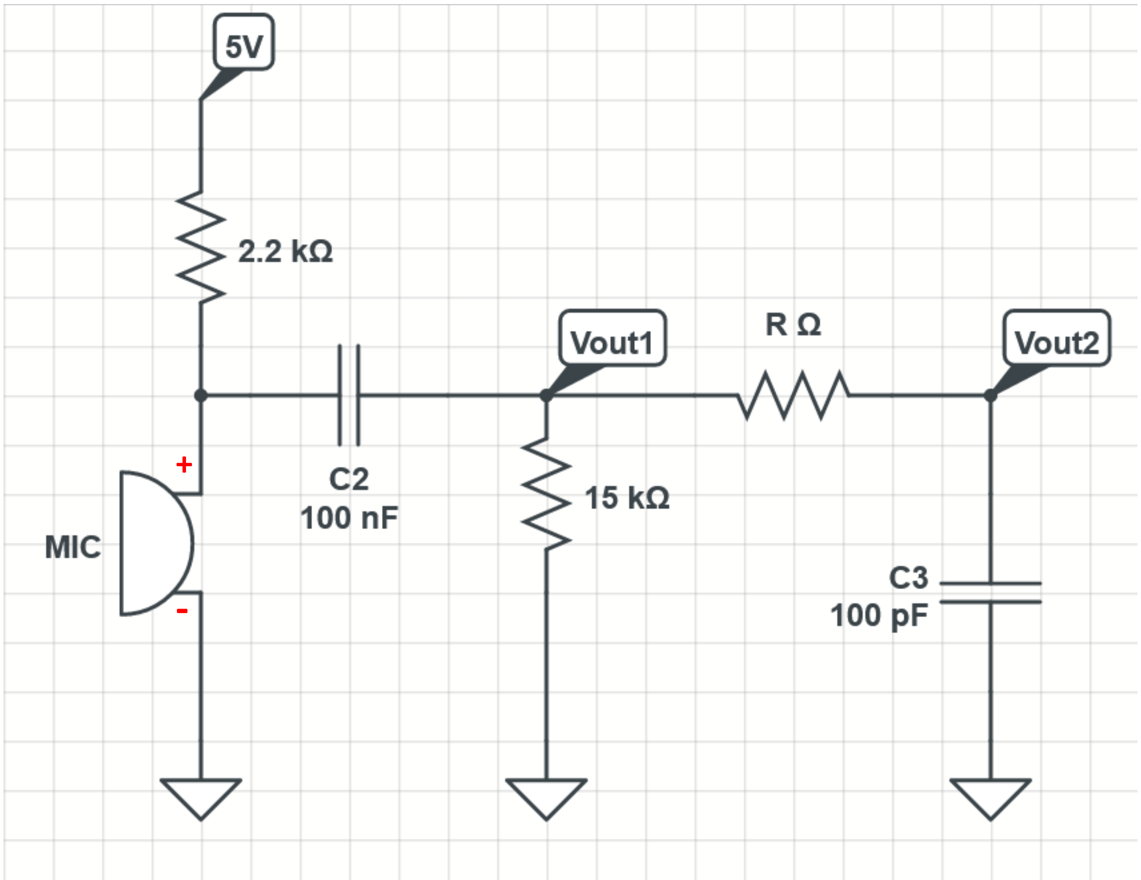


Figure 4: Microphone circuit with the Low Pass Filter

9. Choose the value of R in closest to the calculated Resistor (This should be around 390 kΩ) and connect the circuit as shown in Figure 4. Observe the output waveform by connecting Vout2 to 'CHA' and verify if the noise has been suppressed. You can measure the amplitude of the noise as described in **Setups: Section C. Measurements using Bitscope.** Save the figure frequency plot in your learning journal.
10. Compare and analyze the filtered audio signal with the original audio signal. Note down your findings in the learning journal.

<u>Activity #2: Designing an Amplifier with an Envelope Detector for Processing the Audio Signal in Arduino (60 mins)</u>

In this activity, we will be building an amplifier circuit to amplify the audio signal filtered in Activity 1, before further processing the amplified signal. Our Primary application with amplifying the signal is to match the voltage range of the Arduino's ADC input. The sampled audio signal is used to initiate different control logic in the Arduino.

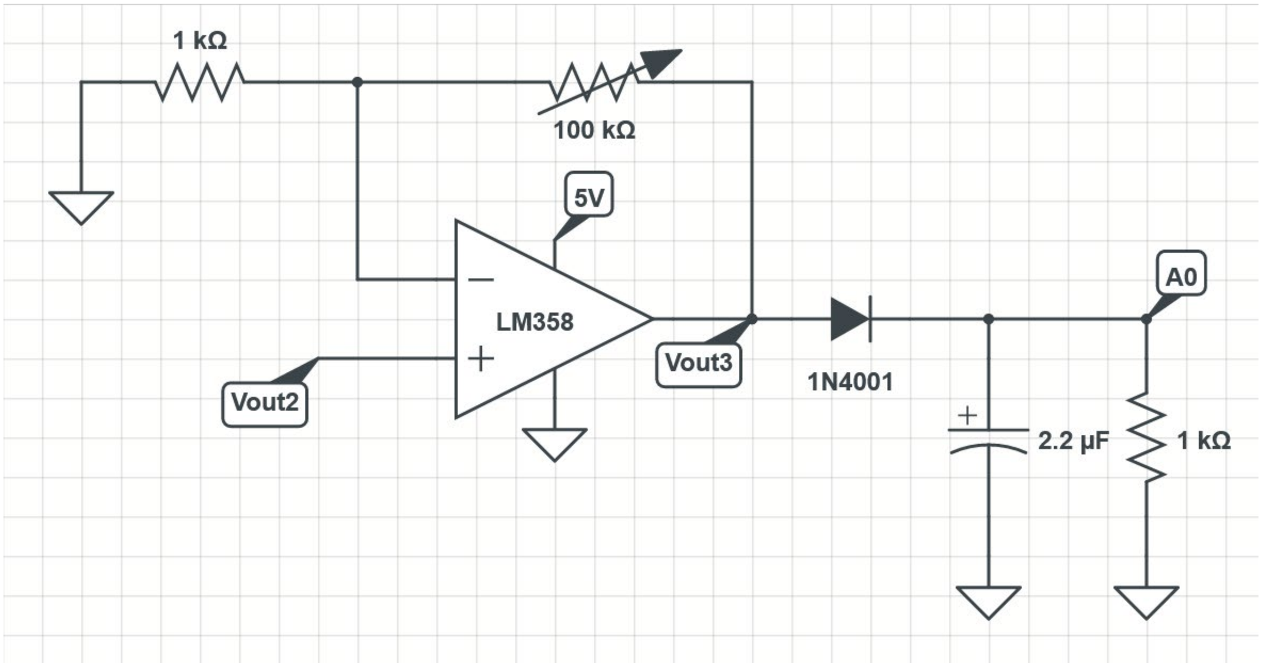


Figure 5: Filter + Amplifier + Envelope Detector. A0 is the analog pin of the Arduino

1. Wire the circuit as shown in Figure 5. We are using the operational amplifier in non-inverting configuration.

   Note: Here we are connecting the op-amp to a single power supply. You can use the USB breakout cable to power up the op-amp

2. The feedback resistor needs to be decided based on the audio input to utilize the full voltage range of the Arduino.
3. Connect 'CHA' to 'Vout3', set the Voltage/Div at 1 V/div, the Timebase to 100ms/s and a zoom of 1:2.
4. Play the audio signal on your mobile and hold it over the microphone. Increase the variable resistor value till the signal is about to saturate. Saturation implies the amplifier outputs a constant voltage (V_{sat}) for **any** given instant of the audio signal in the time domain.

   Note: Since the gain would also depend on the volume of the mobile ensure that you use the same volume throughout the whole studio for the mobile audio out.

5. Remove the variable resistor from the circuit and measure the resistance. Lower the resistance by 1 kΩ and calculate the gain and note down these values in your learning journal. Use this slightly reduced resistance as your feedback resistance.
   Measured Resistance: _______________________________
   Gain for (Measured Resistance – 1 kΩ): ____________________________________

6. Connect the variable resistor back in the circuit ensuring the correct terminals are connected back.

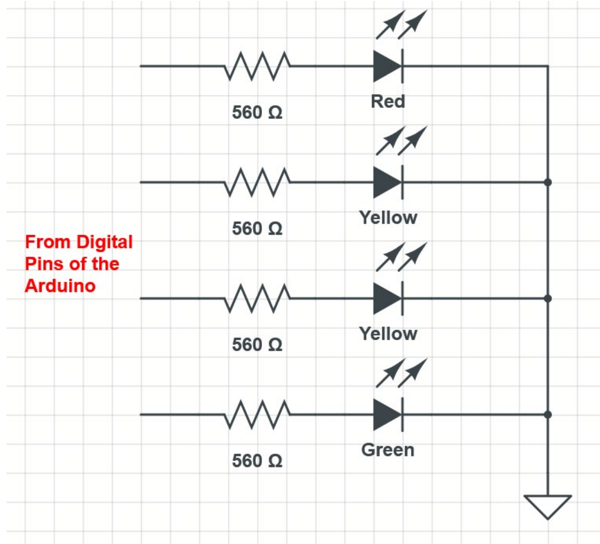


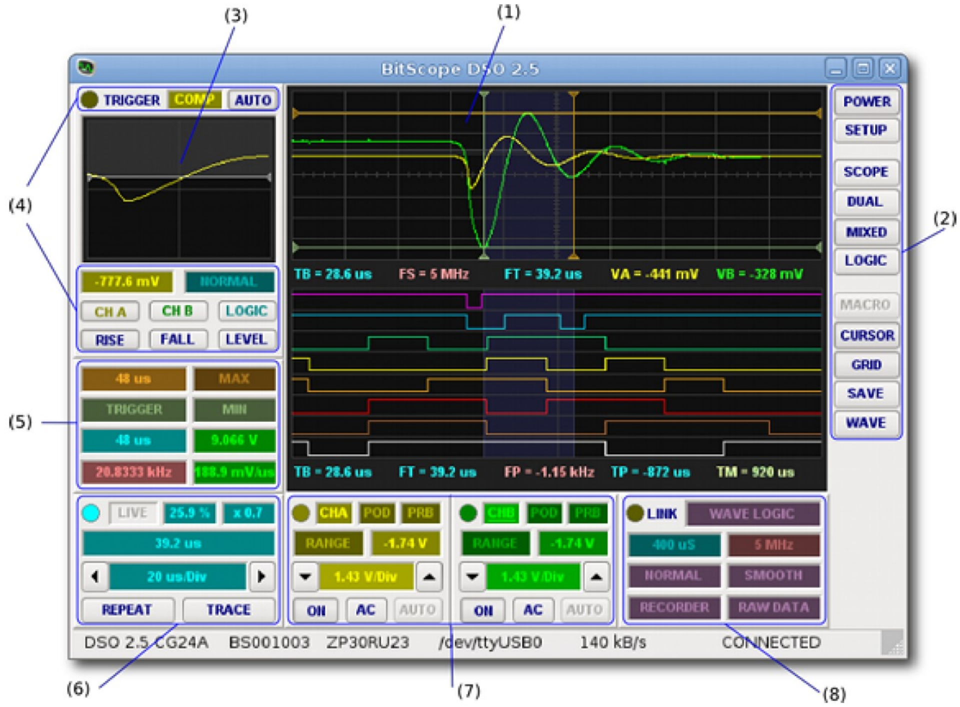
Figure 6: Arduino digital pin output to the LEDs

7. Wire the Arduino with the LEDs and the current limiting resistors according to Figure 6.
8. Connect the output of the Envelope Detector (A0) to any analog pin of the Arduino (Figure 5 says A0 but you are free to choose any other pin).
9. Design the following logic:
    a. Divide the voltage range of the Arduino into four bins. (eg: 0-1.25V, 1.25-2.5V etc…)
    b. Measure the voltage output of the envelope detector in the Arduino.
    c. Switch on only one LED based on the different voltage bins. If one of the LEDs is ON, all the others must be OFF.
    d. The Green LED should be ON for the lowest bin, the yellow LEDs for the middle two bins and the RED LED for the highest bin.
10. Write the code for the corresponding logic in the Arduino IDE, compile and load it into the Arduino. Save a copy of your Arduino code in the learning journal.
11. Observe the LEDs and comment on their working. Does the RED LED turn on? If not, why? How would you rectify this issue? Modify the code to rectify this issue.

Please feel free to ask your TA or the Instructors any questions or doubts you may have.

END OF STUDIO SESSION

APPENDIX: BITSCOPE DSO'S INTERFACE



ID	FEATURE	DESCRIPTION
(1)	Main Display	Waveform, logic and spectrum displays, measurements and cursors.
(2)	Scope Selectors	Virtual instruments, scope tools, presets, cursors, graticule etc.
(3)	Trigger Windows	Shows trigger levels, analog and logic waveforms at the trigger.
(4)	Trigger Controls	Controls trigger setup and displays trigger waveform and data.
(5)	Cursor Measurements	X and Y cursor values, voltage, time and rate measurements.
(6)	Timebase Control	Timebase, Zoom and Time Focus control parameters.
(7)	Channel Controls	Controls input source, range, vertical position and scaling.
(8)	Capture Control	Capture sample rate, duration, frame rate and display modes.