Spectrum Analyzer:

→A **spectrum analyzer** measures the magnitude of an input signal versus frequency within the full frequency range of the instrument.

- The primary use is to measure the power of the spectrum of known and unknown signals. The input signal that most common spectrum analyzers measure is electrical, however, spectral compositions of other signals, such as acoustic pressure waves and optical light waves, can be considered using an appropriate transducer.

By analyzing the spectra of electrical signals, dominant frequency, power, distortion, harmonics, bandwidth, and other spectral components of a signal can be observed that are not easily detectable in time domain waveforms. These parameters are useful in the characterization of electronic devices, such as wireless transmitters.

The display of a **spectrum analyzer** has *frequency* displayed on the horizontal axis and the *amplitude* on the vertical axis. Unlike an **Oscilloscope** which plots *amplitude* on the vertical axis but *time* on the horizontal axis.

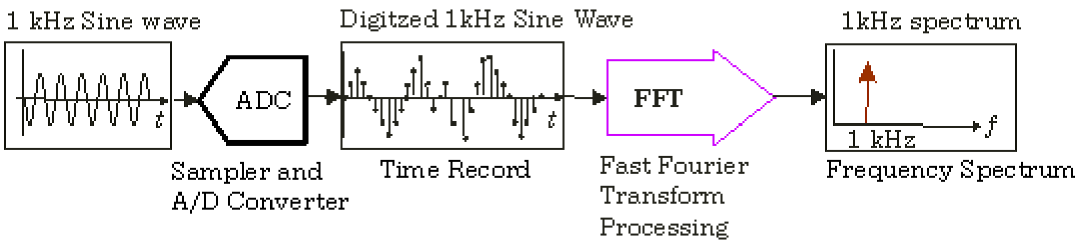
Types:

→Spectrum analyzer types are distinguished by the methods used to obtain the spectrum of a signal. There are swept-tuned and fast Fourier transform (FFT) based spectrum analyzers.

But in our project, we will focus on FFT type based spectrum analyzers.

- FFT analyzer:

computes a time-sequence of periodograms. *FFT* refers to a particular mathematical algorithm used in the process. This is commonly used in conjunction with a receiver and analog-to-digital converter. As above, the receiver reduces the center-frequency of a portion of the input signal spectrum, but the portion is not swept.



- FFT analyzers can process all the samples (100% duty-cycle), and are therefore able to avoid missing short-duration events.

The frequency resolution is $\Delta v = \frac{1}{T}$, the inverse of the time T over which the waveform is measured and Fourier transformed, with Fourier transform analysis in a digital spectrum analyzer,

➔ it is necessary to sample the input signal with a sampling frequency v_s that is at least twice the bandwidth of the signal, due to the Nyquist limit, a Fourier transform will then produce a spectrum containing all frequencies from zero to $\frac{Vs}{2}$. This can place considerable demands on the required analog-to-digital converter and processing power for the Fourier transform.

Example:

For a 16 MHz Arduino nano the ADC clock is set to 16 MHz/128 = 125 KHz. Each conversion in AVR takes 13 ADC clocks so 125 KHz /13 = 9615 Hz. That is the maximum possible sampling rate, and the A/D on the Arduino nano has a resolution of 10 bits, assuming the time window is 4 seconds.

Then we can calculate the number of bits and samples required as follows:

1) $frequency\ resolution = F_r = \frac{1}{4} = 0.25\ Hz.$

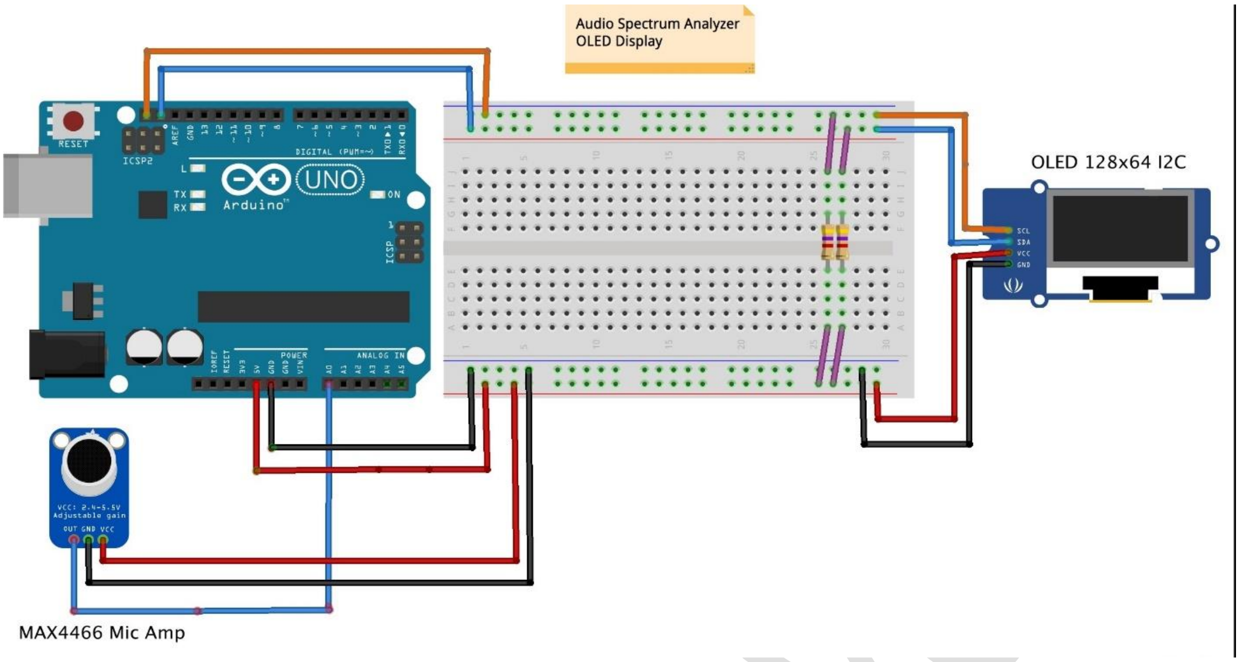
2) $upper\ frequency\ calculated = F_s = \frac{9615}{2} = 4807.5\ Hz\ .$

3) $Since\ 10\ bit\ then\ dynamic\ range = \ R_d = 20log2^{10} = 60\ dB.$

4) Then we can get sample number = $4\ x\ 9615 = \ 38460Hz = 38.4\ Khz.$

5) Finally, Number of bits $= 38460\ x\ 10 = \ 384600\ bits = 384.6\ Kbs.$

Project Schematic:



Components used:

- Arduino Nano or Uno x1
- SSD1306 OLED Display 0.96 inch x1
- Breadboard x1
- Microphone Module (MAX9814) x1
- Male to male wires

Project Theory:

The project idea is to create a crude fixed fast Fourier transform spectrum analyzer using an Arduino library called (“fix_fft.h”).

- The library’s main and only function is that it takes 4 arguments that begins the Fourier transform calculations, that also returns the result and stores them in the same arrays (Real & Imaginary arrays).

→Since our display is 128x64 pixel resolution, then our Real and imaginary array is going to be set to the maximum height of the screen which is 128.

→The input signal will be analog from a microphone module (MAX9814).