

Interface Definitions

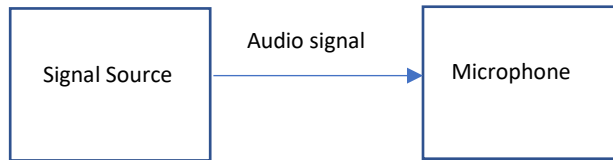


Figure 1: Processing AC Signal

Interface Name	Interface Type	Specifics
Signal Source	Source of the signal (ex. Phone)	<ul style="list-style-type: none"> - Any signal from the outside environment - Best signal would be from a phone or another processor with music
Audio Signal	Traveling through air or a channel	<ul style="list-style-type: none"> - Starting AC signal from the source - This signal is traveling through air and gets picked up by the mic
Microphone	Analog signal processor	<ul style="list-style-type: none"> - Operating voltage: 2V to 10V - Operating frequency: 100 to 20,000 HZ - Sensitivity: -42 ± 3 dB - Operating Temperature: -20 to 70° C



Figure 2: Amplifying the Signal

Interface Name	Interface Type	Specifics
Raw Audio Signal	Analog Signal from the Mic	<ul style="list-style-type: none"> - Voltage Range peak to peak: 30 to 70 mV
Amplifier	Op Amp	<ul style="list-style-type: none"> - Offset voltage 3mV max over temp - Input current 100nA max over temp - Offset current 20nA max over temp - Has twice a gain of the non-inverting input signal

Amplified Signal	Analog Signal increased with gain	<ul style="list-style-type: none"> - AC Signal that is the output of the Amplifier. - Voltage range peak to peak: 60 to 200mV - Doubled the Raw Audio signal
------------------	-----------------------------------	---

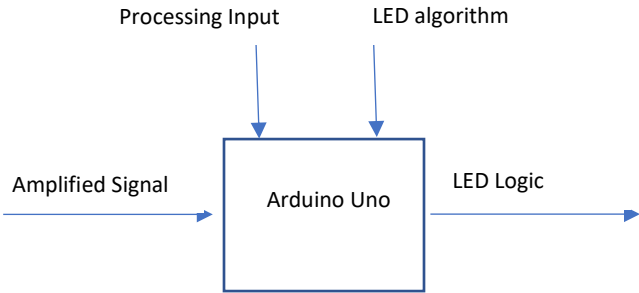


Figure 3: Arduino Uno coding process

Interface Name	Interface Type	Specifics
Amplified Signal	Analog Signal increased with Gain	<ul style="list-style-type: none"> - AC Signal that is the output of the Amplifier. - Voltage range peak to peak: 60 to 200mV - Doubled the Raw Audio signal
Arduino Uno	Microprocessor	<ul style="list-style-type: none"> - Data rate: 9600 baud rate - Logic: 5 Volts - Min Current: 2 uA - Word size: 8 bit
Processing Input	Code	<ul style="list-style-type: none"> - Arduino software: Arduino IDE - Written in C++ - Uses Arduino library: fix_fft.h - Takes input analog signal and transforms it into data
LED algorithm	Code	<ul style="list-style-type: none"> - Arduino software: Arduino IDE - Written in C++ - Uses Arduino library: fix_fft.h - Uses the data from the processing Input and turns that into an algorithm to control the LEDs as outputs

LED Logic	Arduino output Signal to LEDs	<ul style="list-style-type: none"> - Max output voltage: 5V - Sends commands to control the patterns of when the LEDs turn on - Uses the LED algorithm to command the LEDs
-----------	-------------------------------	---

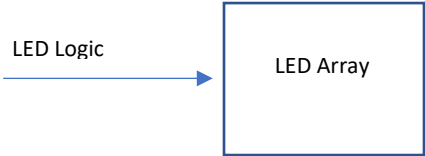


Figure 3: Amplifying the Signal

Interface Name	Interface Type	Specifics
LED Logic	Arduino output Signal to LEDs	<ul style="list-style-type: none"> - Max output voltage: 5V - Sends commands to control the patterns of when the LEDs turn on - Uses the LED algorithm to command the LEDs
LED Array	3 Bands of LEDs	<ul style="list-style-type: none"> - Three bands of 4 LEDs - Size of LED: 5mm - Reverse Voltage: 5V - Peak Forward Current: 1A - Power dissipation: 180mW