Music Synthesizer[a)]

Neil Baker[b)]

Synopsis

The aim of this project is to build a piece of equipment that is capable of taking an input note, auto tuning it, and repeating it with added embellishments through a speaker.

I. PLAN OF IMPLEMENTATION

Concerns regarding the storage space of the standard Arduino have led us to acquire an Arduino Mega, which we feel is more appropriate to the task at hand. "ApproxFFT" code from the web will fourier-transform the analogue microphone signal into a digital frequency. This will take a lot of storage space, since I have to store each sample from the microphone to use the FFT function on it. I will map certain frequencies to certain notes, allowing the program to auto-tune the input frequency. I will then send the frequencies out to analogue using the "tone" library provided by Arduino. I will be able to reproduce input tones, as well as transpose or change tones as desired. Once the basic functionality of the device is achieved, I will attempt to layer sound by adding harmonizing notes onto the top of an input sequence.

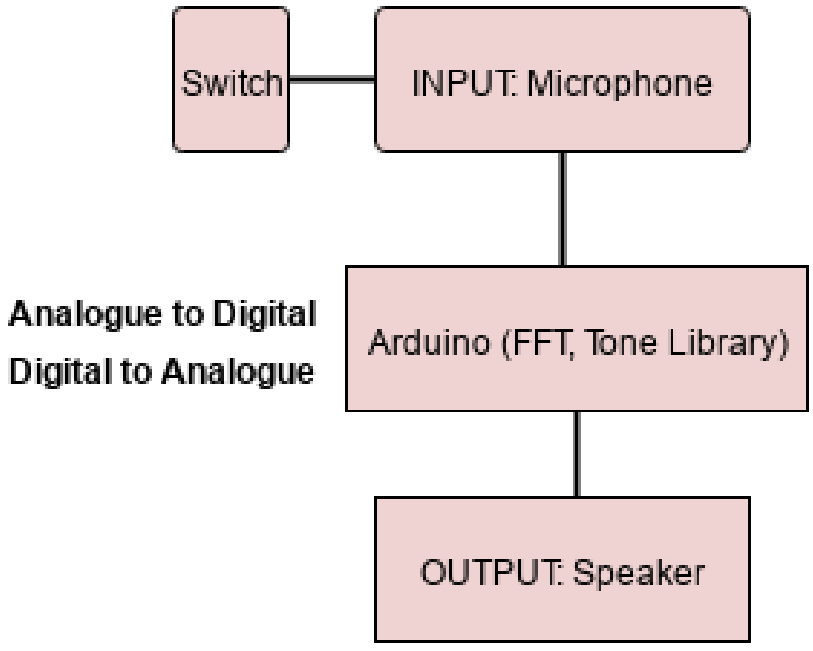

FIG. 1. Block diagram of the synthesizer.

[a)]Thank you to Professor Larry Sulak and the ELab staff at Boston University
[b)]Physics Department, Boston University.

II. LIST OF PARTS

- Electret Microphone Amplifier Module MAX4466
- Arduino Mega
- Speaker (black box or external)
- Switch
- Wires and circuit components

III. RESULTS, ISSUES AND FURTHER WORK

I was successful in the aim of reproducing a tone, but that tone does not always match the input sung note. I embellished the output note with a rising arpeggio. The major issue is the microphone sampling rate, which is limited by the memory space available on the Arduino Mega. The FFT function uses data list of 2^n, and 2048 is too large to all fit in memory. So I have to use 1024 microphone samples, which means I have to have a very small recording time lest the sampling rate be too slow. I have compromised such that the sampling rate is about 1000Hz and the microphone takes data for about a second. This is not the most user friendly solution, but it yields output in the mid range that are within about a tone of the input sung note - i.e., the FFT function is not precisely accurate, but it isn't bad. The higher the input note is, the less likely the output will be good, since the frequency of higher notes is higher and will approach or even surpass the sampling rate.

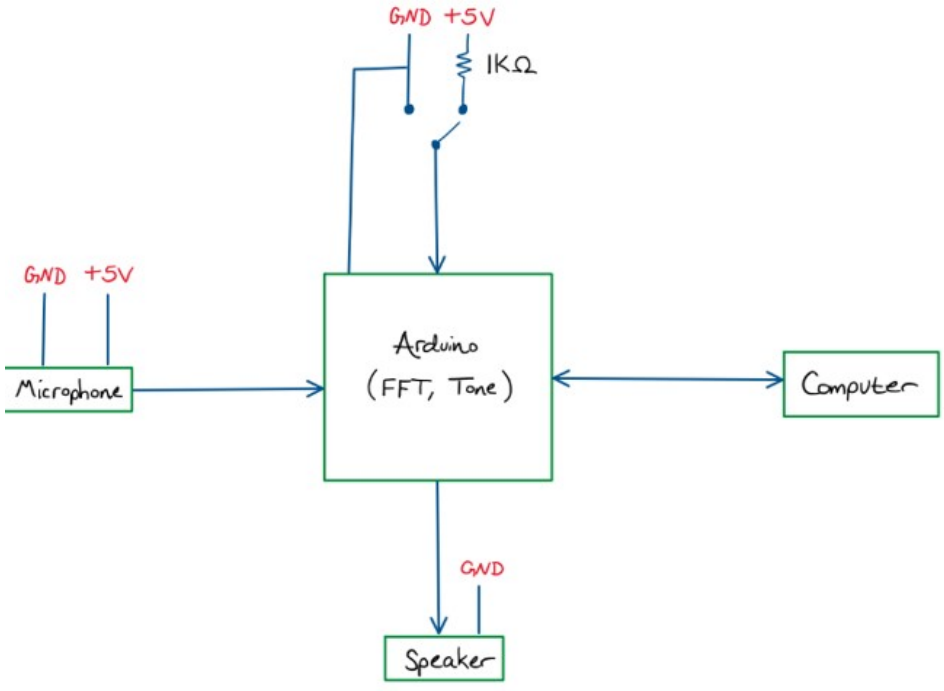

FIG. 2. Circuit diagram of the synthesizer.

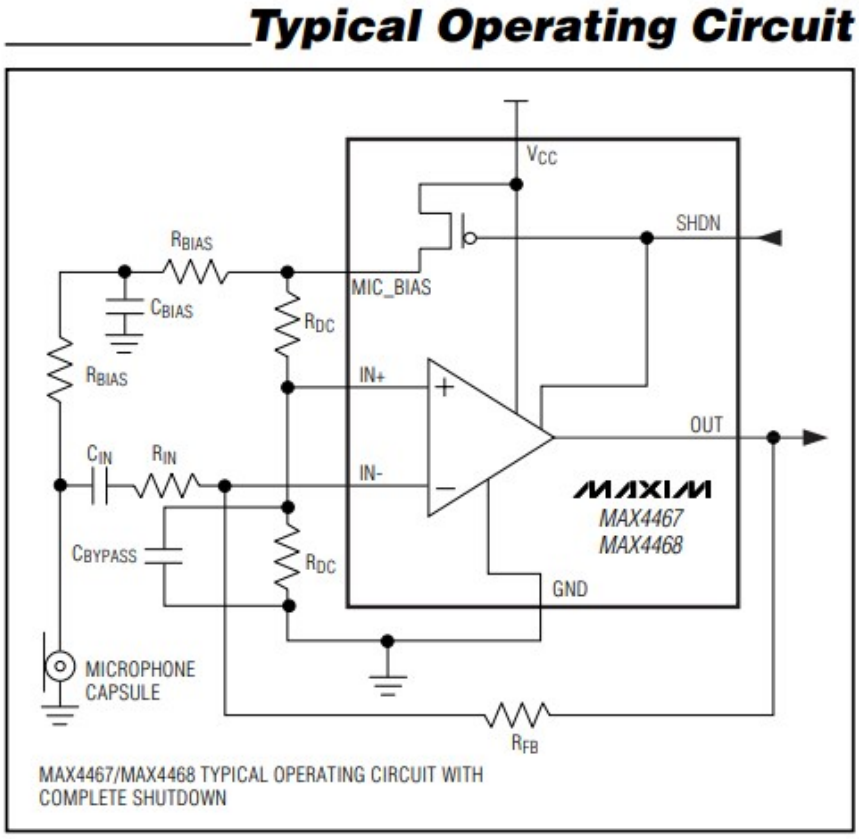

FIG. 3. Circuit diagram of the microphone.