Digitally-Controlled Analog Filter Prototype for Wireless Communication Applications

Jay Park, YSP Student, *Newton South High School*
David Abrahamyan, YSP Student, *Watertown High School*
Mentors: Gaurav Jha, M.S. Student, *Northeastern University*
Mahmoud Ibrahim, Ph.D. Student, *Northeastern University*
Professor Marvin Onabajo, *Electrical & Computer Engineering Department, Northeastern University*

Northeastern University Center for STEM Education

Northeastern University College of Engineering

Abstract

Analog circuits play an important role in devices for the wireless transmission and reception of information. They have to be designed with high performance and reliability. However, the quality of analog circuits depends strongly on variations of electronic component parameters and manufacturing processes. A general research approach aimed at improving analog circuits in the presence of random variations involves the measurement of key performance parameters in combination with automatic tuning for optimum performance through the aid of circuits in the same system. To allow the tuning of analog circuits with digital control, programmable elements have to be incorporated into the analog circuits. This approach was investigated in this research project by prototyping a digitally controllable filter circuit for extraction of the desired information in a wireless receiver while suppressing undesired interference signals.

Introduction

Analog filters are part of wireless receivers, and they are essential in blocking undesired signals from third party sources while allowing essential signals to pass through. A filter is designed to maintain or increase the power levels at desired frequencies, while reducing the power of interfering signals at other frequencies. This reduction of power is called attenuation. To accomplish this specific goal, we programmed an Arduino Uno microcontroller to digitally adjust the bandwidth (desired frequency range) for optimum filtering. We used a prototype board to assemble the electrical filter components. The Arduino language was used to program the microcontroller. In our experiments, we set the optimal cutoff frequency to 1 MHz.

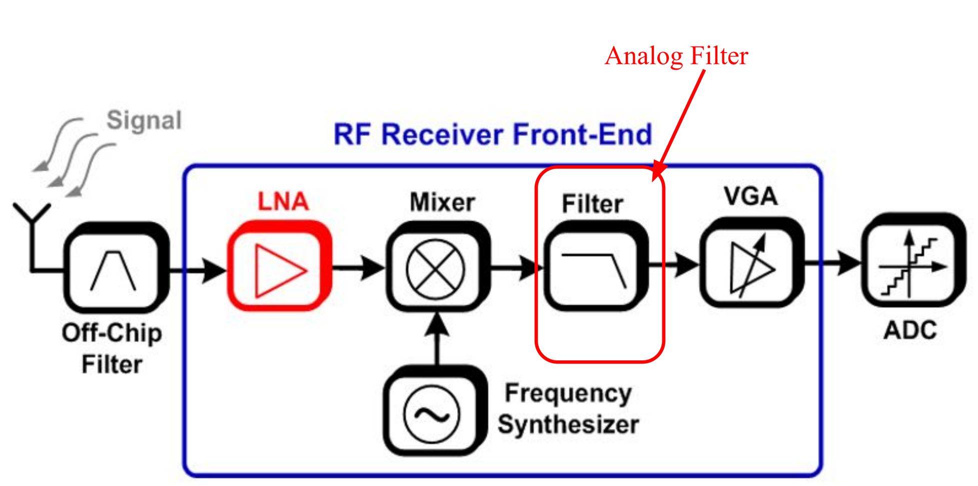

Figure 1: Radio Frequency Receiver Front-End

Development of the Filter Circuit with 1 MHz Cutoff Frequency

The creation of the circuit consisted of three main stages of development:

1. Simulation
The circuit was created in LTspice to simulate it and to predict how different component parameters affect the signal output.

2. Floorplan
A sketch of the final circuit was created to scale to evaluate how all of the parts could fit, while minimizing the length of the wires.

3. Assembly
The components were soldered onto the prototype circuit board, and connected through soldering of wires.

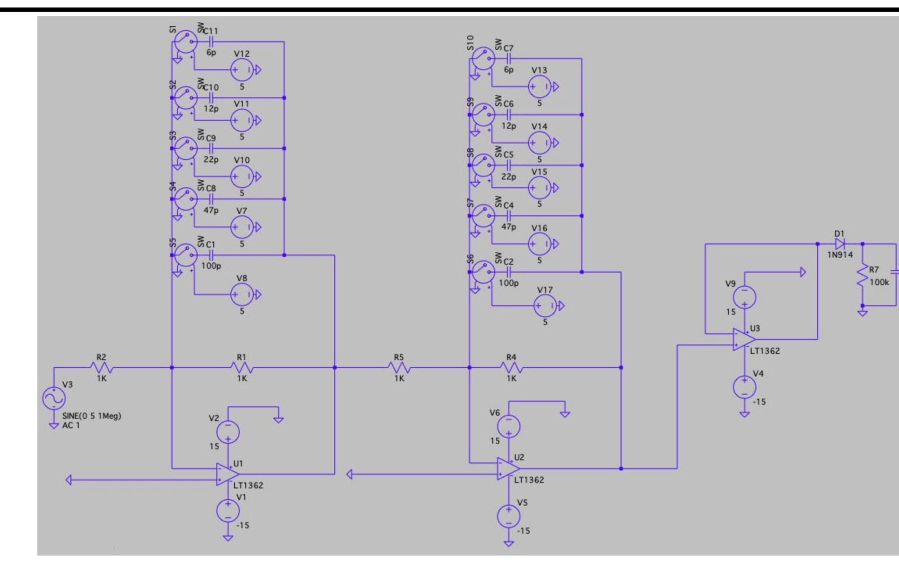

Figure 2: LTspice Simulation Schematic

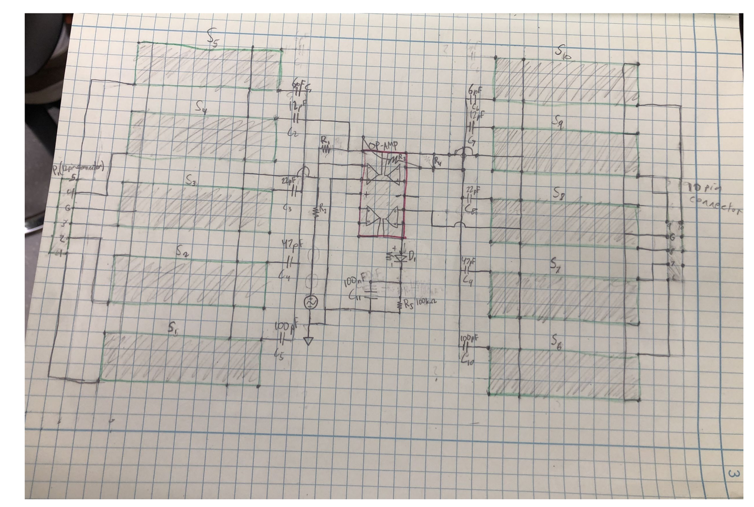

Figure 3: Draft Plan for Soldering

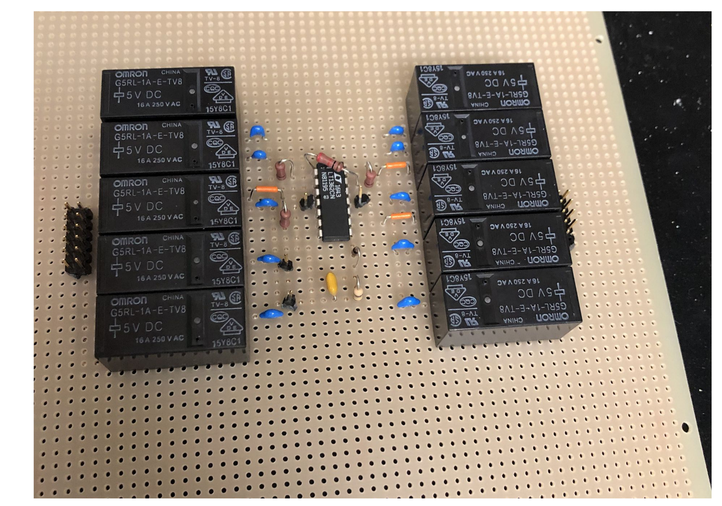

Figure 4: Finished Circuit on a Prototype Board

Development of the Calibration Software

An Arduino Uno microcontroller was programmed to control the filter cutoff frequency based on measured voltages.

1. **Reading of the Analog Voltage**
   A rectifier is used to convert the AC output voltage to a DC voltage. At the beginning of the Arduino program, the rectifier voltage is detected to determine V_{MAX}, which is obtained by outputting the code to produce the lowest capacitance value (i.e., highest cutoff frequency). Afterwards, the program incrementally adjusts the control code until the input voltage is equal to $V_{MAX} / \sqrt{2}$, the value that corresponds to a cutoff frequency of 1 MHz.
2. **Adjusting of the Cutoff Frequency Settings**
   The Arduino was programmed to adjust the cutoff frequency by using capacitor banks with digitally programmable values. According to simulations, increasing the capacitance values reduces the cutoff frequency. The Arduino program cycles through a combination of capacitance values, incrementally decreasing the cutoff frequency until the desired 1 MHz cutoff frequency is set.
3. **Selection of Permutations**
   The capacitor permutations were selected based on simulations with the LTSpice software, an electronic circuit simulator.

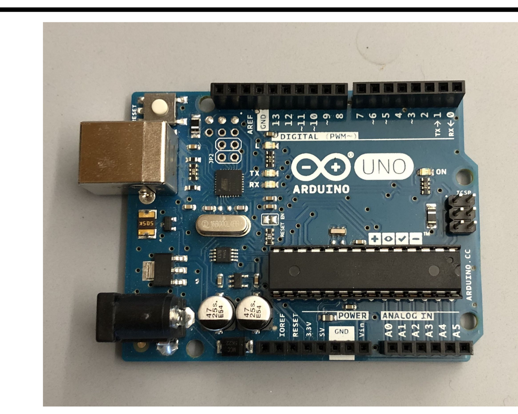

Figure 5: Arduino Uno Microcontroller

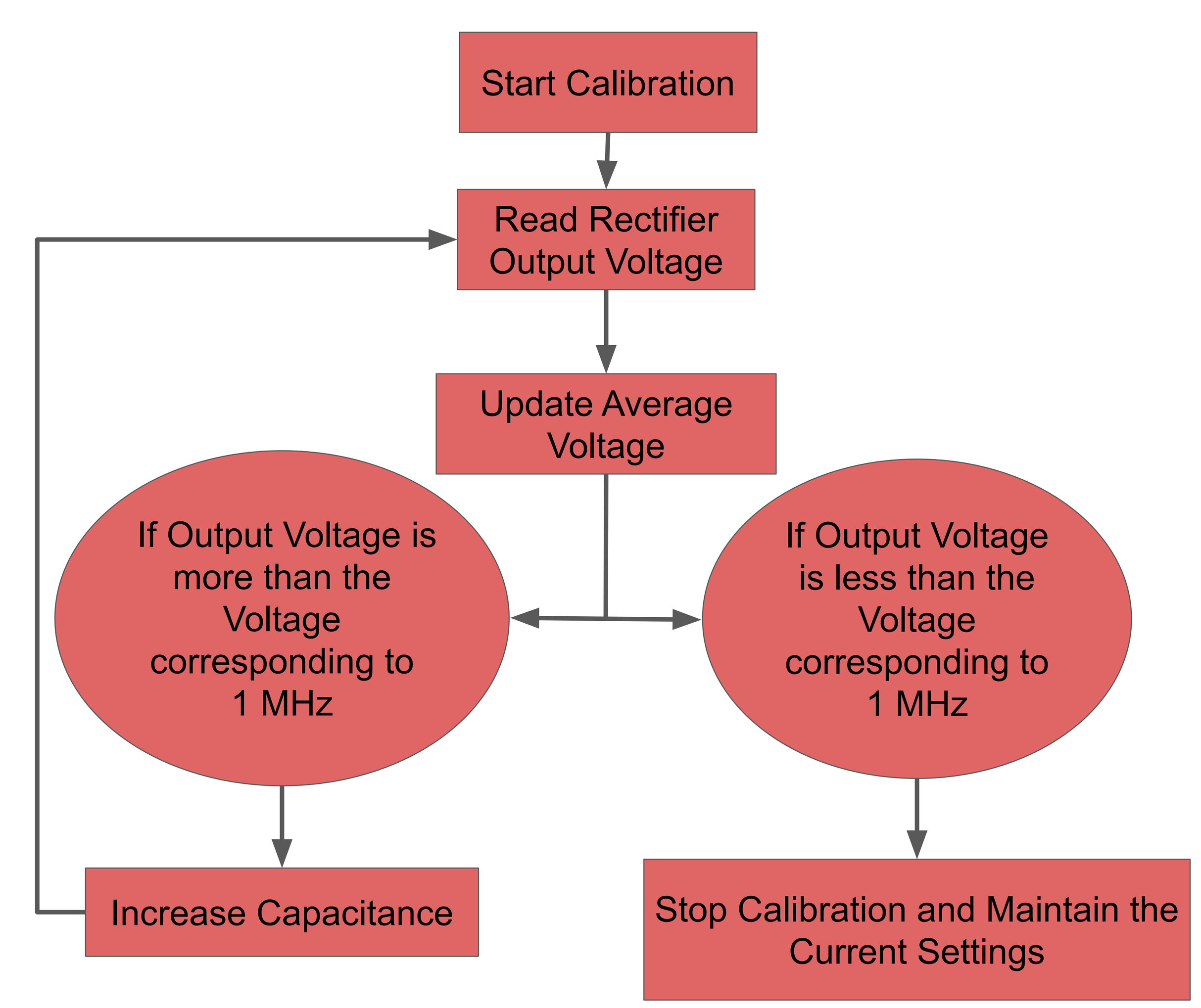

Figure 6: Flowchart of the Calibration Program

Results

Arduino Code Verification

- A DC power supply emulated rectifier output
- The DC voltage was adjusted manually
- The Arduino program stopped the calibration once $V_{MAX} / \sqrt{2}$ was detected

Filter Frequency Response

- Ideal parameters were found for the components using LTspice simulations
- The filter schematic was simulated to confirm that it has a 1 MHz cutoff frequency
- Some errors in the prototype board remain unknown, and require further debugging

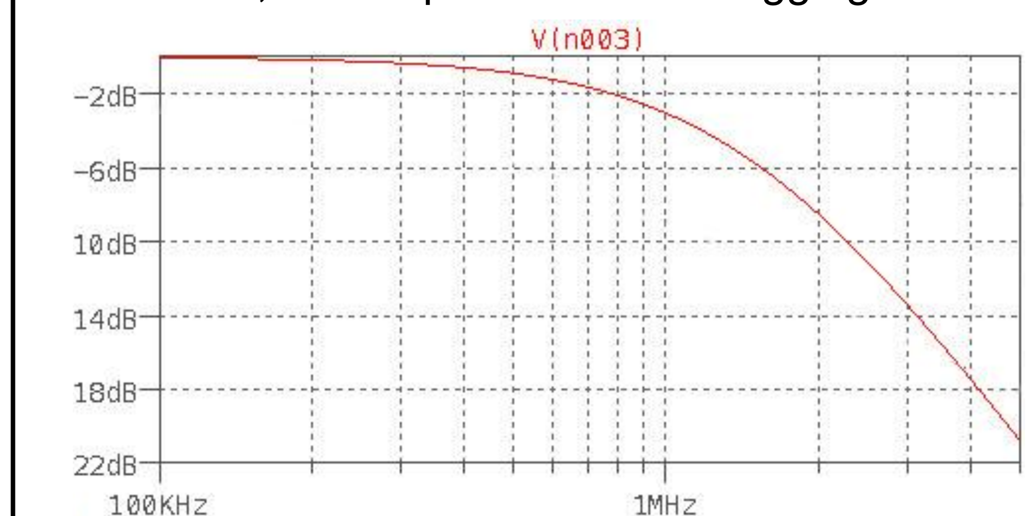

Figure 7: Simulated Filter Frequency Response

Future Work

In order to allow for finer tuning, more relays can be added and capacitors with smaller values can be used. The measurement delay during calibration can be modified more to make the tuning faster. The filter has to be debugged and connected with the Arduino to verify the automatic calibration for cutoff frequency correction.

Acknowledgements

Mahmoud Ibrahim - ASMIC Research Laboratory Ph.D. Student and Mentor
Gaurav Jha - ASMIC Research Laboratory M.S. Student and Mentor
Professor Marvin Onabajo - Dept. of Electrical and Computer Eng. Northeastern University
Claire Duggan - Director, Northeastern University - Center for STEM Education

This work was supported by the National Science Foundation Grant #1451213