Operation USSR
(Ultra Secret Signaling for Reconnaissance)

The Goal of the Project

Our goal of this project was to implement a single frequency tone generator and decoder.

DTMF (Dual Tone Multi-Frequency) signaling is used for telecommunication signaling between telephone handsets and switching centers over analog telephone lines in voice-frequency bands.

Description of the Model

- Dual-Tone Multi-Frequency Signaling (**DTMF**) is a system by which audible tones are used to represent buttons being pressed on a keypad.
- The Discrete Fourier Transform (**DFT**) is used at the receiving end to determine the button pressed. It converts a signal from the time domain representation to its representation in the frequency domain.
- In our implementation, we transmitted frequencies over sound waves (as opposed to a phone line).

DTMF Keypad Frequencies

Frequency	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

Frequencies Used

1 2100 Hz	2 2300 Hz	3 2500 Hz
4 2700 Hz	5 2900 Hz	6 3100 Hz
7 3500 Hz	8 3700 Hz	9 3900 Hz
* 1500 Hz	0 1900 Hz	# 3900 Hz

DFT Equation for Frequency Detection

The Fourier Transform is found for the signal input from the microphone. Since sound waves are quite noisy, the relative magnitudes are compared rather than analyzing peaks.

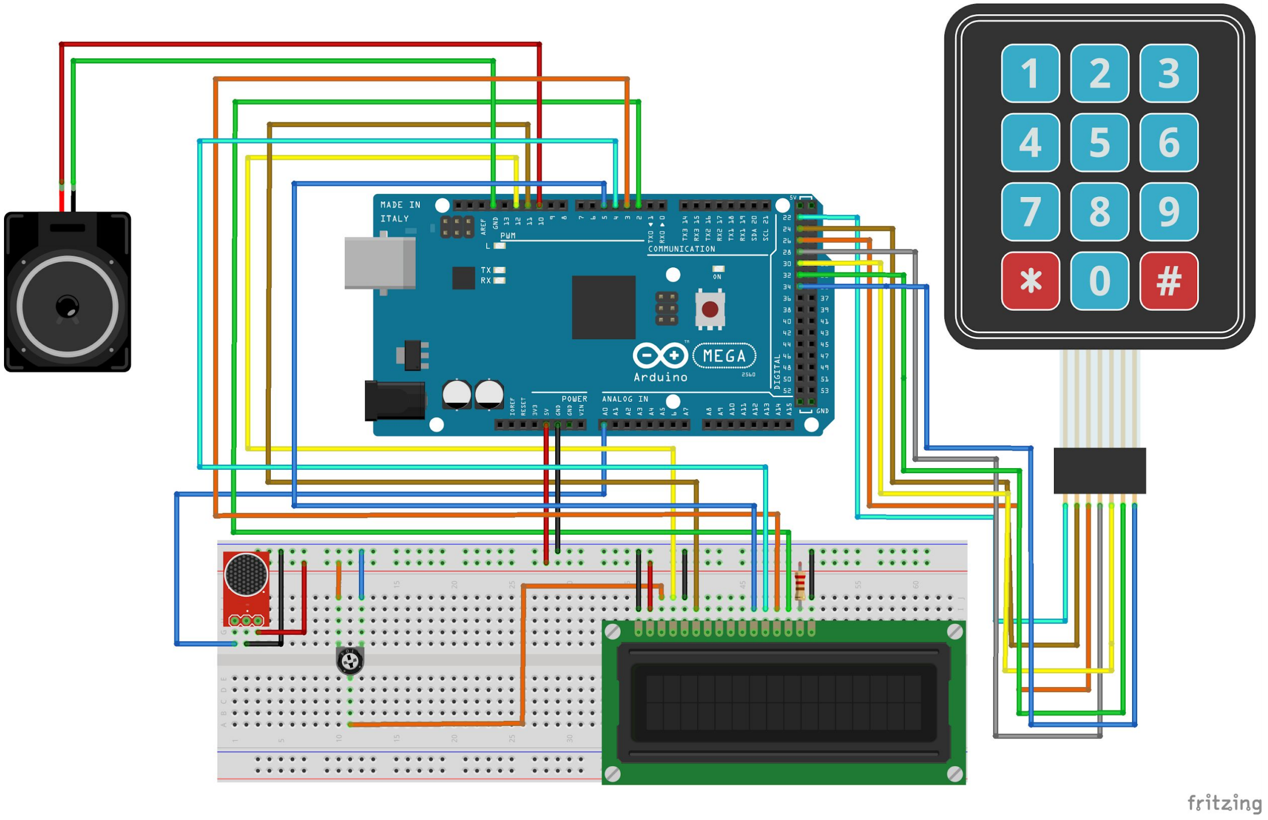
$$DFT(f) = \frac{1}{N}\sqrt{\left(\sum x[n]cos\left(\frac{2\pi fn}{f_s}\right)\right)^2 + \left(\sum -x[n]sin\left(\frac{2\pi fn}{f_s}\right)\right)^2}$$

Where x[n] is the sampled signal, N = number of samples, n = sampling index, f = target frequency (Hz), and f_s = sampling frequency (Hz)

Sketch Code for Implementing DFT

```
double singleFreqDFT(unsigned short signalArray[],  double sampleFreq, float targetFrequency){

  double omega = 2.0 * PI * targetFrequency / sampleFreq;  //find omega0 in radians
  double realPart = 0;                 //real component of DFT, corresponding to cosine
  double imaginaryPart = 0;        //imaginary component of DFT, corresponding to sine
  double currV;                        //current voltage

  //iterate through all samples
  for (int k = 0; k < NUM_SAMPLES; k++) {

    //convert each amplitude of signal from Arduino integer reading to voltage
    currV = signalArray[k] * (5.0 / 1023.0);

    //find real and imaginary components of DFT, using voltage and angle (omega0*k) in radians
    realPart += currV * cos(k * omega);
    imaginaryPart -= currV * sin(k * omega);
  }

  //find and return Fourier magnitude of current sampled signal for target frequency
  return sqrt(realPart*realPart + imaginaryPart*imaginaryPart) / NUM_SAMPLES;
}
```

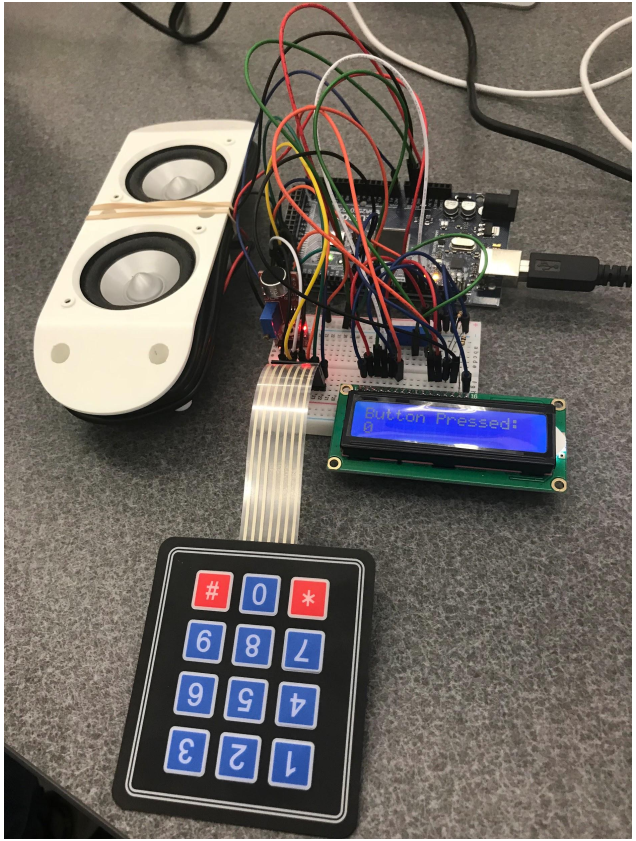
Circuit Diagram

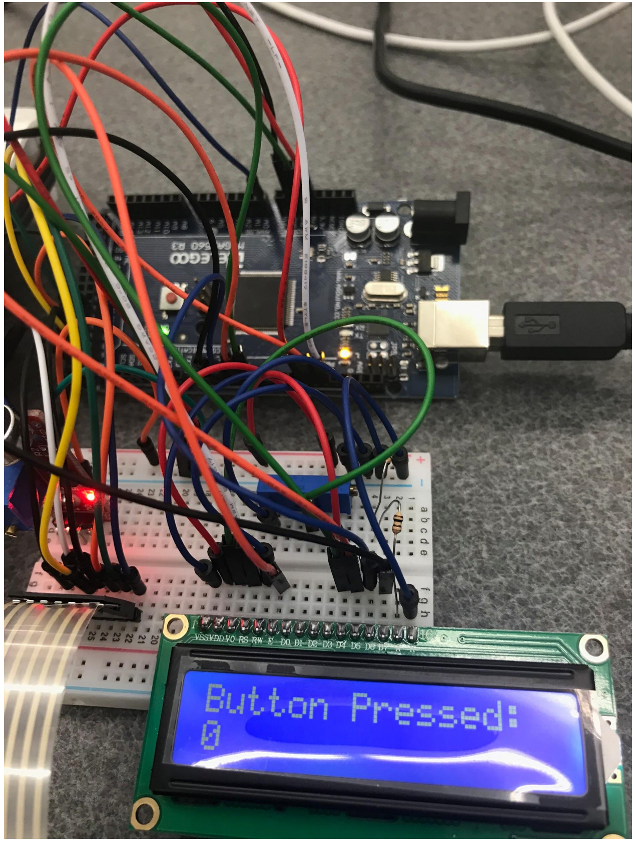


Results

- The results were quite promising for short-distance signaling.
- A better microphone would be required for receiving signals from longer distances.
- A microprocessor better suited for floating point math would speed up the process (like a Raspberry Pi).
- Our prototype, codenamed STFSS (single tone frequency soundwave signaling), has results of 100% classification accuracy.

Prototype Circuit





Summary, what was achieved?

- Dual Band: Our group would have implemented the more complex dual-tone frequency generator and decoder. However, due to the limitation of our hardware functionalities, there is no straightforward way to generate superimposed signals on the Arduino.
- Since our keypad only has 12 keys, single frequency transmitting is adequate for this implementation.
- Applications for this circuit are mainly limited to non-commercial hobby projects.

Thank you!
Спасибо

Ethan Webster

Steven Yoo