PY 681 Final Project:
Alarm Recognition

Matthew Sheinman
Boston University

(Dated: April 30, 2021)

Most alarm systems include an auditory component to alert a user. Older alarm systems often include ONLY an auditory component – e.g., car horns, cooking timers, the alarms on lab equipment, etc. This is not helpful if the user is not present. For this reason, it would be useful to have a device that could listen for certain alarms and recognize them. This project will equip the Arduino to recognize sounds. The Arduino will have two modes – a recording mode, where the frequency spectra of alarms are recorded, and a listening mode, where input data is compared with the stored spectra. In addition, beyond the described use case, the Arduino will in fact be able to recognize any sound with relatively constant frequency character, so there may be more general uses.

I. PLAN OF IMPLEMENTATION

The original plan of implementation is presented here.

March 29th – Demonstrate FFT algorithm on Arduino chip and get input data from the sound sensor.

April 5th – If necessary, build a pre-amp. Store sounds and come up with ways to compare them – probably just a dot product of power spectra would do.

April 12th – Implement a more useful output - maybe the LCD screen.

April 19th - Address any unforeseen problems.

II. PARTS

An op amp (LM358P) is used to create an amplifying circuit. Other parts are already present in the Arduino starter kit, and are marked in the supplied schematic.

III. SOFTWARE OUTLINE

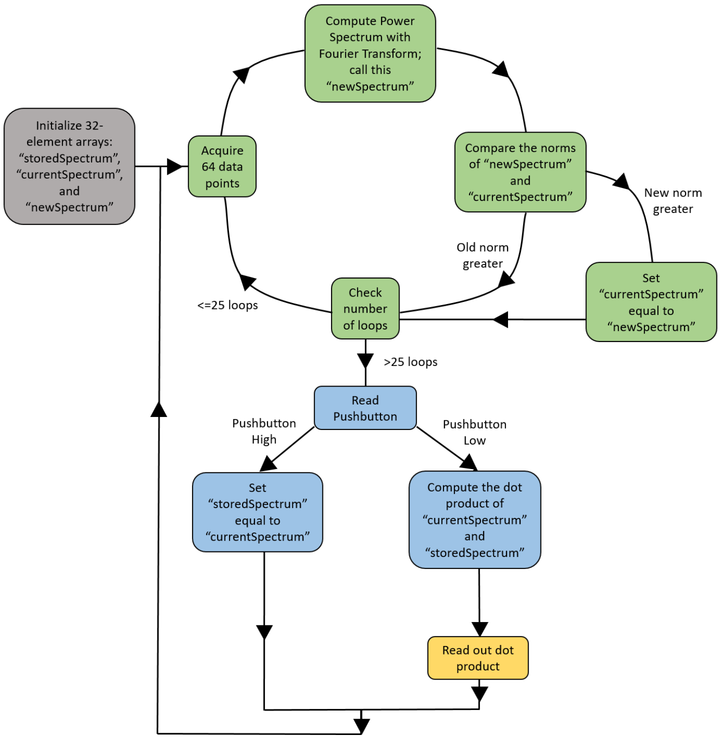
Initialize 32-element arrays: “storedSpectrum”, “currentSpectrum”, and “newSpectrum”
Acquire 64 data points
Compute Power Spectrum with Fourier Transform; call this “newSpectrum”
Compare the norms of “newSpectrum” and “currentSpectrum”
New norm greater
Old norm greater
Set “currentSpectrum” equal to “newSpectrum”
Check number of loops
<=25 loops
>25 loops
Read Pushbutton
Pushbutton High
Pushbutton Low
Set “storedSpectrum” equal to “currentSpectrum”
Compute the dot product of “currentSpectrum” and “storedSpectrum”
Read out dot product

FIG. 1. Software block diagram.

IV. SCHEMATICS

Sounds are measured with a KY-037 sound sensor. The sensor has four pins, but only three will be connected - Analog Out (AO), Ground (-), and Power (+). The remaining unused pin is Digital Out (DO).

Output from AO is sent through an amplifying circuit with gain G=11. The amplifying circuit also includes a blocking capacitor to set DC gain to 0.

For readability, only those Arduino pins used in the project are included in the schematic. All digital pins are used. Pin 1 collects data from a pushbutton. The other 12 pins control a 4-digit 7-segment display.

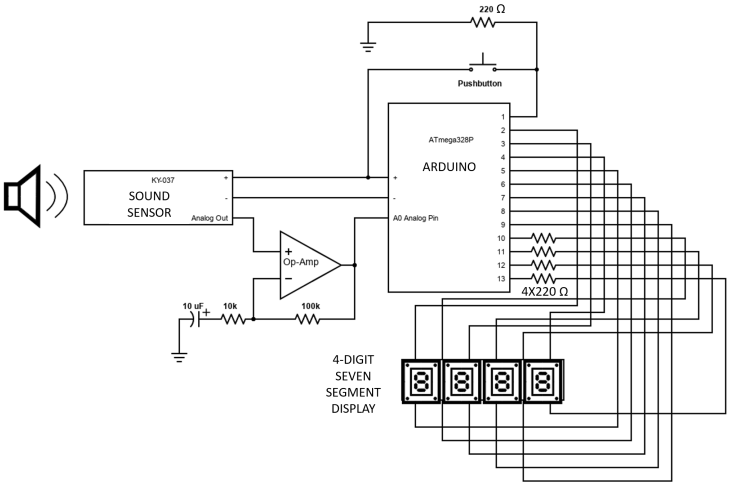

FIG. 2. The schematic.