Patrick Lierle

PY371 Project Final Proposal

Arduino Sequencer Synthesizer

Intro to Sequencer Synthesizers:

Sequencer synthesizers are a branch of synthesizers that do not have a piano-style keyboard as some more familiar synthesizers do, but instead use other inputs to determine the sounds generated. Most sequencers have a given number of steps, usually a multiple of four, at which a note is generated. The user has toggle buttons for each step, allowing them to tell the synth whether or not to play a note at that step. The sequencer cycles through the steps (after step 8, it returns to step 1). In this way, an 8 step sequencer synthesizer can construct 8th note rhythms if we imagine a full cycle of the synth is 1 bar of music in 4/4. The generated notes can then be manipulated with other components such as filters, low frequency oscillators, envelopes, and pitch controls. Below is a picture of an example step synthesizer by Behringer.

My Project:

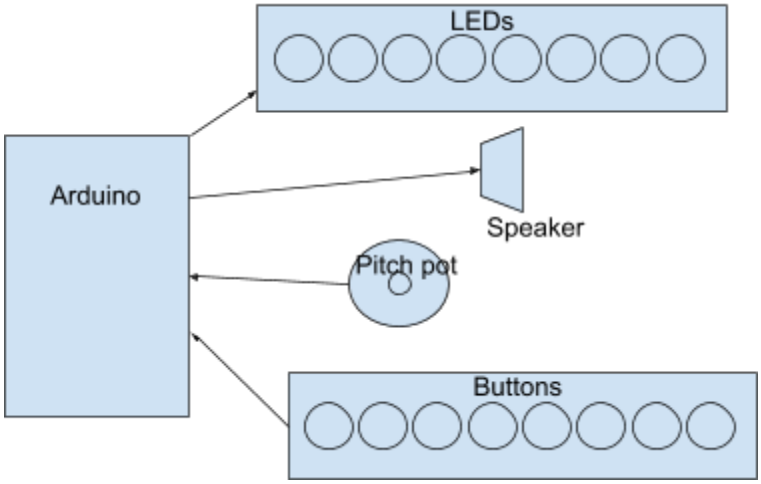
I've split my project into two stages. Completion of the first stage will yield a bare-bones sequencer synthesizer which offers control over rhythm and pitch. Completion of stage two will add flavor to the sound by means of a passive filter circuit and the option of connecting headphones by the addition of a 3.5mm jack, as well as a possible speed control.

Stage 1:

For my basic take on the synthesizer sequencer, I will construct an 8 step sequencer with 8 button switches to toggle whether or not a note is output on that step. I will also include a potentiometer control for the user to select a frequency for the notes. The pitch of all notes will be dictated by the single potentiometer to avoid having to use 8 individual pots. A program could be written where as a setup the user goes through each step and sets its pitch using the pot before using the synth, but that is beyond the scope of stage 1. The output will be sounded through a speaker from an old toy piano. A set of 8 LEDs will also show the current location of the synthesizer in step-space (i.e. what step it is on). By using the toggle switches and the potentiometer, the user will have control over the rhythm and pitch of the synthesizer.

On the programming side, the program will keep a boolean array for the steps, where a 1 means "play" and a 0 means "don't play." The values in this array will dictate whether a square wave is sent to the speaker for that step. The program will be looking for a 5v signal from any of the 8 momentary buttons, which will switch the value for its corresponding step in the array from a 0 to 1 or vice versa. While this is happening, the program will be looping through the steps with a delay between each one, sending out a tone if that step is marked as a 1. The pitch pot will be sending a variable voltage back to the arduino based on its position, and this voltage will be mapped to a logarithmic scale used to determine the frequency of the tone, which will likely

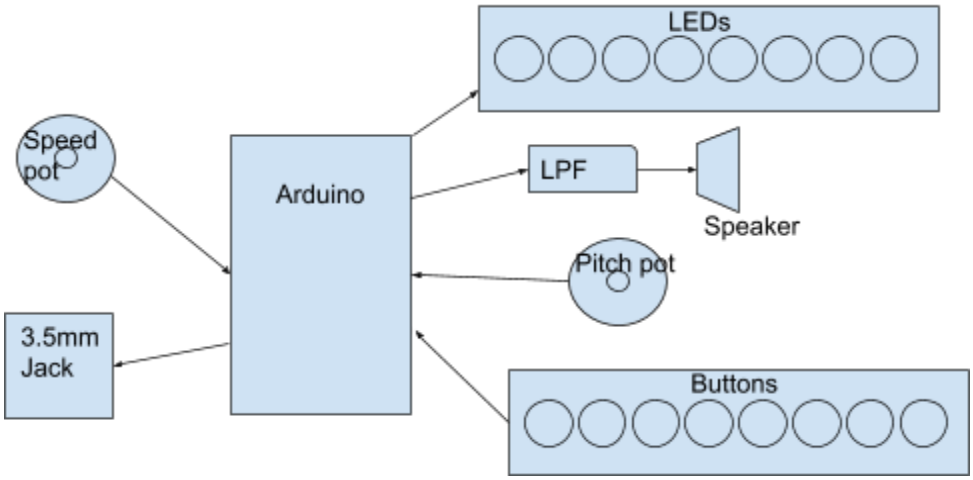
be generated with the arduino tone() function and then sent to the speaker. Finally, the LEDs will need to be lit sequentially. The block diagram is below.



Stage 2:

Stage two introduces a couple more ways to flavor the sound and increase functionality. The first goal of stage two is to add a passive RC (likely) low pass filter (LPF) to the output. This will feature a capacitor and a potentiometer to allow for resistance adjustment and thus cutoff frequency adjustment. Using a force-sensing resistor in the place of the potentiometer will also be explored, time permitting. The second goal of stage two is to add a headphone jack in addition to the speaker. This could either be always on, with the speaker, or another button could control where the output comes from. One more optional goal will be to add another potentiometer which controls the speed at which the synth cycles through its steps.

The programming for stage two will not require anything for the filter, since it is purely analog, but will require a method for alternating between the speaker and headphone jack using the toggle switch, which may use trigger interrupts. It will also need to use the voltage input from the speed potentiometer to determine the delay time between steps. The block diagram with all aforementioned additions is below.


LEDs
Speed pot
Arduino
LPF
Speaker
Pitch pot
3.5mm Jack
Buttons

FULL PARTS LIST

Toggle switches set of 10 $6:

https://www.amazon.com/Uxcell-a12013100ux0116-Position-Vertical-Switch/dp/B007QAJUUS/ref=sr_1_3?dchild=1&keywords=toggle+switch+for+breadboard&qid=1586485255&sr=8-3

LEDs pack of 100 $6:

https://www.amazon.com/eBoot-Pieces-Emitting-Diodes-Assorted/dp/B06XPV4CSH/ref=sr_1_4?dchild=1&keywords=leds&qid=1586468230&sr=8-4

Potentiometers set of 10 $7:

https://www.amazon.com/MCIGICM-Potentiometer-Linear-WH148-Shaft/dp/B0791GYLBZ/ref=sr_1_3?dchild=1&keywords=potentiometer&qid=1586468495&sr=8-3

Jumper wire kit $9:

https://www.amazon.com/Makeronics-Solderless-Breadboard-Prototyping-Electronics/dp/B07P7ZRQX9/ref=sr_1_1_sspa?dchild=1&keywords=awg+22+jumper+wire+kit&qid=1586476047&sr=8-1

Headphone jack $7:

https://www.amazon.com/Aike-Plastic-Stereo-Socket-Connector/dp/B01N5DIZQG/ref=sr_1_13?dchild=1&keywords=headphone+jack+part&qid=1586469087&sr=8-13

Resistors $11:

https://www.amazon.com/AUSTOR-Resistors-Assortment-Resistor-Experiments/dp/B07BKRS4QZ/ref=sr_1_6?dchild=1&keywords=resistor&qid=1586469172&sr=8-6

Caps $10:

https://www.amazon.com/BOJACK-Capacitor-Multilayer-Monolithic-Assortment/dp/B085RLXXBK/ref=sr_1_5?dchild=1&keywords=capacitor+assortment&qid=1586469403&sr=8-5