ArpCube

An Arduino MIDI controller by:
Hanna Thayyil
Hthayyi2
Hthayyi2@uic.edu
Wisam Abunimeh
Wabuni2
Wabuni2@uic.edu

Abstract:

The ArpCube is an Arduino powered MIDI controller. The cube can be plugged into a computer to process the button controls into MIDI, or it can be plugged into a device that understands MIDI such as a synthesizer. The ArpCube can act as a standalone MID trigger or it can be used as an arpeggiator. An arpeggio is a musical term for playing the notes of a chord in succession. When in arpeggiator mode, the ArpCube allows a user to step through a selected sequence of notes based on the input buttons. The ArpCube does not support MIDI-IN or MIDI-THRU signals, only MIDI-OUT.

Description & How to Use:

The ArpCube is inspired by the Novation Launchpad, and the Monome MIDI controller. It can be used with any music software or it can be used with other hardware synthesizers that understand MIDI. The ArpCube is not a standalone instrument, it does not create any audio, it must be used in tandem with a Midi receiving instrument or computer digital audio workspace.

The ArpCube features 32 back-lit buttons used as inputs and outputs. Each button sends a MIDI signal that represents a note. The cube has 3 different modes

-MIDI mode- When in MIDI mode pressing a button will send it's corresponding MIDI On signal out once, and releasing a button will send a MIDI Off signal. Pressing and holding a button will sustain the note until the button is released. When in MIDI Mode the user can use the KEY potentiometer to choose which key the notes are mapped to. The user can also use the OCTAVE potentiometer to transpose the notes on each key up or down octaves.

-Mono Arp Mode- In Mono mode the cube steps through a sequence of 32 steps. Each step is represented by the button, and when a step is reached the button light will turn on. When the arpeggiator reaches a step and the user presses any of the 32 buttons, that step will save the note that was just pressed and play that note when it reaches that step. Every step can only support one note, so if a step already has a note saved and the user presses another note, the old note will be erased and the new note will be saved. The user can transpose octave and keys using the OCTAVE and KEY Potentiometer, and the user can also control the BPM using the BPM Potentiometer.

-Poly Arp Mode- In Poly Mode the cube steps through a sequence of 8 steps. Each step is represented by a column on the button matrix. Each step can hold up to 4 notes. When the sequence is on a step and a user presses a key that note will be added to that step. When the step has reached 4 notes stored, and the user adds another note, all old notes will be cleared and the new note will be stored. The user can transpose octave and keys using the OCTAVE and KEY pot, and control BPM using the BPM pot.

The ArpCube was designed to make use of two Arduino's based on the sheer amount of inputs used. They will be connected using Bluetooth bus. One Arduino will receive information from the various inputs and another will be used to send outputs to both the ArpCube and other connected devices. The ArpCube will use a DIN socket to send MIDI messages to a MIDI understanding device.

INPUT/OUTPUT

Inputs for the device will be 32 buttons, three potentiometers and two switches. The potentiometers allow transposing key, octave and changing BPM. The two switches let you toggle between the three modes. Output will be 32 3mm LED's used as indicators that a button is switched on or off- these LED's back light the buttons. We will be using Adafruit's Trellis keypad driver for simplicity. An LCD screen is used as output to display which mode the user is in, as well as which key they are in and what their current BPM is.

Original Work

The ArpCube is unique because it acts as both a standalone MIDI controller and an arpeggiator. Many documented projects only feature a controller, but the arpeggiator algorithms are completely original. Many DIY projects also only feature 16 or 8 buttons. Since there are 7 notes or 12 semitones in a western scale, we wanted to make the number of buttons on the ArpCube be able to support at least 2 octaves. The Arpcube however supports every possible MIDI octave. We have also not encountered MIDI controller that allows the user to traverse through different Keys, since it is not immediately clear which button is programmed to which note, we wanted to make the MIDI controller easily intuitive and musical. We wrote two original libraries, for Arpeggiating and Note Mapping.

Materials:

2x Arduino Uno
2x Silicone Elastomer Key Pads - https://www.adafruit.com/product/1611
2x Adafruit Trellis PCB Drivers https://www.adafruit.com/product/1616
32x While 3mm LED
1x LCD screen
2x resistors (220 Ohm)
3x Rotary potentiometer (50K)
1x Capacitor (0.05 mF)
2x Toggle Switch
1x DIN socket

References

Sending and Receiving MIDI (we will be using MIDI Send not receive)
https://www.instructables.com/id/Send-and-Receive-MIDI-with-Arduino/

Serial to MIDI References
https://projectgus.github.io/hairless-midiserial/

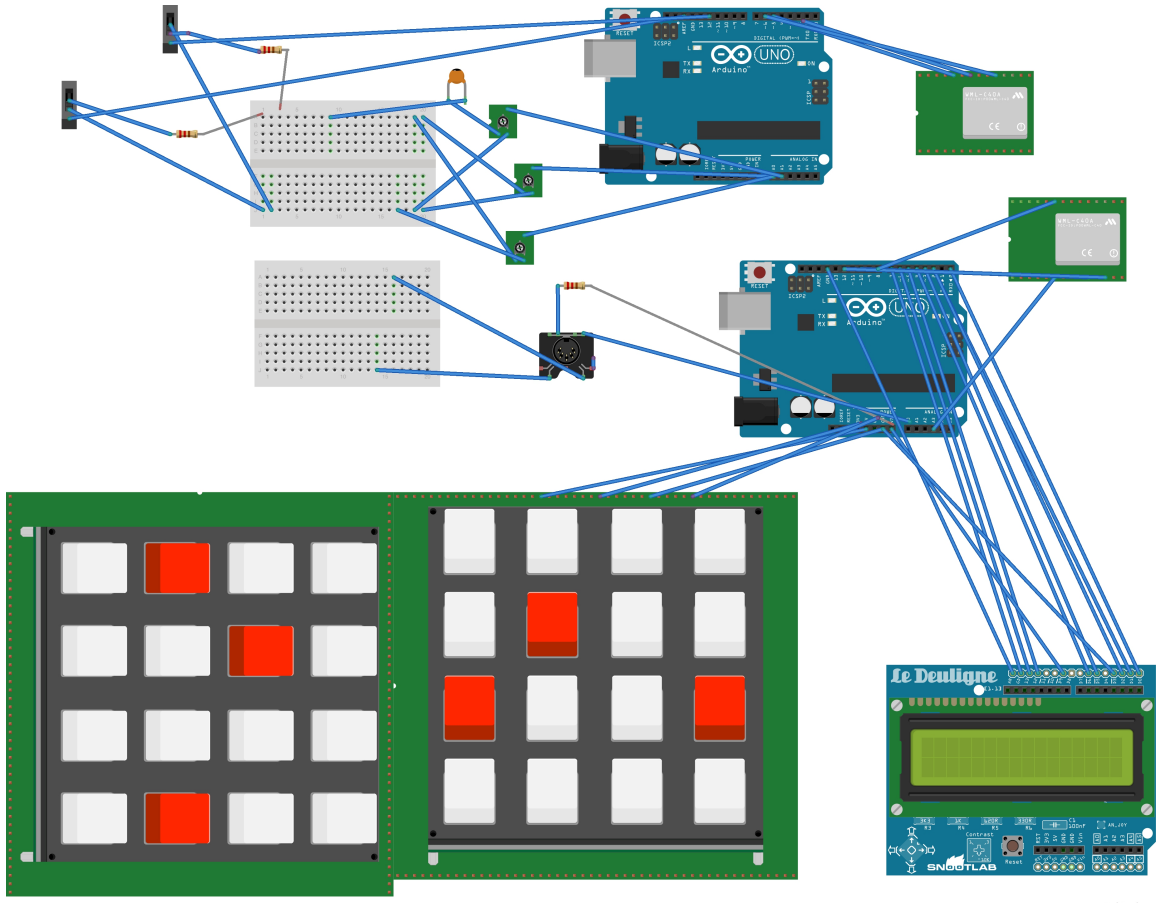
Arduino sequencer used as reference
https://learn.sparkfun.com/tutorials/build-an-auduino-step-sequencer/all

Libraries Used: Software Serial, Arduino STL, LiquidCrystal, Wire.h, Adafruit Trellis, MIDIUSB, MIDI.h

TIMELINE

Proposed Timeline									
WEEK OF									
2/23	3/1	3/8	3/15	3/22	3/29	4/5	4/12	4/19	4/26
Milestone 3 Due 2/28	Sign Up for Expo by 3/7			WEEK OF SPRING BREAK	Milestone 4 Due on 4/3		Progress on Final Presentation	Design Presentation on 4/24	Project Demonstration on 4/27
	Progress on project design			Progress on updating Milestone 3 to Milestone 4				Presentation Slides due on 4/23	Final Report Due on 4/29
				Procurement of Arduino components.			Progress on updating Milestone 4 to Final Report		Teamwork Assessment Due on 5/1
					Implement and build proposed design				
					Write Arduino code				

HOW TO BUILD:



Note: because we were not able to connect our arduinos, our final project uses one Arduino and every single analog and digital pin to connect all inputs and outputs. Our final project also did not end up using the Bluetooth modules, although it could be connected using them . Also, the Fritzing schematic does not show the Trellis PCB's which sit behind the button matrix. These PCB's contain multiplexor and other IC parts that allow the 32 button matrix to only use 4 pins on one Arduino.

Connect Trellis Vin to 5V and Ground to ground.
Connect the INT wire to pin #A2
Connect I2C SDA pin to your Arduino SDA line PIN A4
Connect I2C SCL pin to your Arduino SCL line PIN A5
All Trellises share the SDA, SCL and INT pin.
OCTAVE POT - potPin is on analog pin 0
TEMPO POT is on analog pin 1
SWITCH1 is on digital Pin 6
SWITCH 2 is on digital Pin 7
KEY POT is on pin A2

LCD set up: LiquidCrystal lcd(13, 12, 8, 9, 10, 11);

-Commented in the code sketch are the above instructions on which inputs must be connected to which pins.

Challenges:
Debouncing potentiometers during the arpeggiator was a challenge which was solved using a 0.05 mF capacitor.
Testing the DIN socket with devices other then a digital audio workstation was a challenge because of quarantine restrictions. Currently, if the user connects a MIDI out cable to USB to the computer, it works perfectly fine, but this was not tested connecting to any other device.