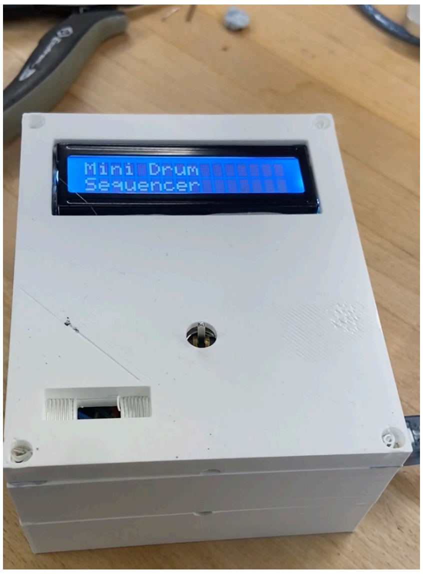
Physical Computing Final Project Writeup

"The mini drum sequencer"

List of components:
1 x 220 Ohm resistor
1 x 16x2 LCD Screen
1 x Joystick Module
1 x Upper Part of downloaded casing
1 x Lower Part of downloaded casing
1 x Custom casing extension



Introduction

My project is a 16 step drum sequencer using a single joystick as an input. The user can access three pattern banks that all trigger samples in Max MSP(Kick, snare and hat). By scrolling left and right the user can choose a step to modify and then click down on the joystick to select or de-select the step. By scrolling up and down the user can change between banks and a short message displayed on screen shows the user which sound this bank will modify. For the step sequence layout I am inspired mostly by drum machine plugins(particularly the "Redrum drum machine").

Loading Screen(setup function)

The loading screen is a custom animation I creating using for loops with a delay in the setup function. This clearly tells the user what the object is and instructions on how to use it.

The main body of code(loop function)

Before the setup function, global variables are declared. These vary from variables for the Arduino pin inputs as well as other things that are used in the loop function. These include custom animated figures that represent the scrolling position, current step in sequence as well as the pattern itself.

The pattern is created using a 3d array that represents 16 steps in 3 different banks and is drawn to the screen using for loops. As the current bank and step position are global variables, it allows the user to easily scroll through patterns and banks. The joystick works by identifying when the x and y inputs have passed a certain threshold and then incrementing or decrementing the current step/bank. The joystick is alternated, this is something I wanted to add as I use these settings when controlling video games. Due to the simplistic, video game like feel, I wanted to incorporate this although this can easily be changed in the code.

Max MSP

In max, the Arduino data is sent as an array and is split in max using the zl.slice object. When each component in the array equals to '1', a sound file is played. I could have sent the data from max to Arduino regarding the current step, as this would most likely result in sturdier time keeping. However, I really wanted to push my boundaries and implement most of the logic on the Arduino itself as I am more experienced in Max MSP.

Electronics

As I did not want to use a breadboard I had to find a way to route the 5v and ground outlets of the Arduino to multiple pins on the lcd controller as well as the joystick. I achieved this by laying copper tape around the sides of the housing. I then cut wires and soldered them to the tape. The tape strip for the ground has 6 pins connected, and the 5v strip has 4. Instead of using a potentiometer to control the brightness of the screen(which is often shown in tutorials), I have used a 220Ohms resistor, soldered to a ground connection that controls the brightness.

Issues and troubleshooting

The original idea for the project was using a touchscreen controller to control steps however due to the nature of the component and the libraries needed to run properly I decided that it was better to use the lcd screen. This also meant that the uno is running less calculations, resulting in more accurate timekeeping. Due to this change of plans, I had to print a different casing. After printing I realised it was not tall enough to include all the wires so had to use Maya to create a custom component that attaches to the lower part of the casing. This extends the Y axis of the model and allowed me to fit all the wires and casing.

After creating the sequencer I had slight problems with timekeeping, it was only after recording my 1st video I fixed this by changing the number of the metro in the max patch(this is why two videos are included). My final issue was the joystick, due to the casing proportions, the attachment to the joystick was too large to use, therefore I have opted not to attach a custom knob due to my very limited 3d modelling knowledge although if I were to extend this project I would definitely design a custom knob that fits the look and feel of the box.

Schematic Note

**In the breadboard pdf I have used a breaboard instead of copper tape. The breadboard in the schematic is used to represent the copper tape which is essentially two rails where wires have been cut and soldered to.*

Final Note

I am very happy with the outcome of the project, I feel that the lcd screen is actually more fitting as a controller due to its simplistic video game look. In my opinion, the machine is very easy to use and fits its purpose tastefully. To extend this project I would consider creating different GUI's that could control envelopes, effects and other aspects of the drum machine.

References

https://www.arduino.cc/reference/en/libraries/liquidcrystal/createchar/
https://www.reasonstudios.com/devices/redrum
https://www.youtube.com/watch?v=dZZynJLmTn8
https://www.youtube.com/watch?v=MlDi0vO9Evg