
Arduino and Max MSP Finger Drum Machine Concept

Jakub Krueger

University of Limerick
Ennis, Co.Clare,Ireland
19267452@studentmail.ul.e

Abstract

This project describes the creation of an interactive performance-based project which makes use of the serial communication between the Arduino and Max MSP in order to create an interactive finger drum machine. This project consists of the usage of buttons, LEDs, and a potentiometer whereby together, along with the use of MAX MSP, creates a finger drum machine where each button allows the user to play a sample. The three samples that are used in this project are the kick, crash, and hi-hat. The user is able to play these samples depending on the pattern he/she uses when pressing these buttons which will in turn create a simple drum beat. The concept behind this project can also be expanded further into a live-performance instrument. If expanded, artists would be able to add extra buttons, which would allow them to play more than three samples. This will give them a better ability to create a more complex type of beat when pressing the buttons depending on the pattern of their choice.

Author Keyword

Interactive Performance; Arduino; Max MSP; Finger Drum Machine;

Introduction

This report will explain the concept behind the project, and it will present a technical description of the piece. Throughout the report, images of the project will also be included along with the project's circuit diagram. Overall, this report will document the current state of the project and its future development.

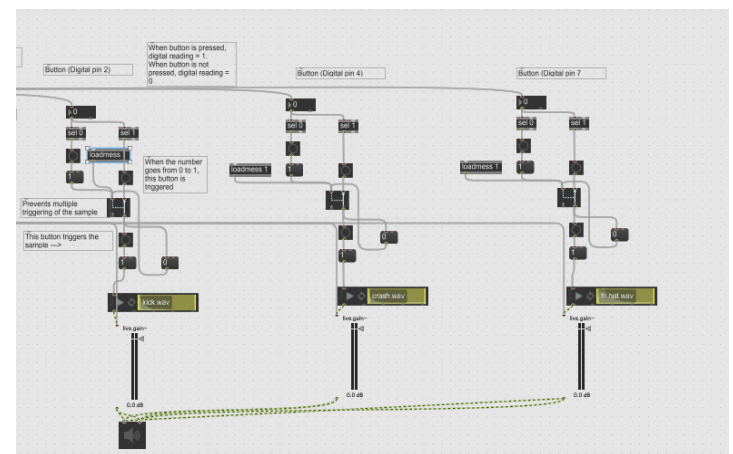
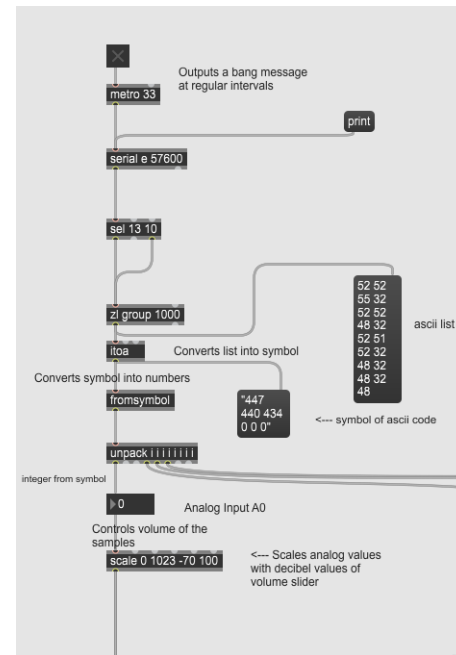
Project Description: Arduino

As mentioned in the introduction, both Arduino and Max MSP has been used to create this project. In this project, only one sensor is being used however I have created three arrays for sensors for the future development of this project. In this project, there are three digital inputs, which consists of three buttons that are connected to the pins 2,4 and 7. The state of these have been assigned to a global variable. Using serial communication between the Arduino and Max, a sample is triggered when the state of each button is high. The pins 9, 10 and 11 are set as digital outputs in this project. These pins are connected to the three LEDs which have been programmed using if statements to light up when its corresponding button is pressed. The potentiometer in this project has been used as an analog input. The value of the potentiometer is read, and its data is sent to Max MSP which is used to control the volume of the three samples. Using pulse width modulation, the LEDs brightness varies depending on the volume of the samples which adds to the performance aspect of the project.

```
1 //Jakub Krueger 19267452
2
3 int valu[3]; //array for sensors.
4
5 //Digital Inputs:
6 const int buttonPin1 = 2; // digital pin 2
7 int buttonState1 = 0; // the state of the button press
8
9 const int buttonPin2 = 4; // digital pin 4
10 int buttonState2 = 0; // the state of the button press
11
12 const int buttonPin3 = 7; // digital pin 7
13 int buttonState3 = 0; // the state of the button press
14
15 int knobValue = 0; // value of potentiometer
16 int light = 0; // value of A0 analog pin which is used to change the brightness of the 3 LED's.
17
18 void setup() {
19   Serial.begin(57600); // set up serial communication
20   pinMode(buttonPin1, INPUT); // digital pin 2 is set as input
21   pinMode(buttonPin2, INPUT); // digital pin 4 is set as input
22   pinMode(buttonPin3, INPUT); // digital pin 7 is set as input
23   pinMode(11, OUTPUT); // digital pin 11 is set as input
24   pinMode(10, OUTPUT); // digital pin 10 is set as input
25   pinMode(9, OUTPUT); // digital pin 9 is set as input
26 }
27
28
29 void loop() {
30   for(int i = 0; i < 3; i++){ // reads analog inputs one by one and sends the value to max.
31     valu[i] = analogRead(i);
32     Serial.print(valu[i]);
33     Serial.print(" ");
34   }
35   light = analogRead(A0); // A0 input is being assigned to a variable.
36
37
38
39   buttonState1 = digitalRead(buttonPin1); // digital pin 2 is being read and it's data is sent to max.
40   Serial.print(buttonState1);
41   Serial.print(" ");
42
43   buttonState2 = digitalRead(buttonPin2); // digital pin 4 is being read and it's data is sent to max.
44   Serial.print(buttonState2);
45   Serial.print(" ");
46
47   buttonState3 = digitalRead(buttonPin3); // digital pin 7 is being read and it's data is sent to max.
48   Serial.print(buttonState3);
49
50   if(buttonState1 == HIGH) // when an individual button is pressed, its corresponding LED is turned on.
51   {
52     analogWrite(11, light/4);
53   }
54   } else {
55     digitalWrite(11, 0);
56   }
57
58
59   if(buttonState2 == HIGH)
60   {
61     analogWrite(10, light/4);
62   } else {
63     digitalWrite(10, 0);
64   }
65
66
67   if(buttonState3 == HIGH)
68   {
69     analogWrite(9, light/4);
70   } else {
71     digitalWrite(9, 0);
72   }
73
74   Serial.println();
75
76   ...
77 }
```

Project Description: MAX MSP

In Max MSP, the data that is being sent from the buttons and the potentiometer in the Arduino through serial communication is converted into an ascii list. This list is then converted into numbers which are used to control the volume of the samples using the potentiometer in Arduino. These numbers are also used to trigger a bang message whenever a button is pressed which in turn triggers each of the samples to be played, depending on which button is pressed. In this project, only three buttons, along with three samples are being played. I have also included a gate in this Max MSP code in order to prevent the samples from playing after the button has been pressed once. This allows the user to play each individual sample without each sample interfering with each other. I believe that the future development of this project would consist of adding extra samples in Max MSP along with extra potentiometers, buttons and LEDs which would give the user a better ability to create more complex beat/instrumental.



Core Concept, Diagrams and Future Development

In the right column, we can see the visual representation of the finger drum machine. As mentioned above, this project consists of three buttons, it's three corresponding LEDs, and a potentiometer. Each of the three LEDs are also different color to add to the performance aspect of the project when the user is using the drum machine to play each of the different samples. In the right column, the circuit diagram of the project is also included. The core concept of this project was to create a finger drum machine where the user is able to play different samples of drum sounds depending on the button that they press. However, I believe that the buttons are not the best for user interaction for this project specifically. I believe that a finger pad or a touch pad would improve on the performance aspect of this project as it would give the user a smoother and more responsive interaction when triggering the samples. To expand on this project further, I believe that adding extra potentiometers would allow to control the volume of each individual samples instead of having one potentiometer that controls all the samples. By adding extra samples, LEDs and buttons, this project would also be able to be used as a midi keyboard, where user input data is sent to a DAW such as Ableton for example and in turn it would allow the user to create a beat or instrumental track.

