4k Labs project documentation

Arduino based MIDI Drum

Team Members: Maedin Seid & Betemariam Assaminew
Team Mentor : Lidetu Tadesse

July 2021

Table of content

Abstract :

In this project, we aimed to create a simple electronic drum that anyone could build using hardware such as a microcontroller, piezo sensors, resistors, and the softwares hairless-midiserial and loopMIDI.The hardware is designed in such a way that, in order to simulate a drum set, piezo sensors are used to represent the cymbals and drums, which produce a voltage when struck.The microcontroller reads this voltage to generate the note that each piezo sensor represents.

Theory :

What is MIDI?

MIDI (Musical Instrument Digital Interface) is a protocol developed in the 1980's which allows electronic instruments and other digital musical tools to communicate with each other. MIDI itself does not make sound, it is just a series of messages like "note on," "note off," "note/pitch," "pitchbend," and many more. These messages are interpreted by a MIDI instrument to produce sound. A MIDI instrument can be a piece of hardware (electronic keyboard, synthesizer) or part of a software environment (ableton, garageband, digital performer, logic...).

How does the MIDI drum work ?

As illustrated in the diagram below, The drum sends a **midi message** to a computer application software known as a DAW; examples of DAW include FL studio, Abelton Live, Adobe Audition, Audacity,Cubase, and others. In this project, we used FL studio because it is free and simple to download and use. As a result, the **midi message** we sent is interpreted by the DAW, which then plays a specific note based on the midi message we sent.

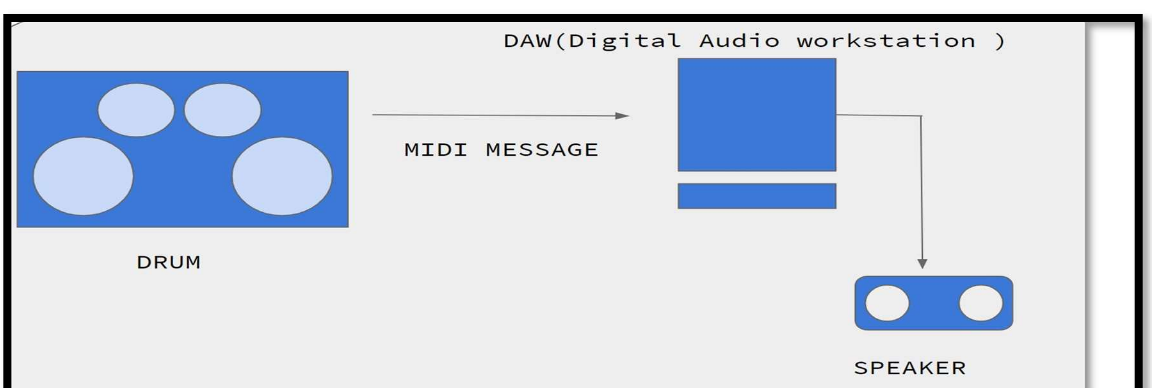


Fig 1

- **MIDI Messages**

MIDI messages are used by MIDI devices to communicate with each other.Structure of MIDI messages:

- MIDI message includes a status byte and up to two data bytes. See fig . 2 .
- Status byte
  - The most significant bit of status byte is set to 1.
  - The 4 low-order bits identify which channel it belongs to (four bits produce 16 possible channels).
  - The 3 remaining bits identify the message.
- The most significant bit of data byte is set to 0.

Classification of MIDI message**:**

Channel messages | **System messages**

Voice message | **mode message** | **common message** | **real-time message** | **exclusive**

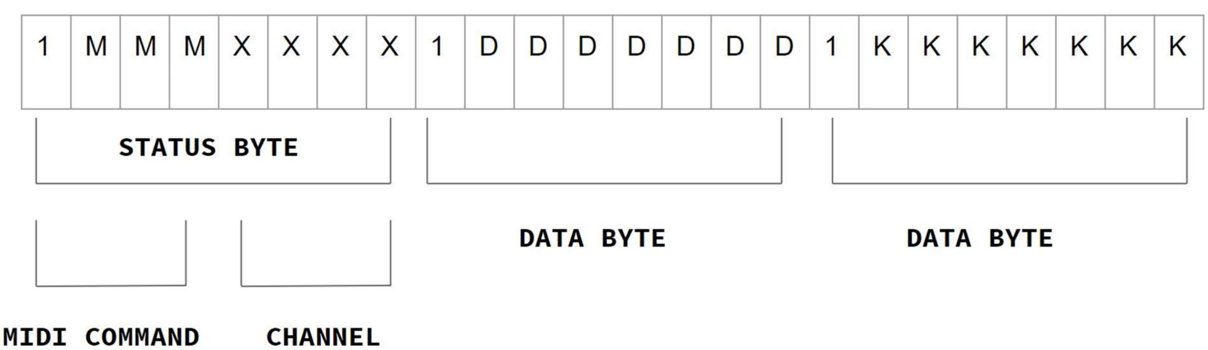


Fig 2

In this project we used channel voice messages to communicate with our DAW . which in this case we only need to use either note on or note off command to communicate with a FL studio ;Here is the table on **Fig 3** for the MIDI message in decimal notation for commands note on and off on different channels . The last two data byte refer to the Midi number(used to represent notes) and velocity (the force with which a note is played) .

Channel	Note On	Note Off	MIDI Number	Velocity
1	144	128	0 - 127	0 - 127
2	145	129	0 - 127	0 - 127
3	146	130	0 - 127	0 - 127
4	147	131	0 - 127	0 - 127
5	148	130	0 - 127	0 - 127
6	149	133	0 - 127	0 - 127
7	150	134	0 - 127	0 - 127
8	151	135	0 - 127	0 - 127
9	152	136	0 - 127	0 - 127
10	153	137	0 - 127	0 - 127
11	154	138	0 - 127	0 - 127
12	155	139	0 - 127	0 - 127
13	156	140	0 - 127	0 - 127
14	157	141	0 - 127	0 - 127
15	158	142	0 - 127	0 - 127
16	159	143	0 - 127	0 - 127

Fig 3

The MIDI Number represents different notes in different octaves, as shown in the table in **Fig 4.**

Octave	Notes											
Number	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127	–	–	–	–

Fig 4

Software requirements:

Arduino IDE

The Arduino Integrated Development Environment (IDE) is a **cross-platform application** (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards. You can download Arduino IDE in the following link https://www.arduino.cc/en/software .

Fig 5

LoopMIDI

Provides a virtual midi channel

How to use it?

As shown the in Fig 5

The button which is indicated by number 1   Used to add ports

The button which is indicated by number 2 Used to delete already created ports

The button which is indicated by number 3 To pause the ports while working

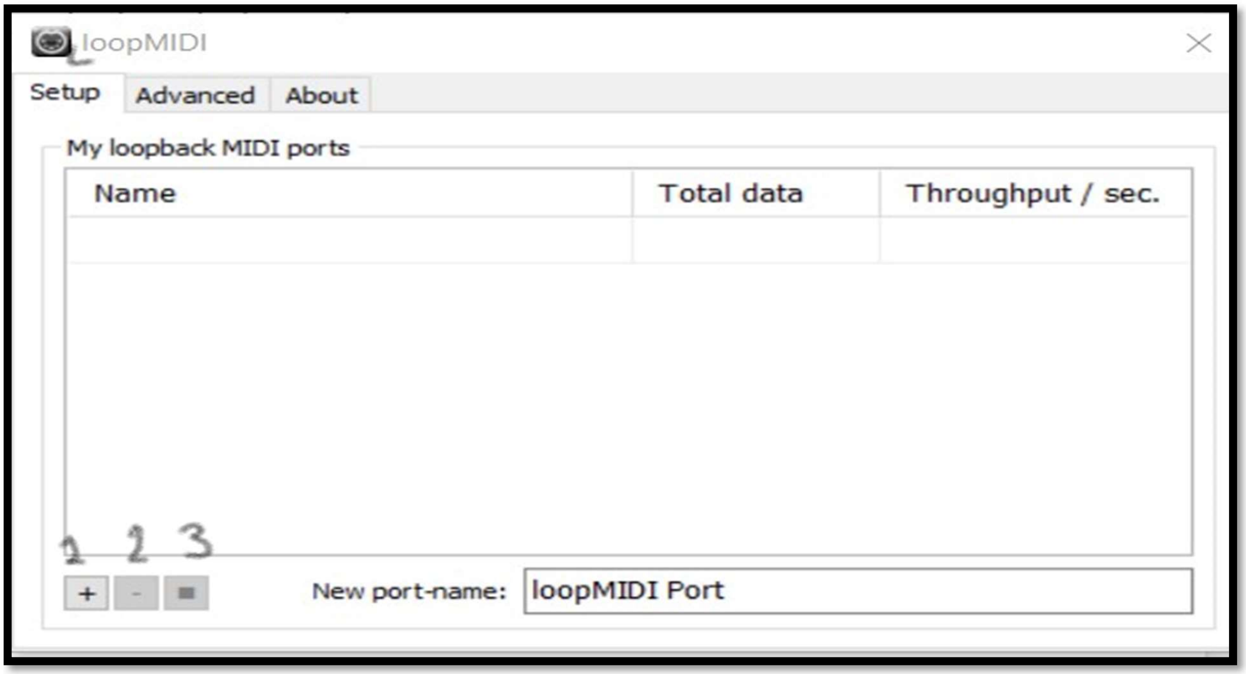


Fig 6

Hairless-midiserial

Convert serial message to midi message

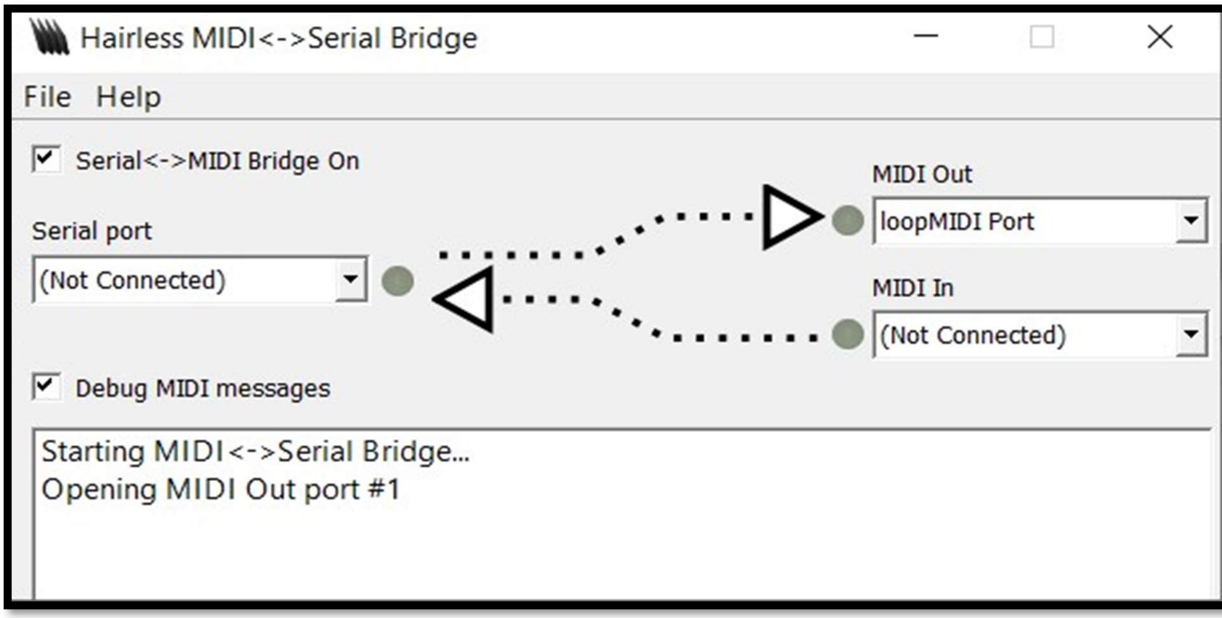


Fig 7

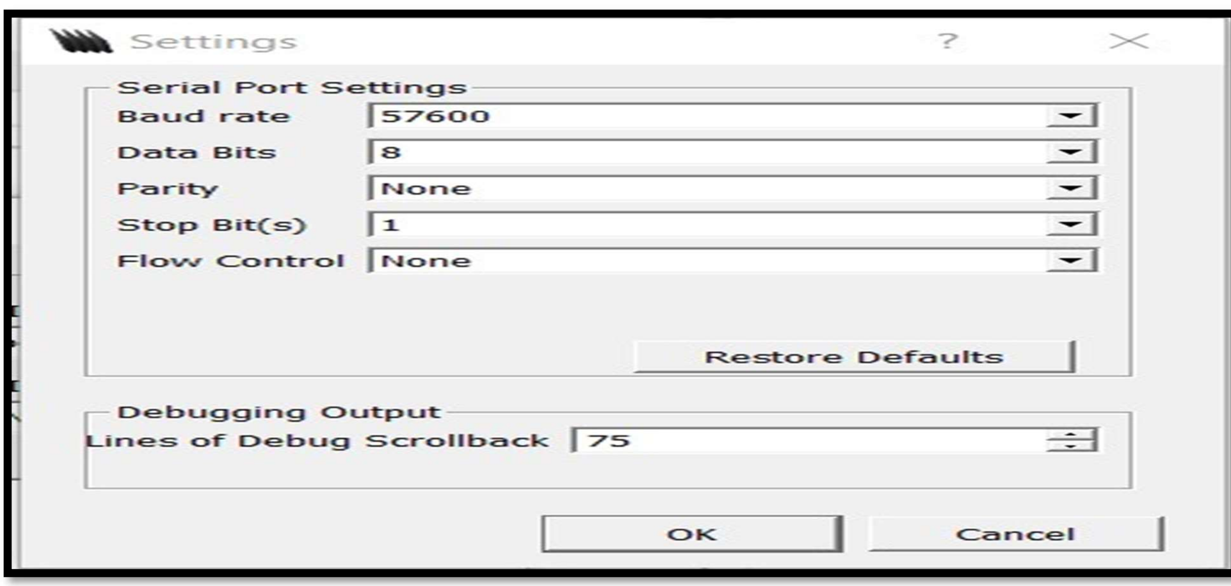


Fig 8

<u>Fl studio</u>

Used to convert midi notes to actual sound they represent we used Fl studio which is DAW software.

You can download FL studio from https://www.image-line.com/fl-studio-download/

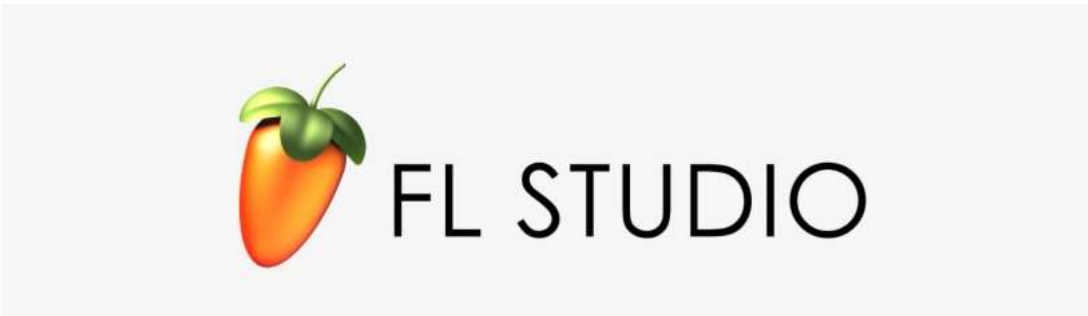


Fig 9

Hardware requirements:

Piezo sensor

Piezo sensors are used to convert a physical parameter; for example acceleration or pressure, into an electrical signal. Piezo sensors are used to measure the change in pressure, acceleration or strain by converting them into electrical charge . in this project we used four piezo sensors .

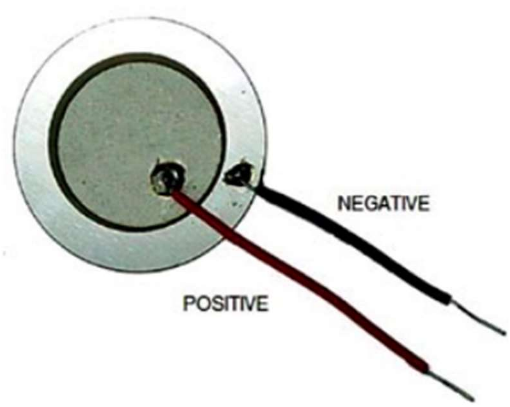


Fig 10

Resistors

In this project Four one mega ohm resistors are used which are connected in parallel to the piezo element to limit the voltage and current produced by the piezoelectric element and to protect the analog input from unwanted vibrations.

Fig 11

MicroController

In this project we used **Arduino uno** as a microcontroller .Arduino is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller.

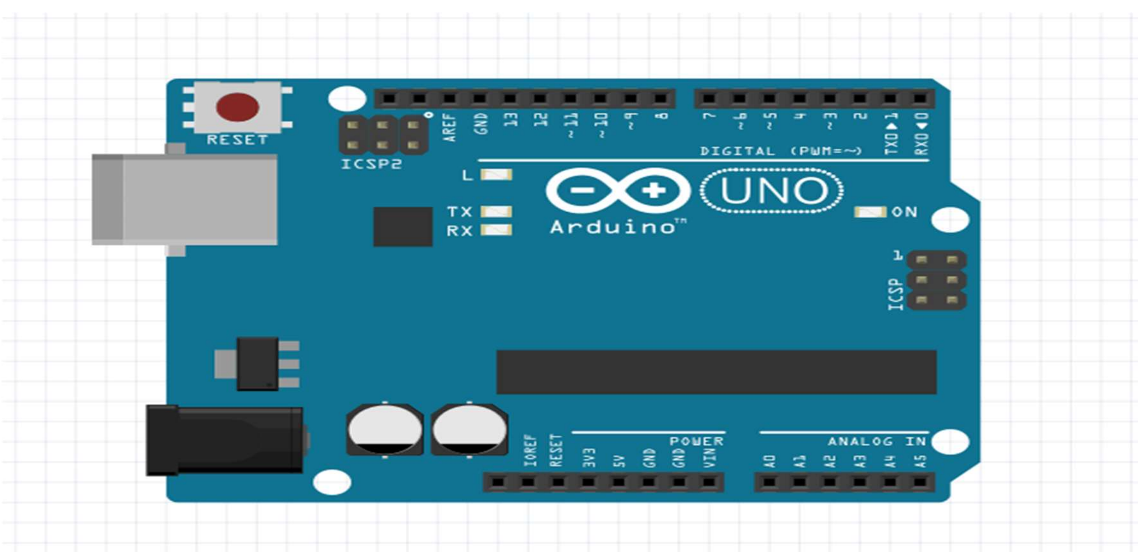


Fig 12

Design :

This design is done using fritzing software you can find the free version form the link below
https://www.filehorse.com/download-fritzing-64/download/

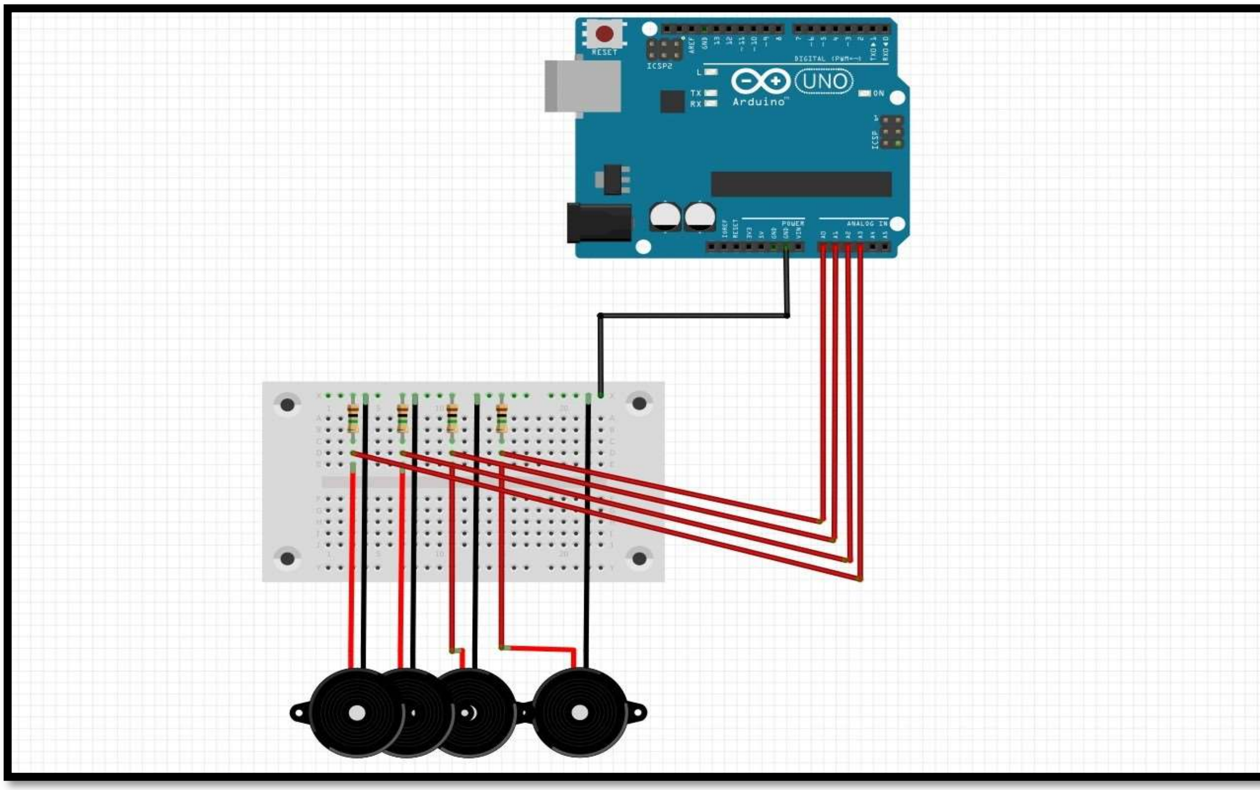


Fig 13

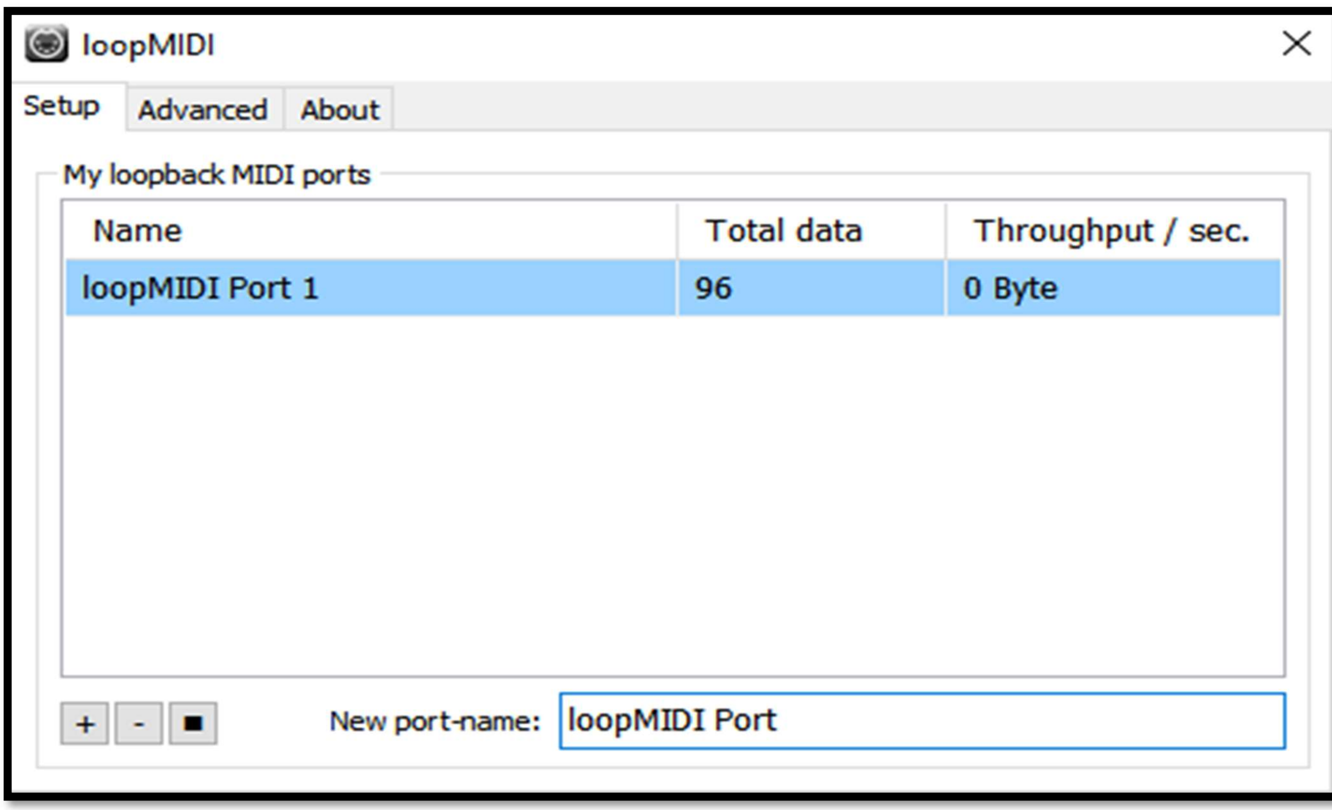
Implementation :

Step 1: Connect the components as shown in the **Fig 13**

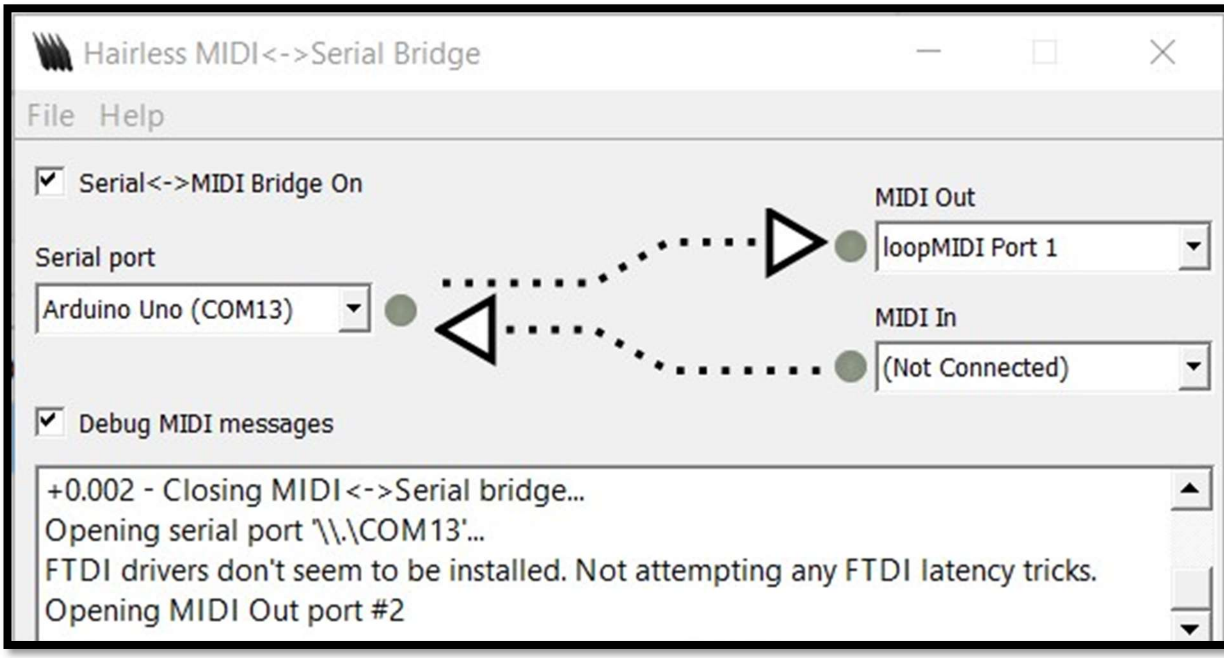
Step 2 : upload the code that is found on

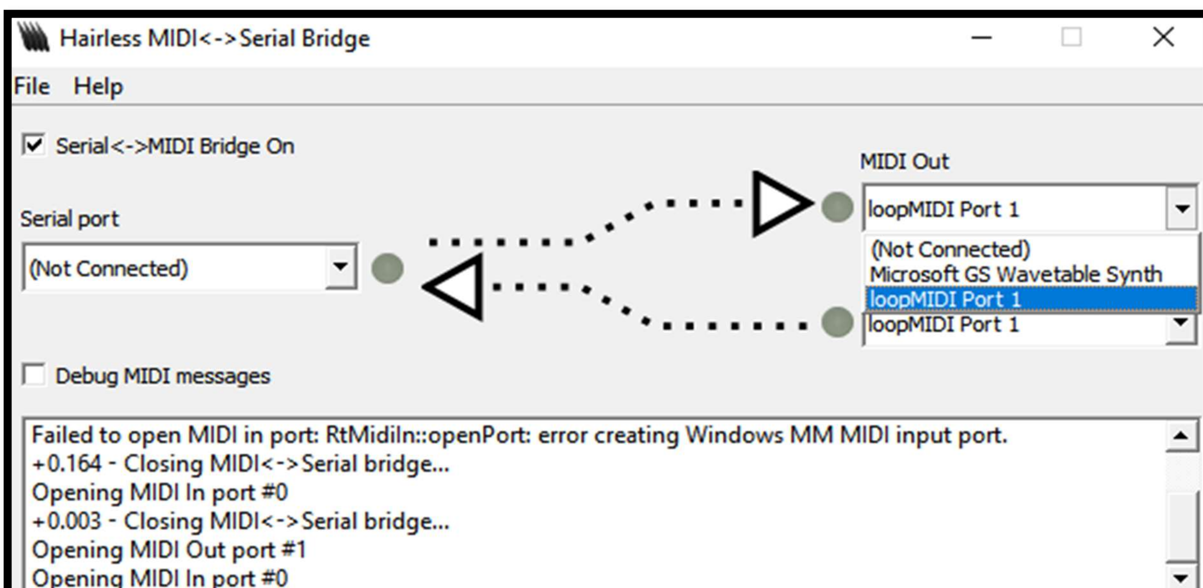
https://github.com/4K-Labs/Arduino-MIDI-Drum/blob/main/e-drum.ino to the Arduino

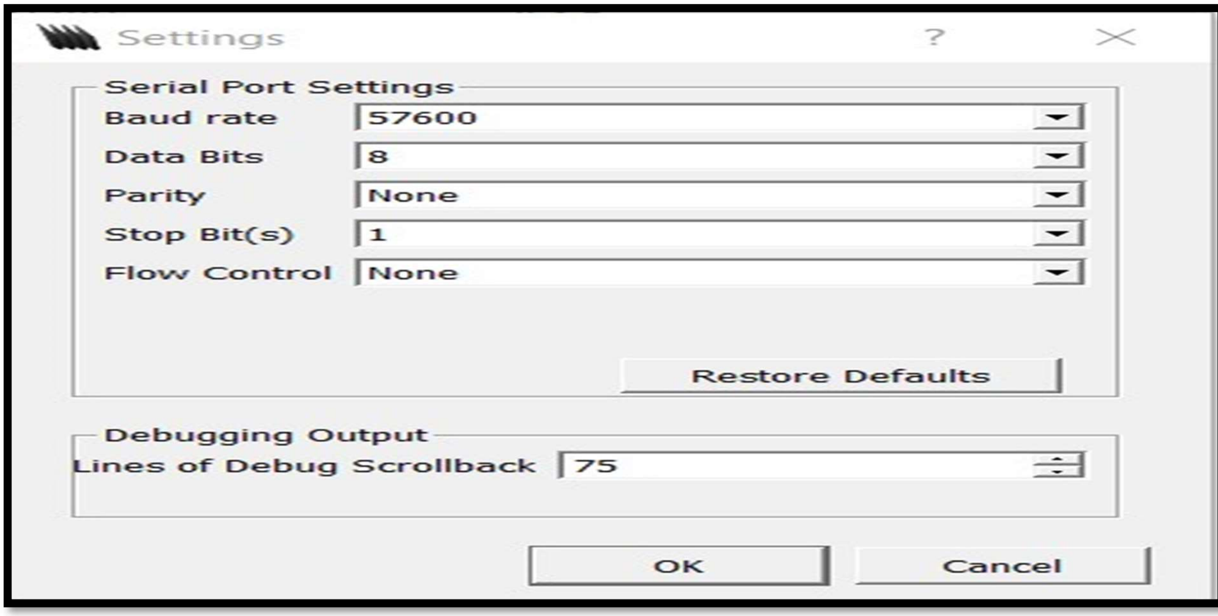
Step 3: open loopmidi and name your port



Step 4: open hairless midi and select the port you named in step 3 and the serial port of the Arduino then go to file and choose preference and adjust the baud rate according to your serial communication setup you set in ,example Serial.begin(baud rate).

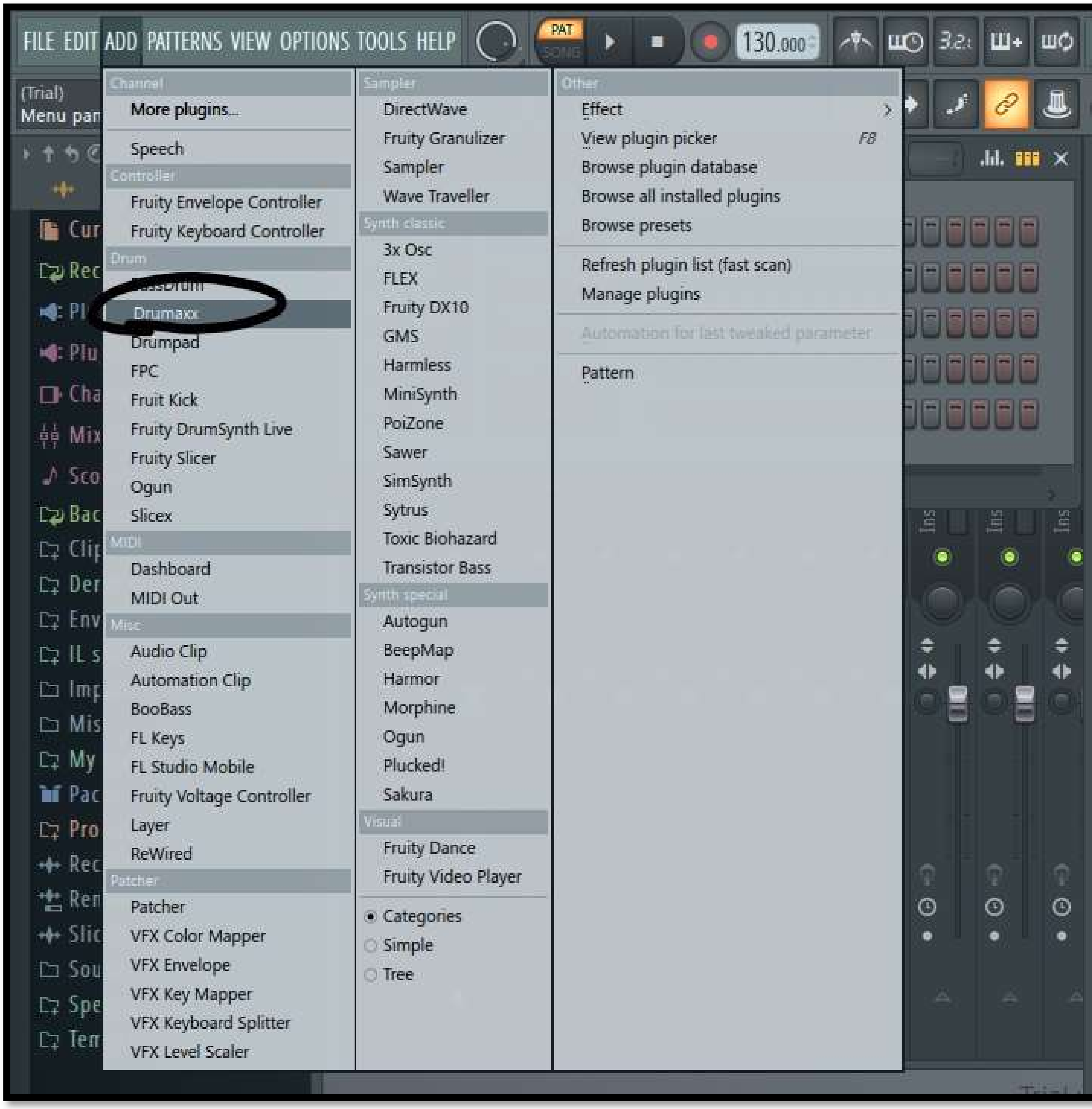






Step 5: open FL studio and at the top section click on **ADD** button

Step 6: After touching the ADD button there are many choices for using the drum, select **Drumaxx.**



Step 7: You can adjust the volume amount and you can also edit what type of note to play referring **Fig 4.**

Challenges :

-locating well-organized and enriched resources on using the MIDI protocol to build an Arduino MIDI controller (which in this case is a drum).
-The second challenge was the drum board and the piezo sensor were not aligning and it was causing problems in the sound.

Reference :

1.About MIDI message
https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node158.html
2.How to build a MIDI a drum
www.youtube.com/watch?v=5S L-W1Ynn3c&t=1s