David Shim

Team 47: Functional Auditory Robotics Therapy

Bhowmik

ECE 120 Honors Lab

Dec. 13, 2021

Final Report

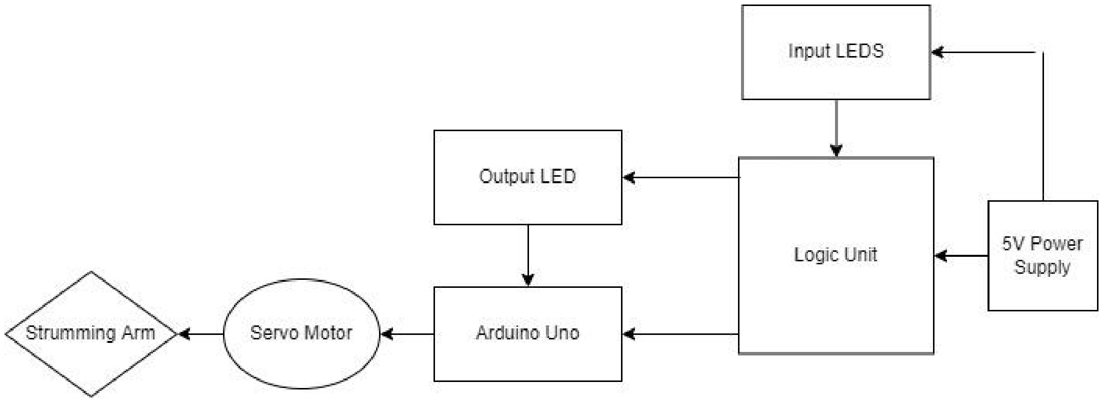
Introduction

The original purpose of Functional Auditory Robotics Therapy (hereinafter referred to as FART) was to build a robot which could be programmed to play 3 chords (G, C, D) on a guitar in a programmable sequence. The robot was to press down on the strings corresponding to the chord inputs and play a single, simple 4/4 strum pattern. Ideally, the robot could be used to play guitar while users sing along to the song of their choice, creating a fun and positive experience for the whole family.

As the project evolved, however, due to circumstances which limited and narrowed down its original scope, the statement of purpose became centered around building a robot which could strum guitar strings based on pre-programmed sequences of inputs. Sort of like a lock for a bicycle, the guitar strumming mechanism would only function if the correct sequence of LED inputs was selected.

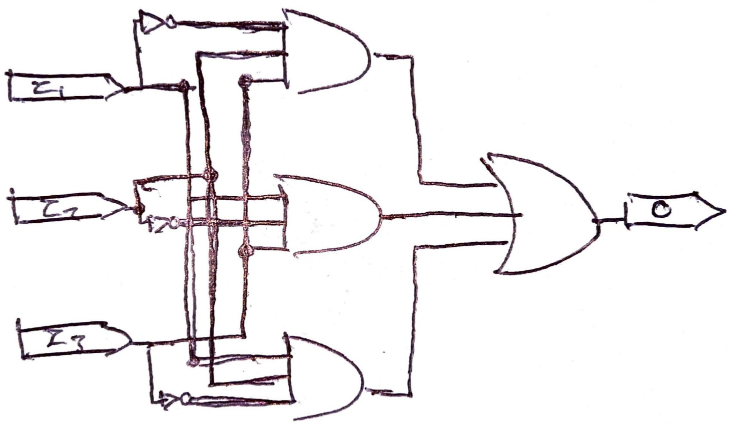
The obvious benefit of this project was building a substitute arm for right hand amputees looking for cheap access to the guitar. Another potential use for this project is practice, allowing beginning guitar players to focus solely on their left hand chord fingerings without having to worry about hand independence while the strumming arm plays a slower, very basic strum for them. Now with regards to the input logic device, it has become a sort of "lock" against people unauthorized to activate the robot. The device also allows one to program multiple strumming patterns under specific LED input sequences, which can be altered in code to change tempo and rhythm.

Design



Appendix 1 - FART Block Diagram

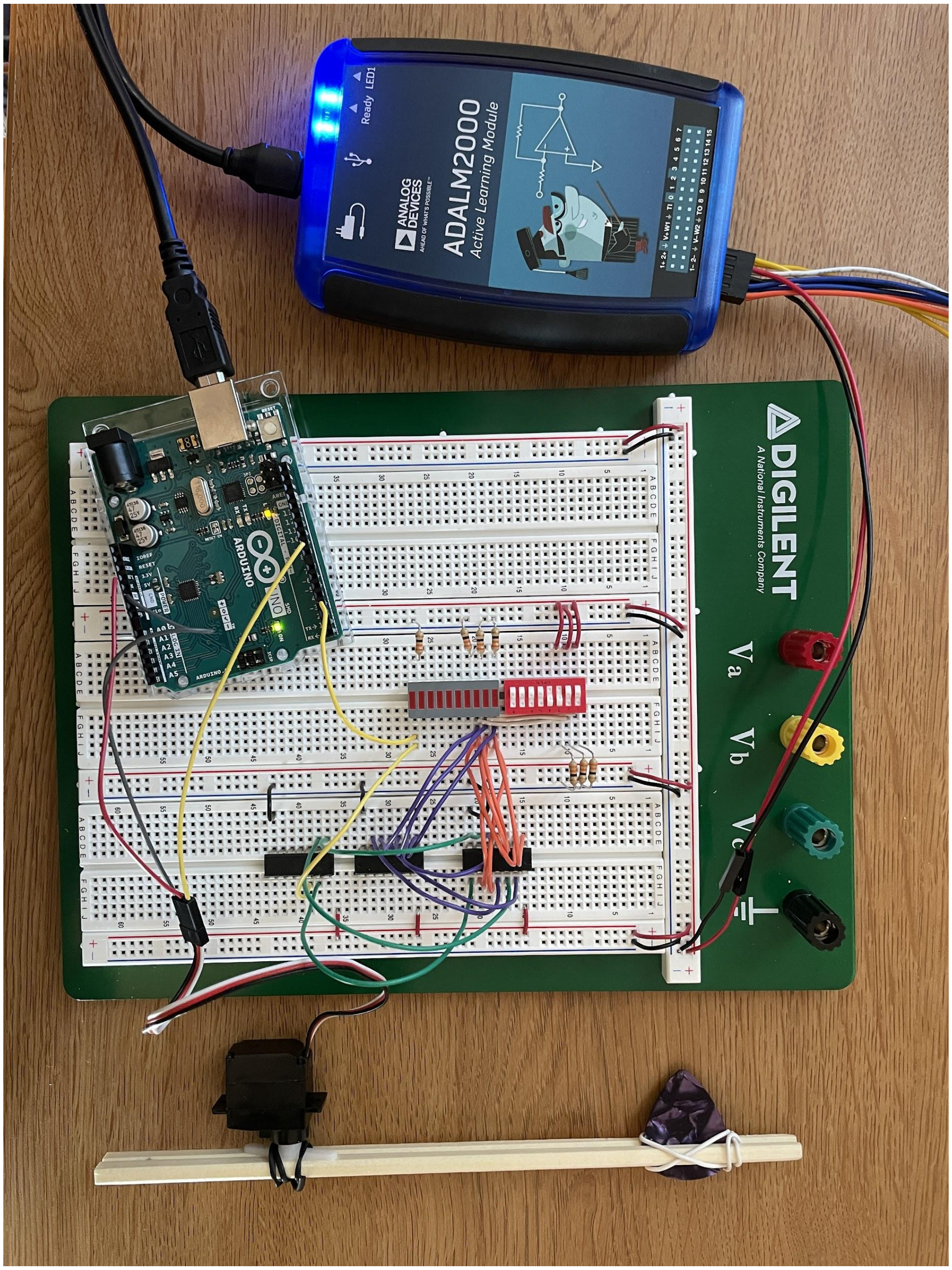
The block diagram for FART can be found in the appendix as Appendix 1. The entire device is powered by a 5V power supply, which provides the charge necessary to run the logic unit, which determines through the Arduino whether or not to run the servo motor which activates the strumming arm.



Appendix 2a - FART Circuit Schematic

The circuit schematic for FART can be found in the appendix as Appendix 2a. The unit consists of 3 3-input AND gates and 1 3-input OR gates, or 4 3-input NAND gates. Essentially, the logic of the unit dictates that the servo motor runs if and only if two of the inputs read HIGH. The equations for the above schematic can be found in the appendix as Appendix 2c.

Results



Appendix 3 - Implemented Design

Depicted above is the completed implementation of the FART design, which can be found in the appendix as Appendix 3. This project utilizes the switch box and LED box provided in the ECE 120 Lab Kit. The first three switches and LEDs are used to dictate the inputs for the logic unit. The last LED is used to signal the output, or whether or not the servo motor should be running. The logic unit uses 2 3-input NAND gate chips and an inverter chip. The output of the logic unit is fed into an Arduino UNO, which reads the output in a loop and uses the read to activate or deactivate a 9g A900 Micro Servo. The code for the Arduino can be found in the appendix as Appendix 4.

Upon testing, the system works through simple logic gate implementation. As long as only two inputs are marked as HIGH inputs, the servo motor executes the strumming code. Upon testing, the output of the circuit reads 1 when inputs are 011, 101, and 110, as expected.

Problems, Challenges, and Future Plans

One problem I faced while working on the project was not knowing how to connect a servo motor to the circuit through the arduino. However, that also contributed to the fun in the project, as each new victory and discovery found through experimentation with the electronics proved to be a fulfilling experience. Using what I learned from labs done in ECE 120, I was able to easily write up and build a circuit that would accomplish the task, but the hard part was figuring out how to connect the circuit hardware with the arduino coding software. However, through the official arduino website and articles from S. Campbell and P. Marian, I figured out that I could simply connect the output to the arduino without having to worry about inputs in the software at all.

If I choose to do more with this project in the future, I plan to build a proper mount for the strumming arm. I would also like to add more possibilities for outputs which lead to a selection of strum patterns.

References

Arduino, "Button | Arduino", *arduino.cc*, 2021. [Online]. Available: https://www.arduino.cc/en/Tutorial/BuiltInExamples/Button. [Accessed: 13- Dec- 2021].

Arduino, 2021. Servo - Arduino Reference. [online] *Arduino.cc*. Available at: <https://www.arduino.cc/reference/en/libraries/servo/>.

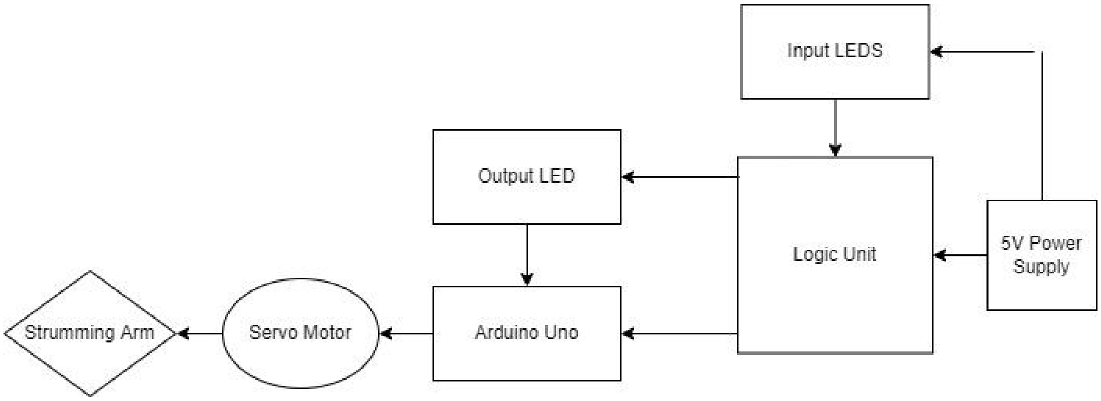
Arduino, "Sweep | Arduino", *arduino.cc*, 2021. [Online]. Available at: <https://www.arduino.cc/en/Tutorial/LibraryExamples/Sweep>.

S. Campbell, "Getting Started with the Arduino - Controlling the LED (Part 1)", *Circuit Basics*, 2021. [Online]. Available: https://www.circuitbasics.com/arduino-basics-controlling-led/.

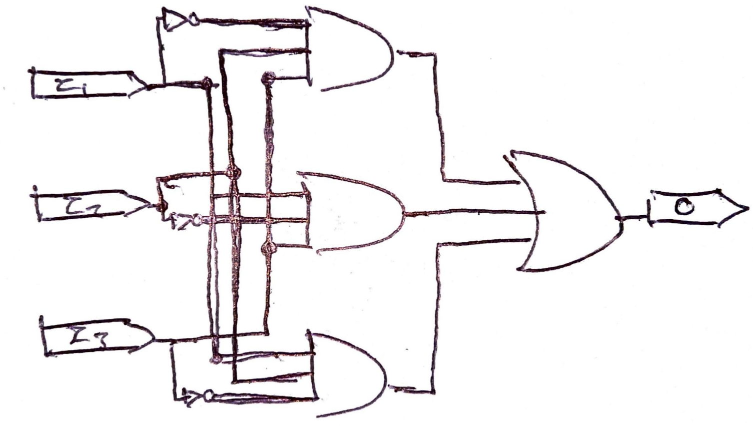
Danilo, D., Alessio, D., Andrea, D. and Giuseppe, P., 2015. Robot Guitar - Arduino - Jessica. [video] Available at: <https://www.youtube.com/watch?v=DBqdnujBSpI>.

P. Marian, "Turn ON an LED with a Button and Arduino - Tutorial #4", *electroschematics.com*, 2021. [Online]. Available: https://www.electroschematics.com/turn-on-led-button-arduino/.

Appendix



Appendix 1 - FART Block Diagram



Appendix 2a - FART Circuit Schematic

I_1	I_2	I_3	O
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

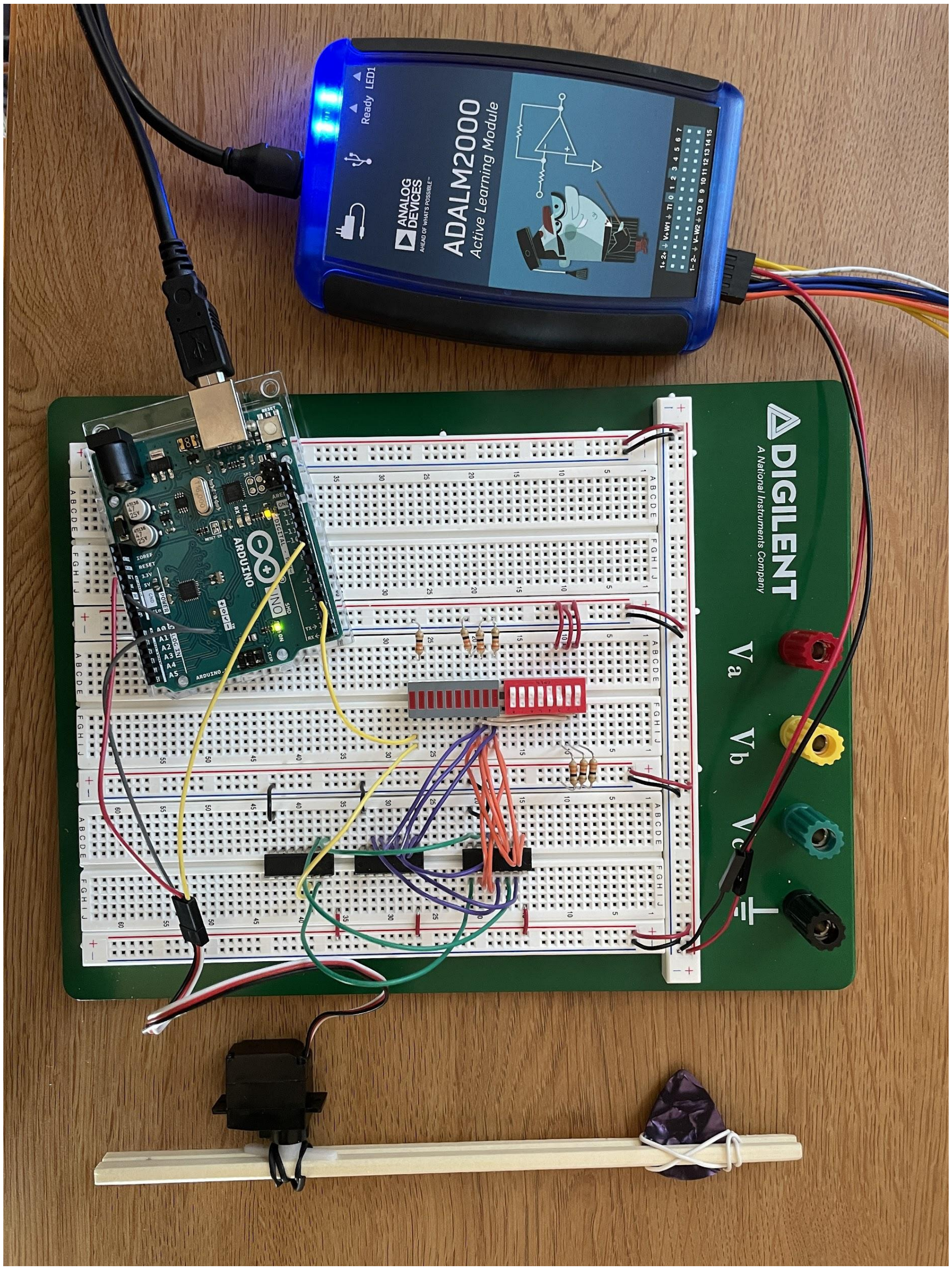
Appendix 2b - FART Logic Table

$I_2 I_3$

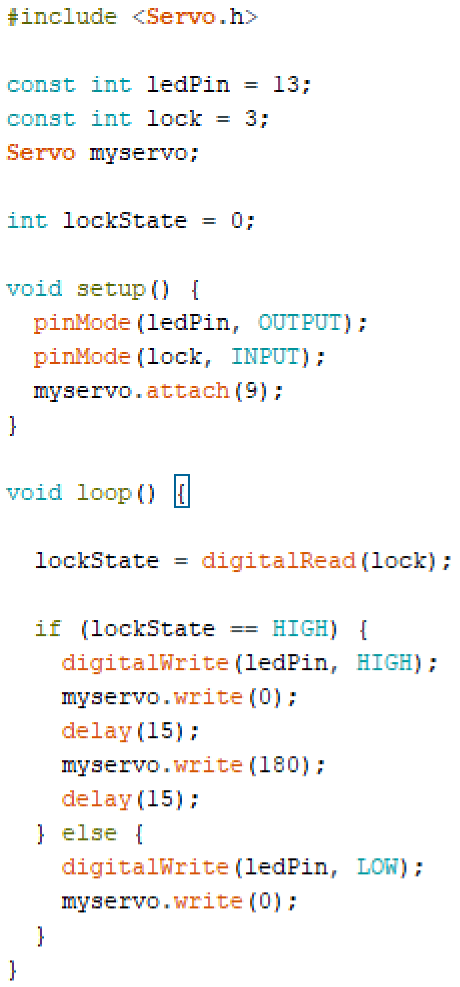
	O	00	01	11	10
I_1	0	0	0	1	0
	1	0	1	0	1

$$O = I_1' I_2 I_3 + I_1 I_2' I_3 + I_1 I_2 I_3'$$

Appendix 2c - FART K-Map and Output Equation



Appendix 3 - Implemented Design



```
#include <Servo.h>

const int ledPin = 13;
const int lock = 3;
Servo myservo;

int lockState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(lock, INPUT);
  myservo.attach(9);
}

void loop() {

  lockState = digitalRead(lock);

  if (lockState == HIGH) {
    digitalWrite(ledPin, HIGH);
    myservo.write(0);
    delay(15);
    myservo.write(180);
    delay(15);
  } else {
    digitalWrite(ledPin, LOW);
    myservo.write(0);
  }
}
```

Appendix 4 - FART Arduino Code