Final Project Report

For project: Pianodo

Leonardo Fusser, 1946995

Project started on **26 January 2021**
Report Submitted on **19 May 2021**

Department of Computer Engineering Technology
Project Planning & Design (247-406-VA)
Dr. Mohamed Tavakoli

VANIER
CÉGEP / COLLEGE
Learning today Leading tomorrow

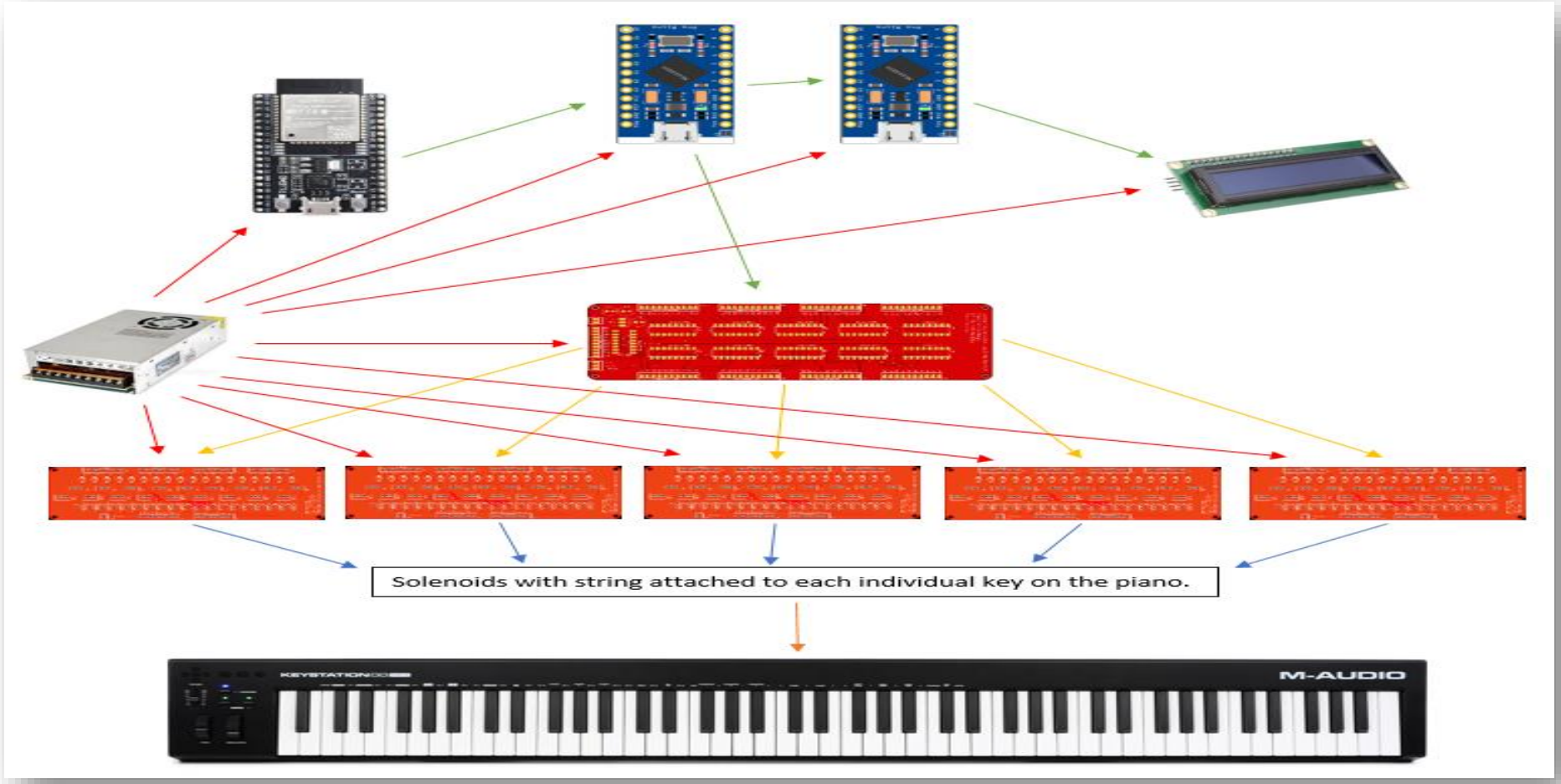
TABLE OF CONTENTS

1.0 Introduction

Quick recap of Pianodo

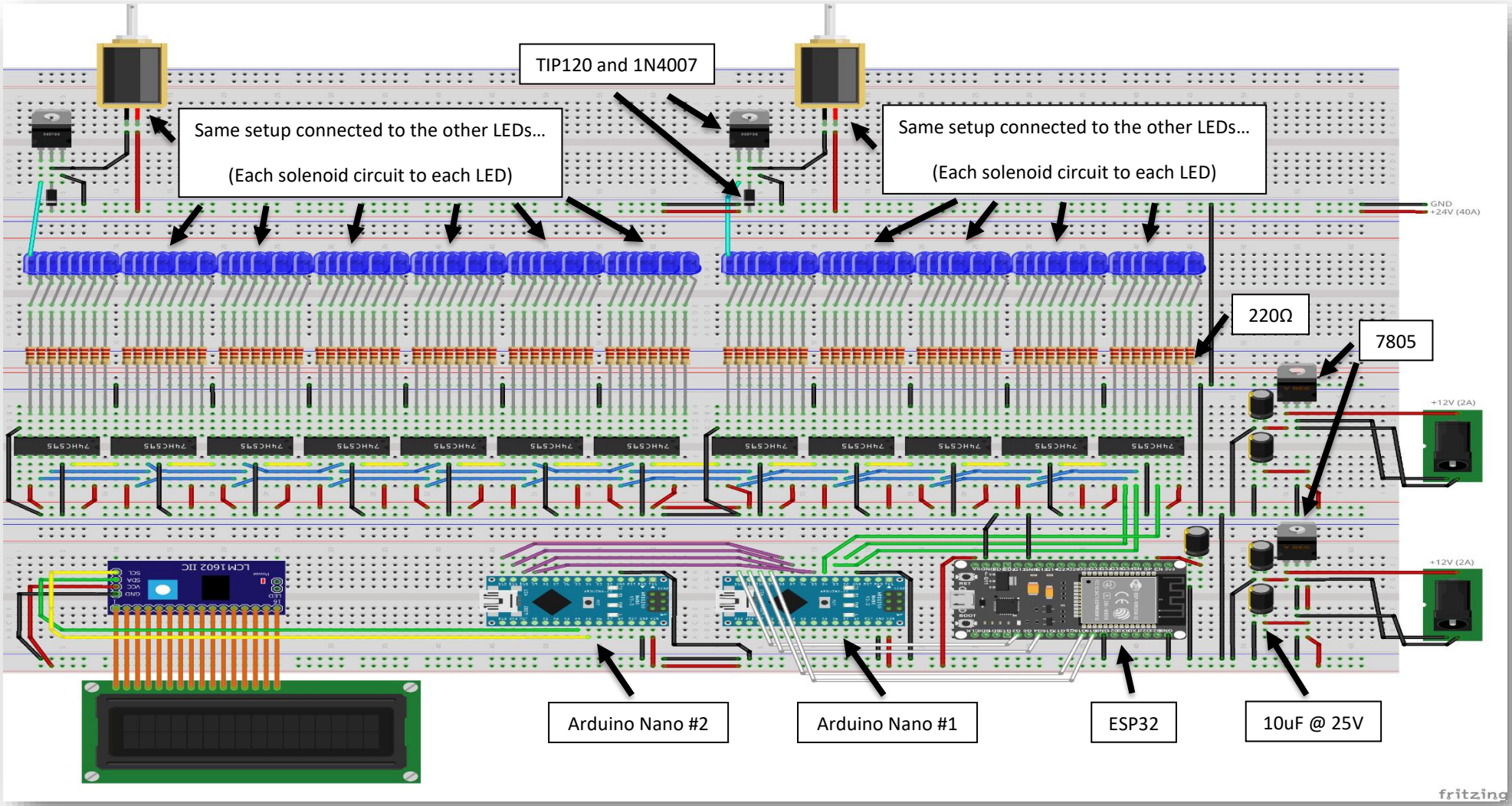
- Pianodo is a device that turns any piano (acoustic or digital) into a self-playing piano that can play a wide range of keys on any standard piano without the need for a human to play it. Pianodo operates separately from the actual piano so the user can still play the piano as if Pianodo was not even a part of it. The user has the flexibility and ability to determine if the whole entire piano or if a certain portion piano can play by itself (by installing the respective number of push-pull solenoids for each individual key). Pianodo is remotely controlled using a mobile phone through the Blynk app and this is how the user is able to control the melodies being played by Pianodo. Unique circuitry and the use of push-pull solenoids make the goal of Pianodo possible. Additionally, there is a 16x2 LCD display that provides additional user interface on top of the remote-control interface for the user.

2.0 Project Assembly



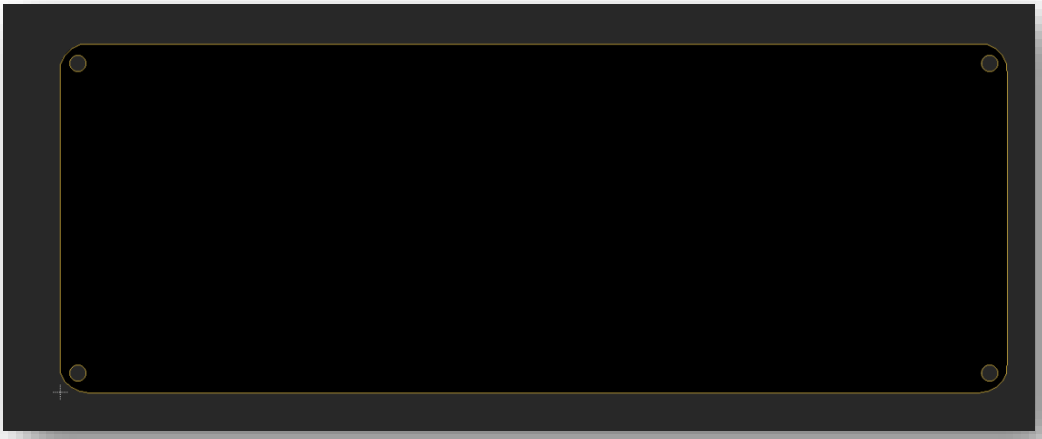
Basic schematic of Pianodo shown above.

- To being the project, a basic overview diagram was created. The basic breakdown of the Pianodo device can be seen in the picture on the previous page. (From top down) An ESP32 development board is used to provide the remote capabilities of Pianodo and interfaces to one of the two Arduino Nano microcontrollers. When the user is using the Blynk app to control Pianodo, they are communicating via Bluetooth to the ESP32 development board and are prompted on the Blynk app for what they can do for Pianodo (prompt is selection of melodies to play). Depending on the user selection in the Blynk app, respective data is sent from the ESP32 development board to one of the two Arduino Nano microcontrollers. This Arduino will take that data and will start playing one of the four melodies the user has selected in the Blynk app (the melodies are stored on this Arduino and not on the ESP32 development board). At the same time, a message will be printed on the 16x2 LCD display, which is connected to the other Arduino. The first Arduino (the one interfacing to the ESP32 development board), also sends out data at the same time when the melody plays to the second Arduino to tell it what is expected to be shown on the 16x2 LCD display (the contents of what is shown on the 16x2 display is stored on the second Arduino). The melody is serially transmitted from the first Arduino to the logic control board (first PCB from top down). The logic control board has 11 74HC595 8-bit shift registers capable of taking the serially inputted melody data and outputting it to 88 parallel outputs (each parallel output is going to each key on the piano). Once the logic is done, each individual parallel output is interfaced to the inputs on the solenoid control boards (the five PCBs under the logic control board shown on the previous page). The solenoid control boards each contain 18 identical BJT circuits, which are basically designed to act as an on-off switches. When an output from the logic control board is 0, the BJT circuit is not turned on and the respective solenoid for its piano key does not press down. The opposite occurs when an output from the logic control board is 1; a sound is produced from the piano. It is important to remember that the outputs from the logic control board are designed to control all the keys on the Piano individually and they all interface individually to each of the 18 BJT circuits on the solenoid control boards. Also, even though the system is designed to work for a full-sized piano (88 keys), other different sized variations of pianos will work as well. The solenoids, which are connected to the outputs of the solenoid control boards, have fish line tied to all the keys on the piano. Power for the ESP32 development board, the two Arduino Nanos, 16x2 LCD display and logic control board come from a 5V regulated source and the solenoid control boards come from a 24V unregulated source. A more detailed schematic can be found on the next page.

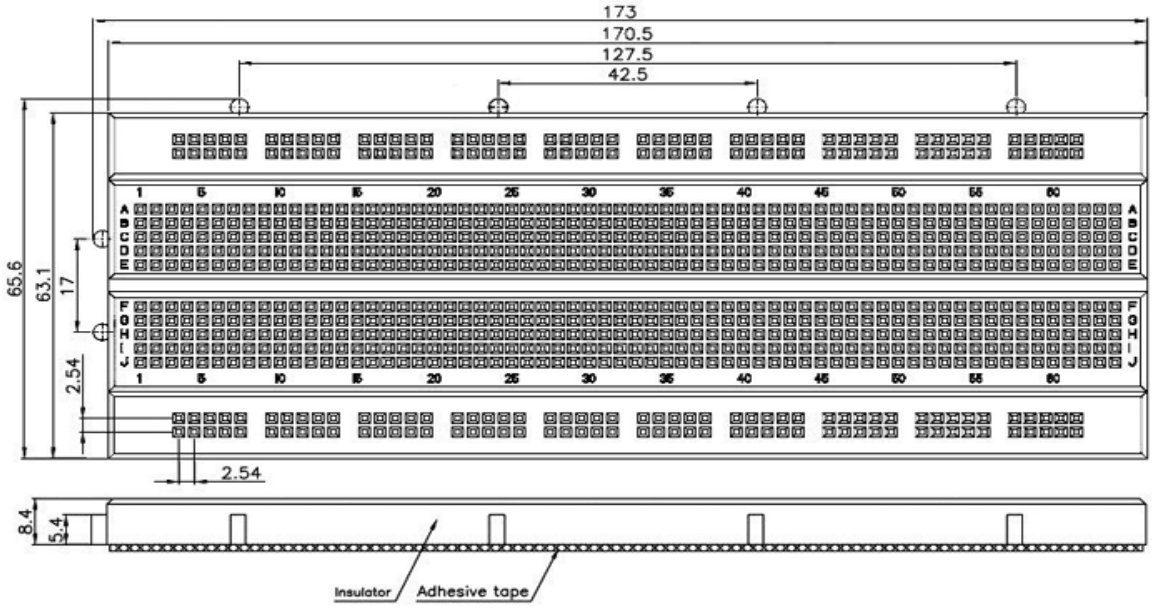


Detailed schematic of Pianodo shown above.

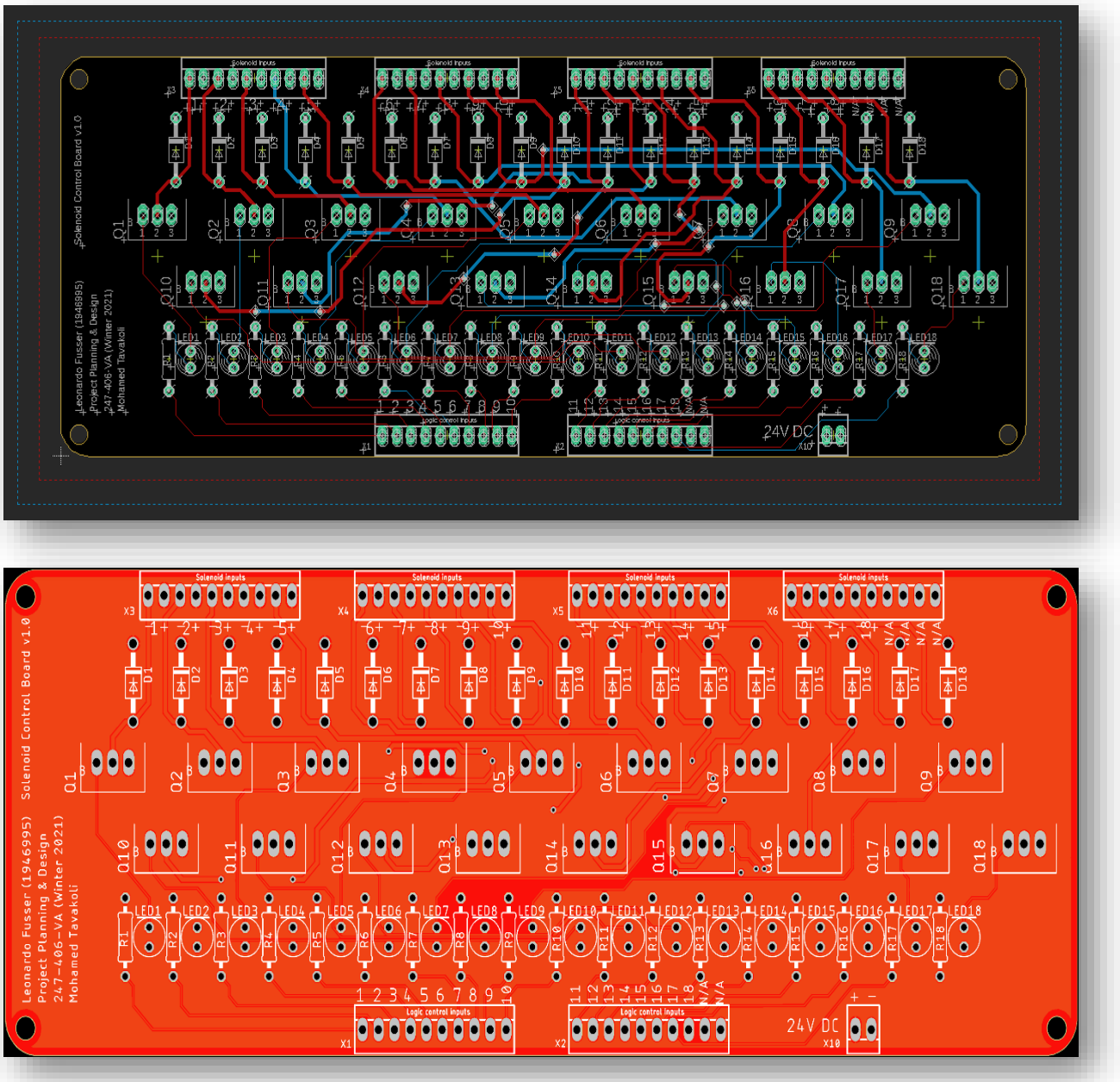
- The complete breakdown of the Pianodo device can be found by looking at the picture on the previous page. The power system consists of a 24V unregulated source for the solenoids and a 12V (regulated to 5V) is for the Arduinos, ESP32 development board, LCD and for the 74HC595 shift registers (this circuitry is on the logic control board). All 88 LEDs connect to all the individual solenoids on the piano (this circuitry is on the solenoid control board). Making sure the overview of Pianodo was good and that all electrical schematics where good, the first step was to produce the actual logic and solenoid control board PCBs in Autodesk EAGLE software.



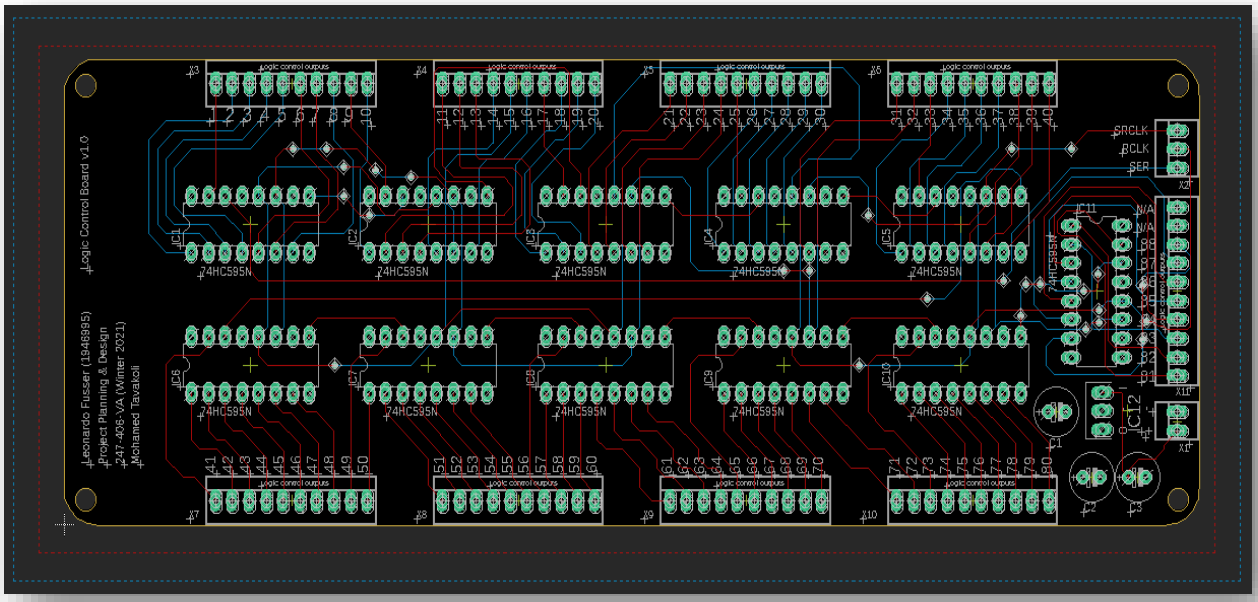
The outline of the two PCBs was created in Autodesk INVENTOR and imported into Autodesk EAGLE software as shown above. The PCBs are a 2-layer board design with appropriate trace sizes.

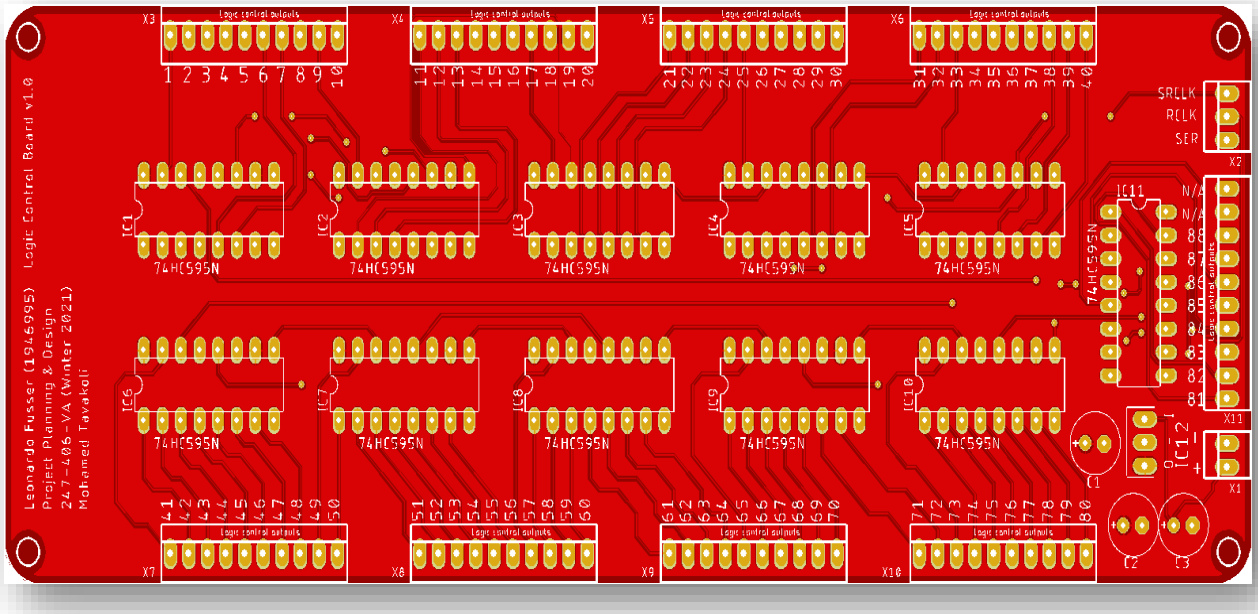


The two board sizes are based off a standard breadboard as seen above.



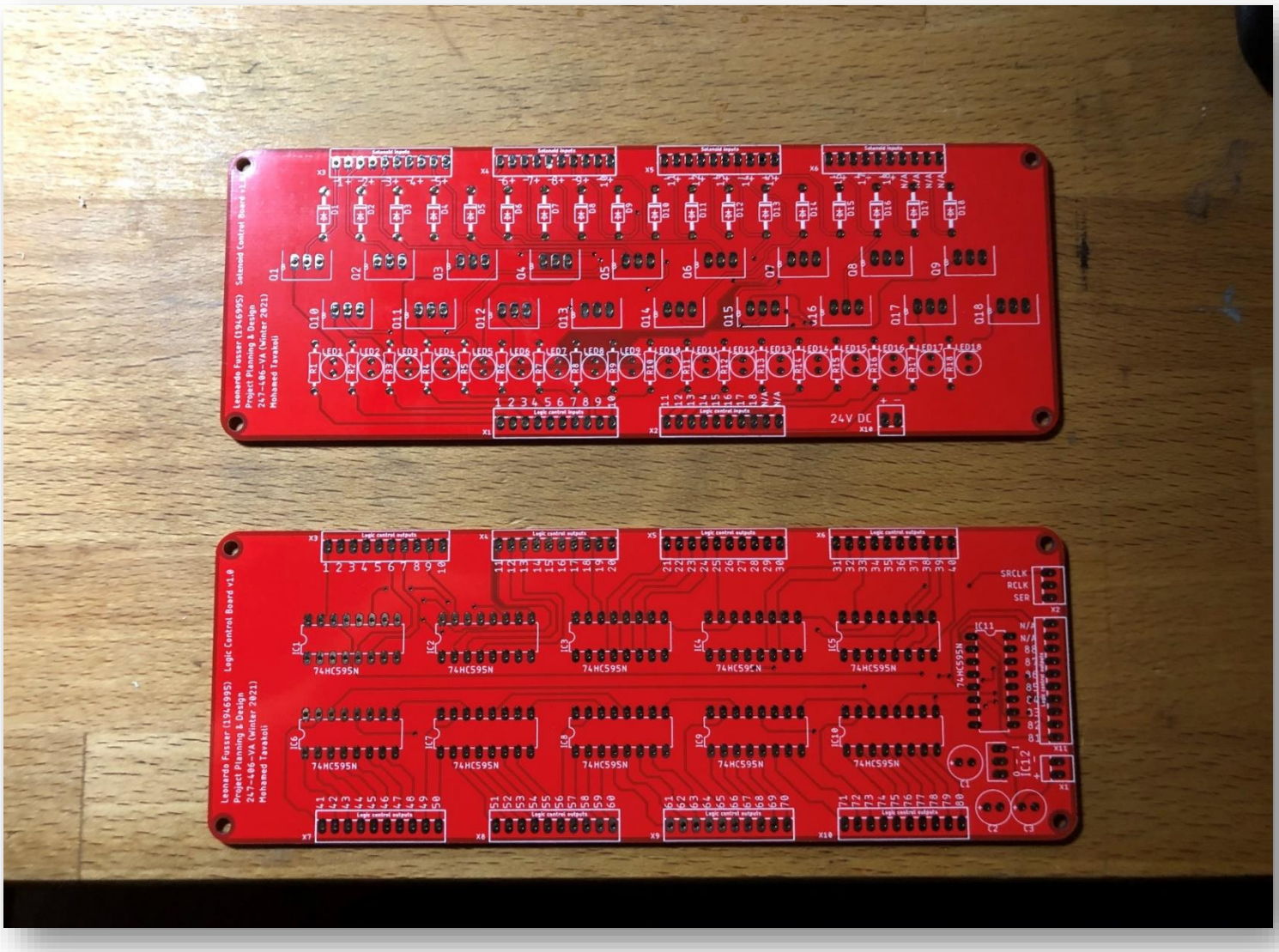
Solenoid control board design (interpreted from diagram on previous page) in EAGLE shown above.





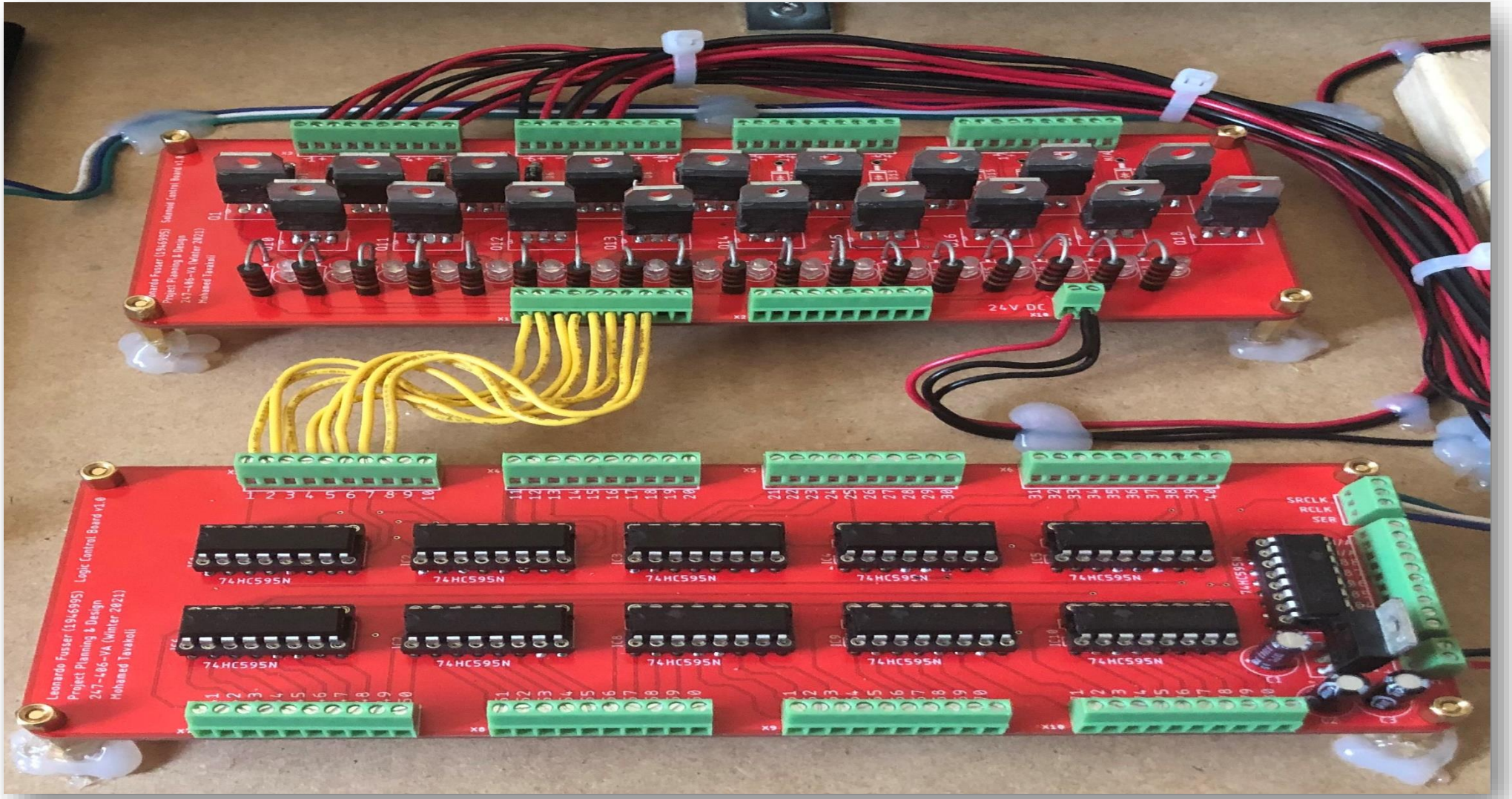
Logic control board design (interpreted from diagram on previous page) in EAGLE shown above.

- After the design of the two PCBs were done in Autodesk EAGLE, the required gerber files were sent to the JLCpcb company in China to produce the boards. The minimum order was five PCBs for each design, so a total of 10 PCBs were produced and sent to me from China. The final result of the two PCBs can be seen on the next page.



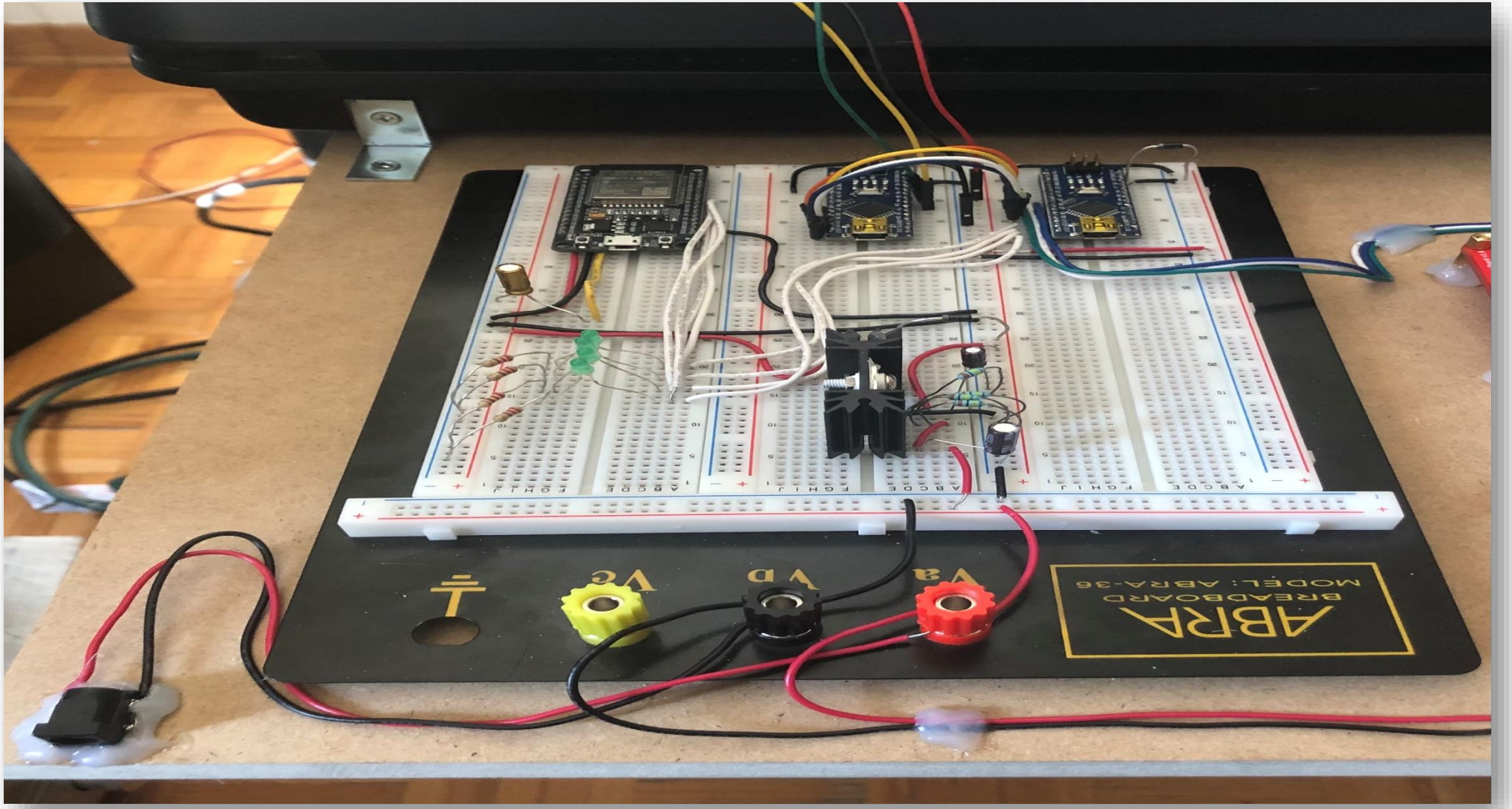
Final result of the solenoid and logic control boards from JLCpcb shown above.

- Once the two boards were received, assembly of components for the boards took place. It is important to note that the components came separate as they were ordered at the same time as the PCBs. The final result of the assembled PCBs can be seen on the next page. It is also important to note that the final result of Pianodo does not utilize all 88 keys on the piano and will be explained why in another section of this report.



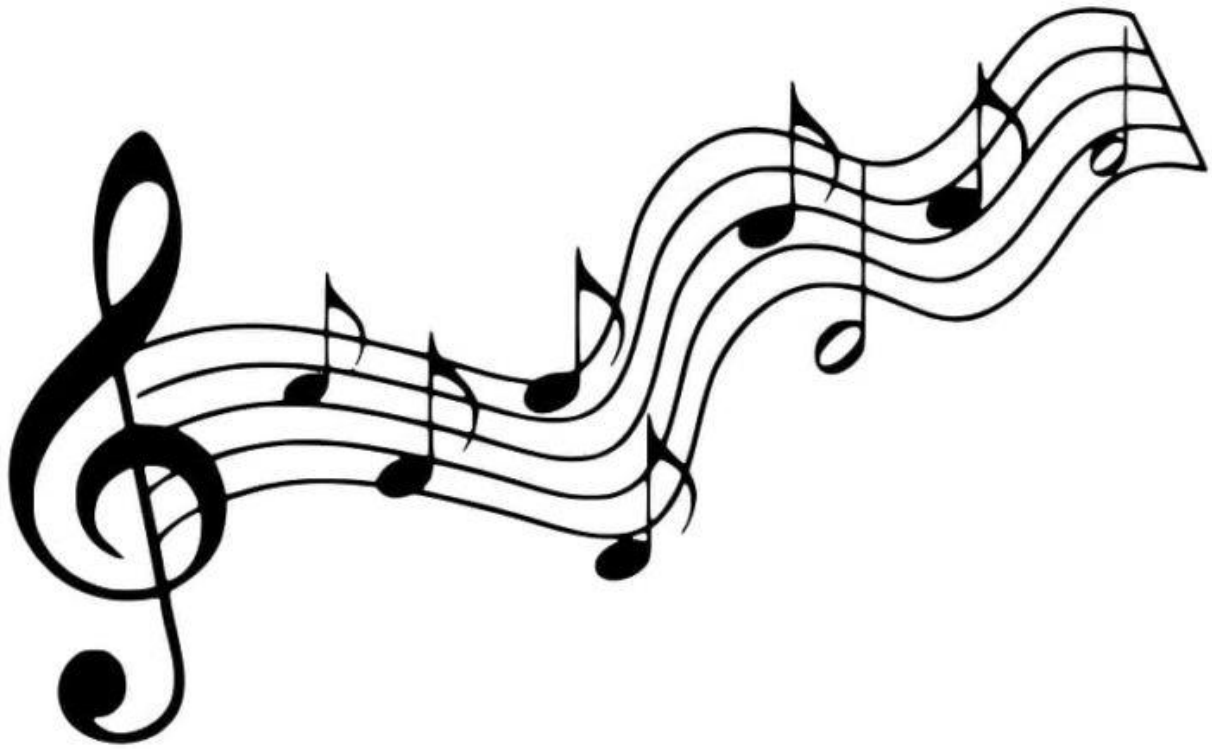
Final result of the two assembled PCBs shown above. Solenoid control board on top and logic control board on bottom.

- Once the two boards were fully assembled, breadboarding was to take place. The components on the breadboard are the two Arduino Nanos, the ESP32 development board, connections to the 16x2 LCD, 5V regulator circuit and wires going to the logic control board. The reason why this was not implemented on its own PCB like with the solenoid and logic control board was because there were uncertainties about how the final design should look like. Power comes in and attaches to the black and red screw terminals on the breadboard. A 12V 2A power supply connects to this breadboard and to the logic control board. The final breadboarded design can be seen on the next page.



Final breadboarding result shown above. The 16x2 LCD display is not shown in the picture.

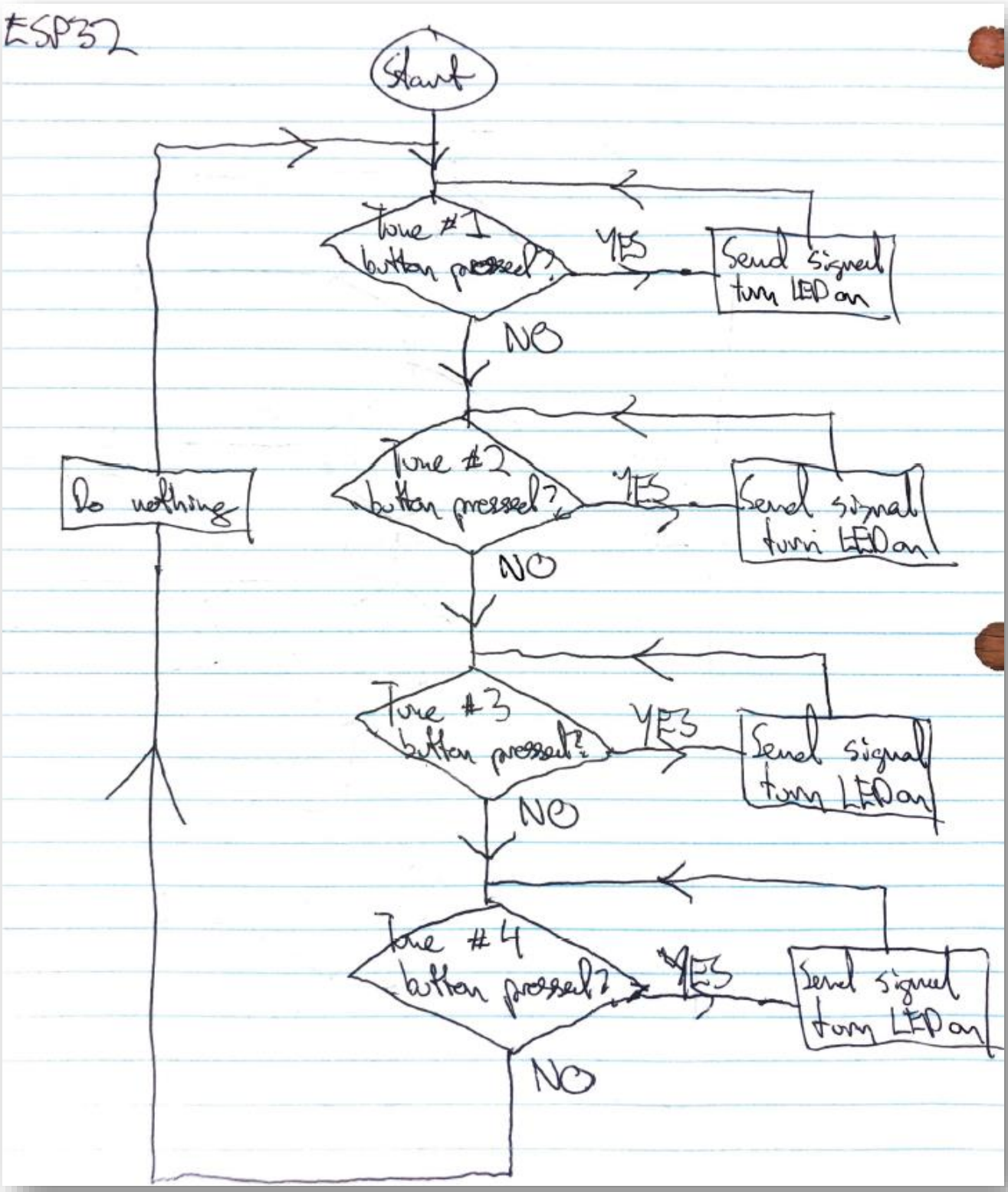
- Once breadboarding was complete, the next step was to proceed with the solenoid assembly to the piano, completing wiring for all the boards and different devices and physically placing all the components on a single piece of MDF board and securing them to it. A combination of hot glue and zip ties were suited to be fit to secure the components to the piece of MDF board. A 24V power supply connects to the solenoid control board and is placed next to the solenoids. Fish line connects the individual solenoids to the keys on the piano (it will be shown that there are only five solenoids as opposed to 88 and will be explained why later in this report). For each solenoid, an elastic is holding back the solenoid plunger from being ejected completely of the shaft inside the solenoid (if this were not present, Pianodo would only be able to play one set of notes once and would require the user to keep placing the solenoid plungers back in the solenoid shaft = not good!). The final physical work can be seen on the next few pages.



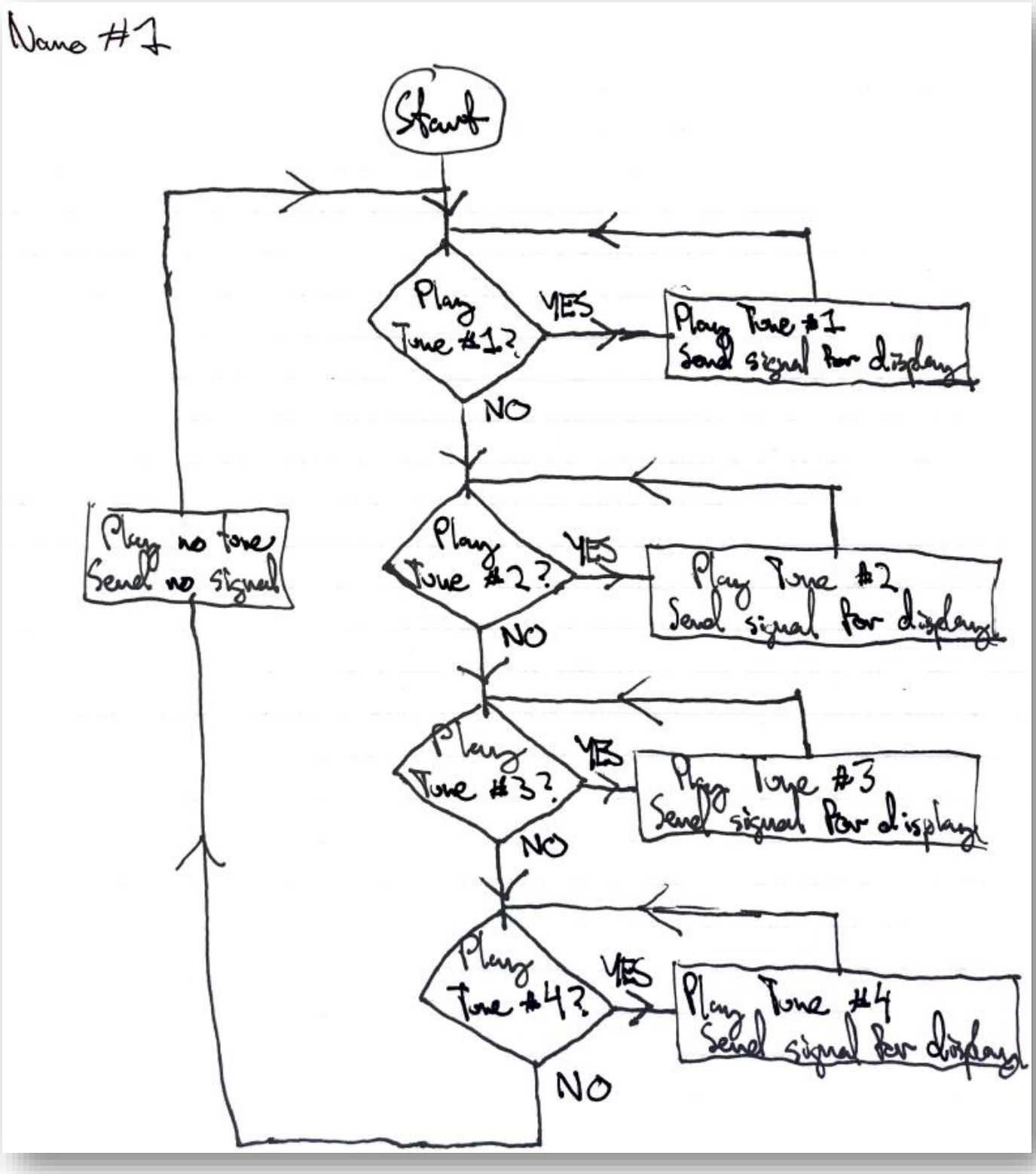
The final solenoid installation, wiring and 24V power supply powering the solenoids shown above.

The final piano installation to the MDF board and all the other components attached to the MDF board shown above.

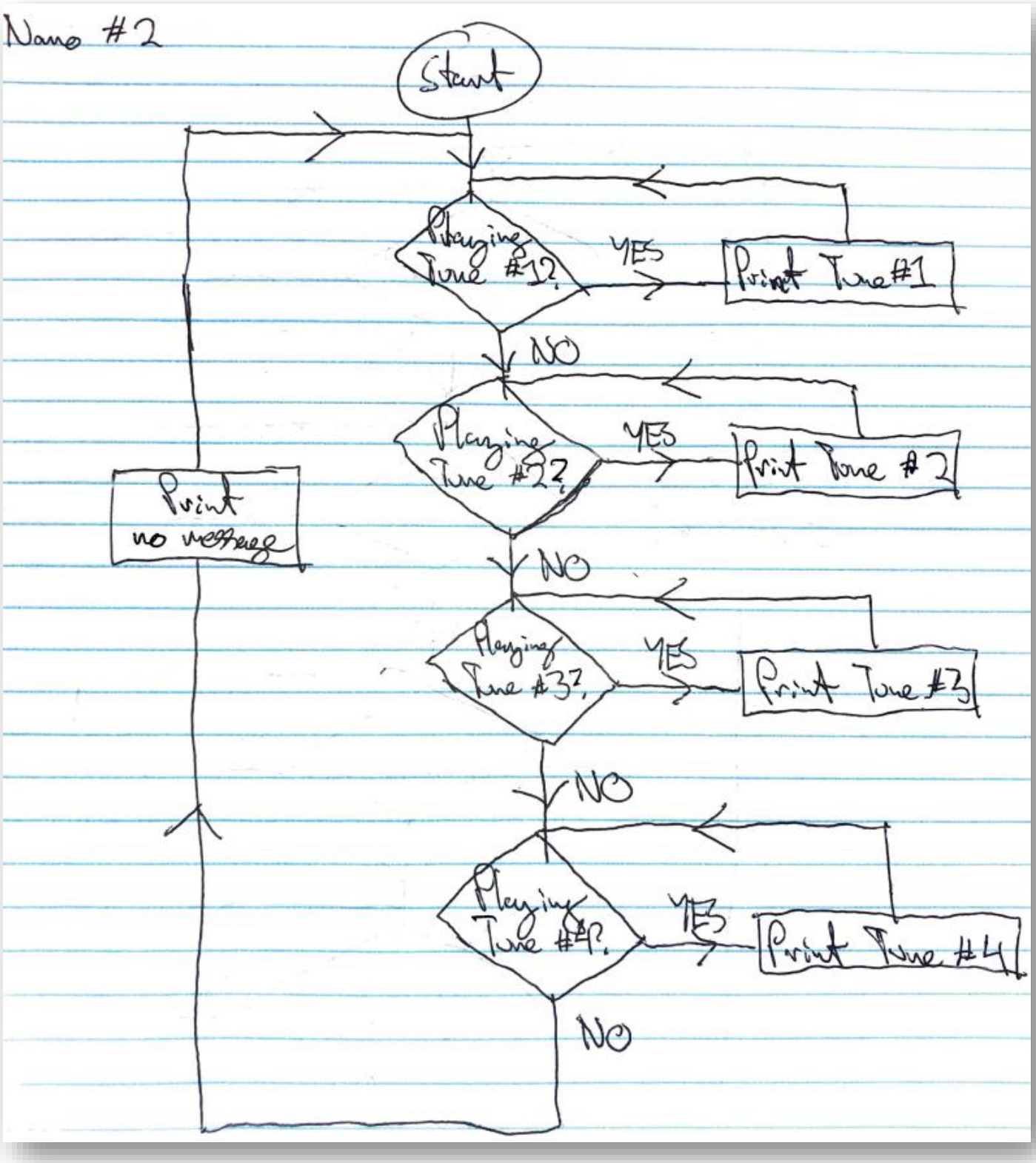
- Once all remaining physical and electrical work was complete, the remaining task was to program the Pianodo device. Three devices needed to be configured and programmed: the 1st Arduino Nano, the 2nd Arduino Nano and the ESP32 development board. For the ESP32 development board, a special library specific for the Blynk app was used in order to get the ESP32 development board to communicate via Bluetooth with my mobile device. At the same time, the ESP32 development board had to understand what to do when the user selected one of four buttons in the Blynk app and output appropriate signals to the 1st Arduino Nano. The reason why there are only four buttons on the Blynk app and not more features will be explained later in this report. For the 1st Arduino Nano, it had to handle the incoming data from the ESP32 development board, output the correct melody serially to the logic control board and output data going to the 2nd Arduino Nano in order to get a message printed on the 16x2 LCD. This had to be programmed so that the 1st Arduino Nano was able to perform all these tasks simultaneously. The 1st Arduino Nano required a unique library in order for the melodies to be transmitted serially to the logic control board. The melodies are stored on this Arduino and were programmed by using specially constructed for loops (and sometimes multiple for loops) to cause the shift registers on the logic control board to create unique outputs and make the solenoids play their respective keys (the datasheet for the 74HC595 shift registers helped with this). This is also visually animated with the LEDs on the solenoid control board (LEDs turn on when a key on the piano is pressed and turn off when no key is pressed). The last Arduino (2nd Arduino) had to be programmed to handle the data coming from the 1st Arduino to print unique messages on the 16x2 LCD display attached to it. A unique message was to be printed depending on the data received. A unique library was needed to get the 16x2 LCD display to print messages to the user. The software used to program all three devices was done in the Arduino IDE software. The corresponding basic logical flow charts are shown on the next few pages for each of the three devices. The code for each of these three devices will be included in the zip folder for this report submission. It is important to note that things such as initialization behavior is not shown in the flowcharts on the next few pages.



Flow chart describing ESP32 development board behavior shown above.



Flow chart describing 1st Arduino Nano behavior shown above.



Flow chart describing 2nd Arduino Nano behavior.

3.0 Conclusions

- The project was a partial success, mainly due to issues related to the solenoids. Even though the design was correct to support Pianodo to play a full-sized piano (88 keys) by itself, the issue was that the solenoids deemed too costly to attach to all the keys on the piano so a reduced number was needed in order to respect the allocated $200 budget. To compensate for this, it was calculated that using 24 solenoids was deemed sufficient to still play certain melodies. 24 solenoids were then purchased from China and according to provided information, was deemed fit to fulfill the needs it was going to be put through (it was calculated that the solenoid needed to be able to handle between 3.5 and 4N of force to successfully press a key on the piano fully down. Upon receiving the 24 solenoids from China, a quick test with one of them showed that in fact they were not able to pull the key to the piano down at all. This happened due to misleading information that was provided to me at the time of purchase. In order to still respect the $200 budget, a small set of solenoids (purchased one before at the very beginning in order to test and view needed requirements to get the piano key pressed down) were purchased in order to get Painodo somewhat working. Due to this, the original idea of implementing MIDI so that the user can use any MIDI file to make Pianodo play MIDI tunes was abandoned due to the unworthy complexity (MIDI can be only taken advantage of when a decent number of keys are played on the piano). Instead, four pre-programmed melodies were programmed on the 1st Arduino Nano so that the user can select Pianodo to play one of 4 different melodies. Aside from solenoid problems, two other issues occurred. First, the 7805-voltage regulator on the breadboard would be getting very warm so in order to prevent the regulator from burning up, a heatsink was put in place to keep the regulator working. Secondly, an issue was encountered with the ESP32 development board upon uploading code to it. The issue was resolved by placing a 10uF capacitor between the EN and GND pin. Only after these multiple issues resolved, was Pianodo finally able to play by itself.

4.0 Final Cost Breakdown

After all work, here is the final cost breakdown:

- 16x2 LCD display: $5.58
- 74HC595 8-bit shift registers: $7.88
- TIP120 power transistors: $4.32
- 10 pin headers: $10.04
- 3-pin headers: $2.75
- 2-pin headers: $2.03

- Blue LEDs (for solenoid control board): $1.19
- 1N4007 diodes (for solenoid control board): $6.77
- DIP16 sockets (for logic control board): $9.33
- Arduino Nanos: $15.99
- ESP32 development board: $15.19
- Custom PCBs from JLCpcb: $50.28
- 5 solenoids from Addison: $22.94
- 24 solenoids from AliExpress: $58.82
- Heatsink and thermal compound for 7805 regulator: $2.00
- Wire and capacitors: Free
- Fish line: Free
- Solder: Free

Grand total: $215.11

5.0 Possible Improvements

- Even with the unexpected results with the solenoids, the possibilities of imperfections were still minimized as careful planning prevented any corners being cut and resulting an overall unsatisfactory result. The one possible improvement that would have been useful is the use of having made a custom enclosure for the 16x2 LCD and creating a PCB to replace the breadboard that was used. The other possible improvement is that if there were more readily available solenoids at a very cheap price so that the true potential of the project can be shown.