FEDERAL UNIVERSITY OF SÃO JOÃO DEL-REI – UFSJ

LABORATORY OF NEUROENGINEERING AND NEUROSCIENCE – LINNCE

OPEN-STIM

OPEN SOURCE HARWARE ARDUINO BASED ELECTRICAL STIMULATOR

ASSEMBLY INSTRUCTION MANUAL

Brazil, May 2021

OPEN SOURCE HARWARE STIMULATOR

ASSEMBLY INSTRUCTION MANUAL

[1] FEDERAL UNIVERSITY OF SÃO JOÃO DEL-REI – UFSJ

[2] NEUROENGINEERING AND NEUROSCIENCE INTERDISCIPLINARY LABORATORY – LINNCE

DEVELOPED BY:

[1 2] Maikon Lorran Santos - maikon.lorran@hotmail.com

[1] João Daniel Nolasco - joaodani@hotmail.com

[1 2] Vinícius Rosa Cota: - vrcota@ufsj.edu.br

Address: Pça. Frei Orlando, 170, Centro – São João del-Rei – Minas Gerais - Brazil

This manual will assist builders in the construction and assembly of the Open Source Hardware Electrical Stimulator. All steps are described in detail, showing the diagrams, PCB, simulation, codes, and photos.

Brazil, May 2021

SUMMARY

CHAPTER 1

DESCRIPTION OF MAIN GOALS

The main objective of the project is to develop and build a low-cost electrical stimulator for experimental electrophysiology based on the Arduino platform. The idea is to build a Shield capable of controlling the output current and keeping it constant even if the load changes, that is, independent of the animal/tissue being stimulated and possible changes of impedance.

In the stimulator configuration, it is possible to vary the frequency, the pulse width, the pulse phase, that is, positive, negative or both and of course, the current. Current amplitude is configured by two wave switches, also known as the rotary switches For frequency, pulse width and pulse phase, the Nextion display will be used, which is a Touch Screen that has its own processor and its own programming language, but which communicates very well with several platforms, being one of most used, the Arduino.

The entire stimulator is powered by four 9 V batteries. These batteries are monitored by an exclusive circuit, which indicates when it is necessary to change or recharge them.

The entire project is built with precision resistors, with error less than or equal to 1%. To accommodate the boards and display, a custom box is built with 3D printing. We highlight that all software used in developing this project have free or student licenses, which do not prevent or compromise the construction of the stimulator or modification of the design.

The project has a CC BY-SA 4.0 license which is accepted by the Open Source Hardware Association (OSHWA); the details of this license can be found in the next chapter.

Finally, with this project, it is expected that:

1) Laboratories from any part of the world can have a stimulator due to its easy and low-cost construction. Standard stimulation protocols (e.g. fixed frequency) is promptly available. Yet, once it is customizable, users can implement their own protocols by changing control software.

2) The design is able to raise the interest of students, academics or even lay people (hobbyists) in the subject of electrical stimulation as a powerful investigational tool and as a means for the treatment of chronic disorders. In this way, we will have more thinking heads devoting themselves to the same important objective.

3) That the stimulator design is improved by qualified and extensive peer review. We want it to have more versions, with its hardware and software improved.

4) Let it be a way of including people in the doing of science. With such a simple and inexpensive design, we believe that even in a school it will be possible to build and teach something about

electronics, programming, 3D printing. In short, it is possible to do science without high cost and include and encourage more people in favor of science.

Given the above, we would like to present the world, OPEN STIM!

CHAPTER 2

SPECIFICATIONS

2.1 GENERAL SPECIFICATIONS:

The stimulator has the following specifications in its current version (1.0):

- **Electric Current:** from 15 µA to 1200 µA
- **Frequency:** from 0.1 Hz to 300 Hz
- **Pulse Duration:** from 4µs to 1000ms
- **Pulses polarity:** positive and negative (cathodic or anodic, depending on wiring)

2.2 WEB SITE:

The entire project is hosted on GitHub. You can access it using the link:

https://github.com/Open-Stim/openstim

In the above GitHub directory, it is possible to have complete access to the files, such as: Shield PCB, diagrams, simulations, Arduino codes and Nextion display, STL files for 3D printing, in addition to the photos of each construction and assembly step.

2.3 LICENCE:

The project is distributed under the **Creative Commons BY-SA 4.0 license**. With this license you will be able to:

Share - copy and redistribute material in any medium or format

Adapt - remix, transform, and create from the material for any purpose, even commercial.

According to the CC BY-AS 4.0 the terms are:

2.4 ARDUINO LEONARDO

The choice for this project was the Arduino Leonardo, due to its low cost and compatibility with the project specifications. The documentation for the arduino Leonardo can be found at: https://www.arduino.cc/en/Main/Arduino_BoardLeonardo. Figure 1 shows the Arduino Leonardo board.

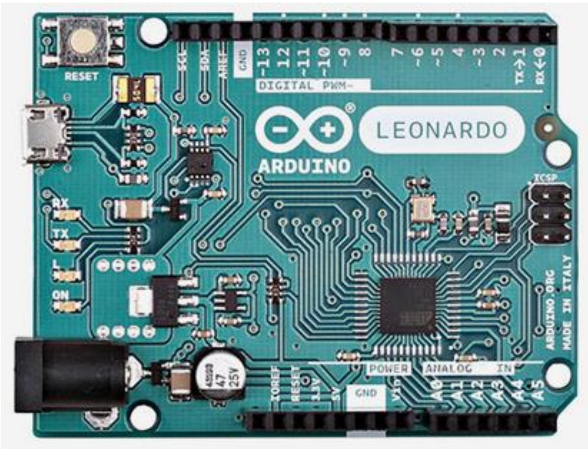


Figure 1 – Arduino Leonardo

Some of the specifications of the Leonardo Arduino are:

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (Recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.3 mm
Weight	20 g

The complete documentation for the Atmel 32u4 processor can be found at: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf

2.5 NEXTION DISPLAY

The tft display chosen was the Nextion 3.2 "model NX4024T032. It is among the most basic models available, and of course has a lower cost. It has a resolution of 400 x 240 Pixels, RAM memory of 3584 bytes, Flash memory of 4 MB, screen resistive touch, 5 V power supply, requiring 85 mA (at maximum brightness) and 15 mA (at minimum brightness). Additionally, it has an input for a microSD memory card of up to 32 GB formatted in Fat32. Figure 2 shows the Nextion display used.

Figure 2 – Display Nextion 3.2”- NX4024T032

The manufacturer's official website is: https://nextion.tech/ and the datasheet of the model in question can be accessed at: https://nextion.tech/datasheets/nx4024t032/

As it can be seen in Figure 2, only 4 connections are required for the display to work, namely: 5V and GND for power and RX and TX for communication with Arduino.

2.6 OPERATIONAL AMPLIFIERS

Only two operational amplifiers are used: TL072 and LM741. The choice of TL072 is due to its low noise and it is responsible for the main part of the project, generating pulses at the output. The LM741 is chosen for its low cost and will be responsible for monitoring the batteries, indicating the time for changing or recharging. The TL072 and LM741 datasheet can be accessed through the links below:

TL072: https://pdf1.alldatasheet.com/datasheet-pdf/view/28775/TI/TL072.html

LM741: https://pdf1.alldatasheet.com/datasheet-pdf/view/66375/INTERSIL/LM741.html

Figure 3 shows the TL072 operational applier and Figure 4 the LM741 operational amplifier. The connection of the TL072 pins is shown in Figure 5 and the LM741 in Figure 6.

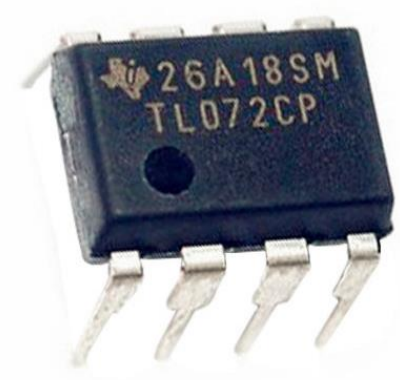


Figure 3 – TL072

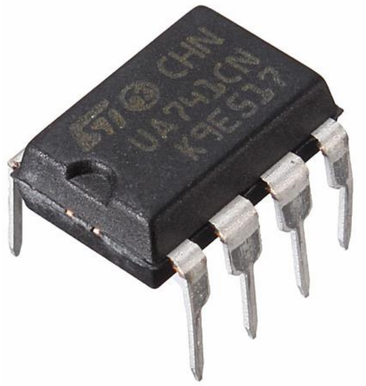
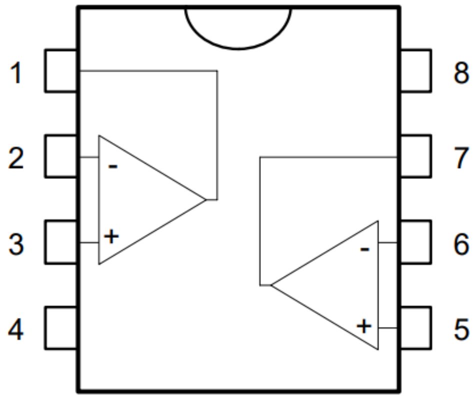
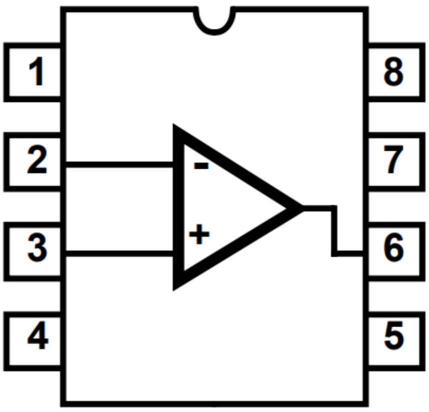


Figure 4 – LM741



1 - Offset null 1
2 - Inverting input 1
3 - Non-inverting input 1
4 - (-) VCC
5 - Non-inverting input 2
6 - Inverting input 2
7 - Output 2
8 - (+) VCC

Figure 5 – Connection pins TL072



1 - Offset null
2 - Inverting input
3 - Non-inverting input
4 - (-) VCC
5 - Non-inverting input 2
6 - Output
7 - (+) VCC
8 - NC

Figure 6 – Connection pins LM741

CHAPTER 3

DIAGRAM AND SIMULATION

3.1 – DIAGRAM CONSTANT CURRENT CIRCUIT

The software used for simulation is LT Spice, which can be downloaded through the link: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html# (available to Windows and Mac).

Figure 7 shows the Open Stim diagram. Next, the circuit's operation will be detailed.

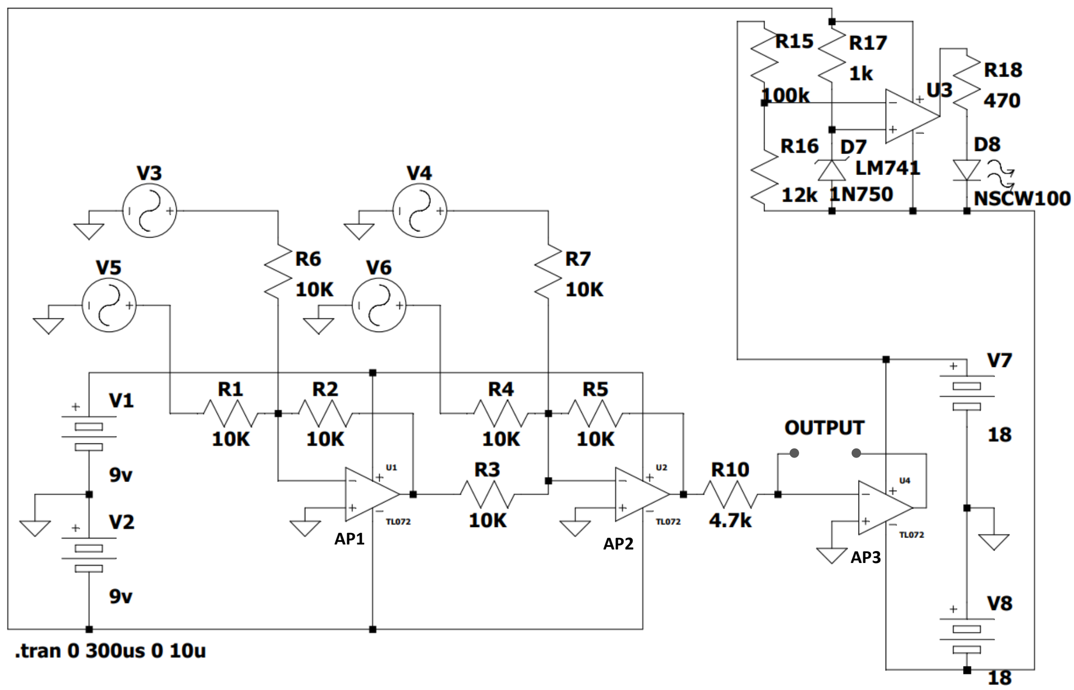


Figure 7 - Electronic Diagram

We started the analysis with the first operational amplifier represented by AP1 (TL072) connected in the configuration of the adder with the voltages of V3 and V5, which has its connection according to Equation 1. As there is no amplification because all the resistors of R1a R7 are all the same at 10 KΩ, we can cancel, and simplify the Vout1 equation.

The voltage output Vout1 is the input of the second operational amplifier (AP2) together with V4 and V6, and therefore, all will be added together. Following the same analogy as Equation 1, we have Vout2 in Equation 2. Simplifying again due to the equal resistors, we come to Equation 3. Equation 4 is just the algebraic manipulation of Equation 3.

Therefore, we have to: V5 and V6 generate the main waves, and V3 and V4 cut the main waves so that the desired pulse widths are obtained. So, we have V5 being cut by V4 and V6 being cut by V3. Without the 4 waves, it would be impossible with a certain fixed frequency to vary the pulse width. Another consideration is that the negative wave is generated in the first stage, that is, in AP1 and the positive wave in the second stage, in AP2.

$$V_{out1} = -\left(\frac{R2}{R1} V_5 + \frac{R2}{R6} V_3\right) = -(V3 + V5) \tag{1}$$

$$V_{out2} = -\left(\frac{R5}{R3} V_{out1} + \frac{R5}{R4} V_6 + \frac{R5}{R7} V_4\right) = -(-(V3 + V5) + V4 + V6) \tag{2}$$

$$V_{out2} = \left((V3 + V5) - V4 - V6\right) \tag{3}$$

$$V_{out2} = (V5 - V4) - (V6 - V3) \tag{4}$$

In the Arduino, the pins 6 and 10 generate the negative pulse, while pins 5 and 12 generate the positive pulse. Pins 5 and 6 are the main waves and the waves from pins 10 and 12 are responsible for cutting the main waves.

The AP3 (TL072) amplifier receives the signal according to the configuration made by the user in relation to the number and polarity of phases, frequency and duration of the pulse. It is connected as a voltage-current converter. Resistor R10 controls the current and is actually the resistor board containing 23 resistors, ranging from 4.02 KΩ to 360 KΩ that translates to a current between 15 µA to 1200 µA. As previously mentioned, the signal received by AP3 is at the inverter port, therefore, the signal will be inverted again, since the configuration in this last stage is an inverter amplifier, which is described by Equation 5.

$$i_{out} = -\frac{V_{out}}{R10} \tag{5}$$

An important observation is that the resistor R11 in the diagram represents the output, that is, the brain of the animal, and values considered possible are 1KΩ and 10KΩ, and these were also considered in the tests. See folder with test files on GitHub for more details.

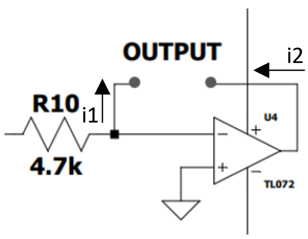


Figure 8 - Current i1 is the same as current i2 considering an ideal operational amplifier

3.2 – BATTERY MONITORING CIRCUIT

In battery powered systems, control of voltage level of the power supplies, to indicate when to recharge or change batteries is essential. As these are 9 V batteries, this project established that at 7.5 V it is necessary to change or recharge them. To better understand how this part of the circuit works, we will go back to analyzing Figure 7.

The responsible for the control of the voltages is the operational amplifier LM741 operating in the comparator configuration. Analyzing each component of the circuit we have: the 3.3 V zener diode, which is represented by D7, the 1 KΩ R17 resistor that limits the current in the zener diode, the 470 Ω R18 resistor that limits the current in the D8 led. The resistor value of 100 KΩ of R15 was chosen so that the current of the circuit was low, improving the efficiency of the circuit. Given the choices made, it is now possible to determine the value of resistor R16 through equation 3 and 4, considering the threshold of 7.5 V and the comparison value of the zener diode of 3.3 V.

Being 4 batteries in 7.5 V, we have 30 V, like this:

$$\frac{R16}{R16 + R15} = \frac{3.3}{30} \tag{6}$$

Doing y = R15 = 100KΩ and we find x:

$$30(R16) = 3.3(R16) + 3.3(100K) \tag{7}$$

$$R16 = 12{,}36\ K\Omega$$

The closest commercial value is 12 KΩ, which is exactly the value used in this project for R16. Therefore, the D8 led will rise when the battery voltage is below 7.5 V. This is easy to understand, while the voltage in the zener is less than the voltage over resistor R16, the output of the LM741 is zero, when the voltage of the zener to be greater than the voltage drop over R16, the output is activated.

3.3 – CONNECTING THE BATTERIES

It is necessary to use 4 different voltages, namely: +18 V, -18 V, -9 V and +9 V, in addition to the reference point. The +9 V and -9 V voltages are necessary to supply the amplifiers that generate the waves and make them go down (AP1 and AP2). The +18 V and -18 V voltages feed the AP3 converts voltage to current, which demands larger voltage amplitudes, considering maximum output current amplitudes and tissue impedance. For instance, to apply 1200 µA to a 10 KΩ output impedance, an amplitude voltage of 12 V is needed which is within our voltage compliance.

The maximum supply voltage of the TL072 is exactly +18 V and - 18 V. According to the manufacturer's datasheet, values are higher may damage the circuit (see datasheet of the maximum voltage and current values of the TL072). These values were chosen so that it is possible to generate all 23 electrical currents of the project. Next in Figure 9, there is the correct way to connect all 9 V batteries in series to obtain all the necessary supply voltages.

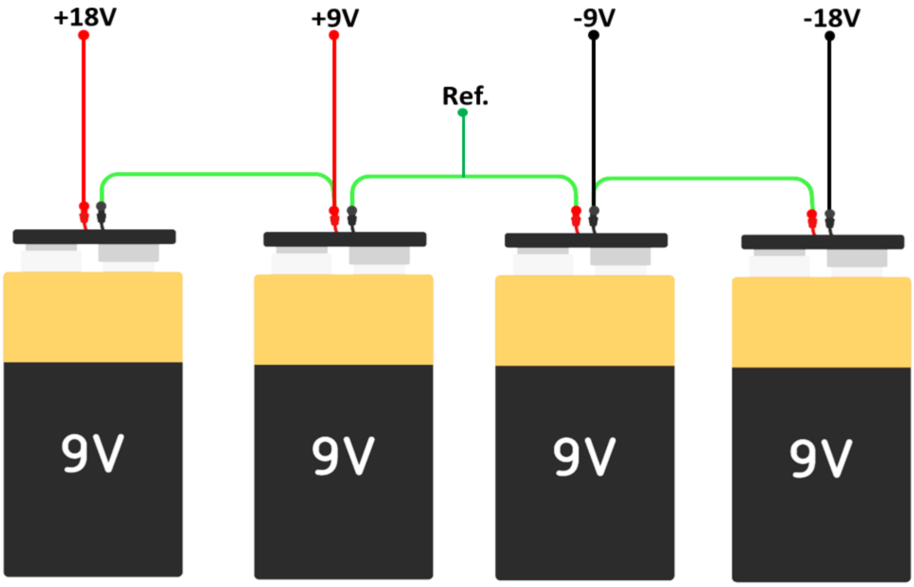


Figure 9 - Connection of 9 V batteries

CHAPTER 4

CONSTRUCTION OF THE POWER SHIELD AND RESISTOR BOARD

After understanding how the circuit works, we are able to start building the Open Stim stimulator. The first step is to build the power shield. For that, it is necessary to access the file on the GitHub PCB. Inside the PCB folder, look for the Eagle folder, and then Shield. The files are in that folder and can be changed if necessary. If you prefer not to make any changes, inside the PCB folder there is the print folder, with PDF file ready to print to use for PCB manufacturing.

4.1 – SHIELD

Figure 10 shows the PCB schema file to be used for making the shield PCB. Print the shield scheme on a laser printer and photo paper and follow the steps below.

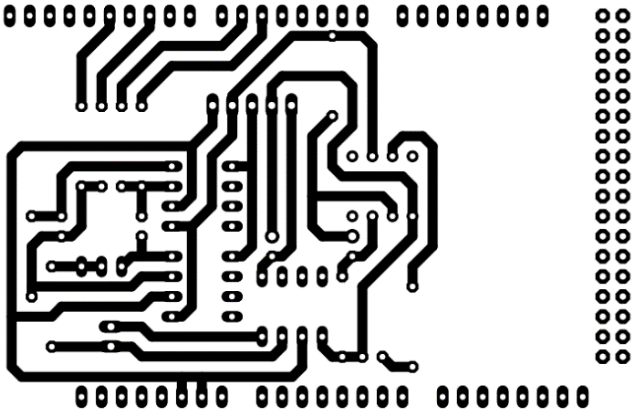


Figure 10 – PCB Shield

1) Cut the sheet of phenolite or copper-plated fiberglass (we recommend fiberglass due to frequency interference) to the size of the shield, but leaving 3 mm more on each side, so that the tracks are not too close to the end the board.

2) After cutting, and with the printing in hand, place the printed side facing the copper plate, as shown in Figure 11.

Figure 11 - Centered drawing on copper plate

3) Center the design on the plate and secure it with tape so that it does not move. With an cloth iron, and without using steam, we will transfer the design to the plate. Place the iron at an average temperature, around 120º C, also place a cloth that can be heated, such as mesh, for example, on the photographic paper, so that it does not come in direct contact with the iron. With the iron heated, wipe over the cloth, heating the plate. Pass it in slow movements, applying for 10 seconds and remove for 5 seconds. Repeat this step for 4 or 5 times, until you notice that the drawing has been transferred to the plate, as in the example in figure 12.

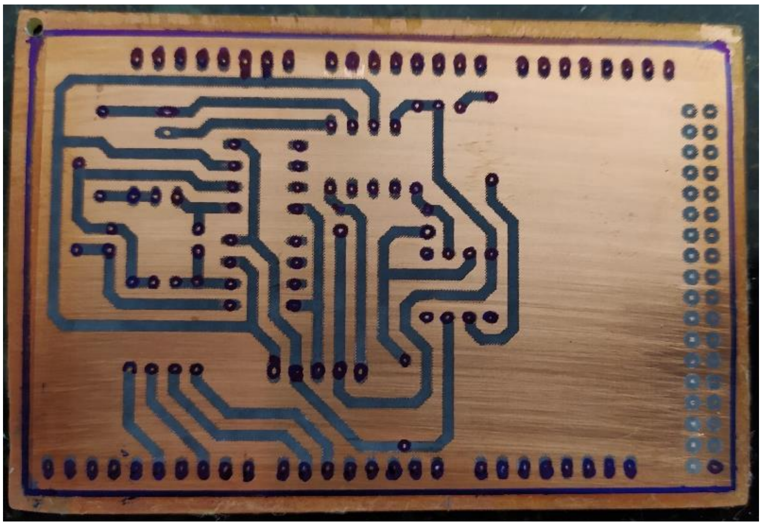


Figure 12 - Transfer of the printed design to the copper plate

4) After verifying the transfer of the file to the board, note if there was no failure, that is, a broken track. If this happens, complete the trail with an overhead projector pen. After that, with the iron perchloride in hand, immerse the plate in the solution and wait until the corrosion process is finished, as shown in Figure 13. The process should take place in 10 minutes if the iron perchloride is in good condition. Bearing in mind that, iron perchloride should be used in a plastic container, never in metal.

Figure 13 - Corrosion process in iron perchloride

5) The expected result should be similar to that of Figure 14, which shows the shield tracks. If there is a problem and the board shows faults in the tracks, it is recommended to do another one, and not to make any kind of patch or jumper.

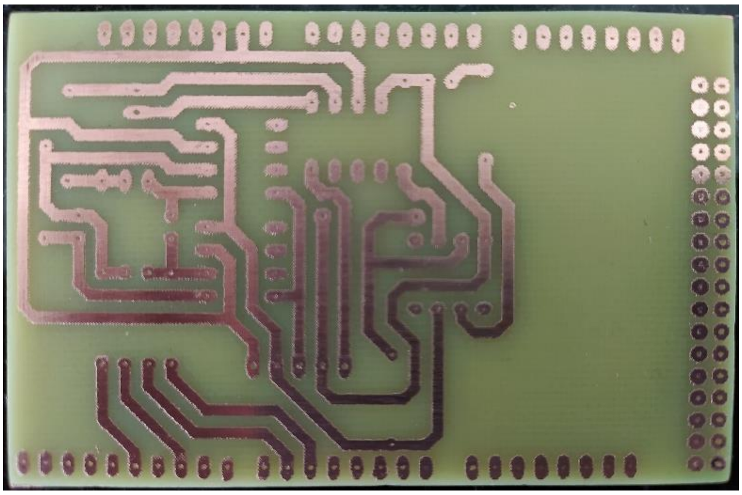


Figure 14 - Shield tracks after corrosion

6) The next step in the process is to drill the plate so that it can receive the components. Use an electronic plate punch or drill with a 1 mm bit. At this stage, it may be necessary to count on the help of one more person or some type of support, in order to avoid wrong holes. To weld the components in the correct places, use the Eagle schematic.

7) After making all the necessary holes, it is time to weld the components. In this step, a 30 W soldering iron and a flat-cut plier is recommended for a better finish. If you prefer, also use solder flux, in order to avoid oxidation at the time of welding. After soldering all components, clean the plate with isopropyl alcohol with the help of an antistatic brush. The expected result is similar to that in Figure 15.

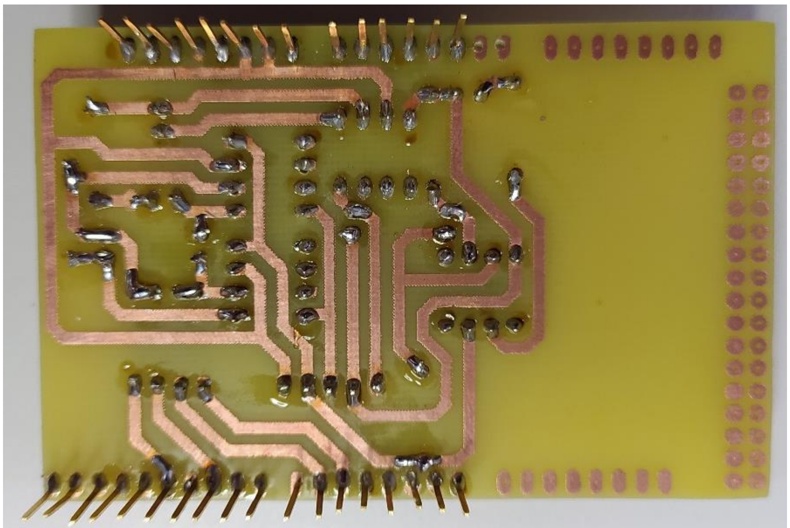


Figure 15 - Soldering all components and cleaning the plate

8) After the shield is ready, check visually first, looking for flaws in the weld and poorly fixed components. If you prefer, you can use a multimeter on the continuity scale and check that everything is connected according to the project. Then, just fit the shield on the Arduino Leonardo, checking the connection of all pins. The expected result should be similar to that in Figure 16 and 17.

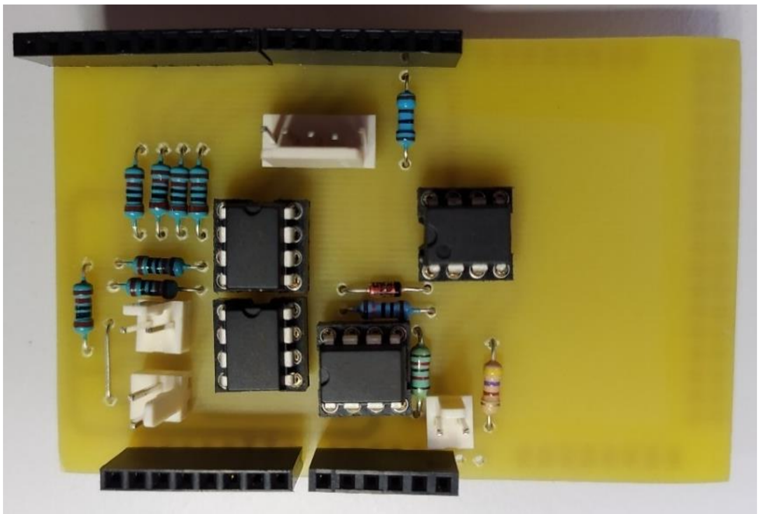


Figure 16 - Shield finished

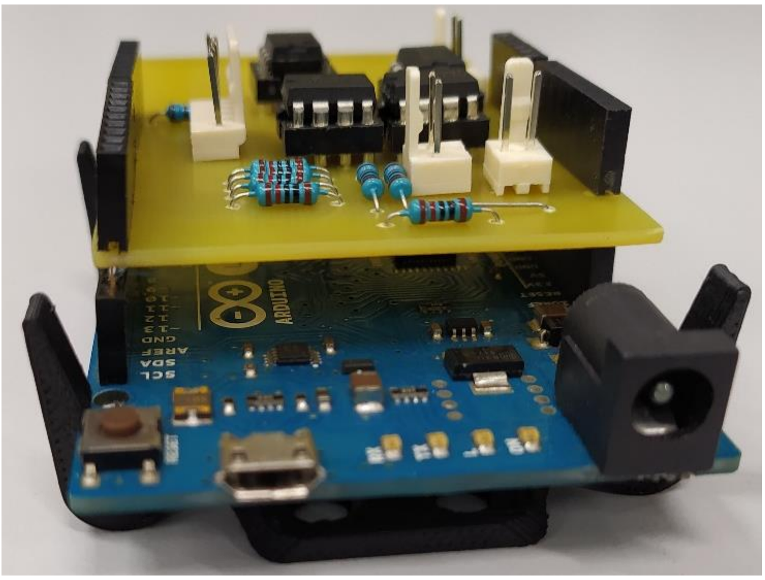


Figure 17 - Arduino Leonardo and Shield

4.2 - RESISTOR BOARD

To build the resistor board, follow the same steps for building the shield. Figure 17 shows the scheme to be printed on a photo paper laser printer.

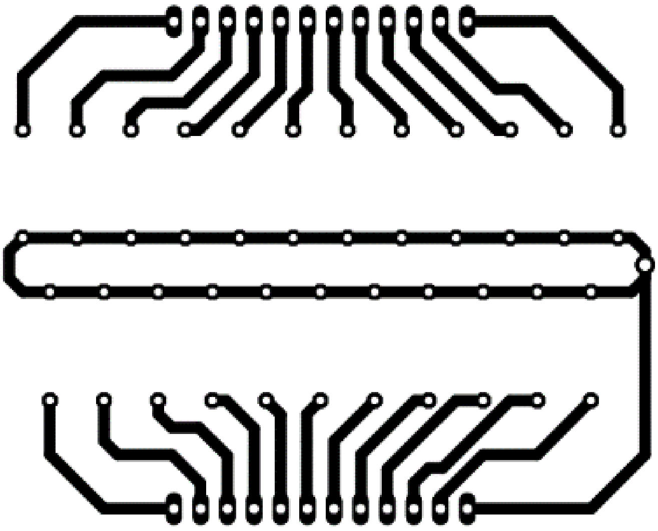


Figure 17 – PCB resistor board

Following the same order as in the previous steps, Figures 18 and 19 show the expected results after corroding the resistor plate.

Remember to clean the plate after welding is finished. KK 12-way connectors were used to facilitate maintenance if necessary. Thus, it is possible to disconnect the resistor board with ease, preserving the board and preventing it from being removed unnecessarily.

Figure 20 shows the resistor board ready, with only the connection to the wave switch missing, which will be made later. Table 1 shows the values of the resistances in the sequence with their respective expected currents. All resistors have 1% tolerance and the male KK connector is soldered directly on the resistor board, while the female is connected by cables that connect to the wave switch.

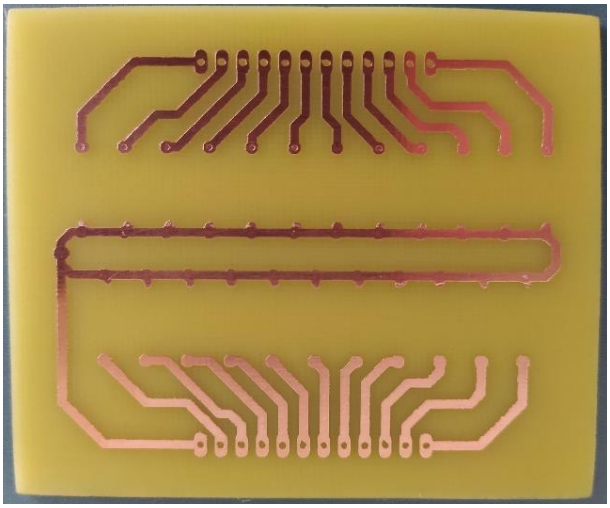


Figure 18 - Resistor plate tracks after corrosion in iron perchloride

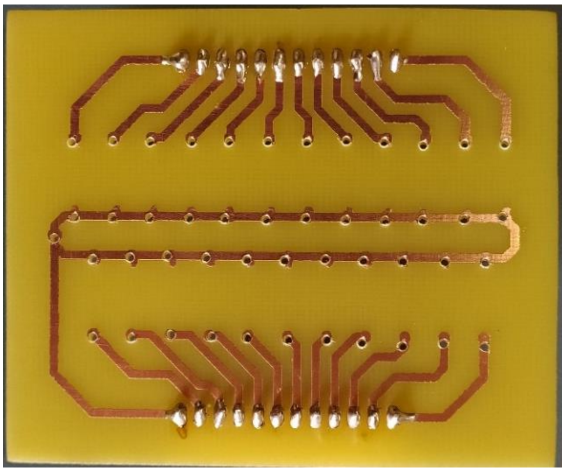


Figure 19 - Component welding

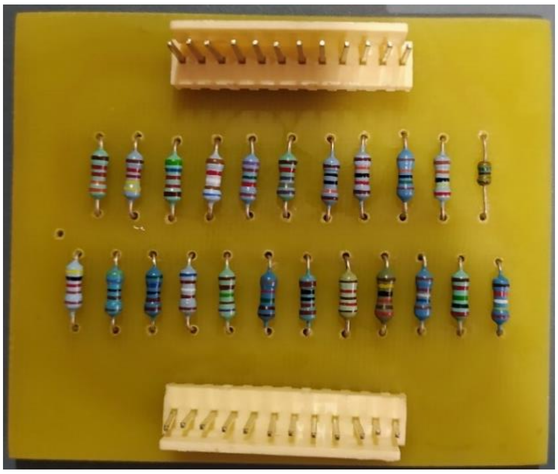


Figure 20 - Resistor board ready to receive the wave switches

Table 1 - Resistance X Currents

RESISTANCE	CURRENT	RESISTANCE	CURRENT
4,02 KΩ	1200 µA	**33,2 KΩ**	150 µA
4,7 KΩ	1000 µA	**44,2 KΩ**	100 µA
5,6 KΩ	900 µA	**56,2 KΩ**	90 µA
6,19 KΩ	800 µA	**61,9 KΩ**	80 µA
7,15 KΩ	700 µA	**71,5 KΩ**	70 µA
8,25 KΩ	600 µA	**78,7 KΩ**	60 µA
10 KΩ	500 µA	**100 KΩ**	50 µA
12,1 KΩ	400 µA	**120 KΩ**	40 µA
14 KΩ	350 µA	**150 KΩ**	30 µA
16,9 KΩ	300 µA	**240 KΩ**	20 µA
20,5 KΩ	250 µA	**360 KΩ**	15 µA
24,9 KΩ	200 µA		

4.3 - ROTARY SWITCHES OR WAVE SWITCHES

Rotary switches are responsible for choosing currents. As we already have the resistor board ready, we will connect the switches to the board. For this, it is necessary to use flexible cables. For this project, we recommend the AWG 26 cable. Each rotary switch, Figure 21, has 12 connection pins and a central pin which is the common pin.

Figure 21 - 12 position rotary switch

To connect all resistors to the rotary switches, follow the steps below:

1) First solder the two 12-way male KK connectors at their locations, as shown in Figure 22.

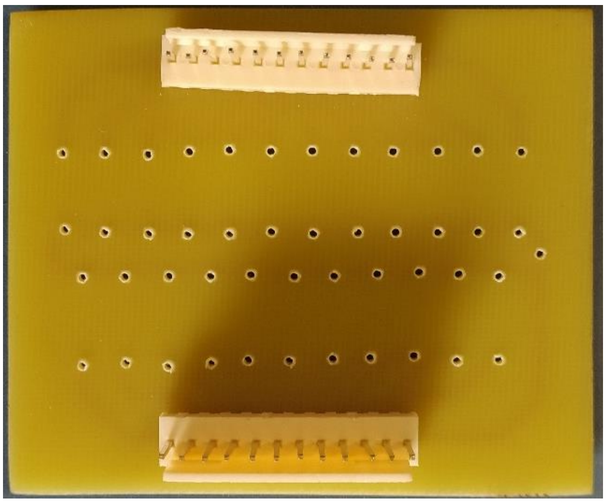


Figure 22 - Welding of 12-way male KK connectors

2) Looking at the resistor plate, locate the last resistor, that is, the 360 KΩ resistor, it will be the first, because it generates the lowest current. With a 25 cm cable in hand, solder the 12-way female KK connector to the first position that will be connected to the male KK connector for this 360 KΩ resistor. At the other end of the cable, solder the first position of the rotary switch.

3) Do this until the 11th current, soldering the second resistor which is the 240KΩ on the second pin of the rotary switch, and so on until the 11th resistor, which is 33,2KΩ. No resistor will be welded to the 12th pin, once it will be used for the connection between the two switches. To make this connection, weld the 12th pin of rotary switch 1 to the common pin of rotary switch 2. In this way, the last position of switch 1, enables switch 2.

4) Now with switch 2, continue soldering the resistors from the 12th to the 23rd, that is, 12 resistors ranging from 24.9KΩ to 4.02KΩ, as shown in Figure 23.

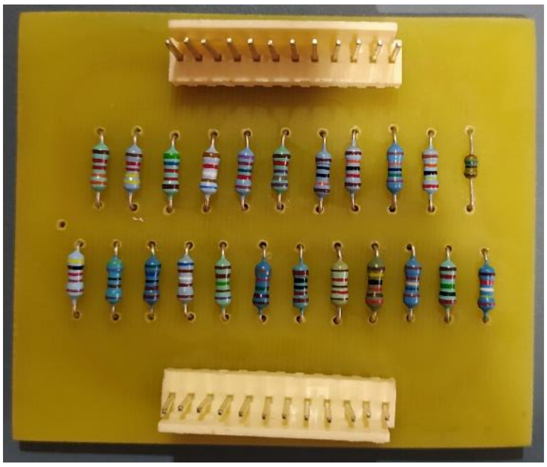


Figure 23 - Resistor board

5) Figure 24 shows the result after welding on the rotary switch and Figure 25 shows what the 12-way female KK connector will look like.

Figure 24 - 12 position rotary switch with connections made

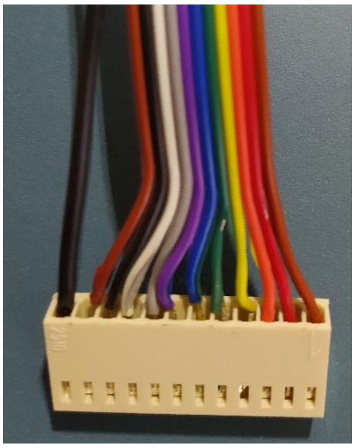


Figure 25 - Female Kk connector with connections made

NOTE: To use the KK connector, you need the Molex 5051 Type KK connector terminal, which is shown in Figure 26. The Molex goes inside the female KK connector, which when connected to the male, completing the connection of the circuit.

Figure 26 - Molex 5051 Type KK Connector Terminal

6) The resistor board has a hole that still does not have a connection, as shown in Figure 27. This hole is the common one on the resistor board and will be connected directly to the shield. The resistor board corresponds to resistor R10 in the diagram in Figure 7, so we must connect it in place. For that, we have to connect two points to the shield, one of them is exactly that common point of the resistor board, as already said, and the other is the common pin of the rotary switch 1. Figure 28 represents this connection in a simplified way. Here we use less resistors, but the purpose is to show how the connection between switch 1 and switch 2 is made and also how the output that will go on the shield is, where the resistor R10 would be.

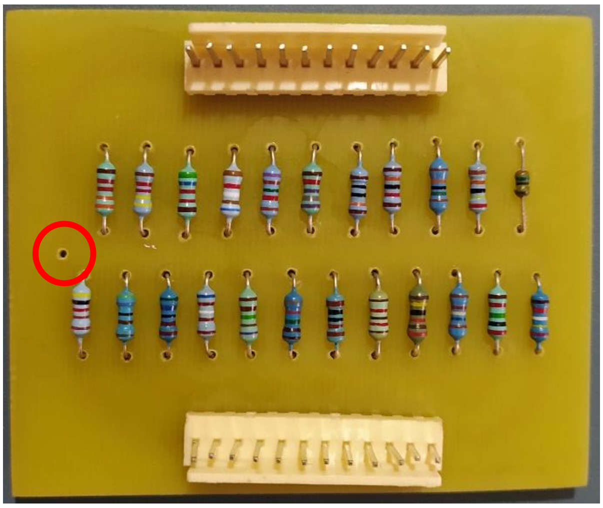


Figure 27 - Common connection point: Connected to output

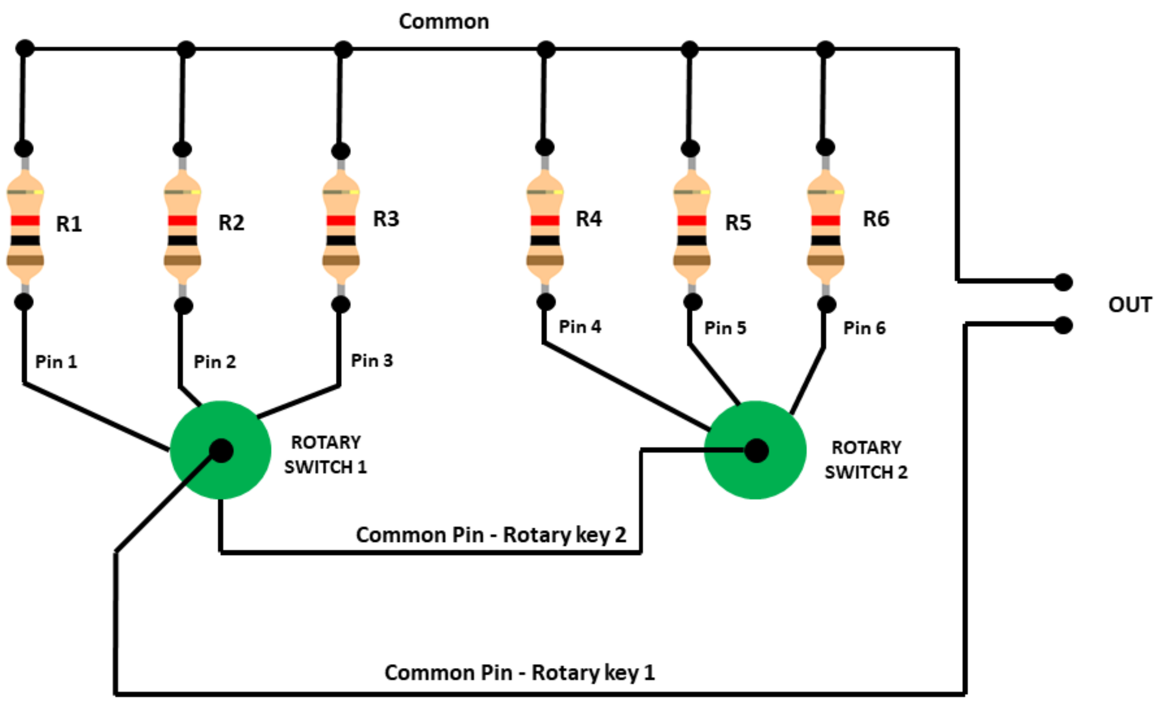


Figure 28 - Connection of rotary keys: interconnection between keys

To connect the resistor board to the shield, the 2-way female KK terminal must be used. The process is the same as previously done using the molex connectors, however, now with only two ways, which is the output of Figure 28.

CHAPTER 5

3D PRINTING OF THE OPEN STIM BOX

To print the box that will accommodate all circuits, go to the 3D printing folder in the GitHub directory and locate the .STL files, if you want to update the drawing, include some text or modify what you think is necessary. There are two ways for this: using the SketchUp software version 2020 or using SolidWorks.

There are two separate pieces for printing: the first is the box and the second is the lid. The box has an average printing time of 17 hours while the lid has an average time of 5 hours. The PLA used is 175 mm in white and black, but this is the user's choice. The brands of our lab and institution (LINNce/UFSJ) are not in the 3D printing files and have been inserted in this documentation only as a means of advertisement and credit. You do not need to include them in your build or derivations.

5.1 – BOX

Figure 29 shows the box seen from the skechtUp software, here it is possible to edit the drawing and generate the .STL files again. It is important to remember that for this, SketchUp needs an extension that makes it possible to export the files in the .STL format. Inside the GitHub directory, in the 3D printing folder, look for the SketchUp folder and inside that folder look for the Extension for STL - SketchUp folder, and install normally using the software.

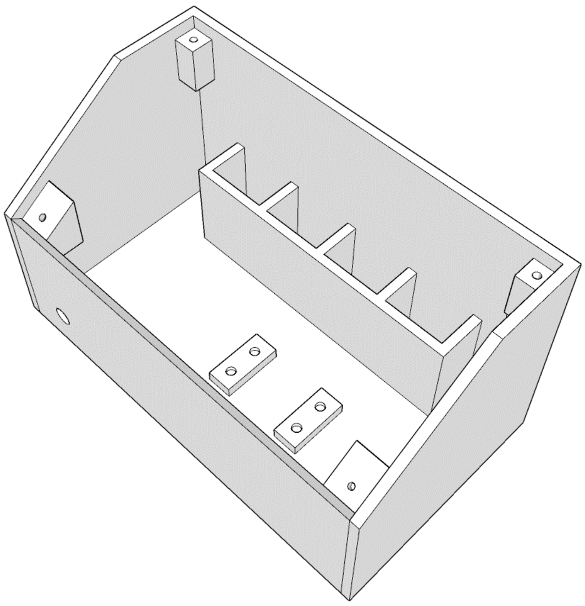


Figure 29 - Open Stim Box

5.2 - BOX COVER

Figure 30 shows the cover of the box, it is there that the Nextion screen will be accommodated, as well as the low battery indication led and the two rotary keys, in addition to the screws that close the cover on the box. It is up to you to put some text to be printed. There is no such text in the files provided.

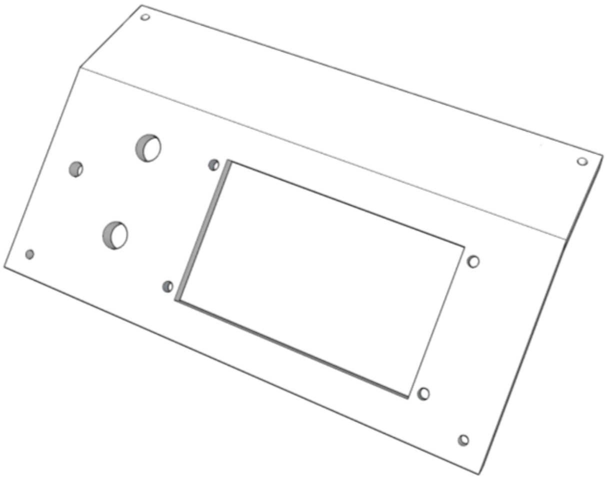


Figure 30 – Cover Box

5.3 - FINAL PRINTED VERSION

After editing files, configuring the 3D printer and with the .STL files ready, just wait for the parts to be printed. Figure 31 shows a printed model of the Open Stim box. Figure 32 shows the printed version of the cover, with text on the top of the cover, identifying the university and laboratory responsible for printing.

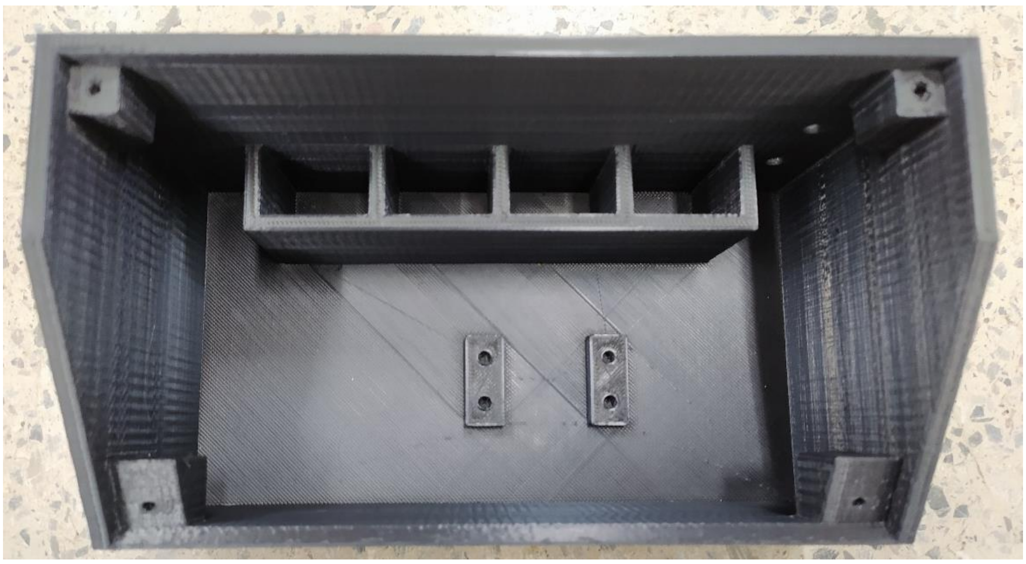


Figure 31 - Print version of the Open Stim box - UFSJ / LINNce

Figure 31 - Printed version of the Open Stim cover - UFSJ / LINNce

CHAPTER 6

CODE: DISPLAY NEXTION AND ARDUINO

All the codes for the Arduino Leonardo, as well as the Nextion display are in the GitHub directory. In the root folder, look for the Nextion code to access the code for the Nextion screen model NX4024T032 of 3.2 ". Look for the Arduino coded folder, to access the files needed to compile and load the code on the Leonardo arduino.

6.1 - NEXTION DISPLAY

The Nextion display is a touch screen that functions as an HMI (man-machine interface). It is there that the values of frequency, pulse width and selection of positive and negative pulses will be made. The first step is to download the Nextion Editor software through the link: https://nextion.tech/nextion-editor/ .

After installation, in the given directory, open the file Nextion_Open_Stim. It is possible to edit the content, to get to know the names of the variables, change images, change values, positions, etc. In short, it is possible to modify the entire layout if deemed necessary. However, when there is any change, it is necessary to check the address of the variable, so that the necessary adjustments are made in the Arduino code. Figure 32 shows the first screen to be displayed, showing the information of the authors who developed the equipment and the university. This screen will appear for a few seconds and then advance to the main menu automatically. This second screen is shown in Figure 33.

Figure 32 - First Nextion display screen

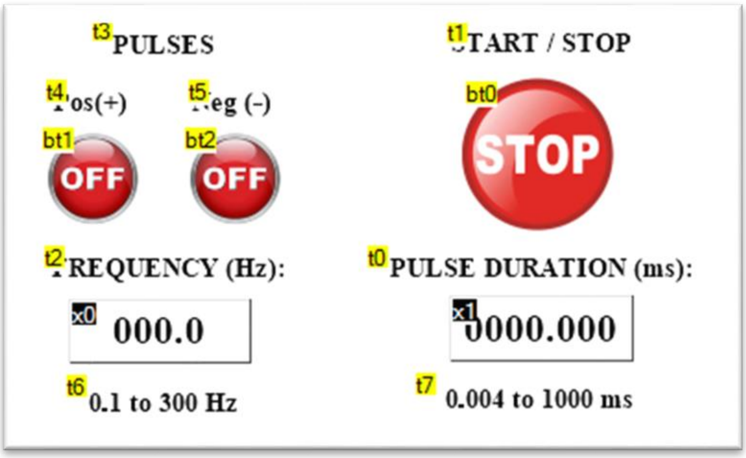


Figure 33 - Nextion main display menu

The Nextion screen has its own programming, so we suggest that before any modification, consult the manufacturer's guide to clarify doubts and get to know the display. This can be done using the link: https://nextion.tech/editor_guide/. It is also possible to access all available functions at: https://nextion.tech/instruction-set/ .

To decrease the brightness of the screen and increase the equipment's autonomy, the DIMS function was used. With it, it is possible to define the brightness of the screen in percentual term. For this project it was set to 40%, therefore, the function is as follows: DIMS = 40. With this value, the current is approximately 36 mA. With this value, the current is approximately 40 mA. The function must be entered in the Preinitialize Event field on the second screen, that is, on the main screen, as shown in Figure 34.

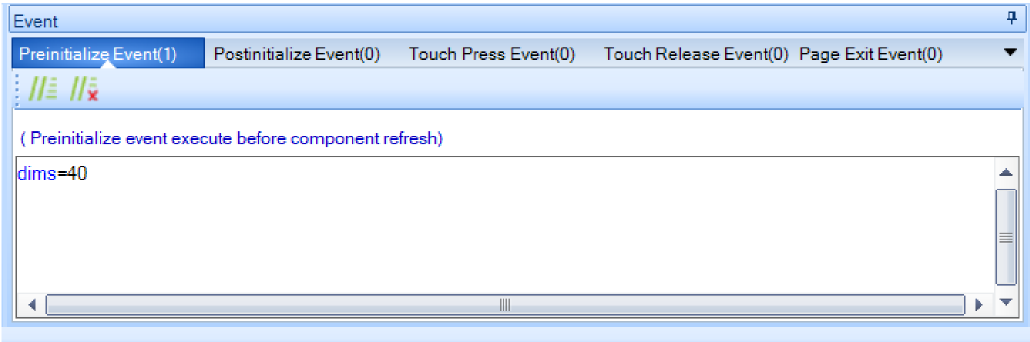
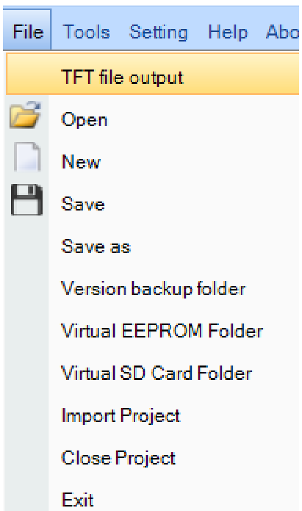


Figure 34 - Inserting DIMS function to control the brightness of the screen

After making all or any modifications you deem necessary, you must send the file to the Nextion screen. A memory card of up to 32 GB formatted in FAT32 is required for this. To generate the file that will be saved on the memory card, go to the File menu and click on TFT file output, as shown in Figure 35. Then, choose the location of the memory card to save the TFT file.

Note: It is necessary to save the file outside any folders and at the root of the memory card.

With the file saved on the microSD card, just insert it into the slot on the Nextion screen, shown in Figure 36. After the card is already inserted, supply the screen with 5 VDC, as shown in Figure 37, as soon as the power is turned on, file upload will start, and it will be saved in Nextion non-volatile memory. When the upload is finished, just turn off the power and remove the microSD card and the code and layout are saved. If you need to modify something in the Nextion Editor, repeat the process described above and upload the TTL file again, so the previous one will be replaced by the new one.



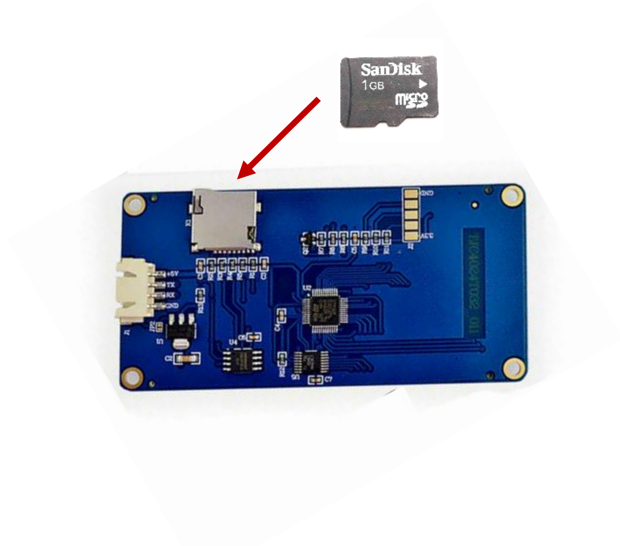


Figure 35 - Exporting TFT file

Figure 36 - MicroSD memory card slot

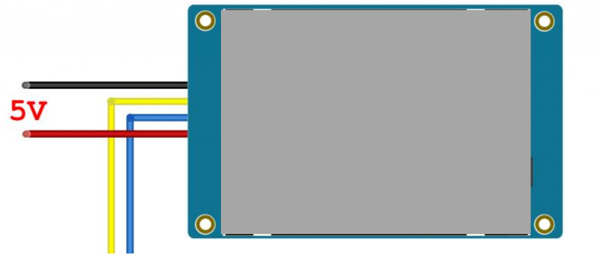


Figure 37 - Power 5VDC display Nextion

There is another upload option, which is using the USB to TTL adapter. The connection to be made at the adapter terminals is shown in Figure 38. The transfer is made directly from the Nextion Editor to the memory of the Nextion display, eliminating the use of the microSD card.

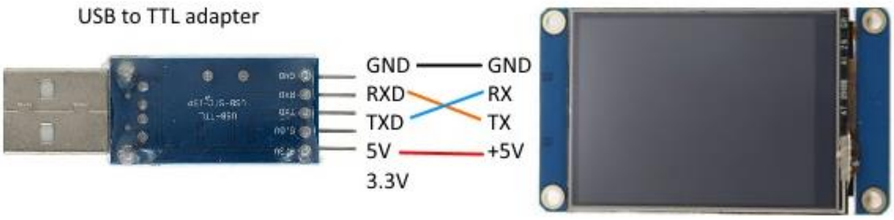


Figure 38 - Connection on Nextion display with USB to TTL adapter

During the sending of the file, either by MicroSD or by the TTL Adapter, observe the percentage of sending and then the success message in the upload, as shown in Figure 39, to be sure that the file was sent without errors.

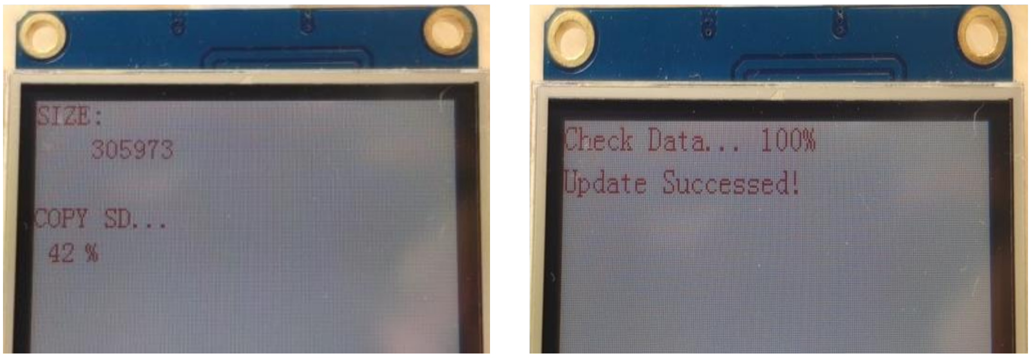


Figure 39 - Upload display Nextion

Next, a summary of the main screen of the Nextion Editor software will be made in Figure 40, to assist in understanding along with the documentation informed in the links above. It is essential that the user studies and practices before making changes to the project. Make all tests, like turning on and off a led, controlling a potentiometer writing a message on the screen, etc.

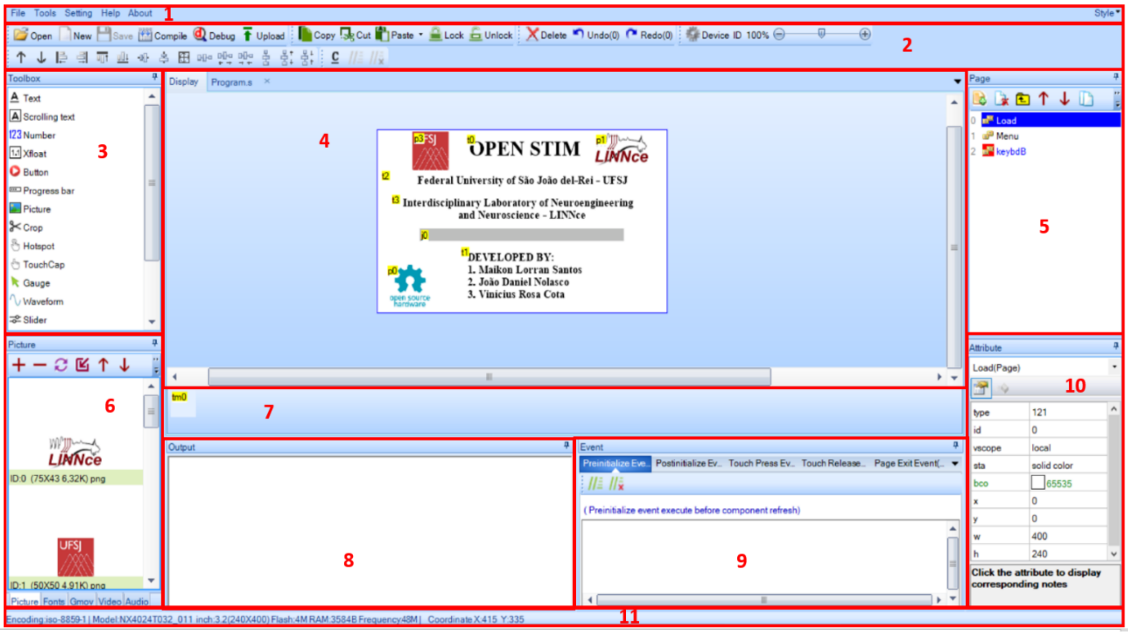


Figure 40 - Main screen Nextion Editor

1) Main Menu

2) Toolbars

3) Toolbox Pane

4) Design Canvas

5) Page Pane

6) Picture Resource Pane

7) Non Visual Components

8) Output

9) User Event Code

10) Attributes Pane

11) Status Bar

6.2 - ARDUINO LEONARDO

First step is to download the Arduino IDE software through the link: https://www.arduino.cc/en/software. Install the program, but before opening for the first time, it is necessary to include the Nextion library in the Arduino Library. To do this, follow the steps below:

1) Inside the GitHub directory, look for the folder Arduino Code, and inside that folder you will have the folder: ITEADLIB_Arduino_Nextion-master. Download it.

2) Copy the folder ITEADLIB_Arduino_Nextion-master and paste it in the location: C: \ Program Files (x86) \ Arduino \ libraries, that is, in the installation location of the Arduino IDE version 32 bits, library. Figure 41 shows what the library folder looks like after the above process.

3) In Figure 42 we have the library already included. To check, open the Arduino IDE, click on the Sketch menu and then Add Library. In the window that opens, check if it appears at the end, after all the other standard Arduino libraries.

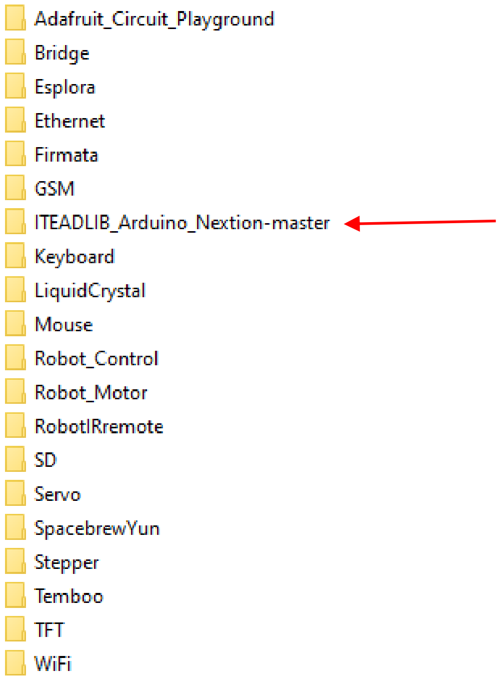


Figure 41 - Arduino library folder after adding the Nextion display library

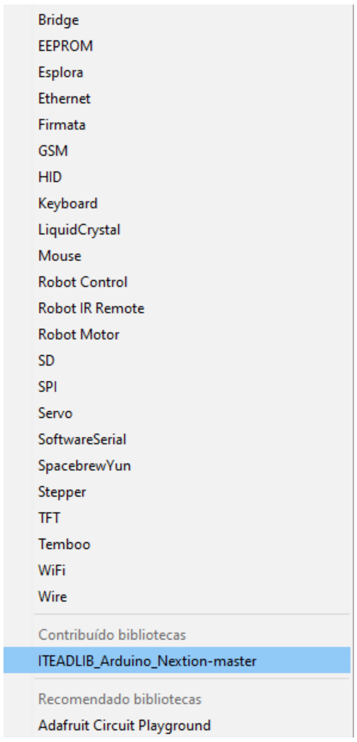


Figure 42 - Verification of inclusion of the Nextion library by the Arduino IDE

It is now possible to compile the Arduino Leonardo code. To open the code, access the directory on GitHub and open the folder Arduino Code. Inside it, open the folder Arduino_Open_Stim and inside it you will find the file that must be uploaded to the Arduino Leonardo.

The Arduino Leonardo documentation was made available in chapter 2 of this Manual. The ports used for communication with the Display Nextion are 0 and 1 (RX and TX, respectively). All connections will be explained and detailed in the next chapter.

Now it is important to have the Nextion screen ready and programmed, and with the Arduino also loaded with the available code.

Remembering again that, in case of modification in any part of the code, whether in Arduino or Nextion, it is necessary to check in both codes if the changes are in agreement, if what has been changed in one will not affect the other and if it does, the necessary modifications are already made for both codes / programming. Regarding the code, all explanations are given directly in the code, in the form of comments.

CHAPTER 7

ASSEMBLING THE OPEN STIM

Before starting the actual assembly, it is recommended to make a Check List of materials and tools:

7.1 - CHECK LIST MATERIALS AND TOOLS

- Flush cutting pliers
- Nose pliers
- Welding suction
- Soldering Iron
- Tin / lead solder
- Multimeter
- Clean contact or Isopropyl Alcohol
- Drill with set of drills
- Stylus
- Screwdriver 1/8 x 3"
- Philips key 1/8 x 3"
- Brush tool
- Double-sided tape

These are the basic tools. However, the user can feel free to use other tools, from the simplest to the most sophisticated. In this final assembly part, two people are recommended, one as the main technician and the other as an assistant.

7.2 - MOUNTING THE BATTERIES IN THE HOUSING

Go back to Figure 9 and connect the 4 batteries according to the diagram shown. To do this, use item 8 of the material list (the material list is also in the GitHub directory, in the root folder), which is the 9 V battery clip. After the connection between the terminals has been made, you can place the batteries in the box in their respective places, as shown in Figure 43.

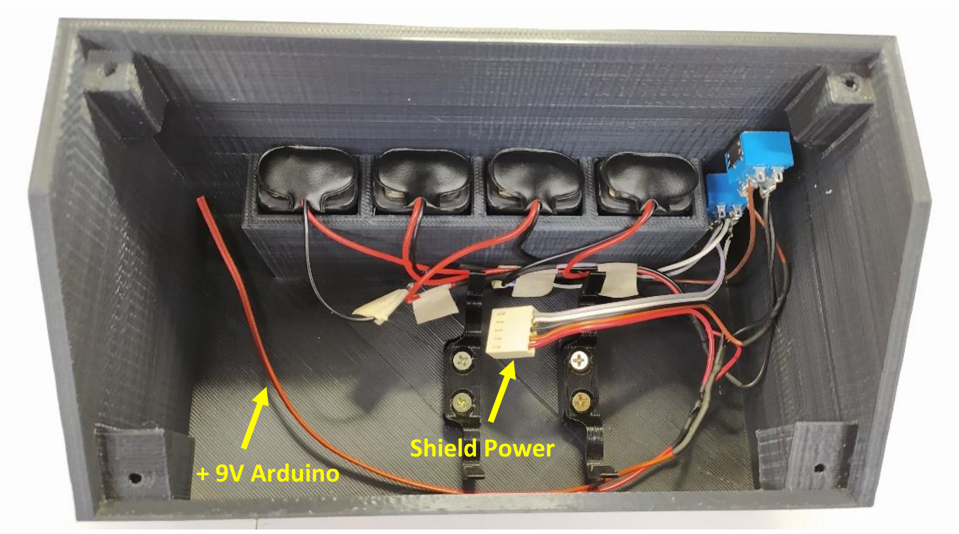


Figure 43 - Connection of batteries and lever switches

Now it is necessary to exit the lever switches and supply the shield, for this, we will use the 5-way female KK connector, since the male is already soldered on the shield. As previously done, it is necessary to use the Molex terminal, item 20 of the bill of materials. The sequence from left to right is:

1) Reference
2) -9 V
3) +9 V
4) -18 V
5) +18 V

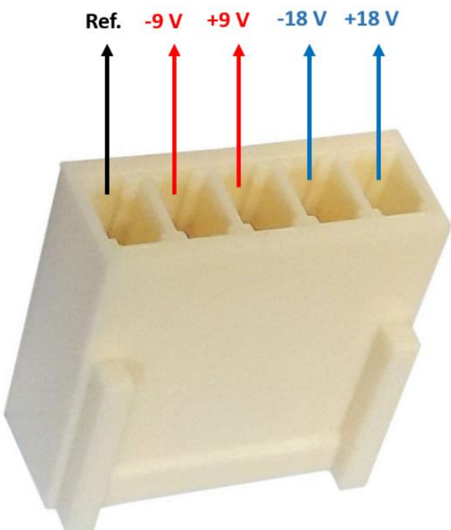


Figure 44 - 5-way female KK connector for shield power

Still on Figure 43, we can also fix the Arduino Leonardo support, using the flat head screw 2.2 x 6.5 mm, which is item 55 in the list of materials, while the support is item 54. To fix, use the philips wrench and a medium torque, to avoid breaking the support that is made of PVC. And finally, take + 9 V in parallel to power the Arduino with the male P4 plug, item 19 of the list of materials. The yellow arrows in Figure 43, indicate the points mentioned above in case of doubt.

Now, attach the shield to the Arduino, as in Figure 45 and attach it to the bracket that was previously fixed. Just press down lightly and the bracket will embrace the Arduino.

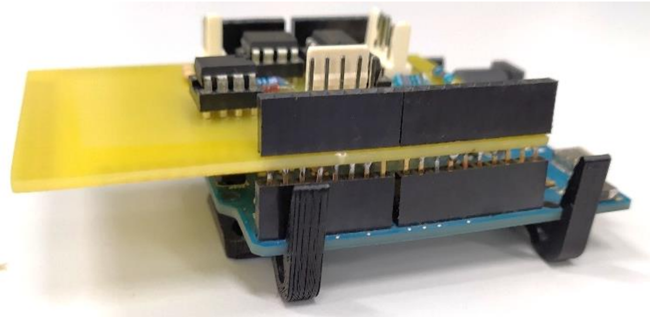


Figure 45 - Arduino + Shield

For now, let's leave the box aside and mount the lid with the Nextion screen. The first step is to fix the Nextion screen to the cover. For that, use the 4 screws with nut, item 57 of the bill of materials. Check if it is well fixed, but without overtightening the screws, avoiding damage to the display and also to the cover. Figure 46 shows how it should look.

Figure 46 - Nextion display attached to the housing cover

Next, fix the two rotary keys, both are accompanied by their nuts, so pass the key from the inside out, insert the nut and tighten it until the key does not move when used. If you prefer, use 2.5 x 100 mm nylon cable ties to better finish the cables that go to the resistor plate, as shown in Figure 47. Do not bend the cables too much, do not make sharp bends and also avoid mechanical stress in the cable close to the welding site.

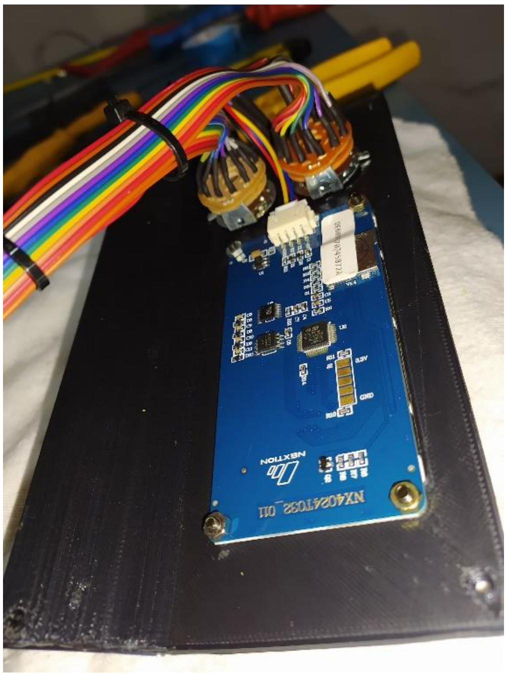


Figure 47 - Attaching the two rotary keys to the housing cover

With the keys in place, place the 3 mm led holder, item 17 on the bill of materials. Then, place the LED inside the holder. Also place the two knobs on the rotary switches, item 9. Figure 48 shows how it looks after these steps.

Figure 48 – Inserting led and knobs

It is also possible to fix the resistor plate. With the box at the front, the board will be on the right side, since the left most is the shielded Arduino. Glue the double-sided tape to the ends of the board, being careful not to stick it on the tracks and fix it on the left side of the box, as shown in Figure 49.

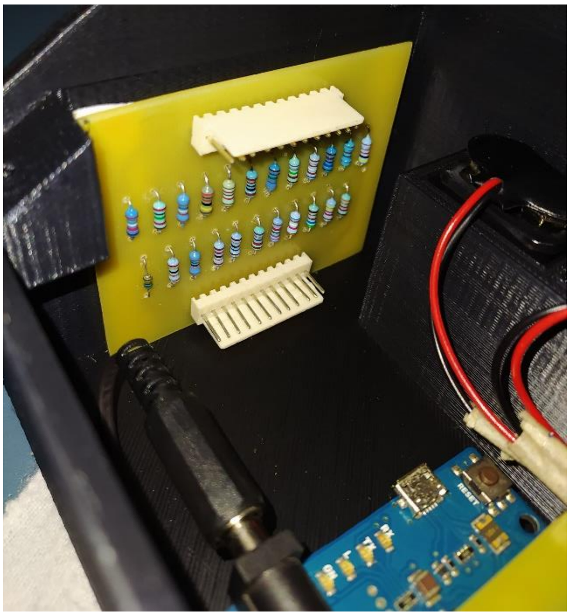


Figure 49 - Fixing the resistor plate to the box

Now solder the ends of the Led with the same AWG 26 cable, and at the other end solder the Molex terminal to be inserted into the 2-way female KK connector, as this will be connected to the shield. Pay attention to the polarity of the led. If necessary, refer to the diagram again. Figure 50 shows where the female KK connectors for all connections should be placed.

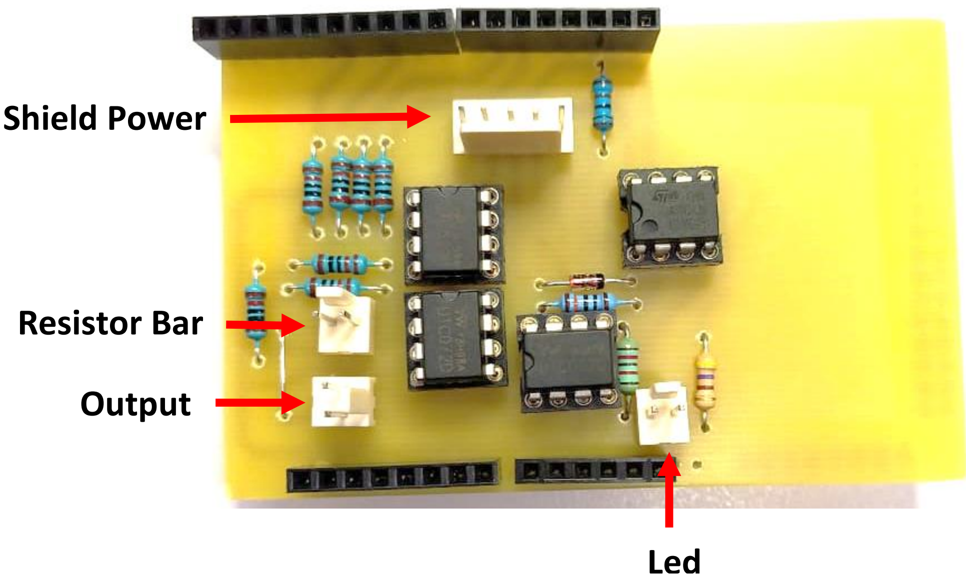


Figure 50 - Shield connections

Remember to connect the current selection rotary switches to the resistor board. As shown in Figure 51.

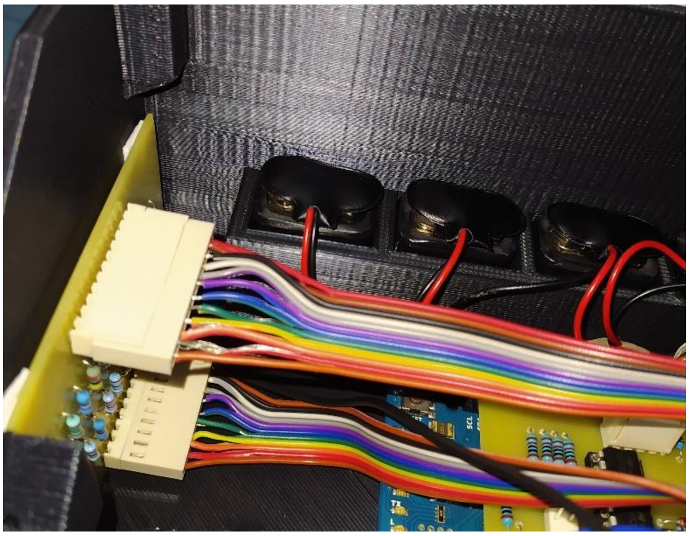


Figure 51 - Connection of rotary switches to the resistor plate

Among the connections to the shield, only the output connection, represented in the diagram by resistor R11, is missing. For hits we will use the female RCA connector, item 5 of the material list, Molex terminal, two-way female KK connector and 20 cm of the AWG 26 cable. This cable must already be placed, being one end into the box for connection in the shield and other end out of the box for connecting the RCA, since the RCA can only be fixed in the box from the outside to the inside. At one end, and with the help of another person or some support, solder the RCA connector, paying attention to the polarity. At the other end, solder the Molex terminal to the two-way female KK connector. After welding, fix the RCA connector to the box, Figure 49 shows what the final result looks like. Finally, connect the output and all other connections to the location shown in Figure 52.

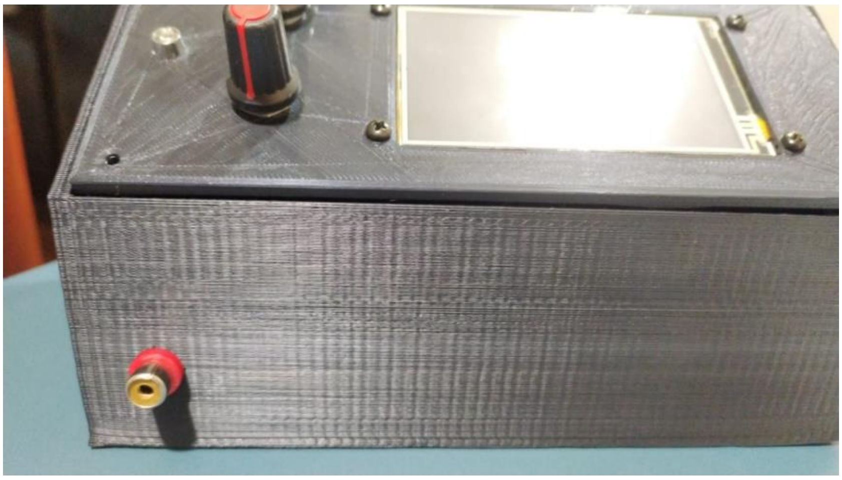


Figure 52 - Output: RCA connector

The last step is to connect the Nextion screen and communicate with the Arduino. It's very simple: supply the Nextion screen with 5 VDC from the Arduino itself. Then connect the TX of the Nextion screen to the RX of the Arduino Leonardo and the RX of the Nextion screen to the TX of the Arduino Leonardo. Logically, this connection is made by the shield, which gives access to the pins of the Arduino Leornado through the female connector bar. Figure 53 shows how this connection is made.

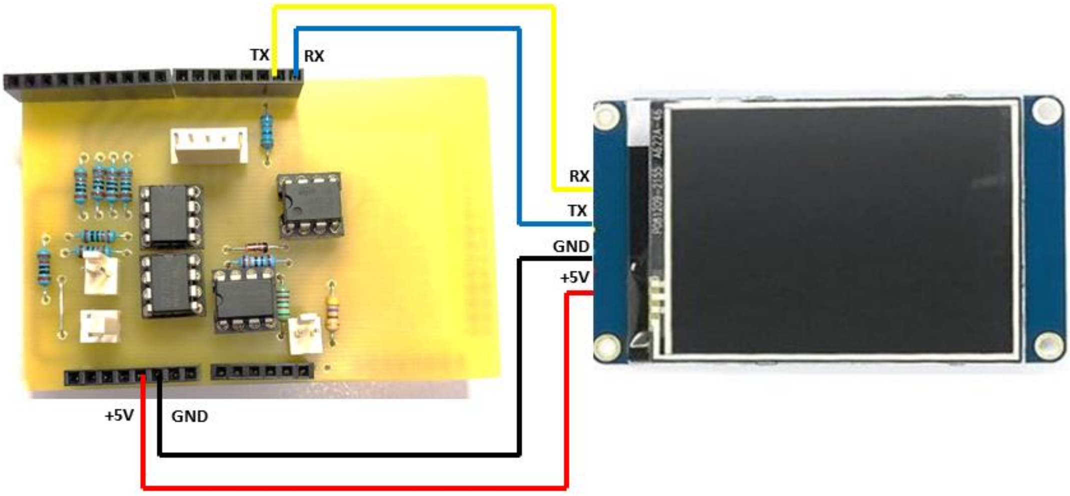


Figure 53 - Connection between Arduino Leonardo and Display Nextion

Turn on the two lever switches to test the equipment, the Arduino Leonardo leds should indicate the energization and the Nextion screen must initialize. If the batteries are under 7.5 V the indication LED will also be on, as shown in Figure 54.

Figure 54 - Open Stim finished

The stimulator is ready, but it is necessary to fully test it. To proceed to the tests, go to the GitBub directory and look for the Tests folder. There you will find a guide to help in the tests and verification of the functioning.

FINAL CONSIDERATIONS

The Laboratory of Neuroengineering and Neuroscience - LINNce of the Federal University of São João del-Rei - UFSJ, hopes that, with this equipment, you can start or continue your research inside or outside a university or technical school. We appreciate your interest in getting to know our stimulator and we count on your contributions to improve the project through GitHub. See you soon and good studies!

LINNce

ACKNOWLEDGMENTS

FAPEMIG

CAPES

UFSJ
Universidade Federal
de São João del-Rei