



**MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL SUL-RIO-GRANDENSE  
CÂMPUS PELOTAS  
CURSO TÉCNICO DE NÍVEL MÉDIO EM ELETRÔNICA**

**Guilherme Zitzke Cardoso da Silva  
Jorge Alberto Gil y Alaniz Silva Júnior  
José Eduardo das Neves da Fonseca**

**Arduino Neptune**

**Pelotas  
2022**

Guilherme Zitzke Cardoso da Silva  
Jorge Alberto Gil y Alaniz Silva Júnior  
José Eduardo das Neves da Fonseca

Relatório do projeto de pesquisa Arduino  
Neptune, submetido à apresentação da 34º  
MOSTRATEC

Orientadoras: Prof. Ulisses Lyra dos Santos  
Prof. Guilherme Schwanke Car-  
doso

Pelotas  
2022

Ao Curso Técnico em Eletrônica do  
câmpus Pelotas - IFSul.

## **AGRADECIMENTOS**

Às nossas famílias pela compreensão.

Aos nossos amigos e colegas, pelo apoio durante as pesquisas.

Ao Professor Ulisses Lyra dos Santos e ao Professor Guilherme Schwanke Cardoso, pela oportunidade acadêmica de realizar este trabalho.

*"O maior bem do homem é uma mente inquieta".*  
**(ISAAC ASIMOV)**

## RESUMO

Um kit didático de baixo custo que se utiliza de um Arduíno Uno/Nano - dependendo da versão - para o estudo de diversos componentes eletrônicos como: Display de Led, display gráfico, circuitos analógicos com controle digital, controle de motor de passo e dois tipos de sensores. Possibilitando a implementação de protótipos e facilitando o aprendizado da plataforma microcontrolada arduino, utilizada no kit, podendo também ser aplicada em diferentes disciplinas estudadas no curso de eletrônica como: eletrônica digital, programação, microcontroladores e qualquer outra disciplina relacionada a projetos que envolvam eletrônica. Nas disciplinas de programação é possível se "abstrair" boa parte do hardware e trabalhar mais a lógica de programação, tendo em vista que é possível programar o kit com linguagem de alto-nível. Em disciplinas de eletrônica digital, por exemplo, pode-se, trabalhar todo o acionamento de um display multiplexado - como o utilizado no projeto - e com simples rotinas demonstrar o funcionamento deste componente, assim como todo o hardware da placa, e por fim qualquer disciplina que tenha como objetivo trabalhar os conhecimentos de eletrônica que envolva sistemas microcontrolados mostrando como se faz o acionamento de cada hardware encontrado na proposta apresentada. No próprio kit é possível separar fisicamente todos os pinos do arduino - ou somente os que forem de interesse - para testar algum dispositivo que não exista no projeto, através da protoboard já integrada, com o objetivo de expansão e testes. Ainda junto da protoboard existem fontes de tensão de: 12v, 5v e 3,3v. A placa poderá ser utilizada pelo professor em sala de aula, buscando estimular a construção de pequenos projetos interdisciplinares no decorrer do curso de eletrônica, devido a esta flexibilidade de utilização estes circuitos integrados possuem. O baixo custo do projeto fica evidente devido à quantidade de hardware que o mesmo dispõe, quando comparado com outros existentes no mercado, além de possuir um layout simples e um diagrama esquemático de fácil entendimento. Como o kit se baseia em componentes de baixo custo, o aluno em sua casa ou laboratório de estudos, poderá desenvolver projetos de eletrônica de seu interesse com maior facilidade. Acredita-se que com esse kit poderá ser melhorado, tanto em qualidade como em inovação, o ensino da eletroeletrônica em sala de aula, proporcionando diferentes formas de pensar os conteúdos e um fácil acesso a todos as novas tecnologias.

**Palavras-chave:** Kit didático, Arduino, Sistemas Microcontrolados.

## **ABSTRACT**

A low cost educational kit that uses an Arduino Uno / Nano - depending on the version - for the study of various electronic components such as: LED display, graphic display, analog circuits with digital control, stepper motor control and two types of sensors. Enabling the implementation of prototypes and facilitating the learning of the Arduino microcontrolled platform used in the kit, it can also be applied in different disciplines studied in the electronics course such as digital electronics, programming, microcontrollers and any other discipline related to projects involving electronics. In the programming disciplines it is possible to "abstract" much of the hardware and work more programming logic, since it is possible to program the kit with high-level language. In digital electronics disciplines you can, for example, learn how to drive a multiplexed display - as used in the project - and with simple routines demonstrate the operation of such component, as well as all the other hardware on the board. In the kit, it is possible to physically separate all the Arduino pins - or only those that are of interest - to test any other module or IC that does not exist in the kit, anything can be integrated on the protoboard, for the purpose of expansion. Also near the protoboard are voltage sources of: 12v, 5v and 3.3v. The board is meant to be used by the teacher in the classroom, seeking to stimulate the construction of small interdisciplinary projects during the course of electronics, due to the flexibility of the circuits on the kit. The low cost of the project is evident due to the amount of passive components it has when compared to others on the market, as well as its simple layout and easy-to-understand schematic diagram. Because the kit is based on low-cost components, the student in his home or the lab can more easily develop electronics projects of interest. It is believed that this kit can improve both quality and innovation in the teaching of electronics in the classroom, providing different ways of thinking about content and easy access to all new technologies.

**Keywords:** Didactic kit, Arduino, Microcontrolled systems.

## **LISTA DE ILUSTRAÇÕES**

Figura 1	Conexão do display 7 Segmentos de 4 dígitos no Arduino. . . . .	13
Figura 2	Circuito do display 7 Segmentos de 4 dígitos na protoboard. . . . .	13
Figura 3	Conexão do display gráfico 128X32. . . . .	14
Figura 4	Circuito do display gráfico 128X32. . . . .	14
Figura 5	Conexão do driver para o motor de passo. . . . .	15
Figura 6	Circuito do driver para conexão do motor de passo. . . . .	15
Figura 7	Conexão dos três botões na entrada analógica. . . . .	16
Figura 8	Circuito dos três botões na entrada analógica. . . . .	16
Figura 9	Conexão do Led RGB. . . . .	17
Figura 10	Circuito do Led RGB. . . . .	17
Figura 11	Conexão do sensor óptico. . . . .	18
Figura 12	Circuito do sensor óptico. . . . .	18
Figura 13	Circuito do sensor infra-vermelho. . . . .	19
Figura 14	Conexão do sequencial de Leds. . . . .	19
Figura 15	Circuito do sensor infra-vermelho. . . . .	20
Figura 16	Shift Register 74HC595. . . . .	20
Figura 17	Sinas do Shift Register 74HC595. . . . .	21
Figura 18	Placa de fenolite cobreada. . . . .	22
Figura 19	Relação da corrente pela largura da trilha. . . . .	22
Figura 20	Placa PCB para o Arduino Neptune UNO. . . . .	23
Figura 21	Render 3D Versão Arduino Neptune NANO - Fusion 360. . . . .	23
Figura 22	Código display 4 dígitos 7 segmentos . . . . .	27
Figura 23	Código display OLED . . . . .	28
Figura 24	Código motor de passo - 1 . . . . .	29
Figura 25	Código motor de passo - 2 . . . . .	30
Figura 26	Layout Arduino Neptune UNO . . . . .	31
Figura 27	Esquemático Arduino Neptune V1.0 . . . . .	32

# SUMÁRIO

<b>1 INTRODUÇÃO . . . . .</b>	9
1.1 OBJETIVOS GERAIS . . . . .	9
1.2 OBJETIVOS ESPECÍFICOS . . . . .	9
1.3 JUSTIFICATIVA . . . . .	9
1.4 PROBLEMA . . . . .	10
1.5 HIPÓTESE . . . . .	10
<b>2 REFERENCIAL TEÓRICO . . . . .</b>	11
<b>3 METODOLOGIA . . . . .</b>	12
3.1 PROTOTIPAGEM . . . . .	12
3.1.1 Display 7 Segmentos de 4 dígitos multiplexado . . . . .	12
3.1.2 Display gráfico 128X32 . . . . .	13
3.1.3 Driver para conexão do motor de passo . . . . .	15
3.1.4 Três botões em entrada analógica . . . . .	16
3.1.5 Led RGB . . . . .	17
3.1.6 Sensor óptico . . . . .	17
3.1.7 Sensor infra-vermelho/controle remoto . . . . .	18
3.1.8 Sequencial de Leds . . . . .	19
3.2 MULTIPLEXAÇÃO . . . . .	20
3.2.1 Shift Register 74HC595 . . . . .	20
3.2.2 Botões divisor de tensão . . . . .	21
3.3 CONFECÇÃO . . . . .	21
3.3.1 Placa de fenolite cobreada . . . . .	21
3.3.2 Placa de circuito impresso . . . . .	22
3.4 INTEGRAÇÃO . . . . .	23
<b>4 CONCLUSÃO . . . . .</b>	24
<b>REFERÊNCIAS . . . . .</b>	25
<b>APÊNDICE A – CÓDIGO DISPLAY 4 DIGITOS 7 SEGMENTOS . . . . .</b>	27
<b>APÊNDICE B – CÓDIGO DISPLAY OLED 128X32 . . . . .</b>	28
<b>APÊNDICE C – CÓDIGO MOTOR DE PASSO . . . . .</b>	29
<b>APÊNDICE D – LAYOUT ARDUINO NEPTUNE UNO . . . . .</b>	31
<b>APÊNDICE E – ESQUEMÁTICO ARDUINO NEPTUNE V1.0 . . . . .</b>	32

# **1 INTRODUÇÃO**

Com o grande aumento da quantidade de circuitos eletrônicos que se baseiam em um microcontrolador para o funcionamento do mesmo, é indispensável o estudo de sistemas digitais programáveis em um curso técnico de eletrônica.

Neste trabalho será mostrado à implementação de um kit didático que teve como objetivo o baixo custo e a facilidade de acesso a replicação de qualquer parte da placa pelo estudante, já que não foram encontrados no mercado projetos com o hardware necessário e que atendessem às demandas citadas.

Para isto, utilizou-se a ferramenta de software "EAGLE"para confeccionar a placa de circuito impresso. Como o kit funciona como um grande shiled, utilizado com o arduino UNO ou NANO dependendo da versão, aproveitou-se o próprio ambiente de programação fornecido pela plataforma arduino para testar cada um dos circuitos colocados na placa.

## **1.1 OBJETIVOS GERAIS**

Construção de um kit didático de baixo custo e fácil acesso, voltado a alunos de diferentes classes sociais, para o ensino de eletroeletrônica.

## **1.2 OBJETIVOS ESPECÍFICOS**

Desenvolver com maior facilidade o estudo de sistemas microcontrolados, no curso técnico de eletrônica, através de um kit didático que proporcione o entendimento de diferentes circuitos eletrônicos, assim como lógica de programação.

## **1.3 JUSTIFICATIVA**

Com o grande aumento da quantidade de circuitos eletrônicos que se baseiam em um microcontrolador para o funcionamento do mesmo, um kit didático que tem como objetivo o baixo custo e a facilidade de acesso para a replicação de integrado à placa pelo estudante, tornando-se indispensável para o estudo de sistemas digitais

programáveis em um curso técnico de eletrônica.

#### **1.4 PROBLEMA**

Encontrar no mercado um produto que atendesse as necessidades evidenciadas em todos os quesitos e ainda pudesse ser aplicado em diferentes disciplinas como: eletrônica digital, programação, microcontroladores e qualquer disciplina relacionada a projetos que envolvam eletrônica. Nesse sentido, muitas escolas técnicas utilizam kits didáticos para a abordagem dos conteúdos, entretanto, na sua grande maioria, são defasados ou tem manutenção/aquisição extremamente caras.

#### **1.5 HIPÓTESE**

Já existe no mercado kits didáticos com características semelhantes, mas pensando na facilidade de montagem, replicação do mesmo e expansão para prototipação de projetos, não foi encontrado nenhuma proposta que atendesse a todas as características solicitadas.

## **2 REFERENCIAL TEÓRICO**

O projeto apresentado teve seu inicio a partir de um estudo feito para a confecção de um kit, que busca possibilitar, tanto ao aluno quanto ao professor, uma ferramenta para melhor desenvolver os conteúdos trabalhados em sala de aula.

Portanto é importante ressaltar que se trata de um projeto didático e que busca abranger os conteúdos abordados em disciplina da área de eletrônica, como: eletrônica digital, programação, microcontroladores e qualquer outra matéria relacionada a projetos que envolvam eletrônica

### **3 METODOLOGIA**

Assim sendo, dividiu-se a metodologia em quatro partes: em um primeiro momento será apresentado como foi realizada o dimensionamento e a escolha dos componentes para a placa, posteriormente, será mostrado como a multiplexação de algumas partes do projeto foram feitas e, por fim, será abordado as etapas de confecção e a integração de todos os circuitos que compõem o kit didático.

#### **3.1 PROTOTIPAGEM**

Buscando melhor organizar as etapas desenvolvidas, dividiu-se a prototipagem os seguintes tópicos: display 7 segmentos de 4 dígitos multiplexado, display gráfico 128X32, driver para conexão de motor de passo, três botões em entrada analógica, Led RGB, sensor óptico, sensor infra-vermelho/controle remoto e, por fim, sequencial de Leds.

##### **3.1.1 Display 7 Segmentos de 4 dígitos multiplexado**

O display de 7 segmentos 4 dígitos utilizado no kit foi escolhido por ser versátil e fácil de ser multiplexado.

Cada segmento do display está conectado junto, e com os pinos D1, D2, D3 e D4, pode-se escolher em qual dos dígitos o número irá aparecer. Assim a multiplexação do display em nível de software fica fácil e intuitiva.

Decidiu-se multiplexar o display com 2 Shift Registers 74HC595, o primeiro com seus 8 bits, servindo como decodificador para o display, e o segundo como MUX para os dígitos. Entretanto, a saída de um está ligada a entrada de dados de outro, logo, com apenas 3 pinos do Arduino, podemos controlar 4 dígitos dos 7 segmentos.

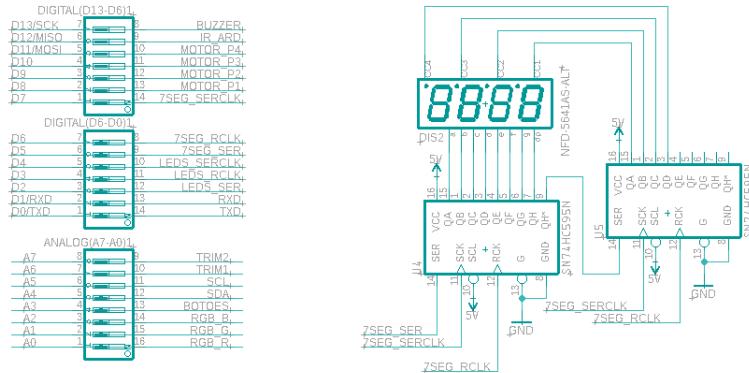


Figura 1: Conexão do display 7 Segmentos de 4 dígitos no Arduino.

Utilizou-se o 74HC595 ao invés de um decodificador e um MUX por alguns motivos, o primeiro seria a menor utilização de pinos do Arduino e o segundo, seria a existencia de bibliotecas no eco-sistema Arduino para controlar o display nesta configuração, como a biblioteca ShiftyDisplay, disponível no gerenciador de bibliotecas da IDE do Arduino.

O circuito foi montado e testado em protoboard, juntamente com o código referido no Apêndice A.

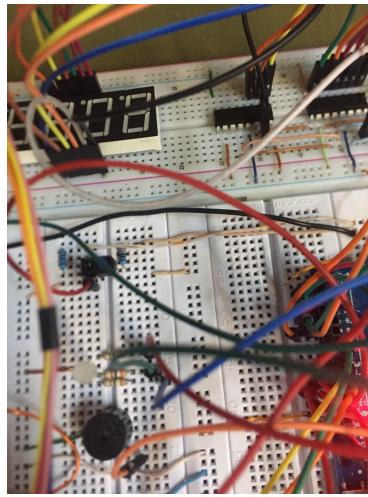


Figura 2: Circuito do display 7 Segmentos de 4 dígitos na protoboard.

### 3.1.2 Display gráfico 128X32

O display gráfico OLED 128x32, baseado no processador SSD1306, representa um aprimoramento sobre os LCDs 16x2 amplamente utilizados em kits didáticos por sua simplicidade.

No Arduino Neptune, o display esta conectado ao barramento I2C

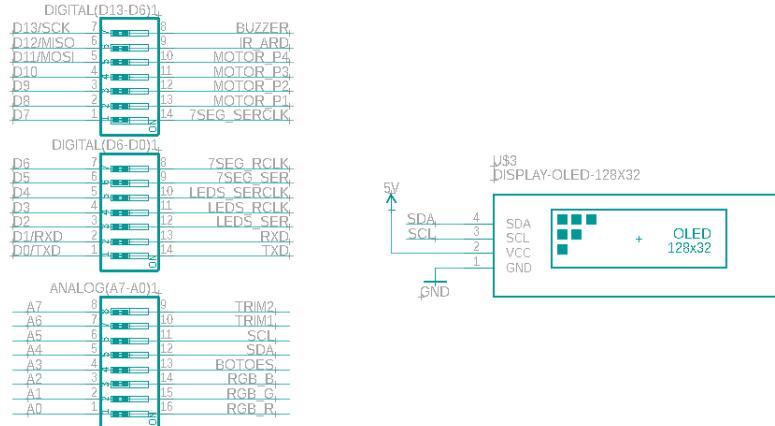


Figura 3: Conexão do display gráfico 128X32.

Apesar de mais simples, atualmente, ambos modelos possuem relativamente o mesmo preço de mercado, fazendo assim a escolha do display OLED a ser considerada, principalmente pela sua vantagem de ser gráfico. Uma das grandes vantagens do display OLED, é a sua capacidade de operar no modo U8X8, não utilizando nenhuma memoria RAM do microcontrolador para suas operações, apenas seu buffer interno.

O display 128x32 foi testado na protoboard em conjunto com varias bibliotecas disponíveis para o mesmo. Um dos códigos utilizados pode ser encontrado no Apêndice B.

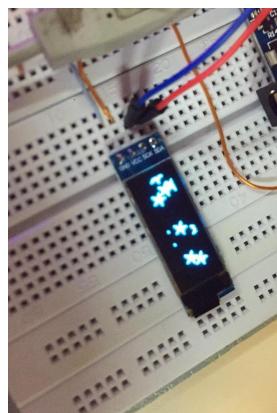


Figura 4: Circuito do display gráfico 128X32.

### 3.1.3 Driver para conexão do motor de passo

O buffer de Corrente ULN2003A permite a troca da corrente das portas do micro-controlador, que esta limitada pela corrente máxima de suas I/O, pela corrente da fonte externa, fazendo assim, a ligação de cargas mais pesadas possíveis no kit, particularmente, motores de passo.

O driver esta diretamente conectado as portas do Arduino, para direto controle dos passos do motor.

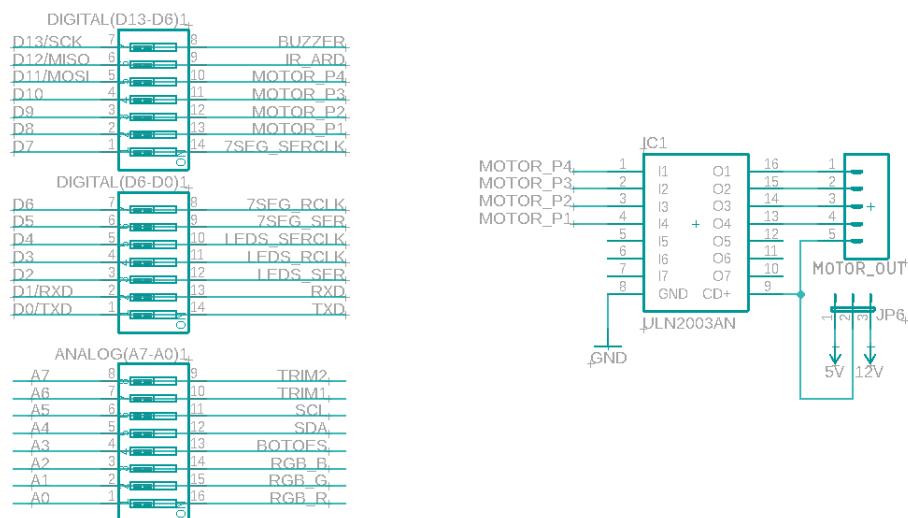


Figura 5: Conexão do driver para o motor de passo.

Na fase de prototipagem, utilizou-se um motor de 5V e uma biblioteca (Apêndice C) para testes do driver e, com um jumper no kit, é possível trocar a tensão comum do motor de 5 Volts para 12 Volts

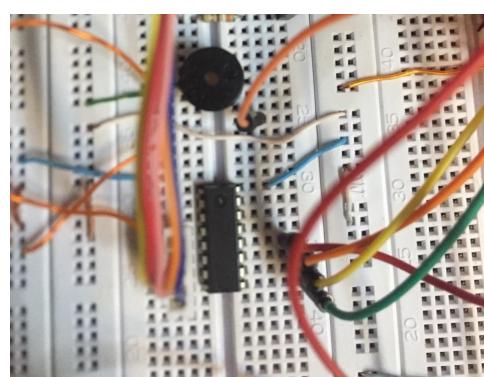


Figura 6: Circuito do driver para conexão do motor de passo.

### 3.1.4 Três botões em entrada analógica

Os botões do kit se assemelham aos antigos botões que eram determinados por divisor de tensão. Os três botões estão conectados em uma mesma entrada analógica que utilizam do conversor A/D do Arduino para funcionar.

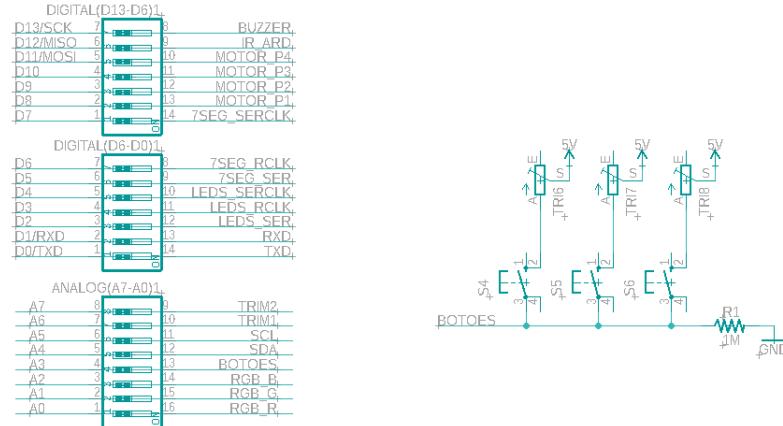


Figura 7: Conexão dos três botões na entrada analógica.

Quando os seus trimpots estão ajustados corretamente, os botões podem ser apertados simultaneamente, gerando um nível de tensão único. Efetivamente, o kit possui 6 botões.

Enquanto os testes estavam sendo feitos na protoboard, os botões estavam com resistores fixos, mas logo se notou que com a troca de USB ou troca de fonte de alimentação, os valores de conversão no código não era como esperado, então foi decidido que na versão final do kit, possuiria trimpots para o ajuste dos botões

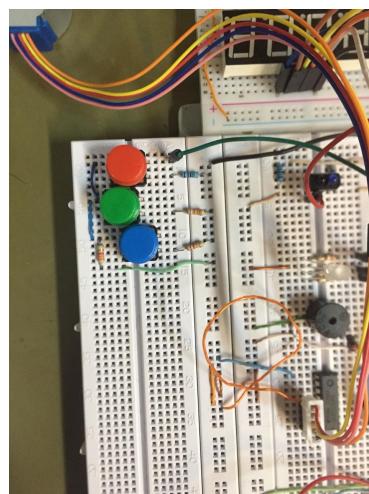


Figura 8: Circuito dos três botões na entrada analógica.

### 3.1.5 Led RGB

O Led RGB do kit está ligado diretamente as portas analógicas do Arduino, para que ocorra uma cobertura completa do espectro de cores por parte de software.

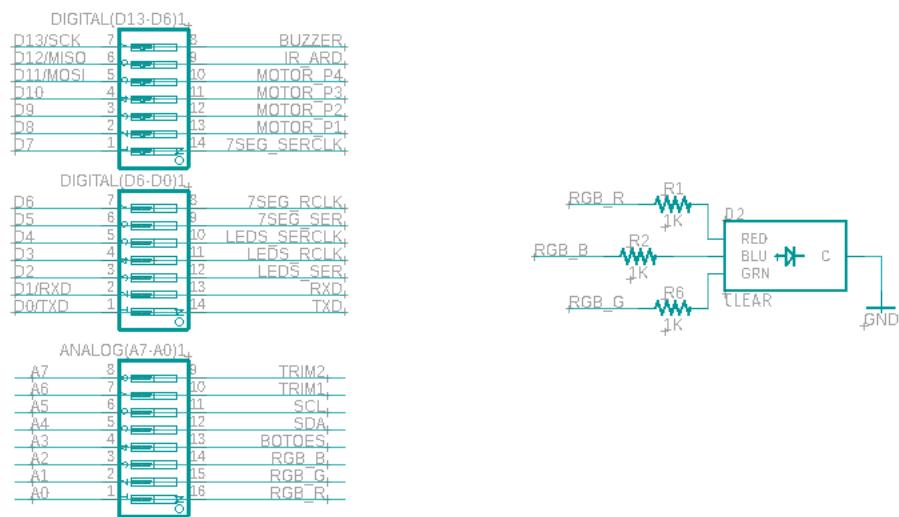


Figura 9: Conexão do Led RGB.

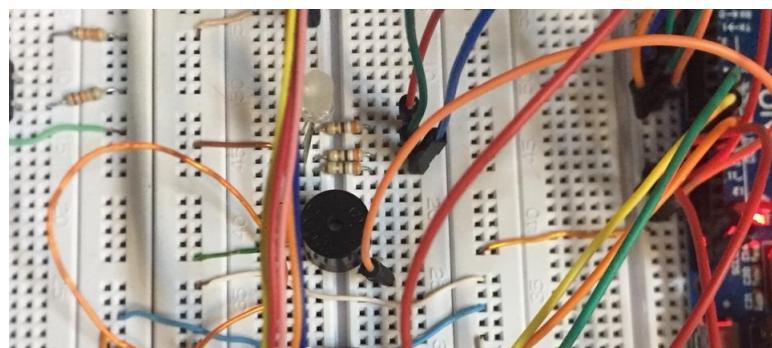


Figura 10: Circuito do Led RGB.

### 3.1.6 Sensor óptico

O sensor óptico TCRT5000, no kit, esta atuando como um sensor de proximidade ou um sensor de movimento. Com o emissor do fototransistor ligado a porta D12 do Arduino, é mais uma ferramenta para se utilizar em sala de aula e exemplificar o conceito de opto-acopladores.

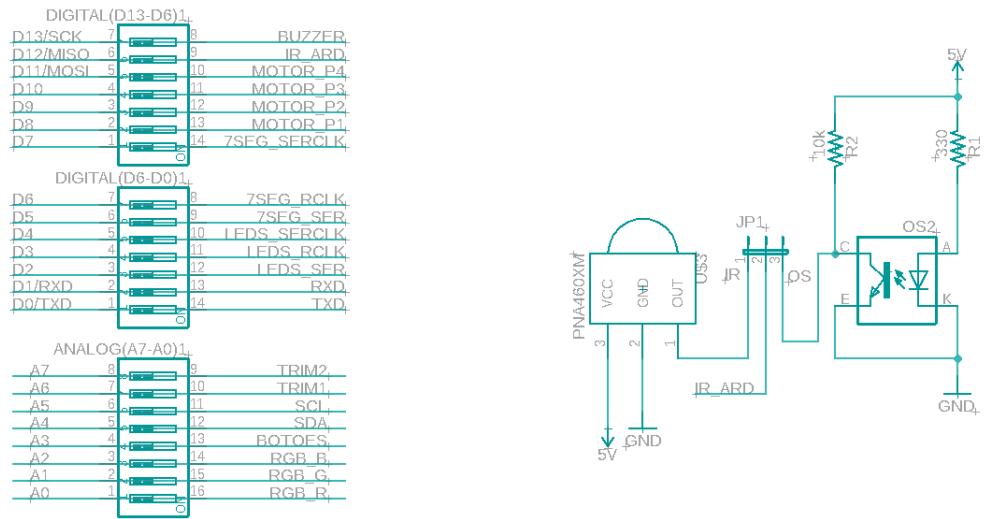


Figura 11: Conexão do sensor óptico.

Na fase de prototipagem, foi decidido que o TCRT5000 estaria multiplexado com o sensor infra-vermelho, através de um jumper físico no kit, devida a natureza sensível, em relação a ruído, dos sinais IR.

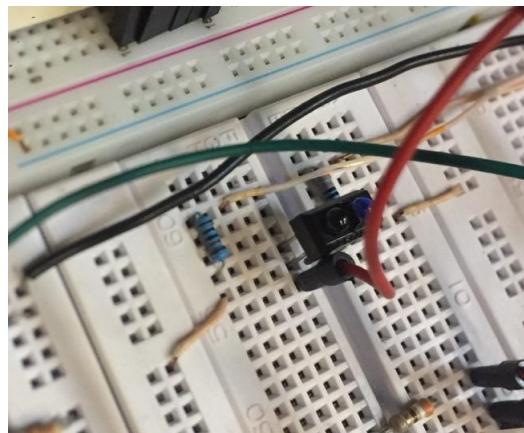


Figura 12: Circuito do sensor óptico.

### 3.1.7 Sensor infra-vermelho/controle remoto

O diodo receptor infra-vermelho de 38KHz é capaz captar sinais IR transmitidos de qualquer controle remoto. A intenção com o sensor, seria adicionar botões extras caso houvesse a necessidade, tendo a opção do controle remoto já integrada.

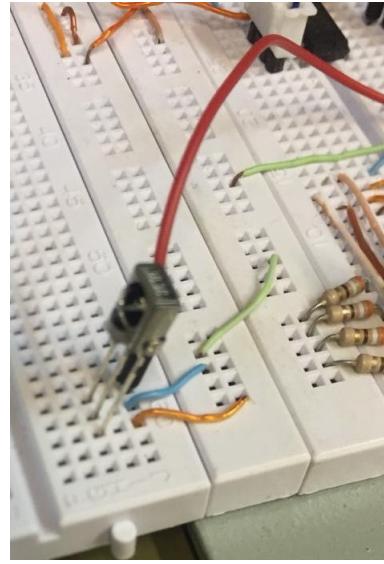


Figura 13: Circuito do sensor infra-vermelho.

### 3.1.8 Sequencial de Leds

No kit, estão multiplexados 8 Leds na saída de um shift register 74HC595. Utilizando apenas três pinos do Arduino para controlar o sequencial.

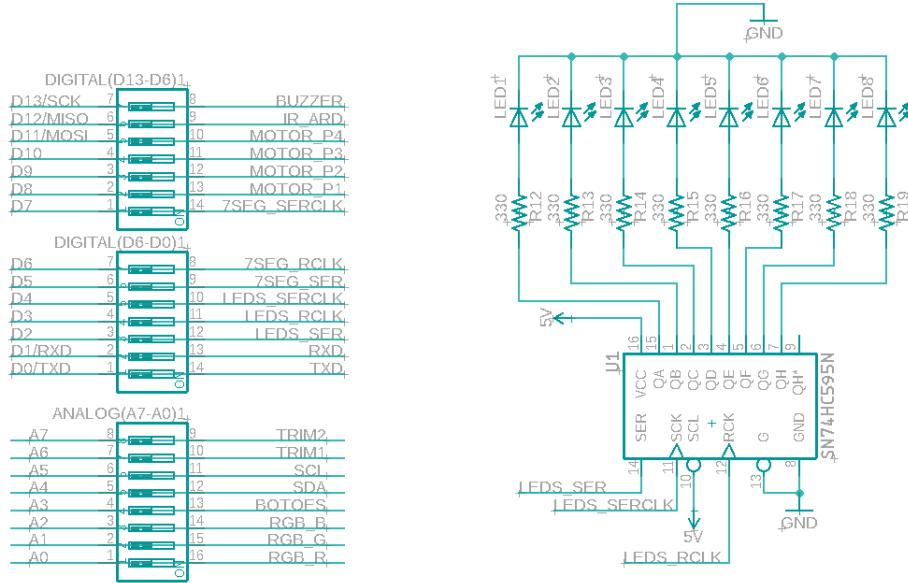


Figura 14: Conexão do sequencial de Leds.

Com a intenção de criar um sequencial intuitivo, foi escolhida uma biblioteca que possui a capacidade de endereçar cada Led individualmente.

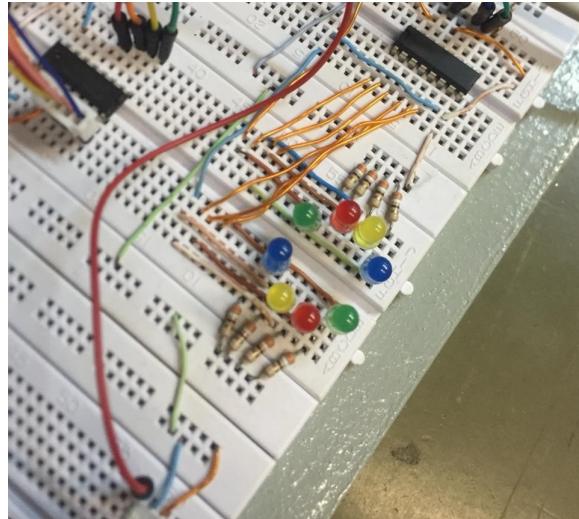


Figura 15: Circuito do sensor infra-vermelho.

## 3.2 MULTIPLEXAÇÃO

Devida a limitada quantidade de pinos do Arduino, alguns métodos foram estudados para tentar maximizar a quantidade de hardware, de interesse do curso, que poderia ser colocado no Arduino Neptune, sem que ficasse muito complexo o entendimento da plataforma.

### 3.2.1 Shift Register 74HC595

Para a multiplexação do kit foi amplamente utilizado o 74HC595, devido a sua vasta aplicação e relativo baixo custo. Hoje, shift registers são o jeito primário de se multiplexar, quando as I/O de um microcontrolador são limitadas.

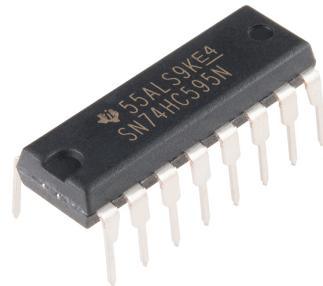


Figura 16: Shift Register 74HC595.

O 74HC595 é um shift register SIPO (Serial In Pararel Out), tendo como input três principais sinais: DATA, LATCH e CLOCK. A principal característica deste componente, é a sua capacidade de agir como uma memoria volátil, atualizando apenas o conteúdo de seus flip-flops internos, quando há um pulso no LATCH.

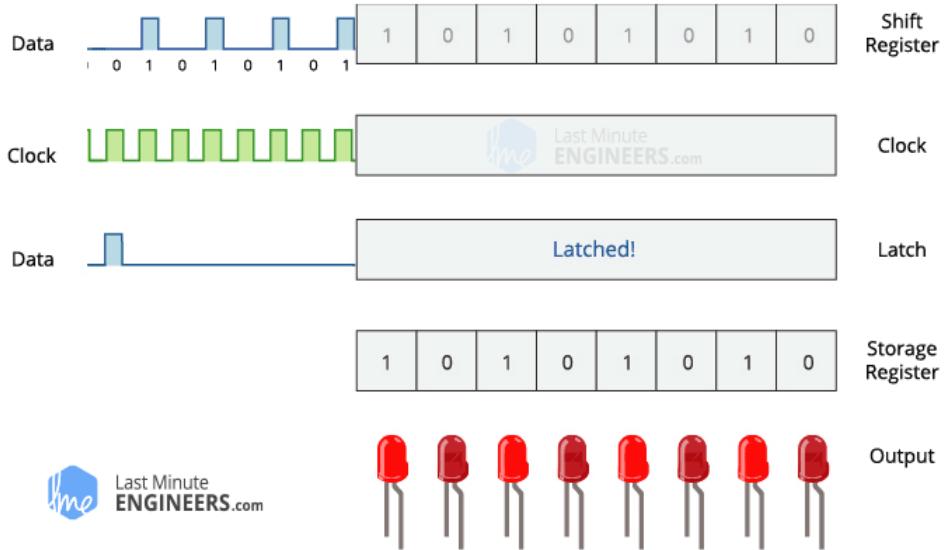


Figura 17: Sinas do Shift Register 74HC595.

### 3.2.2 Botões divisor de tensão

Através do conversor A/D do Arduino de 10 bits, colocou-se três botões em uma única porta analógica do microprocessador, fazendo um divisor de tensão.

Inicialmente, utilizamos resistores de precisão nos botões para tentar contrabalancear a variação de tensão que ocorria quando o kit alterava a conexão USB para outro dispositivo (o kit ainda não possuía fonte externa) mas no fim, decidimos utilizar trimpots para fazer um ajuste mais fino, caso os botões sofram algum tipo de queda de tensão imprevisível.

## 3.3 CONFEÇÃO

Depois que todos os circuitos foram escolhidos e testados em protoboards, com suas respectivas rotinas, partimos para a confecção do protótipo, que passou por duas versões: a placa de fenolite e a PCB.

### 3.3.1 Placa de fenolite cobreada

Inicialmente, nossas habilidades no software CAD Eagle eram suficientes para fazer a placa de fenolite, pois estávamos acostumados a trabalhar com este método de fabricação, apesar do layout da placa ser relativamente complicado. Contudo, enfrentamos desafios com o tamanho da placa, pois não havia percloroeto de ferro ( $\text{FeCl}_3$ ) suficiente para uma placa deste comprimento e largura, tivemos então que corroer a placa aos poucos. No fim, o protótipo foi um sucesso, fazendo o projeto avançar para o próximo etapa.

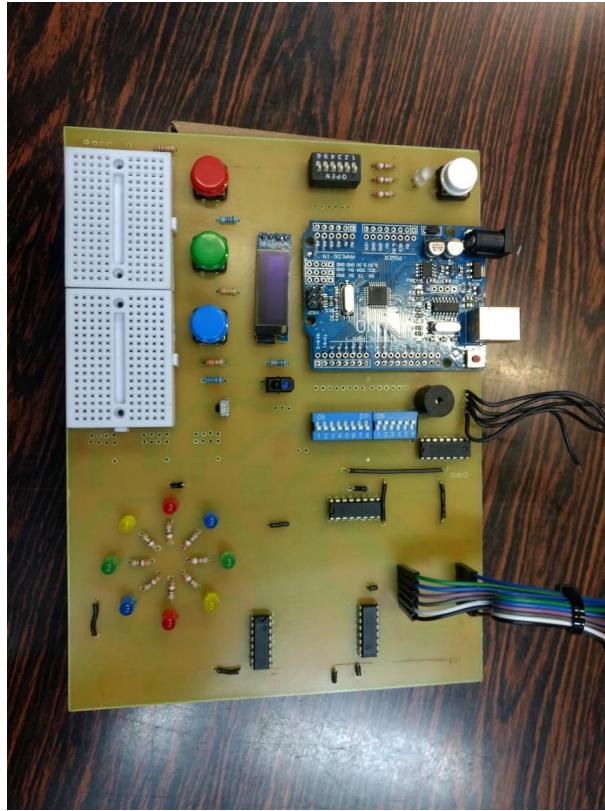


Figura 18: Placa de fenolite cobreada.

### 3.3.2 Placa de circuito impresso

Com o sucesso do primeiro protótipo, focou-se em desenvolver a versão final do kit: uma placa de circuito impresso desenvolvida e fabricada no Brasil. Dessa forma, era necessário possuir um maior conhecimento do software Eagle, já que com essa área da placa, o processo poderia sair relativamente caro e talvez não pudesse se repetir todas as etapas, caso houvesse algum problema.

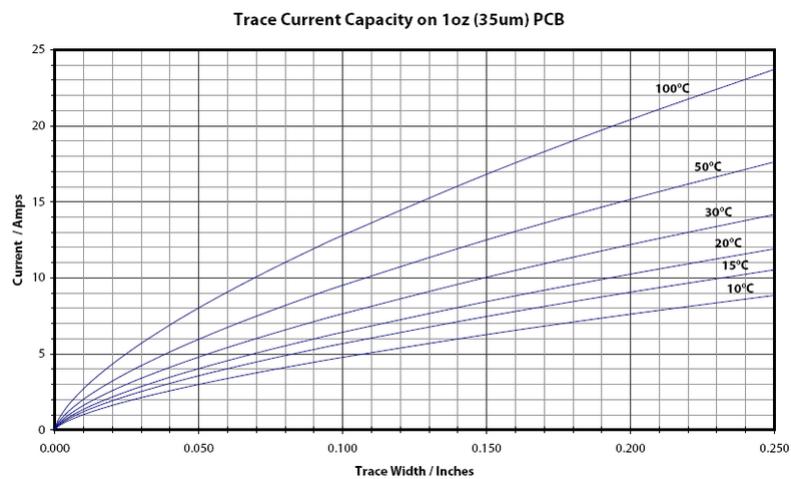


Figura 19: Relação da corrente pela largura da trilha.

Para tanto, começou-se a ler artigos da Autodesk intitulados "Eagle Academy" onde se aprendeu praticamente tudo sobre o software, o que possibilitou a confecção da PCB. O layout da placa foi renovado (Apêndice D), com uma trilha de 0,30mm para suportar 1 Amper e foram adicionados furos para o suporte de acrílico. Além disso, o esquemático foi refeito para estar mais organizado e integrado com o Arduino NANO (Apêndice E).

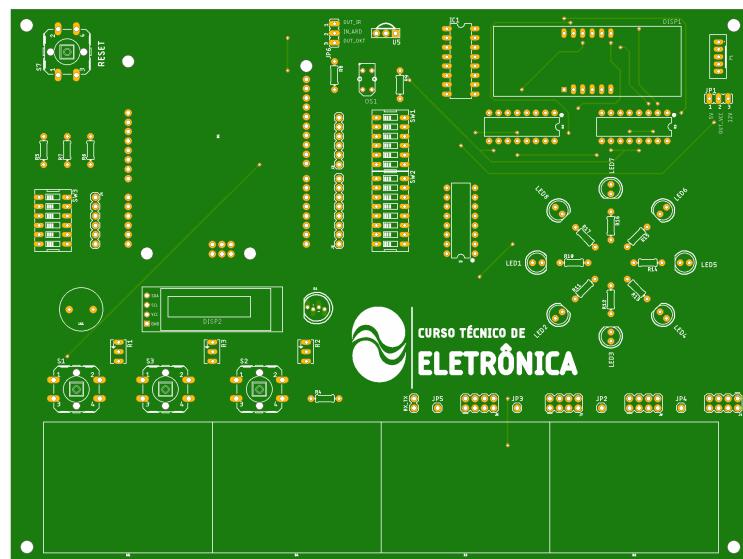


Figura 20: Placa PCB para o Arduino Neptune UNO.

### 3.4 INTEGRAÇÃO

Após o protótipo final estar concluído, optou-se confeccionar uma placa menor e compatível com o Arduino NANO, essencialmente ligando os dois protótipos do kit, fazendo uma versão "do professor".

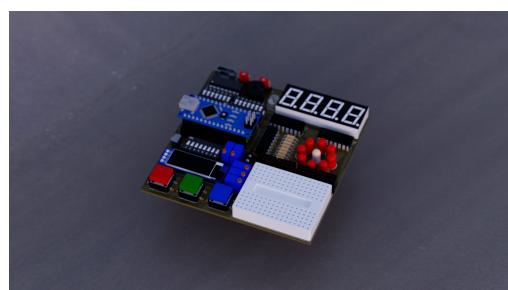


Figura 21: Render 3D Versão Arduino Neptune NANO - Fusion 360.

## **4 CONCLUSÃO**

O kit atende a expectativa de possuir um baixo custo de implementação quando comparado com outros similares encontrados no mercado.

Deve-se levar em consideração que não foi encontrado nenhum outro produto com a mesma quantidade de circuitos existentes no projeto desenvolvido.

Quanto à acessibilidade, o kit também se mostrou satisfatório, já que todo o hardware utilizado na produção da placa pode ser facilmente encontrado - separadamente - para uma possível confecção fora da sala de aula ou projetos extraclasse.

## REFERÊNCIAS

BITS, Brainy. **Connect and use an rgb led with an arduino.** Disponível em: <<https://www.brainy-bits.com/rgb-led-with-an-arduino/>>.

CHU, Alex. **How to define pth and npth in pcb design.** Disponível em: <<https://www.smart-prototyping.com/blog/How-to-Define-PTH-and-NPTH>>.

ENGINEERS, Last Minute. **In-depth how 74hc595 shift register works and interface with arduino.** Disponível em: <<https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial>>.

ENGINEERS, Last Minute. **Interface OLED Graphic Display Module with Arduino.** Disponível em: <<https://lastminuteengineers.com/oled-display-arduino-tutorial>>.

JHON. **Arduino Stepper Motor Interfacing – Circuit Diagram.** Disponível em: <<http://www.circuistoday.com/arduino-stepper-motor-interfacing>>.

MCL. **Pcb trace width vs. current table.** Disponível em: <<https://www.mclpcb.com/pcb-trace-width-vs-current-table>>.

PATTABIRAMAN, Krishna. **How to Set Up an IR Remote and Receiver on an Arduino - Circuit Basics.** Disponível em: <<http://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial>>.

PYNTO, Miguel. **Arduino library for driving 7-segment displays using shift registers.** Disponível em: <<https://miguelpynto.github.io/ShiftDisplay>>.

RAY. **Multiple button inputs using arduino analog pin.** Disponível em: <<https://rayshobby.net/wordpress/multiple-button-inputs-using-arduino-analog-pin>>.

SATTEL, Sam. **Schematic basics part 1 search, place, move eagle blog.** Disponível em: <<https://www.autodesk.com/products/eagle/blog/schematic-basics-part-1>>.

SATTEL, Sam. **Routing autorouting - pcb layout basics 2 | eagle | blog.** Disponível em: <<https://www.autodesk.com/products/eagle/blog/routing-autorouting-pcb-layout-basics-2>>.

SATTEL, Sam. **Design rule check - pcb layout basics 3 | eagle | blog**. Disponível em: <<https://www.autodesk.com/products/eagle/blog/design-rule-check-pcb-layout-basics-3/>>.

VIDAL, Vitor. **Sensor óptico tcrt5000 com arduino aprenda a usar**. Disponível em: <<https://blog.eletrogate.com/sensor-optico-tcrt5000-com-arduino/>>.

(ENGINEERS, 2019a) (ENGINEERS, 2019b) (VIDAL, 2017) (SATTEL, 2017a) (SATTEL, 2017b) (SATTEL, 2017c) (JHON, 2018) (BITS, 2015) (CHU, 2017) (MCL, 2018) (RAY, 2010) (PYNTO, 2017) (PATTABIRAMAN, 2019)

## APÊNDICE A – CÓDIGO DISPLAY 4 DÍGITOS 7 SEGMENTOS

```
#include <ShiftDisplay.h>

ShiftDisplay display(COMMON_CATHODE, 4); //construtor do display, sendo seu
modelo catodo comum e possuindo 4 digitos

void setup() {
    for (int i = 3000; i > 0; i--) {
        display.show(i, 5, ALIGN_RIGHT); //manda um numero(ou caracter) pro display e
        mostra por 5ms
        display.setDot(1, true); //coloca o ponto na posicao 1 no numero(ou caracter)
        guardado no buffer
        display.show(5); //mostra o numero com o ponto por 5 ms
    }
    display.set("boom"); //guarda um numero(ou caracter) no buffer do display
}

void loop() {
    display.show(); //mostra o que esta no buffer indefinidamente
}
```

Figura 22: Código display 4 dígitos 7 segmentos

## APÊNDICE B – CÓDIGO DISPLAY OLED 128X32

```
#include <U8g2lib.h>
#include <Wire.h>

U8G2_SSD1306_128X32_UNIVISION_F_SW_I2C display(U8G2_R0, SCL, SDA,
U8X8_PIN_NONE); //modelo e construtor do display

void setup(void) {
    display.begin(); //inicia o display
}

void loop(void) {
    float x = millis();
    display.clearBuffer();           //limpa o buffer do display
    display.setFont(u8g2_font_9x15_tf); //escolhe a fonte do texto(suporte ate
    fontes de 32px)
    display.drawStr(0,12,"NEPTUNE-V1"); //escreve uma string em uma posicao
    especifica
    display.setCursor(0,26); //move o cursor
    display.print(millis()-x); //printa qualquer tipo de variavel
    display.print("ms");
    display.sendBuffer();           //da update no buffer do display(imprime tudo
    junto)
}
```

Figura 23: Código display OLED

## APÊNDICE C – CÓDIGO MOTOR DE PASSO

```
#include <Stepper.h>
#include <U8g2lib.h>
#include <Wire.h>
const int stepsPerRevolution = 500; //TIRADO DO DATASHEET
U8G2_SSD1306_128X32_UNIVISION_F_SW_I2C display(U8G2_R0, SCL, SDA, U8X8_PIN_NONE);

Stepper motor(stepsPerRevolution, 8, 10, 9, 11); //DEFINE OS PINOS DO MOTOR NO
ARDUINO, ALEM DO NUMERO DE PASSOS POR GRAUS, RETIRADO DO DATASHEET DO MOTOR ROHS
28BYJ-48

void setup () {
    pinMode(A3, INPUT); // DEFINE O PINO DOS BOTOES COMO INPUT
    Serial.begin(9600);
    display.begin(); //inicia o display
}
```

Figura 24: Código motor de passo - 1

```
void loop () {
int z;

z = analogRead(3); //LE O VALOR DO BOTAO NA PORTA ANALOGICA 3
Serial.println(z);
display.clearBuffer();
display.clear();
if (z > 800 && z <900) { // BOTAO VERMELHO
display.setFont(u8g2_font_9x15_tf);
display.drawString(0,12,"7RPM / 180");
display.sendBuffer();
motor.setSpeed(70); // DEFINE A ROTACAO DO MOTOR PARA 7 RPM
motor.step(2048); // MANDA O MOTOR DAR 1 VOLTA COMPLETA NO SENTIDO HORARIO
}
if (z > 600 && z < 800){ // BOTAO VERDE
display.setFont(u8g2_font_9x15_tf);
display.drawString(0,12,"5RPM / -90");
display.sendBuffer();
motor.setSpeed(50); // DEFINE A ROTACAO DO MOTOR PARA 5 RPM
motor.step(-1024); // MANDA O MOTOR DAR MEIA VOLTA VOLTA NO SENTIDO ANTI-HORARIO
}
if (z > 400 && z < 600){ // BOTAO AZUL
display.setFont(u8g2_font_9x15_tf);
display.drawString(0,12,"3RPM / -90");
display.sendBuffer();
motor.setSpeed(10); // SETA A ROTACAO DO MOTOR PARA 3 RPM
motor.step(512); // MANDA O MOTOR ANDAR 90 GRAUS NO SENTIDO HORARIO
}
}
```

Figura 25: Código motor de passo - 2

## APÊNDICE D – LAYOUT ARDUINO NEPTUNE UNO

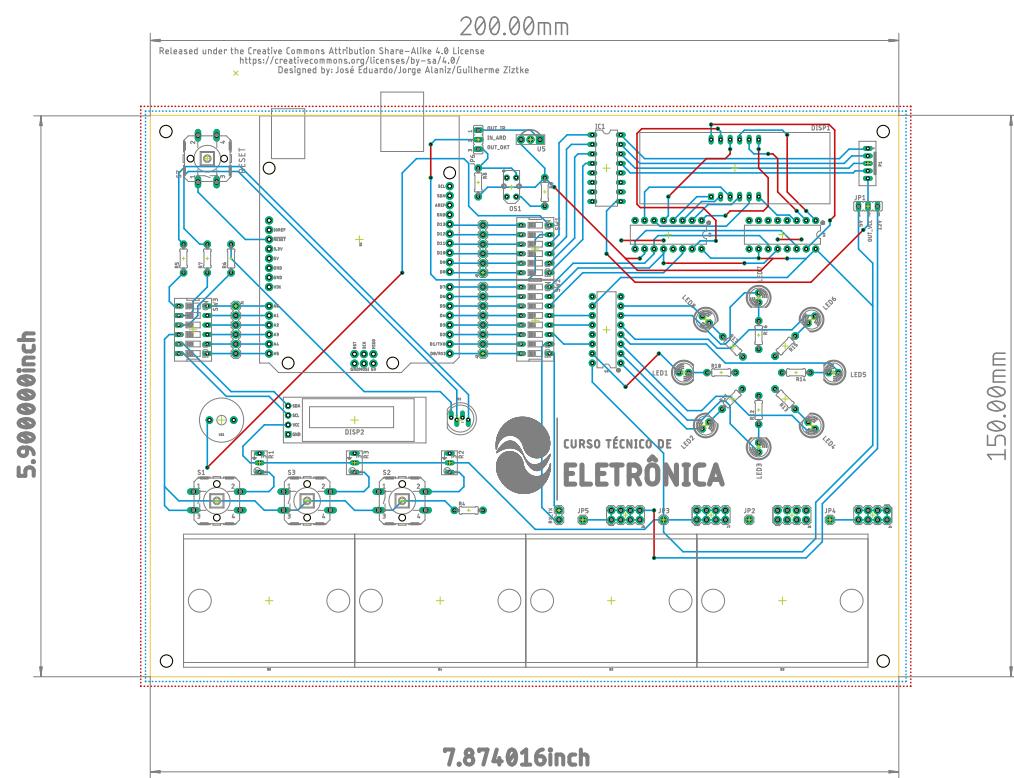


Figura 26: Layout Arduino Neptune UNO

## APÊNDICE E – ESQUEMÁTICO ARDUINO NEPTUNE V1.0

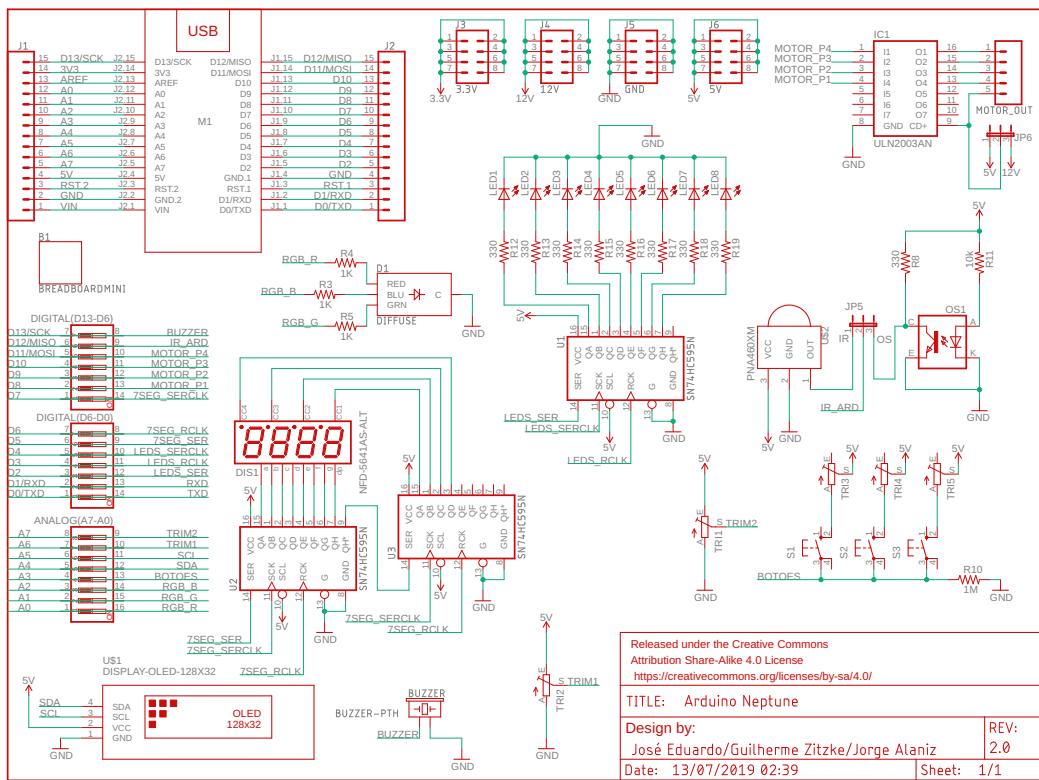


Figura 27: Esquemático Arduino Neptune V1.0