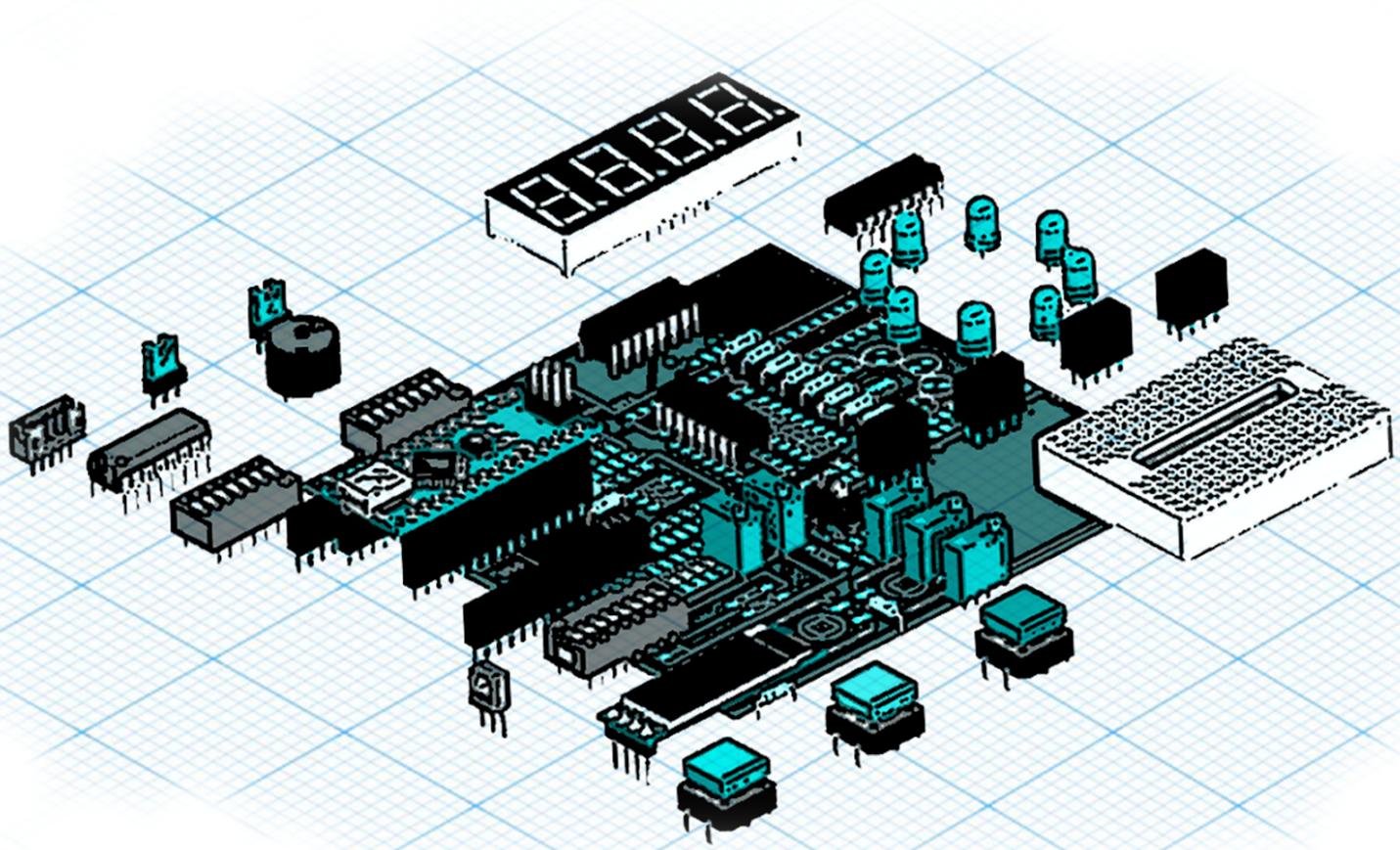
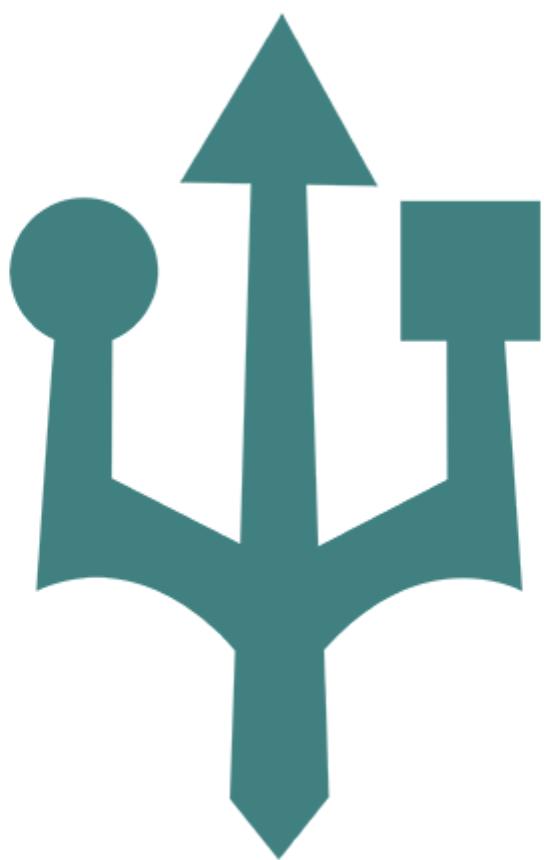


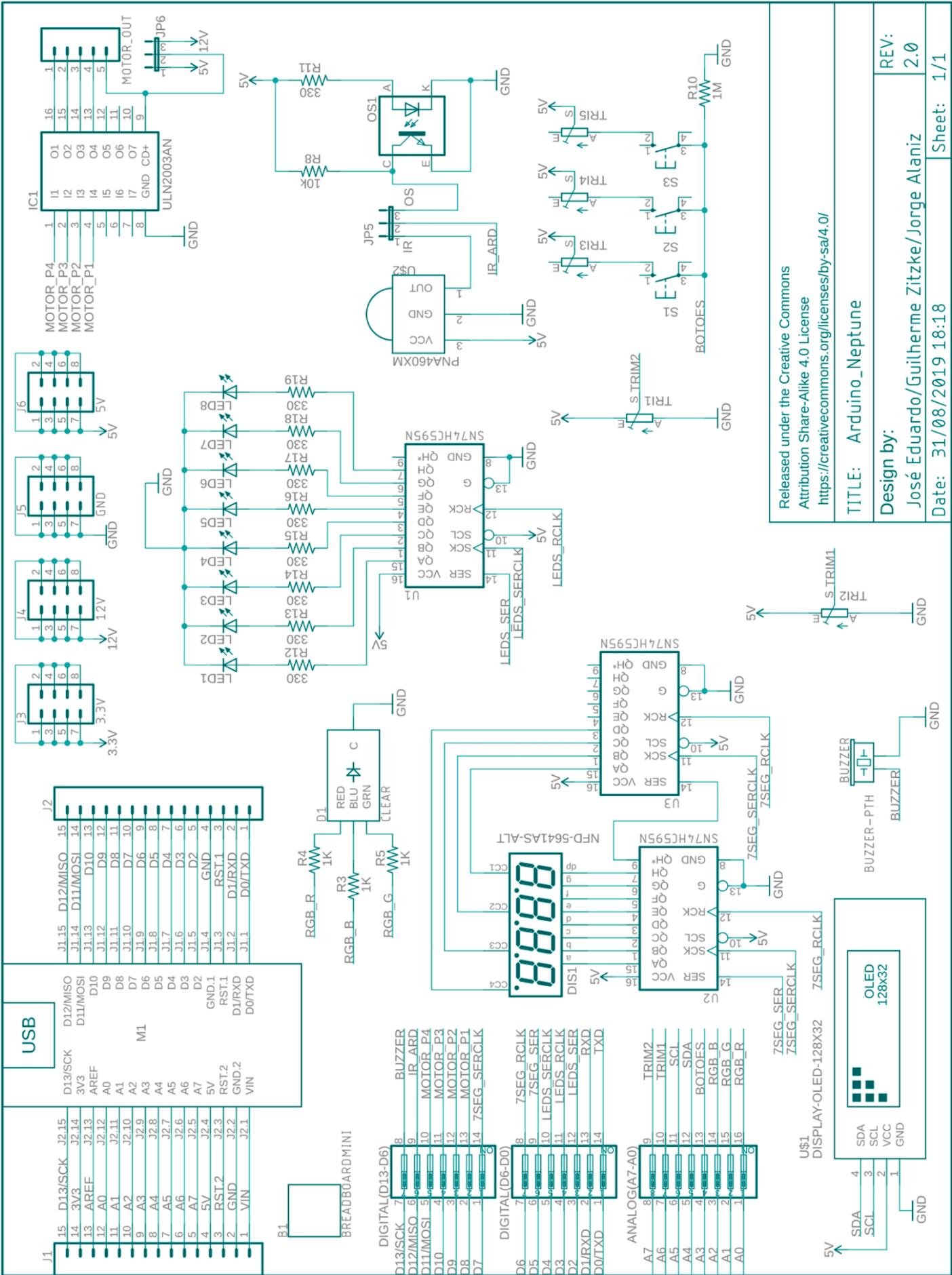


ARDUINO NEPTUNE

LIVRO  SKETCH







Released under the Creative Commons
Attribution Share-Alike 4.0 License
<https://creativecommons.org/licenses/by-sa/4.0/>

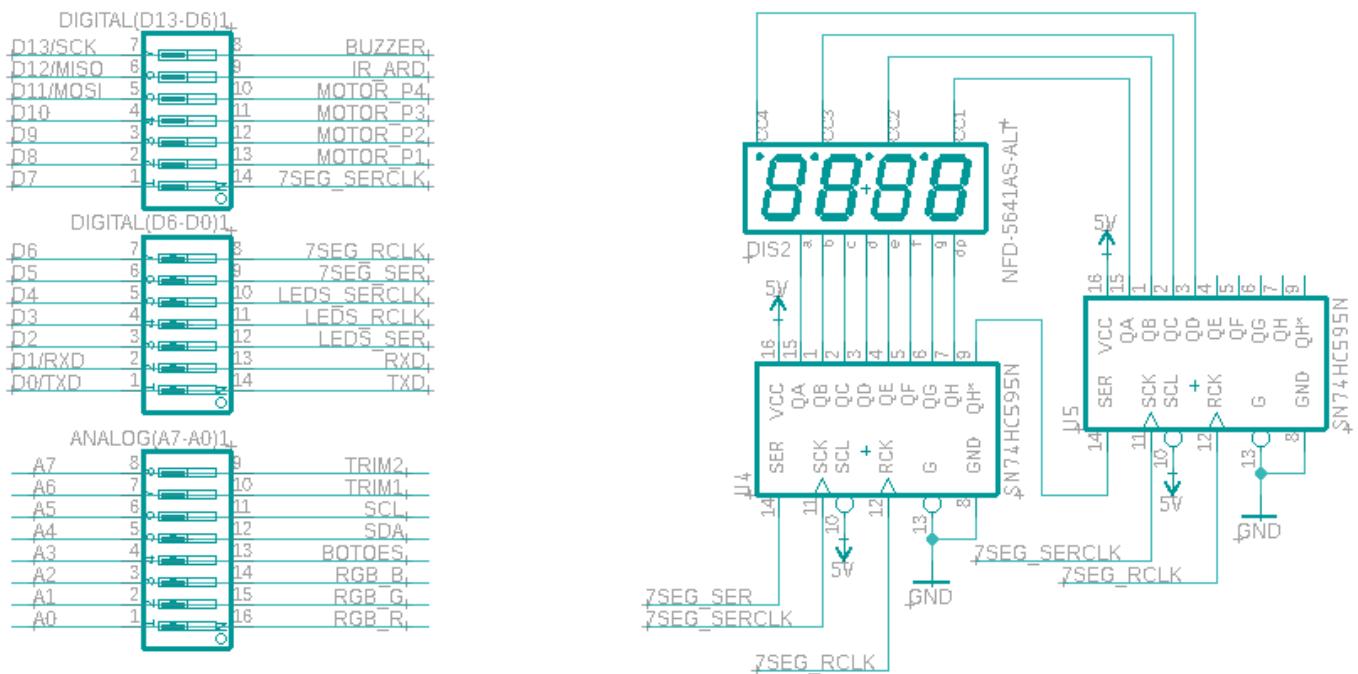
TITLE: Arduino_Neptune

Design by:
José Eduardo/Guilherme Zitzke/Jorge Alaniz

REV:
2.0

Date: 31/08/2019 18:18
Sheet: 1/1

Contador Decrescente



```

#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
    }

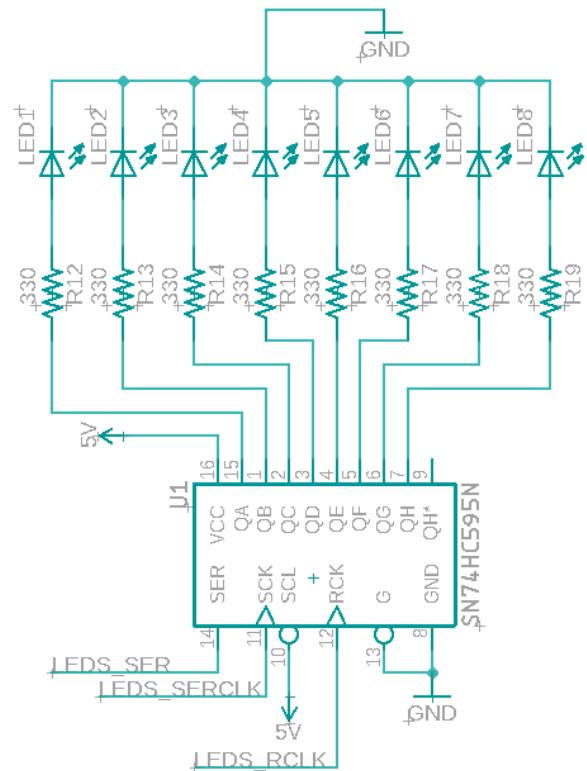
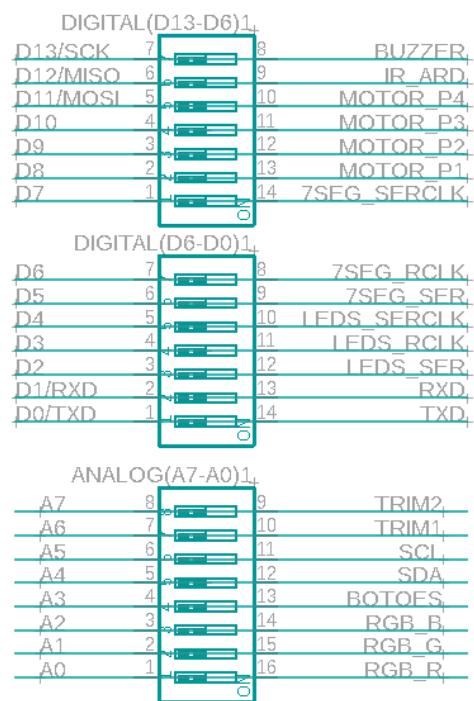
    display.set("boom"); //guarda um numero(ou caracter) no buffer do display
}

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

```

Obs: Utiliza a biblioteca “ShiftDisplay”, disponível no Gerenciador de Bibliotecas na Arduino IDE ou Github.

Sequencial de LEDs



```

#include <ShiftRegister74HC595.h>

ShiftRegister74HC595 sr (1, 2, 4, 3); // contrutor, 1 shift register com seus respectivos
pinos de dados, latch e clock

void setup() {
}

void loop() {
    sr.setAllHigh(); // liga todos
    delay(500);

    sr.setAllLow(); // desliga todos
    delay(500);

    for (int i = 0; i < 8; i++) {

        sr.set(i, HIGH); // liga o led i
        delay(250);
    }
}

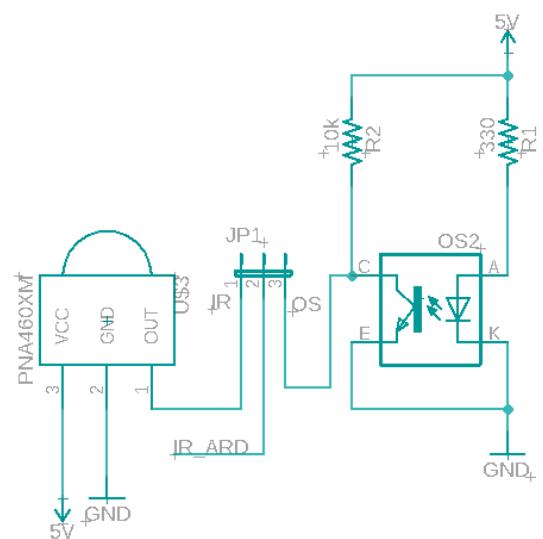
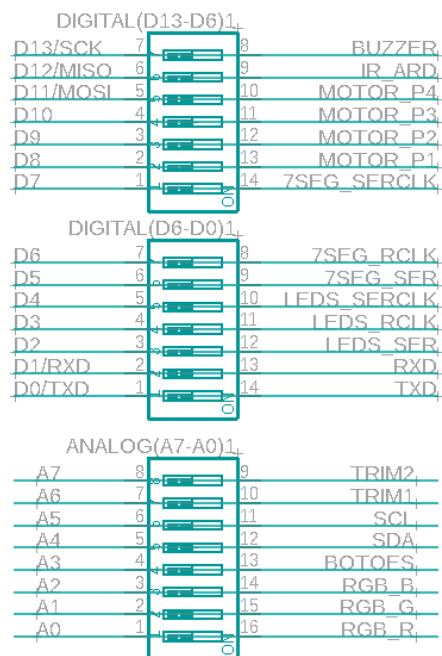
uint8_t pinValues[] = { B10101010 }; // manda um sequencial
sr.setAll(pinValues);
delay(1000);

}

```

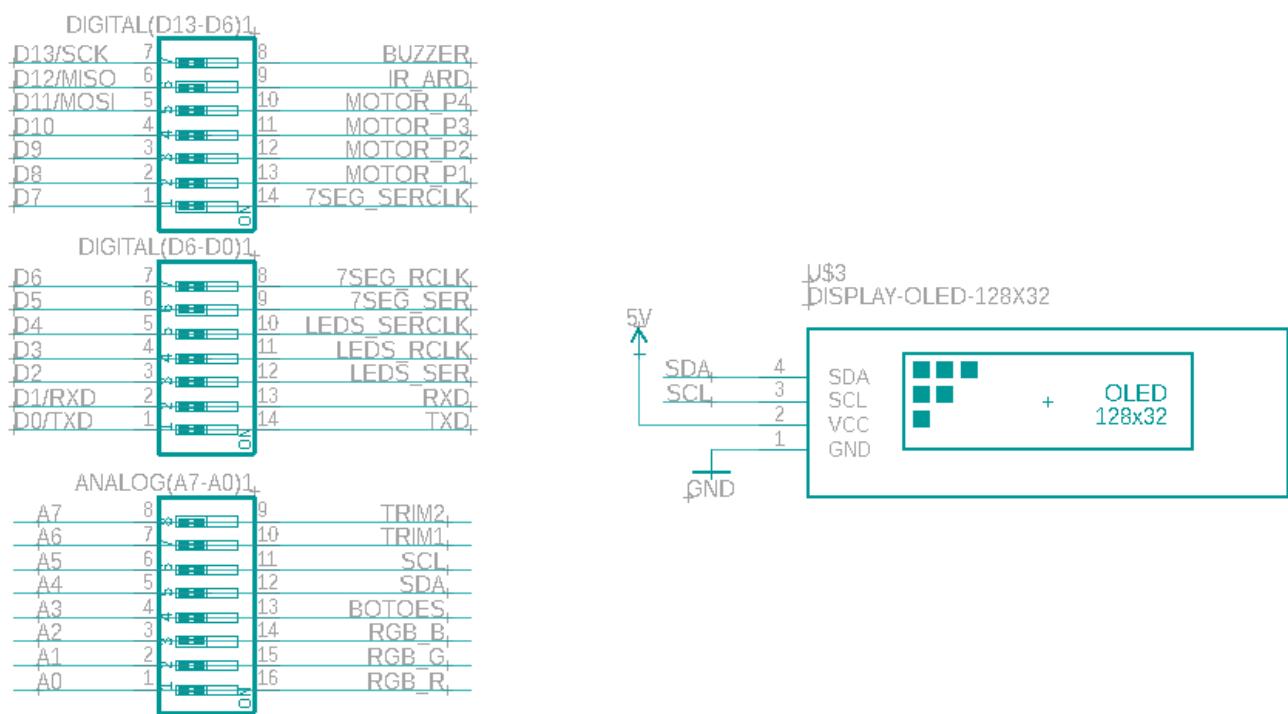
Obs: Utiliza a biblioteca “ShiftRegister74HC595”, disponível no Gerenciador de Bibliotecas na Arduino IDE ou Github.

Sensor de proximidade



```
void setup() {  
}  
  
void loop() {  
    int x = digitalRead(12);  
    if (x==1){  
        analogWrite(A2,255);  
    }  
    else{  
        analogWrite(A2,0);  
    }  
    delay(10);  
}
```

Display OLED



```

#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

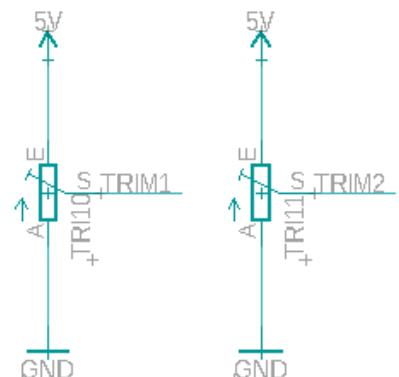
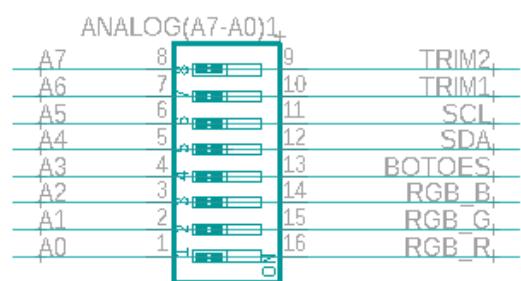
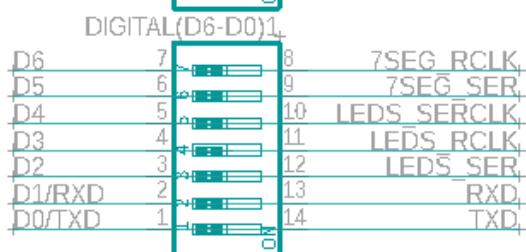
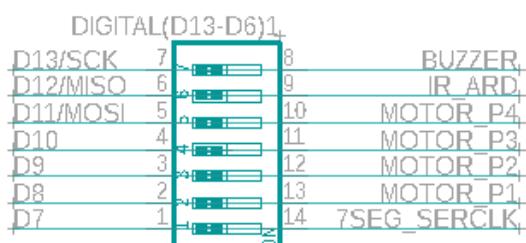
void setup(void) {
    display.begin(); //inicia o display
}

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)
}

```

Obs: Utiliza a biblioteca “U8g2lib”, disponível no Gerenciador de Bibliotecas na Arduino IDE ou Github. Na página do autor da biblioteca, há várias fontes de texto com tamanhos e estilos diferentes.

Conversor A/D



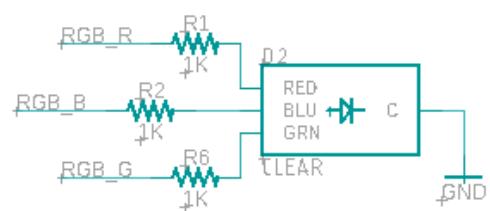
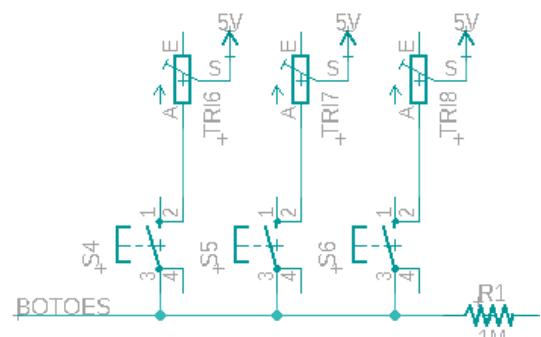
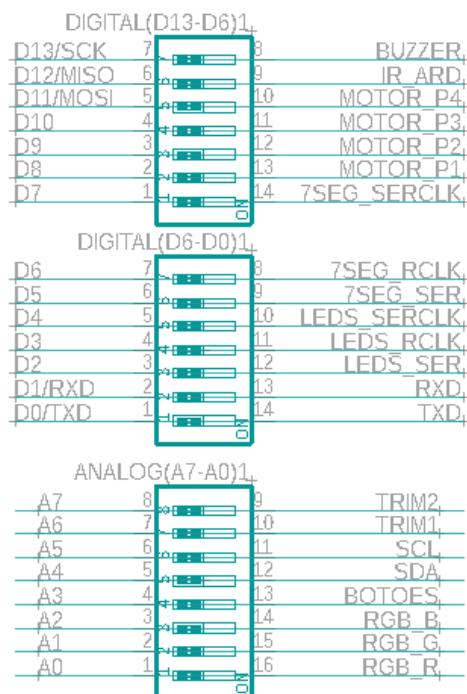
```
#include <ShiftDisplay.h>

ShiftDisplay display(COMMON_CATHODE,4);

void setup() {
    pinMode(A6,INPUT);
}

void loop() {
    float x = analogRead(A6);
    float y = x*(5.0/1023.0);
    display.set(y, ALIGN_LEFT);
    display.setDot(0, true);
    display.show(10);
}
```

RGB com botões



```
#include <AnalogButtons.h>

void b1Click() {
    Serial.print("botao 1 foi pressionado\n");
}

void b1Hold(){
    Serial.print("botao 1 esta sendo segurado\n");
    analogWrite(A0,255);
    analogWrite(A1,0);
    analogWrite(A2,0);
}

void b2Click() {
    Serial.print("botao 2 foi pressionado\n");
}

void b2Hold() {
    Serial.print("botao 2 esta sendo segurado\n");
    analogWrite(A1,255);
    analogWrite(A0,0);
    analogWrite(A2,0);
}

void b3Click() {
    Serial.print("botao 3 foi pressionado\n");
}

void b3Hold() {
    Serial.print("botao 3 esta sendo segurado\n");
    analogWrite(A2,255);
    analogWrite(A0,0);
    analogWrite(A1,0);
}

AnalogButtons analogButtons(A3, INPUT);
```

```
Button b1 = Button(165, &b1Click, &b1Hold);
Button b2 = Button(200, &b2Click, &b2Hold);
Button b3 = Button(288, &b3Click, &b3Hold);

void setup() {
    Serial.begin(9600);
    Serial.println("Testando botoes");
    analogButtons.add(b1);
    analogButtons.add(b2);
    analogButtons.add(b3);
}

void loop() {
    analogButtons.check();
    //configure();
}

void configure() {
    unsigned int value = analogRead(A3);
    Serial.println(value);
    delay(250);
}
```

Obs: Utiliza a biblioteca “AnalogButtons”, disponível no Gerenciador de Bibliotecas na Arduino IDE ou Github.

