

# 2023/24 Projeto 2

Exemplo de Apresentação

Aluno: Eduardo Junqueira nº30241

Aluno: Jorge Pereira n°30248

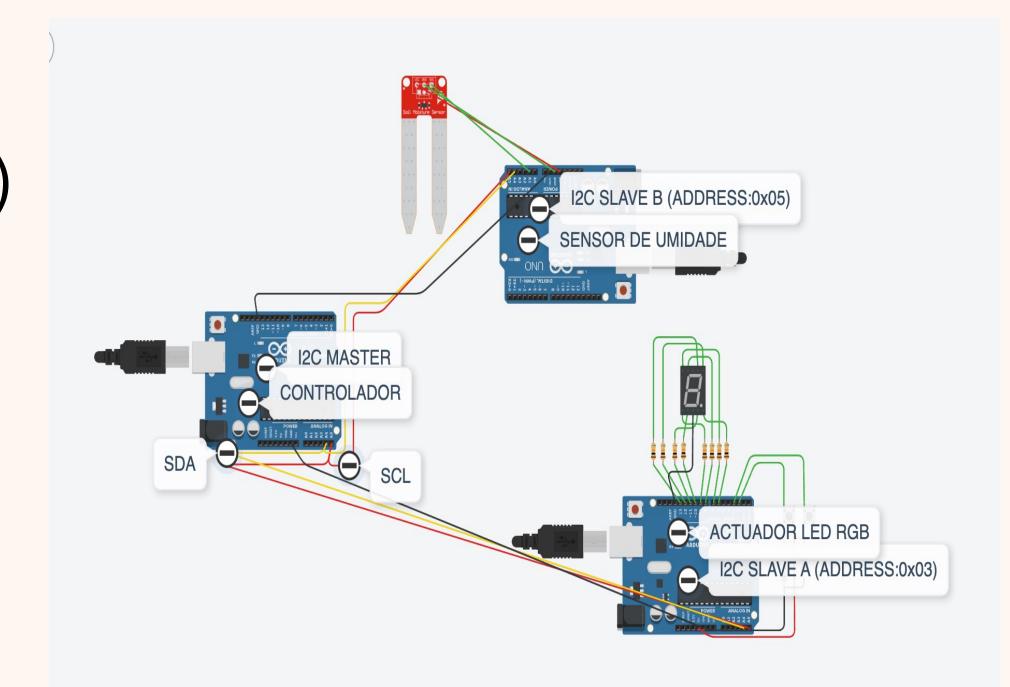
Orientação:



ENGENHARIA DE REDES E SISTEMAS <u>DE COMPUTADORES</u>

### Sumário

- →Introdução ao exercício (Slide 3-4)
  - → Dificuldades de Desenvolvimento(Slide 5)
  - → Desenvolvimento (Slide 5-10)
  - → Resultados(Slide 11)
  - → Demonstração (Slide 12-13)
  - → Conclusões (Slide 14)







# Introdução

- → No âmbito do desafio 2, fomos desafiados ao Controlo de displays de 7 segmentos através de um sensor de humidade do solo.
- → Desenvolvimento do trabalho:
- -Exercício 3 -Visor de 7 Segmentos.
- -Exercício 6-Requisitos base:
- --Exercício 7-2Slaves e 1 Master.

→O sistema será composto por dois elementos principais(Slaves): o sensor de umidade, responsável por medir a umidade, e o visor de sete seguimentos responsável pela cooperação do medida do sensor de umidade



## Trabalhos Relacionados

- -Neste desafio usamos 3 trabalhos para a conclusão total do desafio
- -→Trabalho 3;
- -→Trabalho6;
- -→Trabalho7;





#### Dificuldades de Desenvolvimento

→ Durante o desenvolvimento do nosso projeto sentimos algumas dificuldades em montar o conjunto de microcontroladores e dispositivos para conseguirmos fazer o necessário que seria necessário para resolver o que fora pedido.

→Esse problema foi ultrapassado com a ajuda dos trabalhos 3 e 7 que nos puderam dar uma ideia de como prosseguir para a conclusão deste desafio



#### Desenvolvimento Master

-Aqui está presente a invocação da função do Monitorial Serial do Slave B que é importante:

```
Wire.requestFrom(SLAVE A,17);
while (Wire.available())
 char c = Wire.read();
 Serial.print(c);
// delay de 1 segundo
delay(1000);
// MASTER WRITES to SLAVE B
Wire.beginTransmission(SLAVE_B);
                    // begin transmission to Slave B
Wire.write("MASTER > SLAVE B\n");
Wire.endTransmission();
                     // stop transmitting
// delay de 1 segundo
delay(1000);
// MASTER READS from SLAVE B
Wire.requestFrom(SLAVE_B,17);
while (Wire.available())
 char c = Wire.read();
 Serial.print(c);
// delay de 1 segundo
delay(1000);
```





#### Desenvolvimento Slave A

-Aqui no Slave A temos o visor de 7 segmentos que representa a enumeração dos 7 segmentos

```
// Verifica se o número está dentro do intervalo permitido.
if (num >= 0 && num <= MAX_DISPLAY_NUM) {
     digitalWrite(A, HIGH);
     digitalWrite(B, HIGH);
     digitalWrite(C, HIGH);
     digitalWrite(D, HIGH);
     digitalWrite(E, HIGH);
     digitalWrite(F, HIGH);
     digitalWrite(G, LOW);
     delay(750);
     break;
    case 1:
     digitalWrite(B, HIGH);
     digitalWrite(C, HIGH);
     delay(750);
     break;
   case 2:
     digitalWrite(A, HIGH);
     digitalWrite(B, HIGH);
     digitalWrite(G, HIGH);
     digitalWrite(E, HIGH);
     digitalWrite(D, HIGH);
    delay(750);
     break;
     digitalWrite(A, HIGH);
     digitalWrite(B, HIGH);
     digitalWrite(G, HIGH);
     digitalWrite(C, HIGH);
     digitalWrite(D, HIGH);
    delay(750);
     break;
     digitalWrite(F, HIGH);
     digitalWrite(G, HIGH);
     digitalWrite(B, HIGH);
     digitalWrite(C, HIGH);
    delay(750);
     digitalWrite(A, HIGH);
     digitalWrite(F, HIGH);
     digitalWrite(G, HIGH);
     digitalWrite(C, HIGH);
     digitalWrite(D, HIGH);
    delay(750);
```

```
case 6:
 digitalWrite(A, HIGH);
 digitalWrite(F, HIGH);
 digitalWrite(G, HIGH);
 digitalWrite(C, HIGH);
 digitalWrite(D, HIGH);
 digitalWrite(E, HIGH);
delay(750);
 break;
case 7:
 digitalWrite(A, HIGH);
 digitalWrite(B, HIGH);
 digitalWrite(C, HIGH);
delay(750);
 break;
case 8:
 digitalWrite(A, HIGH);
 digitalWrite(B, HIGH);
 digitalWrite(C, HIGH);
 digitalWrite(D, HIGH);
 digitalWrite(E, HIGH);
 digitalWrite(F, HIGH);
 digitalWrite(G, HIGH);
delay(750);
 break;
case 9:
 digitalWrite(A, HIGH);
 digitalWrite(B, HIGH);
 digitalWrite(C, HIGH);
 digitalWrite(F, HIGH);
 digitalWrite(G, HIGH);
delay(750);
 break;
```





### Desenvolvimento Slave B

-O slave B aqui serve para Monitorizar a umidade e através da implementação da figura seguinte é possível implementar essas funções de acordo com o I2C

```
#include <Wire.h>
#define SLAVE B 0x05
#define sensor A0
//Slave B mede a humidade através do sensor de umidade do solo
void setup() {
   Serial.begin(9600);
   Wire.begin(SLAVE B); // Inicializa o módulo de hardware i2c como SLAVE com o endereço SLAVE A
   Wire.onRequest(requestEvent);
   Wire.onReceive(receiveEvent);
   pinMode(LED BUILTIN, OUTPUT);
   digitalWrite(LED_BUILTIN, HIGH);
void loop() {
float readtemp(){
 float volt=(analogRead(sensor) * 0.004882814);
   float temp = (volt - 0.5) * 100.0;
// requestEvent Handler that executes whenever data is requested by master
void requestEvent() {
   int c=(int)readtemp()+40;
 Serial.println(c);
   Wire.write(c);
// receiveEvent Handler that executes whenever data is received from the master
void receiveEvent(int numBytes) {
   digitalWrite(LED_BUILTIN,!digitalRead(LED_BUILTIN));
   while (Wire.available()) {
       char c = Wire.read();
       Serial.print(c);
```



#### Resultados

```
// Ficha Prática 7 - Comunicação I2C (One Master-Multiple Slave)
// MASTER 1(Arduino Uno R3)
// Pinos I2C no Arduino Uno:
// - SDA: pino A4
// - SCL: pino A5
// - GND: Não esquecer de ligar o sinal de ground, entre os divers
//Nome:Eduardo Junqueira nº30241 / Jorge Pereira n30248
//Link:https://www.tinkercad.com/things/99C1HY0QTAX-copy-of-fp7ex2i
#include <Wire.h> //bliblioteca necessária para I2C
#define SLAVE A 0x03
#define SLAVE B 0x05
void setup()
 Serial.begin(9600);
                       // start serial for output
Wire.begin();
                       // join i2c bus (address option
void loop()
// MASTER WRITES to SLAVE A
 Wire.beginTransmission(SLAVE_A); // begin transmission to Slave
 Wire.write("MASTER > SLAVE_A\n");
 Wire.endTransmission();
                       // stop transmitting
 // delay de 1 segundo
 delay(1000);
 // MASTER READS from SLAVE_A
 Wire.requestFrom(SLAVE_A,17);
 while (Wire.available())
  char c = Wire.read();
  Serial.print(c);
 // delay de 1 segundo
```

```
// delay de 1 segundo
delay(1000);
// MASTER WRITES to SLAVE B
Wire.beginTransmission(SLAVE_B);
                    // begin transmission to Slave
  Wire.write("MASTER > SLAVE B\n");
                     // stop transmitting
  Wire.endTransmission();
1 // delay de 1 segundo
delay(1000);
 // MASTER READS from SLAVE B
Wire.requestFrom(SLAVE B, 17);
  while (Wire.available())
31 {
   char c = Wire.read();
   Serial.print(c);
// delay de 1 segundo
delay(1000);
11
12 }
13
```

```
3 // Ficha Prática 7 - Comunicação I2C (One Master-Multiple Slave)
5 // SLAVE B 3(Arduinjo Uno R3)
7 // Pinos I2C no Arduino Uno:
8 // - SDA: pino A4
9 // - SCL: pino A5
10 // - GND: Não esquecer de ligar o sinal de ground, entre os diversos dispositivos
//Nome:Eduardo Junqueira nº30241 / Jorge Pereira n30248
14 //Link:https://www.tinkercad.com/things/99C1HY0QTAX-copy-of-fp7ex2i2c/editel?sharecode=g00-ID300ixwQR2dVuoIaDRYobs0GQI1uYfxE-y7HdU&
18 #include <Wire.h>
19 #define SLAVE B 0x05
20 #define sensor A0
22 //Slave B mede a humidade através do sensor de umidade do solo
     Wire.begin(SLAVE_B); // Inicializa o módulo de hardware i2c como SLAVE com o endereço SLAVE_A
     Wire.onRequest(requestEvent);
     Wire.onReceive(receiveEvent);
     pinMode(LED BUILTIN, OUTPUT);
      digitalWrite(LED_BUILTIN, HIGH);
32 void loop() {
35 float readtemp(){
36 float volt=(analogRead(sensor) * 0.004882814);
    float temp = (volt - 0.5) * 100.0;
40 // requestEvent Handler that executes whenever data is requested by master
41 void requestEvent() {
    int c=(int)readtemp()+40;
   Serial.println(c);
     Wire.write(c);
47 // receiveEvent Handler that executes whenever data is received from the master
48 void receiveEvent(int numBytes) {
     digitalWrite(LED_BUILTIN,!digitalRead(LED_BUILTIN));
     while (Wire.available()) {
        char c = Wire.read();
         Serial.print(c);
```



# Resultados Continuação

```
//Nome:Eduardo Junqueira nº30241 / Jorge Pereira n30248
3 //Link:https://www.tinkercad.com/things/99C1HY0QTAX-copy-of-fp7ex2i2c/editel?sharecode=g00-ID300ixwQF
7 // Visor de 7 segmentos SLAVE A
#define dW digitalWrite
 #define A 13
3 #define B 12
4 #define C 11
5 #define D 10
6 #define E 9
 #define F 8
8 #define G 7
#define DP 6
 //Botões
 #define btn1 2
 #define btn2 3
 #define MAX_DISPLAY_NUM 9
 int NumG=0;
 void nointerrpts() {
  noInterrupts(); // Desactiva todas as Interrupts
 void num2Display(int num) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
   digitalWrite(E, LOW);
   digitalWrite(F, LOW);
   digitalWrite(G, LOW);
   digitalWrite(DP, LOW);
   // Verifica se o número está dentro do intervalo permitido.
   if (num >= 0 && num <= MAX_DISPLAY_NUM) {
    switch (num)
       digitalWrite(A, HIGH);
       digitalWrite(B, HIGH);
       digitalWrite(C, HIGH);
       digitalWrite(D, HIGH);
```

```
digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, LOW);
  delay(750);
  break;
case 1:
 digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  delay(750);
 break;
case 2:
 digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
 digitalWrite(G, HIGH);
 digitalWrite(E, HIGH);
 digitalWrite(D, HIGH);
delay(750);
 break;
case 3:
 digitalWrite(A, HIGH);
 digitalWrite(B, HIGH);
 digitalWrite(G, HIGH);
 digitalWrite(C, HIGH);
 digitalWrite(D, HIGH);
delay(750);
 break;
case 4:
 digitalWrite(F, HIGH);
 digitalWrite(G, HIGH);
  digitalWrite(B, HIGH);
 digitalWrite(C, HIGH);
delay(750);
 break;
 digitalWrite(A, HIGH);
  digitalWrite(F, HIGH);
 digitalWrite(G, HIGH);
 digitalWrite(C, HIGH);
 digitalWrite(D, HIGH);
delay(750);
 break;
 digitalWrite(A, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
 digitalWrite(E, HIGH);
delay(750);
 break;
 digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
```

```
digitalWrite(B, HIGH);
       digitalWrite(C, HIGH);
     delay(750);
       break;
       digitalWrite(A, HIGH);
       digitalWrite(B, HIGH);
       digitalWrite(C, HIGH);
       digitalWrite(D, HIGH);
       digitalWrite(E, HIGH);
       digitalWrite(F, HIGH);
       digitalWrite(G, HIGH);
     delay(750);
       break;
     case 9:
       digitalWrite(A, HIGH);
       digitalWrite(B, HIGH);
       digitalWrite(C, HIGH);
       digitalWrite(F, HIGH);
       digitalWrite(G, HIGH);
     delay(750);
       break;
void initTimer1Int(float Hz) {
 // Configuração da interrupção associada ao timer 1
       noInterrupts(); // Turn off global interrupts
 // Configurar o TIMER1 para operar em modo CTC (Clear Timer on Compare Match)
 TCCR1A = 0; // Modo normal de operação
 TCCR1B = 0; // Limpe os bits de configuração
 TCNT1 = 0; // Inicialize o contador com zero
 // Comparação (OCR1A) Para Hz calculo:
 // para obter uma interrupção a cada 1 segundo (1 Hz)
 // Frequência do TIMER1 = Clock / (Prescaler * (1 + OCR1A))
 // Clock = 16 MHz (frequência do Arduino Uno)
 // Prescaler = 256 (definido para gerar 1 Hz)
 // \text{ OCR1A} = (16000000 / (256 * 1)) - 1 = 62500
 if (Hz == 0.5) {
    OCR1A = 124999; //Calculado para 0.5Hz
   //OCR1A = (16000000 / (256 * 0.5)) - 1 = 124999
 } else if (Hz == 1 ) {
 OCR1A = 62499; // Calculado para 1Hz
        //OCR1A = (16000000 / (256 * 1)) - 1 = 62499
 } else if (Hz ==2 ) {
   OCR1A = 31249; // Calculado para 2Hz
   //OCR1A = (16000000 / (256 * 2)) - 1 = 31249
```





#### Resultados Final

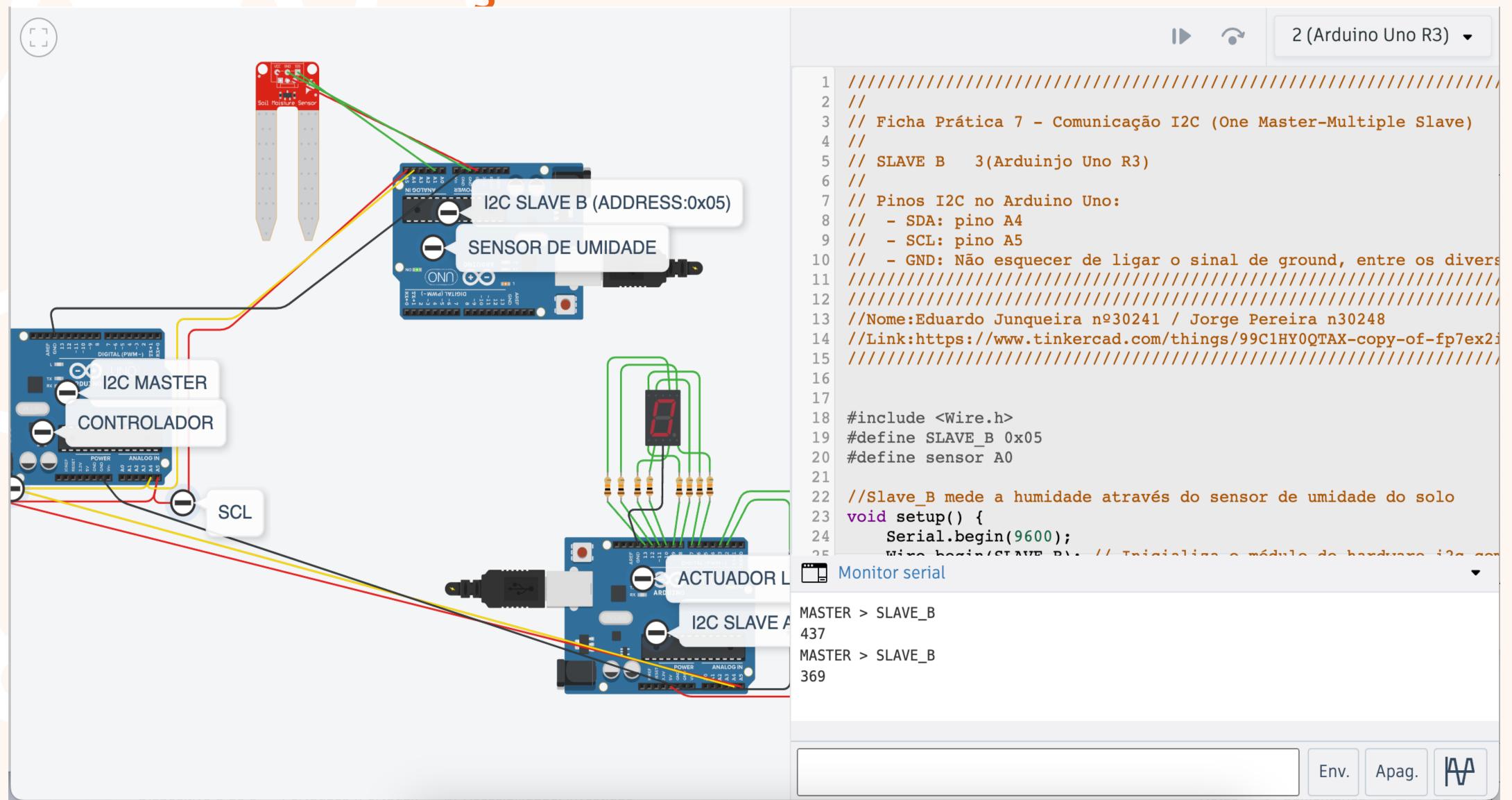
```
// Configurar o TIMER1 para operar com o prescaler 256
 // Isso define a frequência em 1 Hz
 TCCR1B |= (1 << WGM12); // Ative o modo CTC
 TCCR1B |= (1 << CS12); // Defina o prescaler 256
 TIMSK1 = (1 << OCIE1A); // Ative a interrupção de comparação A
                   // Turn on global interrupts
// Rotina de interrupção do TIMER1
//Esta rotina é configurada para ser chamada automaticamente pelo Timer1 quando dá Overflow, o que oc
ISR(TIMER1_COMPA_vect) {
 // (Re)inicialização do Timer 1
 TCNT1 = 0; //para todas as frequências
 // Atualização do valor do display
 num2Display(NumG);
 // Incrementa o contador e garante que ele varie ciclicamente entre 0 e 9
 NumG = (NumG + 1) % 10;
// Função Setup - Inicialização
// Função de Inicialização
void setup() {
                       // Serial Port Baud Rate = 9600 bps
   Serial.begin(9600);
 // Inicialize os pinos do display como saída
 pinMode(A, OUTPUT);
 pinMode(B, OUTPUT);
 pinMode(C, OUTPUT);
 pinMode(D, OUTPUT);
 pinMode(E, OUTPUT);
 pinMode(F, OUTPUT);
 pinMode(G, OUTPUT);
 pinMode(DP, OUTPUT);
 // Inicialização dos inputs
 pinMode(btn1, INPUT);
 pinMode(btn2, INPUT);
 noInterrupts(); // Desativa todas as Interrupções
void loop() {
```

```
) // Função Loop
void loop() {
  // Apenas a rotina de refrescamento do display é executada neste loop
   for (int num = 0; num <= MAX DISPLAY NUM; num++) {
    num2Display(num);
    initTimerlInt(1); // 1HZ
    //initTimer1Int(0.5); // 0.5Hz
    //initTimer1Int(2); // 2Hz
   // Quando o botão 1 é pressionado, reinicia o contador
   if (digitalRead(btn1) == HIGH) {
    NumG = 0;
    delay(750);
    // Quando for pressionado o botão 2, faz
    while (digitalRead(btn2) == HIGH)
      NumG = num;
      delay(750);
3 }
```

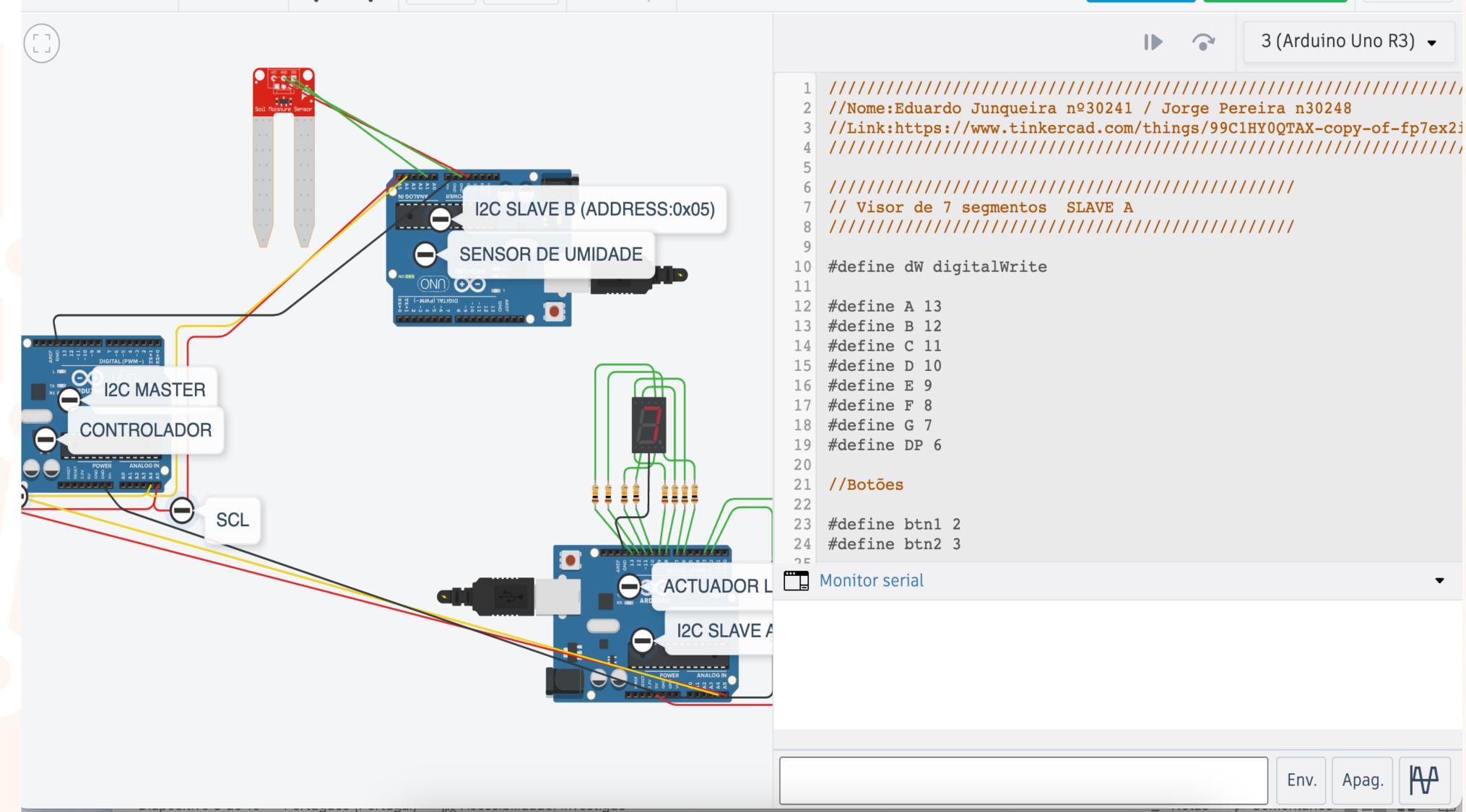
https://www.tinkercad.com/things/99C1HY0QTAX-copy-of-fp7ex2i2c/editel?sharecode=gO0-ID300ixwQR2dVuolaDRYobs0GQI1uYfxE-y7HdU&sharecode=5EtyftTD0uhpJwhqBXyiUcsJprZATIMpiZ9YUlsDpGY



#### Demonstração



# Demonstração Continuação



### Conclusões

-Podemos concluir que foi feito tudo tal como era pedido e que o objetivo do desafio foi cumprido.

-Este desafio foi bastante relevante para a junção de várias Matérias dadas em aula que foram demonstradas aqui neste desafio 2.





FIM