



# **Licenciatura em Engenharia Informática**

## **Tecnologia e Arquitetura de Computadores 2022/2023**

### **Trabalho Prático nº 2**

### **Digital Write/Read**

**Realizado em: 09/03/2023**

**Elaborado em: 09/03/2023**

**Grupo: 5**

**António Dinis - a2021157297**  
**Francisco Figueiras - a2021155919**  
**Mariana Magalhães - a2022147454**

# Índice

<b>1. Introdução .....</b>	<b>3</b>
<b>2. Métodos.....</b>	<b>3</b>
<b>3. Resultados.....</b>	<b>4</b>
<b>3.1. Exercício 1 .....</b>	<b>4</b>
<b>3.2. Exercício 2 .....</b>	<b>5</b>
<b>3.3. Exercício 3 .....</b>	<b>7</b>
<b>3.4. Exercício 4 .....</b>	<b>9</b>
<b>4. Discussão.....</b>	<b>11</b>
<b>5. Conclusão .....</b>	<b>11</b>
<b>6. Referências .....</b>	<b>12</b>

# I. Introdução

Este trabalho tem como objetivo fazer 4 exercícios, no primeiro será ligado um led e este vai ter a transição da direita para a esquerda, no segundo um contador ascendente de binário de 4 bits com o clock a 1 Hz, no terceiro irá ser feito um contador em binário, mas este só vai alterar bits com o clique de um botão e no quarto exercício, com a configuração do **pinMode()** e da função **INPUT\_PULLUP**, irá ser alterado o exercício juntamente com a remoção da resistência ligada ao botão.

## 2. Métodos

O trabalho foi realizado no decorrer das 3 horas de aula de **Tecnologia e Arquitetura de Computadores (TAC)** tendo sido utilizado o **Tinkercad** para projetar o circuito, o **Arduino IDE** para o desenvolvimento do código, todos estes programas foram desenvolvidos num computador com um processador AMD Ryzenn 7 5800H With Radeon Graphics e também foram usados os materiais disponíveis no laboratório para montagem e testagem dos circuitos.

Nome	Quantidade	Componente
UI	1	Arduino Uno R3
D1, D2, D3, D4	4	Red LED
R1, R2, R3, R4	4	1 kΩ Resistor

Tabela 1 - materiais do exercício 1 e 2

Nome	Quantidade	Componente
UI	1	Arduino Uno R3
D1, D2, D3, D4	4	Red LED
R1, R2, R3, R4, R5	5	1 kΩ Resistor
SI	1	Pushbutton

Tabela 2 - materiais do exercício 3

Nome	Quantidade	Componente
UI	1	Arduino Uno R3
D1, D2, D3, D4	4	Red LED
R1, R2, R3, R4, R5	5	1 kΩ Resistor
SI	1	Pushbutton

Tabela 3 - materiais do exercício 4

## 3. Resultados

### 3.1. Exercício I

O objetivo neste exercício é ir acendendo um led de cada vez da direita para a esquerda. Começamos por fazer o projeto do circuito no Tinkercad (Figura 1) passando em seguida para o código no Arduino (Figura 2) para em seguida procedermos para a construção do circuito (Figura 3).

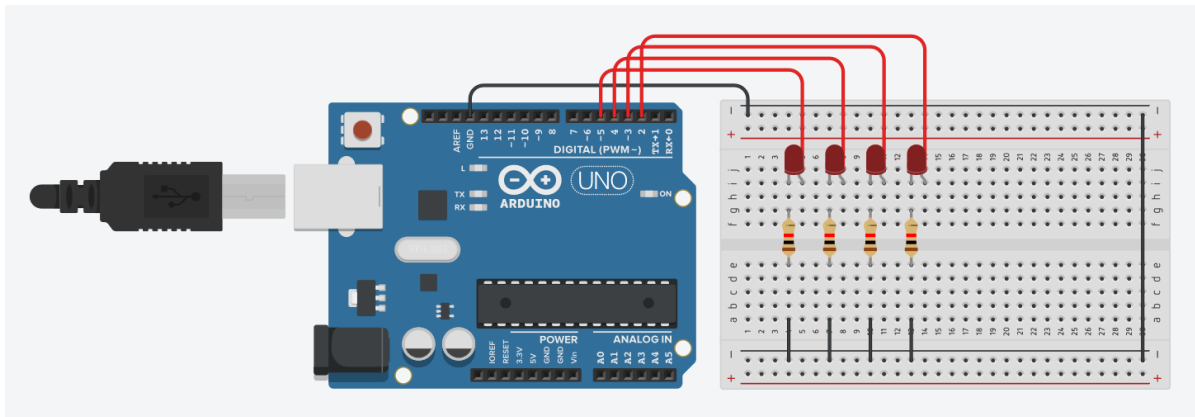


Figura 1 - digitalWrite\_1\_2

```

1 void setup() {
2   for (int i = 2; i < 6; i++)
3     pinMode(i, OUTPUT);
4 }
5
6 void loop() {
7   for (int i = 2; i < 6; i++) {
8     digitalWrite(i, HIGH);
9     delay(500);
10    digitalWrite(i, LOW);
11  }
12 }

```

Figura 2 - digitalWrite\_1(código)

O código ao lado representa o programa utilizado para configurar o arduino onde, na função **setup()**, usamos um *for loop* para iniciar os pinos 2 a 5 (configurados para saída) que posteriormente vão ser utilizados na função **loop()**.

Na função **loop()** os leds vão ser ligados consoante a sua ligação ao arduino vão acender desde o led mais a direita ( que está ligado ao pino 2) até ao que está mais à esquerda (pino 5).

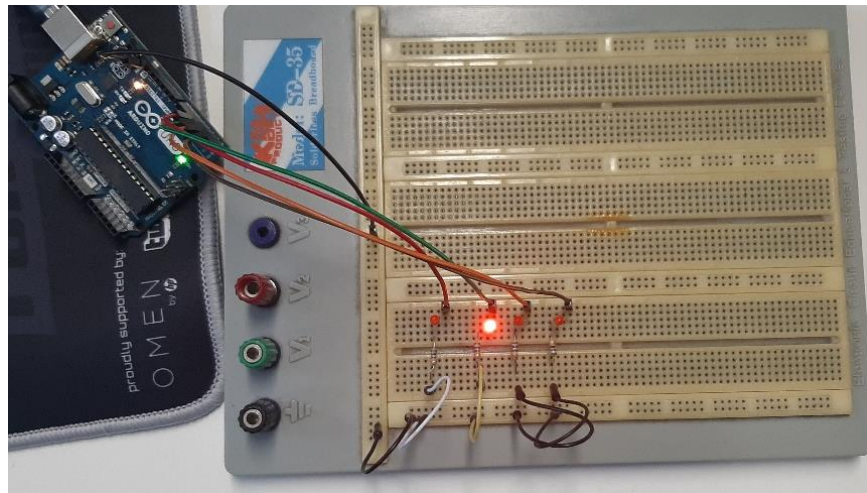


Figura 3 - digitalWrite\_1 breadboard

### 3.2. Exercício 2

Neste exercício o objetivo é simular um contador binário ascendente com quatro bits usando um clock de 1 Hz.

Reutilizando o projeto do circuito no Tinkercad no exercício 1 e a sua respetiva montagem (figura 5) passamos para elaboração do código no Arduino (Figura 4) para isso, utilizamos a tabela 4 como base.

L1	L2	L3	L4
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Tabela 4 - tabela binaria de 4 bits

```
1 void setup() {  
2   for (int i = 2; i < 6; i++)  
3     pinMode(i, OUTPUT);  
4 }  
5  
6 void loop() {  
7   for (int i = 0; i < 16; i++) {  
8     digitalWrite(2, i&1);  
9     digitalWrite(3, i&2);  
10    digitalWrite(4, i&4);  
11    digitalWrite(5, i&8);  
12    delay(1000);  
13  }  
14 }
```

Utilizando a função **setup()** criada anteriormente e reformulando a função loop ficamos com o código ao lado, ou seja, o pino 2 vai estar a alterar a cada pulso do clock, o pino 3 a cada dois pulsos, o pino 4 de quatro em quatro e o pino 5 de oito em oito.

Figura 4 - digitalWrite\_2 código

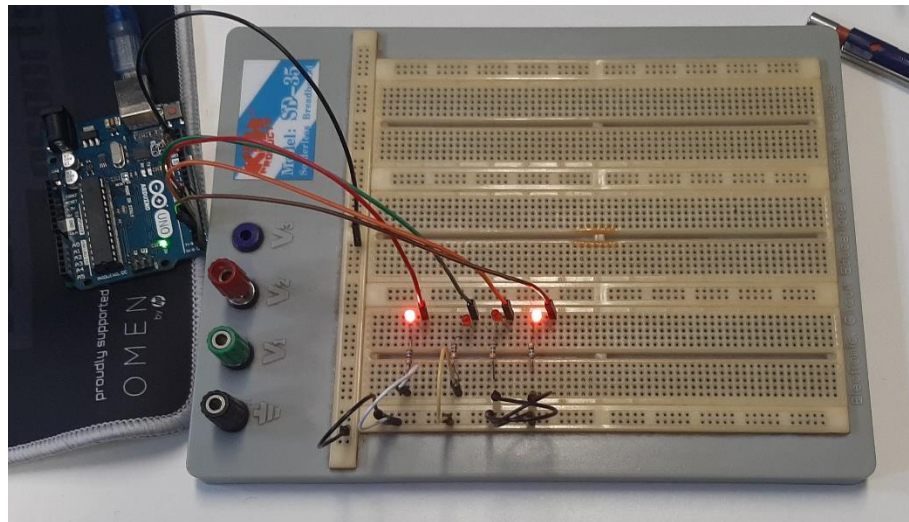


Figura 5 - digitalWrite\_2 breadboard

### 3.3. Exercício 3

Neste exercício o objetivo é o mesmo do exercício 2, mas desta vez o contador só é incrementado cada vez que o botão for pressionado.

Começamos por fazer o projeto do circuito no Tinkercad (Figura 6) passando em seguida para o código no Arduino (Figura 8) para, em seguida, procedermos para a construção do mesmo (Figura 7).

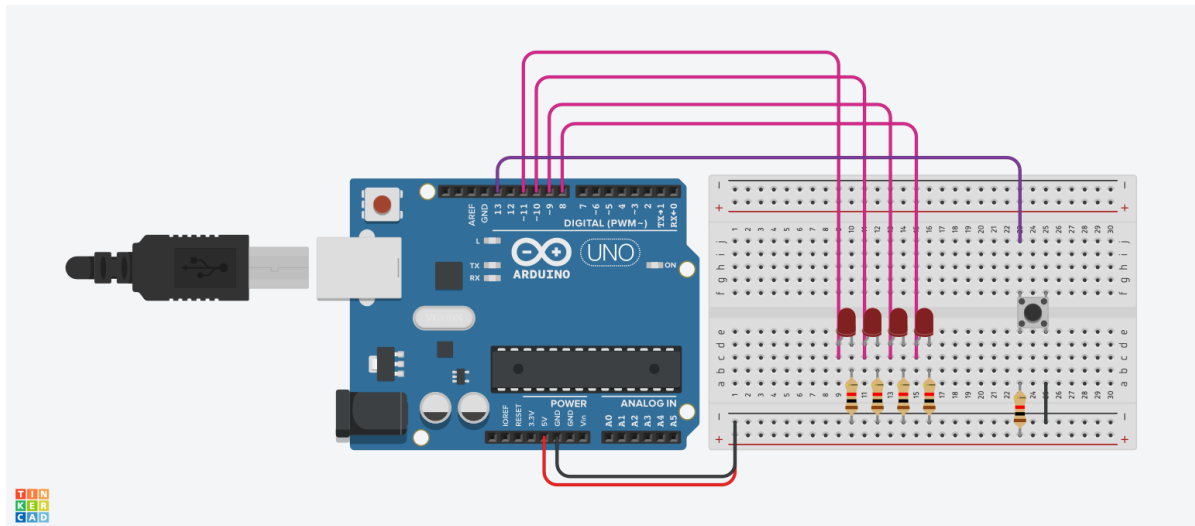


Figura 6 - digitalRead\_3

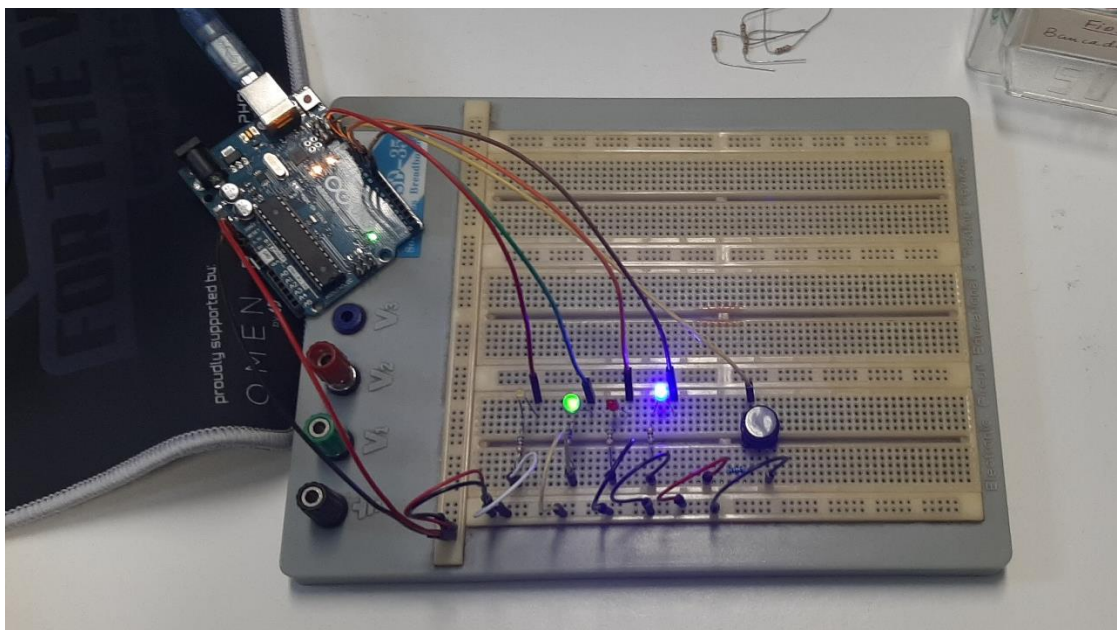


Figura 7 - digitalRead\_3 breadboard



```
1  int buttonState;
2  int counter = 0;
3
4  void setup() {
5      for (int i = 8; i < 12; i++)
6          pinMode(i, OUTPUT);
7      pinMode(13, INPUT);
8      Serial.begin(9600);
9  }
10
11 void loop() {
12
13     buttonState = digitalRead(13);
14     Serial.println("...");
15
16     if (buttonState == LOW) {
17         delay(500);
18         counter++;
19     } else if (counter > 15) {
20         counter = 0;
21     }
22
23     digitalWrite(8, counter&1);
24     digitalWrite(9, counter&2);
25     digitalWrite(10, counter&4);
26     digitalWrite(11, counter&8);
27
28     Serial.println(counter);
29 }
```

O código ao lado representa o programa utilizado para configurar o Arduino onde na função **setup()** prepara do pino 8 ao 11 configurados para saída e onde define o pino 13 (pino a que está ligado o botão) como um *input*, para que depois na função **loop()**, ao ser pressionado o botão, o contador a ele associado aumente o seu valor em 1 até atingir o máximo de 15, voltando de seguida a zero substituindo assim o *for loop* usado no exercício anterior.

Figura 8 - digitalRead\_3 código



### 3.4. Exercício 4

O objetivo neste exercício é igual ao anterior, mas, desta vez, foi necessário usar uma função diferente chamada **pinMode() (INPUT\_PULLUP)** e sem usar a resistência conectada ao botão de pressão.

Reutilizando o projeto do circuito no Tinkercad no exercício 3 e a sua respetiva montagem, mas retirando-lhe a resistência que estava conectada ao botão de pressão (figura 9 e 10) passámos para elaboração do código no Arduino (Figura 11).

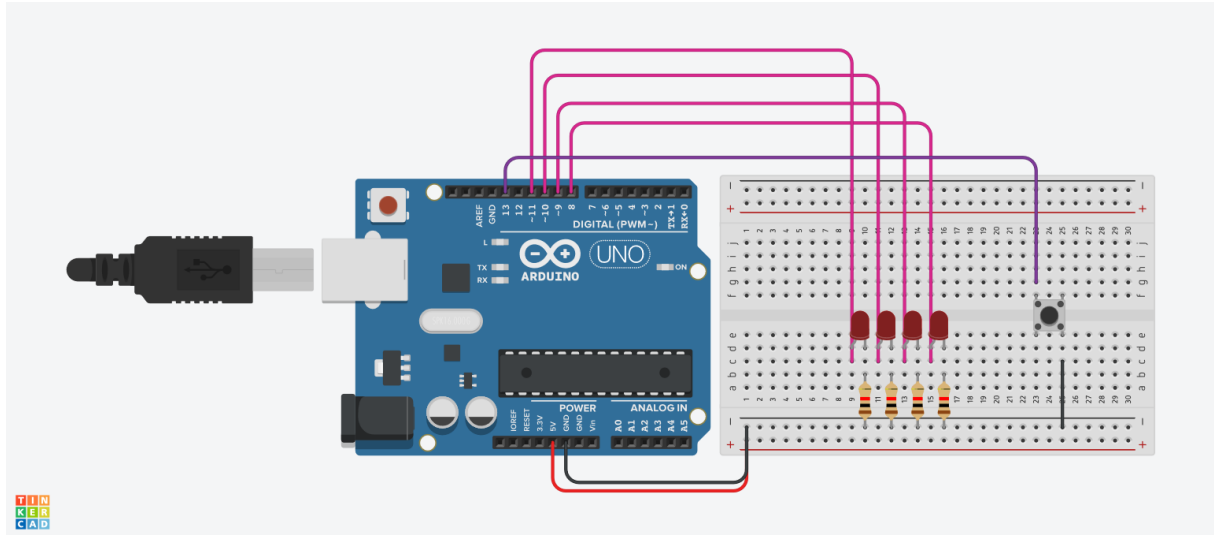


Figura 9 - digitalRead\_4

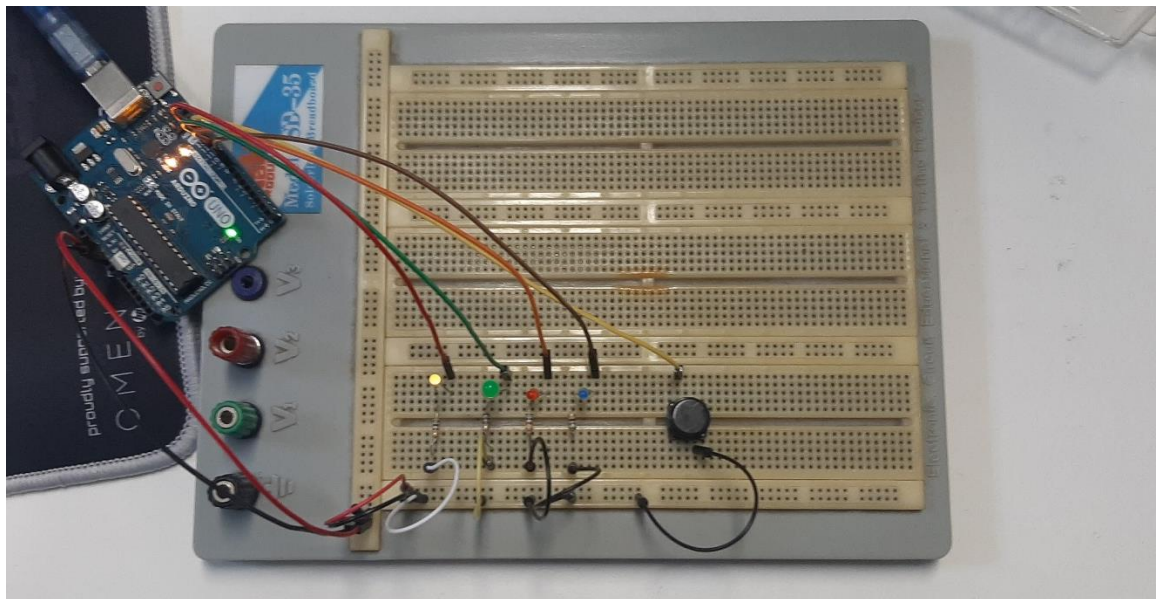


Figura 10 - digitalRead\_4 breadboard

```
1  int buttonState;
2  int counter = 0;
3
4  void setup() {
5      for (int i = 8; i < 12; i++)
6          pinMode(i, OUTPUT);
7      pinMode(13, INPUT_PULLUP);
8      Serial.begin(9600);
9  }
10
11 void loop() {
12
13     buttonState = digitalRead(13);
14     Serial.println("...");
15
16     if (buttonState == LOW) {
17         counter++;
18         delay(500);
19     } else if (counter > 15) {
20         counter = 0;
21     }
22
23     digitalWrite(8, counter&1);
24     digitalWrite(9, counter&2);
25     digitalWrite(10, counter&4);
26     digitalWrite(11, counter&8);
27
28     Serial.println(counter);
29 }
```

Este código em comparação ao exercício anterior só tem uma diferença na linha 7 em vez de *input* implementamos a função **INPUT\_PULLUP**.

Figura 11 - digitalRead\_4 código

## 4. Discussão

Nos exercícios 3 e 4 foi onde estive a maior dificuldade uma vez que no exercício 4 tínhamos de implementar uma nova função chamada **INPUT\_PULLUP** e após alguma pesquisa e ao fim de experimentarmos, percebemos que no fim das contas existe um resistor interno implementado na placa Arduino, este resistor tem um valor estimado entre  $20k\Omega$  e  $50k\Omega$ , por tanto ele fará exatamente a função que a resistência faz no exercício 3.

## 5. Conclusão

Assim sendo podemos concluir que cumprimos os exercícios até ao exercício 4 e que apesar de termos tido alguma dificuldade nos exercícios 3 e 4 conseguimos ultrapassá-los.

## 6. Referências

[https://roboticsbackend.com/arduino-input\\_pullup-pinmode/](https://roboticsbackend.com/arduino-input_pullup-pinmode/)

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Debounce>

[https://docs.arduino.cc/tutorials/generic/digital-input-](https://docs.arduino.cc/tutorials/generic/digital-input-pullup?queryID=dafb20ba8e34f02d6da8444c5a2b123f)

[pullup?queryID=dafb20ba8e34f02d6da8444c5a2b123f](https://docs.arduino.cc/tutorials/generic/digital-input-pullup?queryID=dafb20ba8e34f02d6da8444c5a2b123f)

Vídeos:

<https://www.youtube.com/playlist?list=PLweUCI9fZUoZ3C0hCDGSrBOF5MWIkjBOq>