



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS
GERAIS**

Instituto de Ciências Exatas e de Informática

Exercício prático Arquitetura de Computadores - II

Vitor Costa Oliveira Rolla

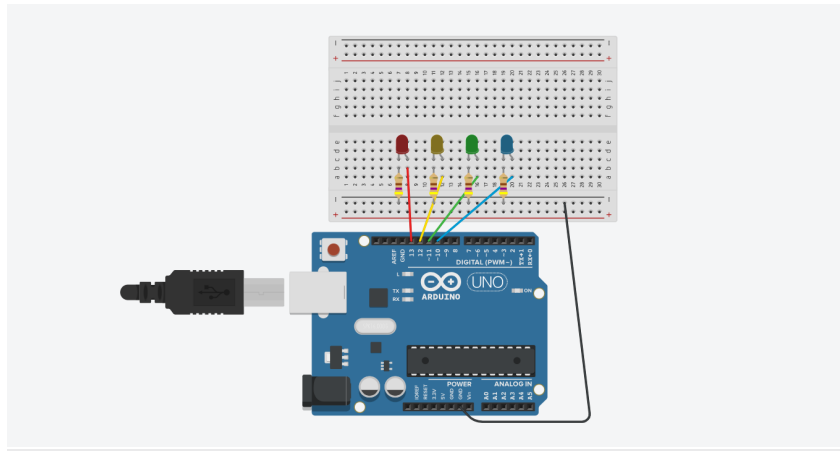
Pontifícia Universidade Católica de Minas Gerais

09 de junho de 2024

Resumo

Este exercício prático tem como objetivo aplicar os conhecimentos adquiridos na disciplina de Arquitetura de Computadores, por meio da simulação de circuitos digitais utilizando o tinkercad. Mais precisamente, serão construídos um display de leds feito para simular um "sinal de trânsito" e os dados de uma ULA de 1 bit

Exercício 1 - Ciclos de luz



```
/*
  Programa: Piscar LEDs com sequência
  Azul pisca de 1 em 1 segundo.
  - Nos 3 primeiros ciclos: Vermelho ligado
  - Nos 4 ciclos seguintes: Verde ligado
  - Nos 2 ciclos seguintes: Amarelo ligado
  - Repete
*/

// Definição dos pinos
int ledVermelho = 13;
int ledVerde    = 12;
int ledAmarelo  = 11;
int ledAzul     = 10;

int ciclo = 0; // contador de ciclos do azul

void setup() {
  pinMode(ledVermelho, OUTPUT);
  pinMode(ledVerde, OUTPUT);
  pinMode(ledAmarelo, OUTPUT);
  pinMode(ledAzul, OUTPUT);
}

void loop() {
  ciclo++; // Avança ciclo

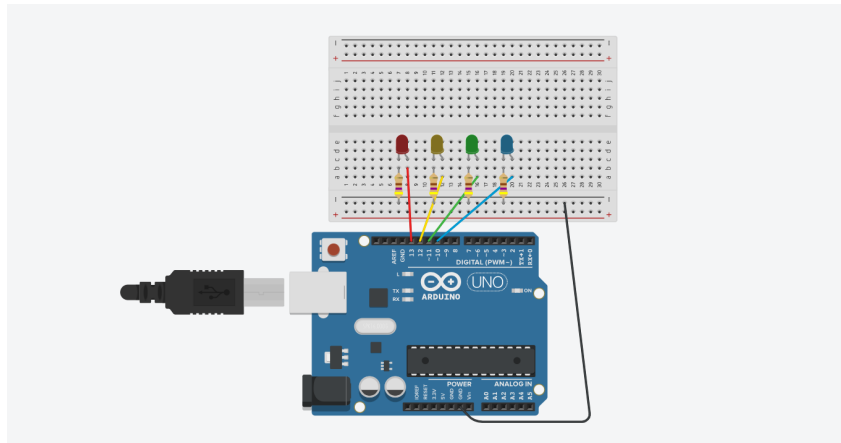
  // Reset do contador depois de 9 ciclos (3 + 4 + 2)
  if (ciclo > 9) {
    ciclo = 1;
  }

  // Determina qual LED ficará aceso junto com o Azul
  if (ciclo >= 1 && ciclo <= 3) { // 3 ciclos com Vermelho
    digitalWrite(ledVermelho, HIGH);
    digitalWrite(ledVerde, LOW);
    digitalWrite(ledAmarelo, LOW);
  }
  else if (ciclo >= 4 && ciclo <= 7) { // 4 ciclos com Verde
    digitalWrite(ledVermelho, LOW);
    digitalWrite(ledVerde, HIGH);
    digitalWrite(ledAmarelo, LOW);
  }
  else if (ciclo >= 8 && ciclo <= 9) { // 2 ciclos com Amarelo
    digitalWrite(ledVermelho, LOW);
    digitalWrite(ledVerde, LOW);
    digitalWrite(ledAmarelo, HIGH);
  }

  // Piscar o azul (1 segundo ligado / 1 segundo desligado)
  digitalWrite(ledAzul, HIGH);
  delay(1000);
  digitalWrite(ledAzul, LOW);
  delay(1000);
}
```

Este é um display de leds utilizado para simular um processo de ciclos. O programa que será exibido executa a seguinte instrução: O led Azul deverá sempre piscar de um em um segundo. Durante os primeiros 3 ciclos de azul ligado, apenas o vermelho deverá estar ligado, em seguida durante 4 ciclos de azul ligado, apenas o verde ficará ligado e finalmente durante 2 ciclos de azul ligado, apenas o amarelo ficará ligado. O ciclo continua indefinidamente.

Exercício 2 - ULA de 1 bit



Através da comunicação serial três valores deverão ser passados ao Arduino. O valor de a, b e o OP. Code. Ao passarmos os valores, deveremos passar os três números de uma vez, observe os exemplos a seguir:

- se enviarmos 100 significa que $a=1$, $b=0$ e queremos a operação 0 ou $\text{and}(a,b)$, a saída será 0 e vai1 também 0.
- se enviarmos 012 significa que $a=0$, $b=1$ e queremos a operação 2 ou $\text{not}(a)$, a saída será 1 e vai1 também 0.
- se enviarmos 113 significa que $a=1$, $b=1$ e queremos a operação 3 ou $\text{soma}(a,b)$, a saída será 0 e vai1 será 1.

Aqui está configuração para mostrar as entradas e saídas:

O valor de a no led vermelho (saída 13)

O valor de b no led amarelo (saída 12)

O valor da saída da ULA no led verde (saída 11)

O valor de vai1 no led azul (saída 10)

```

/*
  ULA de 1 bit com Arduino
  Comunicação Serial: envia 3 dígitos -> a b op
  Exemplo: 113 -> a=1, b=1, op=3 (soma)
*/

int ledA = 13;    // vermelho
int ledB = 12;    // amarelo
int ledS = 11;    // verde (resultado)
int ledC = 10;    // azul (vair)

void setup() {
  pinMode(ledA, OUTPUT);
  pinMode(ledB, OUTPUT);
  pinMode(ledS, OUTPUT);
  pinMode(ledC, OUTPUT);

  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {

    // Entrada
    String entrada = Serial.readString();
    int a = entrada.charAt(0) - '0';
    int b = entrada.charAt(1) - '0';
    int opCode = entrada.charAt(2) - '0';

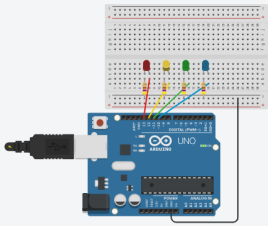
    // Saída
    int cout = 0;
    int s = 0;

    if(opCode == 0){
      s = a & b;
    } else if(opCode == 1){
      s = a | b;
    } else if(opCode == 2){
      s = !a;
    } else if(opCode == 3){
      s = (!a & b) | (a & !b);
      cout = a & b;
    }

    digitalWrite(13, a);
    digitalWrite(12, b);
    digitalWrite(11, s);
    digitalWrite(10, cout);

    // Mostra dados no Terminal Serial
    Serial.print("O valor de a no led vermelho: "); Serial.println(a);
    Serial.print("O valor de b no led amarelo: "); Serial.println(b);
    Serial.print("Op = "); Serial.println(opCode);
    Serial.print("O valor da saída da ULA no led verde: "); Serial.println(s);
    Serial.print("O valor de vair no led azul: "); Serial.println(cout);
    Serial.println("-----")
  }
}

```



```

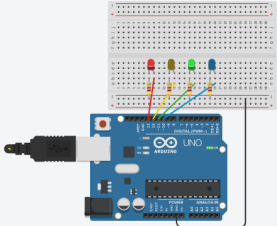
17 Serial.begin(9600);
18 }
19
20 void loop() {
21   if (Serial.available() > 0) {
22
23     // Entrada
24     String entrada = Serial.readString();
25     int a = entrada.charAt(0) - '0';
26     int b = entrada.charAt(1) - '0';
27     int opCode = entrada.charAt(2) - '0';
28
29     // Saida
30     int cout = 0;
31     int s = 0;
32
33     if(opCode == 0){
34       s = a & b;
35     } else if(opCode == 1){
36       s = a | b;
37     } else if(opCode == 2){
38       s = a ^ b;
39     } else if(opCode == 3){
40       s = !a;
41     } else if(opCode == 3){
42       s = !(a & b) || (a & !b);
43       cout = a & b;
44     }
45
46     digitalWrite(13, a);
47     digitalWrite(12, b);
48     digitalWrite(11, s);
49     digitalWrite(10, cout);
50
51   }
52 }
53

```

Monitor serial

O valor de a no led vermelho: 0
 O valor de b no led amarelo: 1
 Op = 0
 O valor de saída de ULA no led verde: 0
 O valor de vaill no led azul: 0

Env Apag



```

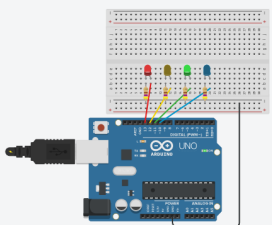
19
20
21 void loop() {
22   if (Serial.available() > 0) {
23
24     // Entrada
25     String entrada = Serial.readString();
26     int a = entrada.charAt(0) - '0';
27     int b = entrada.charAt(1) - '0';
28     int opCode = entrada.charAt(2) - '0';
29
30     // Saida
31     int cout = 0;
32     int s = 0;
33
34     if(opCode == 0){
35       s = a & b;
36     } else if(opCode == 1){
37       s = a | b;
38     } else if(opCode == 2){
39       s = a ^ b;
40     } else if(opCode == 3){
41       s = !a;
42     } else if(opCode == 3){
43       s = !(a & b) || (a & !b);
44       cout = a & b;
45     }
46
47     digitalWrite(13, a);
48     digitalWrite(12, b);
49     digitalWrite(11, s);
50     digitalWrite(10, cout);
51
52   }
53

```

Monitor serial

O valor de a no led vermelho: 1
 O valor de b no led amarelo: 0
 Op = 1
 O valor de saída de ULA no led verde: 1
 O valor de vaill no led azul: 0

Env Apag



```

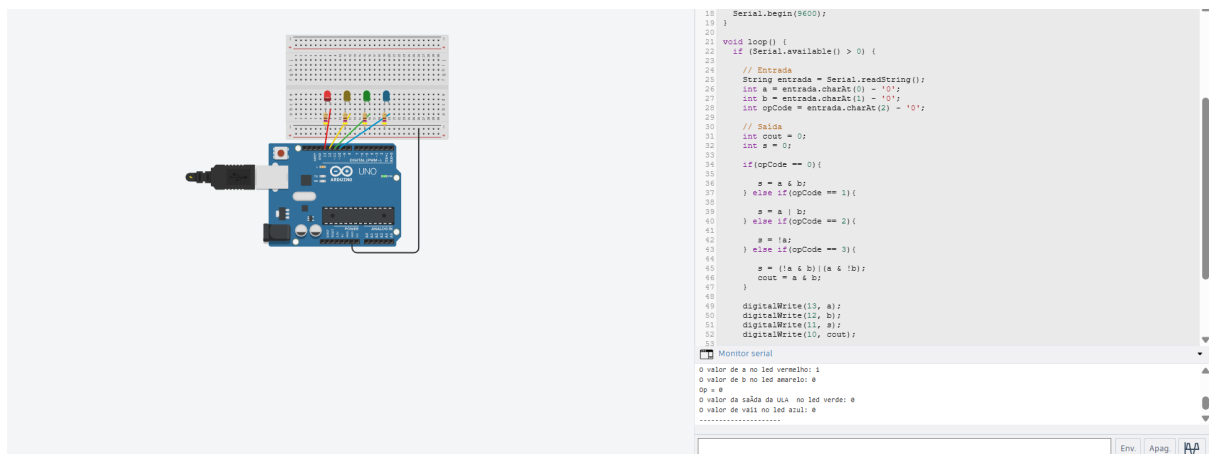
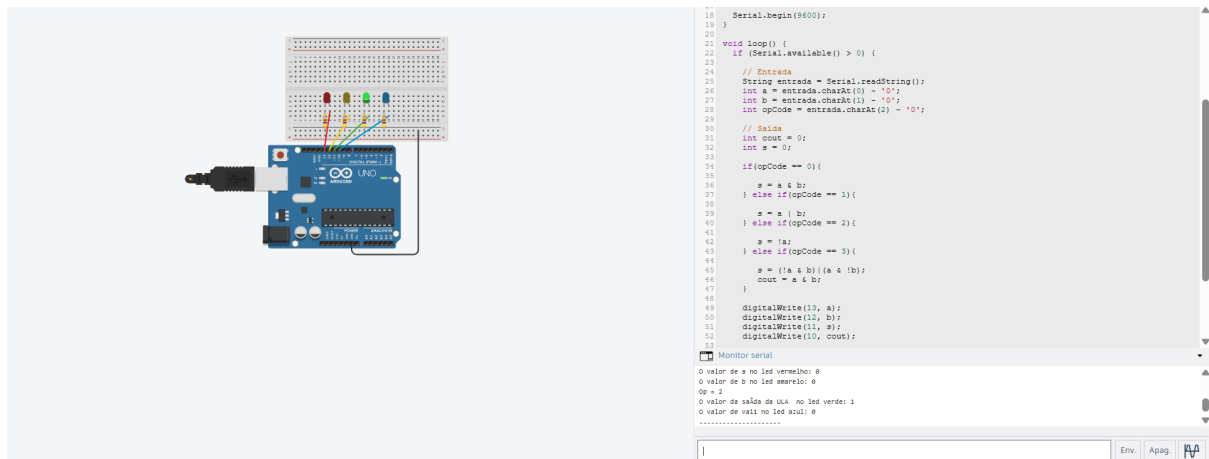
17 Serial.begin(9600);
18 }
19
20 void loop() {
21   if (Serial.available() > 0) {
22
23     // Entrada
24     String entrada = Serial.readString();
25     int a = entrada.charAt(0) - '0';
26     int b = entrada.charAt(1) - '0';
27     int opCode = entrada.charAt(2) - '0';
28
29     // Saida
30     int cout = 0;
31     int s = 0;
32
33     if(opCode == 0){
34       s = a & b;
35     } else if(opCode == 1){
36       s = a | b;
37     } else if(opCode == 2){
38       s = a ^ b;
39     } else if(opCode == 3){
40       s = !a;
41     } else if(opCode == 3){
42       s = !(a & b) || (a & !b);
43       cout = a & b;
44     }
45
46     digitalWrite(13, a);
47     digitalWrite(12, b);
48     digitalWrite(11, s);
49     digitalWrite(10, cout);
50
51   }
52 }
53

```

Monitor serial

O valor de a no led vermelho: 1
 O valor de b no led amarelo: 0
 Op = 3
 O valor de saída de ULA no led verde: 1
 O valor de vaill no led azul: 0

Env Apag



Instrução realizada	Binário (A,B,Op.code)	Valor em Hexa (0x ...)	Resultado em binário
AND(A,B)	0 1 00	0x4	0
OR(A,B)	1 0 01	0x9	1
SOMA(A,B)	1 0 11	0xB	1
NOT(A)	0 0 10	0x2	1
AND(B,A)	1 0 00	0x8	0