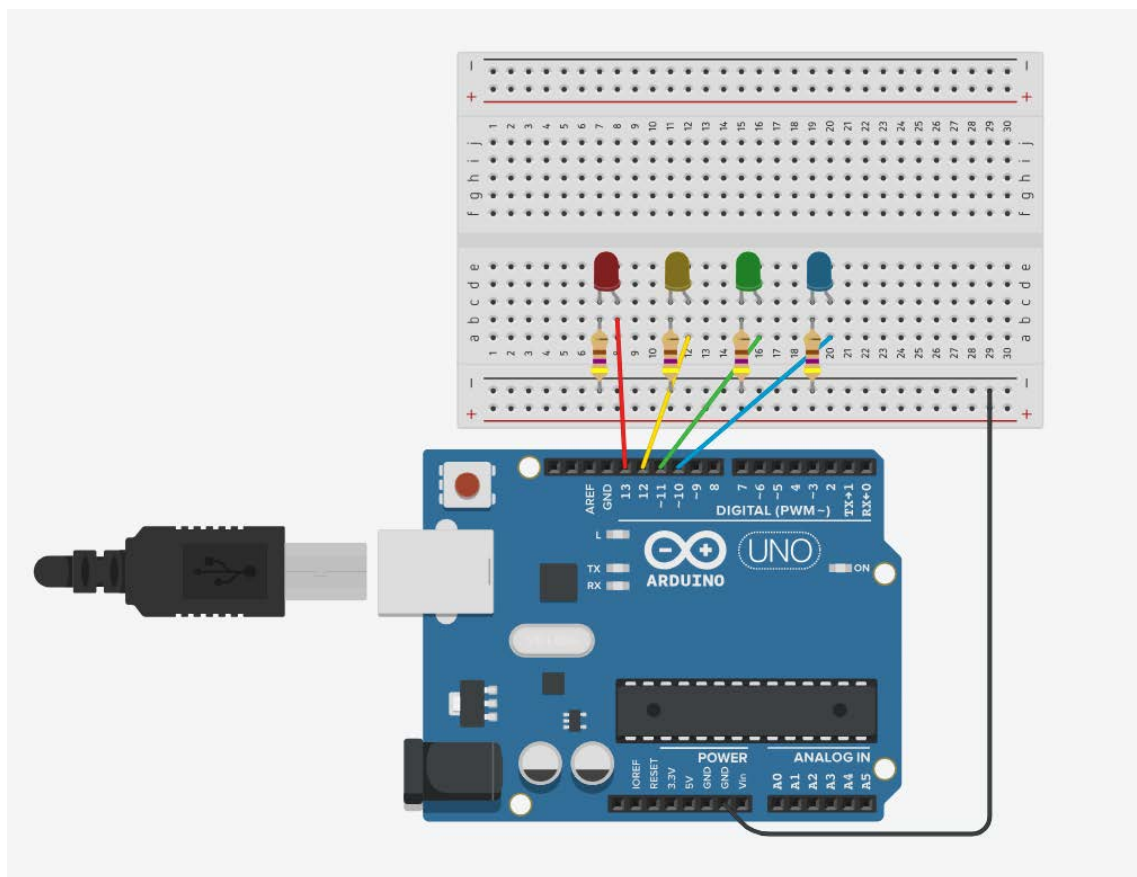


Arquitetura de Computadores 2
Exercício Prático 03 – Prof. Romanelli
812839 – Vinícius Miranda de Araújo

Exercício 1

Circuito

Figura 1 - Circuito Tinkercad



Programa

Figura 2 - Programa de um Semáforo Temporizado para Arduino

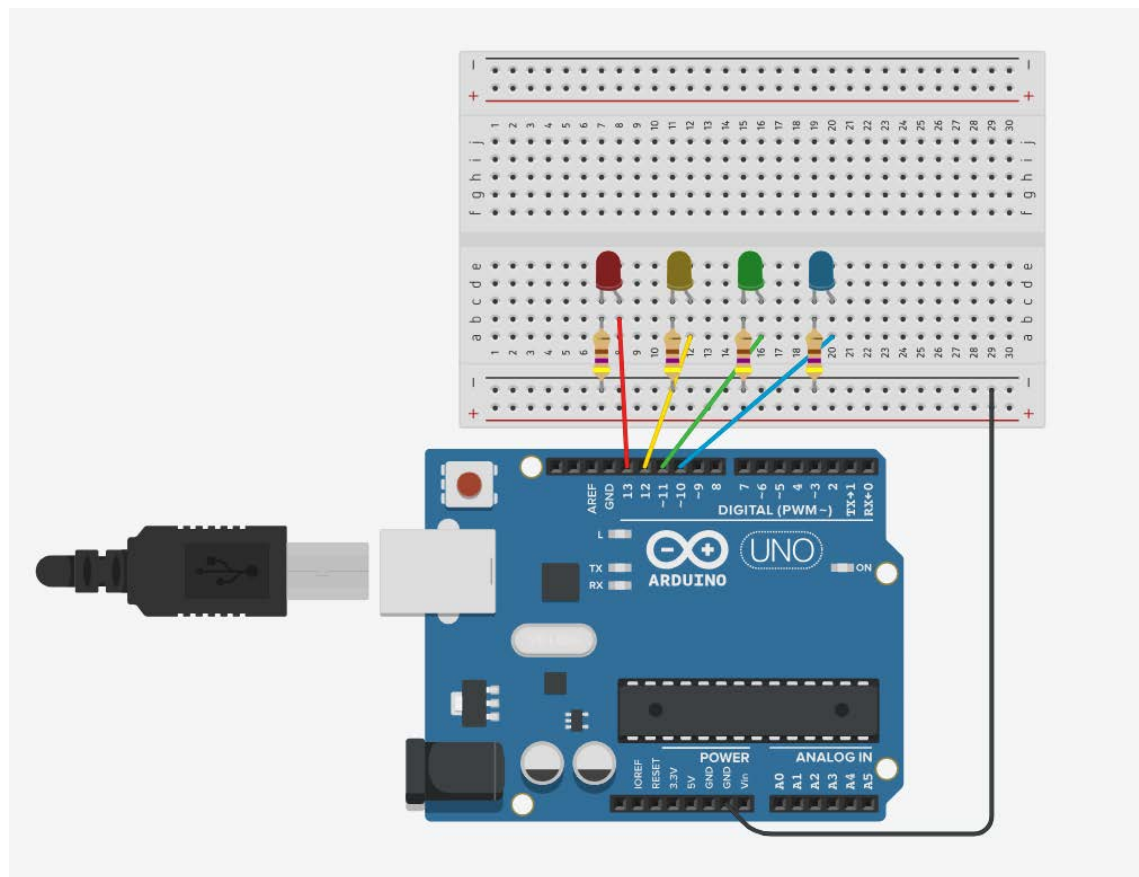
```
1 // definicao de pinos
2 int ledBlue   = 10;
3 int ledGreen  = 11;
4 int ledYellow = 12;
5 int ledRed    = 13;
6
7 // definicao de variaveis
8 const int time    = 1000;
9 const int redClk   = 3;
10 const int greenClk = 4;
11 const int yellowClk = 2;
12
13 // prototipo de funcoes
14 void turnOff ( int led );
15 void turnOn  ( int led );
16
17 void setup ( )
18 {
19     pinMode( ledBlue   , OUTPUT );
20     pinMode( ledGreen  , OUTPUT );
21     pinMode( ledYellow , OUTPUT );
22     pinMode( ledRed    , OUTPUT );
23 } // end setup ( )
24
25 void loop ( )
26 {
27     for ( int i = 0; i < redClk; i++ ) {
28         turnOn( ledBlue );
29         turnOn( ledRed );
30         delay( time );
31         turnOff( ledBlue );
32         delay( time );
33     }
34     turnOff( ledRed );
35
36     for ( int i = 0; i < greenClk; i++ ) {
37         turnOn( ledBlue );
38         turnOn( ledGreen );
39         delay( time );
40         turnOff( ledBlue );
41         delay( time );
42     }
43     turnOff( ledGreen );
44
45     for ( int i = 0; i < yellowClk; i++ ) {
46         turnOn( ledBlue );
47         turnOn( ledYellow );
48         delay( time );
49         turnOff( ledBlue );
50         delay( time );
51     }
52     turnOff( ledYellow );
53 } // end loop ( )
54
55 void turnOff ( int led ) {
56     digitalWrite( led, LOW );
57 } // end turnOff ( )
58
59 void turnOn ( int led ) {
60     digitalWrite( led, HIGH );
61 } // end turnOn ( )
```

Exercício 2

Instrução Realizada	Binário (A, B, Op.Code)	Valor em Hexa	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)	0 1 00	0x4	0

Circuito

Figura 3 - Circuito Tinkercad



Programa

Figura 4 - Programa de uma ULA de 1 bit para Arduino - 1ª Parte

```
1 // definicao de pinos
2 int ledA      = 13;
3 int ledB      = 12;
4 int ledOutput = 11;
5 int ledCarry  = 10;
6
7 // prototipo de funcoes
8 char read      ( );
9 void turnOnOff ( int led, int value );
10 int gate_xor   ( int a, int b );
11 int gate_or    ( int a, int b );
12 int gate_and   ( int a, int b );
13 int gate_not   ( int a );
14
15 void setup ( )
16 {
17     Serial.begin( 9600 );
18     pinMode( ledA , OUTPUT );
19     pinMode( ledB , OUTPUT );
20     pinMode( ledOutput , OUTPUT );
21     pinMode( ledCarry , OUTPUT );
22 } // end setup ( )
23
24 void loop ( )
25 {
26     if( Serial.available( ) >= 3 ) {
27         int a = read( );
28         int b = read( );
29         int op = read( );
30
31         int output = 0;
32         int carry = 0;
33
34         switch ( op ) {
35             case 0:
36                 output = gate_and( a, b );
37                 break;
38             case 1:
39                 output = gate_or( a, b );
40                 break;
41             case 2:
42                 output = gate_not( a );
43                 break;
44             case 3:
45                 output = gate_xor( a, b );
46                 carry = gate_and( a, b );
47                 break;
48             default:
49                 Serial.print( "Operacao Invalida!" );
50                 break;
51         } // end switch
52
53         turnOnOff( ledA , a );
54         turnOnOff( ledB , b );
55         turnOnOff( ledOutput, output );
56         turnOnOff( ledCarry , carry );
57     } // end if
58 } // end loop ( )
59
```

Figura 5 - Programa de uma ULA de 1 bit para Arduino - 2ª Parte

```
60 char read ( ) {  
61     return ( Serial.read( ) - '0' );  
62 } // end read_int ( )  
63  
64 void turnOnOff ( int led, int value ) {  
65     digitalWrite( led, value );  
66 } // end turnOnOff  
67  
68 int gate_xor ( int a, int b ) {  
69     return ( a^b );  
70 } // end gate_xor ( )  
71  
72 int gate_or ( int a, int b ) {  
73     return ( a|b );  
74 } // end gate_or ( )  
75  
76 int gate_and ( int a, int b ) {  
77     return ( a&b );  
78 } // end gate_and ( )  
79  
80 int gate_not ( int a ) {  
81     return ( ~a );  
82 } // end gate_not ( )  
83
```

Testes

Figura 6 – ULA 1 bit – Teste 0 1 0

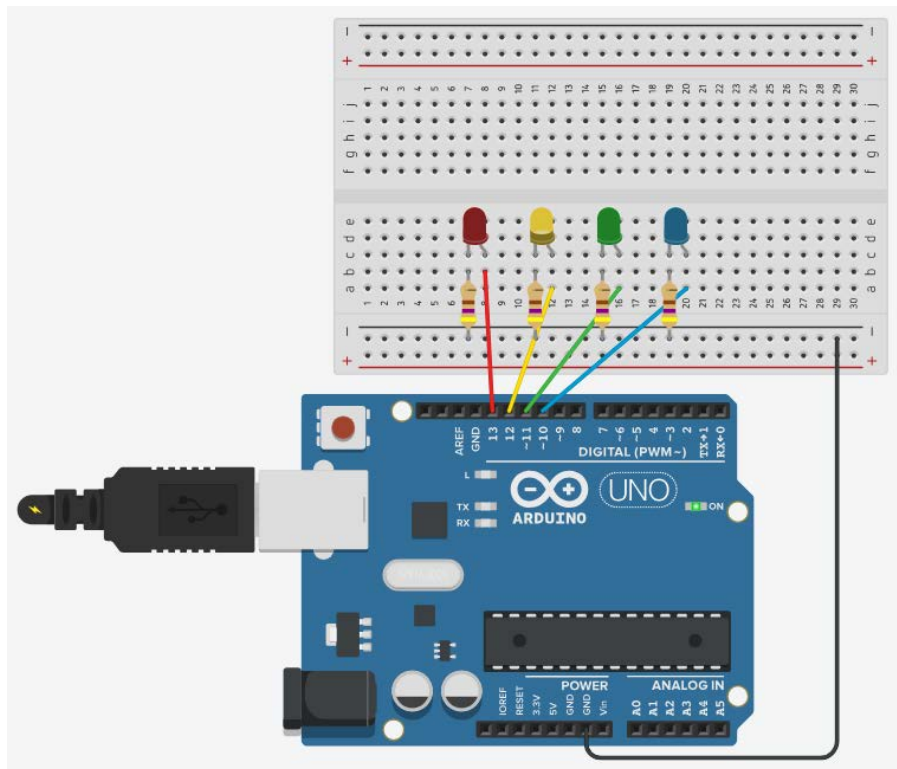


Figura 7 - ULA 1 bit – Teste 1 0 1

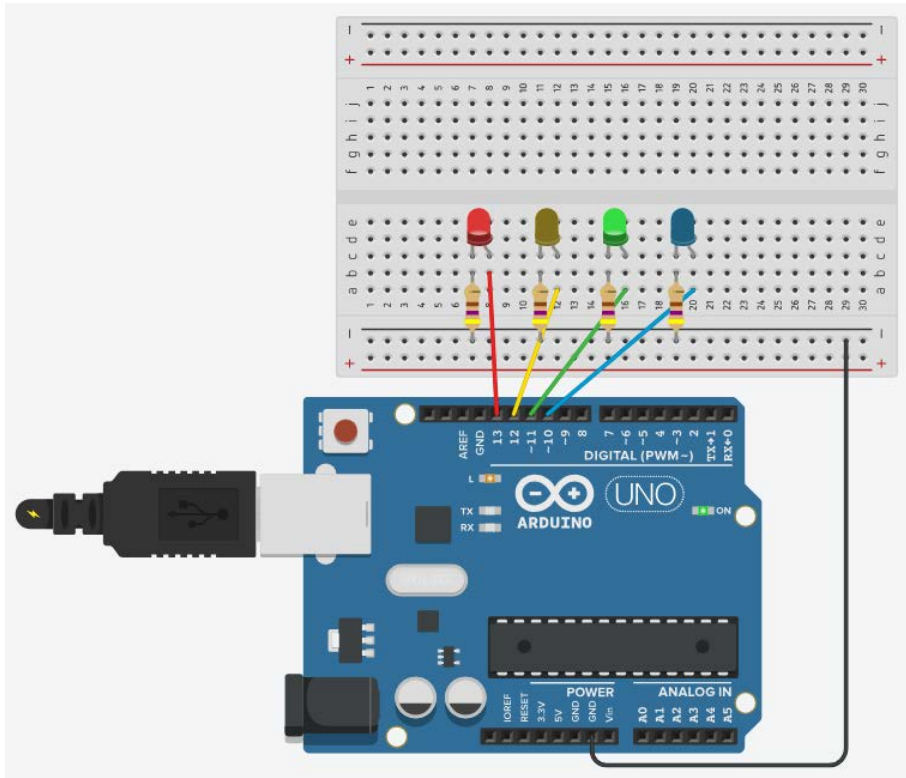


Figura 8 - ULA 1 bit – Teste 1 0 3

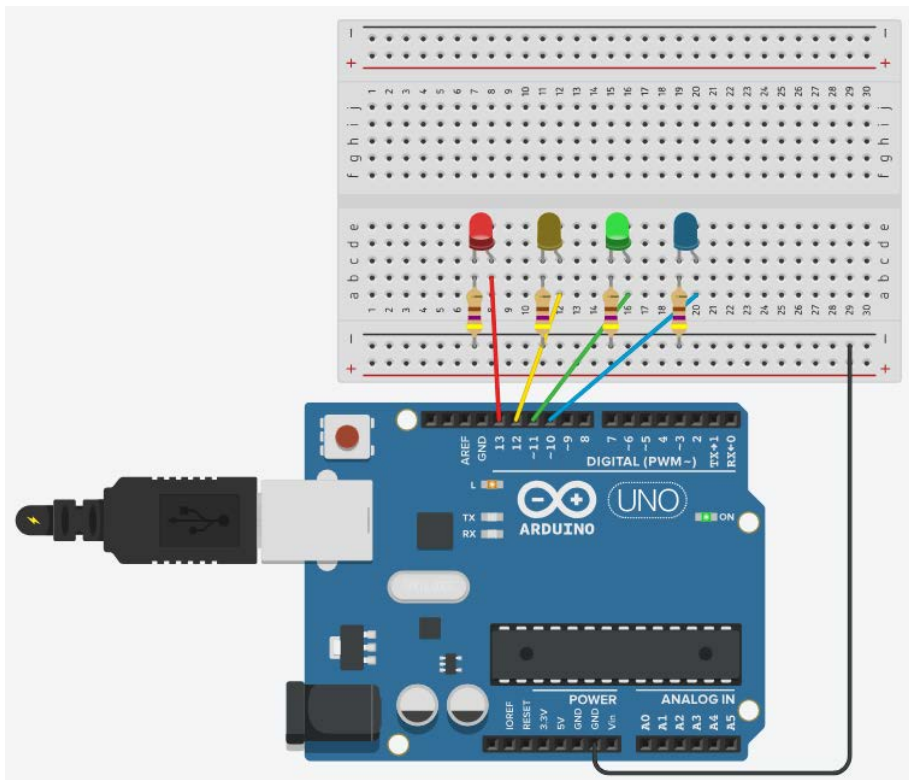


Figura 9 - ULA 1 bit – Teste 0 0 2

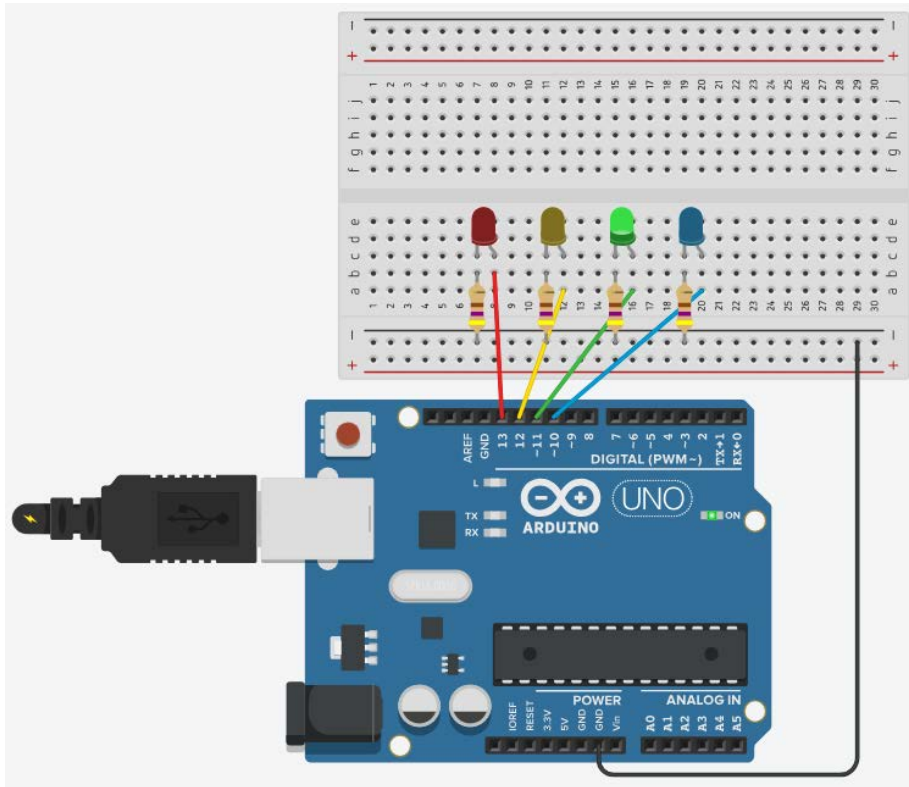
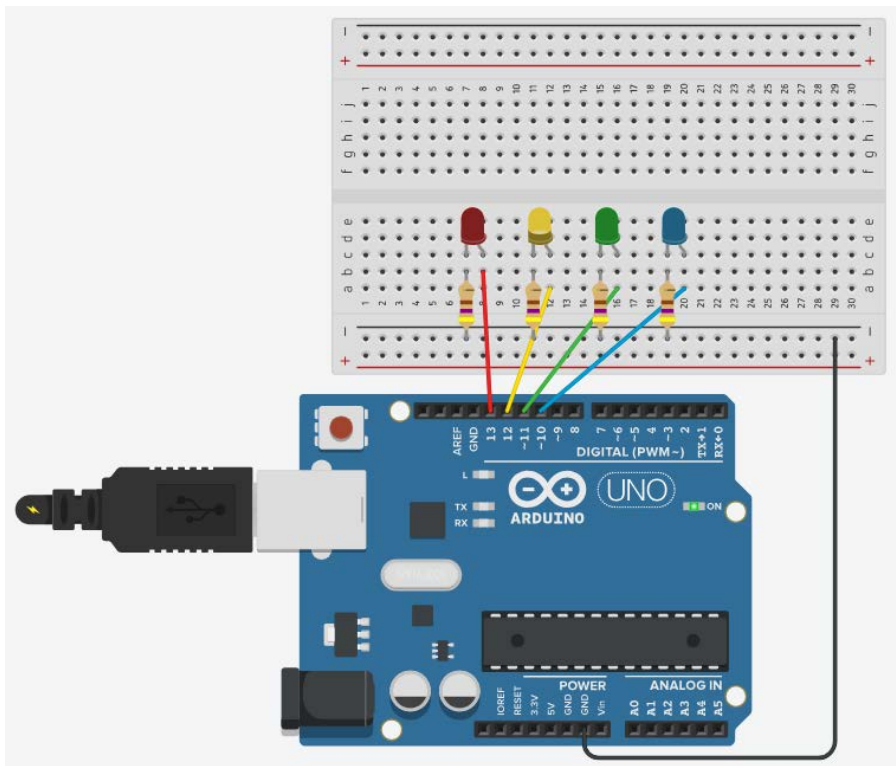


Figura 10 - ULA 1 bit - Teste 0 1 0



FIM