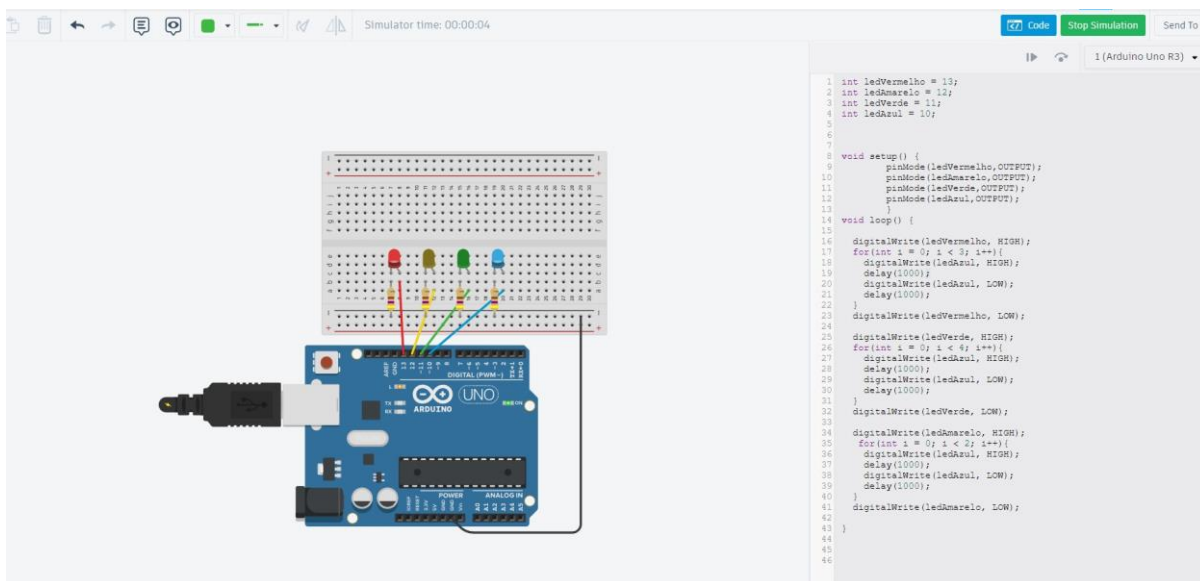


Arquitetura de Computadores 2

Victor Souza Lima

Parte 1:

-Semáforo Arduino



Código:

```
int ledVermelho = 13;
```

```
int ledAmarelo = 12;
```

```
int ledVerde = 11;
```

```
int ledAzul = 10;
```

```
void setup() {
```

```
    pinMode(ledVermelho, OUTPUT);
```

```
    pinMode(ledAmarelo,OUTPUT);
    pinMode(ledVerde,OUTPUT);
    pinMode(ledAzul,OUTPUT);
}
void loop() {

    digitalWrite(ledVermelho, HIGH);
    for(int i = 0; i < 3; i++){
        digitalWrite(ledAzul, HIGH);
        delay(1000);
        digitalWrite(ledAzul, LOW);
        delay(1000);
    }
    digitalWrite(ledVermelho, LOW);

    digitalWrite(ledVerde, HIGH);
    for(int i = 0; i < 4; i++){
        digitalWrite(ledAzul, HIGH);
        delay(1000);
        digitalWrite(ledAzul, LOW);
        delay(1000);
    }
    digitalWrite(ledVerde, LOW);

    digitalWrite(ledAmarelo, HIGH);
    for(int i = 0; i < 2; i++){
        digitalWrite(ledAzul, HIGH);
        delay(1000);
```

```

digitalWrite(ledAzul, LOW);

delay(1000);

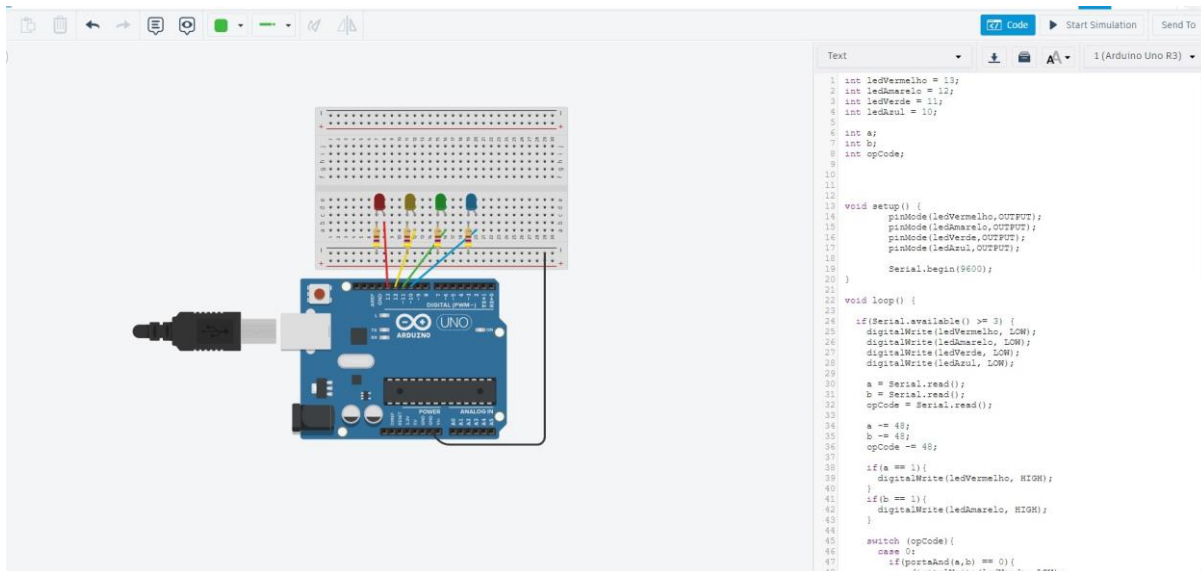
}

digitalWrite(ledAmarelo, LOW);

}

```

Parte 2:



Codigo:

```

int ledVermelho = 13;

int ledAmarelo = 12;

int ledVerde = 11;

int ledAzul = 10;

int a;

int b;

int opCode;

```

```
void setup() {  
    pinMode(ledVermelho,OUTPUT);  
    pinMode(ledAmarelo,OUTPUT);  
    pinMode(ledVerde,OUTPUT);  
    pinMode(ledAzul,OUTPUT);  
  
    Serial.begin(9600);  
}
```

```
void loop() {  
  
    if(Serial.available() >= 3) {  
        digitalWrite(ledVermelho, LOW);  
        digitalWrite(ledAmarelo, LOW);  
        digitalWrite(ledVerde, LOW);  
        digitalWrite(ledAzul, LOW);  
  
        a = Serial.read();  
        b = Serial.read();  
        opCode = Serial.read();  
  
        a -= 48;  
        b -= 48;  
        opCode -= 48;
```

```

if(a == 1){
    digitalWrite(ledVermelho, HIGH);
}

if(b == 1){
    digitalWrite(ledAmarelo, HIGH);
}


switch (opCode){
    case 0:
        if(portaAnd(a,b) == 0){
            digitalWrite(ledVerde, LOW);
        }else{
digitalWrite(ledVerde, HIGH);
            digitalWrite(ledAzul, HIGH);
        }
        break;


    case 1:
        if(portaOr(a,b) == 0){
            digitalWrite(ledVerde, LOW);
        }else{
digitalWrite(ledVerde, HIGH);
            if(portaAnd(a,b) == 1){
                digitalWrite(ledAzul, HIGH);
            }
        }
    }
}

```

```
break;
```

```
case 2:
```

```
    if(portaNot(a) == -2){
```

```
        digitalWrite(ledVerde, LOW);
```

```
    }else{
```

```
digitalWrite(ledVerde, HIGH);
```

```
    }
```

```
    break;
```

```
case 3:
```

```
    if(somador(a,b,1) == 0){
```

```
        digitalWrite(ledVerde, LOW);
```

```
    if(somador(a,b,0) == 1){
```

```
        digitalWrite(ledAzul, HIGH);
```

```
    }
```

```
    }else{
```

```
        digitalWrite(ledVerde, HIGH);
```

```
    }
```

```
    break;
```

```
}
```

```
}
```

```
}
```

```
int portaAnd(int v1, int v2){
```

```
    return(a&b);  
}
```

```
int portaOr(int v1, int v2){  
    return(a|b);  
}
```

```
int portaNot(int v1){  
    return(~v1);  
}
```

```
int portaXor(int v1, int v2){  
    return(v1 ^v2);  
}
```

```
int somador(int v1, int v2, int param){  
    if(param == 0){  
        return portaAnd(v1,v2);  
    }  
    return portaXor(v1,v2);  
}
```

1)

Considere o seguinte programa a ser executado em uma Unidade Lógica e Aritmética (ULA) considerando números de 1 bit:

Início:

A=0;

B=1;

```

AND(A,B);

B=0;

A=1;

OR(A,B);

SOMA(A,B);

A=0;

NOT(A);

B=1;

AND(B,A);

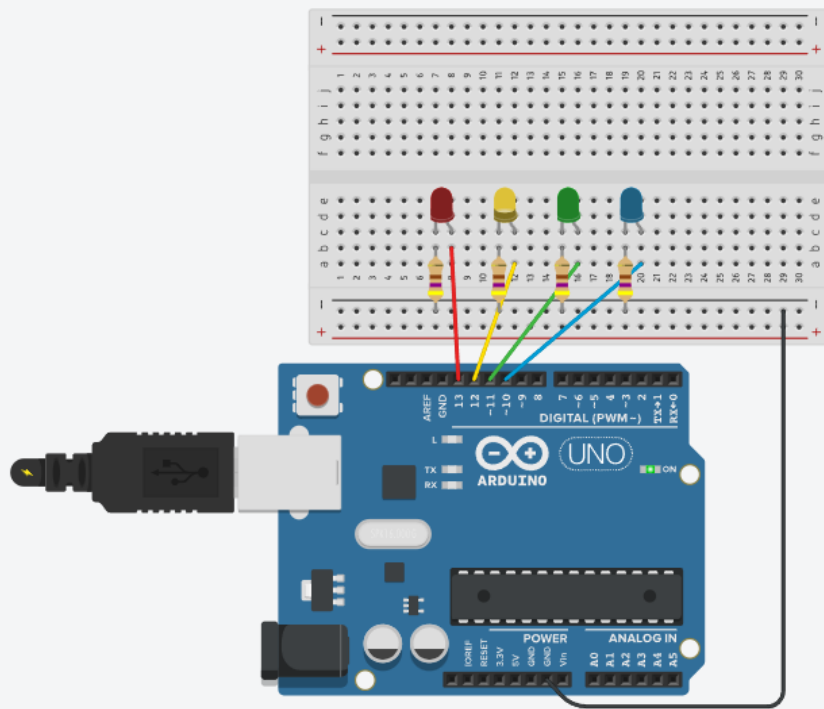
```

Fim.

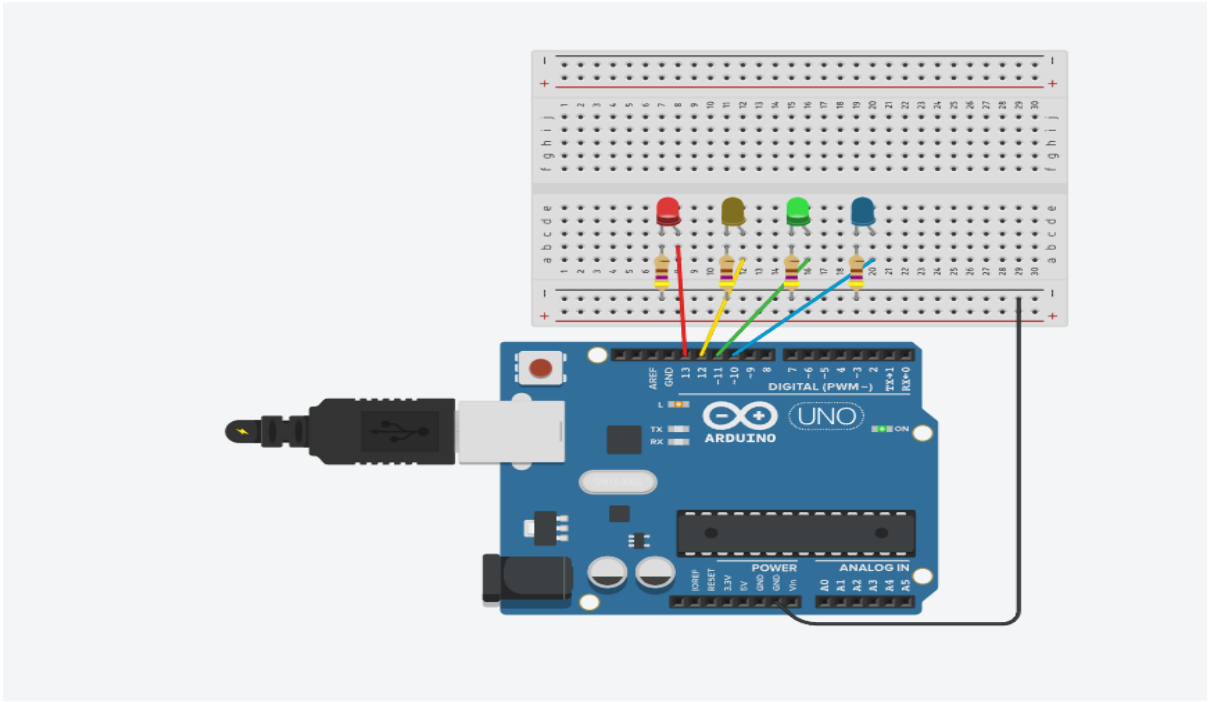
Para o programa de teste acima, preencher a tabela a seguir considerando que cada linha corresponderá à execução de uma instrução (a primeira linha já foi realizada, observe que a palavra de código deverá conter 4 bits, para escrevermos em hexa 0x na frente do número):

Instrução realizada	Binário (A,B,Op.code)	Valor em Hexa (0x ...)	Resultado em binário
AND(A,B)	0 1 00	4	0
OR(A,B)	1 0 01	9	1
SOMA(A,B)	1 0 11	B	1
NOT(A)	0 0 10	2	1
AND(B,A)	1 0 00	8	0

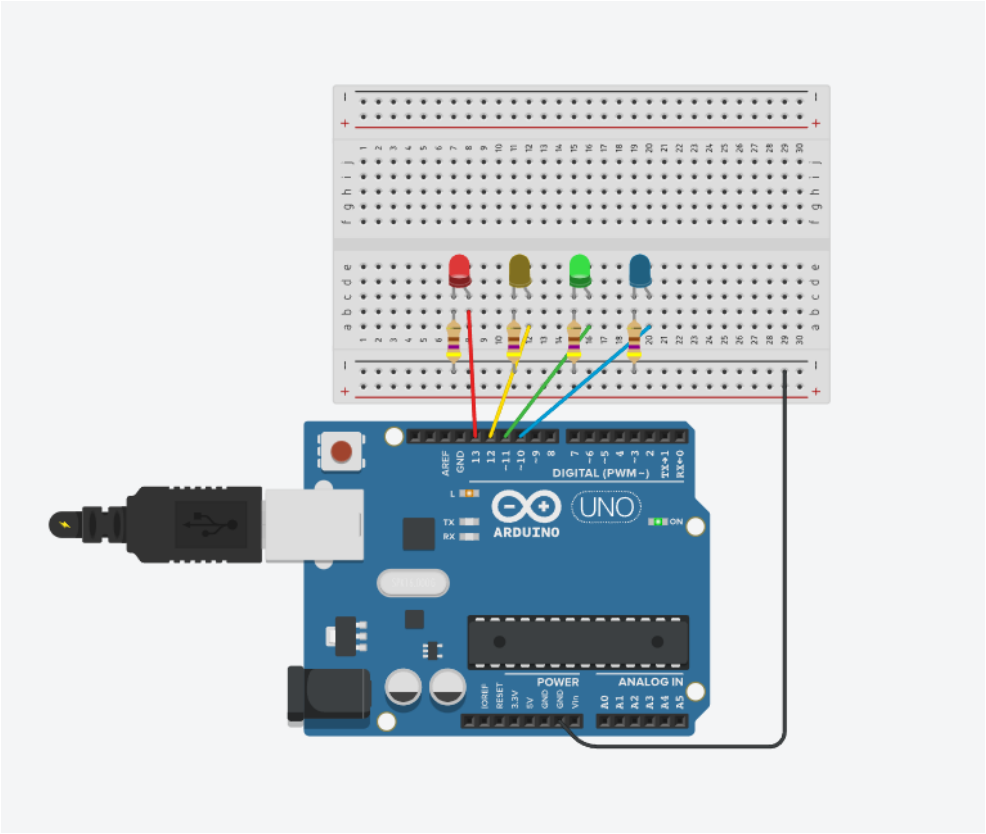
AND(A,B)



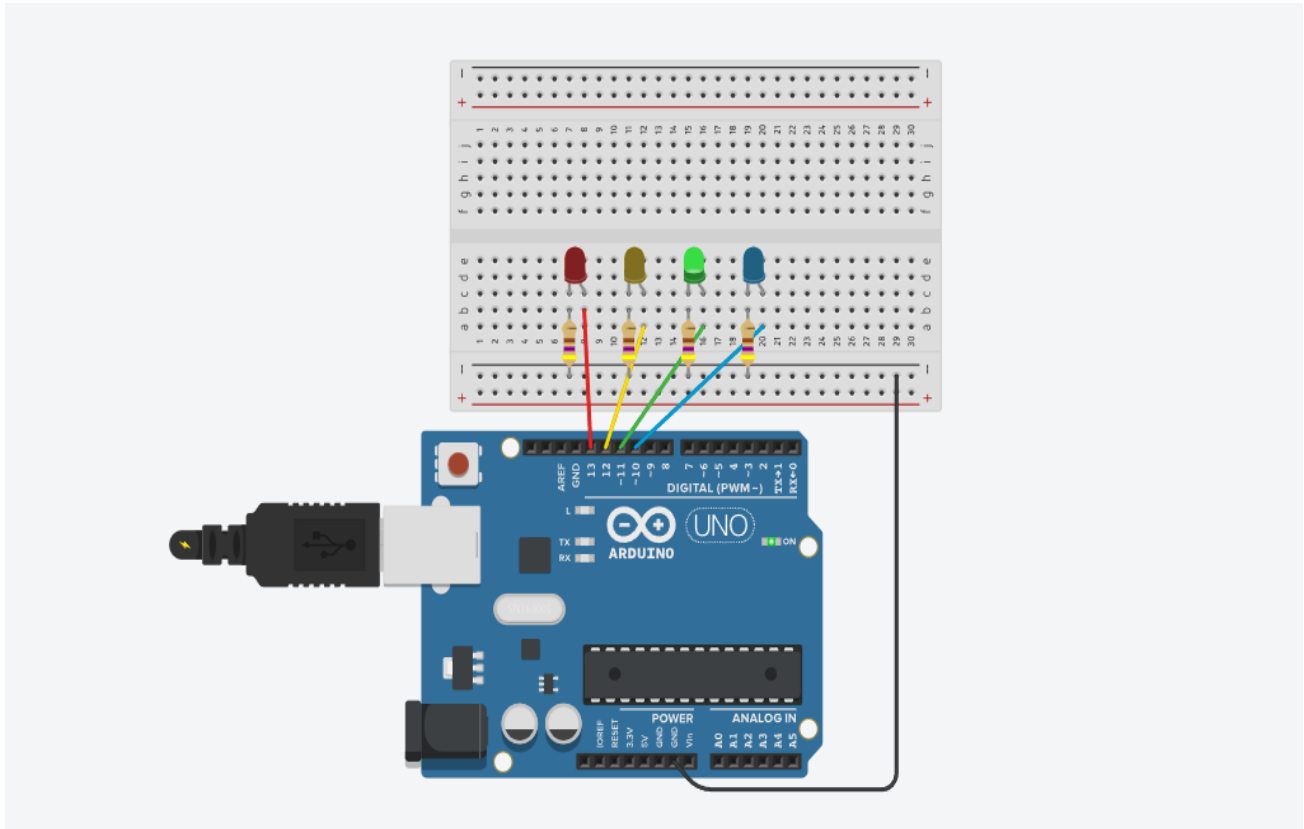
OR(A,B)



SOMA(A,B)



NOT(A)



AND(B,A)

