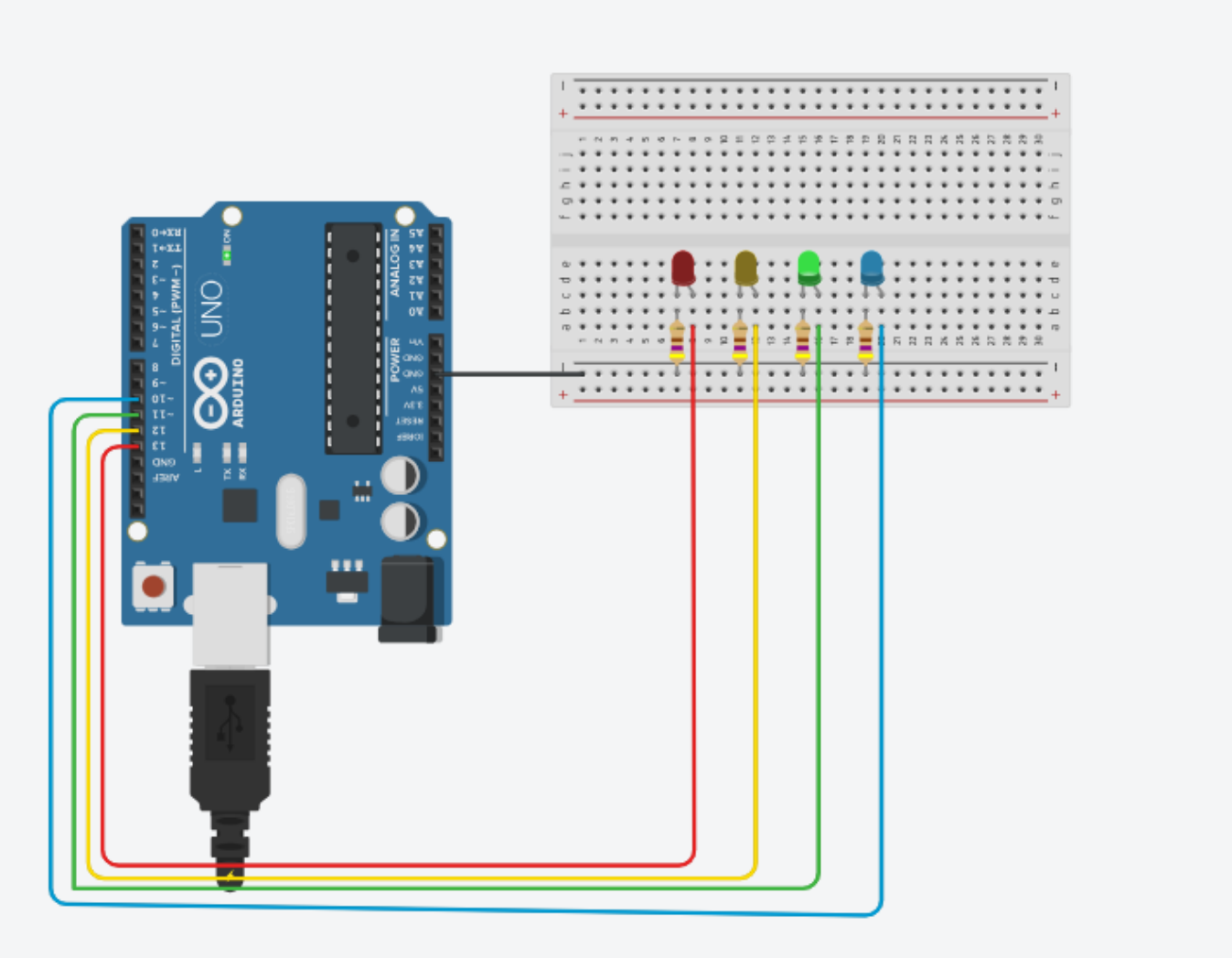
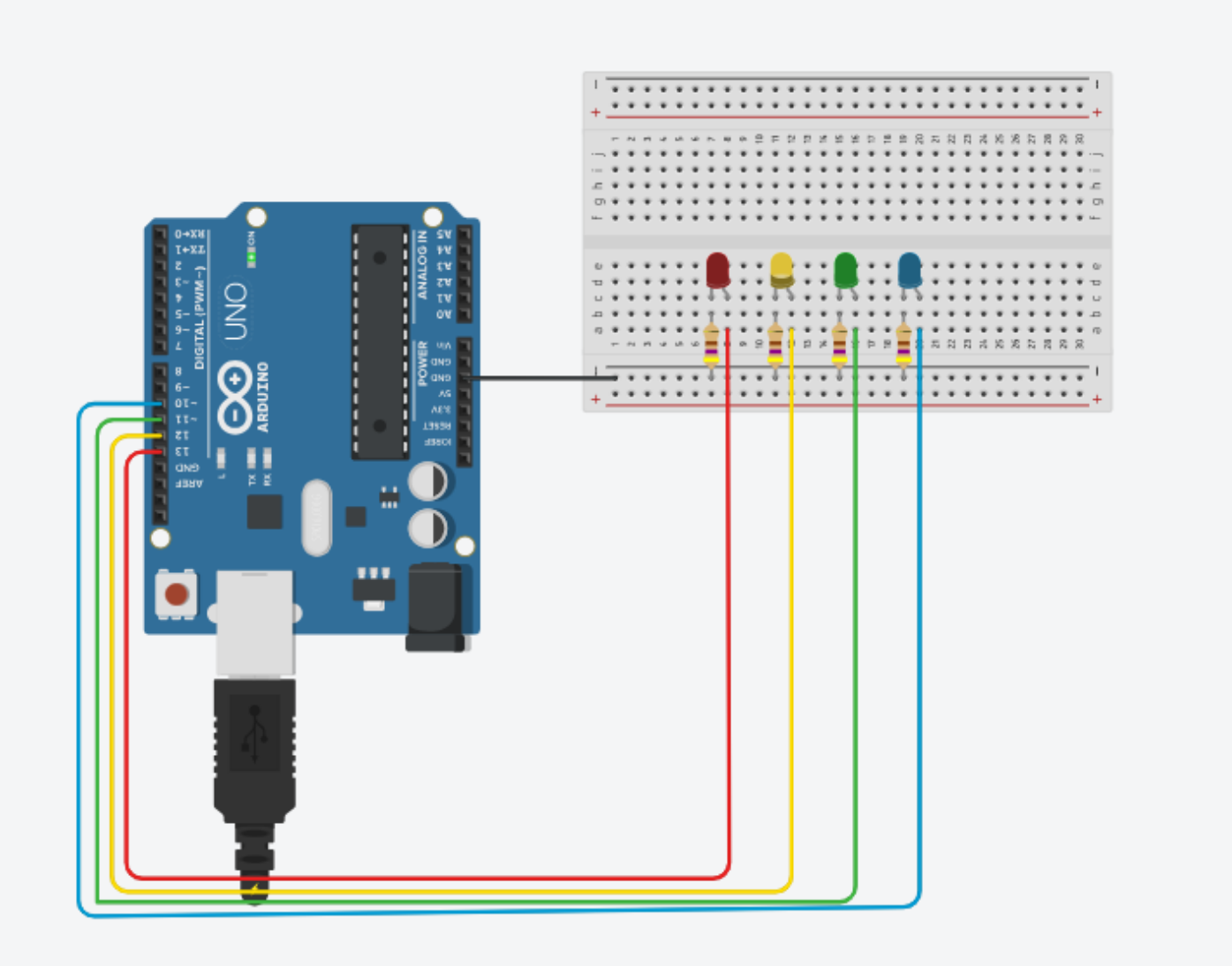
```
/*
  Programa 01
  Semáforo
*/

// Definiçao de valores para variáveis
int led10 = 10;
int led11 = 11;
int led12 = 12;
int led13 = 13;

void cicloLed10(int x){
  for(int i=0;i<x;i++){
   digitalWrite(led10, HIGH);
delay(1000);
digitalWrite(led10, LOW);
   delay(1000);
  }
}
// Rotina executada 1 vez e que em geral configura entradas e saídas
void setup() {
  // configura os pinos como saídas DIGITAIS.
  pinMode(led10, OUTPUT);
  pinMode(led11, OUTPUT);
  pinMode(led12, OUTPUT);
  pinMode(led13, OUTPUT);

}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led13, HIGH);
  cicloLed10(3);
  digitalWrite(led13, LOW);

  digitalWrite(led11, HIGH);
  cicloLed10(4);
  digitalWrite(led11, LOW);

  digitalWrite(led12, HIGH);
  cicloLed10(2);
  digitalWrite(led12, LOW);
 }
```



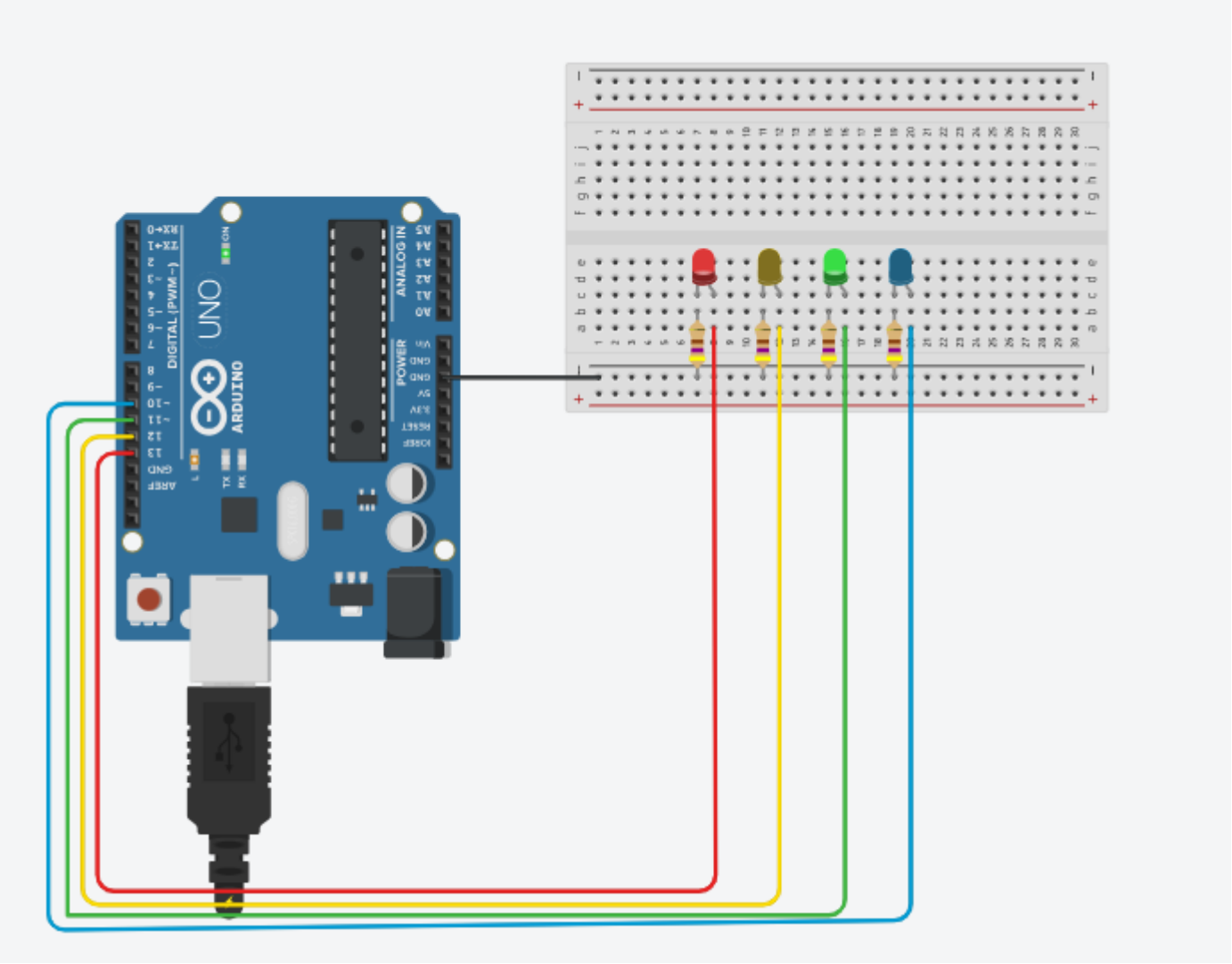

2.

| 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) | 0 1 00 | 0x4 | 0 |

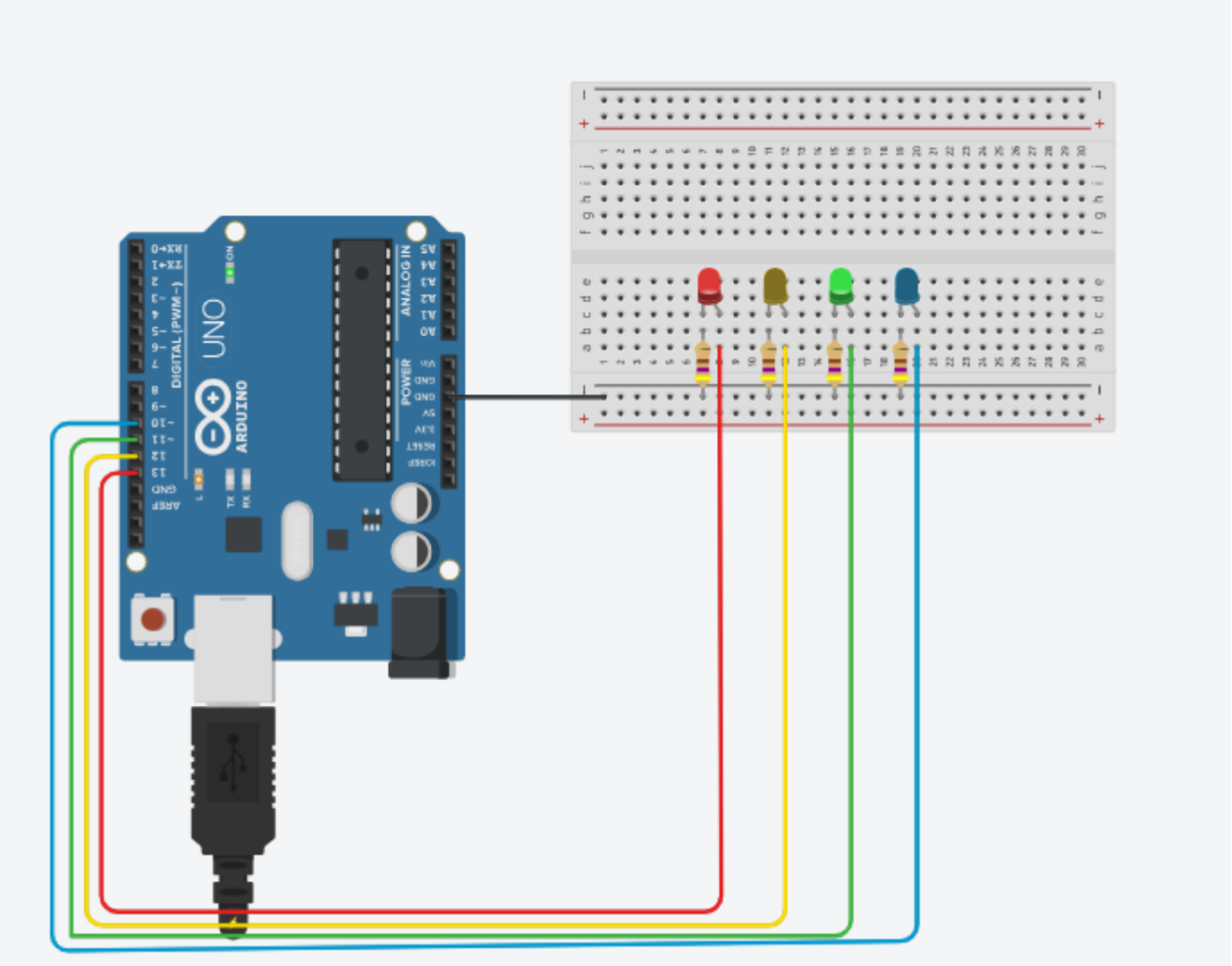
AND(A,B);



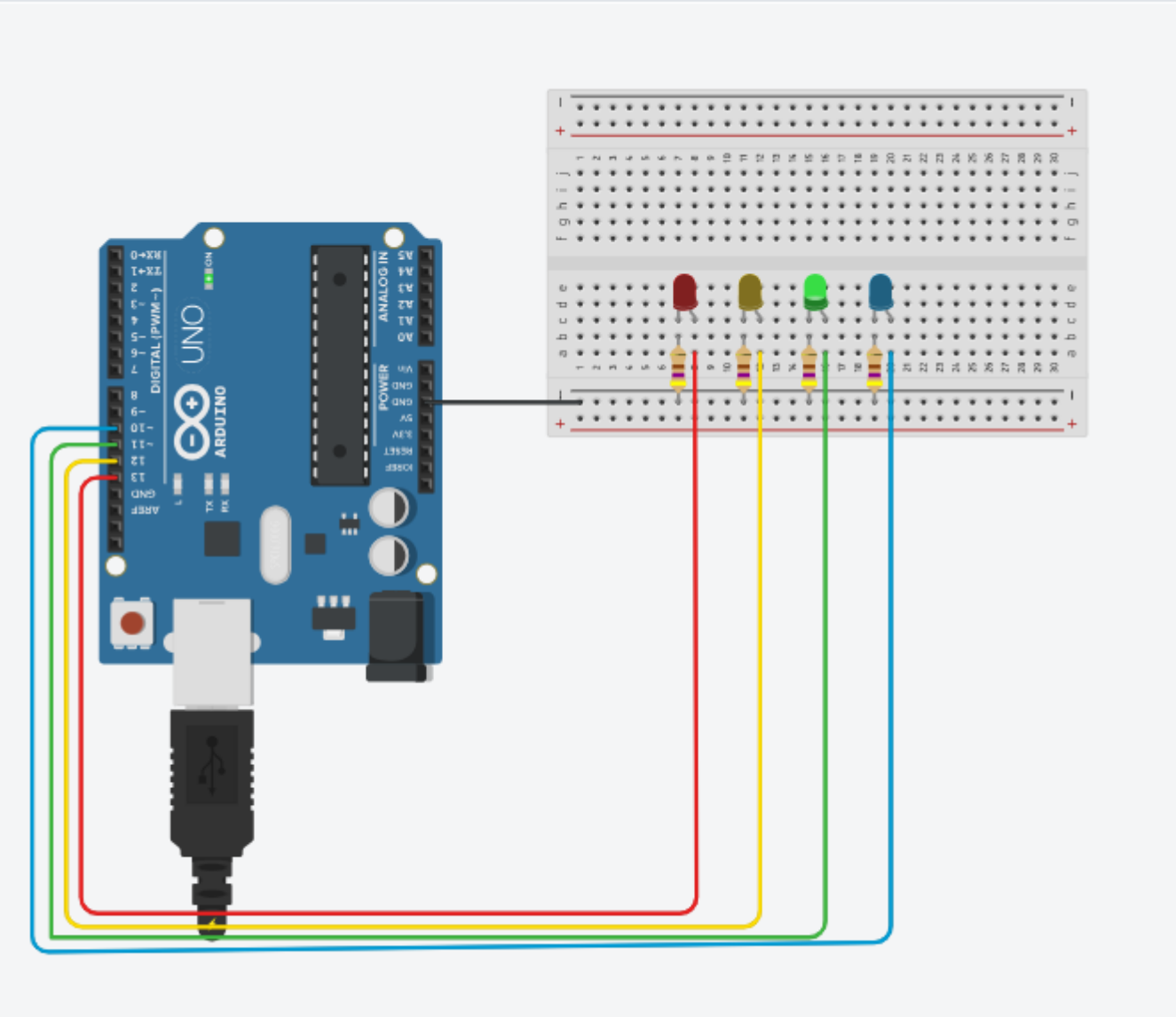
OR(A,B);



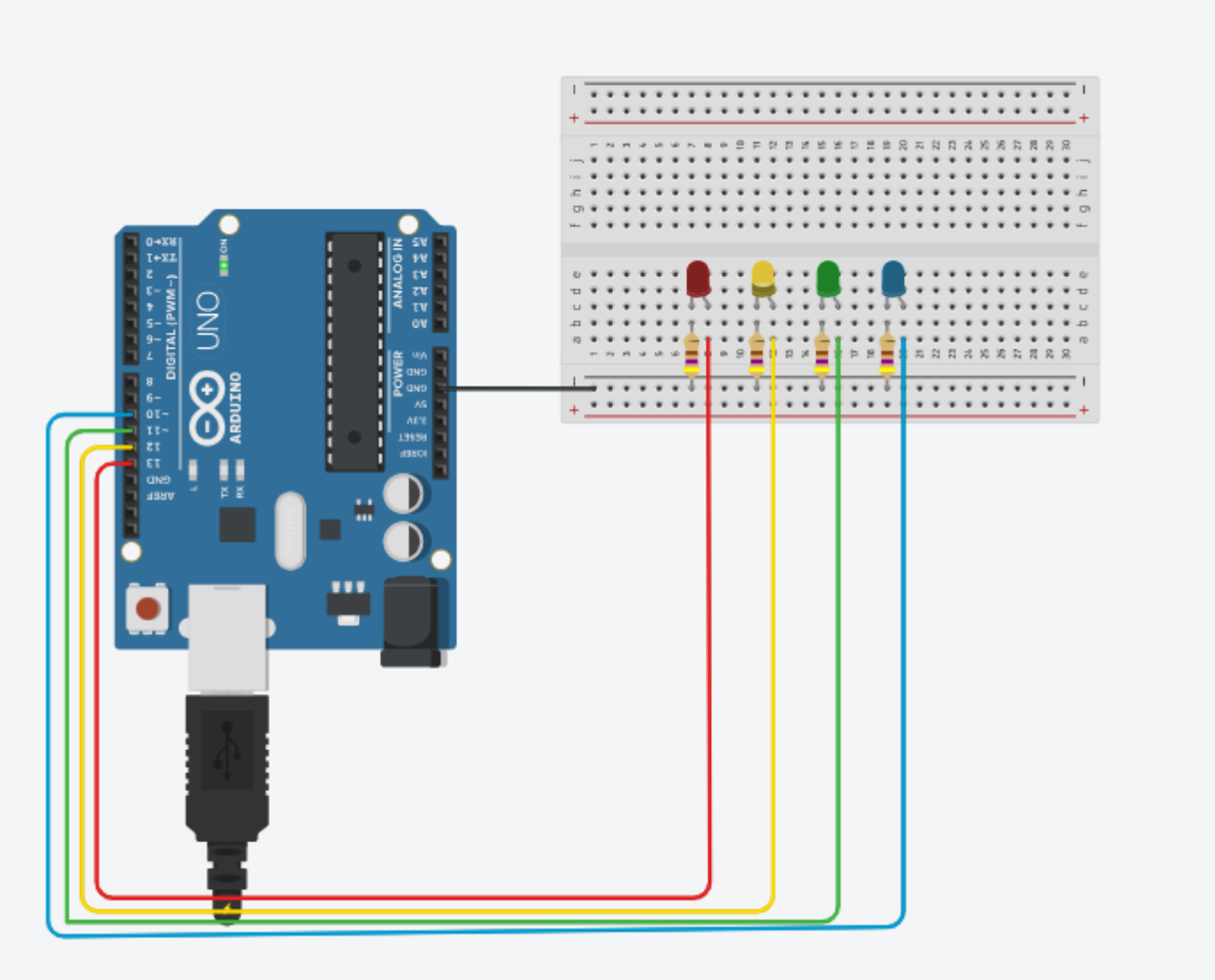
SOMA(A,B);



NOT(A);



AND(B,A);




// Definiçao de valores para variáveis

int led10 = 10;

int led11 = 11;

int led12 = 12;

int led13 = 13;

int a=0;

```
int b=0;
int operacao=0;


// Operação (a AND b)
void andOp(int a, int b){
  if( a==1 && b==1){
    digitalWrite(led11, HIGH);
  }
  else{
    digitalWrite(led11, LOW);
  }
}


// Operação (a OR b)
void orOp(int a, int b){
  if( a==1 || b==1){
    digitalWrite(led11, HIGH);
  }
  else{
    digitalWrite(led11, LOW);
  }
}
```

```
// Operação (NOT a)
void notOp(int a){
  if(a==0){
    digitalWrite(led11, HIGH);
  }
  else{
    digitalWrite(led11, LOW);
  }
}


// Operação (a SOMA b)
void somaOp(int a, int b){
  int soma = a + b;

  if(soma==0){
    digitalWrite(led11, LOW);
    digitalWrite(led10, LOW);
  }
  else if(soma==1){
    digitalWrite(led11, HIGH);
    digitalWrite(led10, LOW);
  }
  else{
    digitalWrite(led11, HIGH);
```

```
    digitalWrite(led10, HIGH);
  }
}


// Função para processar a entrada
void processInput(String input) {
  if (input.length() == 3) {  // Verifica se o comprimento da string é 3
    a = input.charAt(0) - '0';  // Converte o primeiro caractere para número
    b = input.charAt(1) - '0';  // Converte o segundo caractere para número
    operacao = input.charAt(2) - '0';  // Converte o terceiro caractere para número
  } else {
    Serial.println("Entrada inválida. Digite 3 números (ex: 110).");
  }
}


// Rotina executada 1 vez e que em geral configura entradas e saídas
void setup() {
  // configura os pinos como saídas DIGITAIS.
  pinMode(led10, OUTPUT);
  pinMode(led11, OUTPUT);
  pinMode(led12, OUTPUT);
  pinMode(led13, OUTPUT);
```

```
  Serial.begin(9600);
}



// the loop routine runs over and over again forever:
void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');  // Lê a string até o fim da linha
    processInput(input);
  }


  if(a==1){
    digitalWrite(led13, HIGH);
  }
  else{
    digitalWrite(led13, LOW);
  }
  if(b==1){
    digitalWrite(led12, HIGH);
  }
  else{
    digitalWrite(led12, LOW);
  }
```

```
  switch(operacao){
    case 0:
          andOp(a,b);
     break;
    case 1:
     orOp(a,b);
     break;
    case 2:
     notOp(a);
     break;
    case 3:
     somaOp(a,b);
     break;
    default:
     Serial.println("Operacao invalida");
     break;

    }
}
```