

Nome: Gabriel Henrique Vieira de Oliveira

Data: 08/06/2025

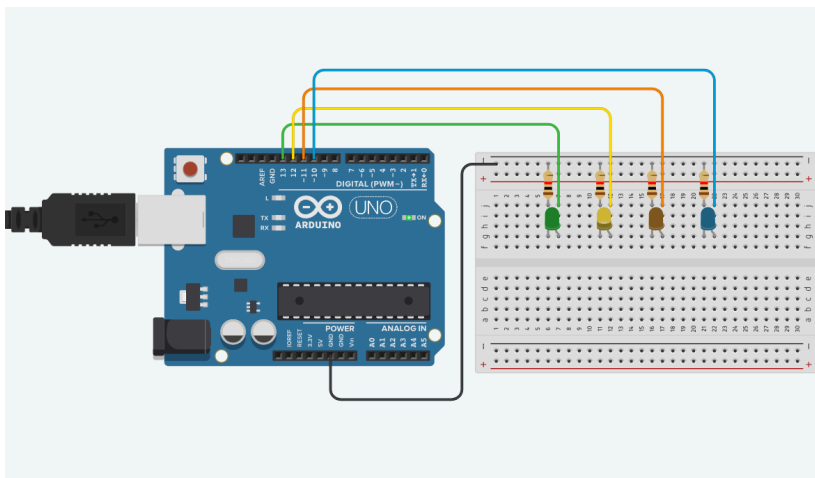
Matéria: Laboratório de Introdução à Engenharia da Computação

Rel 09 – Comunicação Serial e Saídas analógicas

O que deve ser entregue: Para cada exercício, entregue um print da montagem do circuito e a colagem do código desenvolvido.

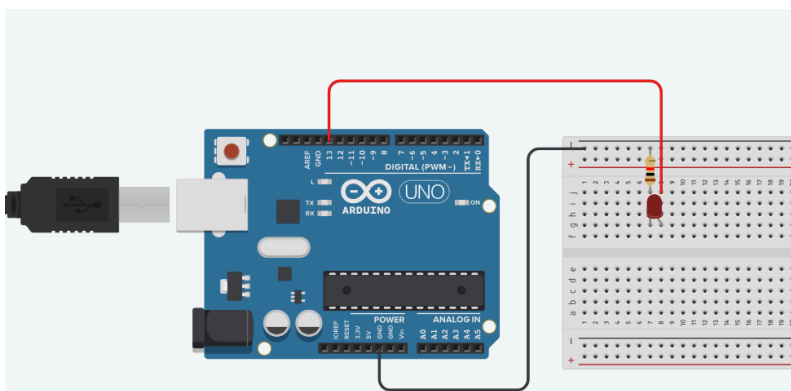
Resolva os seguintes exercícios:

1) Implementar um programa onde 4 letras possam ser digitadas. Cada letra irá acionar apenas um dos leds nos pinos 13, 12, 11 e 10. Use as letras 'r', 'y', 'g', 'b', respectivamente. Por fim, a letra 'a' aciona todos juntos e a letra 'x' apaga todos.



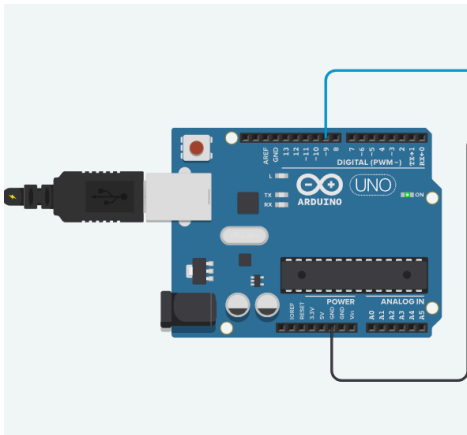
```
1 char letra;
2 int leds[] = {13, 12, 11, 10};
3
4 void setup() {
5   Serial.begin(9600);
6   for (int i = 0; i < 4; i++) {
7     pinMode(leds[i], OUTPUT);
8   }
9 }
10
11 void loop() {
12   if (Serial.available()) {
13     letra = Serial.read();
14
15     // Desliga todos antes de processar nova entrada
16     for (int i = 0; i < 4; i++) {
17       digitalWrite(leds[i], LOW);
18     }
19
20     if (letra == 'r') digitalWrite(13, HIGH);
21     else if (letra == 'y') digitalWrite(12, HIGH);
22     else if (letra == 'g') digitalWrite(11, HIGH);
23     else if (letra == 'b') digitalWrite(10, HIGH);
24     else if (letra == 'a') {
25       for (int i = 0; i < 4; i++) {
26         digitalWrite(leds[i], HIGH);
27       }
28     }
29     // 'x' já desliga tudo no começo
30   }
31 }
```

2) Se o arduino receber a letra 'a' um Led no pino 13 pisca em uma velocidade (1 s em 1 s), se receber uma letra 'b' ele irá piscar em outra velocidade (0,5 s em 0,5 s).



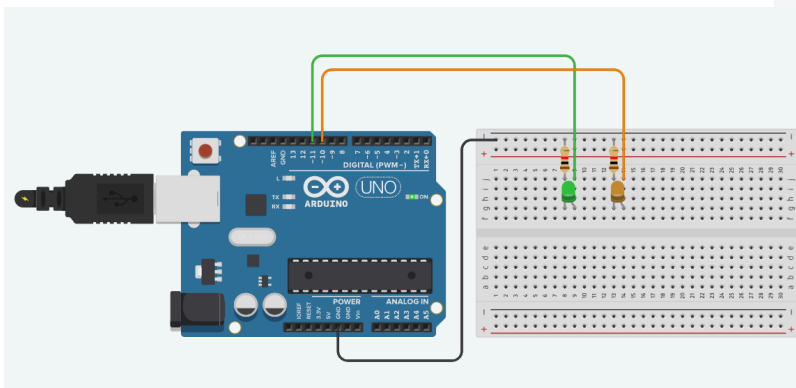
```
1 char letra = 0; // Armazena a última letra válida
2 int led = 13;
3
4 void setup() {
5   Serial.begin(9600);
6   pinMode(led, OUTPUT);
7 }
8
9 void loop() {
10  // Verifica se uma nova letra foi digitada
11  if (Serial.available()) {
12    char entrada = Serial.read();
13
14    // Só aceita 'a' ou 'b' como comandos válidos
15    if (entrada == 'a' || entrada == 'b') {
16      letra = entrada; // Atualiza comando válido
17    } else {
18      letra = 0; // Qualquer outra letra cancela o piscar
19      digitalWrite(led, LOW); // Apaga o LED
20    }
21  } // Se eu colocasse toda essa definição de digitar dentro do serial.read, iria dar
22  // tendo em vista que o led só iria ficar piscando quando recebesse algum serial
23
24  // Pisca conforme a última letra válida recebida
25  if (letra == 'a') {
26    digitalWrite(led, HIGH);
27    delay(1000);
28    digitalWrite(led, LOW);
29    delay(1000);
30  }
31
32  else if (letra == 'b') {
33    digitalWrite(led, HIGH);
34    delay(500);
35    digitalWrite(led, LOW);
36    delay(500);
37  }
38
39  else {
40    digitalWrite(led, LOW); // Se não for 'a' nem 'b', mantém apagado
41    delay(100); // Evita travar o loop
42  }
43 }
44 }
```

3) Implementar um programa onde o led brilhe gradualmente de apagado até brilho máximo e em seguida volte de brilho máximo até apagado.



```
1 int led = 9;           // Pino com suporte PWM
2 //PWM: é uma técnica eletrônica utilizada para
3 //controlar a potência ou a velocidade de um dispositivo,
4 //ajustando a largura dos pulsos de uma forma de onda periódica.
5 //Em vez de desligar completamente a energia, o PWM liga e desliga
6 //rapidamente a energia, criando a ilusão de um nível de tensão
7 //constante, mas com um valor médio ajustado.
8 int brilho = 0;       // Nível de brilho (0 a 255)
9 int passo = 5;        // Variação do brilho (o passo define quanto o brilho do
10 //LED muda a cada ciclo do loop()). 0 a 5, 5 a 10... 225. (x5)|
11
12 void setup() {
13   pinMode(led, OUTPUT);
14 }
15
16 void loop() {
17   analogWrite(led, brilho); // Define o brilho
18
19   brilho = brilho + passo;
20
21   // Inverte o sentido do fade
22   if (brilho <= 0 || brilho >= 255) {
23     passo = -passo;
24   }
25
26   delay(30); // Espera para suavizar o efeito
27 }
```

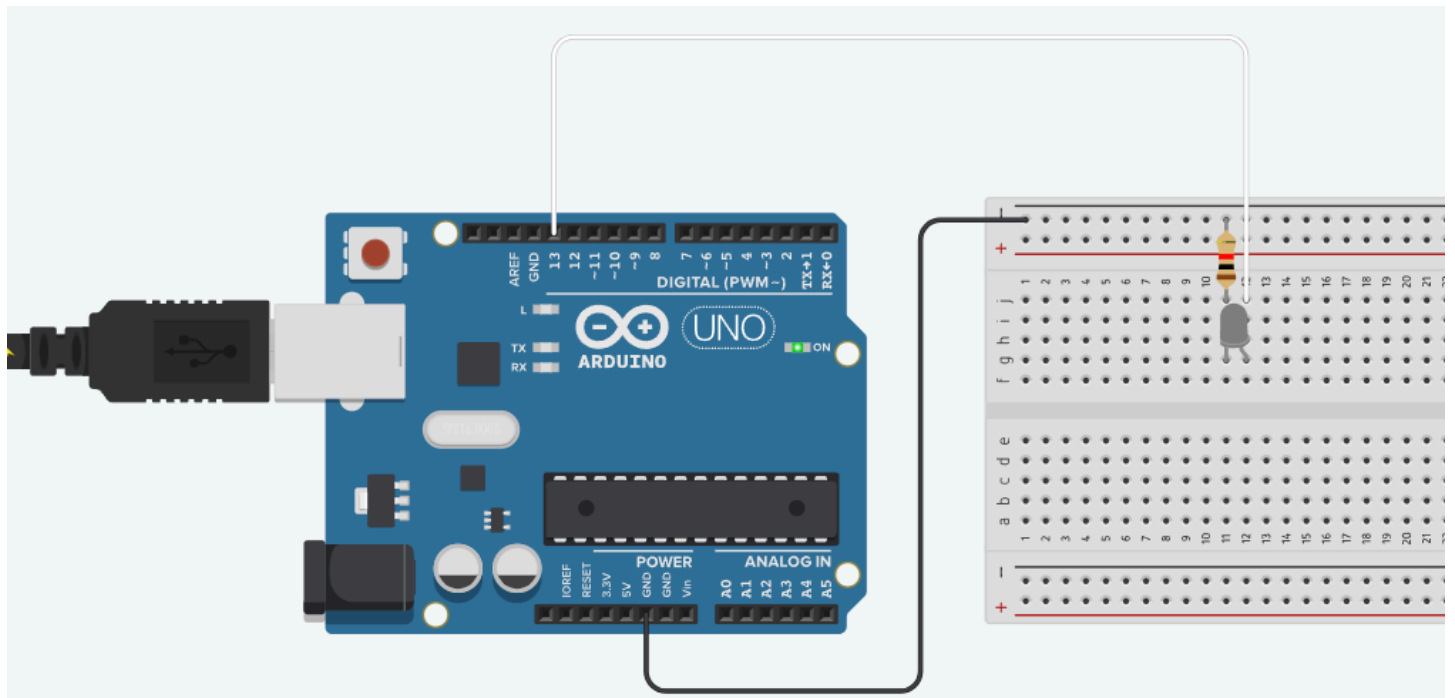
4) Implementar um programa com dois Leds, um no pino 11 outro no 10, quando um aumenta o brilho o outro diminui.



```
1 int led1 = 11;
2 int led2 = 10;
3 int brilho = 0; // 0 a 225
4 int passo = 5;
5
6 void setup() {
7   pinMode(led1, OUTPUT);
8   pinMode(led2, OUTPUT);
9 }
10
11 void loop() {
12   // LED1 aumenta, LED2 diminui
13   analogWrite(led1, brilho);
14   analogWrite(led2, 255 - brilho); // inverso do LED1 (começa
15   // no final, de 225 a 0)
16
17   brilho = brilho + passo; //+=
18
19   // Inverte o sentido do fade
20   if (brilho <= 0 || brilho >= 255) {
21     passo = -passo;
22   }
23
24   delay(30); // Espera para suavizar o efeito
25 }
26
27 }
```

5) Implementar um programa que simule um acionador Morse. Quando uma letra for digitada ele aciona um Led no pino 13 de acordo com o código Morse. Use 100 ms (led ligado) para ponto e 300 ms (led ligado) para o traço e 200 ms (led apagado) entre traços ou pontos.

A ● -	J ● - - -	S ● ● ●
B - ● ● ●	K - ● -	T -
C - ● - ●	L ● - ● ●	U ● ● -
D - ● ●	M - -	V ● ● ● -
E ●	N - ●	W ● - -
F ● ● - ●	O - - -	X - ● ● -
G - - ●	P ● - - ●	Y - ● - -
H ● ● ● ●	Q - - ● -	Z - - ● ●
I ● ●	R ● - ●	

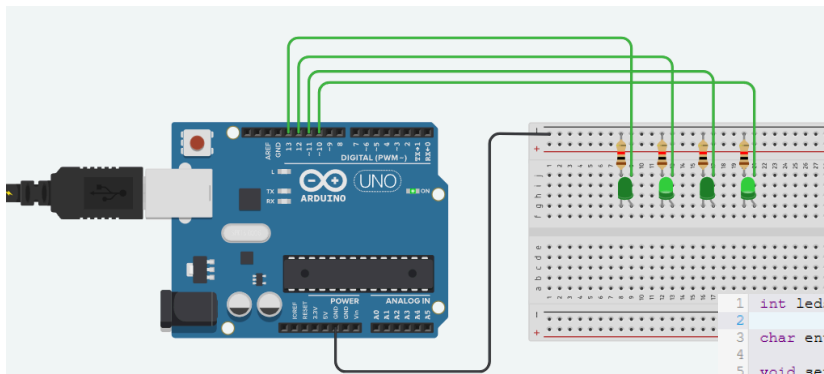


```

1  int led = 13;
2
3  // Estrutura de dados com códigos Morse
4  const char* morseCode[] = {
5      ".-.", // A
6      "-...-", // B
7      "-.-.", // C
8      "-.-.", // D
9      ".-", // E
10     "...-.", // F
11     "--.", // G
12     "....", // H
13     "..", // I
14     "-.-.-", // J
15     "-.-.", // K
16     "-...-", // L
17     "--", // M
18     "-.", // N
19     "---", // O
20     "-.-.-", // P
21     "--.-.", // Q
22     "-.-.", // R
23     "...", // S
24     "-", // T
25     "-.-.", // U
26     "...-.", // V
27     "-.-.-", // W
28     "-...-", // X
29     "-.-.-", // Y
30     "-...-", // Z
31 };
32
33 void setup() {
34     pinMode(led, OUTPUT);
35     Serial.begin(9600);
36
37
38 void loop() {
39     if (Serial.available()) {
40         char letra = Serial.read();
41
42         // Transforma em maiúsculo
43         if (letra >= 'a' && letra <= 'z') {
44             letra = letra - 'a' + 'A';
45         }
46
47         // Só continua se for letra
48         if (letra >= 'A' && letra <= 'Z') {
49             const char* codigo = morseCode[letra - 'A'];
50
51             // Percorre cada caractere do código Morse
52             for (int i = 0; codigo[i] != '\0'; i++) {
53                 if (codigo[i] == '.') {
54                     digitalWrite(led, HIGH);
55                     delay(100); // Ponto = 100 ms
56                 } else if (codigo[i] == '-') {
57                     digitalWrite(led, HIGH);
58                     delay(300); // Traço = 300 ms
59                 }
60
61                 digitalWrite(led, LOW);
62                 delay(200); // Intervalo entre ponto/traço
63             }
64
65             // Pausa maior entre letras
66             delay(400);
67         }
68     }
69 }

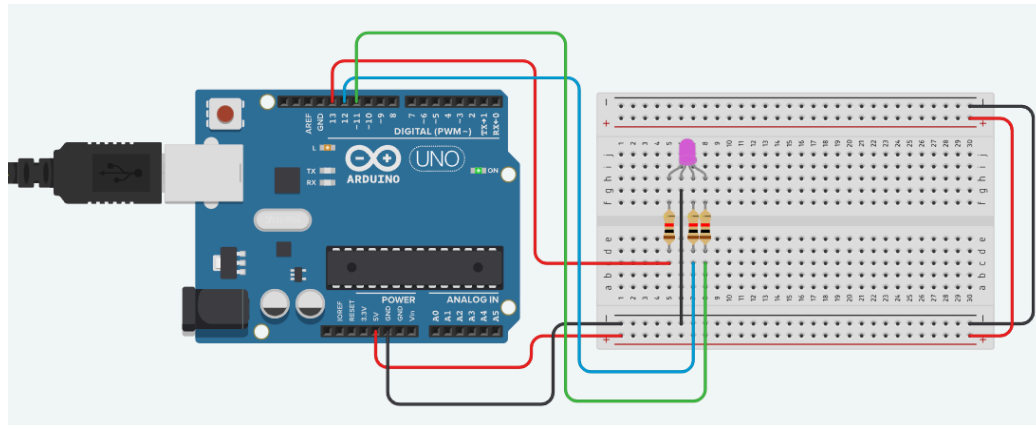
```

6) Implementar um programa em que é digitado um número até 9 e ele mostrará o valor em binário nos leds. Use os leds nos pinos 13, 12, 11 e 10 (valendo 8, 4, 2, 1 respectivamente). Exemplo: Se eu digitar 5 (0101) veremos 13 apagado, 12 aceso, 11 apagado e 10 aceso.



```
1 int leds[] = {13, 12, 11, 10}; // Ordem dos bits: 8, 4, 2, 1
2
3 char entrada;
4
5 void setup(){
6
7     for (int i = 0; i < 4; i++) {
8         pinMode(leds[i], OUTPUT);
9         digitalWrite(leds[i], LOW); // Garante tudo apagado no inicio
10    }
11
12    Serial.begin(9600);
13 }
14
15 void loop(){
16     if(Serial.available()){
17         entrada = Serial.read();
18
19         // Verifica se é um número entre '0' e '9'
20         if (entrada >= '0' && entrada <= '9') {
21             int valor = entrada - '0'; // Converte caractere para número (ex: '5' -> 5)
22
23             // Atualiza LEDs de acordo com os bits
24             for (int i = 0; i < 4; i++) {
25                 int bit = (valor >> (3 - i)) & 1; // Pega o bit correspondente
26                 digitalWrite(leds[i], bit);
27             }
28         }
29     }
30 }
```

7) Crie um programa para alterar o led RGB pegando várias cores possíveis (mais do que 3 cores). Use sua criatividade nesse exercício.



```
1 const int vermelho = 13;
2 const int azul = 12;
3 const int verde = 11;
4 String cor = ""; // Armazena a entrada do usuário
5
6 void setup() {
7     pinMode(vermelho, OUTPUT);
8     pinMode(verde, OUTPUT);
9     pinMode(azul, OUTPUT);
10    Serial.begin(9600);
11    Serial.println("Digite uma cor (vermelho, verde, azul, amarelo, ciano, magenta,");
12 }
13
14 void loop() {
15     if (Serial.available() > 0) {
16         cor = Serial.readStringUntil('\n');
17         cor.trim(); // Remove espaços ou quebras de linha extras
18         cor.toLowerCase(); // Aceita "Verde", "VERDE", etc.
19
20         if (cor == "vermelho") {
21             setCor(true, false, false);
22         } else if (cor == "verde") {
23             setCor(false, true, false);
24         } else if (cor == "azul") {
25             setCor(false, false, true);
26         } else if (cor == "amarelo") {
27             setCor(true, true, false);
28         } else if (cor == "ciano") {
29             setCor(false, true, true);
30         } else if (cor == "magenta") {
31             setCor(true, false, true);
32         } else if (cor == "branco") {
33             setCor(true, true, true);
34         }
35     }
36 }
```

```
34 } else if (cor == "desligar") {
35     setCor(false, false, false);
36 } else {
37     Serial.println("Cor inválida. Tente novamente.");
38 }
39
40 Serial.println("Digite outra cor:");
41 }
42 }
43
44 void setCor(bool r, bool g, bool b) {
45     digitalWrite(vermelho, r ? HIGH : LOW);
46     digitalWrite(verde, g ? HIGH : LOW);
47     digitalWrite(azul, b ? HIGH : LOW);
48 }
49 }
```

Monitor serial

```
digite uma cor (vermelho, verde, azul, amarelo, ciano, magenta, branco, desligar):
digite outra cor:
digite outra cor:
```