



LAB.1 – PORTAS DIGITAIS ARDUINO – OUTPUTS
PISCA LED SIMPLES/SEMAFORO SIMPLES/ SEMAFORO DE PEDESTRES/SINALIZADOR DE
CÓDIGO MORSE

DISCIPLINA: AUTOMAÇÃO DE PEQUENOS PROJETOS
PROFESSOR: MARCELO HILÁRIO SILVEIRA

Nome: _____ RA.: _____
Nome: _____ RA.: _____
Nome: _____ RA.: _____

Turma: _____ Data: ____/____/____

OBS: SALVE TODOS OS SEUS CÓDIGOS EM FORMATO *.INO OU *.TXT E ENVIE PARA SEU EMAIL OU PEN DRIVE.

I. PROJETO 1 – PISCA LED SIMPLES

1. INTRODUÇÃO

Neste projeto você vai desenvolver a montagem dos componentes e a programação do Arduino para fazer um led piscar em um intervalo de tempo definido pelo programador. Conseguiremos compreender melhor a interação do Arduino, através de suas portas digitais, com componentes eletrônicos externos, como o LED e o resistor. Será usado um protoboard como base para a ligação entre o Arduino e os componentes.

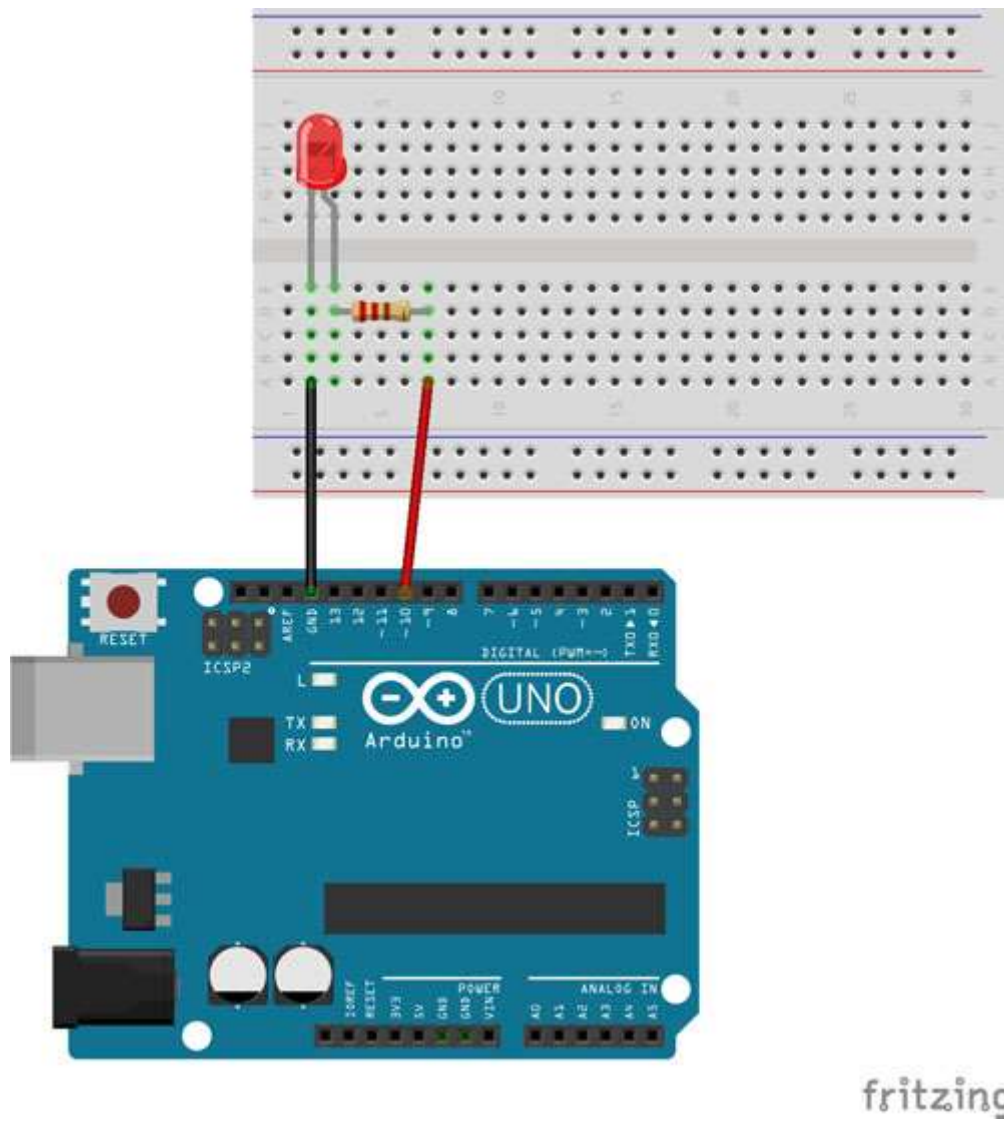
2. MATERIAL NECESSÁRIO

1. 1 ARDUINO;
2. 1 PROTOBOARD;
3. 1 LED;
4. 1 RESISTOR DE 220 OHMS;
5. 2 FIOS DE LIGAÇÃO (JUMPER)

3. INSTRUÇÕES

1. Primeiro coloque o led no protoboard;
2. Coloque o resistor na perna positiva do led (a perna maior)
3. Coloque a outra perna do resistor no pino digital 10 do Arduino;
4. A perna negativa do led (a perna menor) ligue no pino GND do arduino.

4. CONECTANDO OS COMPONENTES



5. DIGITE O CÓDIGO

6. ENTENDENDO O CÓDIGO

- Tudo após um **//** é um comentário dentro do programa.
- **void setup()** é um método e é executado uma vez assim que o arduino é ligado.
- **void loop()** também é um método que é executado, como diz o próprio nome, em loop enquanto o arduino estiver ligado.
- O comando **pinMode(10, OUTPUT)** define o pino digital 10 do arduino como um pino de saída. É o pino em que o led está ligado.
- O comando **digitalWrite(10,HIGH)** liga o led.
- O comando **digitalWrite(10,LOW)** desliga o led.
- O comando **delay(1000)** faz o arduino esperar 1 segundo antes de executar o próximo comando. Para fazer o arduino esperar 2 segundos use **delay(2000)**, para fazer esperar 5 segundos use **delay(5000)**, para fazer esperar 1 minuto use **delay(60000)** , e assim por diante.

7. MODIFIQUE E TESTE

1. Mude o valor da função delay. O que acontece?
2. Mude o número da variável do LED para 12 e mova o jumper do pino 10 para o pino 12. Veja o que ocorre.

II. PROJETO 2- SEMÁFORO SIMPLES

1. INTRODUÇÃO

Neste projeto você criará uma semáforo que irá do verde ao vermelho passando pelo amarelo ciclicamente.

2. MATERIAL NECESSÁRIO

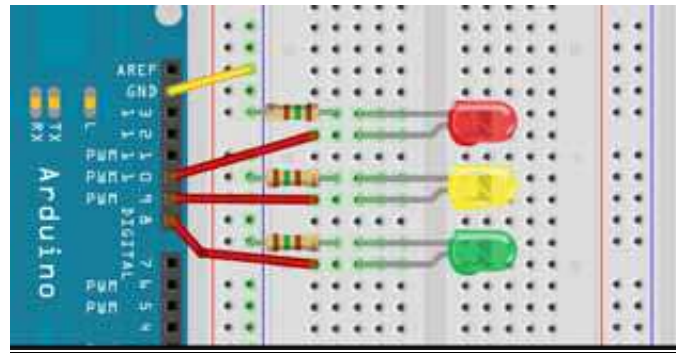
1. PROTOBOARD
2. 1 LED VERMELHO
3. 1 LED AMARELO
4. 1 LED VERDE
5. 3 RESISTORES DE 200 Ohms
6. FIOS JUMPER MACHO-MACHO

3. INSTRUÇÕES

1. Conecte seu circuito como mostra a figura do item 4. Você deve conectar 3 LEDs com o anodo de cada um (perna maior) indo para os pinos digitais 8, 9 e 10.

2. Leve um fio jumper do terra do Arduino (GND) para o barramento do terra no topo do protoboard.
3. Um fio terra vai do terminal catodo (perna menor) de cada LED para o barramento terra comum por meio de um resistor. O objetivo do resistor é limitar a corrente que passa pelo LED para que ele não queime.

4. CONECTANDO OS COMPONENTES



5. DIGITE O CÓDIGO

//PROJETO 2 – SEMÁFORO SIMPLES

//Nomeia os pinos 8, 9 E 10

int ledgreen = 8;

int ledyellow = 9;

int ledred = 10;

void setup() {

// Define os pinos como saida;

pinMode(ledgreen,OUTPUT);

pinMode(ledyellow,OUTPUT);

pinMode(ledred,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(ledgreen,HIGH); //ACENDE FAROL VERDE

delay(5000); //ESPERA 5 SEGUNDOS

digitalWrite(ledgreen,LOW); //APAGA O FAROL VERDE

digitalWrite(ledyellow,HIGH); //ACENDE FAROL AMARELO

delay(2000); //ESPERA 2 SEGUNDOS

digitalWrite(ledyellow,LOW); // APAGA O SINAL AMARELO

```
digitalWrite(ledred,HIGH); //ACENDE FAROL VERMELHO

delay(5000); //ESPERA 5 SEGUNDOS
digitalWrite(ledred,LOW); // APAGA O SINAL VERMELHO

}
```

III. PROJETO 3- SEMÁFORO COM SEMÁFORO DE PEDESTRES

1. INTRODUÇÃO

Agora você estenderá o projeto anterior para incluir um semáforo de pedestres

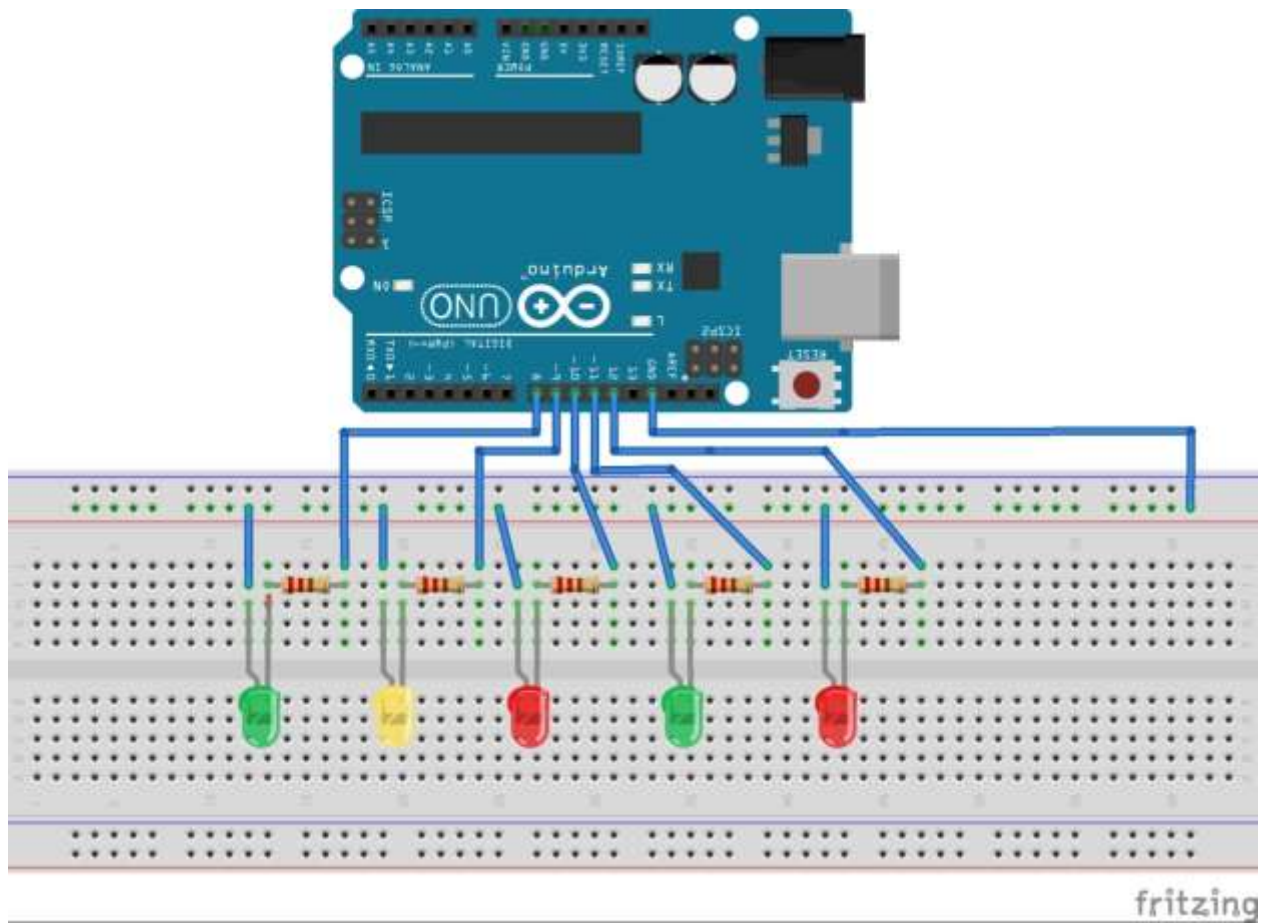
2. MATERIAL NECESSÁRIO

1. 1 PROTOBOARD
2. 2 LEDs VERMELHO
3. 1 LED AMARELO
4. 2 LEDs VERDE
5. 5 RESISTORES DE 200 Ohms
6. FIOS JUMPER MACHO-MACHO

3. INSTRUÇÕES

1. Conecte seu circuito como mostra a figura do item 4. Você deve conectar 5 LEDs com o anodo de cada um (perna maior) indo para os pinos digitais 8, 9, 10, 11 e 12, através do resistor de 220 Ohms.
2. Leve um fio jumper do terra do Arduino (GND) para o barramento do terra no topo do protoboard.
3. Um fio terra vai do terminal catodo (perna menor) de cada LED para o barramento terra comum.
4. Perceba que desta vez ligamos os resistores aos pinos do Arduino ao invés de ligá-los ao GND para mostrar que ambas as formas funcionam.

4. CONECTANDO OS COMPONENTES



5. DIGITE O CÓDIGO

```
//nomeia os pinos 8, 9, 10, 11 e 12
```

```
int ledgreen = 8;
```

```
int ledyellow = 9;
```

```
int ledred = 10;
```

```
int pedgreen = 11;
```

```
int pedred = 12;
```

```
void setup() {
```

```
    // define os pinos como saida;
```

```
    pinMode(ledgreen,OUTPUT);
```

```
    pinMode(ledyellow,OUTPUT);
```

```
    pinMode(ledred,OUTPUT);
```

```
    pinMode(pedgreen,OUTPUT);
```

```
    pinMode(pedred,OUTPUT);
```

```
}
```

```

void loop() {
  // put your main code here, to run repeatedly:
  //acende verde
  digitalWrite(ledgreen,HIGH);
  digitalWrite(pedgreen,LOW);
  digitalWrite(pedred,HIGH);
  delay(5000);
  digitalWrite(ledgreen,LOW);
  //acende amarelo
  digitalWrite(ledyellow,HIGH);
  digitalWrite(pedgreen,LOW);
  digitalWrite(pedred,HIGH);
  delay(2000);

  //acende vermelho
  digitalWrite(ledyellow,LOW);
  digitalWrite(ledred,HIGH);
  digitalWrite(pedgreen,HIGH);
  digitalWrite(pedred,LOW);
  delay(5000);
  digitalWrite(ledred,LOW);

}

```

6. MODIFIQUE E TESTE

1. Mude o tempo que o sinal fica em amarelo para 500ms.
2. Mude o tempo que os sinais vermelho e verde ficam ligados para 10 segundos cada.

IV. PROJETO 4 – SINALIZADOR DE CÓDIGO MORSE

1. INTRODUÇÃO

Para este projeto, voce reutilizara o circuito que preparamos na prática 1, mas voce utilizará um codigo diferente para fazer com que o LED sinalize as letras S.O.S., sinal de socorro internacional em codigo Morse. O codigo Morse é um tipo de codificacao de caracteres que transmite letras e numeros utilizando padroes de ligado e desligado. Portanto, ele é muito adequado ao seu sistema digital, uma vez que voce pode acender e apagar o LED no padrao necessario para soletrar uma palavra ou série de caracteres. Nesse caso, o padrao S.O.S. é formado de tres pontos (sinais curtos), seguidos por tres tracos (sinais longos), seguidos por tres pontos novamente.

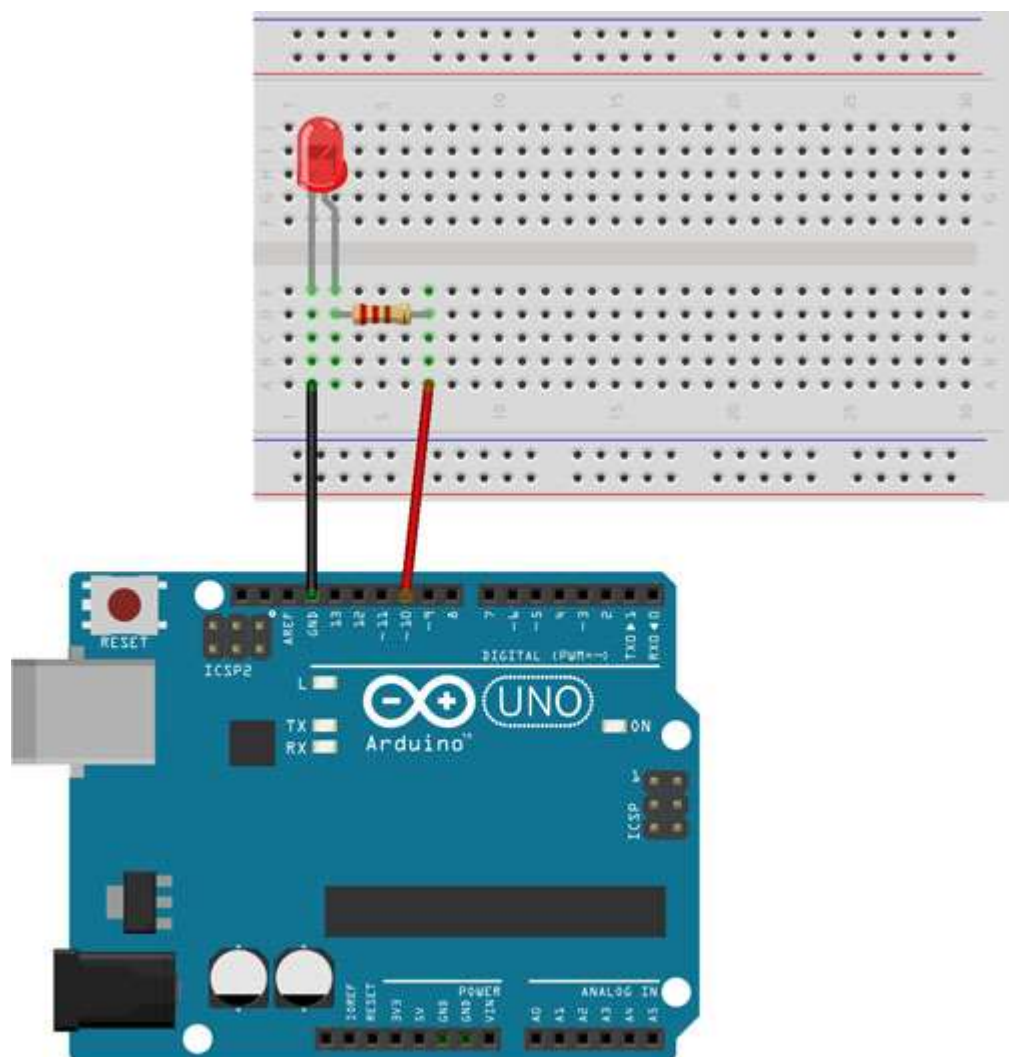
2. MATERIAL NECESSÁRIO

1. 1 ARDUINO;
2. 1 PROTOBOARD;
3. 1 LED;
4. 1 RESISTOR DE 220 OHMS;
5. 2 FIOS DE LIGAÇÃO (JUMPER)

3. INSTRUÇÕES

Para piscar o LED, acendendo e apagando nesse padrao e sinalizando S.O.S., utilize o codigo da seção 5 a seguir.

4. CONECTANDO OS COMPONENTES



5. DIGITE O CÓDIGO

```
// LED conectado ao pino 10
int ledPin = 10;
// executa uma vez, quando o sketch inicia
void setup() {
  // define o pino como saída
  pinMode(ledPin, OUTPUT);
}
// executa repetidas vezes
void loop() {
  // 3 pontos
  for (int x=0; x<3; x++) {
    digitalWrite(ledPin, HIGH); // acende o LED
    delay(150); // espera 150ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
  }

  // espera 100ms para marcar o intervalo entre as letras
  delay(100);
  // 3 traços
  for (int x=0; x<3; x++) {
    digitalWrite(ledPin, HIGH); // acende o LED
    delay(400); // espera 400ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
  }

  // espera 100ms para marcar o intervalo entre as letras
  delay(100);
  // 3 pontos novamente
  for (int x=0; x<3; x++) {
    digitalWrite(ledPin, HIGH); // acende LED
    delay(150); // espera 150ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
  }

  // espera 5 segundos antes de repetir o sinal de SOS
  delay(5000);
}
```

6. MODIFIQUE E TESTE

1. Dada a tabela do código morse envie a sigla ETEP.

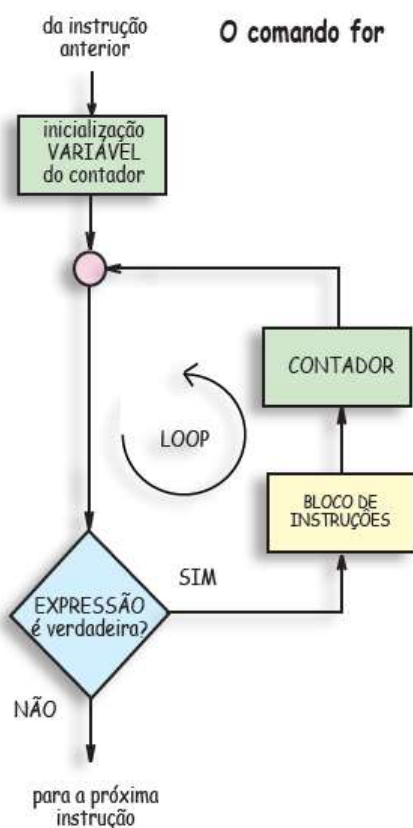
A	.-	J	.-.-.-	S	...	2	..-.-
B	-...	K	-.-	T	-	3	...-.-
C	-.-.-	L	.-...	U	...	4-
D	...	M	--	V-	5
E	.	N	--.	W	.-.-	6	-....
F	...-	O	---	X-	7	-....
G	...-	P-	Y-	8	-----
H	Q-	Z-	9	-----
I	..	R	...	1	0	-----

V. APÊNDICE

A instrução **for** age como um contador e deve ter uma variável de controle que deve ser previamente inicializada com um tipo e um valor. A sua sintaxe é a seguinte:

```
for (variável; expressão; incremento) {  
bloco de instruções;  
}
```

A **variável** é inicializada normalmente com 0 ou 1; o parâmetro **expressão** deve conter o valor máximo (ou mínimo) que o contador deve alcançar; e **incremento** é o valor que será incrementado (ou decrementado) da variável cada vez que o bloco de instruções é executado. Observe que cada parâmetro entre parênteses é separado por ponto e vírgula.



Exemplo:

```
for (int i = 0; i <= 10; i++) {  
    println (contador);  
    delay(1000);  
}
```

Nesse exemplo a variável de controle *i* do contador é inicializada com 0; o contador é testado e se o valor nele acumulado for menor que 10 seu valor é enviado para o Terminal, e depois de 1 segundo, o contador é incrementado e seu novo valor é testado novamente.