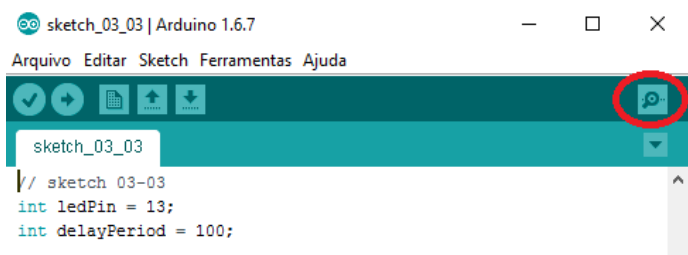


Serial Monitor

Sua finalidade é atuar como um canal de comunicação entre o computador e o Arduino



- Função que será usada para isto:

Serial.println

Exemplo 3

Faça o Arduino mostrar na tela do computador o resultado de uma soma de 2 variáveis

```
// sketch 03-04
void setup()
{
  Serial.begin(9600);
  int a = 2;
  int b = 2;
  int c = a + b;
  Serial.println(c);
}

void loop()
{}
```



Obs.: Lembre-se de colocar no monitor serial a mesma porta que foi colocado no código.

Operadores

Operador	Significado	Exemplos	Resultado
<	Menor do que	9 < 10 10 < 10	verdadeiro falso
>	Maior do que	10 > 10 10 > 9	falso verdadeiro
<=	Menor do que ou igual a	9 <= 10 10 <= 10	verdadeiro verdadeiro
>=	Maior do que ou igual a	10 >= 10 10 >= 9	verdadeiro verdadeiro
==	Igual a	9 == 9	verdadeiro
!=	Não igual a, diferente de	9 != 9	falso

Comando IF

Se houver uma condição com a necessidade de verificação se uma condição é verdadeira ou falsa, o comando IF é uma boa solução.

Exemplo 4

```
// sketch 03-06
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
  delayPeriod = delayPeriod + 100;
  if (delayPeriod > 3000)
  {
    delayPeriod = 100;
  }
}
```

o pino do LED para que seja uma saída.

p.



d.

de 3.000, torne-o novamente igual a 100.

FOR

É uma estrutura de repetição, parecido com o loop do Arduino, porém com este comando é possível prender o programa em um determinado laço até que uma contagem tenha sido atingida

Exemplo 5

Escreva um programa em um sketch que faz um LED piscar 20 vezes, em seguida fazer uma pausa de 3 segundos e então começar tudo de novo.

Resolução

Forma Simples

```
// sketch 03-07
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);

  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);

  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);

  // repita as 4 linhas anteriores mais 17 vezes
  delay(3000);
}
```

Forma Eficiente

```
// sketch 03-08
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  for (int i = 0; i < 20; i ++){
    digitalWrite(ledPin, HIGH);
    delay(delayPeriod);

    digitalWrite(ledPin, LOW);
    delay(delayPeriod);
  }
  delay(3000);
}
```


Problema que podem ocorrer na forma Eficiente

A função loop vai precisar de muito tempo para executar o laço FOR. Desta forma pode ocorrer de:

- Não conseguir ler um botão se ele for pressionado e o programa estiver dentro de um laço FOR;
- Não receber nenhuma comunicação serial, se ela for enviada no momento que o programa estiver dentro do laço FOR;
- Qualquer outra coisa que tenha sido acionado no Arduino e estiver dentro do laço FOR pode ocorrer de o programa não executar.

Solução:

A função loop deve ser executada tão rapidamente quanto possível para que ela seja executada o maior número possível de vezes.


Resolução Melhorada

Forma Eficiente

```
// sketch 03-08
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}


void loop()
{
  for (int i = 0; i < 20; i ++){
    digitalWrite(ledPin, HIGH);
    delay(delayPeriod);
    digitalWrite(ledPin, LOW);
    delay(delayPeriod);
  }
  delay(3000);
}
```



Forma Eficiente Melhorada

```
// sketch 03-09
int ledPin = 13;
int delayPeriod = 100;
int count = 0;
void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
  count ++;
  if (count == 20)
  {
    count = 0;
    delay(3000);
  }
}
```



While

Também é um laço de repetição, que pode ser usada no lugar do comando FOR.

- A diferença que a verificação da contagem é feita antes de entrar no programa
- Para que o programa permaneça dentro do laço de repetição, a expressão do comando while deve ser verdadeira.

```
int i = 0;
while (i < 20)
{
    digitalWrite (ledPin, HIGH);
    delay(delayPeriod);
    digitalWrite (ledPin, LOW);
    delay(delayPeriod);
    i ++;
}
```

Funções

Desenvolva Funções no algoritmo para facilitar sua organização.

Exemplos de Funções:

- digitalWrite
 - Delay
 - Setup
 - loop
- } Funções interna

O que é uma Função:

Uma Função é algo um pouco parecido com um programa dentro de um programa. Usado para “empacotar” algumas coisas que você deseja realizar.

Como Funciona:

A função poderá ser chamada de qualquer lugar do seu sketch e contém as suas próprias:

- Variáveis
- Lista de comando

Quando os seus comandos terminam de ser efetuados, a execução do sketch continua no ponto imediatamente após a linha no código em que a função foi chamada.

Exemplo 6

Faça um LED piscar 20 vezes utilizando para isto uma função com a atribuição de acender e apagar o LED.

Resolução

```
// sketch 04-01
int ledPin = 13;
int delayPeriod = 250;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  for (int i = 0; i < 20; i ++)
  {
    flash();
    delay(3000);
  }
}

void flash()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
}
```

Flash()

20X

Depois de 20X entrando em flash, finalmente o código vai para a próxima linha em delay(3000).