

## Placa Arduino MEGA 2560 – IDE – funções de I/O

A IDE (ambiente para desenvolvimento integrado) de desenvolvimento das aplicações para o Arduino pode ser obtida em <https://www.arduino.cc/en/software>.

Uma síntese da linguagem utilizada pode ser obtida em <https://www.arduino.cc/reference/en/>.

Nesta nota iremos tratar especificamente das funções de entrada e saída, gerando códigos para ilustrar a aplicação das funções, apresentando o protótipo em placa de *proto-board* do circuito e o respectivo esquema elétrico. A nota prossegue com a apresentação do uso da pinagem apresentada na nota 1.

Não pretendemos explorar detalhadamente ainda outros aspectos da linguagem da IDE. Apresentamos somente os aspectos básicos de sintaxe e da semântica da IDE para a compreensão dos exemplos realizados.

Lembrando que a placa Arduino MEGA 2560 é baseada no microcontrolador Atmel 2560, com a seguinte pinagem:

- 8 (oito) entradas e saídas para alimentação;
- 54 (cinquenta e quatro) entradas e saídas digitais;
- 16 (dezesesseis) entradas analógicas.

## Funções de entradas e saídas digitais

### 1. Função *pinMode()*

#### - Descrição:

Configura o pino especificado para se comportar como uma entrada ou uma saída. É possível habilitar ou desabilitar os resistores internos de *pull-up* da entrada – não discutiremos esse ponto por enquanto.

#### - Sintaxe:

*pinMode(pino, modo)*

#### - Parâmetros:

*pino*: o número do pino a ser configurado – entre 0 e 53.

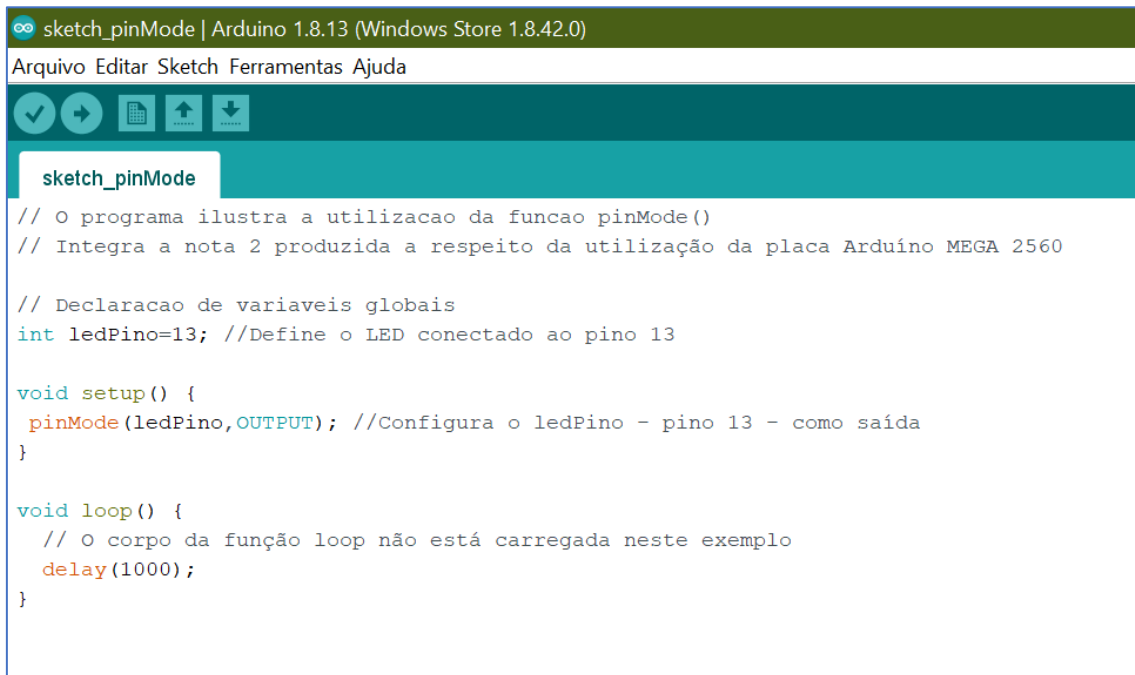
*modo*: INPUT (entrada), INPUT\_PULLUP (entrada com configuração dos resistores *pull-up*) ou OUTPUT – saída de até 40 mA.

#### - Retorno:

Não retorna valor.

### - Exemplo

A Figura 1 ilustra exemplo da utilização da função `pinMode()`.



```
sketch_pinMode | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

sketch_pinMode
// O programa ilustra a utilizacao da funcao pinMode()
// Integra a nota 2 produzida a respeito da utilização da placa Arduino MEGA 2560

// Declaracao de variaveis globais
int ledPino=13; //Define o LED conectado ao pino 13

void setup() {
  pinMode(ledPino,OUTPUT); //Configura o ledPino - pino 13 - como saida
}

void loop() {
  // O corpo da função loop não está carregada neste exemplo
  delay(1000);
}
```

Figura 1 – Exemplo de código na IDE da placa Arduino MEGA 2560 de uso da função `pinMode()`

### - Comentários:

- a. Semelhante à linguagem C, as funções na IDE têm a forma

```
tipo-retorno nome-função (declaração de argumentos)
{
    Declarações e comandos
}
```

A palavra reservada `void` indica que a função retorna vazio.

- b. A dupla barra `//` indica uma linha de comentário simples  
A função `setup()` é chamada quando um *sketch* inicia. É utilizada para inicializar variáveis, modos dos pinos, bibliotecas etc. É executada somente uma vez, após a inicialização ou reinicialização da placa.
- c. A função `loop()` controla ativamente a placa para que a inicialização do *sketch* ocorra.
- d. A função `delay(tempo)` pausa a execução do programa durante o *tempo* especificado como parâmetro – em milissegundos.

## 2. Função *digitalWrite()*

### - Descrição:

A função escreve em um pino digital especificado. A tensão do pino será definida para o valor 5V ou 0V.

### - Sintaxe:

*digitalWrite(pino,valor)*

### - Parâmetros:

*pino*: o número do pino de onde será lido o sinal – 0 a 53.

*valor*: 5 V (HIGH) ou 0 V (LOW)

### - Retorno:

Não retorna valor.

### - Exemplo:

A Figura 2 apresenta um exemplo do uso da função *digitalWrite()*.



```
// O programa ilustra a utilizacao da funcao digitalWrite()
// Integra a nota 2 produzida a respeito da utilização da placa Arduino MEGA 2560

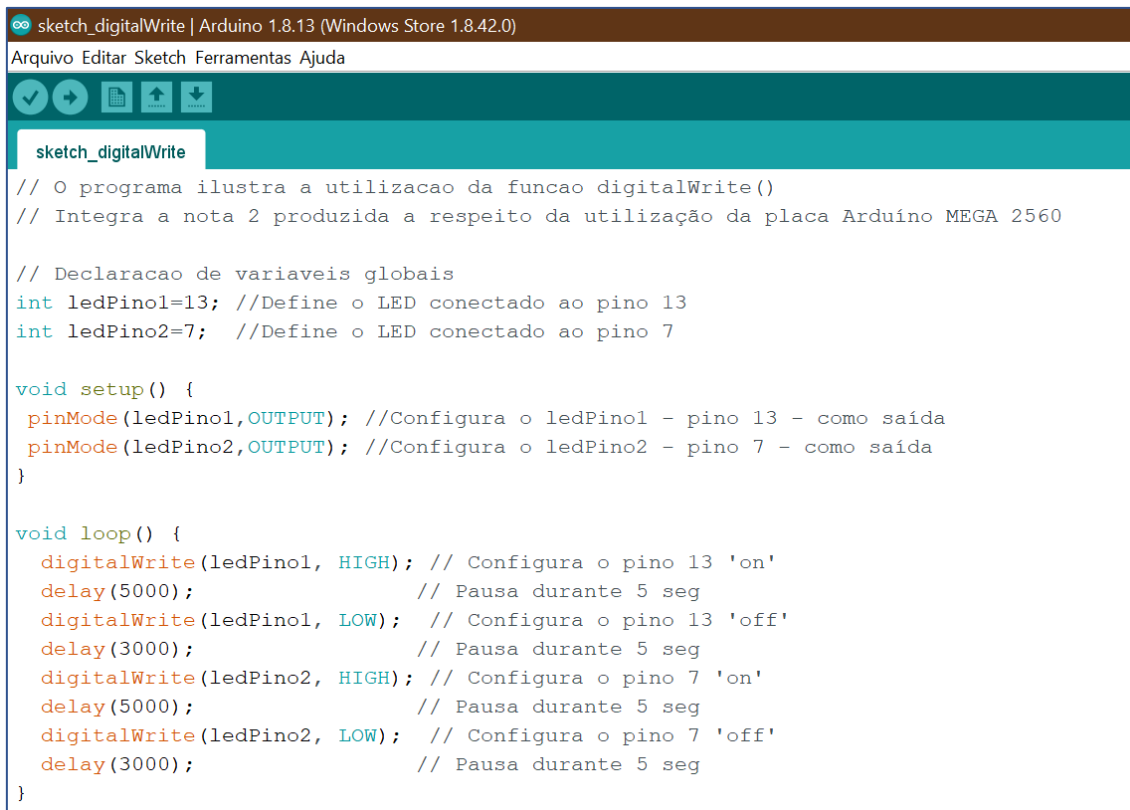
// Declaracao de variaveis globais
int ledPino=13; //Define o LED conectado ao pino 13

void setup() {
  pinMode(ledPino,OUTPUT); //Configura o ledPino - pino 13 - como saída
}

void loop() {
  digitalWrite(ledPino, HIGH); // Configura o pino 13 'on'
  delay(5000);                // Pausa durante 5 seg
  digitalWrite(ledPino, LOW); // Configura o pino 13 'off'
  delay(5000);                // Pausa durante 5 seg
  digitalWrite(ledPino, HIGH); // Configura o pino 13 'on'
  delay(3000);                // Pausa durante 3 seg
}
```

Figura 2 – Exemplo 1 de código na IDE da placa Arduino MEGA 2560 de uso da função *digitalWrite()*

A Figura 3 apresenta um exemplo que incorpora mais um LED à placa.



```
sketch_digitalWrite | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

sketch_digitalWrite

// O programa ilustra a utilizacao da funcao digitalWrite()
// Integra a nota 2 produzida a respeito da utilização da placa Arduino MEGA 2560

// Declaracao de variaveis globais
int ledPino1=13; //Define o LED conectado ao pino 13
int ledPino2=7;  //Define o LED conectado ao pino 7

void setup() {
  pinMode(ledPino1,OUTPUT); //Configura o ledPino1 - pino 13 - como saída
  pinMode(ledPino2,OUTPUT); //Configura o ledPino2 - pino 7 - como saída
}

void loop() {
  digitalWrite(ledPino1, HIGH); // Configura o pino 13 'on'
  delay(5000);                  // Pausa durante 5 seg
  digitalWrite(ledPino1, LOW);  // Configura o pino 13 'off'
  delay(3000);                  // Pausa durante 5 seg
  digitalWrite(ledPino2, HIGH); // Configura o pino 7 'on'
  delay(5000);                  // Pausa durante 5 seg
  digitalWrite(ledPino2, LOW);  // Configura o pino 7 'off'
  delay(3000);                  // Pausa durante 5 seg
}
```

Figura 3 – Exemplo 2 de código na IDE da placa Arduino MEGA 2560 de uso da função digitalWrite()

#### - Comentários:

- Os códigos de ambos os exemplos, em linhas gerais, consistem da declaração das variáveis globais (ledPino 1 e ledPino2); configuração dos pinos; e escrita dos valores HIGH e LOW nos pinos (pino 13 no exemplo 1; pino 7 e pino 13 no exemplo 2), com pausa entre as operações.
- O código do exemplo 1 (Figura 2) faz com que, alternadamente, o LED conectado ao pino 13 acenda e apague, mantendo-o ON ou OFF nos tempos definidos pela função *pause()*.
- O código do exemplo 2 (Figura 3) faz com os LEDs conectados aos pinos 13 e 7 acendam e apaguem alternadamente. Na condição ON, cada LED permanece aceso durante 5 segundos; na condição OFF cada LED permanece apagado durante 3 segundos.
- O protótipo do circuito correspondente ao código da Figura 2 (exemplo 1) montado em placa de *protoboard* está apresentado na Figura 4. A placa está alimentada pela saída USB do *desktop* na entrada de alimentação USB da placa Arduino. O pino 13 está conectado ao anodo do LED. O catodo do LED está conectado ao terra (*ground*) da placa de *protoboard* e da placa – todos utilizando o mesmo referencial de potencial.

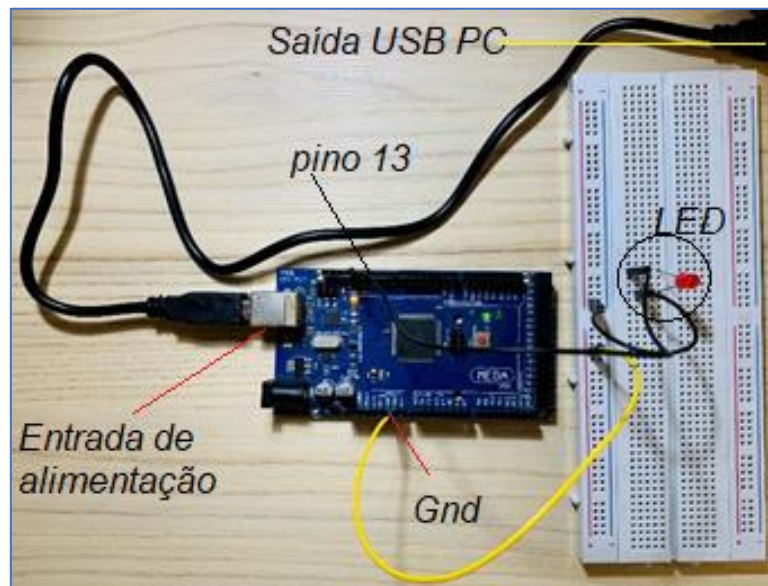


Figura 4 – Protótipo em placa de protoboard para teste do circuito do exemplo de uso da função `digitalWrite()`

- e. O esquema elétrico relativo ao exemplo 1 é apresentado na Figura 5, elaborado utilizando-se a ferramenta EasyEDA. Utiliza-se o ARDUÍNO MEGA obtido da biblioteca da ferramenta proposto por Marcos Postali.

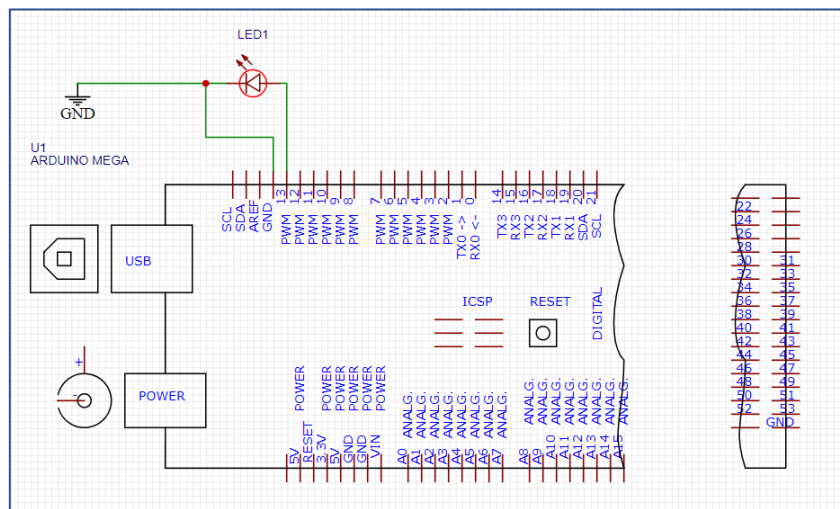


Figura 5 – Esquema elétrico do exemplo 1 de uso da função `digitalWrite()`

- f. O esquema elétrico relativo ao exemplo 2 é apresentado na Figura 6.

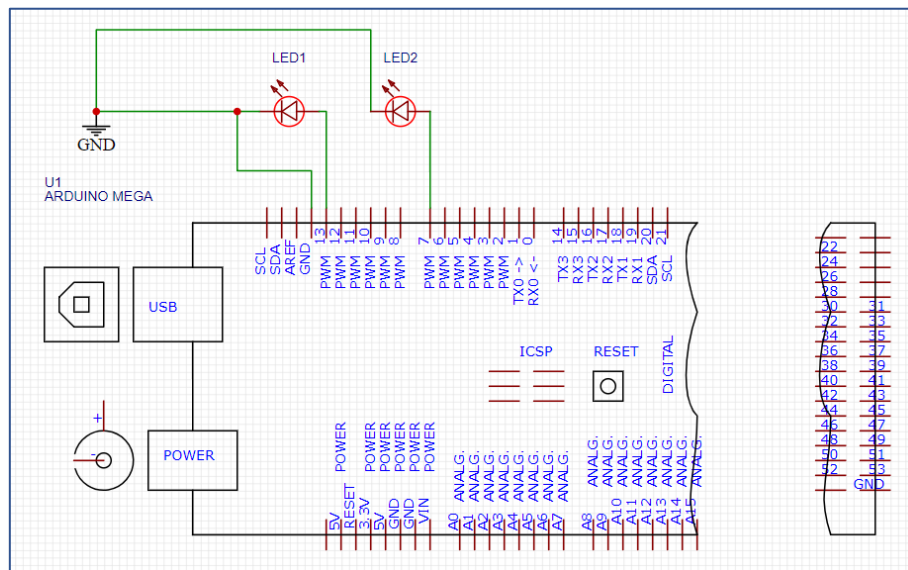


Figura 6 - Esquema elétrico do exemplo 2 de uso da função `digitalWrite()`

- g. É importante lembrar que a corrente de saída é de 40 mA. Componentes que precisam operar com uma corrente maior do que essa podem não operar adequadamente.

### 3. Função `digitalRead()`

#### - Descrição:

A função lê um sinal de um pino digital especificado.

#### - Sintaxe:

`digitalRead(pino)`

#### - Parâmetros:

*pino*: o número do pino de onde será lido o sinal – 0 a 53.

#### - Retorno:

HIGH (5V) ou LOW (0V).

#### - Exemplo:

A Figura 7 apresenta um exemplo da utilização da função `digitalRead()`.



```
sketch_digitalRead | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

sketch_digitalRead
// O programa ilustra a utilizacao da funcao digitalRead()
// Integra a nota 2 produzida a respeito da utilização da placa Arduino MEGA 2560

// Declaracao de variaveis globais
int ledPinol=11; //Define o LED conectado ao pino 11
int fontePino2=7; //Define o circuito de fonte de sinal conectado ao pino 7
int val=0;        //Define uma variável para armazenar o dado de entrada - por exemplo, um sinal de uma fonte

void setup() {
  pinMode(ledPinol,OUTPUT); //Configura o ledPinol - pino 11 - como saída
  pinMode(fontePino2,INPUT); //Configura o fontePino2 - pino 7 - como entrada
}

void loop() {
  val=digitalRead(fontePino2); //Carrega o valor da fonte conectada ao pino 7 na variável val
  digitalWrite(ledPinol, val); // Carrega o valor de val na saída conectada - LED - ao pino 11
}
```

Figura 7 - Exemplo de código na IDE da placa Arduino MEGA 2560 de uso da função digitalRead()

- Comentários:

- O código do exemplo da Figura 7 lê um sinal pela entrada digital do pino 7. O valor lido é carregado na saída digital do pino 11. Como nos exemplos anteriores, utiliza-se a estrutura de declaração das variáveis, inicialização do sketch e execução da rotina.
- O protótipo do circuito correspondente ao código da Figura 7 montado em placa de *protoboard* está apresentado na Figura 8.

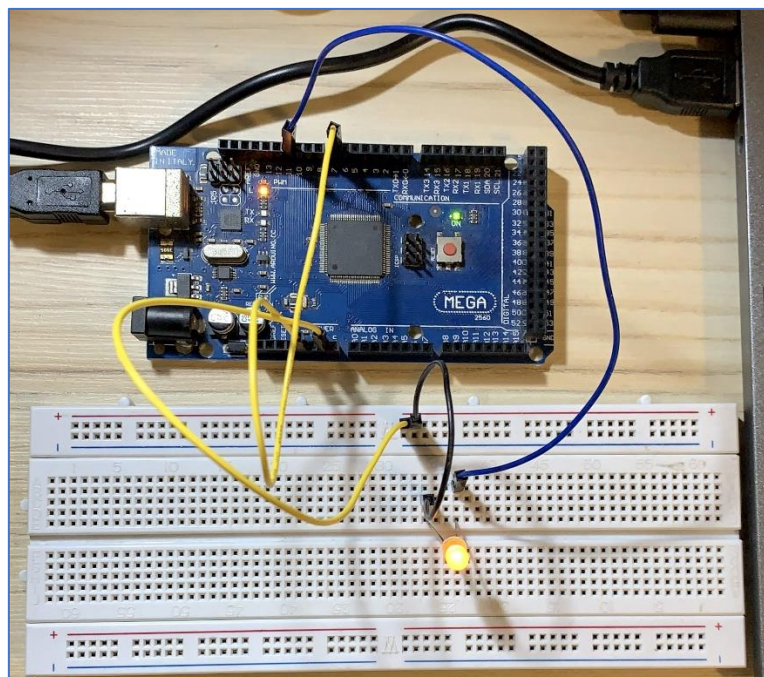


Figura 8 – Protótipo em placa de protoboard para teste do circuito do exemplo de uso da função digitalRead()

- O esquema elétrico relativo ao exemplo é apresentado na Figura 9. O esquema evidencia o que não está tão claro no protótipo. Foram

utilizadas a tensão de 5V da saída da placa do Arduíno MEGA e a saída de Gnd para simular o sinal de entrada no pino 7.

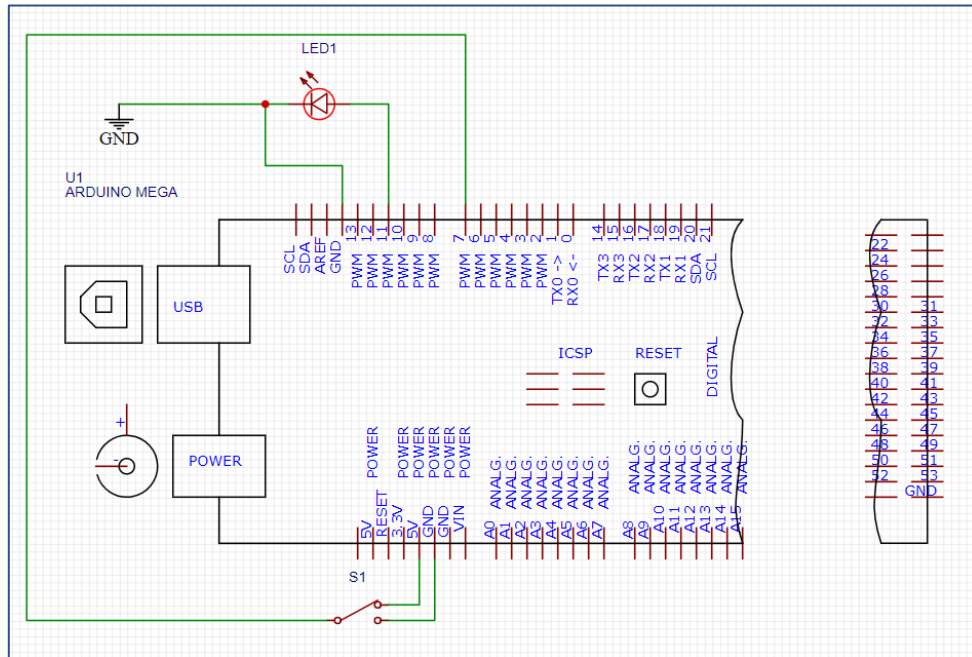


Figura 9 – Esquema elétrico do exemplo de uso da função `digitalRead()`

## Funções de saídas analógicas

#### 4. Função *analogWrite()*

- Descrição:

A função gera um sinal de saída no padrão PWM (*Pulse Widht Modulation*).

- Sintaxe:

*analogWrite(pino, valor)*

- Parâmetros:

*pino*: o número do pino em que será gerado o sinal de saída. No caso da placa Arduino MEGA, pinos 2-13, 44-46.

*valor*: entre 0 e 255.

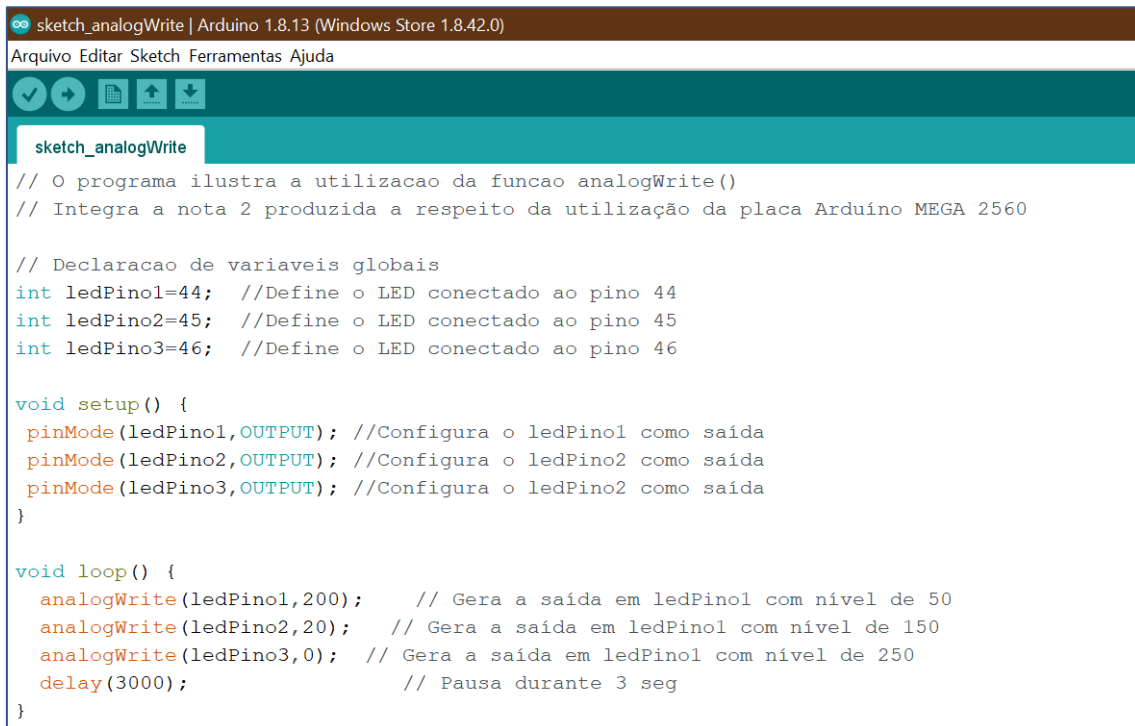
- Retorno:

Não retorna valor.



- Exemplo:

A Figura 10 apresenta o exemplo de código da função *analogWrite()*.



```
sketch_analogWrite | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

sketch_analogWrite

// O programa ilustra a utilizacao da funcao analogWrite()
// Integra a nota 2 produzida a respeito da utilização da placa Arduino MEGA 2560

// Declaracao de variaveis globais
int ledPino1=44; //Define o LED conectado ao pino 44
int ledPino2=45; //Define o LED conectado ao pino 45
int ledPino3=46; //Define o LED conectado ao pino 46

void setup() {
  pinMode(ledPino1,OUTPUT); //Configura o ledPino1 como saída
  pinMode(ledPino2,OUTPUT); //Configura o ledPino2 como saída
  pinMode(ledPino3,OUTPUT); //Configura o ledPino2 como saída
}

void loop() {
  analogWrite(ledPino1,200); // Gera a saída em ledPino1 com nível de 50
  analogWrite(ledPino2,20); // Gera a saída em ledPino1 com nível de 150
  analogWrite(ledPino3,0); // Gera a saída em ledPino1 com nível de 250
  delay(3000); // Pausa durante 3 seg
}
```

Figura 10 - Exemplo de código na IDE da placa Arduino MEGA 2560 de uso da função *analogWrite()*

- Comentários:

- O código do exemplo da Figura 10 gera um sinal PWM nas saídas digitais dos pinos 44, 45 e 46. Os sinais gerados são, respectivamente, do nível 2, 20 e 250 – lembrando que a escala se refere aos 255 níveis possíveis da quantização (resolução) do sinal produzido. O sinal digital possui uma porcentagem do sinal alto em relação ao baixo durante um ciclo de pulso (*duty-cycle*) dependendo do valor definido. Como nos exemplos anteriores, utiliza-se a estrutura de declaração das variáveis, inicialização do sketch e execução da rotina.
- O protótipo do circuito correspondente ao código do exemplo montado em placa de *protoboard* está apresentado na Figura 11. A imagem busca evidenciar os diferentes níveis de luminosidade dos diodos, que ocorre em decorrência nos níveis de sinal gerados (2, 20 e 250) pelo sinal PWM.

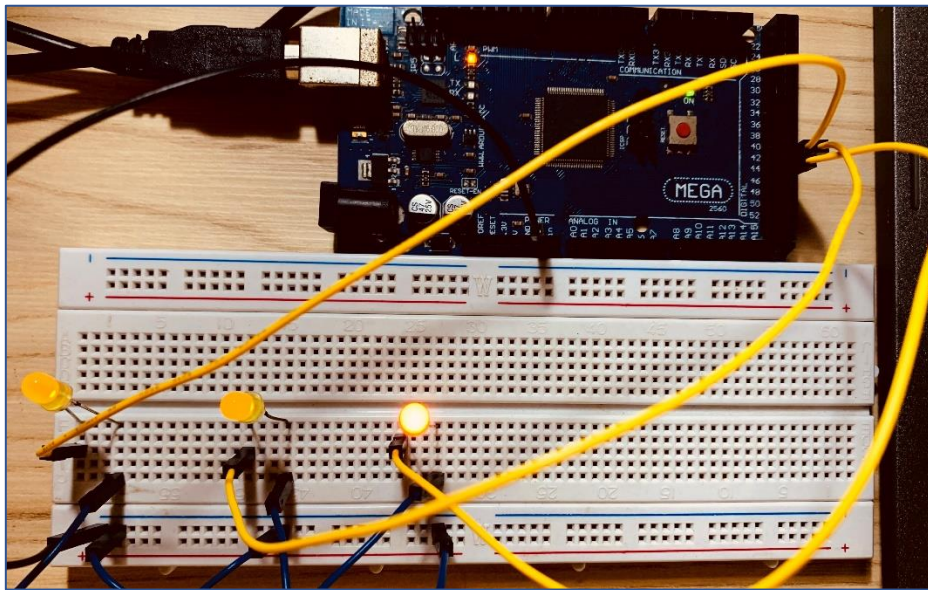


Figura 11 - Protótipo em placa de protoboard para teste do circuito do exemplo de uso da função `analogWrite()`

c. O esquema elétrico relativo ao exemplo é apresentado na Figura 12.

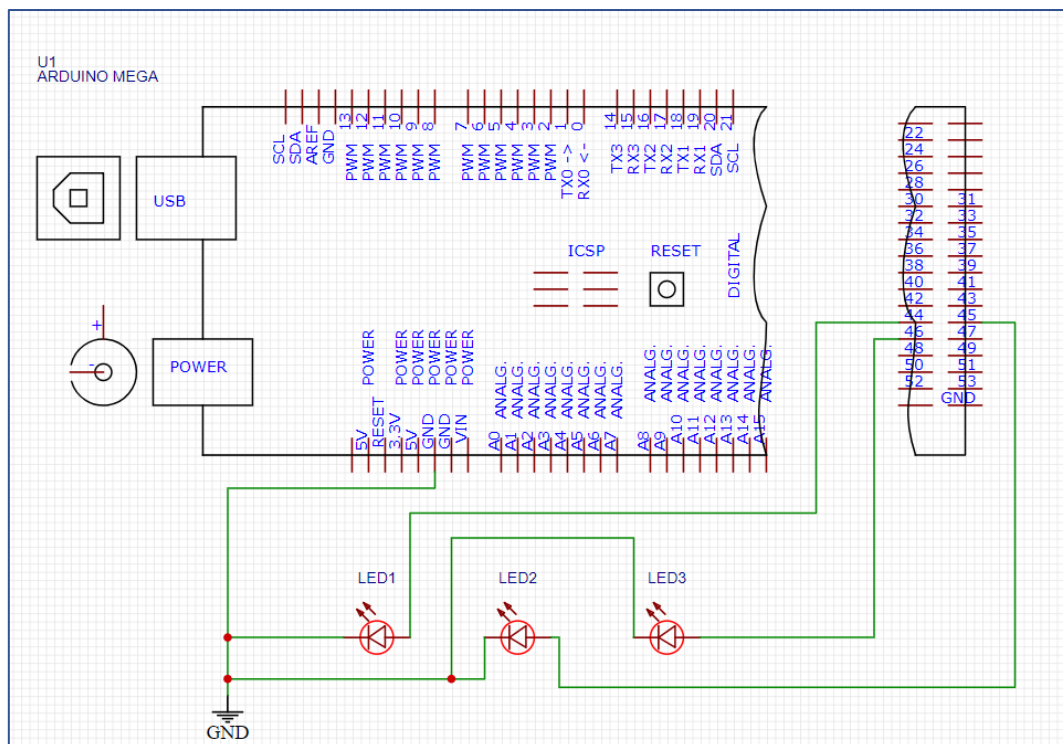


Figura 12 - Esquema elétrico do exemplo de uso da função `analogWrite()`

## 5. Função `analogRead()`

### - Descrição:

Lê o valor do pino analógico especificado. Lembrando que as entradas analógicas são convertidas em digitais com de 10 bits.

- Sintaxe:

*analogRead(pino)*

- Parâmetros:

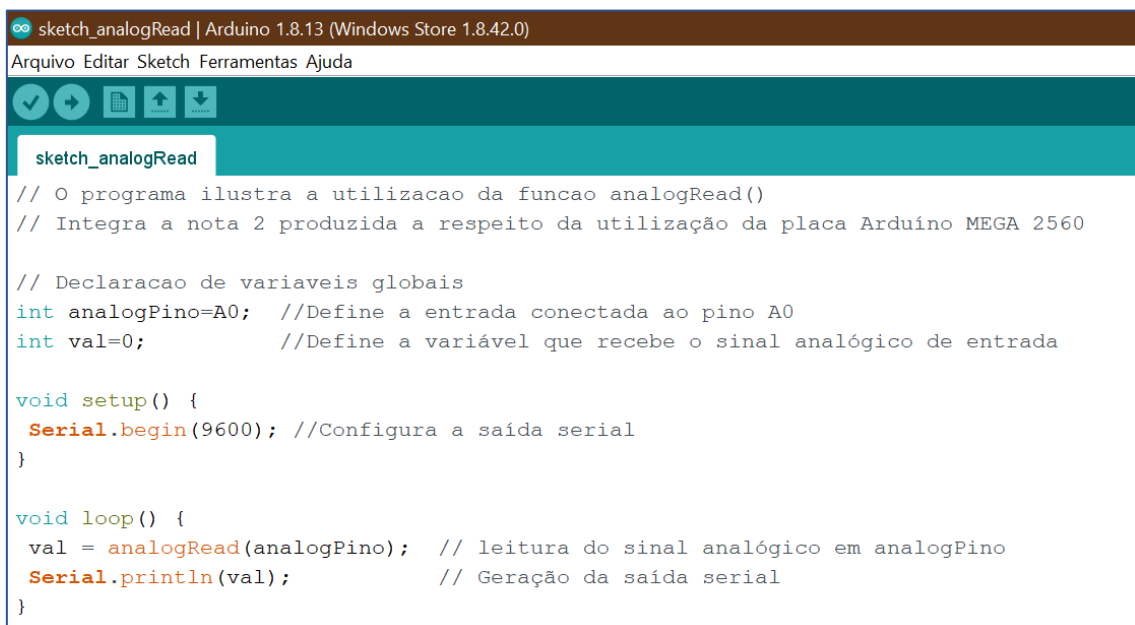
*pino*: o número do pino em que será lido o sinal analógico. No caso da placa Arduino MEGA, pinos A0-A15.

- Retorno:

A leitura analógica no pino, limitado à resolução do conversor analógico ao digital de 10 bits.

- Exemplo:

A Figura 13 apresenta o exemplo de código com o uso da função *analogRead()*.



```
// sketch_analogRead | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

sketch_analogRead

// O programa ilustra a utilizacao da funcao analogRead()
// Integra a nota 2 produzida a respeito da utilização da placa Arduino MEGA 2560

// Declaracao de variaveis globais
int analogPino=A0; //Define a entrada conectada ao pino A0
int val=0;          //Define a variável que recebe o sinal analógico de entrada

void setup() {
  Serial.begin(9600); //Configura a saída serial
}

void loop() {
  val = analogRead(analogPino); // leitura do sinal analógico em analogPino
  Serial.println(val);          // Geração da saída serial
}
```

Figura 13 - Exemplo de código na IDE da placa Arduino MEGA 2560 de uso da função *analogRead()*

- Comentários:

- O código do exemplo da Figura 13 lê um sinal na entrada analógica A0 e carrega o sinal lido com uma resolução de 10 bits na variável *val*. Como nos exemplos anteriores, utiliza-se a estrutura de declaração das variáveis, inicialização do sketch e execução da rotina.
- O código utiliza uma classe da linguagem da interface IDE de comunicação serial da placa Arduino MEGA 2560. Lembrando que os pinos 0(RX), 1(TX) - Serial; 19(RX), 18(TX) – Serial 1; 17(RX), 16(TX) – Serial 2; e 15(RX), 14(TX) – Serial 3 da placa são destinados a comunicação serial. Obs. O nível lógico das saídas é de 5V, logo não deve se conectar diretamente essas saídas em uma porta RS-232, que opera com 12V, sob pena de poder danificar a placa.

- c. A função *Serial.begin(velocidade)* configura a taxa em bits por segundo (*bauds*) para a transmissão.
- d. A função *Serial.println(valor)* imprime o dado na porta serial no formato ASCII, seguido de um caractere de retorno de transporte (ASCII13 ou 'r') e um caractere de nova linha ('\n').
- e. Aprofundaremos as operações de comunicação em outra nota sobre o Arduino.

A tabela apresentada a seguir contém uma síntese das funções I/O descritas nesta nota.

Função	Descrição	Sintaxe	Parâmetros
pinMode()	Configura o pino especificado para se comportar como uma entrada ou uma saída.	pinMode(pino,modo)	0-53, INPUT, INPUT_PULLUP ou OUTPUT
digitalWrite()	Escreve em um pino digital especificado	digitalWrite(pino,valor)	0-53, HIGH ou LOW
digitalRead()	Lê um sinal de um pino digital especificado. Retorna LOW ou HIGH.	digitalRead(pino)	0-53
analogWrite()	Gera um sinal de saída no padrão PWM	analogWrite(pino,valor)	2-13 e 44-46, 0-255
analogRead()	Lê o valor do pino analógico especificado. Retorna o valor da entrada em 10 bits.	analogRead(pino)	A0-A15