

Placa Arduino MEGA 2560 – IDE – ambiente e linguagem de programação

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

O ambiente da IDE está bem descrito em [Arduino - Environment](#)

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

Nesta nota iremos tratar especificamente de tópicos específicos do ambiente e da linguagem da IDE. A nota prossegue com a apresentação do uso da pinagem apresentada na nota 1, aplicando as funções de entrada e saída apresentadas na nota 2.

Ambiente da IDE

A figura apresenta a tela principal da IDE.

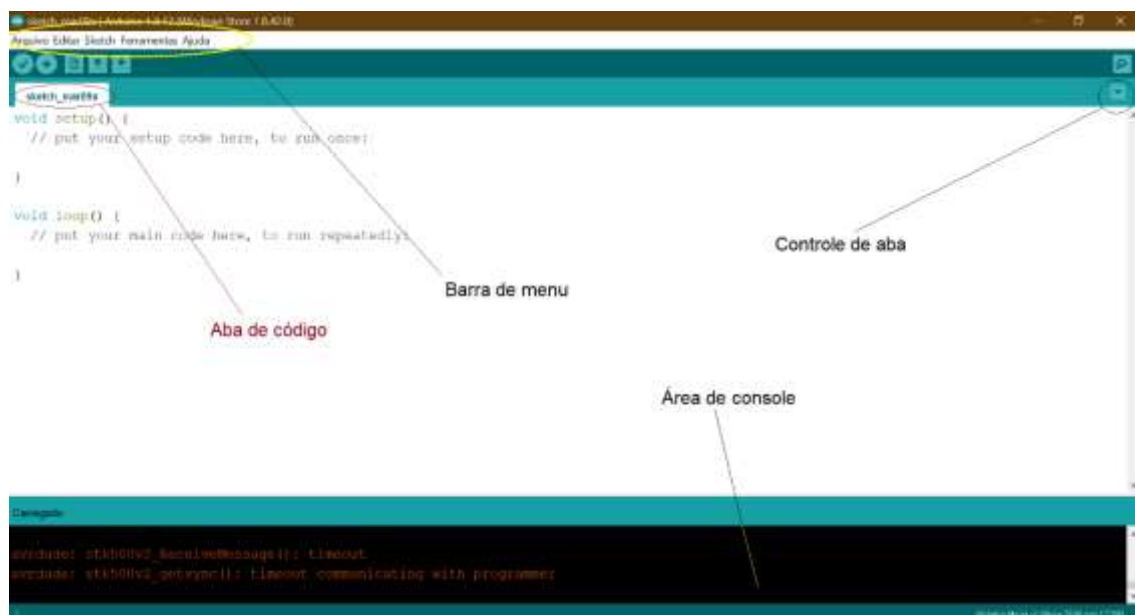


Figura 1 – Tela inicial da IDE do Arduino

A barra de menu é constituída por: *Arquivo*, *Editar*, *Sketch*, *Ferramentas* e *Ajuda*. Vamos nos concentrar somente em alguns elementos da barra de menu.

Os elementos mais simples não estão discutidos. Não discutimos também os elementos que se referem ao programador. O programador é o código de inicialização da placa. Por enquanto, se desejar aprofundar os detalhes, pode começar consultando [Arduino - Bootloader](#).

No menu Sketch, conforme apresenta a Figura 2, focaremos nos elementos detalhados a seguir.

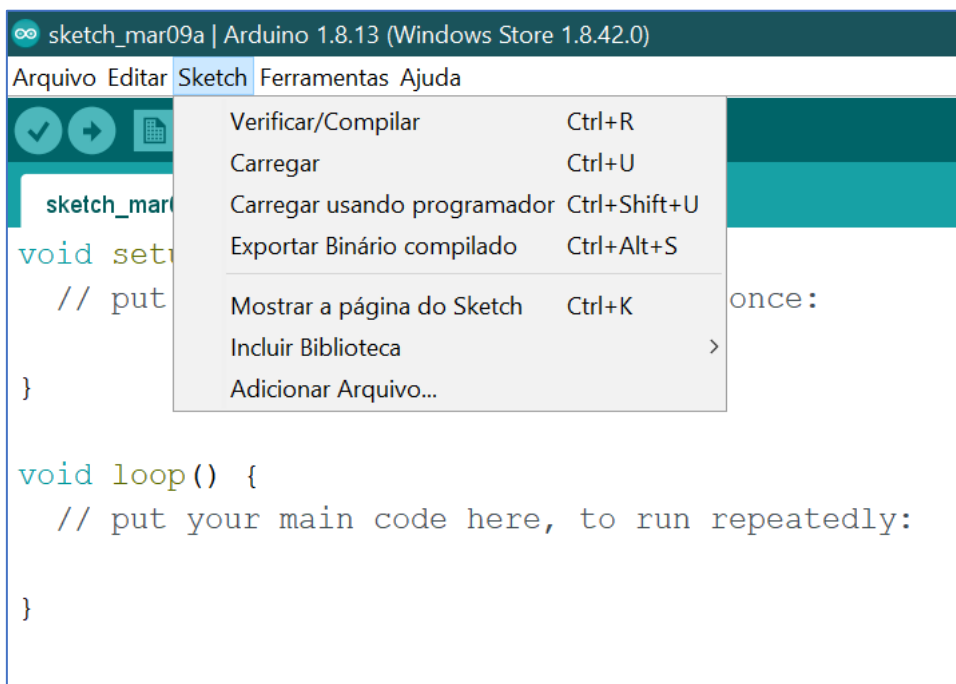


Figura 2 – Menu > Sketch

- *Sketch>Verificar/Compilar* – verifica erros no código, compilando-o; reporta na área do console o uso de memória e as variáveis;
- *Sketch>Carregar* – compila e carrega o arquivo binário na placa;
- *Sketch>Exportar binário compilado* – salva um arquivo .hex, que pode ser mantido ou enviado para a placa usando outras ferramentas;
- *Sketch>Incluir bibliotecas* – adiciona uma biblioteca ao seu sketch pela inserção do comando `#include` do pré-processador da linguagem; adicionalmente, a partir deste item de menu, pode-se acessar o *Gerenciador de Biblioteca* e importar novas bibliotecas de arquivos .zip, conforme ilustra a Figura 3;

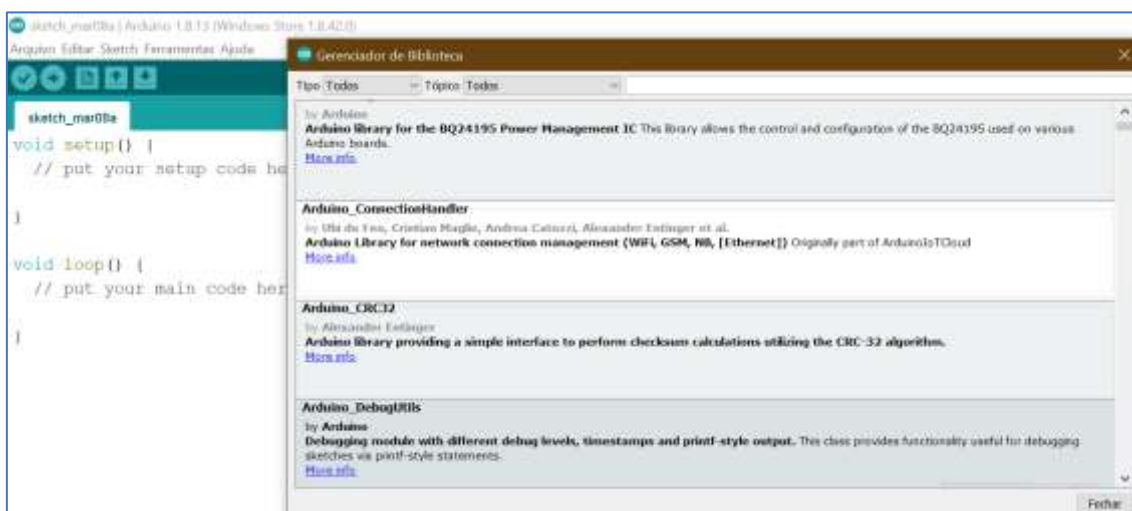


Figura 3 – Formulário de acesso ao Gerenciador de Biblioteca, pelo menu > Sketch > Incluir Biblioteca

- *Sketch>Adicionar Arquivo* – adiciona um arquivo fonte ao *sketch*, que é carregado como uma nova aba e pode ser apagado usando o triângulo invertido na parte superior direita da janela.

No menu Ferramentas, conforme apresenta a Figura 4, focaremos nos elementos detalhados a seguir.

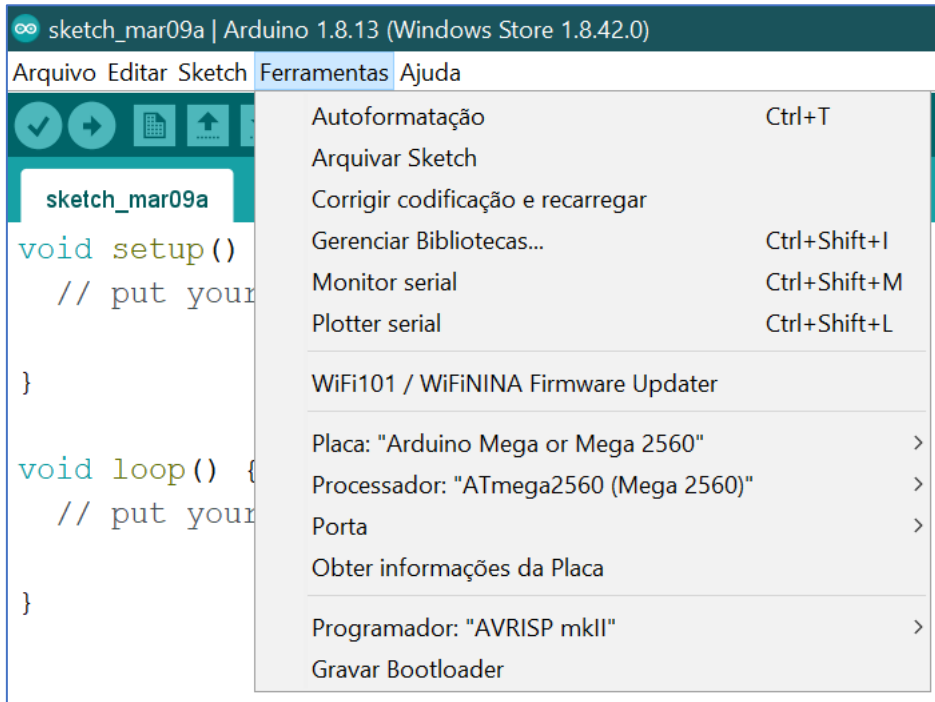


Figura 4 – Menu Ferramentas

- *Ferramentas>Corrigir codificação e recarregar* – corrige possíveis discrepâncias entre o mapa de codificação de caracteres do editor e mapas de caracteres de outros sistemas;
- *Ferramentas>Monitor serial* – abre uma janela de monitor serial e inicia a troca de dados com qualquer placa conectada na porta selecionada; normalmente reinicializa a placa se ela suportar a reinicialização pela porta serial – é necessário que a placa esteja alimentada e conectada;
- *Ferramentas>Plotter serial* – abre uma janela que mostra graficamente informações de comunicação serial das portas analógicas e digitais – é necessário que a placa esteja alimentada e conectada.

Os *sketchs* são disponibilizados em Sketchbook, uma pasta padrão de uso da IDE, que podem ser acessados em *Arquivo>Sketchbook* e pode ser configurada em *Arquivo>Preferências*.

Antes de carregar o *sketch* é necessário habilitar a porta serial correta. No Windows pode ser COM1, COM2 ou COM4, ... Para encontrar a porta adequada consultar a aplicação Gerenciador de Dispositivos do Windows.

Uma vez selecionada a porta serial e a placa carregar o sketch desejado. A placa irá reiniciar automaticamente e fazer o 'upload'. O botão LED de TX irá piscar.

Após realizada a conexão da placa na porta adequada automaticamente o programador é executado (código de inicialização da placa). É possível ver o LED de TX piscando. A IDE apresentará uma mensagem que o sketch foi carregado ou que houve um erro.

Para ilustrar os pontos específicos da linguagem será utilizado um circuito de conexão com o LCD 1602, uma família de LCD de 16 caracteres com 2 linhas, monocromático, cuja pinagem está apresentada na Figura 5.

A Tabela 1 sintetiza as funções da pinagem do LCD 1602, apresentando sua função e descrição. É essencial para conexão com a pinagem da placa Arduino MEGA 2560, utilizada no curso.

Tabela 1 – Pinagem do LCD 1602

Id	Função	Descrição
1	Gnd	Terra
2	Vcc	Alimentação DC – 3,3 V a 5V (típico)
3	VO	Tensão para ajuste de contraste
4	RS	'Register Select', Low=comando, High=dados
5	RW	'Read/Write', Low=gravação, High=leitura
6	E	'Enable', Low, gatilhada pela borda
7-14	DB0-7	'Data Bit', High/Low
15	A	'Anodo', +5V
16	K	'Catodo', Gnd

A Figura 5 apresenta o LCD com a pinagem e a disposição dos caracteres.

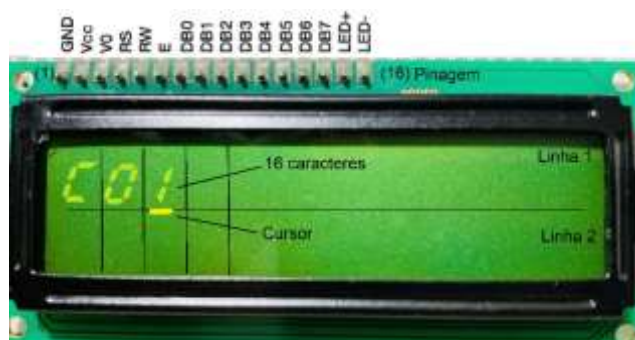


Figura 5 – LCD 1602 com apresentação da pinagem e da disposição dos caracteres

Os caracteres são apresentados em duas linhas de 16 caracteres cada uma. O LCD 1602 possui também um cursor, que indica a posição do caractere lido/escrito em um instante. A pinagem inicia-se a partir do pino 1 – na esquerda superior da figura -, até o pino 16, cujas funções estão especificadas na Tabela 1.

Algumas observações específicas à simulação realizada:

- No pino V0 do LCD 1602 a tensão de entrada será regulada por um potenciômetro de 10 kohms;
- O pino RW será conectado ao Gnd porque a operação a ser realizada será somente de escrita no LCD 1602;
- Será utilizado o modo de 4-bits do LCD 1602, que, por default, opera em DB4-7;
- No *backlight* do LCD 1602 (pinos de anodo e catodo), o pino de anodo será alimentado por uma tensão de + 5VDC com uma proteção dada por um resistor.

A Figura 6 apresenta o esquema das ligações utilizadas.

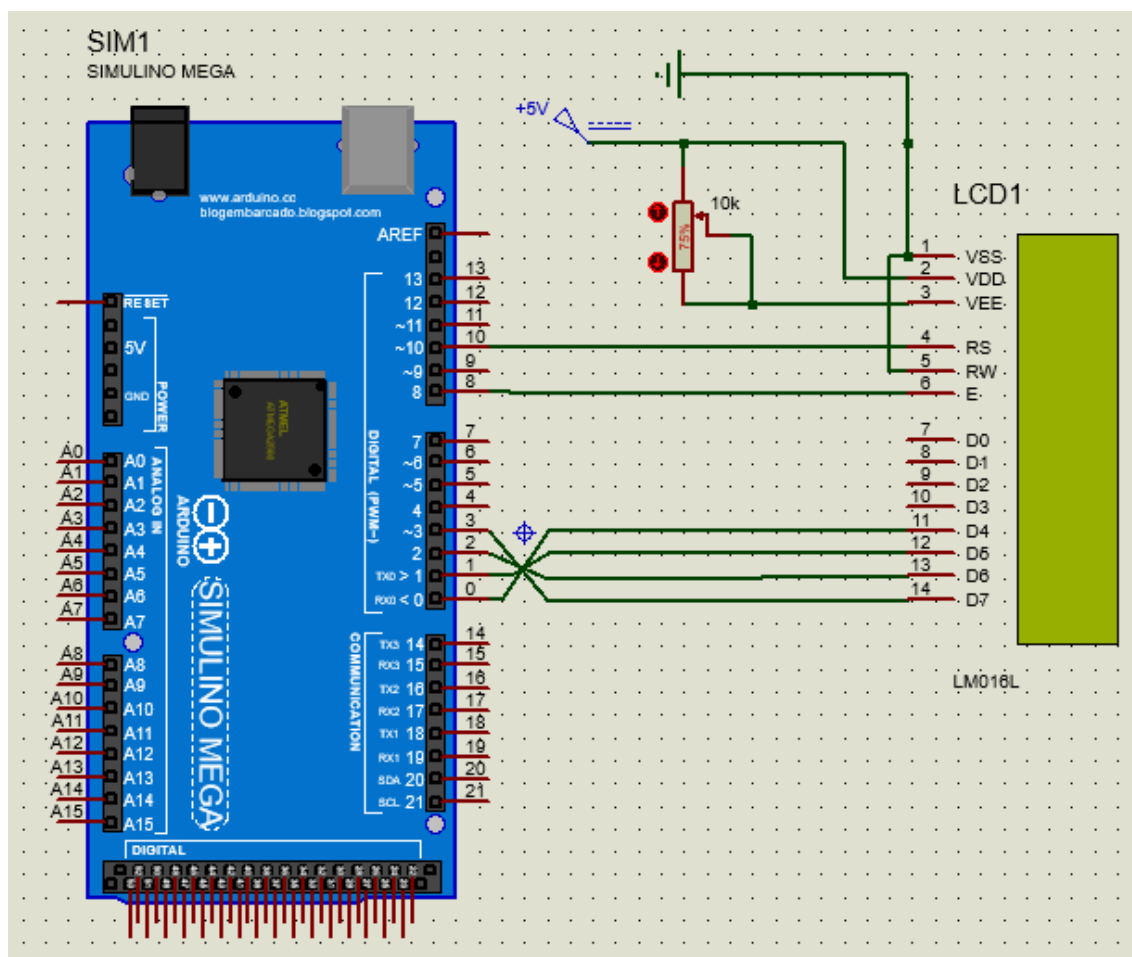


Figura 6 – Esquema elétrico das ligações da placa do Arduino MEGA2560 com o LCD 1502

O esquema foi elaborado com o software Proteus, versão 8.5. Não estão apresentadas as ligações ao anodo e ao catodo do LCD 1602. O detalhe das ligações está apresentado na Figura 7. O modelo do Proteus usa a notação VSS para Gnd e VEE para V0.

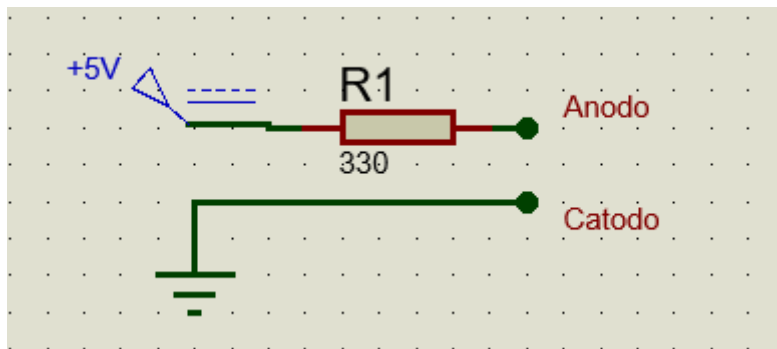


Figura 7 – Esquema das ligações com o backlight (anodo e catodo) do LCD 1602

Observe que estão sendo usados os pinos 0 a 3 do Arduino para alimentar a entrada de 4-bits de dados do LCD 1602. O pino 8 alimenta a entrada de *Enable* e o pino 10 alimenta a entrada de RS do LCD 1602.

O LCD 1602 trabalha com um microcontrolador cujo set de instruções não será objeto desta nota, pois foi utilizada a biblioteca [LiquidCrystal](#) do Arduino para gerar o programa a ser carregado no Arduino.

A Figura 8 apresenta o código utilizado para a montagem do modelo no *protoboard*.

```
sketch_pratica4 | Arduino 1.8.13
Arquivo Editar Sketch Ferramentas Ajuda

sketch_pratica4 $
#include <LiquidCrystal.h>

// inicializa a biblioteca
// com os números dos pinos do arduino
const int rs = 10, en = 8, d4 = 0, d5 = 1, d6 = 2, d7 = 3;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // configura o número de colunas e linhas do lcd:
  lcd.begin(16, 2);
  lcd.print("Uso do lcd1602");
}

void loop() {
  // set o cursor para a coluna 0, linha 1
  lcd.setCursor(0, 1);

  // escreve o número de segundos após o reset
  lcd.print(millis() / 1000);
}
```

Figura 8 – Sketch para uso do Arduino MEGA 2560 com o LCD 1602

A utilização da biblioteca LiquidCrystal simplifica sobremaneira a utilização do LCD.

O protótipo criado está apresentado na Figura 9.

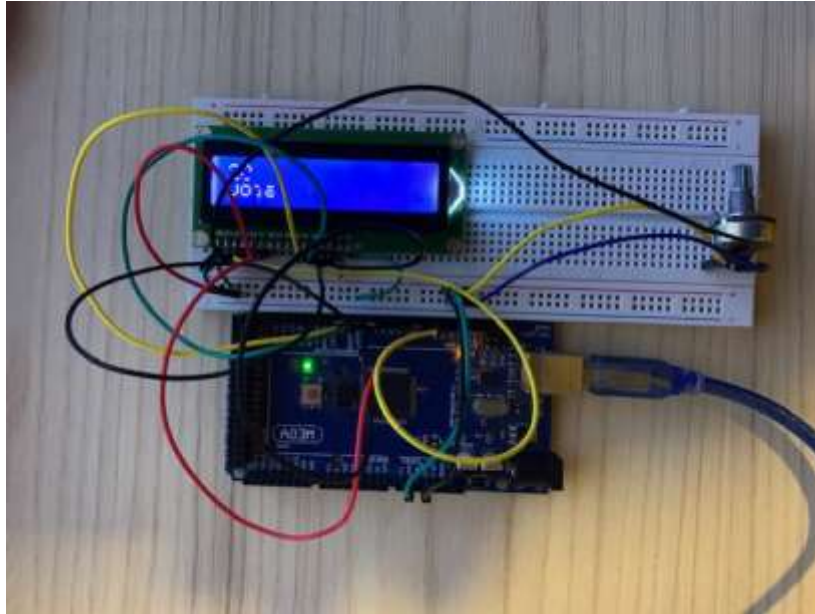


Figura 9 – Protótipo em protoboard da ligação do Arduíno com o LCD 1602