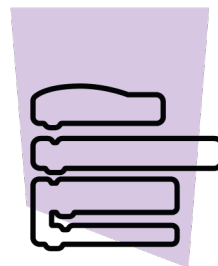
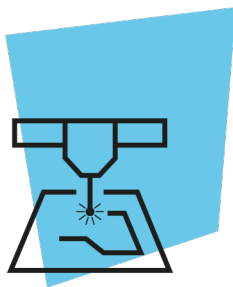
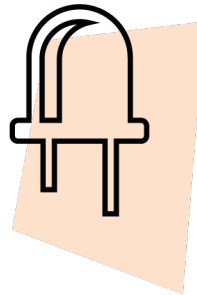
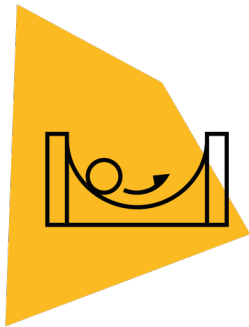


ROBÓTICA

Módulo 2



Escrevendo mensagens [Display OLED]

AULA 09

*Disponível no kit distribuído em 2023

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Adilson Carlos Batista

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Orlando de Macedo Junior

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Ilustração

Jocelin Vianna (ASCOM)

Modelagem 3D Blender

Cleiton Rosa

Roberto Carlos Rodrigues

2023



Sumário

Introdução	2
Objetivos desta aula	2
Competências gerais previstas na BNCC	3
Habilidades do século XXI a serem desenvolvidas	4
Lista de materiais	4
Roteiro da aula	5
1. Contextualização	5
2. Montagem e programação	6
3. Feedback e finalização	19
Referências	20



Introdução

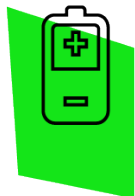
Em nosso kit de robótica você teve contato com o display OLED, um modelo de painel com várias aplicações e necessário para o nosso dia a dia, pois em muitos lugares é utilizado como recurso visual, seja para controlar a chegada emitindo número de senhas, conforme você viu na **Aula 8 - Painel de senhas**, seja para dar avisos de segurança, como ocorre no trânsito, ou ainda transmitir uma mensagem qualquer.

Nesta aula, seguiremos prototipando com o display OLED para apresentar mensagens enviadas por um computador.



Objetivos desta aula

- Programar o funcionamento do display OLED com uma mensagem;
- Introduzir as funções **Serial.available()** e **Serial.readString()** na programação.



Competências gerais previstas na BNCC

[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

[CG04] - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

[CG05] - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

[CG09] - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

[CG10] - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.





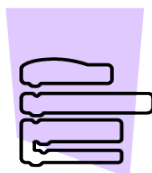
Habilidades do século XXI a serem desenvolvidas

- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



Lista de materiais

- 1 placa protoboard;
- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 4 jumpers macho-macho;
- 1 display OLED;
- Notebook;
- Software Arduino IDE.



Roteiro da aula

Contextualização

Os displays OLED são componentes bastante importantes e úteis, com aplicações variadas devido a sua versatilidade. Por meio delas podemos exibir mensagens usando letras, números e símbolos, bem como formar desenhos a partir de pixels, assim podemos aplicar o display em muitos projetos como interface visual entre o homem e a máquina.

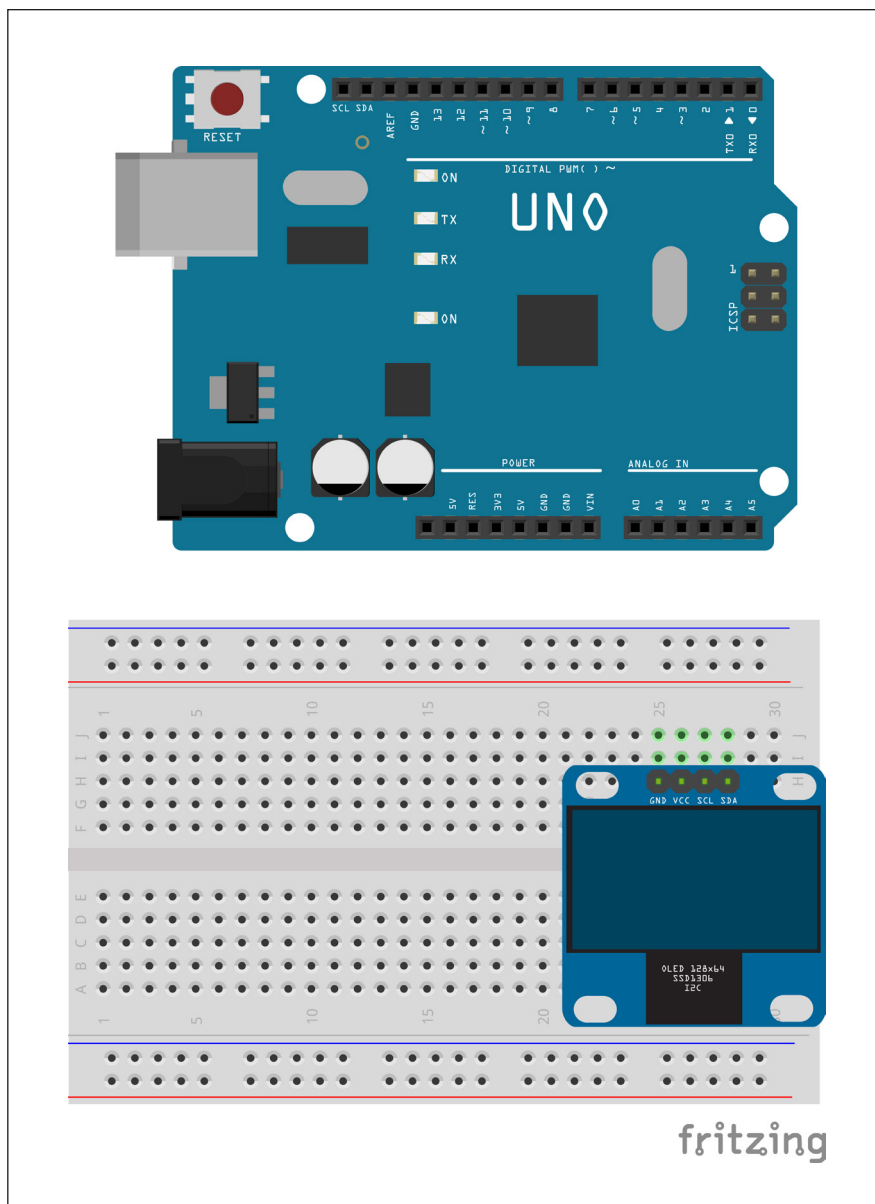
Toda vez que você for criar uma comunicação com o seu usuário, será esse o componente responsável para exibir a mensagem, ou, quando quiser visualizar os dados lidos por um sensor, por exemplo, seja ele de temperatura, de chuva, de fumaça, por exemplo, estes valores poderão ser exibidos no display OLED.

Para esta aula, iremos usar o display OLED, o qual, como vimos em aulas anteriores, possui grande nitidez e contraste devido ao controle individual de seus pixels, facilitando a exibição de mensagens programadas.

Montagem e programação

Iniciaremos nosso projeto conectando o display OLED à protoboard e deixando o Arduino posicionado próximo ao display (figura 1).

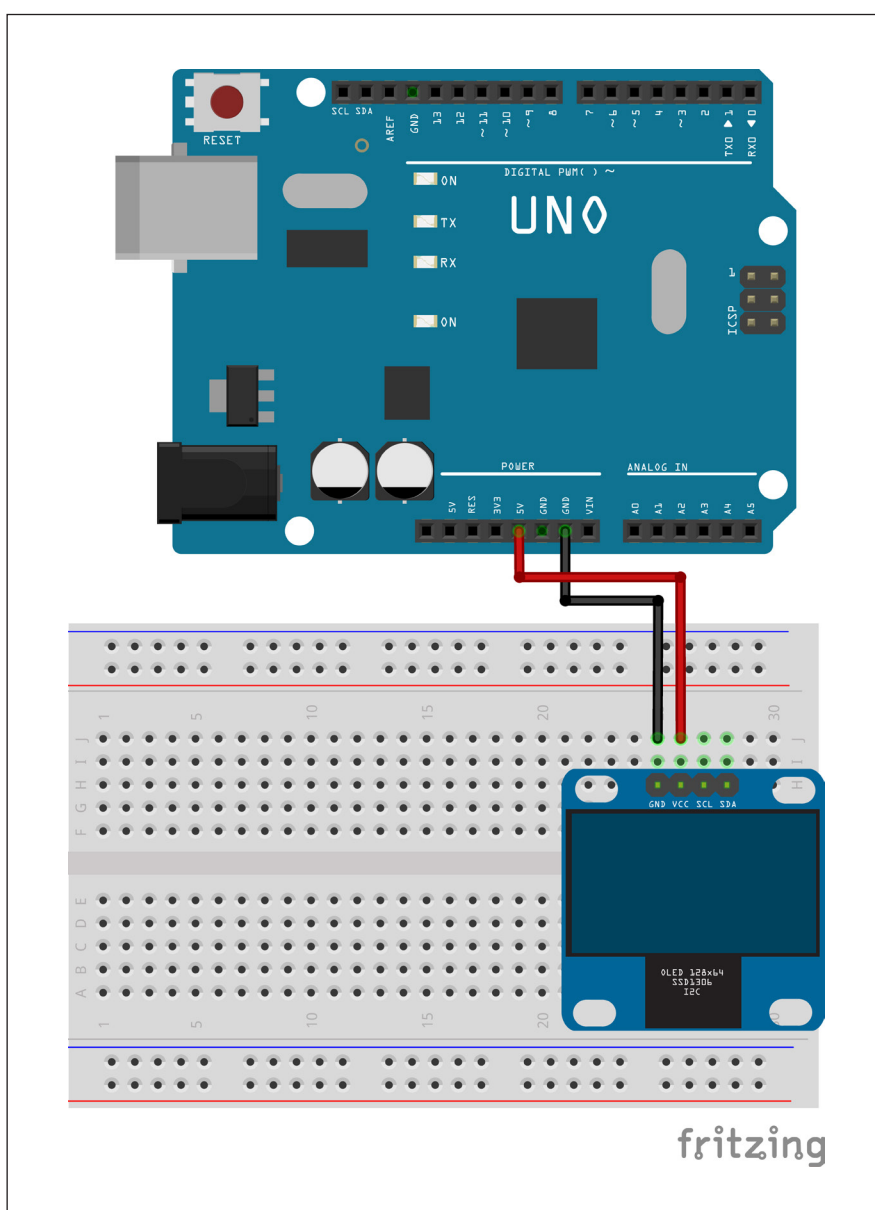
Figura 1 - Conexão dos pinos VCC e GND do display OLED ao Arduino



Fonte: SEED/DTI/CTE

Vamos iniciar as conexões de alimentação do display OLED ao Arduino: conecte o pino VCC do display à porta 5 V do Arduino e o pino GND à porta GND (figura 2), observando e seguindo a disposição dos pinos VCC e GND do display OLED disponível em seu kit de robótica.

Figura 2 - Conexão dos pinos VCC e GND do display OLED ao Arduino

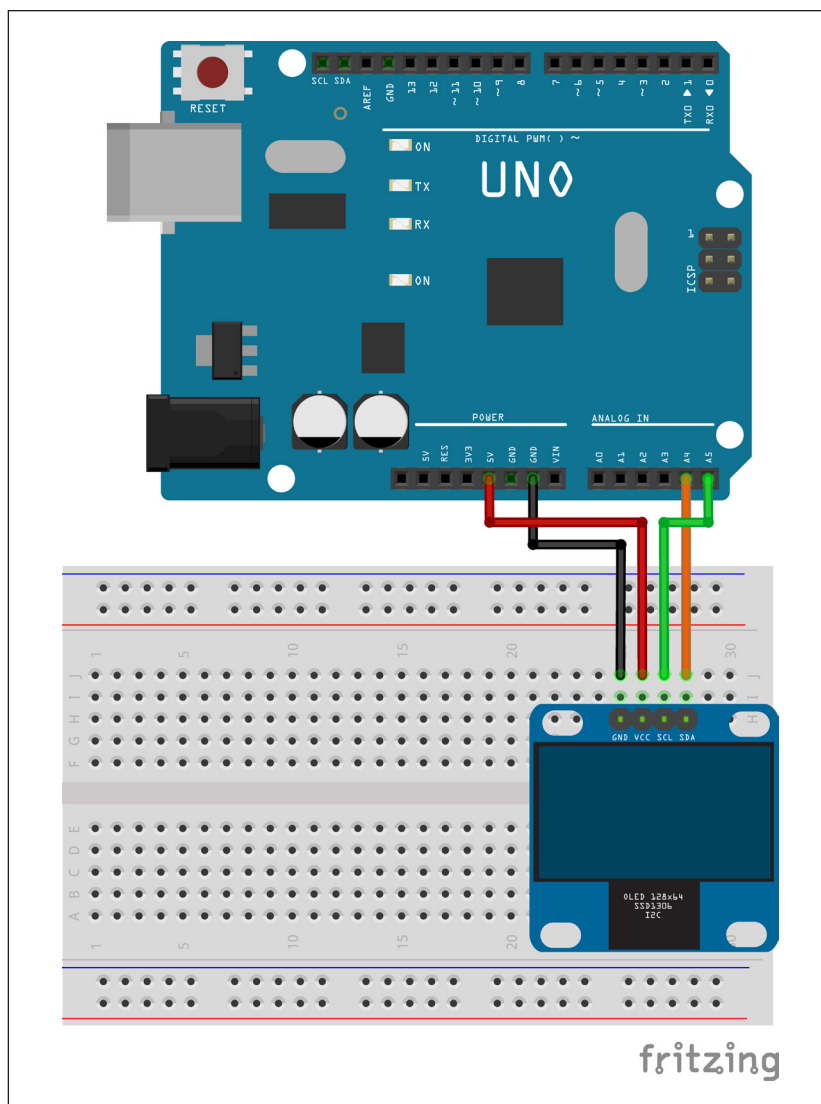


Fonte: SEED/DTI/CTE



Por fim, faremos as conexões de transmissão de dados: conecte o pino SCL (serial clock) do display OLED, referente ao sincronismo do display e do Arduino, à porta analógica A5 do Arduino e o pino SDA (serial data), referente à troca de dados, à porta analógica A4 do Arduino (figura 3). Como vimos na última aula, se a versão do seu Arduino Uno possuir as portas SCL e SDA, você poderá utilizá-las na montagem de seu projeto como opção à conexão direta às portas analógicas A5 e A4.

Figura 3 - Conexão dos pinos SCL e SDA do display OLED ao Arduino



Fonte: SEED/DTI/CTE

Com os componentes eletrônicos devidamente montados, vamos programar o nosso protótipo, conectando a placa Arduino ao computador, com o uso de um cabo USB, para que ocorra a comunicação do Arduino.

Para esta programação, utilizaremos as bibliotecas **Adafruit SSD1306** e **Adafruit GFX Library**, desenvolvidas pela Adafruit. Caso o Arduino IDE (versão Windows) indique que são necessárias outras bibliotecas para a utilização destas, clique na opção “instalar todas” (*Install all*).

No software Arduino IDE ou no Arduino Editor Online IDE, escreva ou copie e cole o código-fonte de programação (quadro 1).

Quadro 1 - Código-fonte da programação na linguagem do Arduino (Wiring)

```
/*
*****
/* Aula 9 - Enviando Mensagens [display OLED] */
/* Programação do projeto enviando mensagens. */
/* Ao transferir o código abaixo para seu Arduino, */
/* será possível, através do monitor serial do */
/* Arduino IDE, enviar mensagens de texto para o */
/* display OLED. */
*****
/* Se necessário, baixe as bibliotecas através dos */
/* links abaixo. */
/*
/* http://librarymanager/All#SSD1306#Adafruit */
/* Biblioteca: Adafruit SSD1306 by Adafruit */
/*
/* http://librarymanager/All#GFX#class#that */
/* Biblioteca: Adafruit GFX Library by Adafruit */
*****
/* Inclusão das bibliotecas necessárias. */
*/
```



```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

/* Variável para armazenar a mensagem. */
String msg;
/* Variável para controlar a linha do display. */
int linha;
/* Cria o objeto de controle para o display OLED. */
Adafruit_SSD1306 display(128, 64, &Wire, -1);

/* Início das configurações. */
void setup() {
  Serial.begin(9600);
  /* Tempo que o Arduino utilizará para "ler" os caracteres armazenados
na string msg. Se este tempo for curto para a quantidade de caracteres a
ser processado, a mensagem será enviada incompleta. */
  Serial.setTimeout(500);
  /* Inicializa o display OLED com o endereço 0x3C. */
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  /* Limpa o display. */
  display.clearDisplay();
  tela_inicial();
}

void loop() {
  /* Verifica dados disponíveis na porta serial. */
  if (Serial.available()) {
    /* Lê o próximo caractere da porta serial. */
    msg = Serial.readString();
    /* Se o caractere for %, limpa o display. */
    if (msg == "%") {
      display.clearDisplay();
      linha = 0;
    }
    /* Se o caractere for >, pula linha no display.*/
    else if (msg == ">") {
      linha += 8;
    }
  }
}
```

```
    else {
        display.setCursor(0, linha);
        display.print(msg);
    }
}
display.display();
}

void tela_inicial() {
    display.drawRect(0, 0, 128, 64, 1);
    display.drawFastHLine(0, 14, 128, WHITE);
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(4, 4);
    display.print("ESCREVA SUA MENSAGEM");
    display.setCursor((128 - 6 * 8) / 2, 20);
    display.setTextSize(1);
    display.print("Aten");
    display.write(135);
    display.write(97);
    display.setCursor(70, 20);
    display.write(126);
    display.write(111);
    display.write(19);
    display.setCursor(4, 30);
    display.setTextSize(1);
    display.print(" Utilize o comando % para nova mensagem e > para nova li-
nha.");
    display.display();
    linha = 0;
}
```

Fonte: SEED/DTI/CTE

Com o código-fonte inserido no Arduino IDE, compile o programa e carregue-o ao Arduino.

Após a transferência do programa para o Arduino, o display OLED exibirá uma mensagem inicial (figura 4), programada pela função `void tela_inicial()`, solicitando que você escreva a mensagem desejada.

Figura 4 - Exibição da mensagem inicial do display OLED

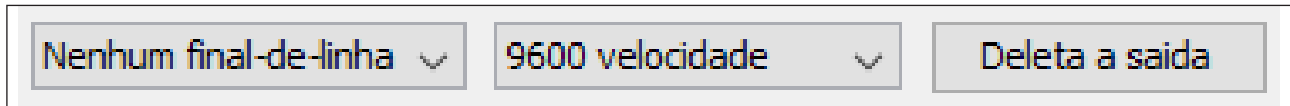


Fonte: SEED/DTI/CTE

Para enviar as mensagens que serão exibidas no display OLED, abra o **monitor serial** do Arduino e certifique-se que a velocidade de comunicação está de acordo com a programação (9600 bauds).

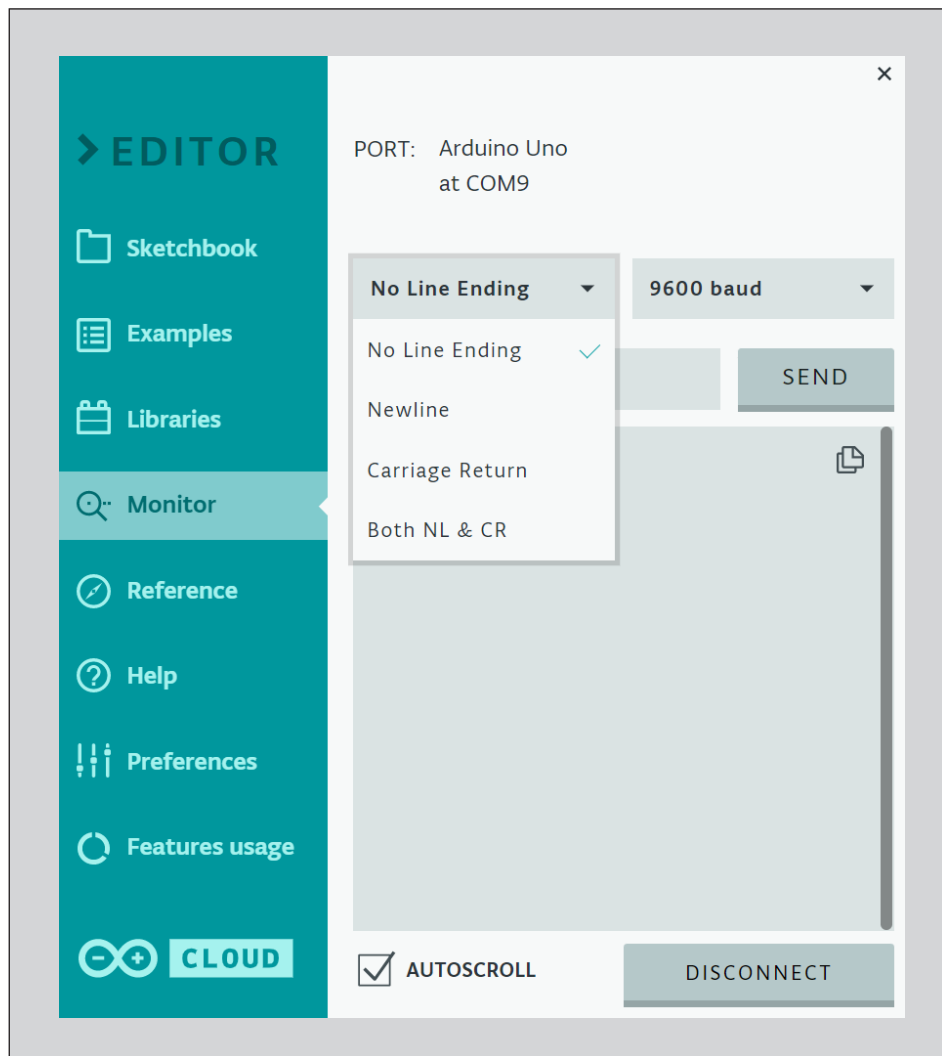
Não se esqueça ainda de selecionar a opção “Nenhum final-de-linha” (figura 5) ou “No Line Ending” (figura 6), conforme versão do Arduino que você está utilizando. Do contrário, as mensagens não serão exibidas no display.

Figura 5 - Seleção da opção “Nenhum final-de-linha” no monitor serial do software Arduino IDE



Fonte: SEED/DTI/CTE

Figura 6 - Seleção da opção “No Line Ending” no monitor serial do Arduino Editor Online



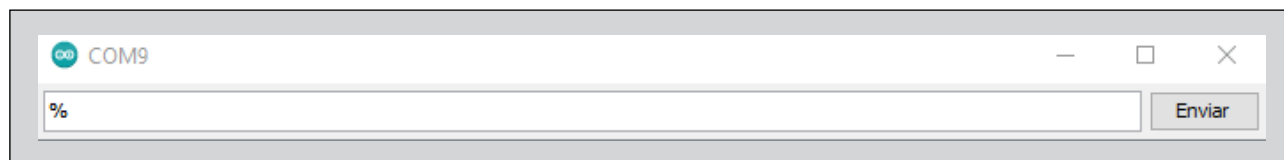
Fonte: SEED/DTI/CTE

Na caixa de texto do monitor serial das versões software Arduino IDE e Arduino Editor Online, você poderá enviar os comandos via porta serial. O Arduino estará, pela função `Serial.available()`, verificando os dados disponíveis na porta serial.

No nosso projeto, utilizaremos três recursos relacionados a caracteres armazenados na variável `msg`, os quais possibilitarão a escrita e visualização de mensagens quando o Arduino fizer a leitura destes caracteres pela função `Serial.readString()`: **limpar a tela**, para apagar as mensagens já impressas no display; **escrever mensagem**, utilizando o espaço para digitação de texto; **quebra de linha**, para possibilitar a escrita de mensagens linha a linha. Caso você não apague o display ou limpe a linha, poderá haver sobreposição de caracteres.

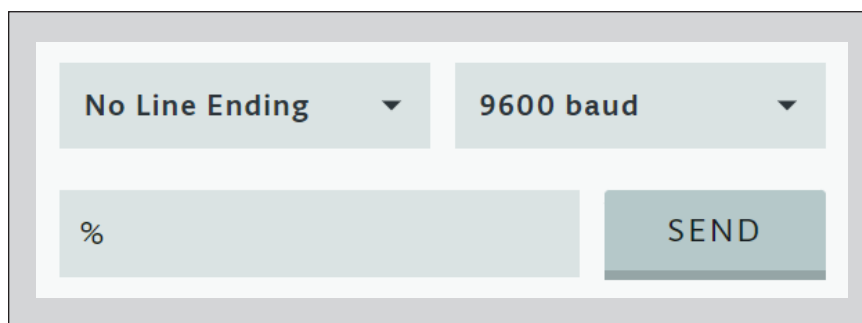
Para limpar a tela do display, insira o caracter `%` na área de digitação e envie o comando ao Arduino (figuras 7 e 8).

Figura 7 - Limpeza do display pelo monitor serial do software Arduino IDE



Fonte: SEED/DTI/CTE

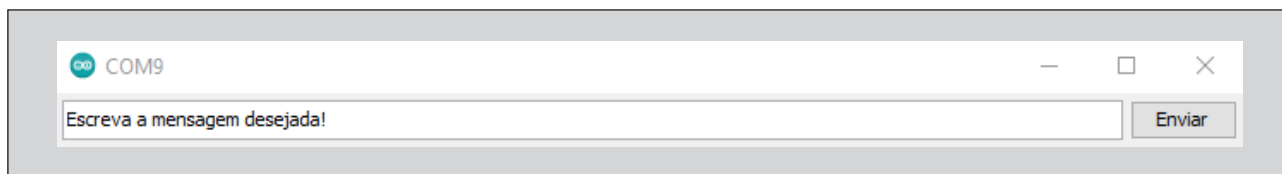
Figura 8 - Limpeza do display pelo monitor serial do Arduino Editor Online



Fonte: SEED/DTI/CTE

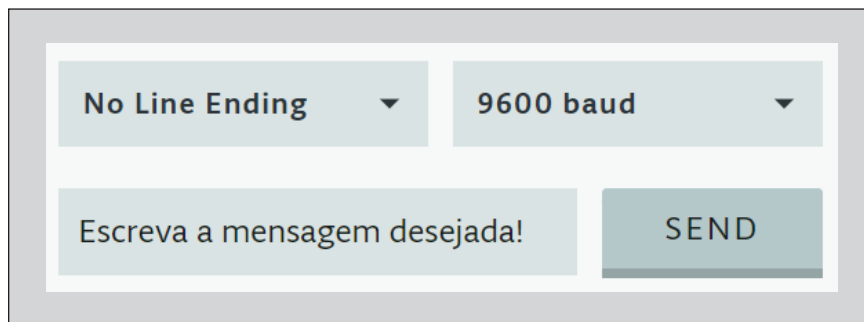
Na sequência, você já poderá escrever sua primeira mensagem (figuras 9 e 10). Na programação desta aula, o Arduino “lerá” em 500 milissegundos a mensagem que você pretende enviar. Caso você demore para digitá-la, ultrapassando o tempo que o Arduino aguarda para “leitura” dos caracteres digitados, ele não a exibirá completamente por precisar de um tempo maior para esta leitura, considerando apenas as informações recebidas no tempo de espera de 500 milissegundos, o qual você poderá adequar, no setup da programação, pela função `Serial.setTimeout()`.

Figura 9 - Escrita de mensagem pelo monitor serial do software Arduino IDE



Fonte: SEED/DTI/CTE

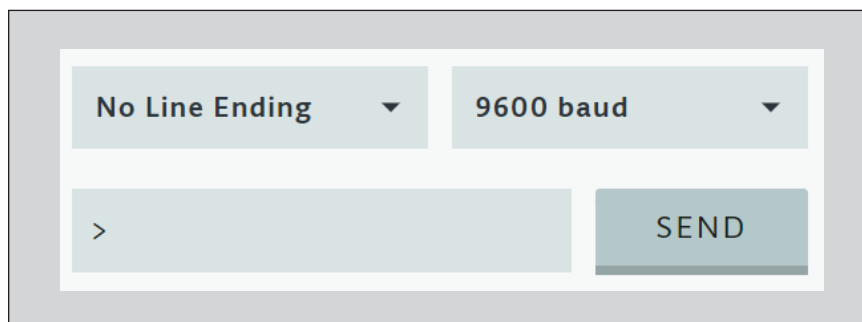
Figura 10 - Escrita de mensagem pelo monitor serial do Arduino Editor Online



Fonte: SEED/DTI/CTE

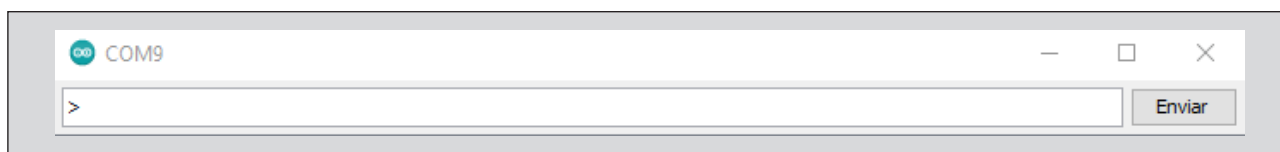
Para escrever uma nova mensagem com quebra de linha, ou seja, abaixo da mensagem que já estiver escrita no display, insira o caractere `>` no monitor serial e clique em Enviar (figuras 11 e 12). Você também pode utilizar este recurso para posicionar o cursor mais abaixo no display, mesmo sem escritas anteriores.

Figura 11 – Quebra de linha pelo monitor serial do software Arduino IDE



Fonte: SEED/DTI/CTE

Figura 12 – Quebra de linha pelo monitor serial do Arduino Editor Online



Fonte: SEED/DTI/CTE

Enquanto estiver escrevendo suas mensagens, lembre-se que a biblioteca utilizada na programação do projeto desta aula não aceita palavras com acentuação e “ç” escritas pelo monitor serial ou como texto pela função `display.print(“ ”)`.



Curiosidade!

Alguns caracteres presentes na mensagem inicial do display OLED, quando o código é carregado ao Arduino, foram impressos mediante **tabela de códigos ASCII**, na qual tem-se códigos numéricos decimais para a representação de caracteres. Observe, nas funções `display.write()` presentes no código da nossa programação estes códigos numéricos e identifique, pela mensagem da tela inicial do display OLED (figura 4), os caracteres correspondentes.

Explore o display OLED e capriche nas mensagens!



**Desafio:**

Que tal exibir no display OLED uma mensagem com efeito animado? Para isso, você pode utilizar a função `startscrollright()`, para efeito de rolagem à direita, e a função `startscrollleft()`, para efeito de rolagem à esquerda, como no exemplo a seguir (quadro 2 - Função `RP_animado`). Conforme o tamanho de sua mensagem, adeque as posições do cursor e tempo de rolagem:

Quadro 2 - Função `RP_animado`

```
/* Programação com função criada para efeito de mensagem rolante (scroll)
à direita e à esquerda. */

/* Inclui as bibliotecas para utilização do display OLED. */
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

/* Cria o objeto de controle para o display OLED. */
Adafruit_SSD1306 display(128, 64, &Wire, -1);

void setup() {
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C); /* Inicializa o display. */
    display.clearDisplay(); /* Limpa o display. */
}
```



E se...?

O projeto não funcionar?

Verifique se os jumpers estão na mesma coluna dos terminais dos componentes, fazendo assim a conexão;

Verifique se os jumpers estão ligados aos pinos corretos no Arduino;

Verifique se a programação está adequada a cada porta utilizada no projeto.

3. Feedback e finalização

Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.

Analise seu projeto desenvolvido, de modo a atender aos requisitos para o funcionamento do display OLED.

Refleta se as seguintes situações ocorreram:

i. Colaboração e cooperação: você e os membros de sua equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?

ii. Pensamento crítico e resolução de problemas: você conseguiu identificar os problemas, analisar informações e tomar decisões de modo a contribuir para o projeto desenvolvido?

Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de robótica.



Referências

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 25 de abril de 2023.

AUTOCORE ROBÓTICA. **Conhecendo o protocolo I2C com Arduino**. Disponível em: <https://autocorerobotica.blog.br/conhecendo-o-protocolo-i2c-com-arduino/>. Acesso em: 25 de abril de 2023.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 12 jul. 2022.

MECCOM. **I2C - Protocolo de Comunicação**. Disponível em: <https://www.mecco-meletronica.com.br/data/uploads/i2c-protocolo-de-comunicacao.pdf>. Acesso: 28 abr. 2023.

VIDA DE SILÍCIO. **Protocolo I2C - Comunicação entre Arduinos**. Disponível em: <https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos/>. Acesso em: 25 de abril de 2023.

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

Andrea da Silva Castagini Padilha
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edgar Cavalli Júnior
Edna do Rocio Becker
José Feuser Meurer
Marcelo Gasparin
Michele Serpe Fernandes
Michelle dos Santos
Orlando de Macedo Júnior
Roberto Carlos Rodrigues

Os materiais, aulas e projetos da “Robótica Paraná” foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.
Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilhável 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

