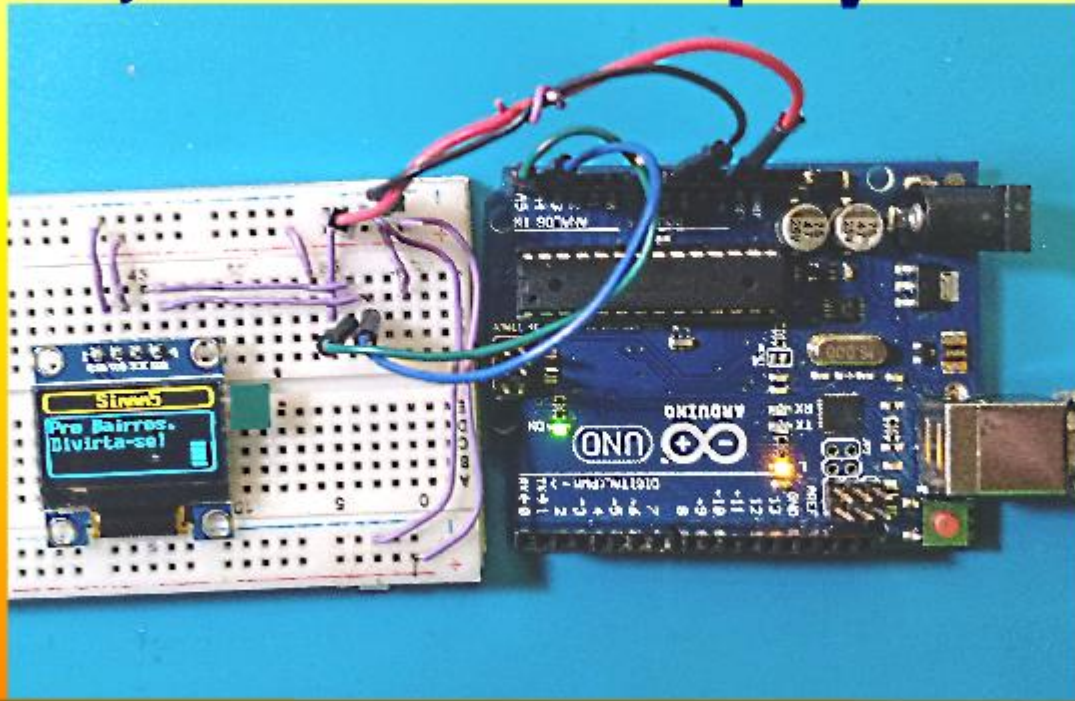


DESBRAVE ESSE FANTÁSTICO DISPLAY OLED AZUL E AMARELO COM COMUNICAÇÃO I2C!

## Veja como usar o display OLED



Professor Bairros (01/05/2023)

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!



The screenshot shows the homepage of 'bairrospd.com'. At the top left is the logo 'bairrospd' with the text 'BAIRROS PROJETOS DIDÁTICOS E ELETRÔNICOS'. Below the logo is a green banner with the text 'ESTUDE ELETRÔNICA NO SITE WWW.BAIRROSPD.COM!'. The main content area features a navigation menu with 'HOME', 'CURSOS', 'BIBLIOTECA', 'TUTORIAIS', 'VOCÊ SABIA?', and 'CONTATO'. A prominent yellow banner reads 'APRENDA A LER RESISTORES' with an illustration of a person working with a resistor. To the right of this banner is a section titled 'Procure aqui:' with the text 'O QUE SIGNIFICA GASTAR ENERGIA ELÉTRICA: Uma questão de Potência.' Below this is a blue banner that says 'AULAS OU ASSESSORIA COM O ENGENHEIRO E PROFESSOR ROBERTO BAIROS?' and a 'CLIQUE AQUI' button.

**VISITE  
O NOSSO  
SITE e  
CANAL  
YOUTUBE**

[www.bairrospd.com](http://www.bairrospd.com)  
Professor Bairos

[www.bairrospd.com](http://www.bairrospd.com)

VISITE O SITE DO PROFESSOR BAIROS LÁ EM O PDF E MUITO MAIS.  
PARA AULAS ONLINE CONTATE VIA SITE.

[www.bairrospd.com](http://www.bairrospd.com)

<https://www.youtube.com/@professorbairros>

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### Sumário

1. Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C! .....	4
2. O circuito.....	5
3. O software. ....	6
4. A biblioteca. ....	7
5. Incluindo os drivers no programa. ....	9
6. A comunicação I2C.....	10
7. O setup .....	11
8. As funções.....	12
9. Programa 1 Desenhando Linhas .....	13
10. O Loop.....	14
11. A rotina desenha linhas e escreve. ....	15
12. Fim da área amarela. ....	17
13. Escrevendo um texto.....	18
14. A linha vertical.....	20
15. Resumo das áreas do display. ....	21
16. Função drawLine .....	22
17. A função drawFrame .....	23
18. O programa desenhado frames. ....	24
19. Função frame com frames redondos.....	25

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

20.	A função print que mostra uma variável. ....	26
21.	A função drawBox. ....	27
22.	Incrementando a variável no loop. ....	29
23.	Conclusão. ....	30
24.	Apêndice ....	31
	1 desenhando linhas. ....	32
	2 Desenhando Frames.....	34
	3 Desenhando frame com raio.....	36
	4 Print variável ..... 38	
	Desenhando BOX variável ..... 40	
25.	Créditos.....	43

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 1. DESBRAVE ESSE FANTÁSTICO DISPLAY OLED AZUL E AMARELO COM COMUNICAÇÃO I2C!

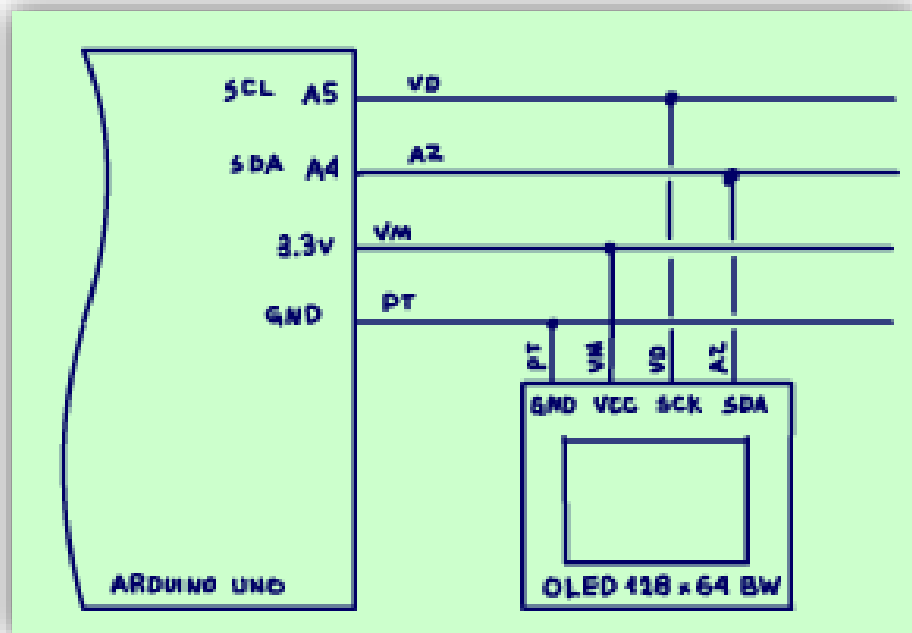


Nesse tutorial eu vou desvendar os segredos do Display OLED azul e amarelo com comunicação I2C, um display muito simples e colorido que pode dar um charme extra aos seus projetos e claro que eu vou explicar, bem explicadinho.

Vamos lá.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 2. O CIRCUITO.



Como amantes que somos da eletrônica tudo deve começar com o diagrama de verdade, nada daqueles desenhos do Tinkercard, veja na figura como ligar o Display OLED no Arduino.

O modelo que eu vou mostrar nesse tutorial é o modelo conhecido como display OLED 128x64 0,96 I2C azul e amarelo.

Esse 0,96 é o tamanho da tela 0,96 polegadas.

A cor das mensagens podem ser duas, amarelo e azul.

Esse I2C é o modelo DA comunicação.

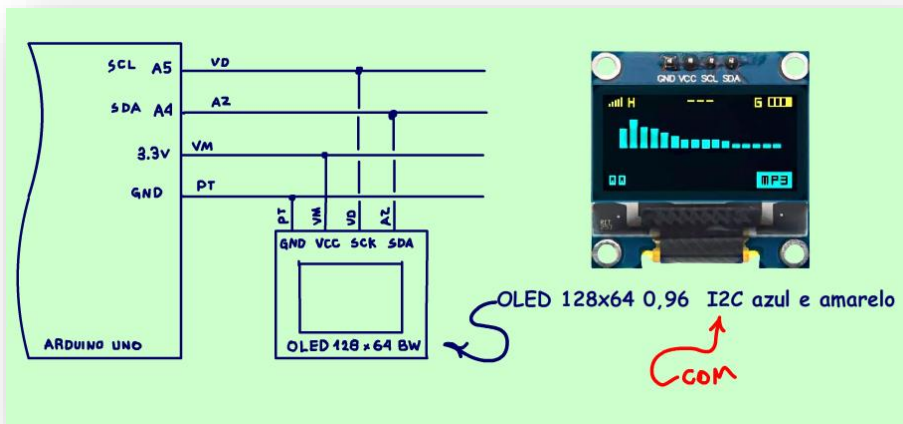
E pronto eletricamente é isso!

Arthurzinho:

Não podia ser mais simples, e o software?

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 3. O SOFTWARE.



Quando a gente estuda um novo dispositivo para usar com o Arduino, depois de aprender a conexão precisamos aprender a programar esse dispositivo.

A conexão na maioria dos casos é muito simples, com neste caso, com diz o Arthurziho, não podia ser mais simples.



Quanto a programação, o software, a maioria simplesmente copiam e colam, isso não está errado, desde que você entenda os detalhes dessa programação e possa adaptar a sua necessidade, a sua aplicação, então é isso que eu acho de mais importante num tutorial como esse, o desvendar da

programação, mostrar os detalhes, é isso que eu chamo explicar, bem explicadinho, então vamos lá.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 4. A BIBLIOTECA.

```
Blink$
1 // Blink
2
3
4 // the setup function runs once when you press reset or power
5 void setup() {
6   // initialize digital pin LED_BUILTIN as an output.
7   pinMode(LED_BUILTIN, OUTPUT);
8 }
9
10 // the loop function runs over and over again forever
11 void loop() {
12   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the active level)
13   delay(1000); // wait for a second
14   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the pin output voltage LOW
15   delay(1000); // wait for a second
16 }
```



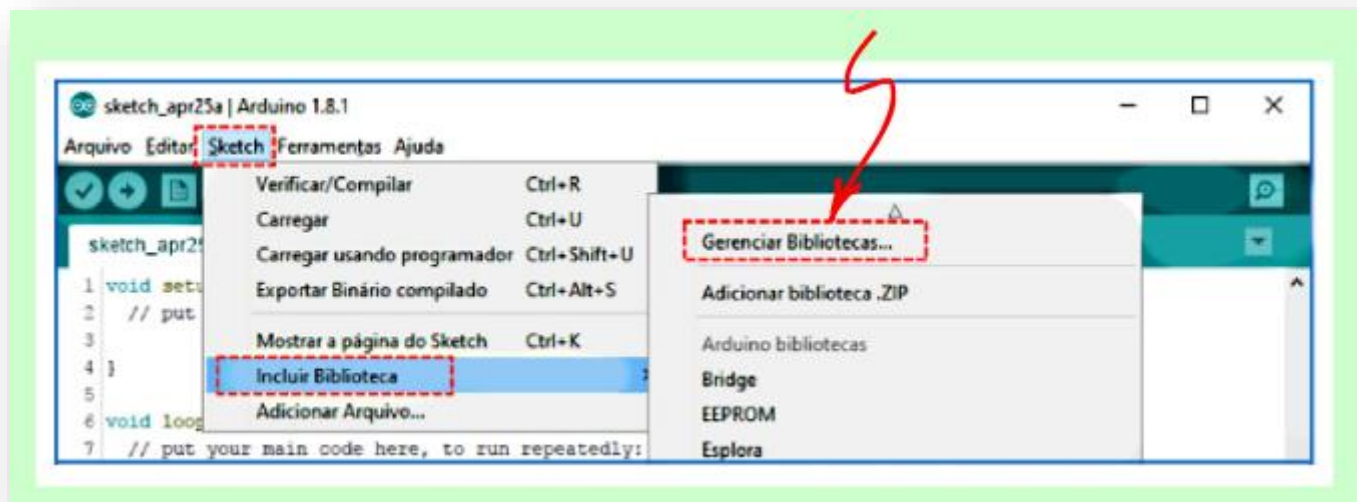
Programar um dispositivo é usar as funções que estão prontinhas para ativar esse dispositivo.

Essas funções estão armazenadas num programa colocado dentro da biblioteca do Arduino, esse programa pode ser chamado de driver. Então, cada vez que uma empresa produz um dispositivo, junto manda fazer a biblioteca com as funções para usar esse dispositivo, simples assim.

Então quando você compra um dispositivo, tem que baixar da internet a biblioteca e depois instalar na Interface de programação do Arduino, isso vale para qualquer outro microcontrolador.



## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!



Para usar o display você precisa instalar a biblioteca U8GLIB, essa é a mais simples, existem outras mais modernas, mas isso ficará para outro tutorial, vou começar com a mais simples.

Para isso na Interface de programação do Arduino, vou chamar de IDE para simplificar, clique em Sketch, incluir biblioteca e gerenciar biblioteca, isso vale para qualquer Driver.



Na janela de texto digite o nome da biblioteca, u8glib, pode ser em letras minúsculas e espere um pouco para a IDE procurar, então a IDE mostra a janela com as bibliotecas que podem ser usadas, escolha a u8glib que já vem com o Arduino e depois é só clicar em instalar e pronto, você tem tudo para usar o seu Display.

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

## 5. INCLUINDO OS DRIVERS NO PROGRAMA.

```
1_OLE_teste_Linhas
1 //Programa: TESTE DAS FUNÇÕES u8g DO DISPLAY OLE
2 //Programa teste Desenhando Linhas
3 //Autor: Professor Bairros
4 #include <U8glib.h>
5 U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); //Definicoes do display Oled
```

Você deverá incluir esse driver no seu programa para usar as suas funções, esse é o primeiro passo a fazer no programa escrever a linha include como na figura.

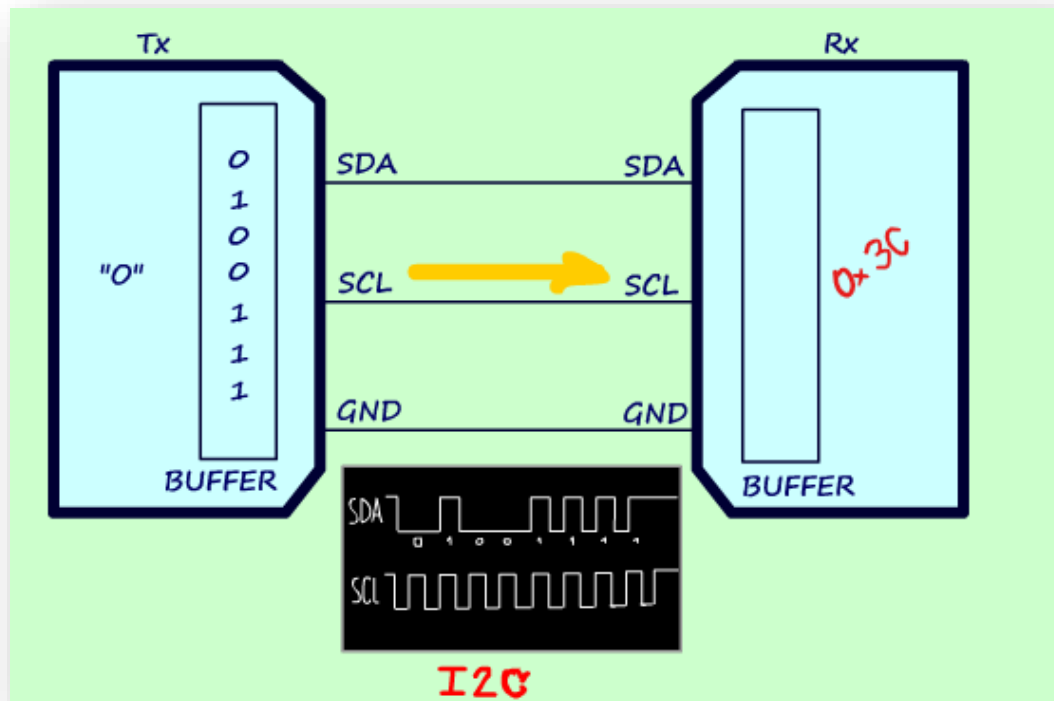
As funções estão todas agrupadas com um nome muito comprido, então você dá uma espécie de apelido a esse nome, um apelido mais fácil de identificar e que

you will use in your program, in this example it will be u8g, it is better not to be the same.

Now whenever you want to use a function from the display control driver, first call it by u8g dot, then after the dot call the desired function, this programming is also not so complicated as you might think, is it not?

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 6. A COMUNICAÇÃO I2C.



Esse display usa a comunicação I2C.

Como você já viu no meu tutorial sobre os sistemas de comunicação, claro que está na descrição desse vídeo, a comunicação I2C é feita a dois fios, o Arduino é o mestre e todos os outros dispositivos conectados na rede tem um endereço, são os escravos, para a maioria dos módulos desse tipo, display OLED o endereço é 0x3C.

A linha SDA é a linha de dados e a linha SCL são os pulsos de clock, essa é uma comunicação síncrona.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 7. O SETUP

```
22 void setup()
23 {
24   Serial.begin(9600);
25   //inicializa Display Oled
26   if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
27     u8g.setColorIndex(255); //volta 40 white
28   }
29   else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
30     u8g.setColorIndex(3); //volta 2max intensity
31   }
32   else if ( u8g.getMode() == U8G_MODE_BW ) {
33     u8g.setColorIndex(1); //volta 1 pixel on esse é o mais comum
34   }
35   else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {
36     u8g.setHiColorByRGB(255, 255, 255); //volta 48
37   }
38 }
```

Depois de incluir a biblioteca, dizer qual o tipo de display você está usando, agora você deverá ativar o driver, isso é feito no setup.

Para o controle exato do display o programa deve saber qual o modelo está sendo usado, isso é chamado de inicializar o display.

A rotina descrita aqui é um padrão que identifica o tipo do display e inicializa esse tipo.

O que esse conjunto de instruções faz é perguntar ao display, que modelo você é, é a instrução getmode, esse get depois do apelido u8g indica que é uma pergunta, o mode indica que a pergunta é me diga o seu tipo.

A rotina faz várias perguntas só uma será verdadeira.

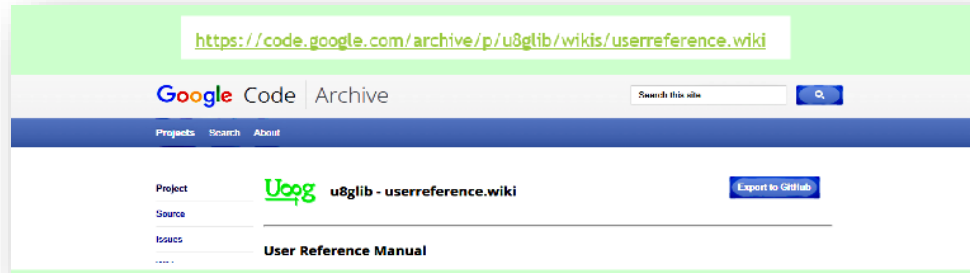
Uma vez identificado o tipo a rotina SET liga as funções apropriadas.

A linha serial.begin inicializa a serial padrão do Arduino, você deve fazer isso sempre para deixar a serial pronta para ser usada.

Note que essa operação de ativar os drivers eu vou chamar de inicializar, ela é feita dentro do setup.

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

## 8. AS FUNÇÕES.



As funções disponíveis são muitas e todas estão amplamente descritas na internet no endereço da figura:

<https://code.google.com/archive/p/u8glib/wikis/userreference.wiki>

esse endereço está descrição desse vídeo, abra esse endereço e todas as funções ficarão nas suas mãos.

São muitas, mas todas seguem a mesma lógica, então vou mostrar as mais usadas e a filosofia será a mesma para todas as outras.

Primeiro vou mostrar como funciona a tela desse display.

O display é do tipo 128x64, esse 128 indica o número de pixels na largura e o 64 indica os pixels na altura, vou mostrar essa área usando a função desenhar linhas.

Conhecer a área disponível é importante, e causa muita confusão, então vou explicar, bem explicadinho, vou fazer isso desenhando linhas ao redor do display.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 9. PROGRAMA 1 DESENHANDO LINHAS

```

1_OLE_teste_Linhas
1 //Programa: TESTE DAS FUNÇÕES u8g DO DISPLAY OLE
2 //Programa teste Desenhando Linhas
3 //Autor: Professor Bairros
4 #include <U8glib.h>
5 U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); //Definições do display Oled
6 void draw() ← draw()
7 {
8   //Comandos graficos para o display devem ser colocados aqui
9   u8g.setFont(u8g_font_8x13B); //normal
10  u8g.drawHLine(0,0,127); // (x,y,L) linha H começando no pixel xy mais L pixel
11  u8g.drawHLine(0,15,127); // (x,y,L) linha H começando no pixel xy mais L pixel
12  u8g.drawHLine(0,16,127); // (x,y,L) linha H começando no pixel xy mais L pixel
13  u8g.drawHLine(0,63,127); // (x,y,L) linha H começando no pixel xy mais L pixel
14  u8g.drawVLine(0,0,15); // (x,y,L) linha V começando no pixel xy mais L pixel
15  u8g.drawVLine(0,16,47); // (x,y,L) linha V começando no pixel xy mais L pixel
16  u8g.drawVLine(127,0,16); // (x,y,L) linha V começando no pixel xy mais L pixel
17  u8g.drawVLine(127,16,48); // (x,y,L) linha V começando no pixel xy mais L pixel
18  u8g.drawString(40, 12, "Simmm1"); // escreve na posição (x,y, "strg") xy é a base
19  u8g.drawString(3, 29, "Pro Bairros."); // escreve na posição (x,y, "strg") xy é a base
20  u8g.drawString(3, 42, "Divirta-se!"); // escreve na posição (x,y, "strg") xy é a base
21 }
22 void setup()

```

rotina, executando as instruções uma após a outra.

A gente já viu o setup.

Agora vou mostrar o loop.

O programa do Arduino é dividido em partes, a primeira parte antes do setup, vou chamar de cabeçalho, você inclui os seus drivers, declara as variáveis e monta as suas sub-rotinas, que são rotinas auxiliares a serem usadas no seu programa, nesse programa tem a rotina draw que é muito importante, ela escreve ou desenha no display.

No setup você ativa todos os drivers, liga tudo.

Por último o Loop que é o lugar onde você monta o seu programa propriamente dito, aqui que tudo acontece, essa parte é chamada loop, porque o programa fica rodando em loop, em ciclo dentro dessa

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 10.0 LOOP

```
39 void loop()
40 {
41   u8g.firstPage(); //Chama a rotina de desenho na tela
42   do
43   {
44     draw(); //chama a rotina que desenha a tela
45   }
46   while ( u8g.nextPage() ); //fica desenhado até o fim da memória.
47   delay(10000); //delay em milisegundos
48 }
```

O que o loop faz é chamar a rotina que desenha no display, essa rotina foi criada nesse programa, é ela que muda conforme o aplicativo, você pode dar qualquer nome a essa rotina, pode ser desenhar, display ou coisa assim, é nessa rotina que tudo acontece.

Depois de escrever tem o DELAY de 10000ms, para fixar a tela na retina.

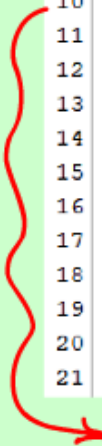
Esse é um padrão de rotina que poderá ser usado para outros aplicativos que use esse display.

Então, vamos ver a rotina que escreve no display.

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

## 11. A ROTINA DESENHA LINHAS E ESCREVE.

```
6 void draw()
7 {
8
9   u8g.setFont(u8g_font_8x13B);
10  u8g.drawLine(0,0,127);
11  u8g.drawLine(0,15,127);
12  u8g.drawLine(0,16,127);
13  u8g.drawLine(0,63,127);
14  u8g.drawLine(0,0,15);
15  u8g.drawLine(0,16,47);
16  u8g.drawLine(127,0,16);
17  u8g.drawLine(127,16,48);
18  u8g.drawString(40, 12, "Simmm1");
19  u8g.drawString(3, 29, "Pro Bairros.");
20  u8g.drawString(3, 42, "Divirta-se!");
21 }
```

 u8g.drawLine(0,0,127); //(x,y,w) linha H começando no pixel xy mais w pixel

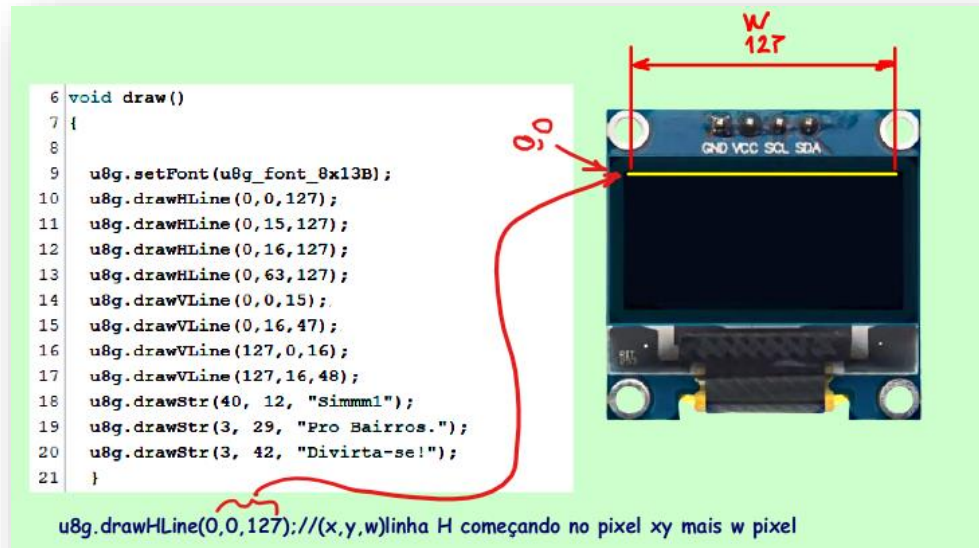
Vou tentar mostrar tudo passo a passo, montando vários programinhas, do mais simples ao mais enroladinho um pouco, tudo para que você não tropece.

Esse é o primeiro exemplo, vamos desvendar os segredos por trás da área desse display desenhando linhas nas bordas.

A instruções podem ser gráficas ou de escrita em um texto, que na linguagem de computação é chamado de string, os textos sempre deve ser escrito entre aspas duplas.



## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!



As instruções drawHLine é uma instrução gráfica que desenha uma linha horizontal, veja como é descrita no manual de referência, ela têm três parâmetros x,y,w.

O x,y indica a posição do primeiro pixel, a referência, o w indica o número de pixels para completar a largura da linha.

O x é a distância da borda lateral até o pixel, seria o número da colunas.

O y é a distância da borda superior até o pixel, seria o número da linhas.

A primeira instrução desenha uma linha horizontal do pixel na posição 0,0 mais 127 pixels.

Note que a largura total dessa linha é de 128, exatamente a largura da tela.

Note a cor, nessas linhas iniciais a cor é amarela.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

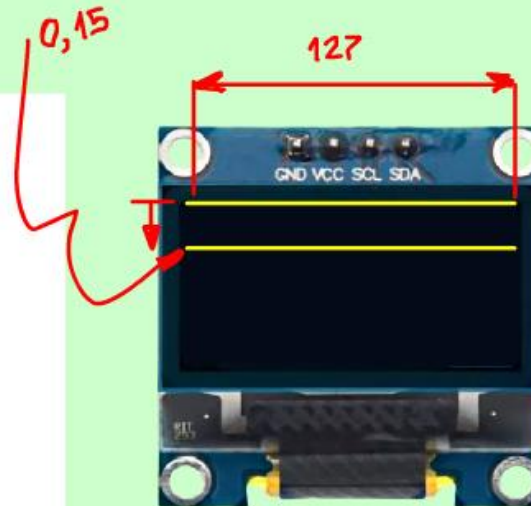
### 12. FIM DA ÁREA AMARELA.

```

6 void draw()
7 {
8
9   u8g.setFont(u8g_font_8x13B);
10  u8g.drawLine(0,0,127);
11  u8g.drawLine(0,15,127);
12  u8g.drawLine(0,16,127);
13  u8g.drawLine(0,63,127);
14  u8g.drawLine(0,0,15);
15  u8g.drawLine(0,16,47);
16  u8g.drawLine(127,0,16);
17  u8g.drawLine(127,16,48);
18  u8g.drawString(40, 12, "Simmm1");
19  u8g.drawString(3, 29, "Pro Bairros.");
20  u8g.drawString(3, 42, "Divirta-se!");
21 }

```

`u8g.drawLine(0,15,127);` //(x,y,w) linha H começando no pixel xy mais w pixel



A segunda linha tem os parâmetros 0,15,127.

Inicia na coluna zero, só é deslocada 15 linhas para baixo, e continua com a largura de mais 127 pixel, esse é o fim da área desenhada em amarelo.

Tudo que for desenhado ou escrito nessa área vai sair na cor amarela, é uma área reservada para o cabeçalho da tela, achei interessante esse truque para dar um brilho ao display sem complicar.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 13. ESCREVENDO UM TEXTO.


```

6 void draw()
7 {
8
9   u8g.setFont(u8g_font_9x13B);
10  u8g.drawHLine(0,0,127);
11  u8g.drawHLine(0,15,127);
12  u8g.drawHLine(0,16,127);
13  u8g.drawHLine(0,63,127);
14  u8g.drawVLine(0,0,15);
15  u8g.drawVLine(0,16,47);
16  u8g.drawVLine(127,0,16);
17  u8g.drawVLine(127,16,48);
18  u8g.drawString(40, 12, "Simmm1");
19  u8g.drawString(3, 29, "Pro Bairros.");
20  u8g.drawString(3, 42, "Divirta-se!");
21 }

```

*Handwritten annotations:*  
 - A red arrow labeled "setFont" points to line 9.  
 - A red arrow labeled "STRING" points to the string "Simmm1" in line 18.  
 - Below line 21, the text "x y" is written in red, with a red arrow pointing to the coordinates (40, 12) in line 18.  
 - Below line 21, the text "STRING" is written in red, with a red arrow pointing to the string "Simmm1" in line 18.

`u8g.drawString(40, 12, "Simmm1");// //escreve na posição (x,y, "strg") xy é a base`

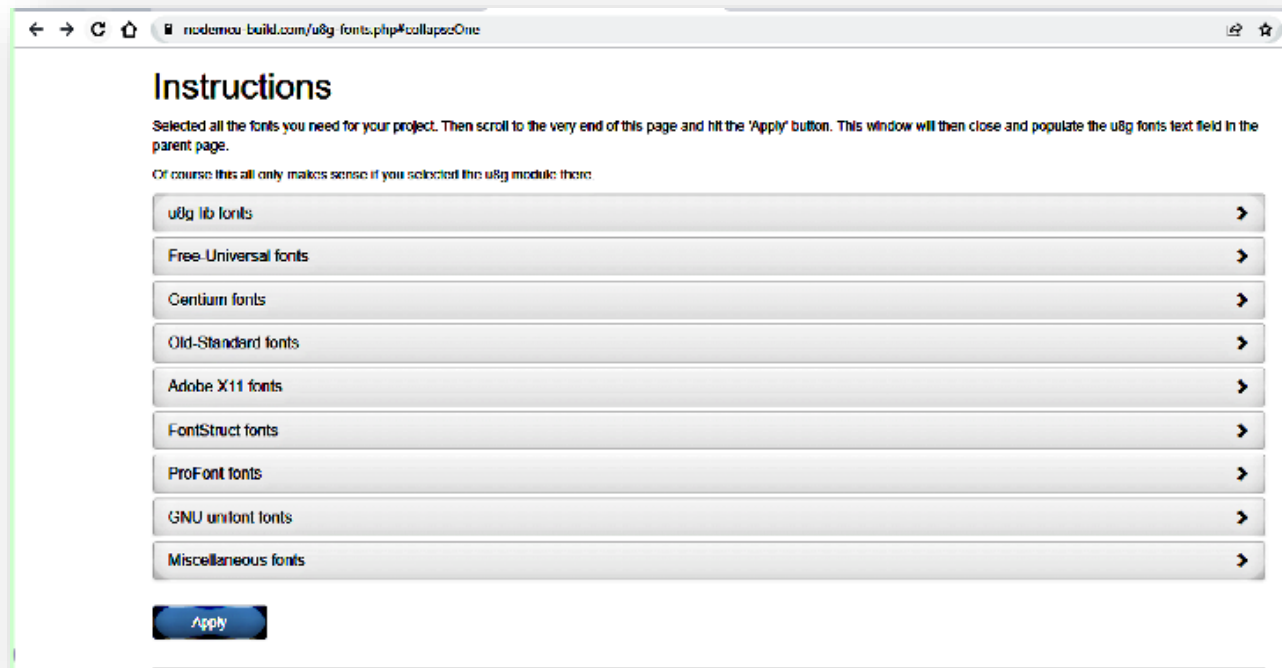


Note que tem um texto escrito na área em amarelo, a função de escrever um texto é chamada de drawstring, string significa texto lembra.

Essa instrução tem três parâmetros também, xy são a coluna e a linha do pixel inicial e o último parâmetro entre aspas duplas é a string, o texto propriamente dito.

Note que antes de escrever o texto eu tenho que declarar o tipo de fonte, isso é feito na função setfont.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!



Existem mais fontes disponíveis, eu deixei na descrição desse vídeo o endereço onde você pode ver todas as fontes disponíveis.

Um detalhe interessante é que o pixel inicial está posicionado no pé mais a esquerda da letra, eu gostaria que fosse no topo.

Olha as fontes: <https://nodemcu-build.com/u8g-fonts.php#collapseOne>

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 14.A LINHA VERTICAL.

```

6 void draw()
7 {
8
9  u8g.setFont(u8g_font_8x13B);
10 u8g.drawLine(0,0,127);
11 u8g.drawLine(0,15,127);
12 u8g.drawLine(0,16,127);
13 u8g.drawLine(0,63,127);
14 u8g.drawLine(0,0,15);
15 u8g.drawLine(0,16,47);
16 u8g.drawLine(127,0,16);
17 u8g.drawLine(127,16,48);
18 u8g.drawStr(40, 12, "Simmm1");
19 u8g.drawStr(3, 29, "Pro Bairros.");
20 u8g.drawStr(3, 42, "Divirta-se!");
21 }

```

`u8g.drawLine(0,0,15);` // (x,y,L) linha V começando no pixel xy mais L pixel

Veja que depois de desenhar as quatro linhas horizontais o programa desenha 4 linhas verticais, é isso que faz a instrução `drawVline`.

Essa instrução tem os parâmetros similares aos parâmetros da linha horizontal, mas agora são `x,y,L`, `x` o número da coluna, `y` o número da linha do pixel inicial, mas agora tem o parâmetro `L`, a altura, o número de pixel na vertical para completar da altura a linha.

Note que a primeira linha vertical está posicionada no início da área amarela, coluna zero, linha zero e altura de 15 pixel a largura da área amarela.

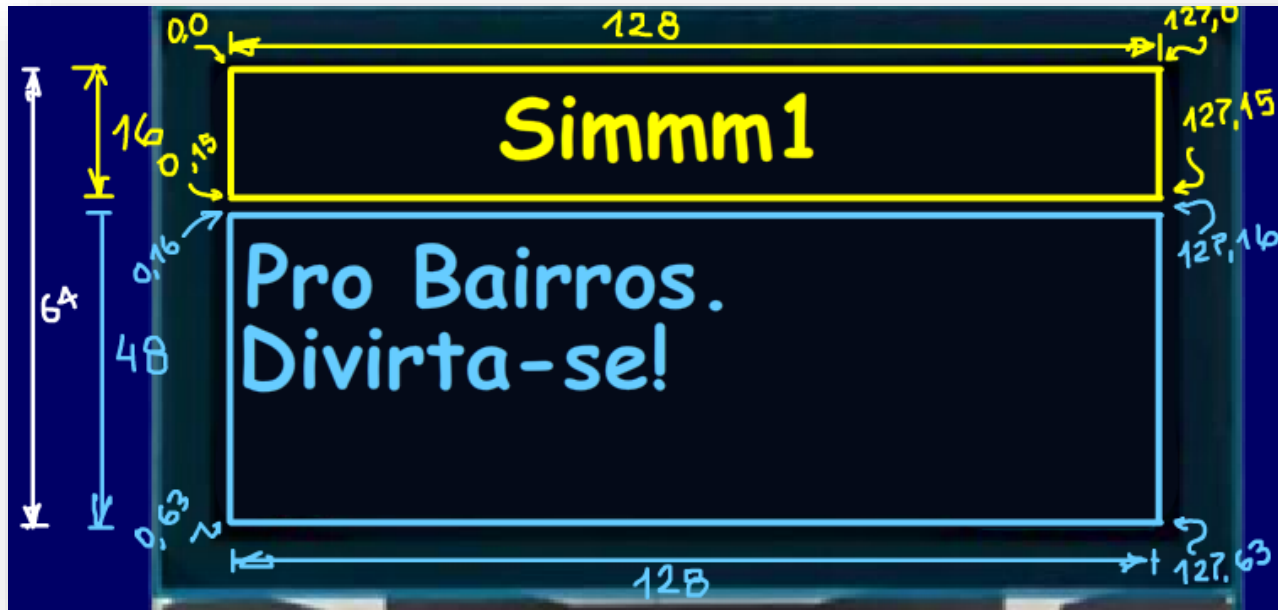
Viu como a lógica se repete, então conhecendo essas funções básicas você vai saber interpretar todas as outras

funções, essa é a intenção desse tutorial dar a base para que você monte o seu conhecimento.

Todos os programas exemplos deste tutorial estão no PDF, é só copiar e colar na sua ide, estão nos anexos.

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 15. RESUMO DAS ÁREAS DO DISPLAY.



Na figura você pode ver as áreas do display mostrando os pixels dos pontos extremos com largura e altura.

Veja os limites da área em amarelo, sua largura total é de 128 pixels, mas o limite da primeira linha é 127, claro um pixel foi desenhado na origem dada por x,y, esse é um detalhe importante que pouca gente fala por aí.

Veja os limites da área azul.

Veja agora algumas outras funções úteis bem explicadinhas

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 16. FUNÇÃO DRAWLINE

#### drawLine

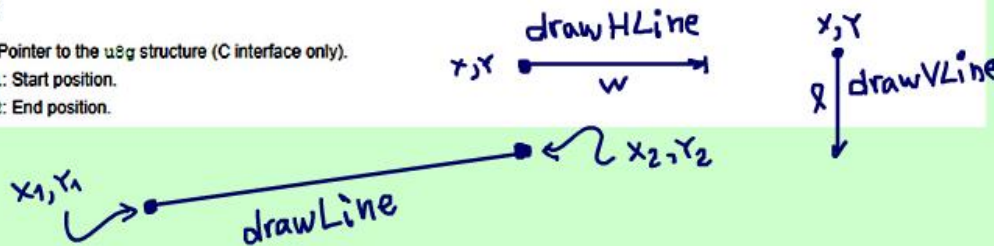
- C++ Prototype `void U8GLIB::drawLine(u8g_uint_t x1, u8g_uint_t y1, u8g_uint_t x2, u8g_uint_t y2)`
- C Prototype `void u8g_DrawLine(u8g_t *u8g, u8g_uint_t x1, u8g_uint_t y1, u8g_uint_t x2, u8g_uint_t y2)`

#### Description

Draw a line from  $(x1, y1)$  to  $(x2, y2)$ . There are no restrictions on the start end end position. This procedure uses the current color index to draw the line. For a monochrome display, the color index 0 will usually clear a pixel and the color index 1 will set a pixel.

#### Arguments:

- `u8g`: Pointer to the `u8g` structure (C interface only).
- `x1, y1`: Start position.
- `x2, y2`: End position.



Existem um monte de funções que desenham figuras, drawLine desenha uma linha a partir da informação do pixel inicial  $x1, y1$  e do pixel final  $x2, y2$ .

Essa é outra forma de desenhar linhas, o que você acha, essa função facilita ou não?

Se for desenhar linhas verticais e horizontais eu acho mais simples usar as funções drawHLine e drawVLine, mas se você quiser desenhar uma linha inclinada, então essa é a solução.

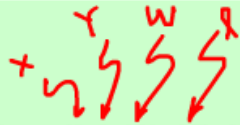
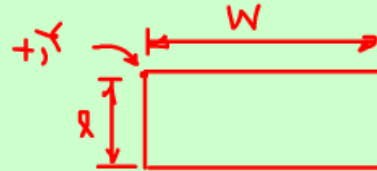
## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 17.A FUNÇÃO DRAWFRAME

```

6 void draw()
7 {
8
9   u8g.setFont(u8g_font_8x13B);
10  u8g.drawFrame(0,0,127,15);
11  u8g.drawFrame(0,16,127,47);
12  u8g.drawStr(40, 12, "Simmm2");
13  u8g.drawStr(3, 29, "Pro Bairros.");
14  u8g.drawStr(3, 42, "Divirta-se!");
15 }

```



u8g.drawFrame(0,0,127,15); // (x,y,w,L) linha R com raio R começando no pixel xy, largura W altura L

Agora vou montar outro programinha que mostra uma segunda forma de desenhar os retângulos, usando a função drawframe.

Essa função desenha um retângulo completo, veja os parâmetros, x,y,w,l.

Você consegue identificar todos os parâmetros?

Isso mesmo desenha um retângulo com o pixel inicial na posição x,y; com largura w e altura L, note que o pixel inicial está no topo.



Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

## 18.O PROGRAMA DESENHADO FRAMES.

```
6 void draw()
7 {
8
9   u8g.setFont(u8g_font_8x13B);
10  u8g.drawFrame(0,0,127,15);
11  u8g.drawFrame(0,16,127,47);
12  u8g.drawStr(40, 12, "Simmm1");
13  u8g.drawStr(3, 29, "Pro Bairros.");
14  u8g.drawStr(3, 42, "Divirta-se!");
15 }
```



Veja a versão da rotina draw do programa anterior usando a função drawframe, desenhar frame facilita, reduz o número de instruções.

Esse programa está no pdf também.

Não mudou nada na tela, mas o software ficou mais limpo.

Você pode copiar somente a função draw e colar no seu programa, note que o resto continua a mesma coisa.

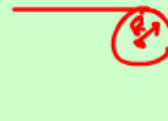
## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 19. FUNÇÃO FRAME COM FRAMES REDONDOS

```

6 void draw()
7 {
8
9   u8g.setFont(u8g_font_8x13B);
10  u8g.drawRFrame(0,0,127,15,4);
11  u8g.drawRFrame(0,16,127,47,4);
12  u8g.drawStr(40, 12, "Simmm3");
13  u8g.drawStr(3, 29, "Pro Bairros.");
14  u8g.drawStr(3, 42, "Divirta-se!");
15 }

```



`u8g.drawRFrame(0,0,127,15,4);` //(x,y,w,L,R) linha R com raio R começando no pixel xy, largura W altura L

Agora novo exemplo de programa, vou enfeitar tudo arredondar os cantos dos retângulos.

O número de funções são muitas, outra bem interessante é a função Frame que arredonda os cantos, nessa função tem um parâmetro a mais, o parâmetro R que significa o raio do arredondamento do canto.

Veja o terceiro programa agora desenhado os frames arredondados, é só acrescentar um parâmetro e a sua tela ficou cheia de estilo, tá lá no pdf.



## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 20.A FUNÇÃO PRINT QUE MOSTRA UMA VARIÁVEL.

```

6  int n=0;//variável
7  void draw()
8  {
9
10 u8g.setFont(u8g_font_8x13B);
11 u8g.drawRFrame(0,0,127,15,4);
12 u8g.drawRFrame(0,16,127,47,4);
13 u8g.drawStr(40, 12, "Simmm4");
14 u8g.drawStr(3, 29, "Pro Bairros.");
15 u8g.drawStr(3, 42, "Divirta-se!");
16 u8g.setPrintPos(50, 55);
17 u8g.println(n);
18 }

```

*int n=0*

**u8g.setPrintPos(50, 55);**  
**u8g.println(n);**

```

36 void loop()
37 {
38   u8g.firstPage();
39   do
40   {
41     draw();
42   }
43   while ( u8g.nextPage() );
44   delay(1000);
45   n=n+1;
46 }

```

*incremento soma +1*  
*(1000)*



Veja agora o programa exemplo que escreve o valor numérico de uma variável no display.

Se você quiser mostrar uma variável terá que usar a função print, que é uma função padrão da linguagem “C”, antes de imprimir o valor na tela você deverá posicionar o ponto inicial desse valor.

Veja o exemplo da figura, aqui eu criei uma variável n do tipo inteiro,

No programa eu imprimo essa variável.

Depois de imprimir eu incremento a variável n, isso dentro do loop depois da rotina draw, dentro da rotina draw o DELAY não funciona bem.

Note que para deixar a contagem mais dinâmica eu apressei o DELAY, essa rotina pode ser a base para um temporizador, ou contador ou quantas coisas mais a sua mente inventar!

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

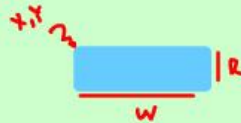
### 21. A FUNÇÃO DRAWBOX.

```

6  int n=0;//variável
7  void draw()
8  {
9
10  u8g.setFont(u8g_font_8x13B);
11  u8g.drawRFrame(0,0,127,15,4);
12  u8g.drawRFrame(0,16,127,47,4);
13  u8g.drawStr(40, 12, "Simm5");
14  u8g.drawStr(3, 29, "Pro Bairros.");
15  u8g.drawStr(3, 42, "Divirta-se!");
16  if (n>0){u8g.drawBox(110,55,10,4);}
17  if (n>1){u8g.drawBox(110,50,10,4);}
18  if (n>2){u8g.drawBox(110,45,10,4);}
19  if (n>3){u8g.drawBox(110,40,10,4);}
20  }

```

$x$   $y$   $w$   $h$   
**u8g.drawBox(110,55,10,4);**



E agora o último programinha desse tutorial que mostra a função drawbox que fora o nome tem a mesma estrutura do drawFrame, mas agora preenche o retângulo desenhado, dá para desenhar bloquinhos, como um barrgráfico.

Veja na figura o que cada parâmetro significa, são os mesmos do drawFrame.

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

```

6  int n=0;//variável ← int n
7  void draw()
8  {
9
10     u8g.setFont(u8g_font_8x13B);
11     u8g.drawRFrame(0,0,127,15,4);
12     u8g.drawRFrame(0,16,127,47,4);
13     u8g.drawStr(40, 12, "Simmm5");
14     u8g.drawStr(3, 29, "Pro Bairros.");
15     u8g.drawStr(3, 42, "Divirta-se!");
16     if (n>0){u8g.drawBox(110,55,10,4);}
17     if (n>1){u8g.drawBox(110,50,10,4);}
18     if (n>2){u8g.drawBox(110,45,10,4);}
19     if (n>3){u8g.drawBox(110,40,10,4);}
20 }

```

$x$   $y$   $w$   $h$   
**`u8g.drawBox(110,55,10,4);`**

Agora para deixar mais dinâmico eu vou aumentar e diminuir a barrinha como um VU e para alterar a posição do bloquinho eu vou usar uma variável do tipo inteiro, então pode ser alterada pelo programa.

Com que eu faço isso, usando a variável  $n$ , claro que existem muitas outras formas de fazer isso, o objetivo desse exemplo é mostrar que o limite é a sua imaginação.

Dentro do draw eu pergunto o valor da variável e conforme o valor eu vou desenhando os bloquinhos um sobre os outros.

Quando a variável  $n$  for maior do que 4 eu faço ela voltar para o valor 0 e pronto, começa tudo novamente.

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

## 22. INCREMENTANDO A VARIÁVEL NO LOOP.

```

6  int n=0;//variável
7  void draw()
8  {
9
10     u8g.setFont(u8g_font_8x13B);
11     u8g.drawRFrame(0,0,127,15,4);
12     u8g.drawRFrame(0,16,127,47,4);
13     u8g.drawStr(40, 12, "Simmm5");
14     u8g.drawStr(3, 29, "Pro Bairros.");
15     u8g.drawStr(3, 42, "Divirta-se!");
16     if (n>0){u8g.drawBox(110,55,10,4);}
17     if (n>1){u8g.drawBox(110,50,10,4);}
18     if (n>2){u8g.drawBox(110,45,10,4);}
19     if (n>3){u8g.drawBox(110,40,10,4);}
20 }

```

*int n*

```

38 void loop()
39 {
40     u8g.firstPage();
41     do
42     {
43         draw();
44     }
45     while ( u8g.nextPage() );
46     delay(100);
47     n=n+1;
48     if (n>4){
49         n=0;
50     }

```

*incremento*  
*compara*  
*zera*

`u8g.drawBox(110,55,10,4);`

A variável n é incrementada no loop.

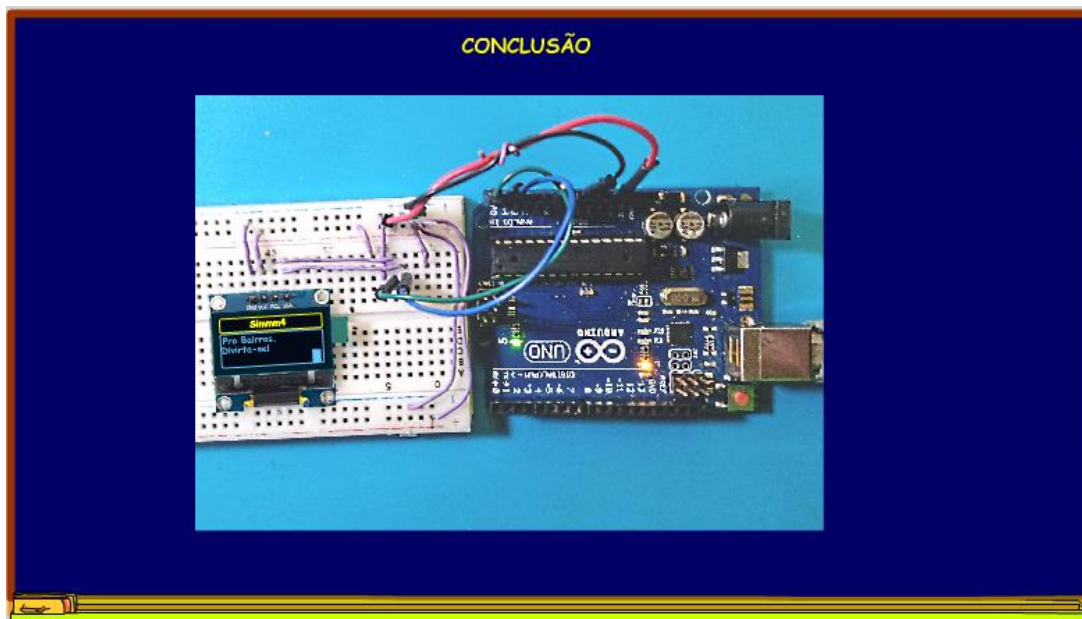
É lá que é feita a comparação se a variável n é maior do 4 e então é zerada.

Veja que efeito, então é possível desenhar figuras em movimento, fazer joguinhos e muito mais, tudo aquilo que a sua mente puder imaginar.



Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 23. CONCLUSÃO.



Você viu nesse tutorial como usar as principais funções do display OLE 128x64 com comunicação I2C Azul e amarelo, esse vai ser um display que eu vou usar muito em outros tutoriais, agora vai ficar fácil construir instrumentos como voltímetros, amperímetros e até a carga ativa, com esse display vai ficar muito fácil passar as informações para o operador, aguarde...

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

## 24. APÊNDICE

Aqui estão todos os programas



## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 1 DESENHANDO LINHAS.

```
//Programa: TESTE DAS FUNÇÕES u8g DO DISPLAY OLE
//Programa teste Desenhando Linhas
//Autor: Professor Bairros
#include <U8glib.h>
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); //Definicoes do display Oled
void draw()
{
  //Comandos graficos para o display devem ser colocados aqui
  u8g.setFont(u8g_font_8x13B); //normal
  u8g.drawHLine(0,0,127); // (x,y,L) linha H começando no pixel xy mais L pixel
  u8g.drawHLine(0,15,127); // (x,y,L) linha H começando no pixel xy mais L pixel
  u8g.drawHLine(0,16,127); // (x,y,L) linha H começando no pixel xy mais L pixel
  u8g.drawHLine(0,63,127); // (x,y,L) linha H começando no pixel xy mais L pixel
  u8g.drawVLine(0,0,15); // (x,y,L) linha V começando no pixel xy mais L pixel
  u8g.drawVLine(0,16,47); // (x,y,L) linha V começando no pixel xy mais L pixel
  u8g.drawVLine(127,0,16); // (x,y,L) linha V começando no pixel xy mais L pixel
  u8g.drawVLine(127,16,48); // (x,y,L) linha V começando no pixel xy mais L pixel
  u8g.drawStr(40, 12, "Simmm1"); // // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 29, "Pro Bairros."); // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 42, "Divirta-se!"); // escreve na posição (x,y, "strg") xy é a base
}
void setup()
{
  Serial.begin(9600);
  //inicializa Display Oled
  if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
    u8g.setColorIndex(255); //volta 40 white
  }
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

```
else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
  u8g.setColorIndex(3); //volta 2max intensity
}
else if ( u8g.getMode() == U8G_MODE_BW ) {
  u8g.setColorIndex(1); //volta 1 pixel on esse é o mais comum
}
else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {
  u8g.setHiColorByRGB(255, 255, 255); //volta 48
}
}
void loop()
{
  u8g.firstPage(); //Chama a rotina de desenho na tela
  do
  {
    draw(); //chama a rotina que desenha a tela
  }
  while ( u8g.nextPage() ); //fica desenhado até o fim da memória.
  delay(10000); //delay em segundos
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 2 DESENHANDO FRAMES

```
//Programa: TESTE DAS FUNÇÕES u8g DO DISPLAY OLE
//Programa teste Desenhando Frames
//Autor: Professor Bairros
#include <U8glib.h>
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); //Definicoes do display Oled
void draw()
{
  //Comandos graficos para o display devem ser colocados aqui
  u8g.setFont(u8g_font_8x13B); //normal
  u8g.drawFrame(0,0,127,15); //(x,y,w,L)linha R começando no pixel xy, largura W altura L
  u8g.drawFrame(0,16,127,47); //(x,y,w,L)linha R começando no pixel xy, largura W altura L
  u8g.drawStr(40, 12, "Simmm2"); // //escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 29, "Pro Bairros."); //escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 42, "Divirta-se!"); //escreve na posição (x,y, "strg") xy é a base
}
void setup()
{
  Serial.begin(9600);
  //inicializa Display Oled
  if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
    u8g.setColorIndex(255); //volta 40 white
  }
  else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
    u8g.setColorIndex(3); //volta 2max intensity
  }
  else if ( u8g.getMode() == U8G_MODE_BW ) {
    u8g.setColorIndex(1); //volta 1 pixel on esse é o mais comum
  }
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

```
else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {  
    u8g.setHiColorByRGB(255, 255, 255); //volta 48  
}  
}  
void loop()  
{  
    u8g.firstPage(); //Chama a rotina de desenho na tela  
    do  
    {  
        draw(); //chama a rotina que desenha a tela  
    }  
    while ( u8g.nextPage() ); //fica desenhado até o fim da memória.  
    delay(10000); //delay em segundos  
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 3 DESENHANDO FRAME COM RAI0

```
//Programa: TESTE DAS FUNÇÕES u8g DO DISPLAY OLED
//Programa teste Desenhando Frames com raio
//Autor: Professor Bairros
#include <U8glib.h>
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); //Definicoes do display Oled
void draw()
{
  //Comandos graficos para o display devem ser colocados aqui
  u8g.setFont(u8g_font_8x13B); //normal
  u8g.drawRFrame(0,0,127,15,4); // (x,y,w,L,R) linha R com raio R começando no pixel xy, largura W altura L
  u8g.drawRFrame(0,16,127,47,4); // (x,y,w,L,R) linha R com raio R começando no pixel xy, largura W altura L
  u8g.drawStr(40, 12, "Simmm3"); // // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 29, "Pro Bairros."); // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 42, "Divirta-se!"); // escreve na posição (x,y, "strg") xy é a base
}
void setup()
{
  Serial.begin(9600);
  //inicializa Display Oled
  if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
    u8g.setColorIndex(255); //volta 40 white
  }
  else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
    u8g.setColorIndex(3); //volta 2max intensity
  }
  else if ( u8g.getMode() == U8G_MODE_BW ) {
    u8g.setColorIndex(1); //volta 1 pixel on esse é o mais comum
  }
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

```
else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {  
    u8g.setHiColorByRGB(255, 255, 255); //volta 48  
}  
}  
void loop()  
{  
    u8g.firstPage(); //Chama a rotina de desenho na tela  
    do  
    {  
        draw(); //chama a rotina que desenha a tela  
    }  
    while ( u8g.nextPage() ); //fica desenhado até o fim da memória.  
    delay(10000); //delay em segundos  
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 4 PRINT VARIÁVEL

```
//Programa: TESTE DAS FUNÇÕES u8g DO DISPLAY OLED
//Programa teste Print variável
//Autor: Professor Bairros
#include <U8glib.h>
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); //Definicoes do display Oled
int n=0; //variável
void draw()
{
  //Comandos graficos para o display devem ser colocados aqui
  u8g.setFont(u8g_font_8x13B); //normal
  u8g.drawRFrame(0,0,127,15,4); // (x,y,w,L,R) linha R com raio R começando no pixel xy, largura W altura L
  u8g.drawRFrame(0,16,127,47,4); // (x,y,w,L,R) linha R com raio R começando no pixel xy, largura W altura L
  u8g.drawStr(40, 12, "Simmm4"); // // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 29, "Pro Bairros."); // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 42, "Divirta-se!"); // escreve na posição (x,y, "strg") xy é a base
  u8g.setPrintPos(50, 55); // escreve na posição (x,y, "strg") xy é a base
  u8g.println(n);
}
void setup()
{
  Serial.begin(9600);
  //inicializa Display Oled
  if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
    u8g.setColorIndex(255); //volta 40 white
  }
  else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
    u8g.setColorIndex(3); //volta 2max intensity
  }
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

```
else if ( u8g.getMode() == U8G_MODE_BW ) {
  u8g.setColorIndex(1); //volta 1 pixel on esse é o mais comum
}
else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {
  u8g.setHiColorByRGB(255, 255, 255); //volta 48
}
}
void loop()
{
  u8g.firstPage(); //Chama a rotina de desenho na tela
  do
  {
    draw(); //chama a rotina que desenha a tela
  }
  while ( u8g.nextPage() ); //fica desenhado até o fim da memória.
  delay(1000); //delay em segundos
  n=n+1;
}
```



## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### DESENHANDO BOX VARIÁVEL

```
//Programa: TESTE DAS FUNÇÕES u8g DO DISPLAY OLE
//Programa teste Box variável
//Autor: Professor Bairros
#include <U8glib.h>
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); //Definicoes do display Oled
int n=0; //variável
void draw()
{
  //Comandos graficos para o display devem ser colocados aqui
  u8g.setFont(u8g_font_8x13B); //normal
  u8g.drawRFrame(0,0,127,15,4); // (x,y,w,L,R) linha R com raio R começando no pixel xy, largura W altura L
  u8g.drawRFrame(0,16,127,47,4); // (x,y,w,L,R) linha R com raio R começando no pixel xy, largura W altura L
  u8g.drawStr(40, 12, "Simmm5"); // // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 29, "Pro Bairros."); // escreve na posição (x,y, "strg") xy é a base
  u8g.drawStr(3, 42, "Divirta-se!"); // escreve na posição (x,y, "strg") xy é a base
  if (n>0){u8g.drawBox(110,55,10,4);}
  if (n>1){u8g.drawBox(110,50,10,4);}
  if (n>2){u8g.drawBox(110,45,10,4);}
  if (n>3){u8g.drawBox(110,40,10,4);}
}
void setup()
{
  Serial.begin(9600);
  //inicializa Display Oled
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

```
if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
  u8g.setColorIndex(255); //volta 40 white
}
else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
  u8g.setColorIndex(3); //volta 2max intensity
}
else if ( u8g.getMode() == U8G_MODE_BW ) {
  u8g.setColorIndex(1); //volta 1 pixel on esse é o mais comum
}
else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {
  u8g.setHiColorByRGB(255, 255, 255); //volta 48
}
}

void loop()
{
  u8g.firstPage(); //Chama a rotina de desenho na tela
  do
  {
    draw(); //chama a rotina que desenha a tela
  }
  while ( u8g.nextPage() ); //fica desenhado até o fim da memória.
  delay(100); //delay em segundos
```

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

```
n=n+1;  
if (n>4){  
  n=0;  
}  
}
```

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

### 25. CRÉDITOS

E por favor, se você não é inscrito, se inscreva e marque o sininho para receber as notificações do canal e não esqueça de deixar aquele like e compartilhar para dar uma força ao canal do professor bairros.

**Arthurzinho:** E não tem site.

Tem sim é [www.bairrospd.com](http://www.bairrospd.com) lá você encontra o PDF e tutoriais sobre esse e outros assuntos da eletrônica

E fique atento ao canal do professor bairros para mais tutoriais sobre eletrônica, até lá!

INSCRIÇÃO YOUTUBE: <https://www.youtube.com/@professorbairros>

VISITE O SITE DO PROFESSOR BAIROS LÁ TEM O PDF E MUITO MAIS

PARA AULAS ONLINE CONTATE VIA SITE

[www.bairrospd.com](http://www.bairrospd.com)

SOM: pop alegre Mysteries -30 (fonte YOUTUBE)

Comunicação (Parte 1): <https://youtu.be/A1HmMtz9QGg>

Comunicação (Parte 2 com I2C): <https://youtu.be/RLFshE9qzJM>

Manual de referência Funções u8glib: <https://code.google.com/archive/p/u8glib/wikis/userreference.wiki>

## Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

20230429 Desbrave esse fantástico display OLED azul e amarelo com comunicação I2C

Desbrave esse fantástico Display OLED azul e amarelo com comunicação I2C!

Nesse tutorial eu vou desvendar os segredos do Display OLE azul e amarelo com comunicação I2, um display muito simples e colorido que pode dar um charme extra aos seus projetos.

Assuntos relacionados.

Comunicação (Parte 1): <https://youtu.be/A1HmMtz9QGg>

Comunicação (Parte 2 com I2C): <https://youtu.be/RLFshE9qzJM>

Manual de referência Funções u8glib: <https://code.google.com/archive/p/u8glib/wikis/userreference.wiki>

Olha as fontes: <https://nodemcu-build.com/u8g-fonts.php#collapseOne>

Quanta teoria eu preciso para trabalhar com eletrônica?: <https://youtu.be/-5T6T3sljDo>

SEO: <https://youtu.be/iQN3k7VLOqM>

Display oled, display oled azul e amarelo, display oled com comunicação I2C,

YOUTUBE: <https://youtu.be/iQN3k7VLOqM>