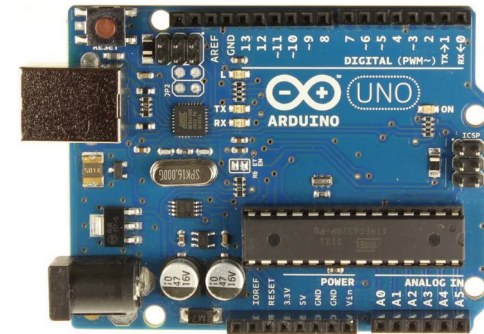


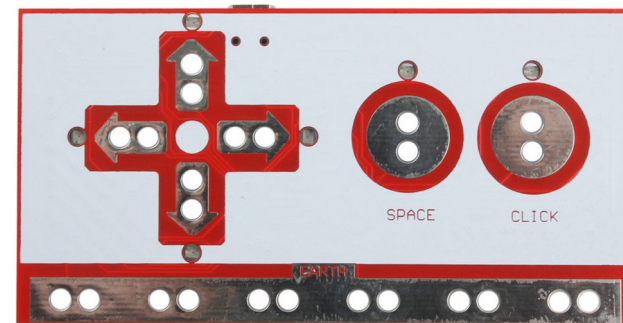


Ambiente (IDE) Arduino

E



LudoBot/ *Itty Bitty Buggy*



Esta aula está em:

[https://drive.google.com/drive/folders/
1EJycEY4pUVkyc5r631IpNtSLY2LQpM1y?
usp=sharing](https://drive.google.com/drive/folders/1EJycEY4pUVkyc5r631IpNtSLY2LQpM1y?usp=sharing)

e

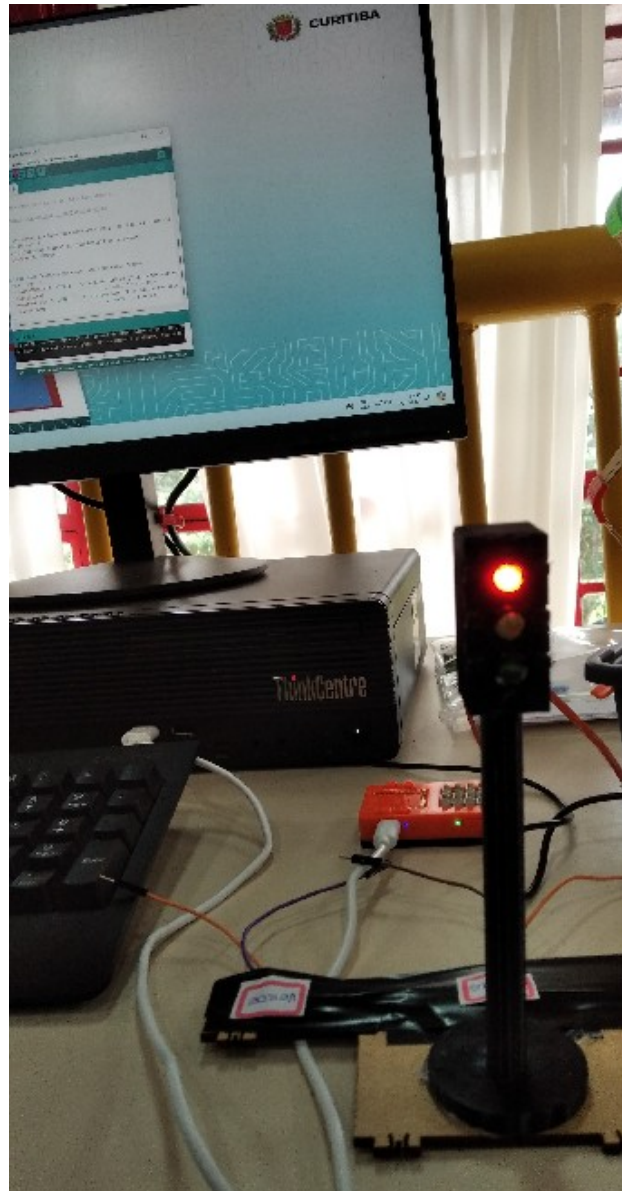
[https://tiaplicada.ufpr.br/wp-content/uploads/
2024/10/aula25outubro.zip](https://tiaplicada.ufpr.br/wp-content/uploads/2024/10/aula25outubro.zip)

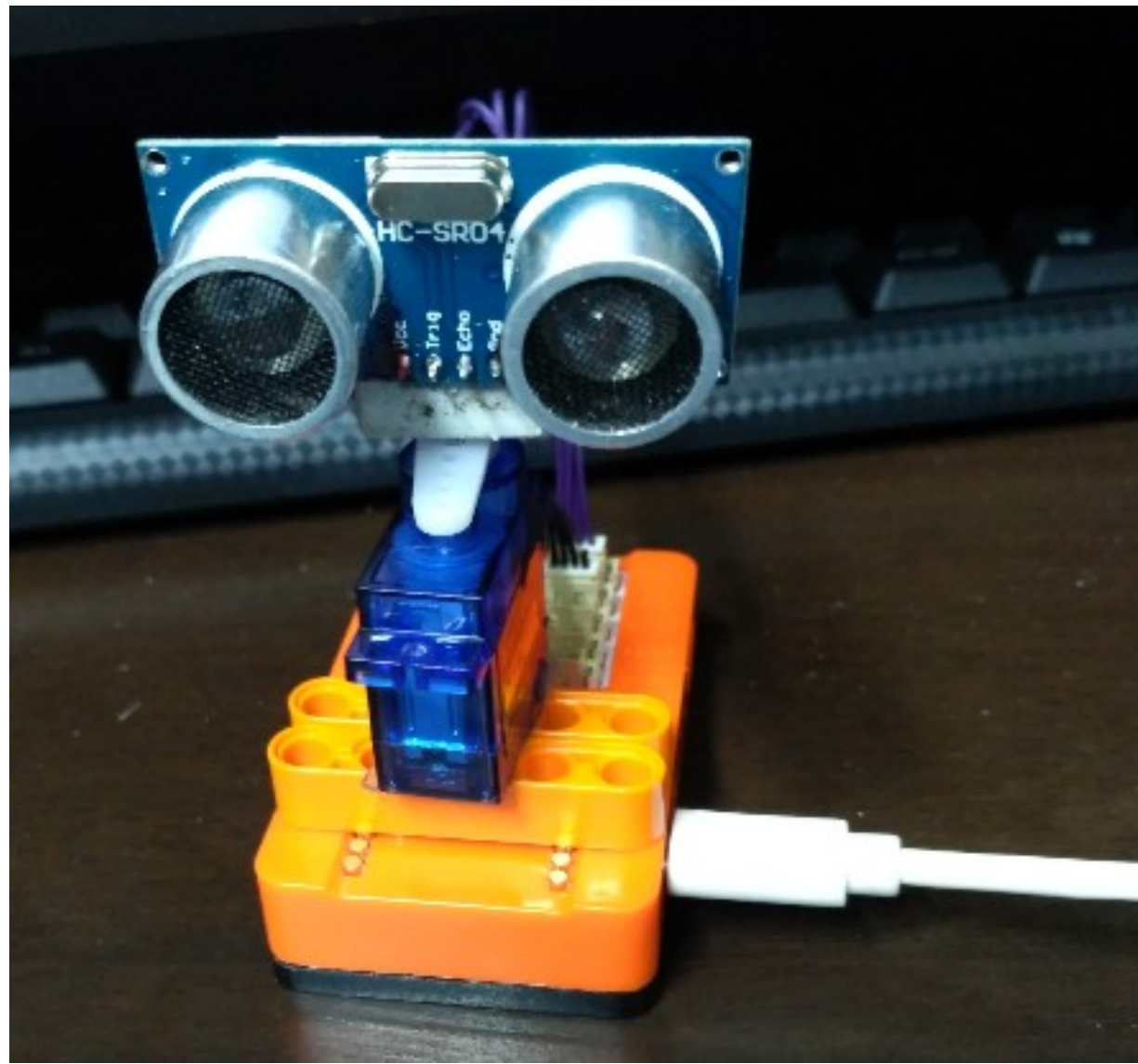
Ideia:

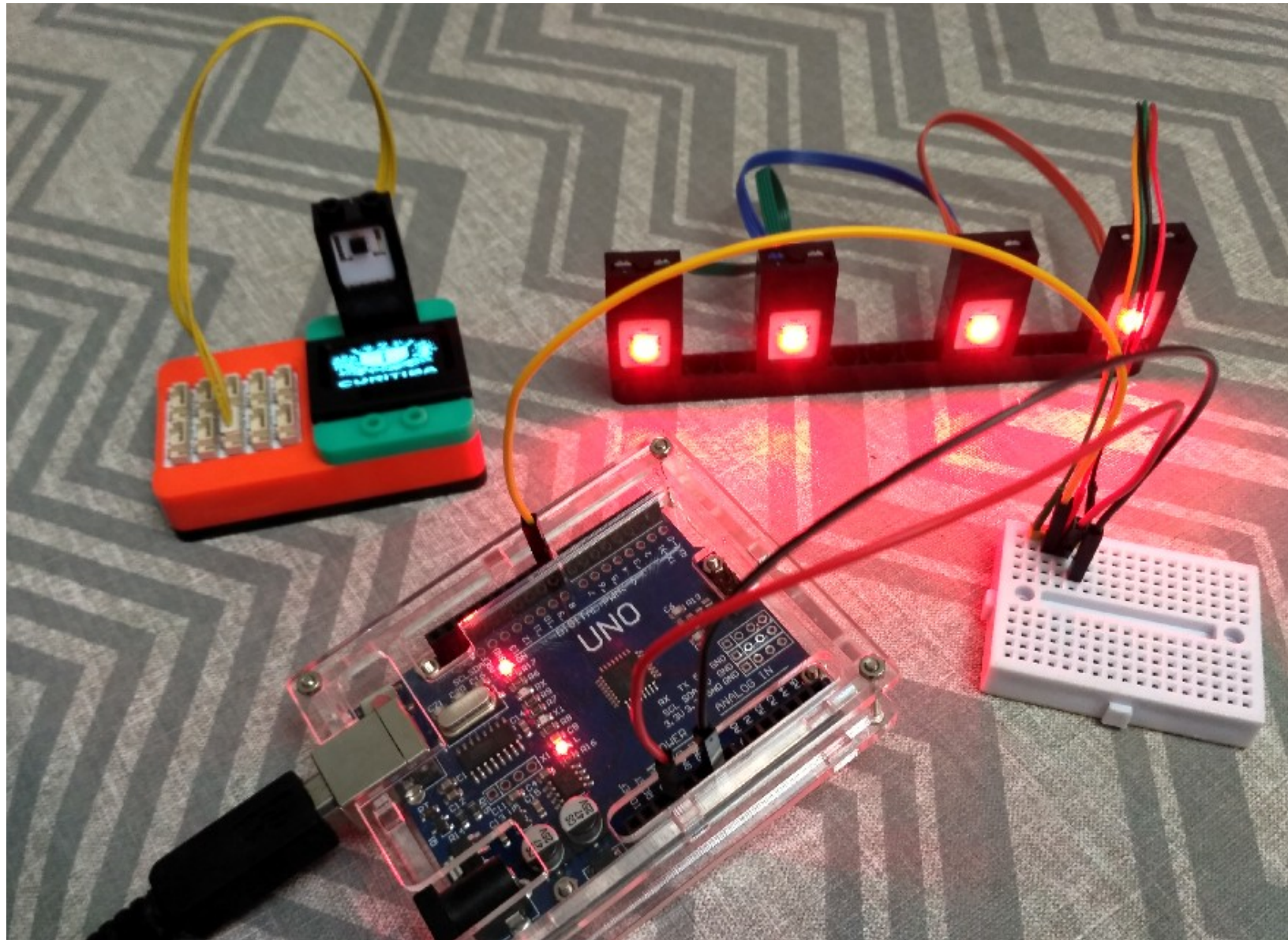
- utilizar um único ambiente/ um mesmo código para controle de criações no Arduino e no Ludobot/ Itty Bitty
- aproveitar os componentes disponíveis no *kit* Arduino e nos *kits* Ludobot e *Itty Bitty Buggy*
- ‘incrementar’ projetos com peças LEGO

STACKABLE WITH LEGO®









O que é necessário ?

Instalar a IDE do Microduino



The image shows the Microduino IDE installation window. It features the Arduino and Genuino logos at the top left. Below them is the text 'AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE'. The Microduino logo is prominently displayed in the center. To the right is a stack of four Microduino boards, each with a different colored top cover (red, blue, yellow, and teal). At the bottom, there is a row of twelve small robot icons in various colors. The status bar at the very bottom indicates 'Inicializando pacotes...'.

ARDUINO | **Genuino**

AN OPEN PROJECT WRITTEN, DEBUGGED,
AND SUPPORTED BY ARDUINO.CC AND
THE ARDUINO COMMUNITY WORLDWIDE

MICRODUINO

Arduino IDE for Microduino - 1.8.4-V 6.0

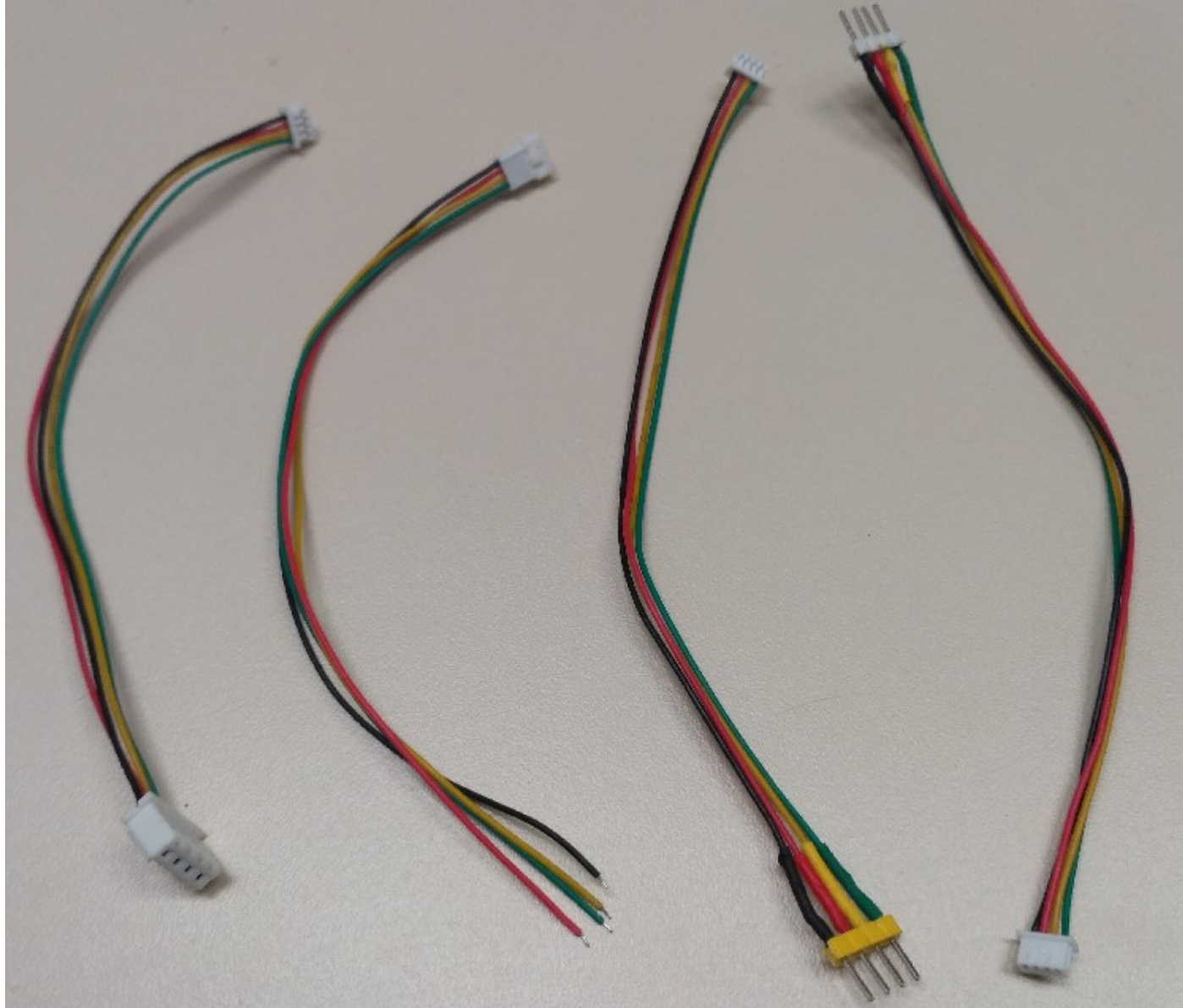
We do a lot of work related to the
libraries for better support of
Microduino and mCookie Modules

Inicializando pacotes...

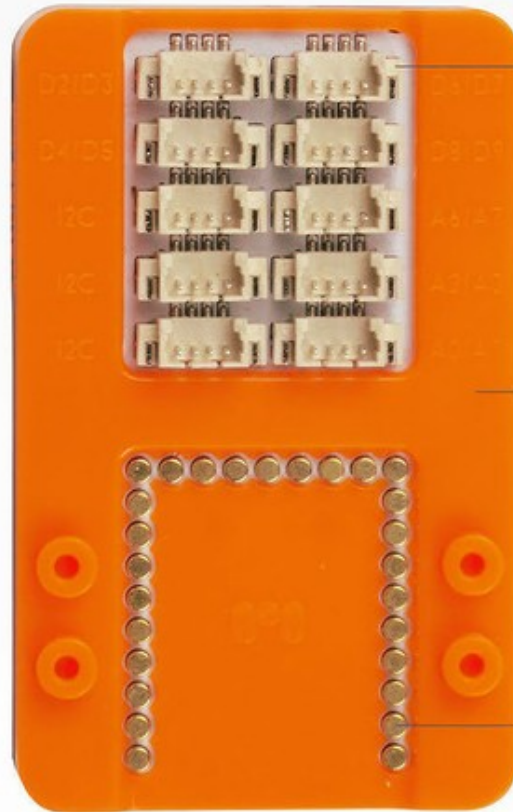
Usa a IDE do Arduino

- Configurar arquivos para as placas e importá-las
 - Ludobot / *Itty Bitty Buggy*
 - Makey Makey
 - BBC:microbit / IDEAbit
- Instalar bibliotecas
 - no caso do microbit é necessário usar conversores de níveis de tensão (e pode ocorrer de nem todas as bibliotecas funcionarem)

Cabos/ fios



mCenter+



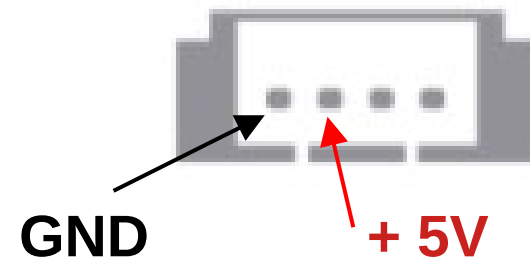
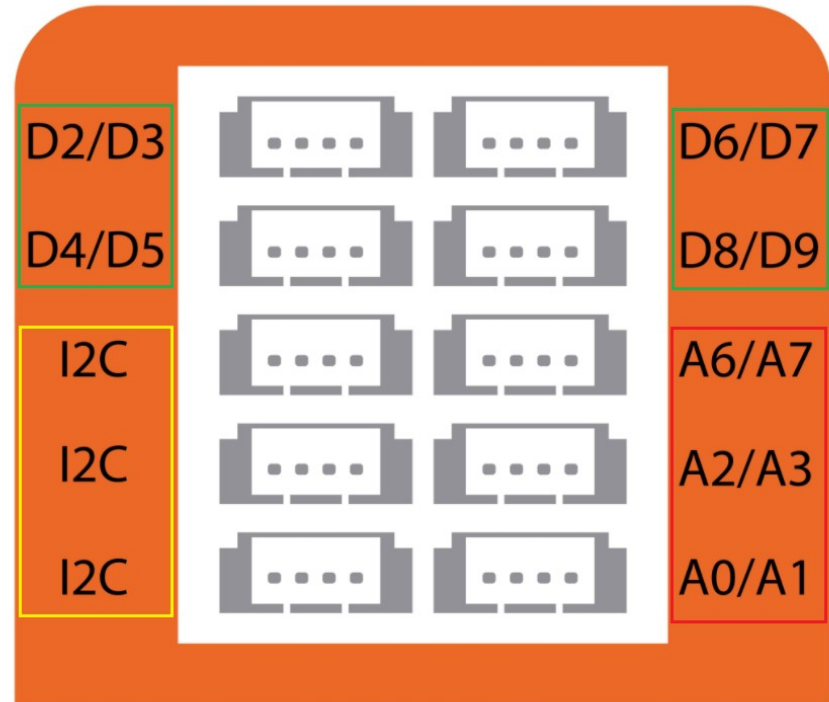
Portas:
Digitais
Analógicas
I2C

Microcontrolador
ATMEGA644PA
Clock 16Mhz

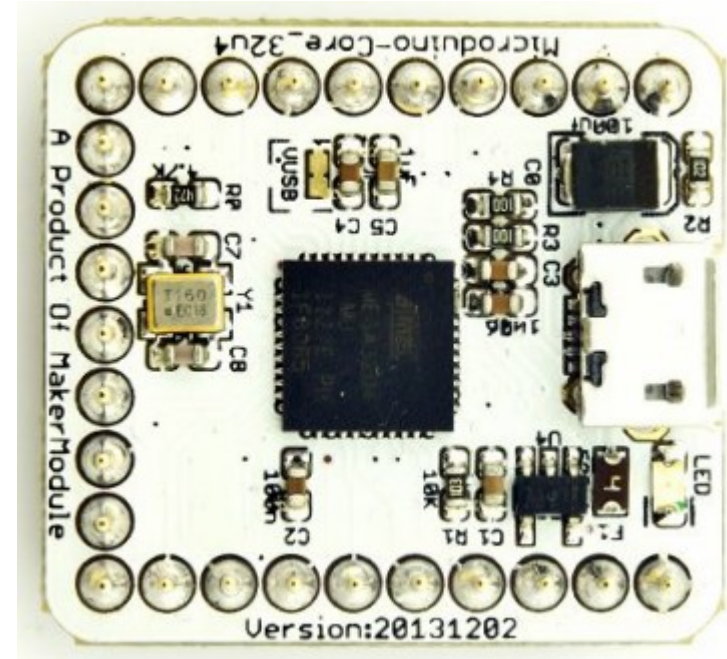
Conexão
Bluetooth

Bateria
850 mA

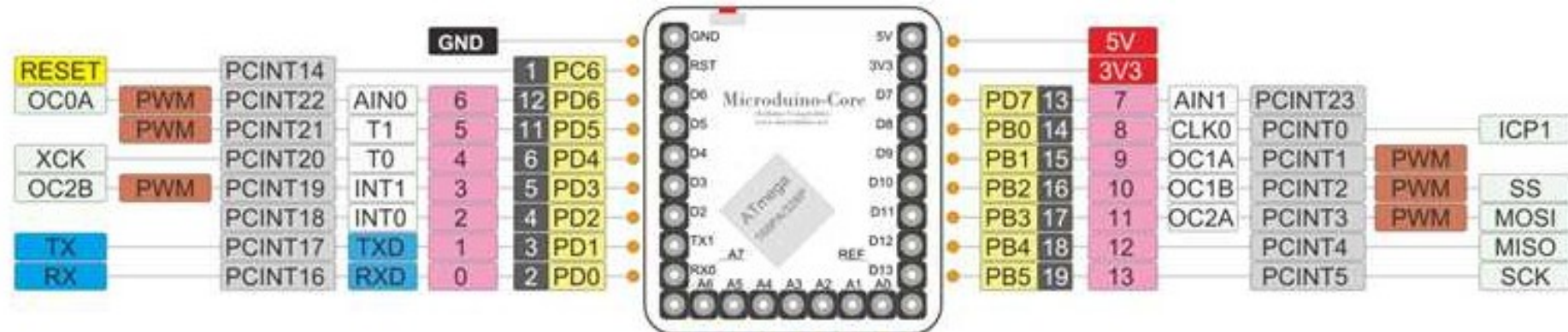
Conector
uPin27 magnético



Microduino



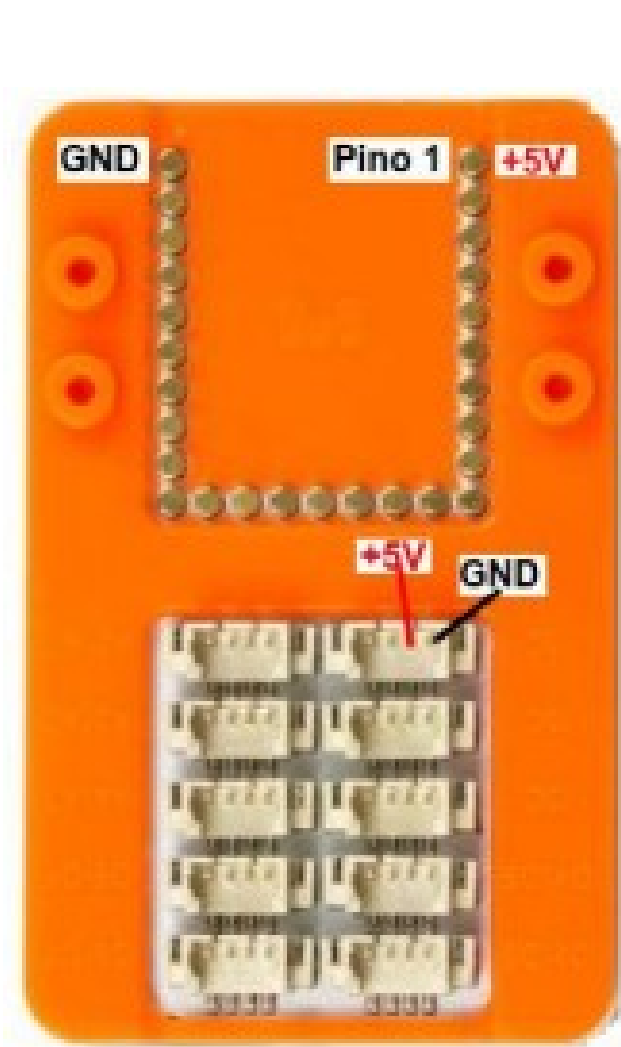
Visto por cima



Visto por baixo



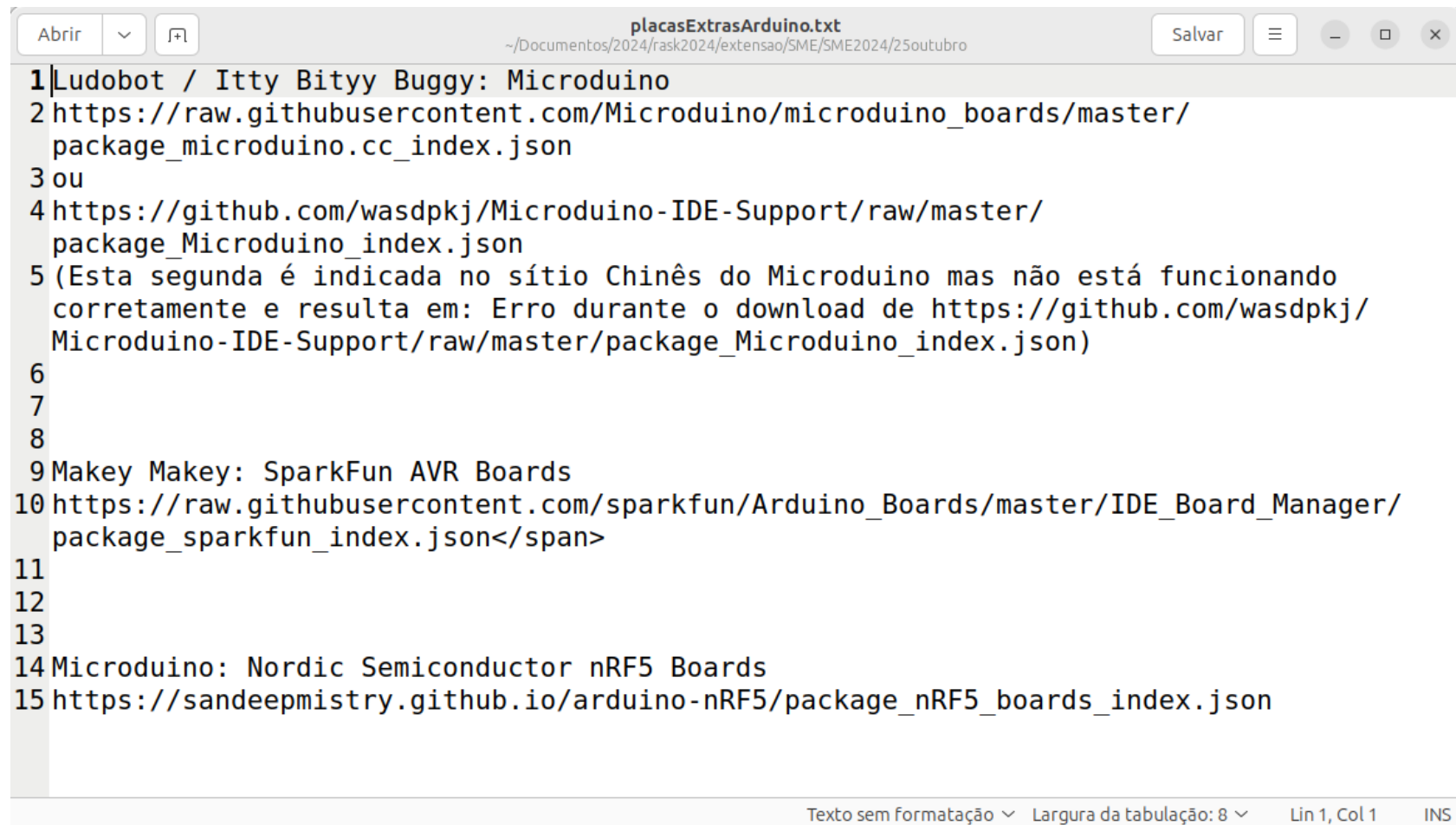
Pino	Nome	UNO	Digital	Analogico	Int	PWM	Serial	SPI	I2C	Energia
1	VCC	+5V								+5V
2	VCC	+3V3								+3.3V
3	(OC0A/AIN1)PB3	D7	D7			Sim				
4	(OC2B/ICP)PD6	D8	D8			Sim				
5	(OC1A)PD5	D9	D9			Sim				
6	(OC0B/SS)PB4	D10	D10			Sim		SS		
7	(MOSI)PB5	D11	D11					MOSI		
8	(MISO)PB6	D12	D12					MISO		
9	(SCK)PB7	D13	D13					SCK		
10	AREF	AREF								
11	(ADC7)PA7	A0	D24	A0						
12	(ADC6)PA6	A1	D25	A1						
13	(ADC5)PA5	A2	D26	A2						
14	(ADC4)PA4	A3	D27	A3						
15	(SDA)PC1	SDA	D20						SDA	
16	(SCL)PC0	SCL	D21						SCL	
17	(ADC1)PA1	A6	D30	A6						
18	(ADC0)PA0	A7	D31	A7						
19	(RXD0)PD0	D0	D0				0(RX)			
20	(TXD0)PD1	D1	D1				0(TX)			
21	(INT0)PD2	D2	D2		0		1(RX)			
22	(INT1)PD3	D3	D3		1		1(TX)			
23	(XCK0/T0)PB0	D4	D4							
24	(CLKO/T1)PB1	D5	D5							
25	(INT2/AIN0)PB2	D6	D6		2					
26	RESET	RST								
27	GND	GND								GND



Configuração

(só precisa fazer no primeiro uso da placa)

Configurar arquivos para as placas e importá-las

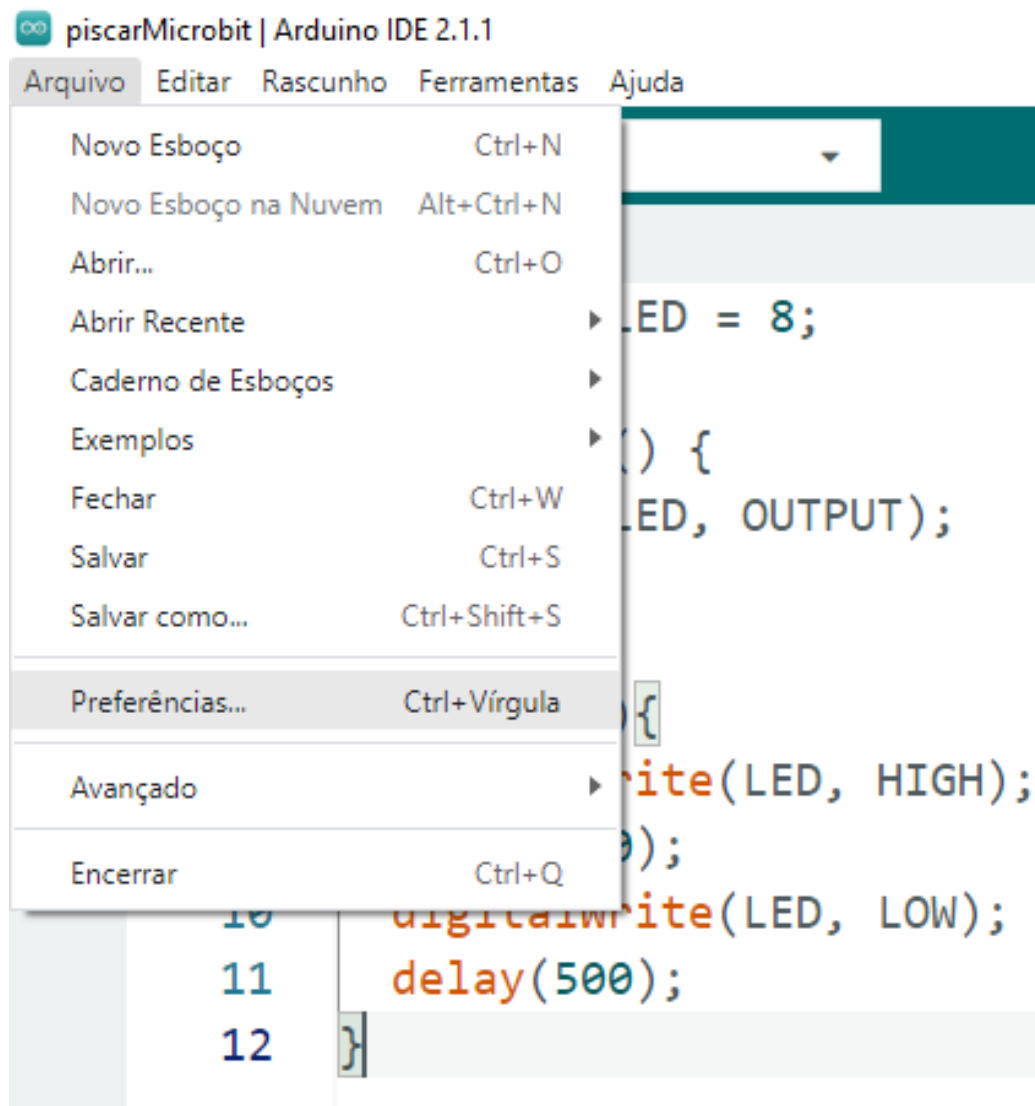


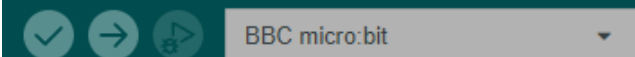
```
1|Ludobot / Itty Bitty Buggy: Microduino
2|https://raw.githubusercontent.com/Microduino/microduino_boards/master/
  package_microduino.cc_index.json
3|ou
4|https://github.com/wasdpkj/Microduino-IDE-Support/raw/master/
  package_Microduino_index.json
5|(Esta segunda é indicada no sítio Chinês do Microduino mas não está funcionando
  corretamente e resulta em: Erro durante o download de https://github.com/wasdpkj/
  Microduino-IDE-Support/raw/master/package_Microduino_index.json)
6|
7|
8|
9|Makey Makey: SparkFun AVR Boards
10|https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/
   package_sparkfun_index.json
11|
12|
13|
14|Microduino: Nordic Semiconductor nRF5 Boards
15|https://sandeepmistry.github.io/arduino-nRF5/package_nRF5_boards_index.json
```

Texto sem formatação ▾ Largura da tabulação: 8 ▾ Lin 1, Col 1 INS

Arquivo / Preferências

(a tela pode mudar em função da versão)





```
piscarMicrobit.ino
1  const int LED = 1;
2
3  void setup()
4  {
5      pinMode(LED, OUTPUT);
6  }
7
8  void loop()
9  {
10     digitalWrite(LED, HIGH);
11     delay(500);
12     digitalWrite(LED, LOW);
13     delay(500);
14 }
```

Saída

Preferências



Configurações

Rede

Localização do Caderno de Esboços:

c:\Users\DECIGI\Documents\Arduino

PROCURAR

☐ Mostrar arquivos dentro dos Esboços

Tamanho da fonte do editor: 20

Dimensões da interface: ☒ Automático 100 %

Tema de Cores: Light

Idioma do Editor: português (Brasil) (Recarregamento Necessário)

Mostrar a saída detalhada durante ☐ compilar ☐ enviar

Avisos do compilador Nenhum

☐ Verificar o código após enviar☒ Salvamento Automático☐ Dicas do editorURLs do Gerenciador de Placas Adicionais: http://drazzy.com/package_drazzy.com_index.json, https://sandeepmishra.in/package_index.json

CANCELAR

OK

Aqui serão informadas as placas, uma por linha

URLs do Gerenciador de Placas Adicionais

✕

Insira URLs adicionais, uma por linha

http://drazzy.com/package_drazzy.com_index.json

https://sandeepmistry.github.io/arduino-nRF5/package_nRF5_boards_index.json

Clique para obter uma lista de URLs de placas não oficiais suportadas

CANCELAR

OK

Faça uma de cada vez, pois há um passo adicional

Instalar Ludobot / Itty Bitty Buggy / Microduino / mCookie / mPie

[https://raw.githubusercontent.com/
Microduino/microduino_boards/
master/
package_microduino.cc_index.json](https://raw.githubusercontent.com/Microduino/microduino_boards/master/package_microduino.cc_index.json)

Abzir

placasExtrasArduino.txt

~/Documentos/2024/ask2024/extensao/5ME/5ME2024/25outubro

Salvar

```
1 Ludobot / Itty Bitty Buggy: Microduino
2 https://raw.githubusercontent.com/Microduino/microduino_boards/master/
  package_microduino.cc_index.json
3 ou
4 https://github.com/wasdpkj/Microduino-IDE-Support/raw/master/package_Microduino_index.json
5 (Esta segunda é indicada no sítio Chinês do Microduino mas não está funcionando corretamente e
  resulta em: Erro durante o download de https://github.com/wasdpkj/Microduino-IDE-Support/raw/
  master/package_Microduino_index.json)
6
7
8
9 Makey Makey: SparkFun AVR Boards
10 https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/
  package_sparkfun_index.json
11
12
13
14 Microduino: Nordic Semiconductor nRF5 Boards
15 https://sandeepmistry.github.io/arduino-nRF5/package_nRF5_boards_index.json
```

“CTRL + C”

“CTRL + V”

URLs do Gerenciador de Placas Adicionais

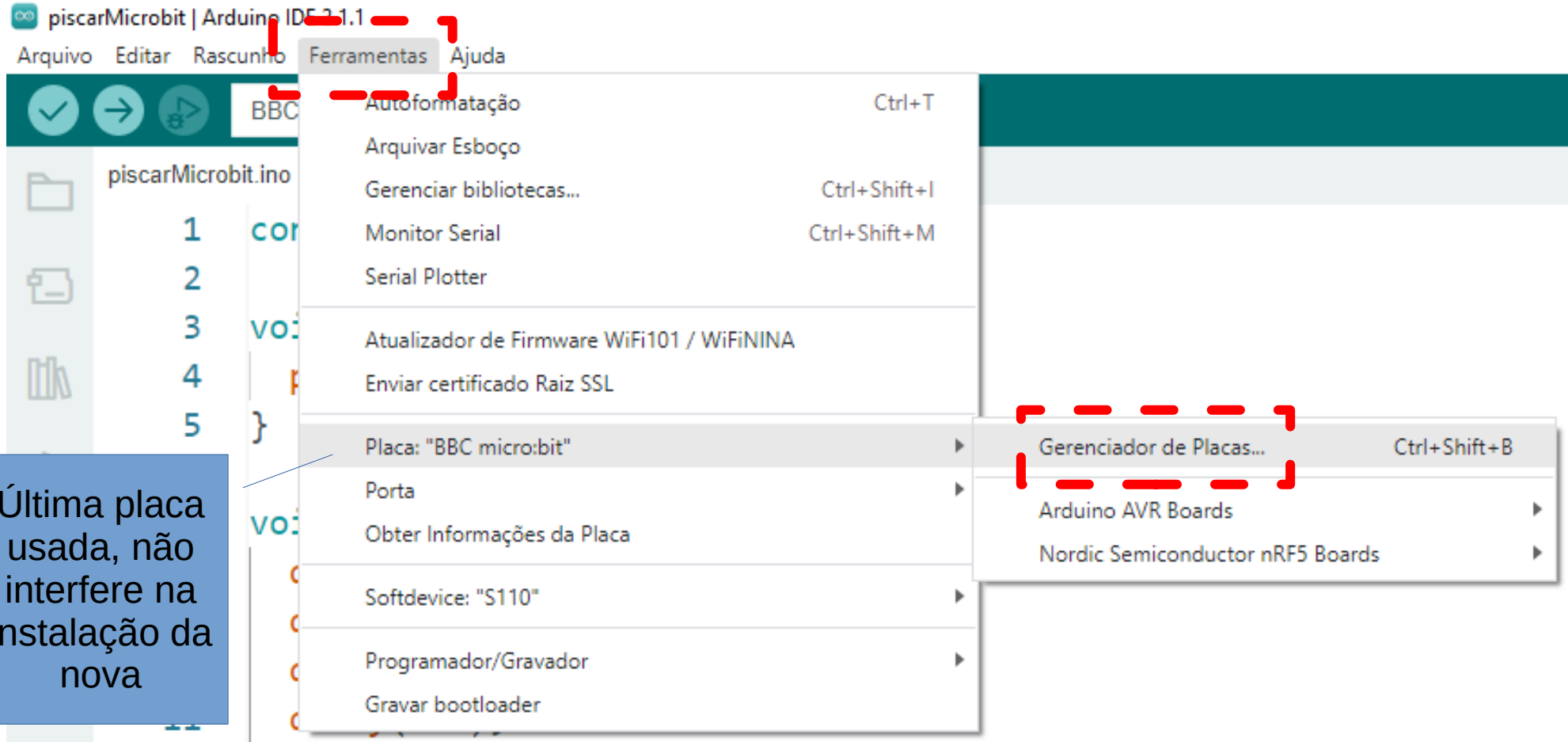
Insira URLs adicionais, uma por linha

```
http://drazzy.com/package_drazzy.com_index.json
https://sandeepmistry.github.io/arduino-nRF5/package_nRF5_boards_index.json
https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_ind
|
```

Clique para obter uma lista de URLs de placas não oficiais suportadas

CANCELAR OK

Instalar os arquivos das placas





Última placa usada – não é referente à sua instalação atual

- Digite 'Microduino'
- ...aguarde a lista...
- Selecione *Microduino AVR Boards*
- Clique instalar

Saída



Downloading packages

Microduino:avr@1.0.1

Installing platform Microduino:avr@1.0.1

Configuring platform.

Platform Microduino:avr@1.0.1 installed

Em processamento Microduino AVR Boards:1.0.1: Platform Microduino:avr@1.0.1 installed

Selecione a placa

Ferramentas Ajuda

- Autoformatação Ctrl+T
- Arquivar Esboço
- Gerenciar bibliotecas... Ctrl+Shift+I
- Monitor Serial Ctrl+Shift+M
- Serial Plotter
- Atualizador de Firmware WiFi101 / WiFiNINA
- Enviar certificado Raiz SSL

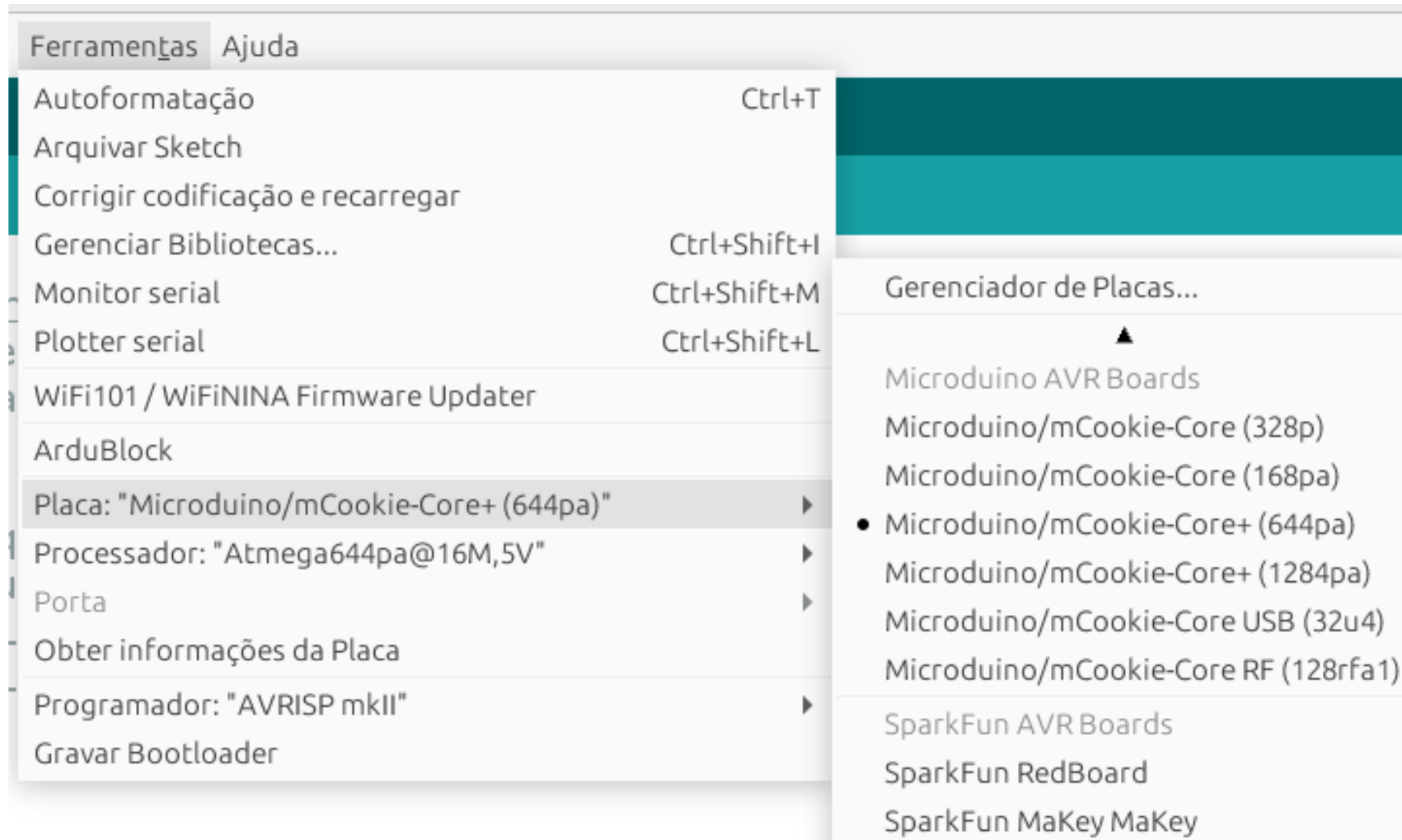
Placa: "SparkFun MaKey MaKey"

- Gerenciador de Placas... Ctrl+Shift+B
- Microduino AVR Boards
 - Microduino/mCookie-Core (328p)
 - Microduino/mCookie-Core (160pa)
 - Microduino/mCookie-Core+ (644pa)
 - Microduino/mCookie-Core+ (1284pa)
 - Microduino/mCookie-Core USB (32u4)
 - Microduino/mCookie-Core RF (128rfa1)

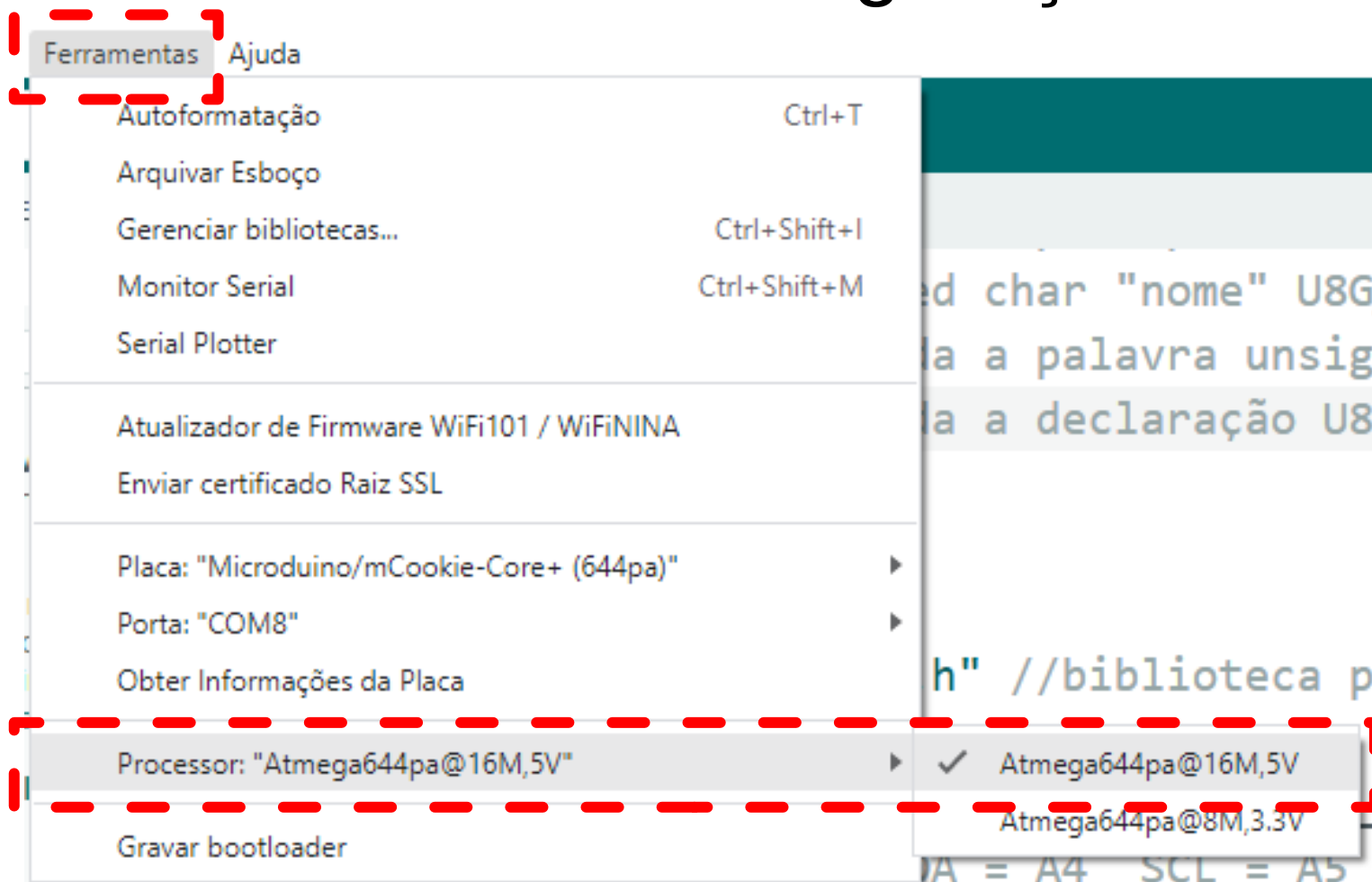
Última utilizada

```
ed char "nome" U8G_PROGMEM = {  
la a palavra unsigend  
la a declaração U8G_PROGMEM
```

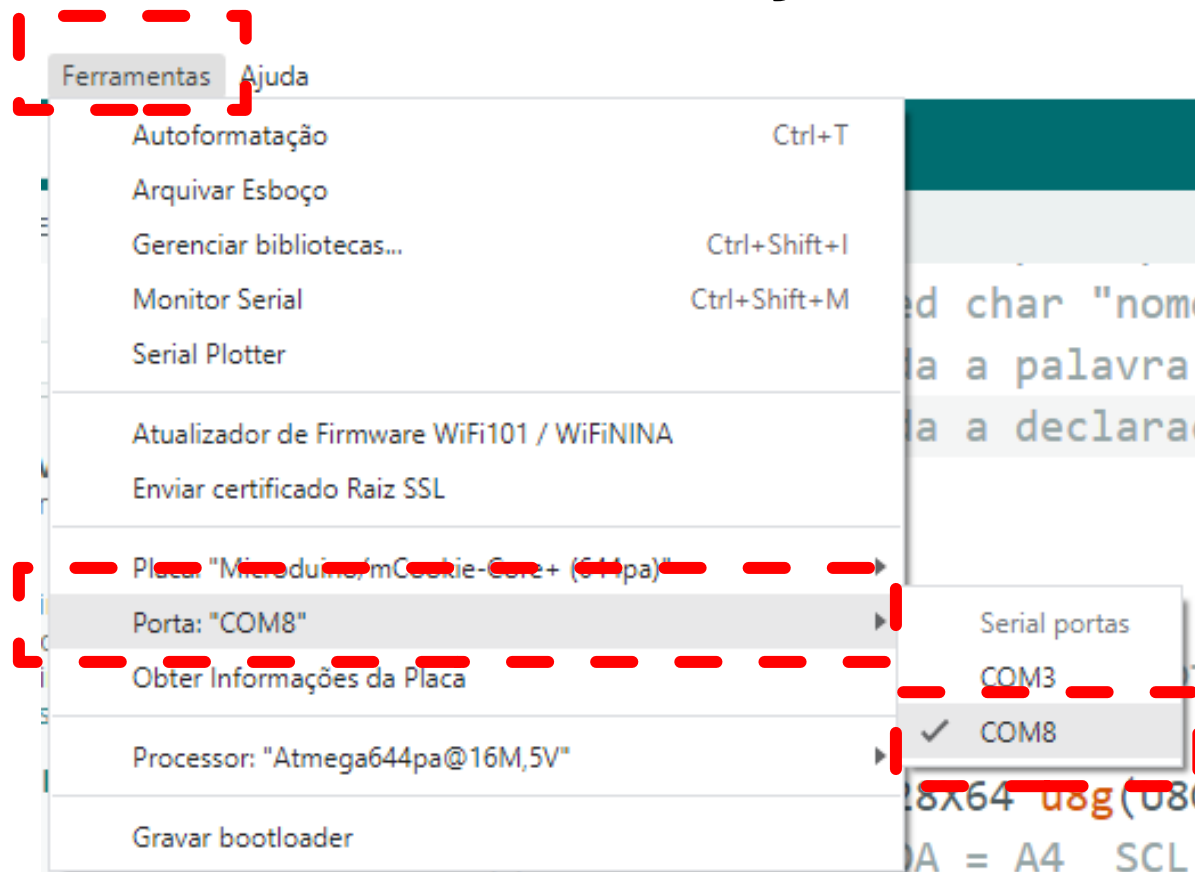
Platform Microduino:avr@1.0.1 installed



Acerte a configuração



Conecte o Ludobot e acerte a porta de comunicação



Não sabe qual porta usar ?

No Windows, pode usar o gerenciador de dispositivos



Portas (COM e LPT)

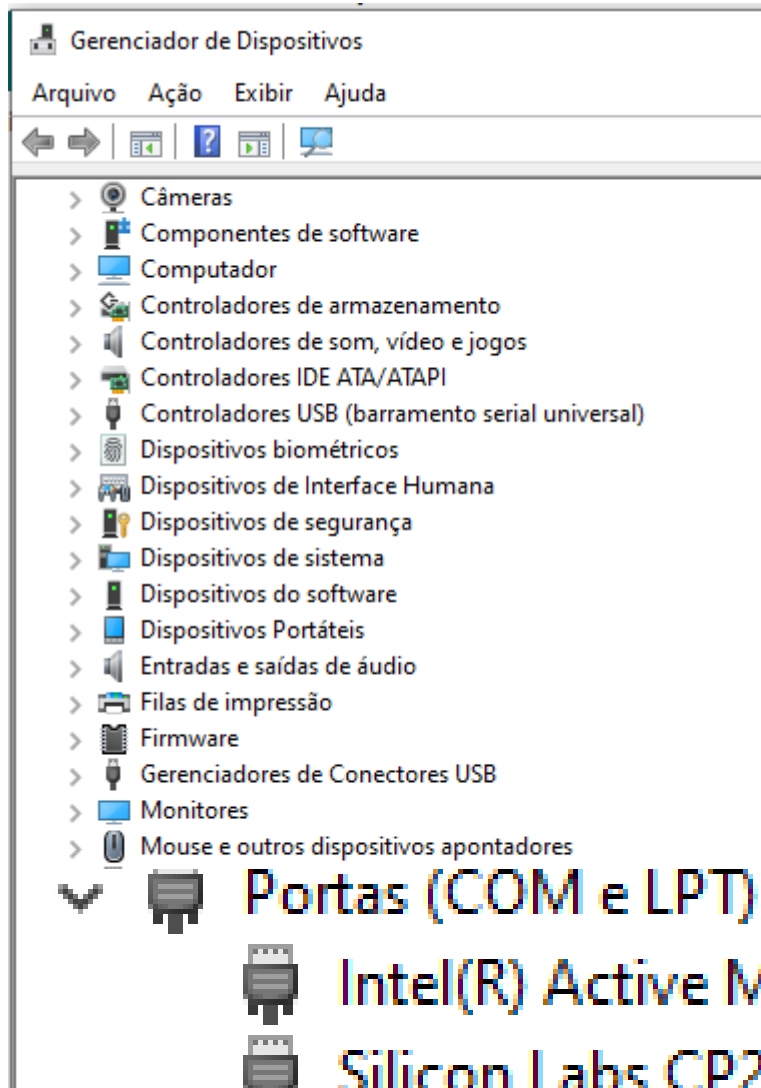


Intel(R) Active Management Technology - SOL (COM3)



Silicon Labs CP210x USB to UART Bridge (COM8)





Portas (COM e LPT)



Intel(R) Active Management Technology - SOL (COM3)

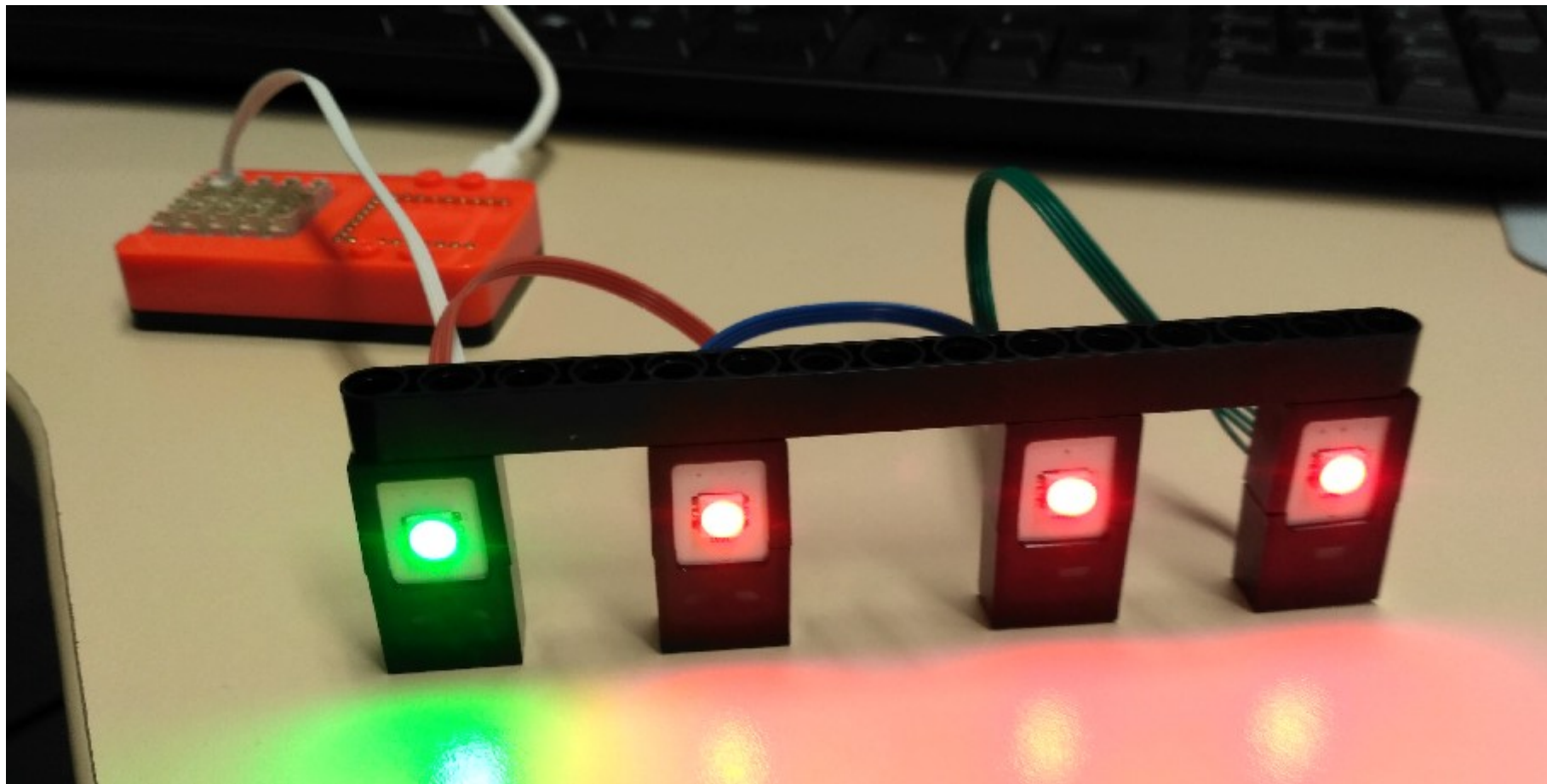


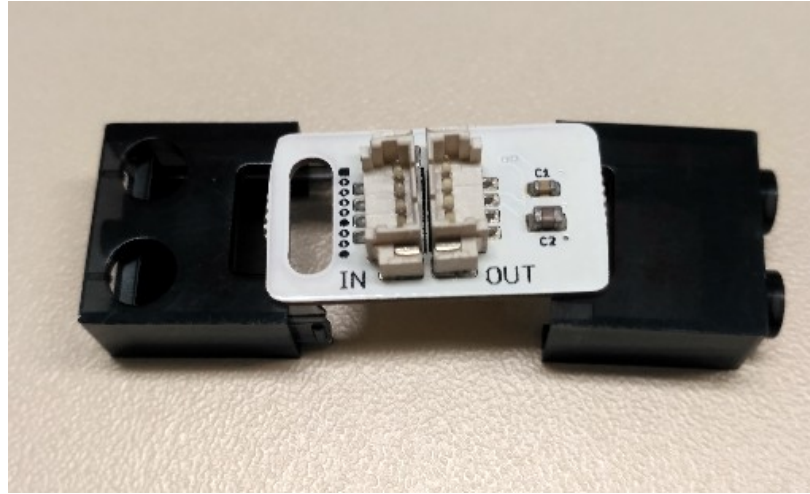
Silicon Labs CP210x USB to UART Bridge (COM8)

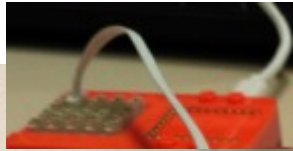


Agora, é só usar...

Exemplo 1 – Fita de LEDs

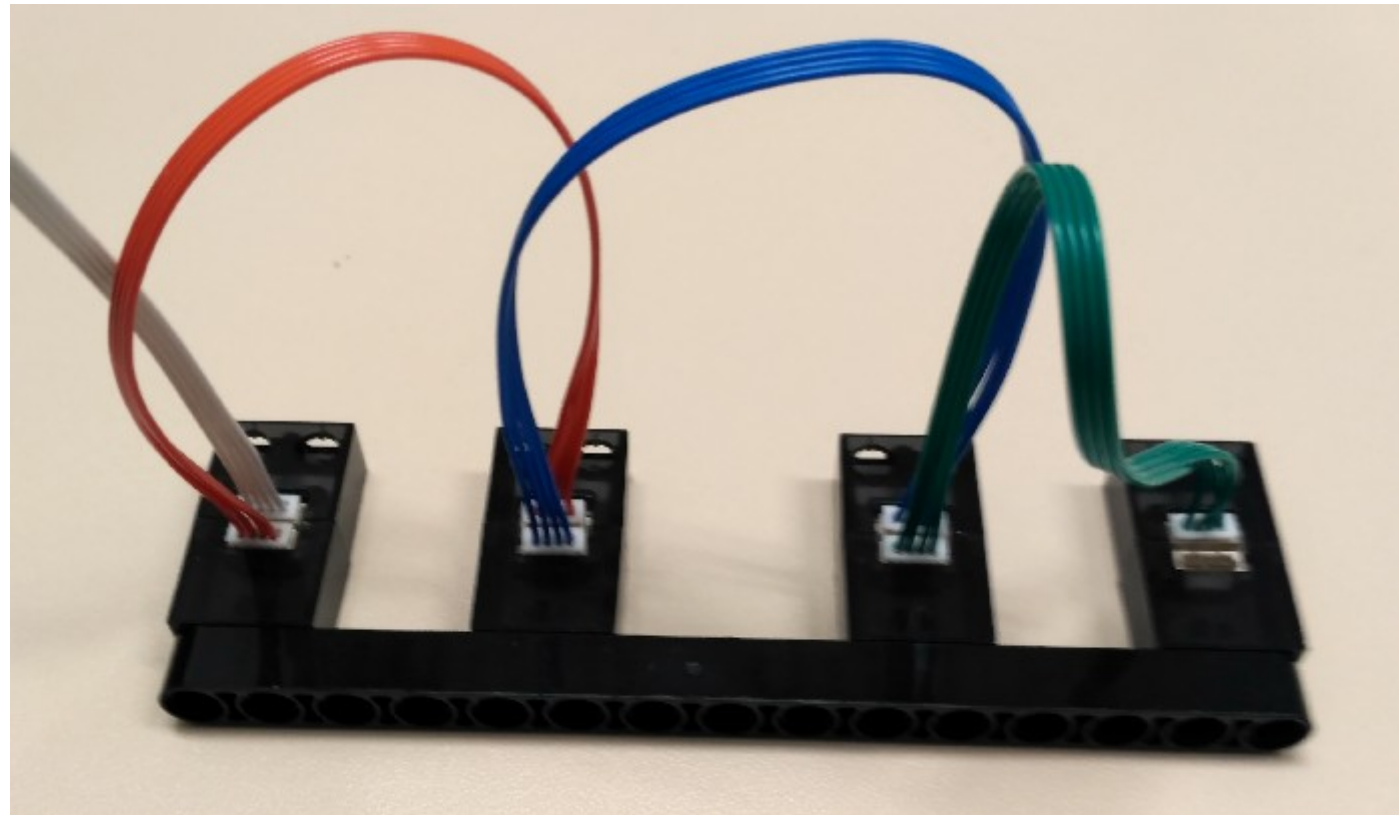






Conector '*IN*' do primeiro LED ligado no Ludobot (usei a porta D8/D9, mas pode ser outra é só ajustar o código)

Conector '*OUT*' do primeiro LED ligado no '*IN*' do segundo LED; '*OUT*' do segundo LED no '*IN*' do terceiro LED; ...





Eu escolhi ligar o conector 'I/N' do primeiro LED na porta D8/D9, mas pode ser outra é só ajustar o código.

Observação -na figura ao lado, no conector, temos:

D9 - porta digital 9

D8 - porta digital 8

VCC (+5V)

GND (0V)

Arquivo com o código



Faixa de LEDs

faixaDeLEDS

```
1 #include <Adafruit_NeoPixel.h> //biblioteca da Adafruit
2
3 #define PinoControle 8 //controle do primeiro LED
4 //0 conector pega o primeiro contato somente; por exemplo,
5 //conectado em D8/D9 com o pino 9 não funciona, só com o 8
6 #define QtdadeLEDs 4 //número de LEDs utilizados
7 #define valorMaximo 255 //maior valor de cor
8 #define valorMinimo 0 //menor valor de cor
9
10 //cria um objeto com o nome faixaLEDs|
11 Adafruit_NeoPixel faixaLEDs = Adafruit_NeoPixel(QtdadeLEDs, PinoControle, NEO_GRB + NEO_KHZ800);
12
13 void setup() {
14     faixaLEDs.begin(); //inicializa o objeto faixaLEDs
15     faixaLEDs.show(); //inicia a exibição de cores em faixaLEDs
16 }
```

```
18 void loop() {
19   // 'mudaCor' tem dois parâmetros: cor desejada, tempo
20   // 'cor' (RGB) para o primeiro LED; depois do 'tempo' muda a cor para o segundo LED, ...
21   mudaCor(faixaLEDs.Color(255, 0, 0), 500); //Vermelho -> R=255, G=0, B=0
22   mudaCor(faixaLEDs.Color(0, 255, 0), 500); //Verde
23   mudaCor(faixaLEDs.Color(0, 0, 255), 500); //Azul
24   mudaCor(faixaLEDs.Color(255, 255, 0), 500); //Amarelo
25   mudaCor(faixaLEDs.Color(255, 255, 255), 500); //Branco
26   devanescer(255, 0, 0, 10); //varia luminosidade em vermelho
27   devanescer( 255, 255, 0, 10); //varia luminosidade em amarelo
28   devanescer( 0, 255, 0, 10); //varia luminosidade em verde
29   devanescer( 0, 0, 255, 10); //varia luminosidade em azul
30 }
```

```
32 void mudaCor(uint32_t cor, uint8_t tempo) {
33     for (uint16_t qualLED = 0; qualLED < faixaLEDs.numPixels(); qualLED++) {
34         faixaLEDs.setPixelColor(qualLED, cor);
35         faixaLEDs.show();
36         delay(tempo);
37     }
38 }
39
40 void ajustaLED(uint32_t cor) {
41     for (uint16_t quem = 0; quem < faixaLEDs.numPixels(); quem++) {
42         faixaLEDs.setPixelColor(quem, cor);
43     }
44     faixaLEDs.show();
45 }
```

```
47 void devanescer( int vermelho, int verde, int azul, uint8_t tempo) {
48     for (int valor = 0; valor < valorMaximo; valor++)
49     {
50         ajustaLED(faixaLEDs.Color(map(valor, valorMinimo, valorMaximo, 0, vermelho),
51                                     map(valor, valorMinimo, valorMaximo, 0, verde),
52                                     map(valor, valorMinimo, valorMaximo, 0, azul))));
53         delay(tempo);
54     }
55     for (int valor = valorMaximo; valor >= 0; valor--)
56     {
57         ajustaLED(faixaLEDs.Color(map(valor, valorMinimo, valorMaximo, 0, vermelho),
58                                     map(valor, valorMinimo, valorMaximo, 0, verde),
59                                     map(valor, valorMinimo, valorMaximo, 0, azul))));
60         delay(tempo);
61     }
```

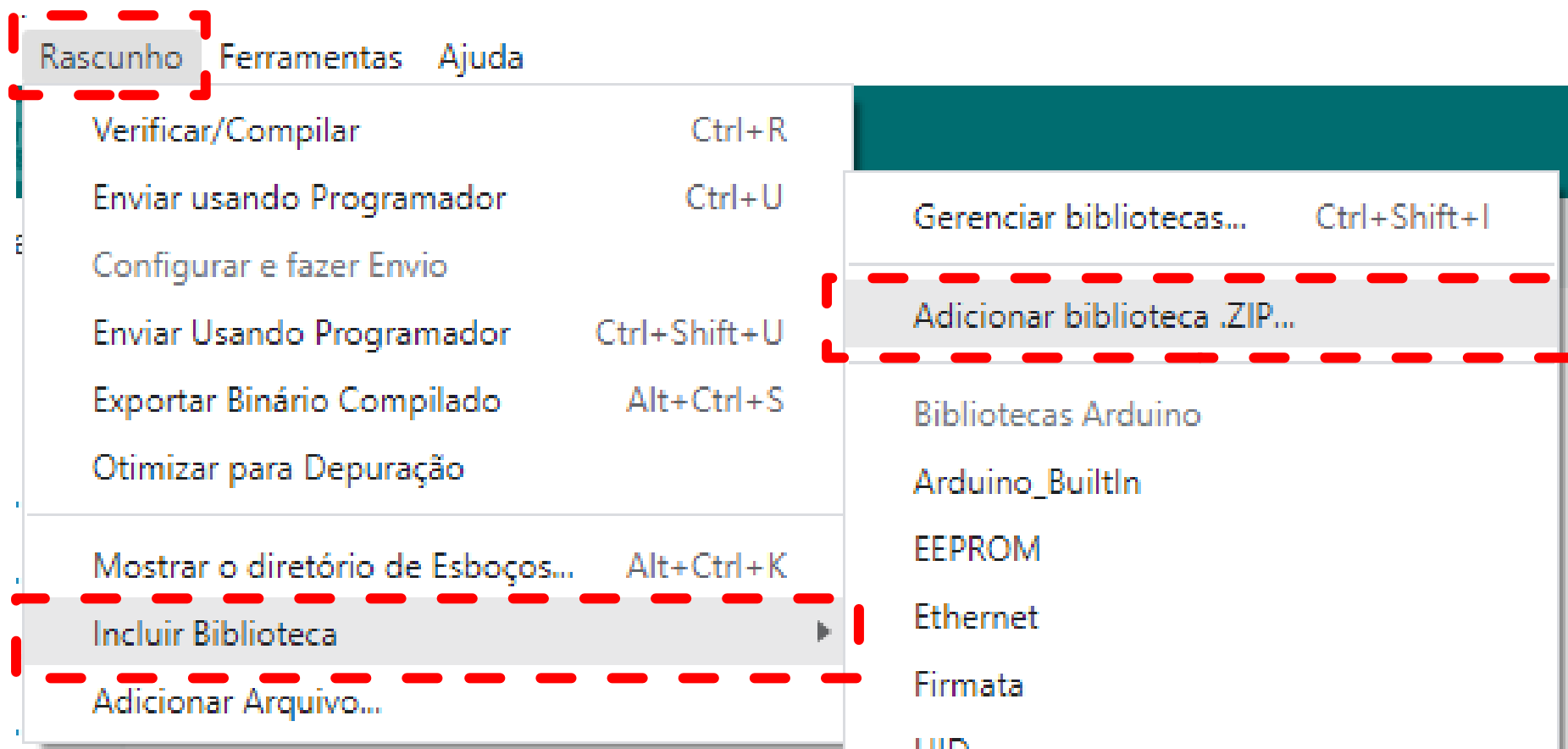
Compliar e testar

- Pode ser necessária a instalação da biblioteca , dependendo de se você já a utilizou ou não

```
<Adafruit_NeoPixel.h>
```

Se necessário, pode ser realizado com o arquivo disponível ou pela internet

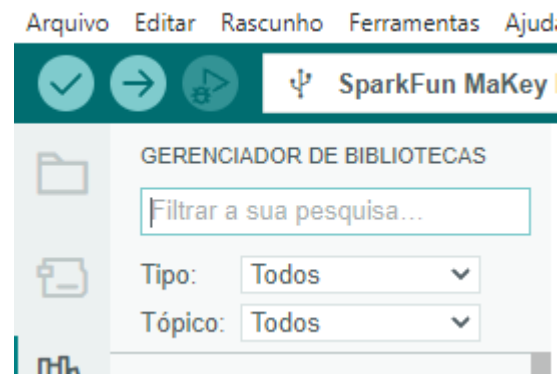
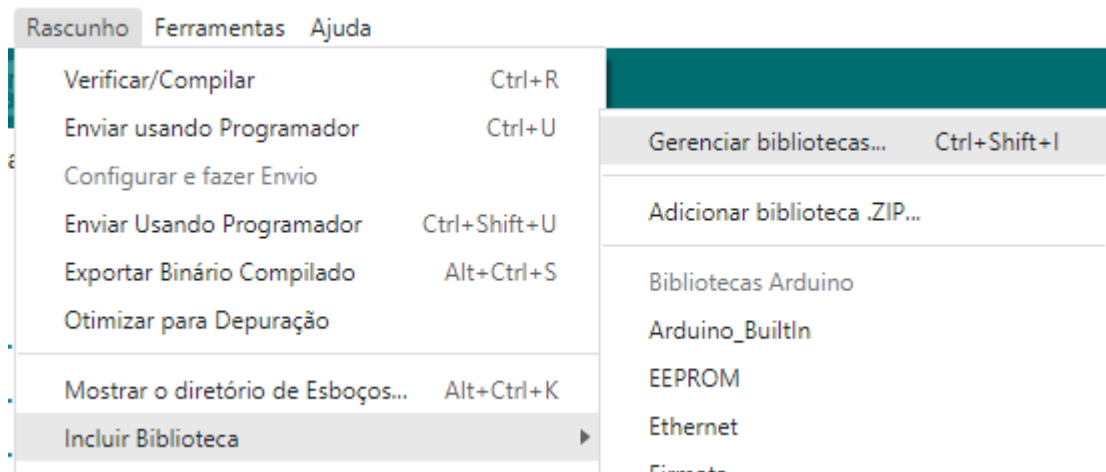
Se você tiver o arquivo



Se você não tiver o arquivo

Durante a compilação
o nome do arquivo
necessário é indicado

Adafruit_NeoPixel

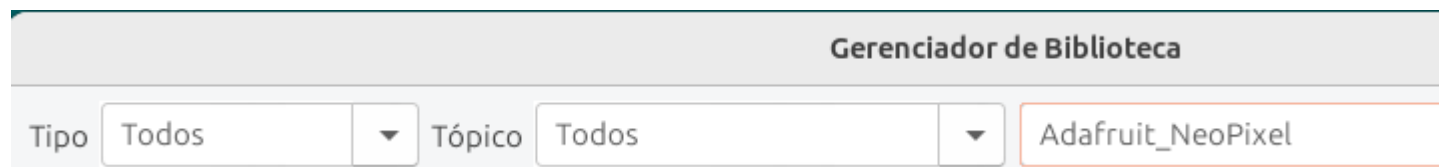


Se você não tiver o arquivo

Informe o
arquivo e
clique em
instalar



Interface
nova



Interface
antiga

Adafruit NeoPixel by **Adafruit** Versão **1.12.3** **INSTALLED**

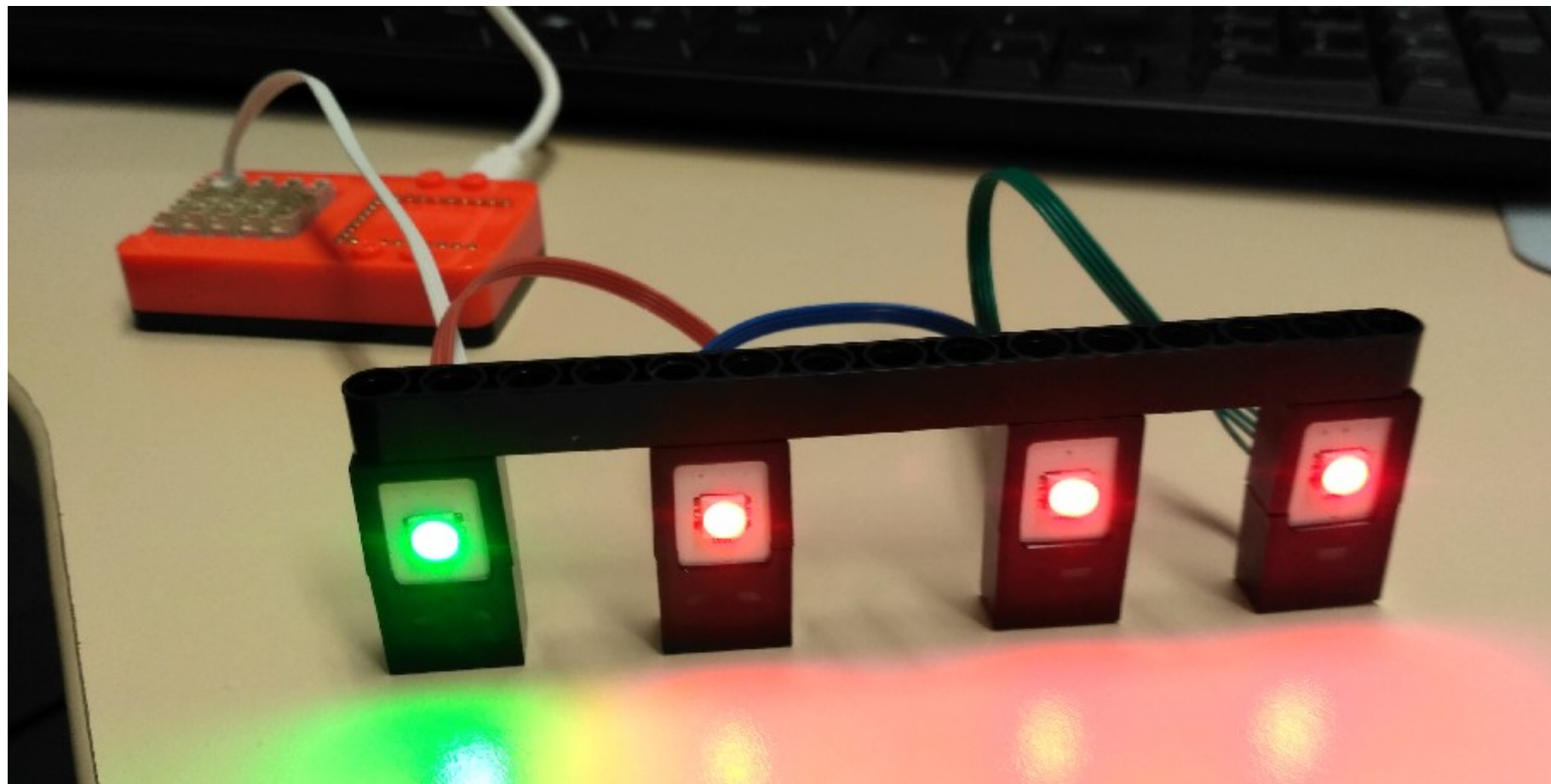
Arduino library for controlling single-wire-based LED pixels and strip. Arduino library for controlling single-wire-based LED pixels and strip.

[More info](#)

Selecionar vers...

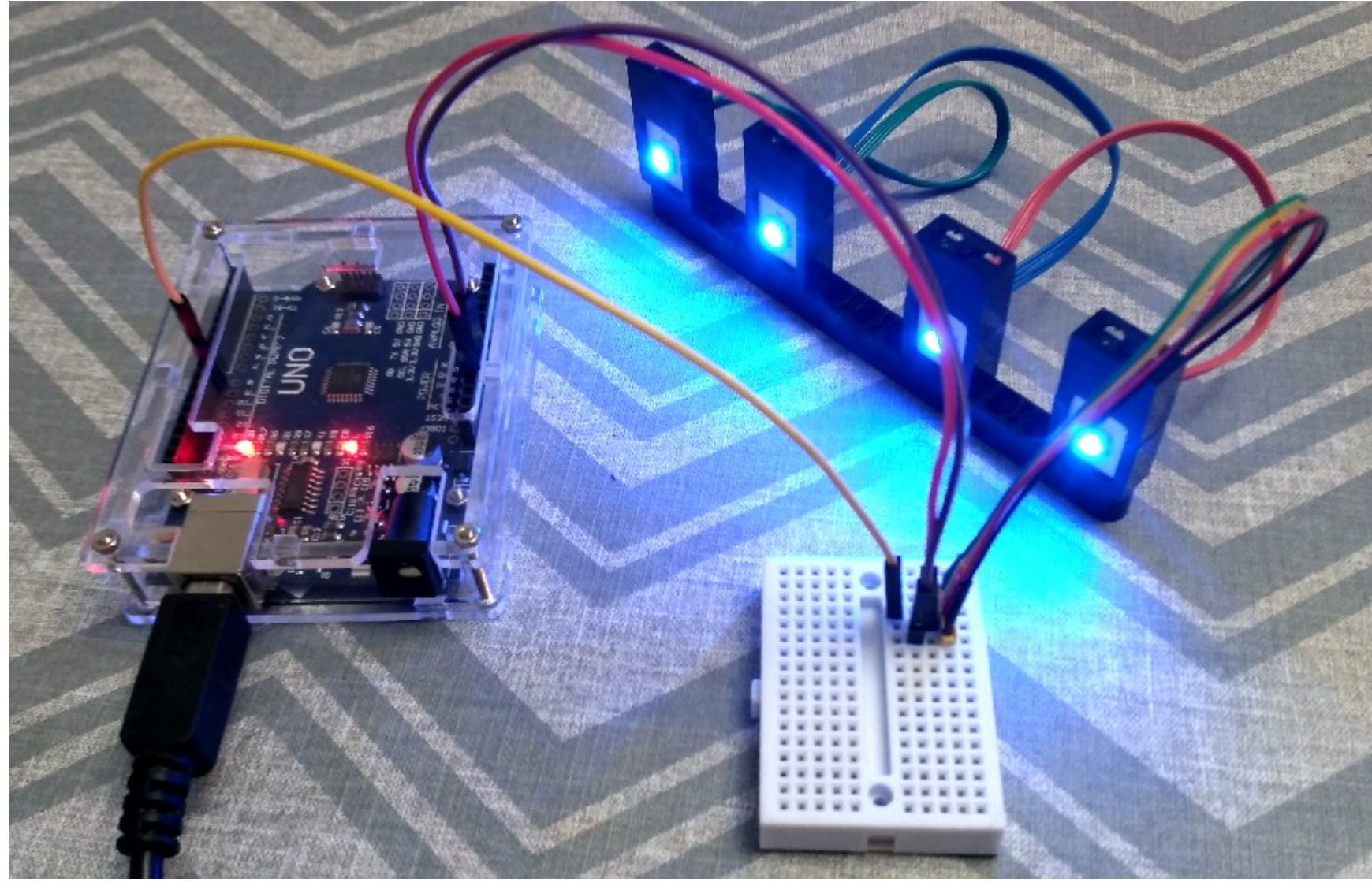
Instalar

Conecte os LEDs, conecte o Ludobot no computador, compile e teste



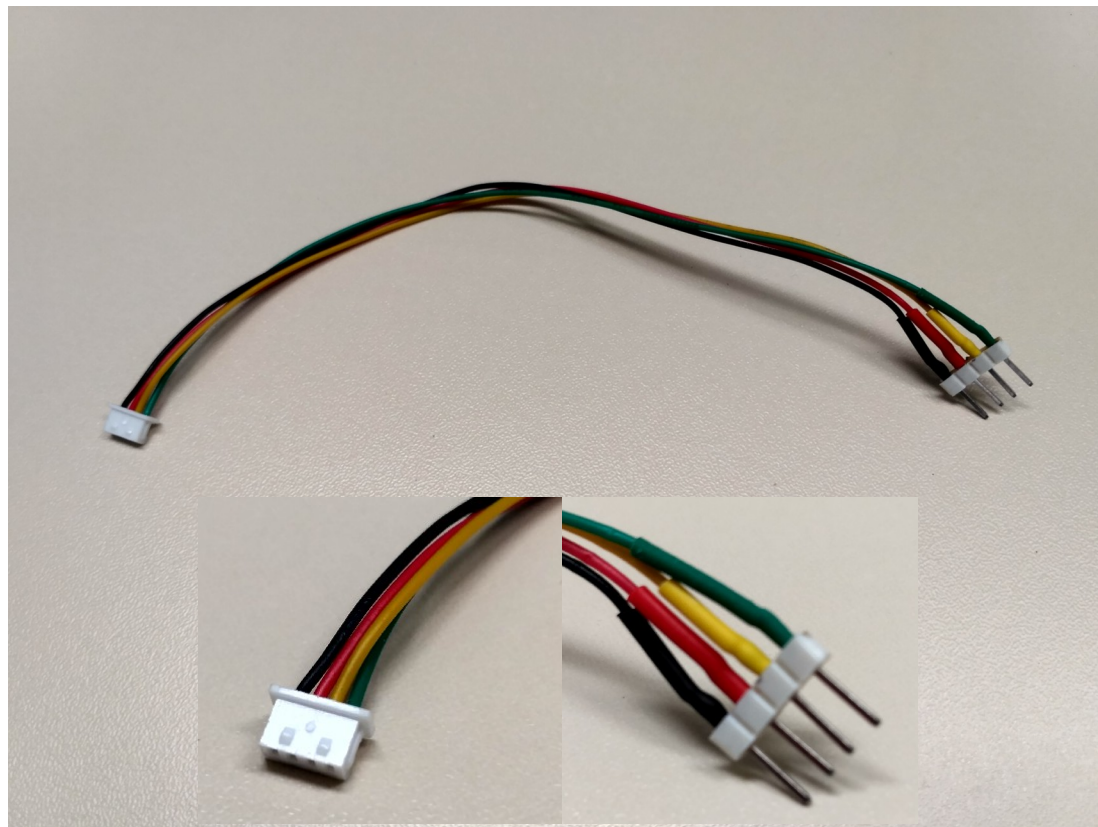
Depois que tudo estiver funcionando, se você quiser ligar no Arduino sem mudar nada do código, é só ligar o primeiro LED na porta '8' (e o VCC no VCC e o GND no GND).

Depois, compilar o mesmo código mudando a palca para Arduino UNO (ou outro)

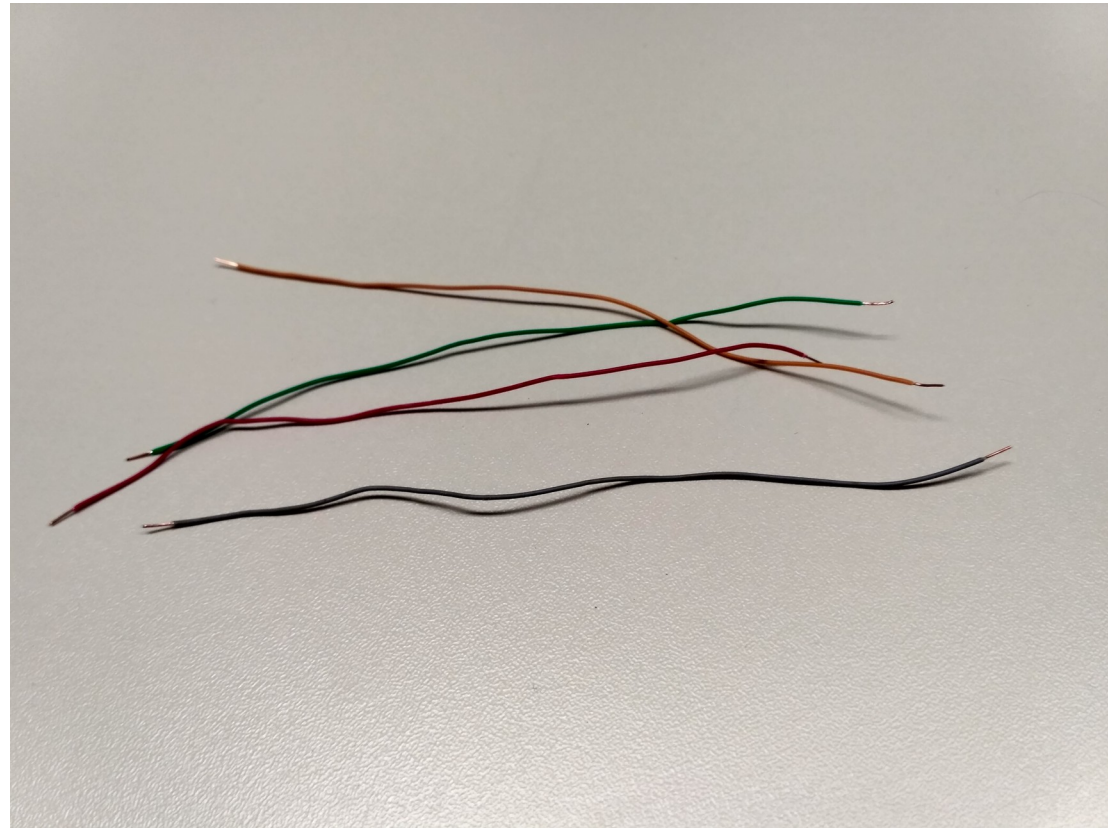


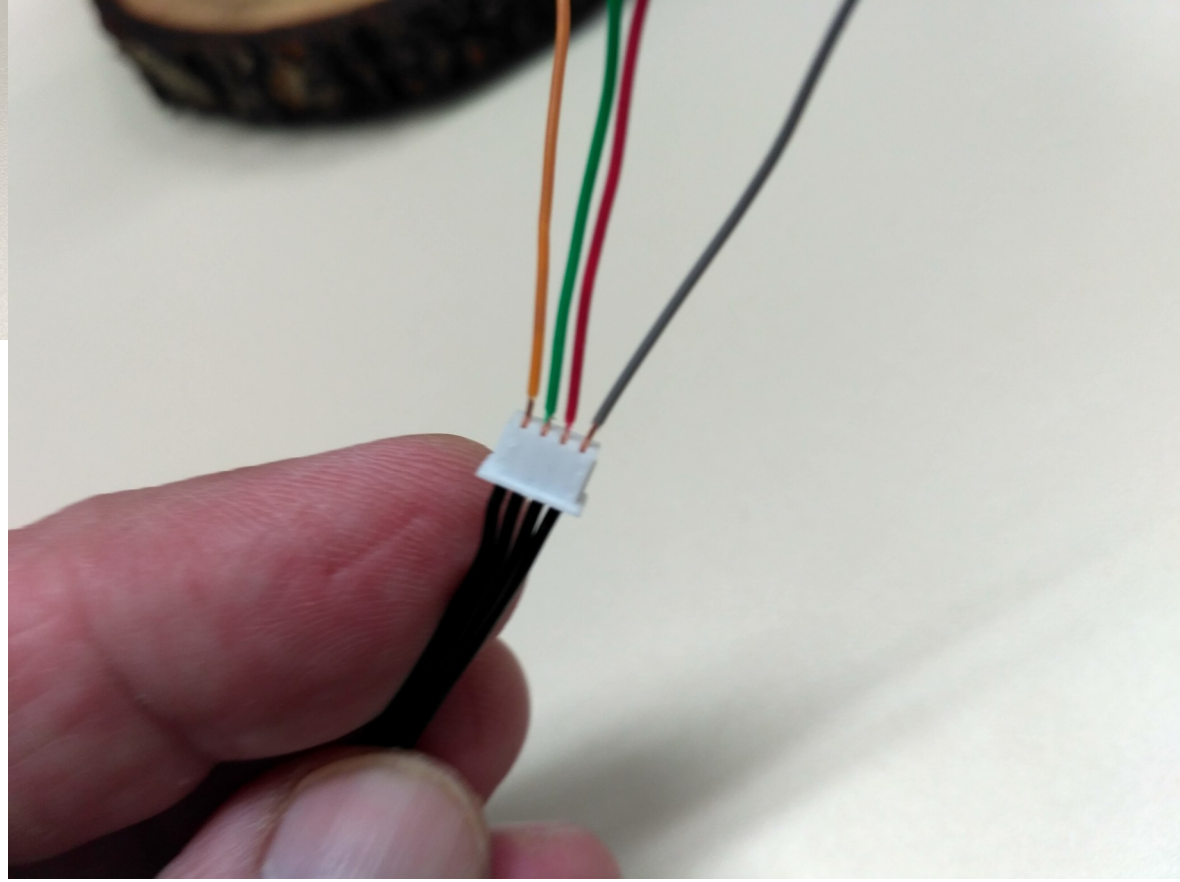
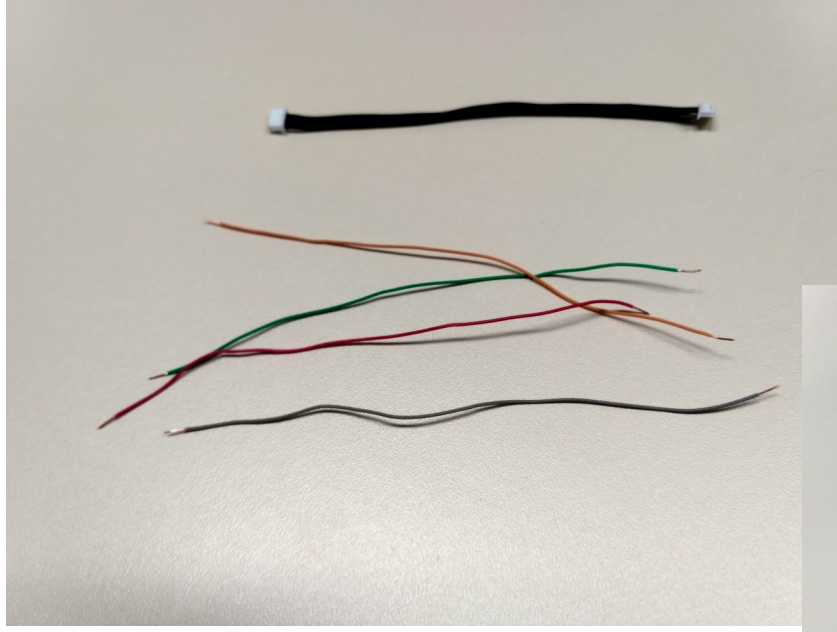


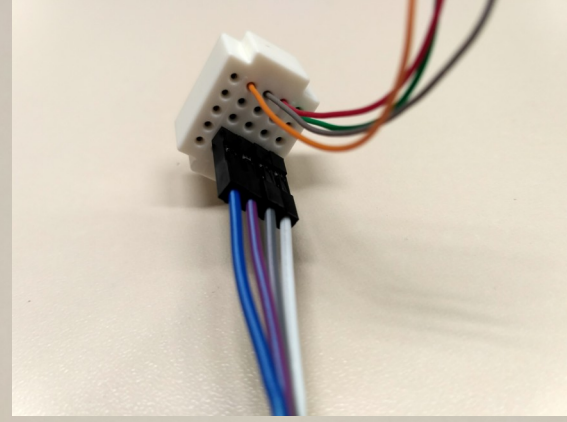
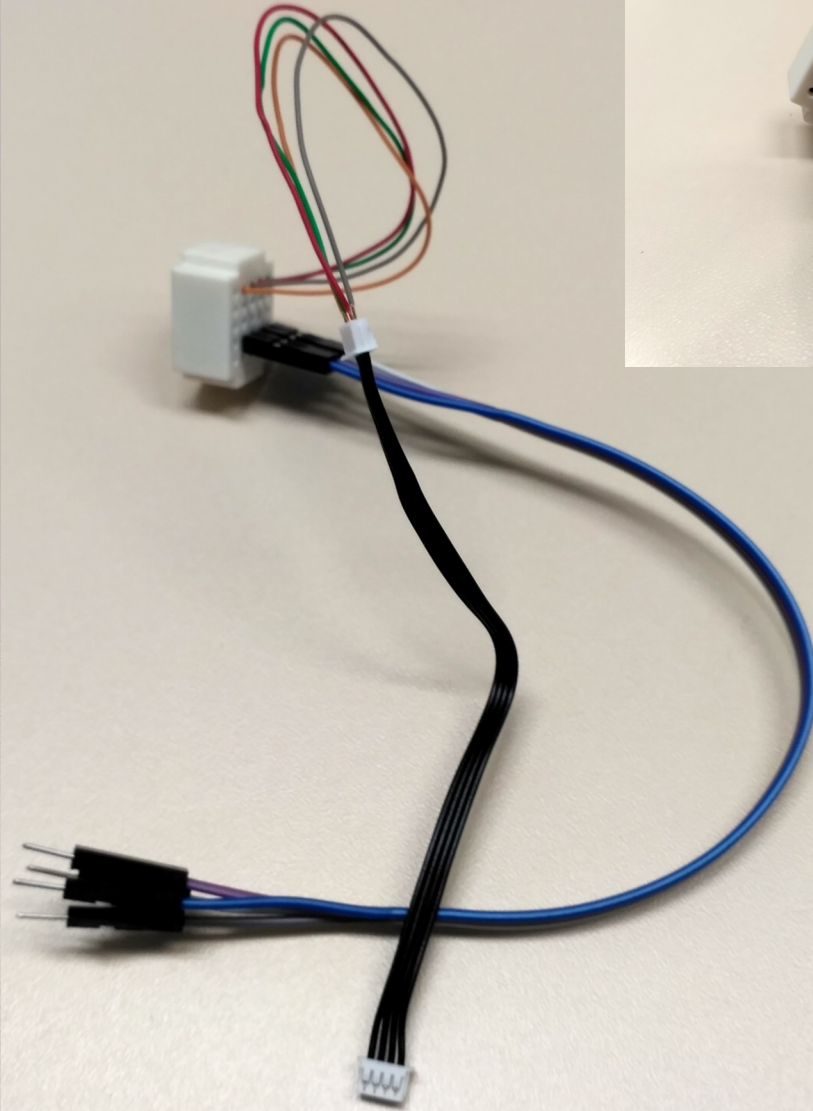
Tem que ter um cabo próprio ou fazer um.

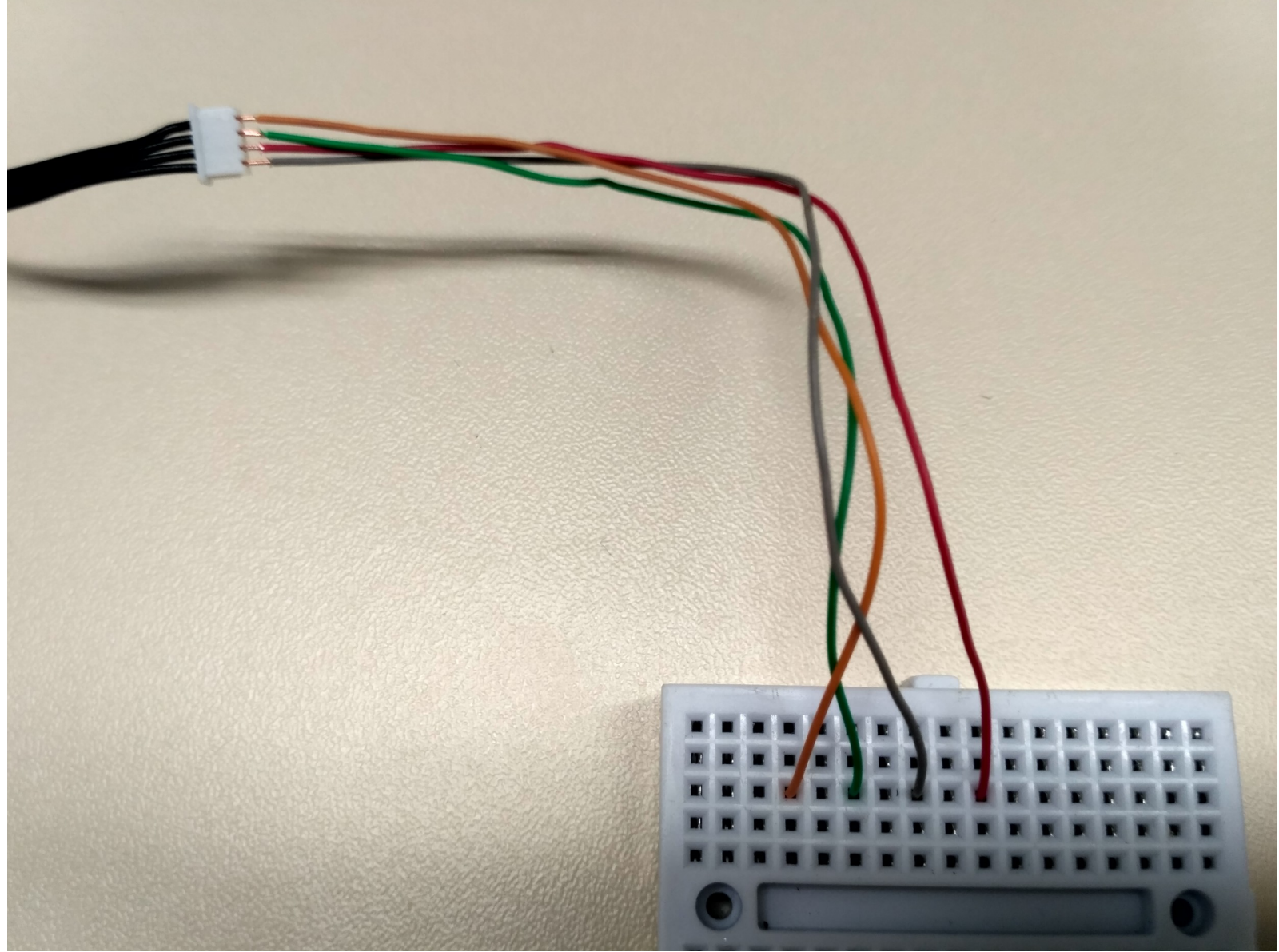


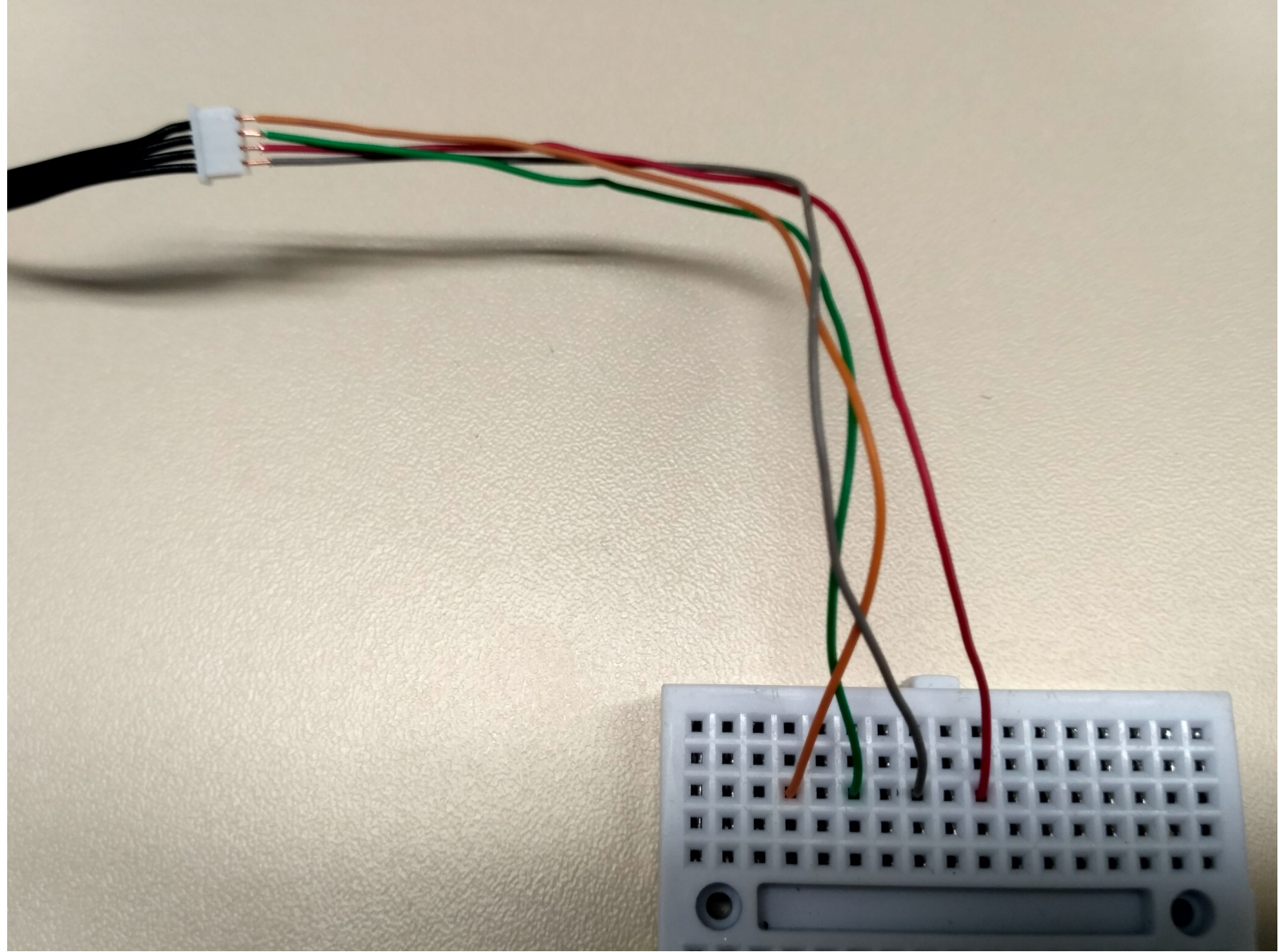
Dá para improvisar para testar:

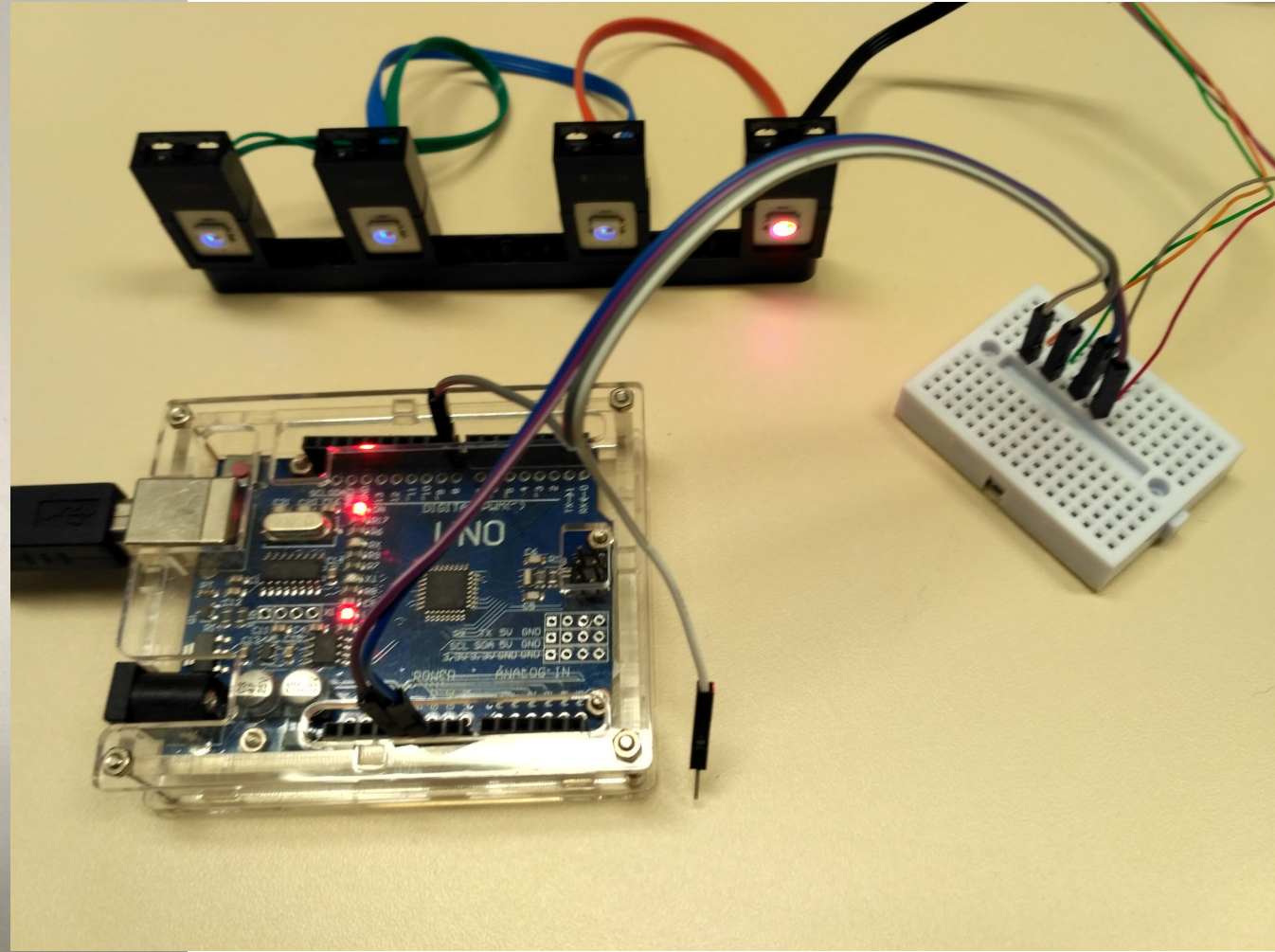
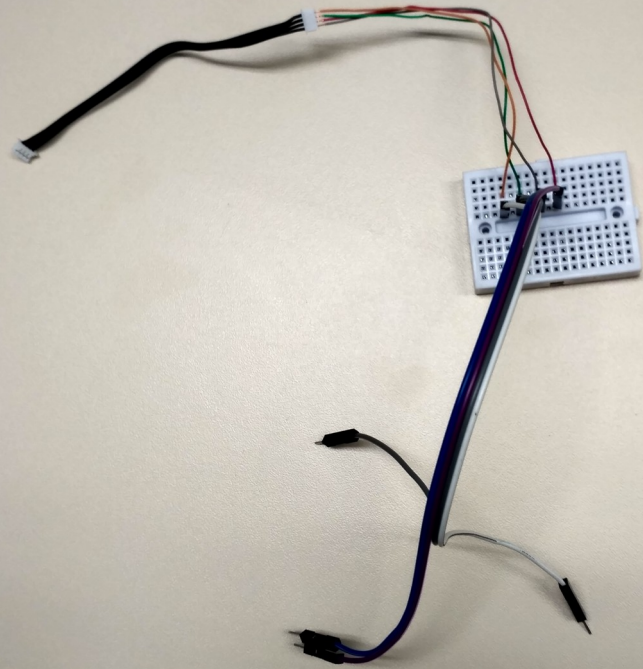




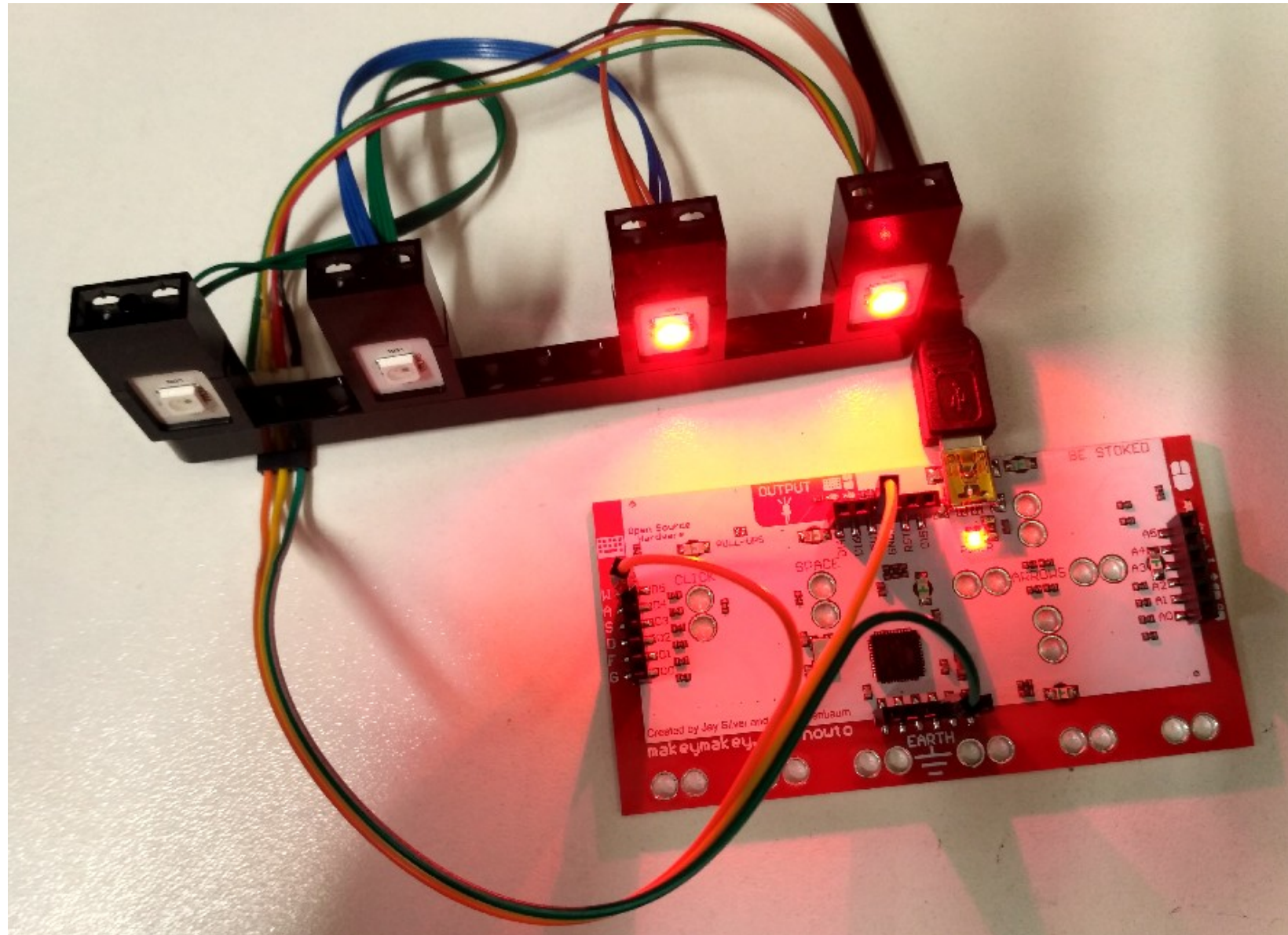




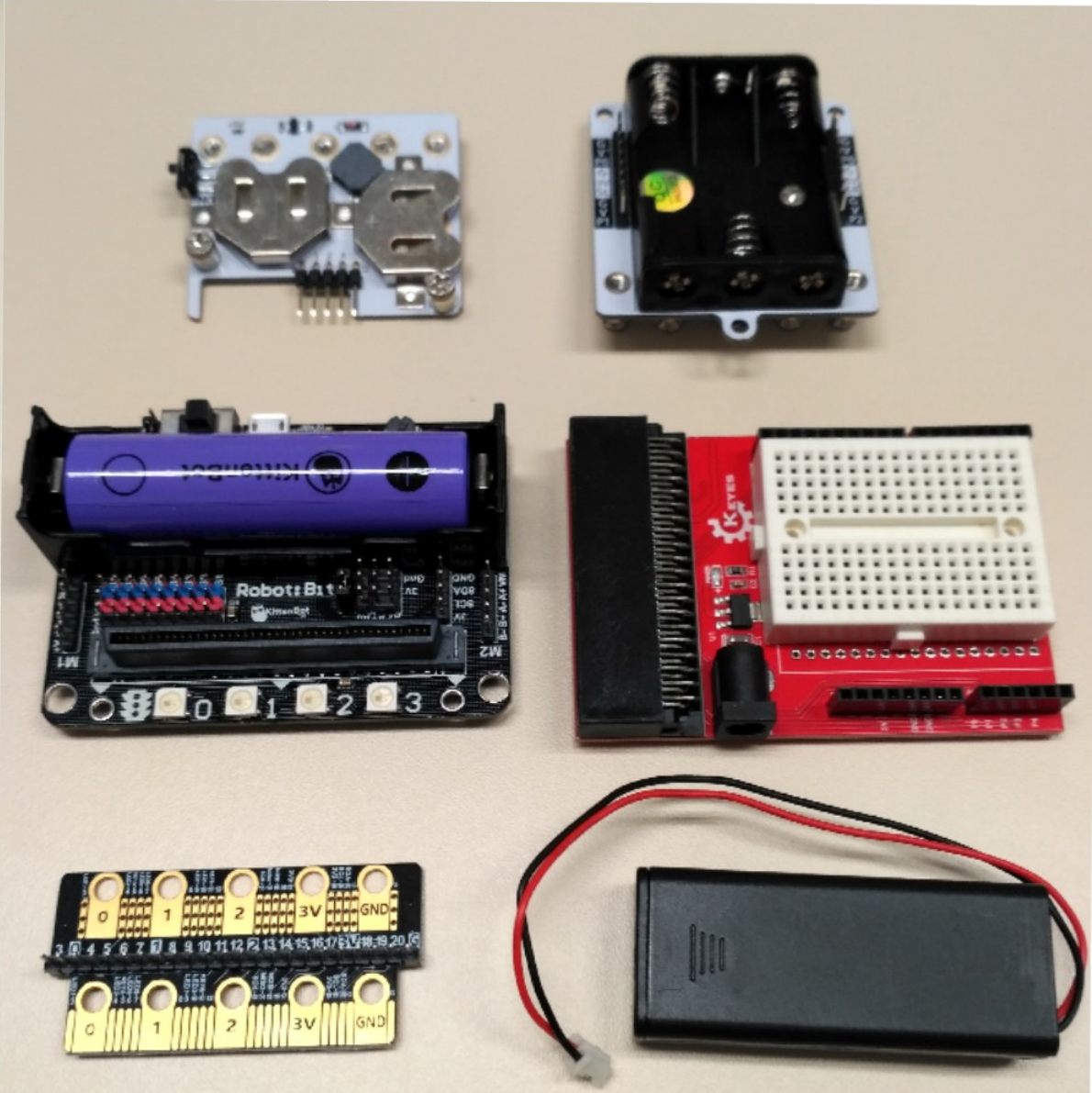


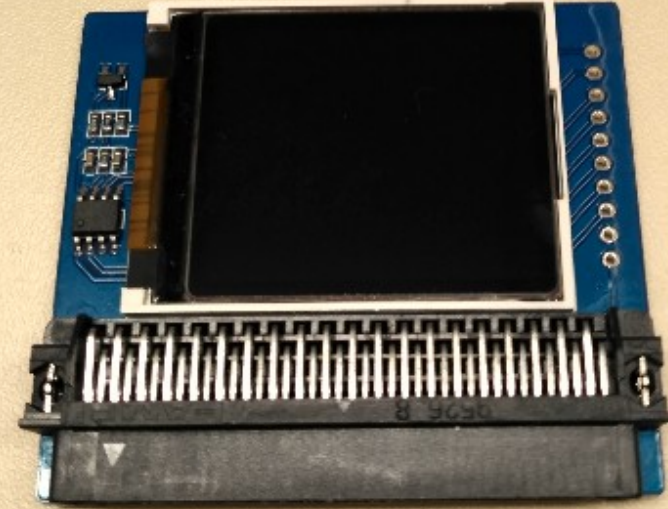


Makey Makey



Microbit



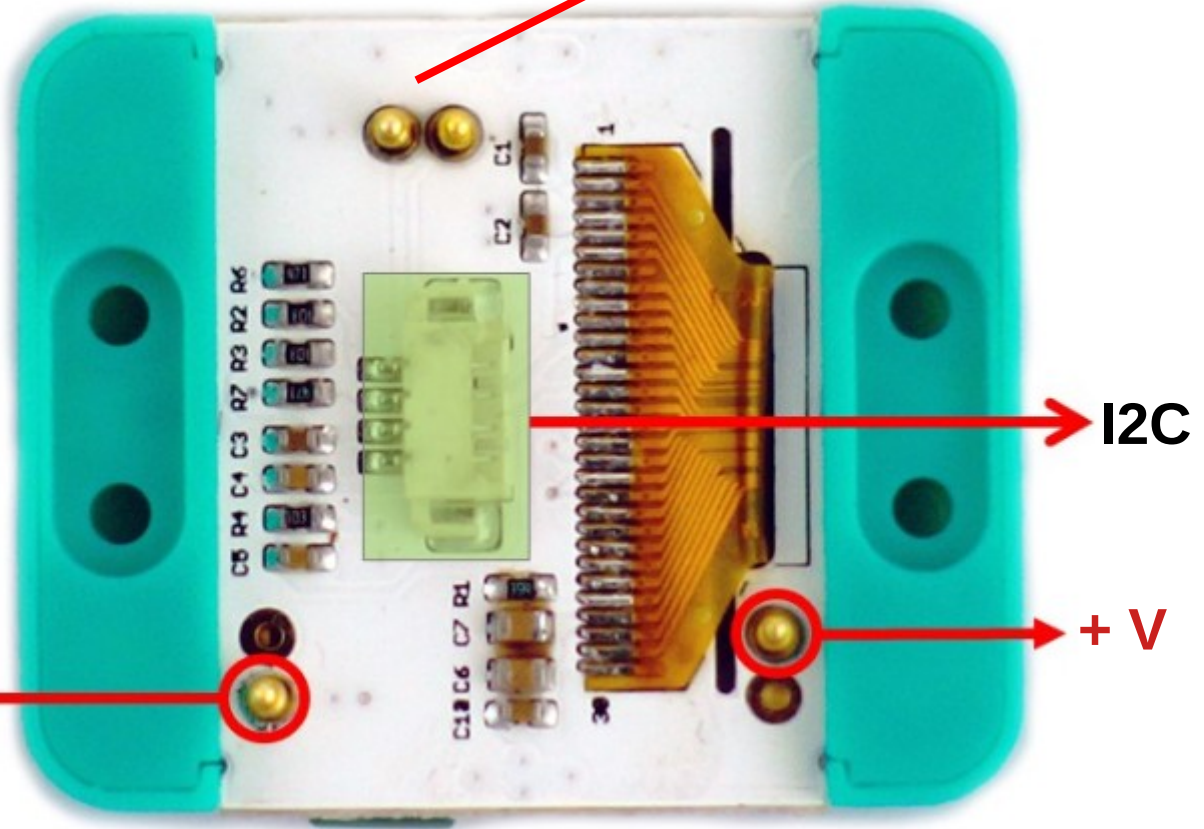


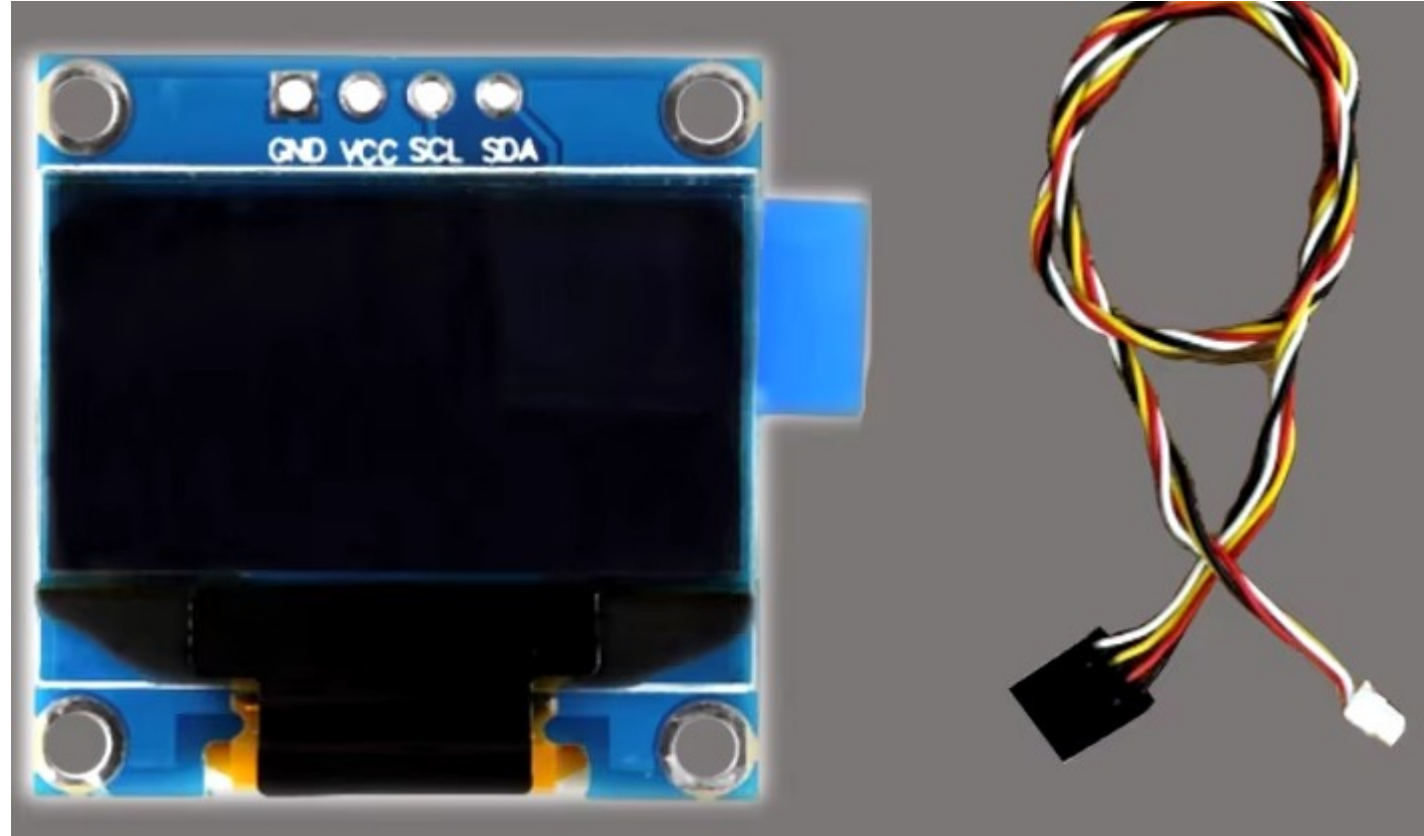


Exemplo 2

Sensor de temperatura
e umidade

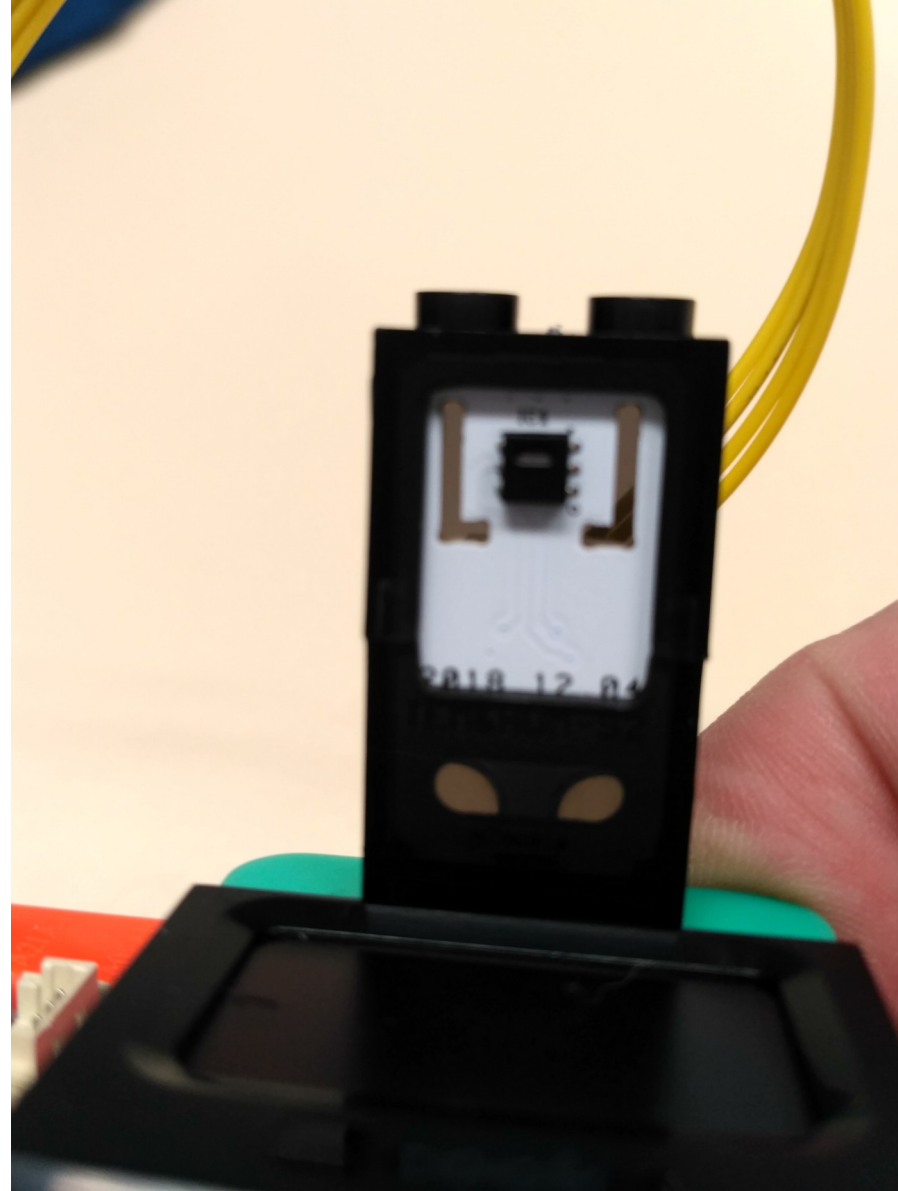
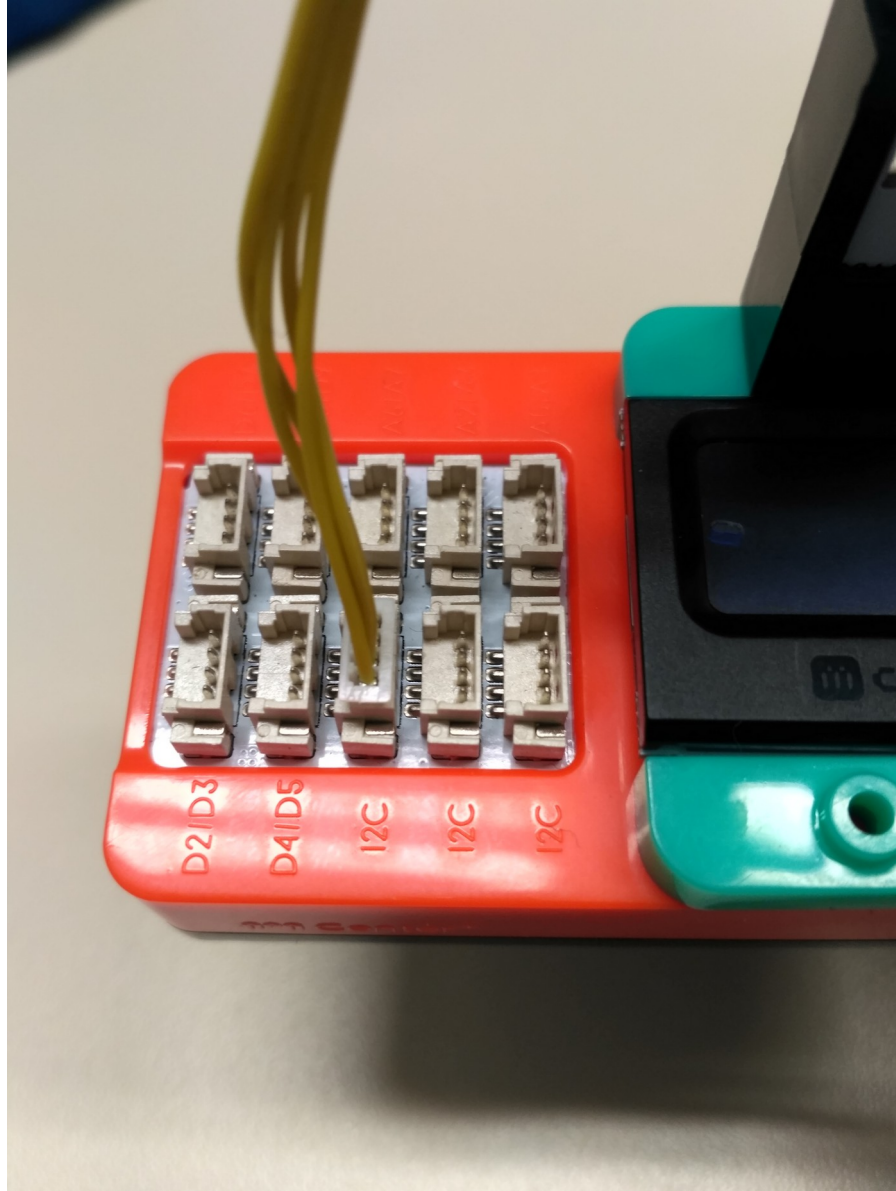
GND





Montagem

- O *display* fica na *interface* microduino (o painel em U); como é magnético ele só se conecta se estiver na posição correta – os contatos são automaticamente selecionados
- O sensor de temperatura e umidade ficará conectado em uma porta identificada como I2C (o *display* também é 'I2C', mas a conexão dele é automática nos contatos do 'U')



Arquivo com o código



buggy



CuritibaBitmap



faixaDeLEDS



temperaturaDisplay



temperaturaDisplayPaginasImagem

Código

temperaturaDisplay

```
1 #include "U8glib.h" //biblioteca padrão do Google
2 #include <Wire.h> //biblioteca de comunicação (I2C)
3 #include "DFRobot_SHT20.h" //biblioteca do sensor de temperatura e umidade - verifique qual o seu!
4
5 DFRobot_SHT20 sensorS2; //sensor de temperatura e umidade S2 / SHT20 - existem outros modelos!
6
7 U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE); // conexão do display OLED

28 void setup()
29 {
30     sensorS2.initSHT20();
31 }
```

```
34 void loop()  
35 {  
36     float umd = sensorS2.readHumidity();  
37     float temp = sensorS2.readTemperature();  
38  
39     u8g.firstPage();  
40     do {  
41         mostraTemperatura(temp);  
42         mostraUmidade(umd);  
43     }  
44     while( u8g.nextPage() );  
45     delay( 1000 );  
46  
47 }
```

Estrutura
padrão
da
biblioteca

```
9 void mostraTemperatura(float temp) {  
10     u8g.setFont(u8g_font_courB14);  
11     //u8g.setFont(u8g_font_courR10);  
12     u8g.drawStr( 0, 10, "Temperatura:"); //coluna, linha  
13     u8g.setPrintPos(15, 28);  
14     u8g.print(temp);  
15     u8g.print(" ");  
16     u8g.print(char(176));  
17     u8g.print("C");  
18 }
```

```
20 void mostraUmidade(float umd) {  
21     u8g.setFont(u8g_font_courB14); // fonte  
22     u8g.drawStr( 0, 48, "Umidade:");  
23     u8g.setPrintPos(15, 63);  
24     u8g.print(umd);  
25     u8g.print(" %");  
26 }
```


Bibliotecas necessárias

- "U8glib.h"
- "DFRobot_SHT20.h"
- Como anteriormente, usando o arquivo da biblioteca ou digitando o nome da biblioteca no gerenciador e clicando em instalar



GERENCIADOR DE BIBLIOTECAS

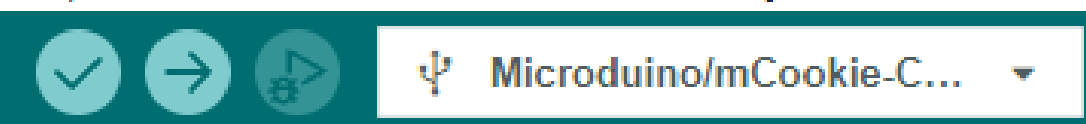
Tipo: Todos

Tópico: Todos

U8glib por oliver
<olikraus@gmail.com>
Supported display controller:
SSD1306, SSD1309, SSD1322,
SSD1325, SSD1327, SH1106,...
[Mais informações](#)

1.19.1

INSTALAR



GERENCIADOR DE BIBLIOTECAS

Tipo: Todos

Tópico: Todos

DFRobot_SHT20 por
DFRobot
This is a SHT20 I2C temperature &
humidity sensor with waterproof
probe. It comes with the 4C...
[Mais informações](#)

1.0.0

INSTALAR

temperaturaDi:

1	#
2	#
3	#
4	#
5	#
6	s
7	
8	
9	
10	

Se faltar uma biblioteca...

```
Saída
```

```
#include "U8glib.h" //biblioteca padrão do Google
||||| ^~~~~~
compilation terminated.

exit status 1

Compilation error: U8glib.h: No such file or directory
```

✖ Compilation error: U8glib.h: No such file or directory

MENSAGEM DE ERRO COPIAR

Ln 14, Col 1 SparkFun MaKey MaKey em COM6 2

Se faltar uma biblioteca...

temperaturaDisplay:3:10: error: DFRobot_SHT20.h: No such file or directory
#include "DFRobot_SHT20.h" //biblioteca do sensor de temperatura e umidade - verifique qual o seu!

^~~~~~

compilation terminated.

Foram encontradas múltiplas bibliotecas para "U8glib.h"

Usado: /home/simao/.arduino15/packages/Microduino/hardware/avr/1.0.1/libraries/
_01_Microduino_OLED_U8glib

Foram encontradas múltiplas bibliotecas para "Wire.h"

Usado: /home/simao/.arduino15/packages/Microduino/hardware/avr/1.0.1/libraries/
_10_System_Wire

exit status 1

DFRobot_SHT20.h: No such file or directory

1. Compilar e testar
2. Explicações do desenho da imagem

Desenhar no *display*

```
#include "U8glib.h" //biblioteca padrão do Google
```

Existem outras bibliotecas, o tratamento que vem a seguir é muito semelhante entre elas

```
#include "U8glib.h" //biblioteca padrão do Google
```

```
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);
```

```
#include "U8glib.h" //biblioteca padrão do Google
```

```
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);
```

U8GLIB_SSD1306_128X64 biblioteca, tipo de controlador (SSD1306) e tamanho (128 colunas, 64 linhas – lembre-se que, em ‘C’, vetores começam em zero; portanto, os endereços vão de 0-127 para colunas e 0-63 para linhas)

Modelos diferentes de *display* utilizam controladores diferentes, o que exige declarações (e bibliotecas) diferentes

(Na documentação em Chinês - grato ao Google pela tradução - há dois modelos de *display*, com controladores diferentes)

```
#include "U8glib.h" //biblioteca padrão do Google
```

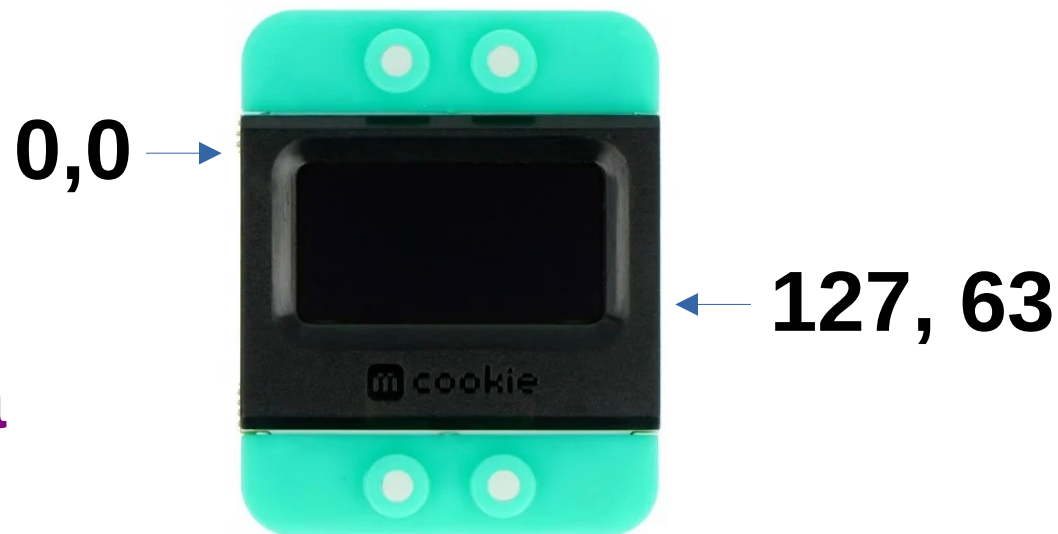
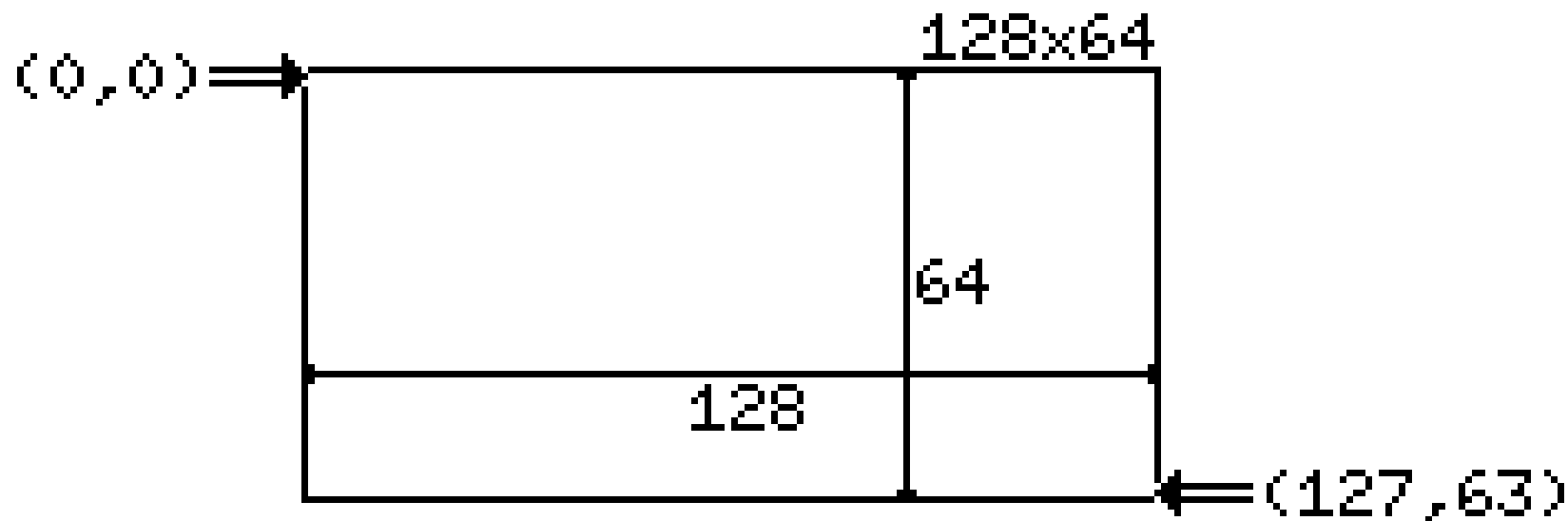
```
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);
```

u8g nome do objeto (você pode trocar para qualquer outro, por exemplo 'meuDisplay' – o código padrão do Google em geral virá com u8g). Este objeto vai se comportar como uma cópia do *display* na memória.

```
#include "U8glib.h" //biblioteca padrão do Google
```

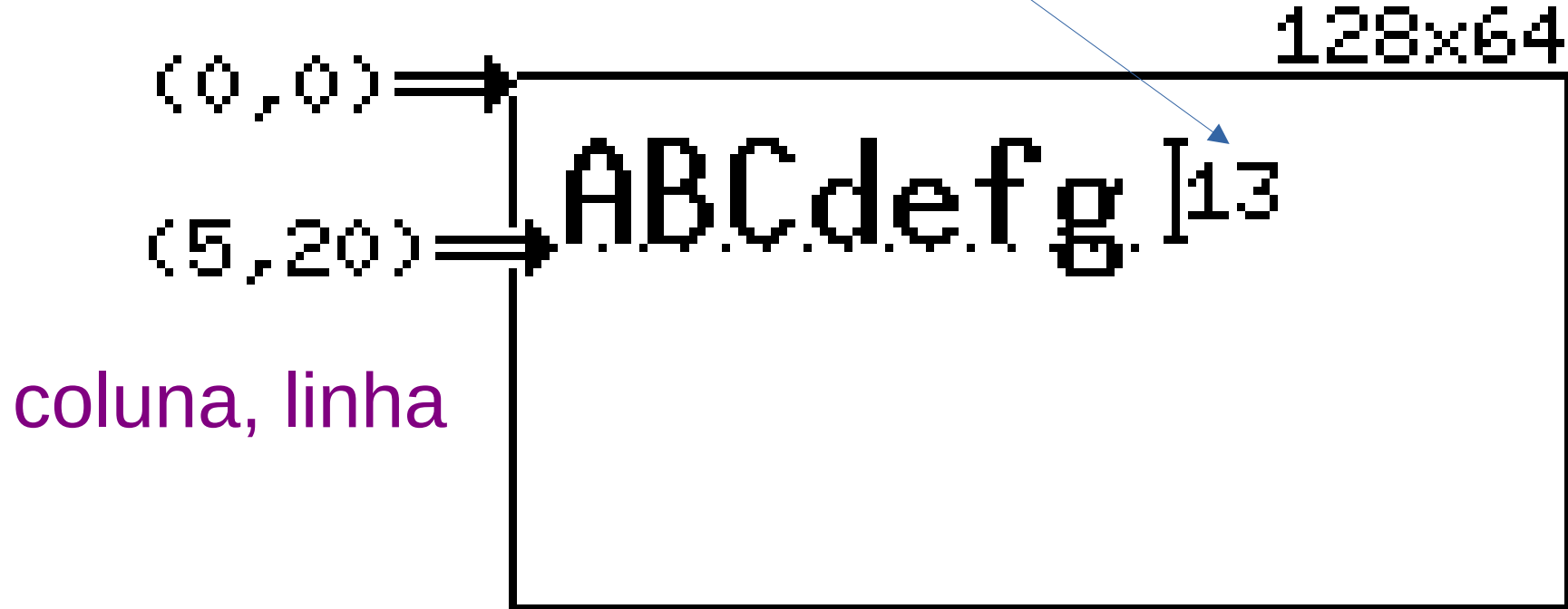
```
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);
```

U8G_I2C_OPT_NONE é uma constante para o tipo de conexão utilizado; neste caso, é para I2C (que utiliza os pinos SDA/ SCL para dados e sincronismo, respectivamente)



coluna, linha

```
u8g_SetFont(u8g, u8g_font_10x20);  
u8g_DrawStr(u8g, 5, 20, "ABCdefg");  
Int a = u8g_GetFontAscent(u8g);
```




```
u8g_SetFont(u8g, u8g_font_10x20);
```

u8g_SetFont função para informar qual conjunto de fontes de caracteres será utilizado

```
u8g_SetFont(u8g, u8g_font_10x20);
```

u8g nome do objeto utilizado em seu programa.

(ele foi declarado no início do programa:

```
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);)
```

```
u8g_SetFont(u8g, u8g_font_10x20);
```

u8g_font_10x20 nome da fonte de caracteres a ser utilizada (no mesmo programa, você poderá utilizar várias)

u8g_font_courR24: 123 ABC abcdefg [Adobe X11 Font](#)

u8g_font_helvB18: 123 ABC abcdefg [Adobe X11 Font](#)

u8g_font_helvR18: 123 ABC abcdefg [Adobe X11 Font](#)

u8g_font_profont29: 123 ABC abcdefg [ProFont](#)

20 Pixel Height

u8g_font_fur20: 123 ABC abcdefg [Free Universal](#)

u8g_font_fub20: 123 ABC abcdefg [Free Universal](#)

u8g_font_gdr20: 123 ABC abcdefg [Gentium](#)

u8g_font_gdb20: 123 ABC abcdefg [Gentium](#)

u8g_font_courB24: 123 ABC abcdefg [Adobe X11 Font](#)

21 Pixel Height

u8g_font_osb21: 123 ABC abcdefg [Old Standard](#)

u8g_font_osr21: 123 ABC abcdefg [Old Standard](#)

Exemplos
de fontes de
caracteres

https://github.com/olikraus/u8glib/wiki/fontsize

12 Pixel Height

u8g_font_gdr12: 123 ABC abcdefg [Gentium](#)

u8g_font_gdb12: 123 ABC abcdefg [Gentium](#)

u8g_font_helvB12: 123 ABC abcdefg [Adobe X11 Font](#)

u8g_font_helvR12: 123 ABC abcdefg [Adobe X11 Font](#) →

u8g_font_ncenB12: 123 ABC abcdefg [Adobe X11 Font](#)

u8g_font_ncenR12: 123 ABC abcdefg [Adobe X11 Font](#)

u8g_font_courB10, courB10

BBX Width 13, Height 21, Capital A 9

Font data size: 3355

32/0x20	! " # \$ % & ' () * + , - . /
48/0x30	0 1 2 3 4 5 6 7 8 9 : ; < = > ?
64/0x40	@ A B C D E F G H I J K L M N O
80/0x50	P Q R S T U V W X Y Z [\] ^ _
96/0x60	` a b c d e f g h i j k l m n o
112/0x70	p q r s t u v w x y z { } ~
128/0x80	
144/0x90	
160/0xa0	ı ċ € ₣ ¥ ¦ § ¨ © ª « ¬ ® ¯
176/0xb0	° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
192/0xc0	À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
208/0xd0	Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
224/0xe0	à á â ã ä å æ ç è é ê ë ì í î ï
240/0xf0	ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ

<https://github.com/olikraus/u8glib/wiki/fontgroupadobex11>

<https://github.com/olikraus/u8glib/wiki/fontgroupadobex11>

u8g_font_courB10

u8g_SetFont(u8g, **u8g_font_courB10**);

u8g_font_courB10, courB10

BBX Width 13, Height 21, Capital A 9

Font data size: 3355

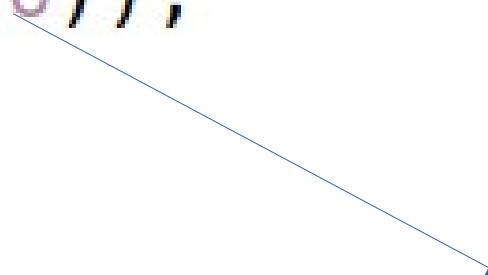
32/0x20	! " # \$ % & ' () * + , - . /
48/0x30	0 1 2 3 4 5 6 7 8 9 : ; < = > ?
64/0x40	@ A B C D E F G H I J K L M N O
80/0x50	P Q R S T U V W X Y Z [\] ^ _
96/0x60	` a b c d e f g h i j k l m n o
112/0x70	p q r s t u v w x y z { } ~
128/0x80	
144/0x90	
160/0xa0	ı ċ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯
176/0xb0	° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
192/0xc0	À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
208/0xd0	Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
224/0xe0	à á â ã ä å æ ç è é ê ë ì í î ï
240/0xf0	ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ

O primeiro caractere da linha é o 224, que é o 'à'. Então, o seguinte é o 225 (á) e o caractere 'ã' é o 227, o 'é' tem o código 233 e assim por diante.


```
9 void mostraTemperatura(float temp) {  
10     u8g.setFont(u8g_font_courB14);  
11     //u8g.setFont(u8g_font_courR10);  
12     u8g.drawStr( 0, 10, "Temperatura:");  
13     u8g.setPrintPos(15, 28);  
14     u8g.print(temp);  
15     u8g.print(" ");  
16     u8g.print(char(176));  
17     u8g.print("C");  
18 }
```

u8g_font_courB14, courB14
BBX Width 17, Height 26,
Font data size: 4784

32/0x20	! " # \$
48/0x30	0 1 2 3 4
64/0x40	@ A B C D
80/0x50	P Q R S T
96/0x60	` a b c d
112/0x70	p q r s t
128/0x80	
144/0x90	
160/0xa0	i ¢ £ ¤
176/0xb0	° ± ² ³ ´



(Outra biblioteca)

Usar uma letra minúscula

```
void limpa(void)
{
    displayOLED.clear();
    displayOLED.drawString(3, 0, "Temperatura"); //coluna, linha
    displayOLED.inverse();
    displayOLED.setCursor(4,1);
    displayOLED.print("Aleatoria");
    displayOLED.noInverse();
    displayOLED.setCursor(12, 5);
    displayOLED.print("o");
    displayOLED.draw2x2String(13, 5, "C");
}
```

Também dá para desenhar um pequeno círculo...

```
u8g_DrawStr(u8g, 5, 20, "ABCdefg");
```

u8g_DrawStr desenha uma *string* (sequência de caracteres) no objeto desejado (neste caso **u8g**), na posição coluna / linha passada (neste caso coluna **5** e linha **20**); a *string* vem indicada entre aspas duplas.

```
u8g_DrawStr(u8g, 5, 20, "ABCdefg");
```

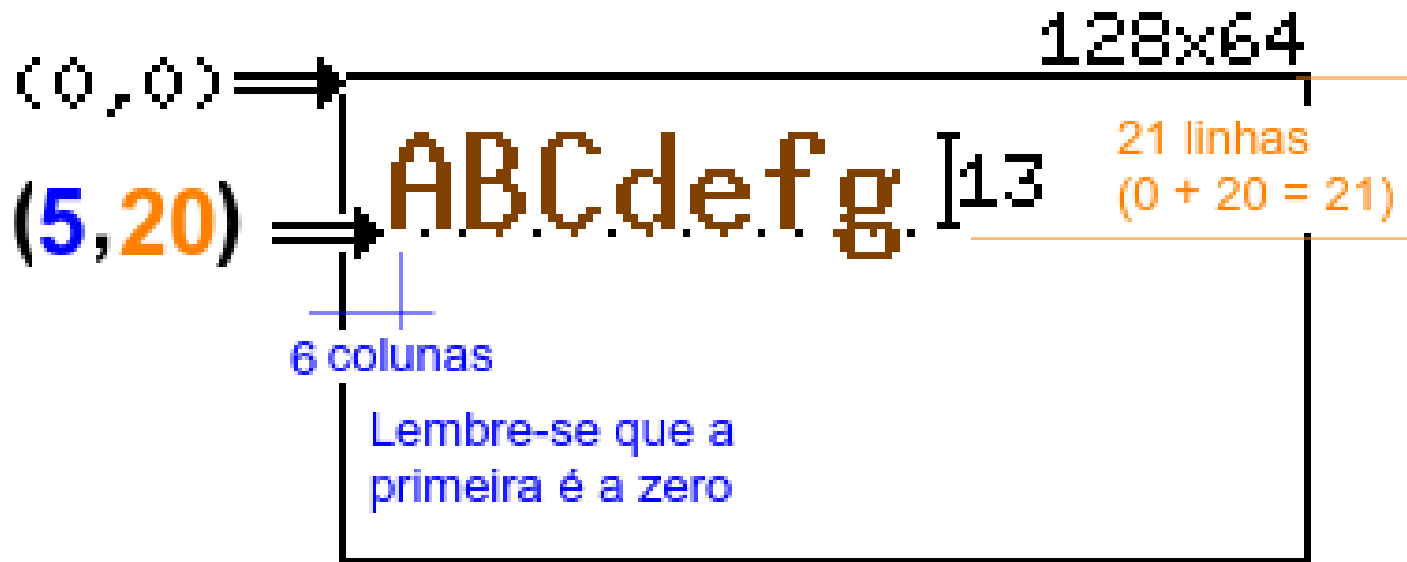
Objeto

coluna

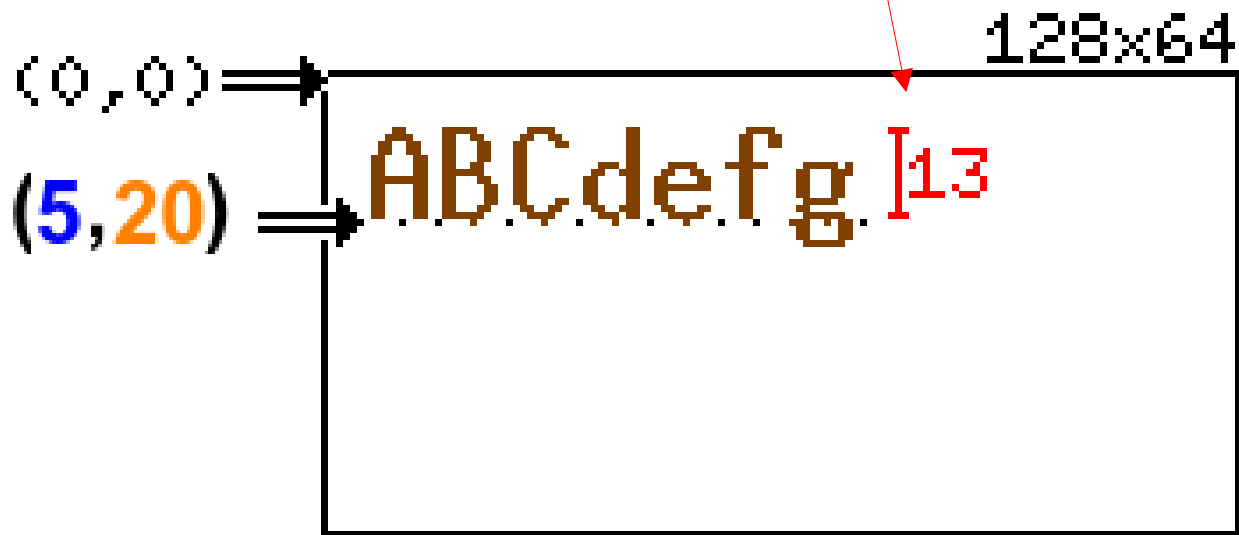
linha

string

```
(U8GLIB_SSD1306_128X64  
u8g(U8G_I2C_OPT_NONE);)
```



```
Int a = u8g_GetFontAscent(u8g);
```



A distância recomendada acima da linha de base para texto com espaçamento simples.

■ Topo/ top

■ Ascensão / Ascent

■ Linha base / Baseline

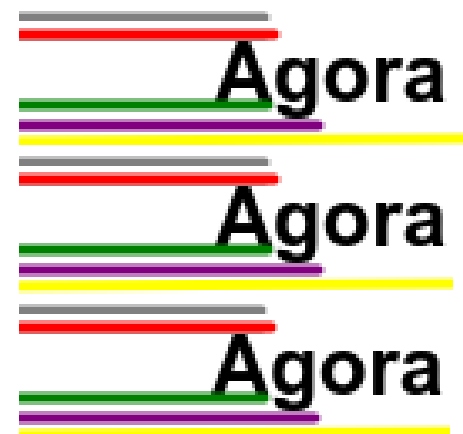
■ Descida / Descent

■ Inferior / Bottom



Agora

Várias linhas



Agora

Agora

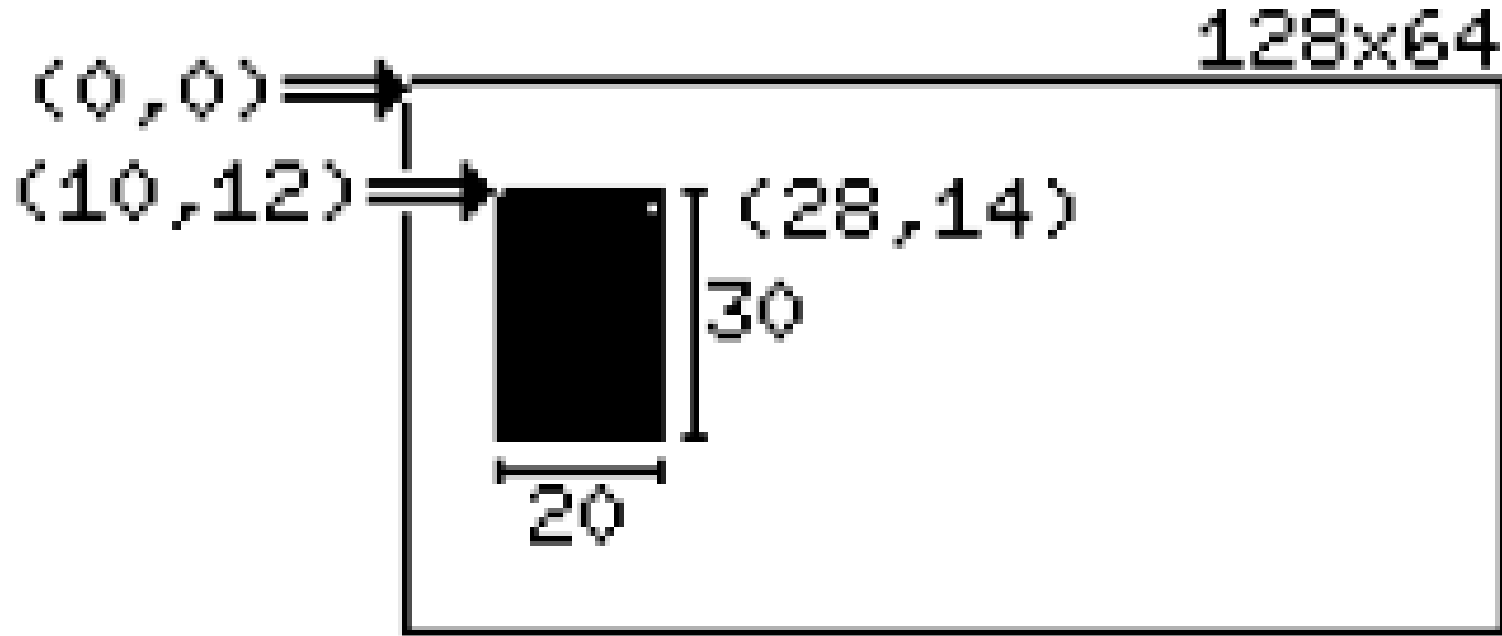
Agora

Entrelinha / Leading - Espaço adicional recomendado para adicionar entre linhas de texto.


```
9 void mostraTemperatura(float temp) {  
10     u8g.setFont(u8g_font_courB14);  
11     //u8g.setFont(u8g_font_courR10);  
12     u8g.drawStr( 0, 10, "Temperatura:");  
13     u8g.setPrintPos(15, 28);  
14     u8g.print(temp);  
15     u8g.print(" ");  
16     u8g.print(char(176));  
17     u8g.print("C");  
18 }
```

Desenhar formas e *pixels*

```
u8g.setColorIndex(1); //preto  
u8g.drawBox(10, 12, 20, 30);  
u8g.setColorIndex(0); //branco  
u8g.drawPixel(28, 14);
```



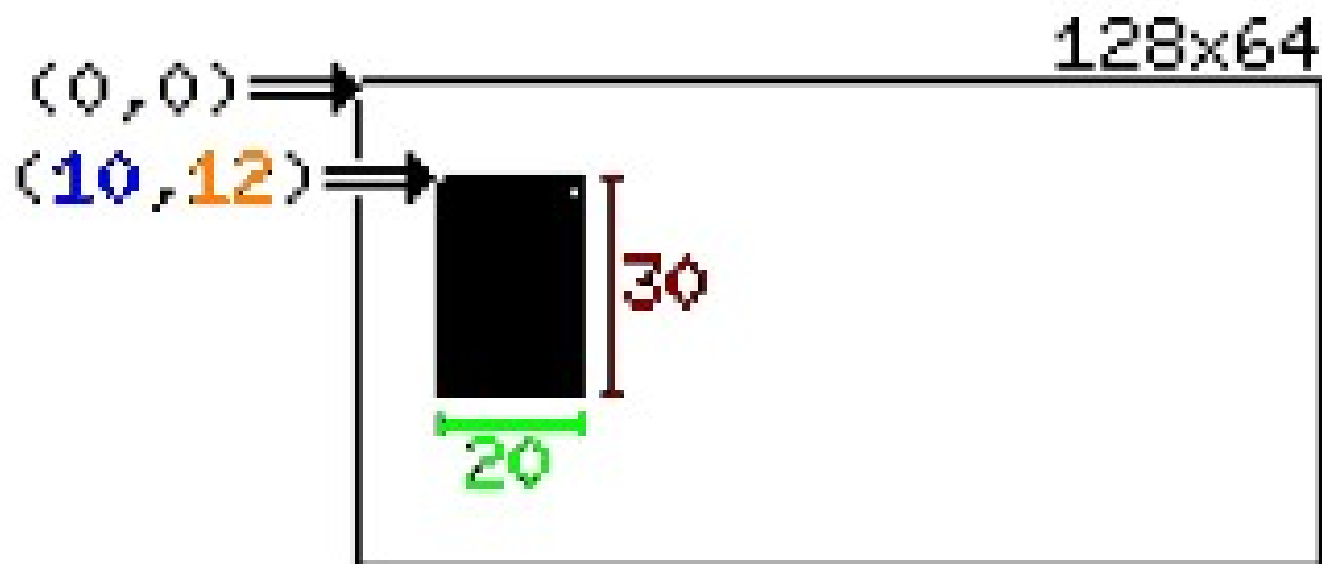
```
u8g.setColorIndex(1);
```

Estabelece a cor 1 = preto

```
u8g.drawBox(10, 12, 20, 30);
```

Desenha uma 'caixa' começando na coluna 10 e linha 12, com 20 colunas de largura e 30 linhas de altura

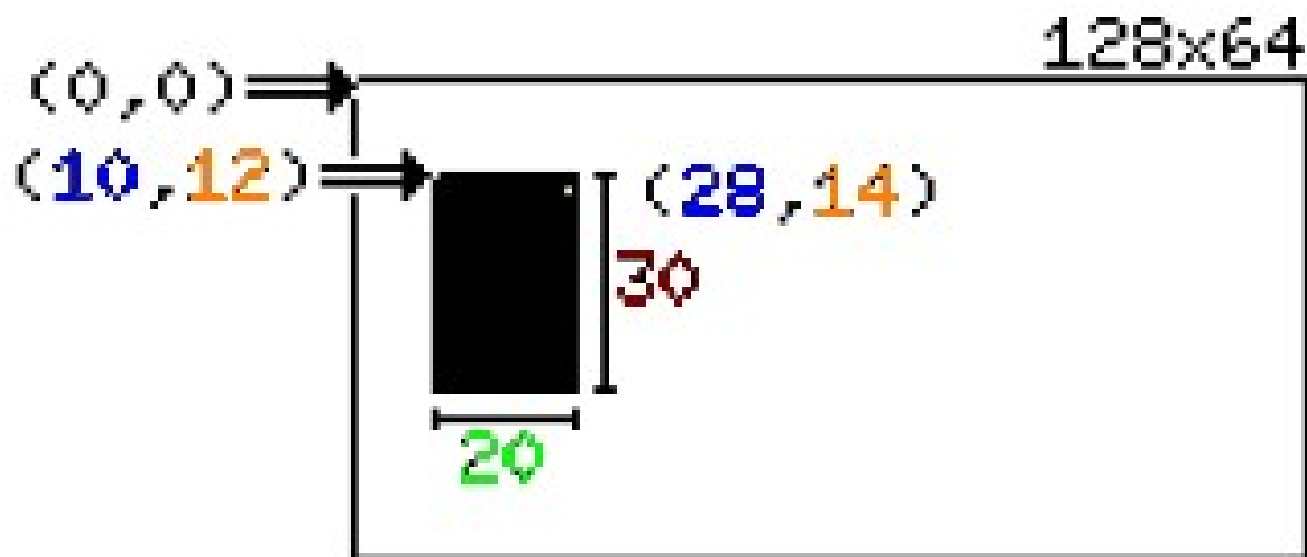
```
u8g.setColorIndex(1); //preto  
u8g.drawBox(10, 12, 20, 30);
```



`u8g.setColorIndex(0);`
Estabelece a cor 0 = branco

`u8g.drawPixel (28, 14);`
Desenha um único *pixel* na tela, na
coluna 28 e linha 14


```
u8g.setColorIndex(1); //preto  
u8g.drawBox(10, 12, 20, 30);  
u8g.setColorIndex(0); //branco  
u8g.drawPixel(28, 14);
```



Exemplo 3 - Exibindo uma imagem *bitmap*



Arquivo com o código



A imagem deve estar no formato XBM.

O conteúdo do arquivo será copiado para dentro do código.

Em síntese, a biblioteca vai desenhar pontinho por pontinho na tela em relação àquilo que for lido na sequência copiada.

Tipo de imagem
png (PNG)

Largura
1668 pixels

Altura
2045 pixels



CURITIBA

Tipo de imagem
png (PNG)

Largura
128 pixels

Altura
64 pixels



Tipo de imagem
xbm (XBM)

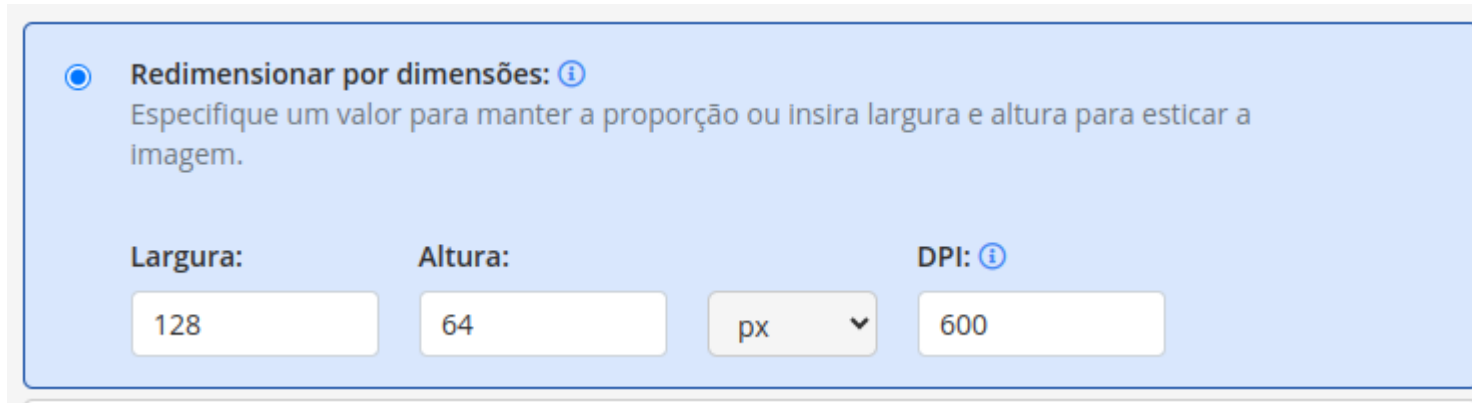
Largura
128 pixels

Altura
64 pixels



Para redimensionar a imagem, eu usei:

<https://www.img2go.com/pt/redimensionar-imagem>



The screenshot shows the 'Redimensionar por dimensões' (Resize by dimensions) section of the img2go.com website. It includes a radio button selection, a descriptive text, and input fields for width, height, unit, and DPI.

☒ **Redimensionar por dimensões:** ⓘ
Especifique um valor para manter a proporção ou insira largura e altura para esticar a Imagem.

Largura: Altura: DPI: ⓘ

Para transformar a imagem em 'XBM', eu usei:

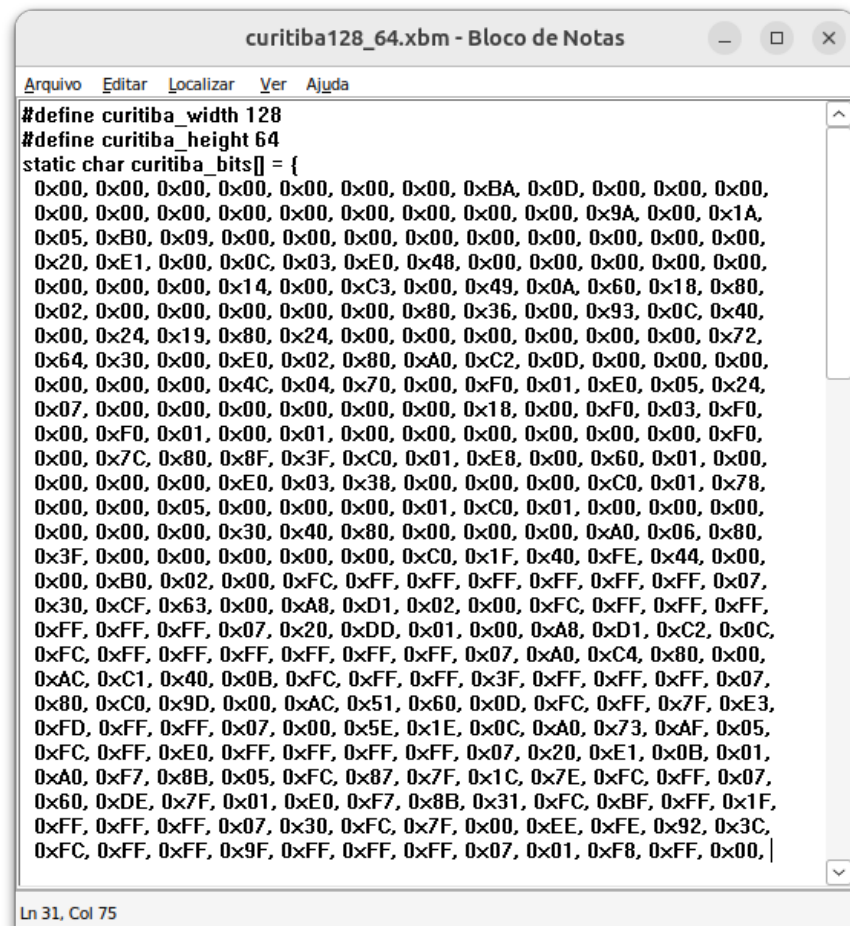
<https://convertio.co/pt/>



The screenshot shows the file upload and format selection area of the convertio.co website. It includes a file icon, the filename, a 'para' (to) label, and a dropdown menu for the output format.

 curitiba128_64.png para

Abra o arquivo XBM em um editor de textos:

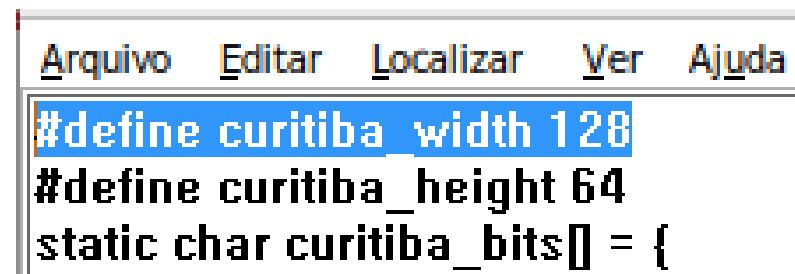


```

curitiba128_64.xbm - Bloco de Notas
Arquivo Editar Localizar Ver Ajuda
#define curitiba_width 128
#define curitiba_height 64
static char curitiba_bits[] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBA, 0x0D, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x9A, 0x00, 0x1A,
    0x05, 0xB0, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x20, 0xE1, 0x00, 0x0C, 0x03, 0xE0, 0x48, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x14, 0x00, 0xC3, 0x00, 0x49, 0x0A, 0x60, 0x18, 0x80,
    0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x36, 0x00, 0x93, 0x0C, 0x40,
    0x00, 0x24, 0x19, 0x80, 0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x72,
    0x64, 0x30, 0x00, 0xE0, 0x02, 0x80, 0xA0, 0xC2, 0x0D, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x4C, 0x04, 0x70, 0x00, 0xF0, 0x01, 0xE0, 0x05, 0x24,
    0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x00, 0xF0, 0x03, 0xF0,
    0x00, 0xF0, 0x01, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0,
    0x00, 0x7C, 0x80, 0x8F, 0x3F, 0xC0, 0x01, 0xE8, 0x00, 0x60, 0x01, 0x00,
    0x00, 0x00, 0x00, 0xE0, 0x03, 0x38, 0x00, 0x00, 0x00, 0xC0, 0x01, 0x78,
    0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x01, 0xC0, 0x01, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x30, 0x40, 0x80, 0x00, 0x00, 0x00, 0xA0, 0x06, 0x80,
    0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x1F, 0x40, 0xFE, 0x44, 0x00,
    0x00, 0xB0, 0x02, 0x00, 0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07,
    0x30, 0xCF, 0x63, 0x00, 0xA8, 0xD1, 0x02, 0x00, 0xFC, 0xFF, 0xFF, 0xFF,
    0xFF, 0xFF, 0xFF, 0x07, 0x20, 0xDD, 0x01, 0x00, 0xA8, 0xD1, 0xC2, 0x0C,
    0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07, 0xA0, 0xC4, 0x80, 0x00,
    0xAC, 0xC1, 0x40, 0x0B, 0xFC, 0xFF, 0xFF, 0x3F, 0xFF, 0xFF, 0xFF, 0x07,
    0x80, 0xC0, 0x9D, 0x00, 0xAC, 0x51, 0x60, 0x0D, 0xFC, 0xFF, 0x7F, 0xE3,
    0xFD, 0xFF, 0xFF, 0x07, 0x00, 0x5E, 0x1E, 0x0C, 0xA0, 0x73, 0xAF, 0x05,
    0xFC, 0xFF, 0xE0, 0xFF, 0xFF, 0xFF, 0xFF, 0x07, 0x20, 0xE1, 0x0B, 0x01,
    0xA0, 0xF7, 0x8B, 0x05, 0xFC, 0x87, 0x7F, 0x1C, 0x7E, 0xFC, 0xFF, 0x07,
    0x60, 0xDE, 0x7F, 0x01, 0xE0, 0xF7, 0x8B, 0x31, 0xFC, 0xBF, 0xFF, 0x1F,
    0xFF, 0xFF, 0xFF, 0x07, 0x30, 0xFC, 0x7F, 0x00, 0xEE, 0xFE, 0x92, 0x3C,
    0xFC, 0xFF, 0xFF, 0x9F, 0xFF, 0xFF, 0xFF, 0x07, 0x01, 0xF8, 0xFF, 0x00,

```

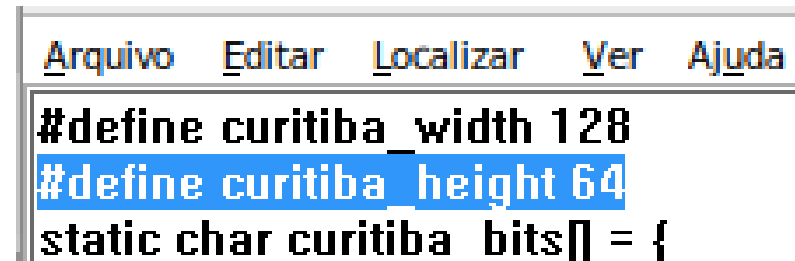
Ln 31, Col 75



```

Arquivo Editar Localizar Ver Ajuda
#define curitiba_width 128
#define curitiba_height 64
static char curitiba_bits[] = {

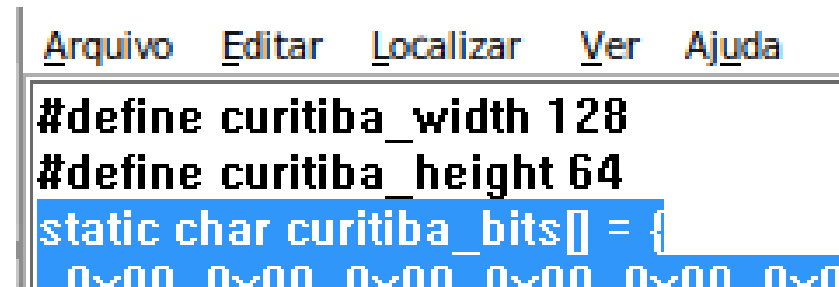
```



```

Arquivo Editar Localizar Ver Ajuda
#define curitiba_width 128
#define curitiba_height 64
static char curitiba_bits[] = {

```

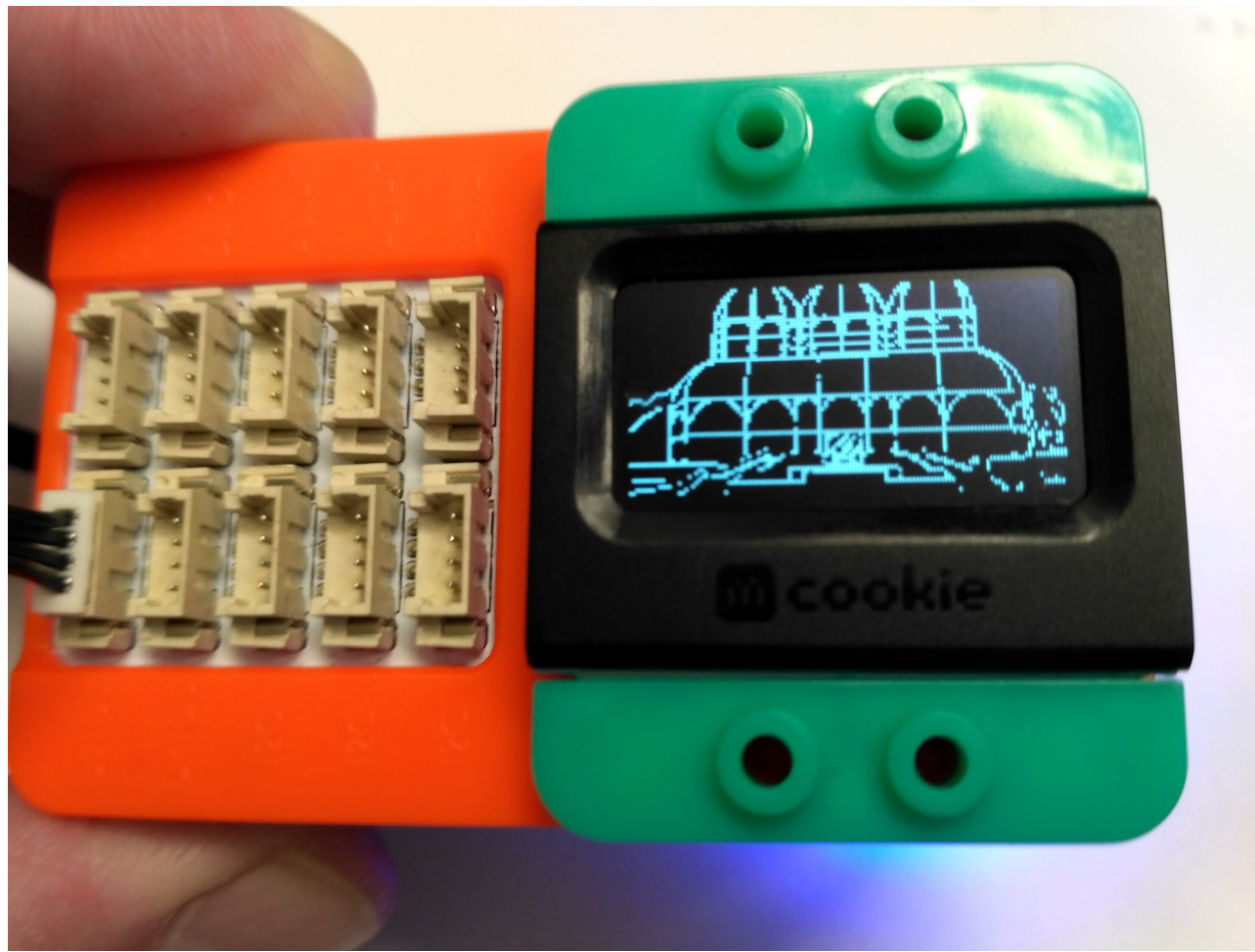


```

Arquivo Editar Localizar Ver Ajuda
#define curitiba_width 128
#define curitiba_height 64
static char curitiba_bits[] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00

```


Algumas imagens dão melhor resultado

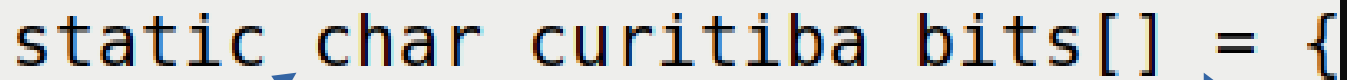


```
1 /*
2 https://forum.arduino.cc/t/u8glib-and-bitmap-creation-display/148125/2
3 1- Carregue sua imagem no gimp
4 2- Converta-a em um bitmap indexado, 2 níveis, preto e branco, talvez use dithering
5 3- Exporte como XBM (extensão .xbm)
6 4- Copie o conteúdo do arquivo .xbm para seu arquivo .ino (abra como texto e copie)
7 5- Mude o que você copiou para
8     static unsigned char "nome" U8G_PROGMEM = {
9     - Foi incluída a palavra unsigend
10    - Foi incluída a declaração U8G_PROGMEM
11 */

14 #include "U8glib.h" //biblioteca padrão do Google
15
16 // conexão do display OLED - GND, VCC, SDA, SCL no conector
17 // Arduino UNO SDA = A4  SCL = A5
18 U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);
```

```
1 /*
2 https://forum.arduino.cc/t/u8glib-and-bitmap-creation-display/148125/2
3 1- Carregue sua imagem no gimp
4 2- Converta-a em um bitmap indexado, 2 níveis, preto e branco, talvez use dithering
5 3- Exporte como XBM (extensão .xbm)
6 4- Copie o conteúdo do arquivo .xbm para seu arquivo .ino (abra como texto e copie)
7 5- Mude o que você copiou para
8     static unsigned char "nome" U8G_PROGMEM = {
9     - Foi incluída a palavra unsigend
10    - Foi incluída a declaração U8G_PROGMEM
11 */
```

```
20 #define curitiba_width 128
21 #define curitiba_height 64
22 static unsigned char curitiba_bits[] U8G_PROGMEM = {
```



```
static char curitiba_bits[] = {
```

```
static unsigned char curitiba_bits[] U8G_PROGMEM = {
```

Copiado (CTRL +C, CTRL+V) do arquivo XBM

```

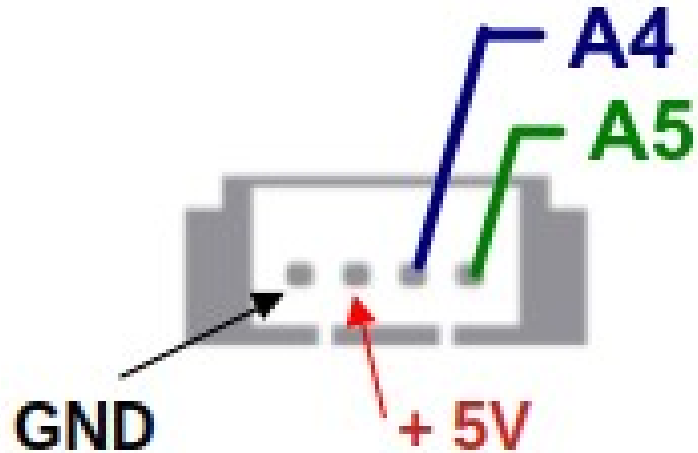
22 static unsigned char CuritibaBits[] URG PROGMEM = {
23     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
24     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
25     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
26     0x20, 0xE1, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
27     0x00, 0x00, 0x00, 0x14, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
28     0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
29     0x00, 0x24, 0x10, 0x00, 0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
30     0x64, 0x30, 0x00, 0xE0, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
31     0x00, 0x00, 0x00, 0x40, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
32     0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
33     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
34     0x00, 0x7C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
35     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
36     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
37     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
38     0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
39     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
40     0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
41     0xFF, 0xFF, 0xFF, 0x07, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
42     0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
43     0xAC, 0xC1, 0x40, 0x00, 0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
44     0x80, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
45     0xFD, 0xFF, 0xFF, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
46     0xFC, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07, 0x20, 0xE1, 0x00,
47     0xA0, 0xF7, 0x80, 0x05, 0xFC, 0x87, 0x7F, 0x1C, 0x7E, 0xFC, 0xFF, 0x07,
48     0x60, 0xDE, 0x7F, 0x01, 0xE0, 0xF7, 0x80, 0xFC, 0x00, 0xFF, 0xFF, 0x07,
49     0xFF, 0xFF, 0xFF, 0x07, 0x30, 0xFC, 0x7F, 0x00, 0x00, 0xFF, 0x92, 0x3C,
50     0xFC, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0x00, 0xFF, 0x00,
51     0xFE, 0xD2, 0xF2, 0x25, 0xFC, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0xFF, 0x07,
52     0x07, 0xF1, 0x7F, 0x00, 0xFC, 0x52, 0xD2, 0x04, 0xFC, 0xFF, 0xFF, 0x0F,
53     0xFF, 0xFF, 0xFF, 0x07, 0x00, 0xFE, 0x1F, 0x10, 0xDC, 0x21, 0x51, 0x02,
54     0xFC, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0x07, 0xC0, 0xFE, 0xC0, 0x00,
55     0xF4, 0xFC, 0x70, 0x62, 0xFC, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0x07, 0x07,
56     0x00, 0xC2, 0x0F, 0x43, 0x60, 0xFC, 0xF0, 0x31, 0xFC, 0xFF, 0xFF, 0x07,
57     0xFF, 0xFF, 0xFF, 0x07, 0x00, 0xF8, 0x0F, 0x10, 0x40, 0xCE, 0xF8, 0x18,
58     0xFC, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0xFD, 0x1F, 0x10,
59     0x78, 0x0F, 0xFF, 0x0F, 0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07,
60     0x30, 0xFC, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
61     0xFF, 0xFF, 0xFF, 0x07, 0x10, 0xFC, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00,
62     0xFD, 0xFF, 0xFF, 0x1F, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0x00, 0x00, 0x00,
63     0x30, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x07, 0x00,
64     0xFD, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0xFE, 0xF3, 0x20, 0xFF, 0x1F,
65     0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
66     0xF6, 0xFF, 0xFF, 0x1F, 0xFF, 0xFF, 0xFF, 0x01, 0x00, 0x00, 0x00, 0x00,
67     0x18, 0x00, 0x78, 0x52, 0x00, 0xFF, 0xFF, 0x0F, 0xFE, 0xFF, 0xFF, 0x00,
68     0x04, 0xFF, 0x2F, 0x00, 0x00, 0x00, 0x04, 0xE3, 0x00, 0x00, 0xFF, 0x7F,
69     0xFF, 0xFF, 0x3F, 0x00, 0x00, 0xFF, 0x47, 0x00, 0x00, 0x00, 0x00, 0x00,
70     0x01, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F, 0x00, 0x00, 0xFF, 0x01, 0x00,
71     0x00, 0x00, 0xFF, 0x3F, 0xF2, 0xF8, 0xFF, 0xFF, 0xFF, 0xFF, 0x03, 0x00,
72     0x1E, 0xF3, 0x0F, 0x00, 0x00, 0x00, 0x06, 0xF0, 0x7F, 0xF2, 0xC1, 0xFF, 0xFF,
73     0xFF, 0x7F, 0x00, 0x00, 0x1E, 0xE3, 0x02, 0x00, 0x00, 0x02, 0x70, 0xFC,
74     0x77, 0x00, 0xFE, 0xFF, 0xFF, 0x07, 0x00, 0x00, 0x66, 0x40, 0x00, 0x00, 0x00,
75     0x00, 0xFE, 0xCF, 0xC7, 0xFF, 0x01, 0x02, 0xFC, 0x07, 0x00, 0xF8, 0x3F,
76     0x6A, 0xF8, 0x0F, 0x00, 0x00, 0x7C, 0xFC, 0xE1, 0xFF, 0xF1, 0x07, 0x00,
77     0x00, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0x00, 0x3F, 0x00, 0x08,
78     0xFF, 0x24, 0x12, 0xFE, 0x08, 0x60, 0xF8, 0xFF, 0x1F, 0x0E, 0x1F, 0x00,
79     0x00, 0x1F, 0x06, 0xFF, 0xFF, 0x30, 0x13, 0xC8, 0x88, 0x40, 0xF0, 0xFF,
80     0x0F, 0x0C, 0x7F, 0x00, 0x00, 0xC0, 0x87, 0x00, 0x00, 0xC7, 0x3F, 0x18, 0xCC,
81     0x98, 0xC1, 0x3F, 0x7C, 0x20, 0x7C, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x10,
82     0x0E, 0xF0, 0xFF, 0xFF, 0xFD, 0xFF, 0x37, 0x67, 0xE0, 0x03, 0x00, 0x00,
83     0x00, 0x00, 0x00, 0x83, 0x3F, 0xFE, 0x7F, 0x00, 0xF8, 0xFF, 0xC3, 0x07,
84     0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,
85     0xF8, 0xBF, 0x10, 0x7F, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
86     0x07, 0x60, 0x00, 0xC0, 0xBF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
87     0x00, 0x00, 0x00, 0x00, 0x03, 0x40, 0x00, 0xF8, 0x87, 0x7F, 0x00, 0x38,
88     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06,
89     0x3C, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
90     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
91     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
92     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
93     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
94     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
95     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
96     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
97     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
98     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
99     0x00, 0xFF, 0x1F, 0x0F, 0x7C, 0xFC, 0xFF, 0x78, 0xFE, 0x7F, 0x1E, 0xFE,
100    0x7F, 0xC0, 0x1F, 0x00, 0xC0, 0x07, 0x0C, 0x0F, 0x7C, 0x3C, 0xF0, 0x79,
101    0xC0, 0x07, 0x1E, 0x1E, 0x78, 0xE0, 0x3F, 0x00, 0xC0, 0x03, 0x00, 0x00,
102    0x7C, 0x3C, 0x00, 0x79, 0xC0, 0x03, 0x1E, 0xFE, 0x7F, 0xF0, 0x7D, 0x00,
103    0xC0, 0x03, 0x00, 0x0F, 0x7C, 0xFC, 0x7F, 0x78, 0xC0, 0x03, 0x1E, 0xFE,
104    0x7F, 0xF0, 0x78, 0x00, 0xC0, 0x07, 0x00, 0x0F, 0x7C, 0x3C, 0x3E, 0x78,
105    0xC0, 0x03, 0x1E, 0x1E, 0xF0, 0xF8, 0xFF, 0x00, 0x80, 0xFF, 0x1F, 0xFF,
106    0x3F, 0x3C, 0xF8, 0x78, 0xC0, 0x03, 0x1E, 0xFE, 0xFF, 0xFC, 0xFF, 0x01,
107    0x00, 0xFE, 0x07, 0xFC, 0x0F, 0x3C, 0xF0, 0x79, 0xC0, 0x03, 0x1E, 0xFE,
108    0x3F, 0x3E, 0xE0, 0x03, };

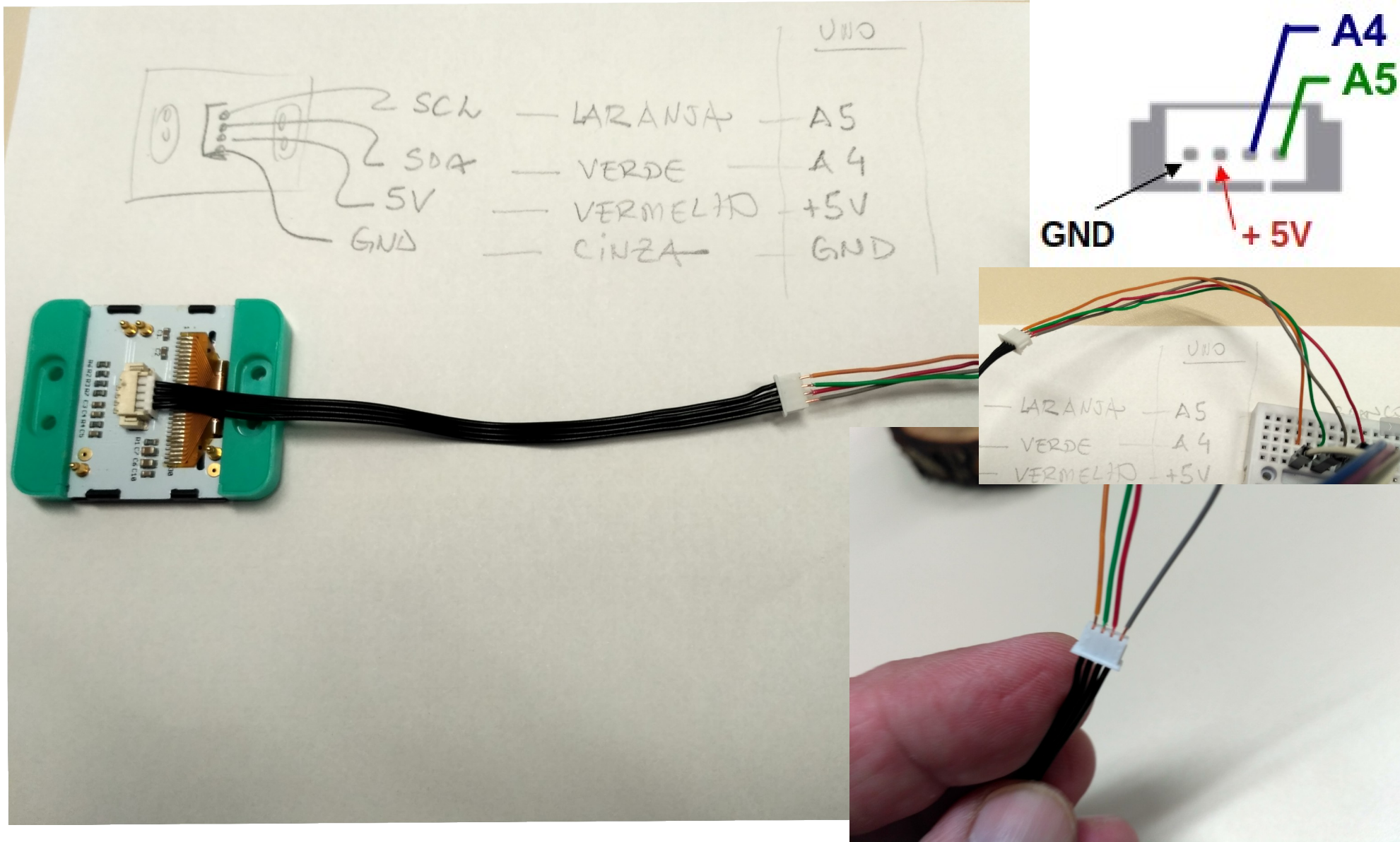
```

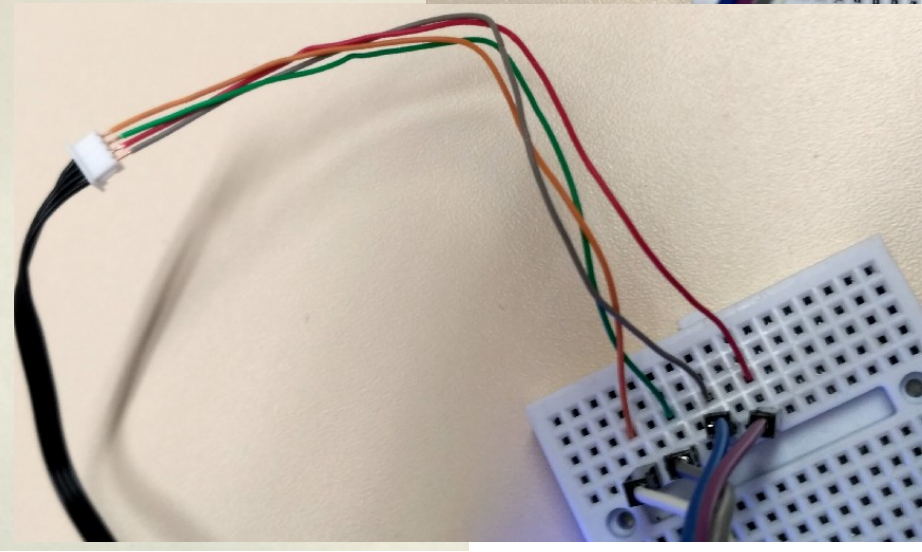
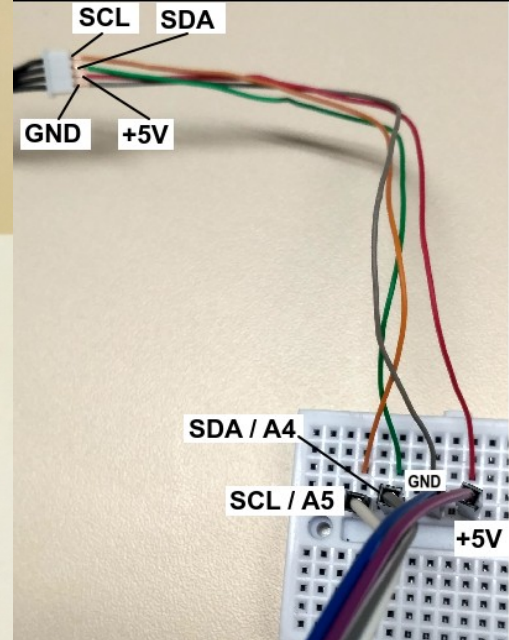
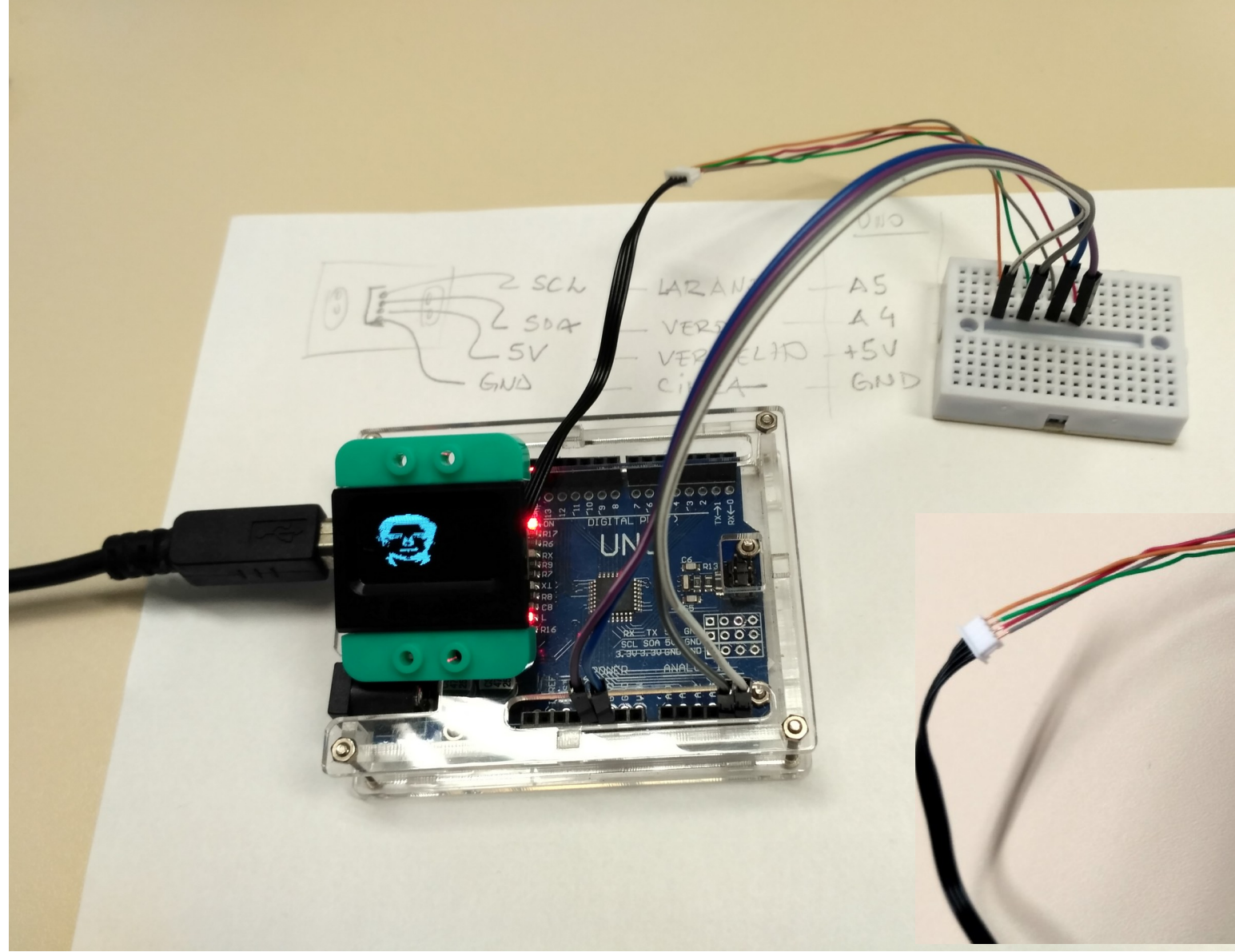
```
111 void setup() {
112     u8g.setColorIndex(1);
113 }
114
115
116 void loop() {
117     u8g.firstPage();
118     do {
119         u8g.drawXBMP( 0, 0, curitiba_width, curitiba_height, curitiba_bits);
120     }
121     while(u8g.nextPage());
122 }
```

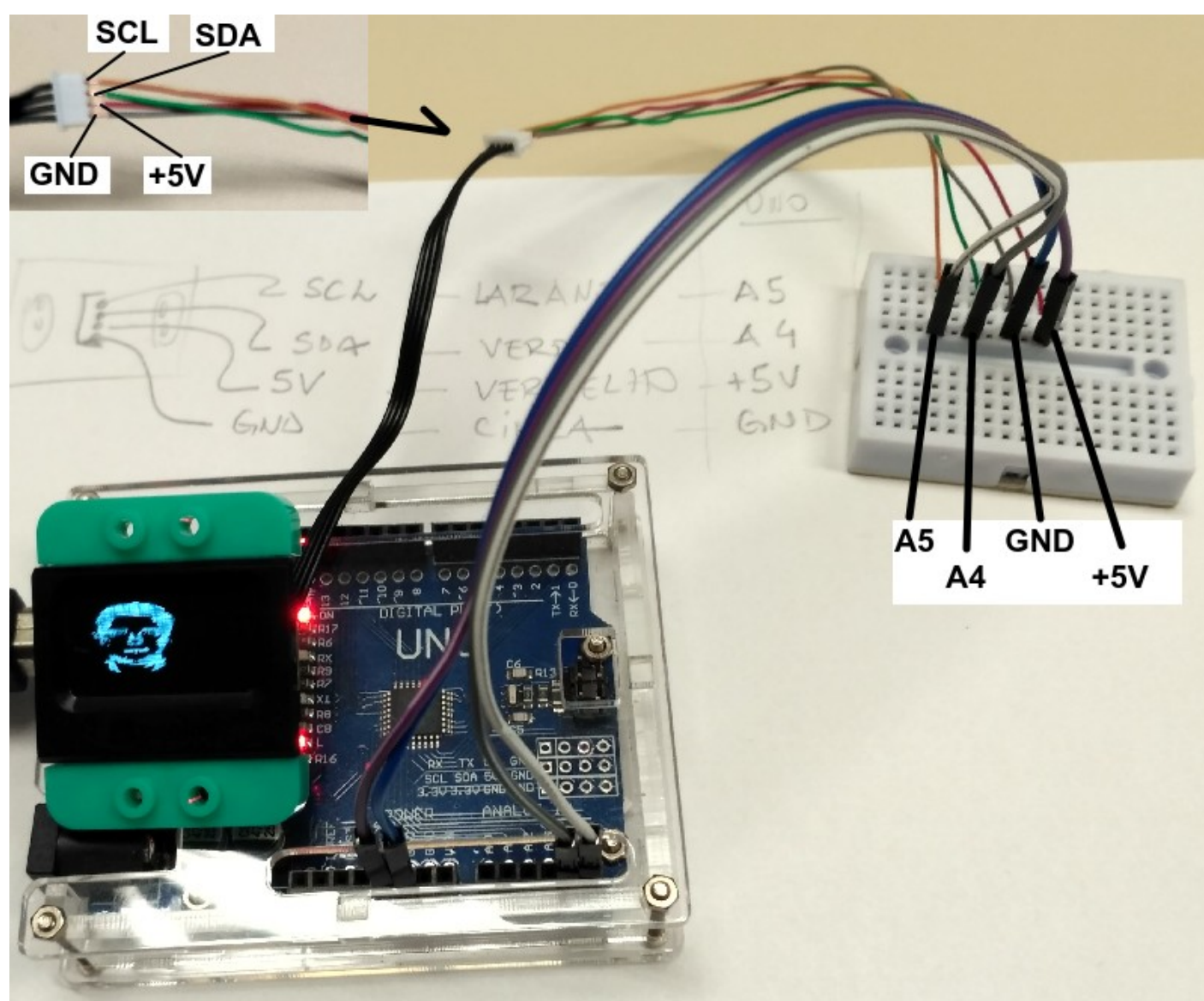
Arduino UNO

- Para ligar no Arduino UNO temos que utilizar a interface I2C, que são os pinos 'A4' para SDA e 'A5' para SCL

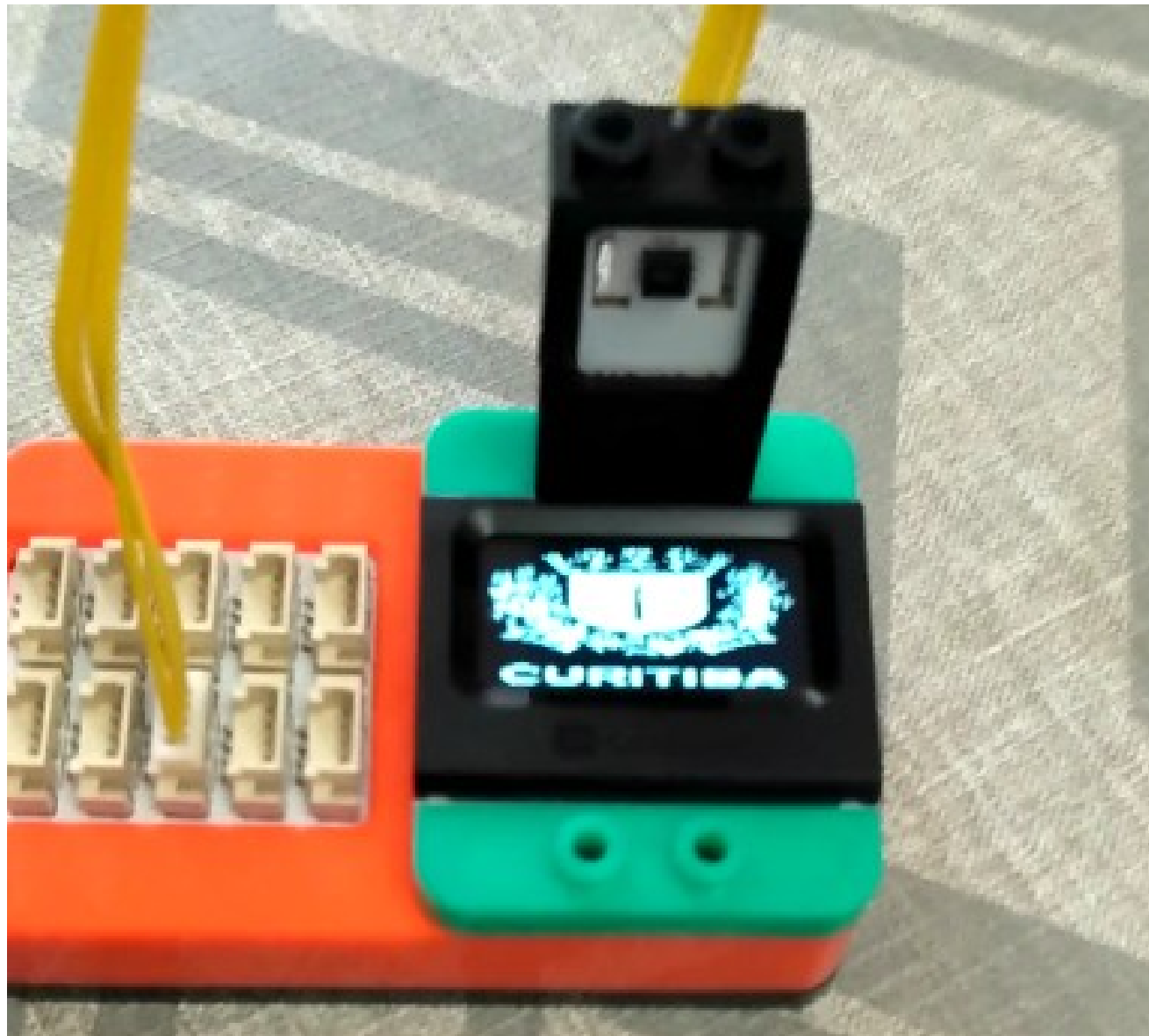








Compilar e testar



Exemplo 4

Exibir a
temperatura,
umidade, imagem
e informações

Arquivo com o código



buggy



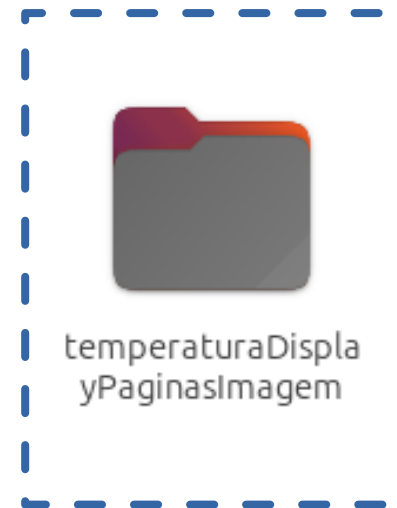
CuritibaBitmap



faixaDeLEDS



temperaturaDisplay



temperaturaDisplayPaginasImagem

temperaturaDisplayPaginasImagem

```
1 #include "U8glib.h" //biblioteca padrão do Google
2 #include <Wire.h> //biblioteca de comunicação
3 #include "DFRobot_SHT20.h" //biblioteca do sensor de temperatura e umidade
4 #define curitiba_width 128
5 #define curitiba_height 64
6 static unsigned char curitiba_bits[] U8G_PROGMEM = {
7     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xBA, 0x0D, 0x00, 0x00, 0x00,
8     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x9A, 0x00, 0x1A,
9     0x05, 0xB0, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
10    0x20, 0xE1, 0x00, 0x0C, 0x03, 0xE0, 0x48, 0x00, 0x00, 0x00, 0x00, 0x00,
11    0x00, 0x00, 0x00, 0x14, 0x00, 0xC3, 0x00, 0x49, 0x0A, 0x60, 0x18, 0x80,
12    0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x36, 0x00, 0x93, 0x0C, 0x40,
```

```
94 DFRobot_SHT20 sensorS2; //sensor de temperatura e umidade S2 / SHT20 - existem outros modelos!
95
96 U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE); // conexão do display OLED
97
98 int paginaAtual = 0;
99 const int qtdadePaginas = 5;
100 float umd = 0.0;
101 float temp = 0.0;
```



```
103 void pagApresenta();
104 void pagTemperatura();
105 void pagUmidade();
106 void pagProf();
107 void pagCuritiba();
108 void (*paginas[qtdadePaginas])() = {pagApresenta,pagTemperatura,pagUmidade,pagProf,pagCuritiba}
109 int tempoExibicao[qtdadePaginas] = {800,1500,1500,1000,2000};
```

```
111 void pagApresenta(){
112     u8g.setFont(u8g_font_courB14);
113     u8g.setPrintPos(28, 20);
114     u8g.print("V");
115     u8g.print(char(225));
116     u8g.print("rias");
117     u8g.setPrintPos(23, 50);
118     u8g.print("P");
119     u8g.print(char(225));
120     u8g.print("ginas");
121 }
```

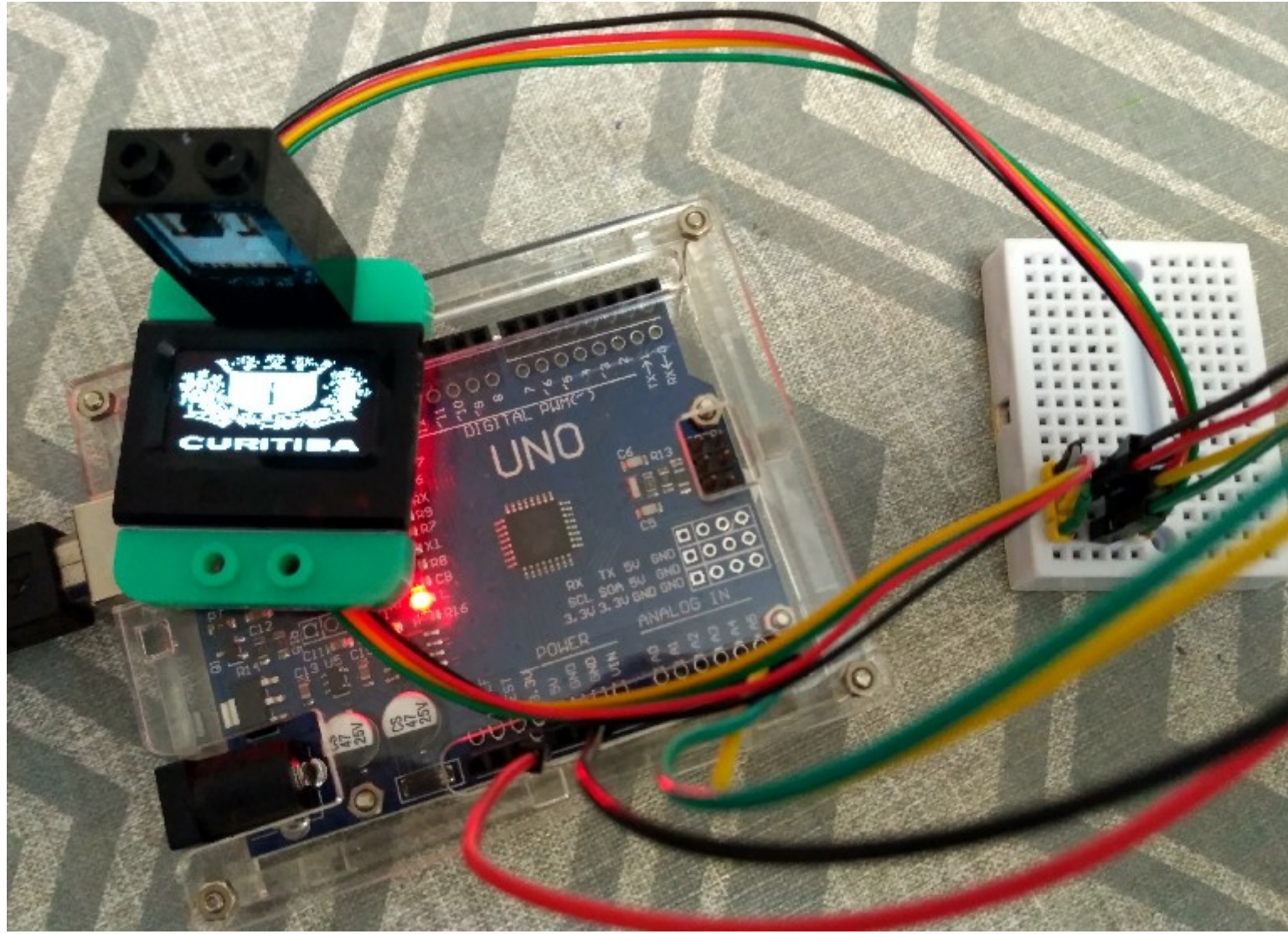
```
123 void pagTemperatura() {
124     u8g.setFont(u8g_font_courR14);
125     u8g.drawStr( 5, 12, "Temperatura:"); //coluna, linha
126     u8g.setPrintPos(15, 52);
127     u8g.setFont(u8g_font_fur20);
128     u8g.print(temp);
129     u8g.print(" ");
130     u8g.print(char(176));
131     u8g.print("C");
132 }
```

```
134 void pagUmidade() {  
135     u8g.setFont(u8g_font_courR14); // fonte  
136     u8g.drawStr(25, 12, "Umidade");  
137     u8g.setFont(u8g_font_fur20);  
138     u8g.setPrintPos(15, 52);  
139     u8g.print(umd);  
140     u8g.print(" %");  
141 }
```

```
143 void pagProf(){
144     u8g.setFont(u8g_font_courB14);
145     u8g.setPrintPos(35, 32);
146     u8g.print("Prof.");
147     u8g.setPrintPos(35, 48);
148     u8g.print("Sim");
149     u8g.print(char(227));
150     u8g.print("o");
151 }
152
153 void pagCuritiba(){
154     u8g.drawXBMP( 0, 0, curitiba_width, curitiba_height, curitiba_bits);
155 }
```

```
157 void setup()  
158 {  
159     sensorS2.initSHT20();  
160     u8g.setColorIndex(1);  
161 }  
162  
163 void atualizaSensor(){  
164     temp = sensorS2.readTemperature();  
165     umd = sensorS2.readHumidity();  
166 }
```

```
168 void loop()  
169 {  
170     atualizaSensor();  
171     u8g.firstPage();  
172     do {  
173         (*paginas[paginaAtual])();  
174     }  
175     while(u8g.nextPage());  
176     delay(tempoExibicao[paginaAtual]);  
177     paginaAtual++;  
178     if (paginaAtual == qtdadePaginas)  
179         paginaAtual = 0;  
180 }
```

Bônus

Instalar *Makey Makey*

Arquivo / Preferências

[https://raw.githubusercontent.com/
sparkfun/Arduino_Boards/master/
IDE_Board_Manager/
package_sparkfun_index.json](https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json)

piscarMicrobit | Arduino IDE 2.1.1

Arquivo Editar Rascunho Ferramentas Ajuda



BBC micro:bit



GERENCIADOR DE PLACAS

SparkFun AVR



Tipo:

Todos

**SparkFun AVR Boards**

por SparkFun Electronics

Boards included in this package:
RedBoard, MaKey MaKey, Pro
Micro, Fio v3, Qduino Mini, Digit...

[Mais informações](#)

1.1.13

**INSTALAR**

[-bitmap-creation-display/148125/2](#)

Atualizador de Firmware WiFi101 / WiFiNINA

Enviar certificado Raiz SSL

2 níveis, preto e branco, talvez use dithering

Placa: "BBC micro:bit"

Gerenciador de Placas...

Ctrl+Shift+B

Porta

Arduino AVR Boards

Obter Informações da Placa

Nordic Semiconductor nRF5 Boards

Softdevice: "S110"

SparkFun AVR Boards

Programador/Gravador

PROGRAM

Gravar bootloader

sparkFun Redboard

SparkFun MaKey MaKey

sparkFun Pro micro

SparkFun Fio v3

Qduino Mini

SparkFun Digital Sandbox

SparkFun Mega Pro

SparkFun RedBot

SparkFun Serial 7-Segment Display

SparkFun ATmega128RFA1 Dev Board

LilyPad USB Plus

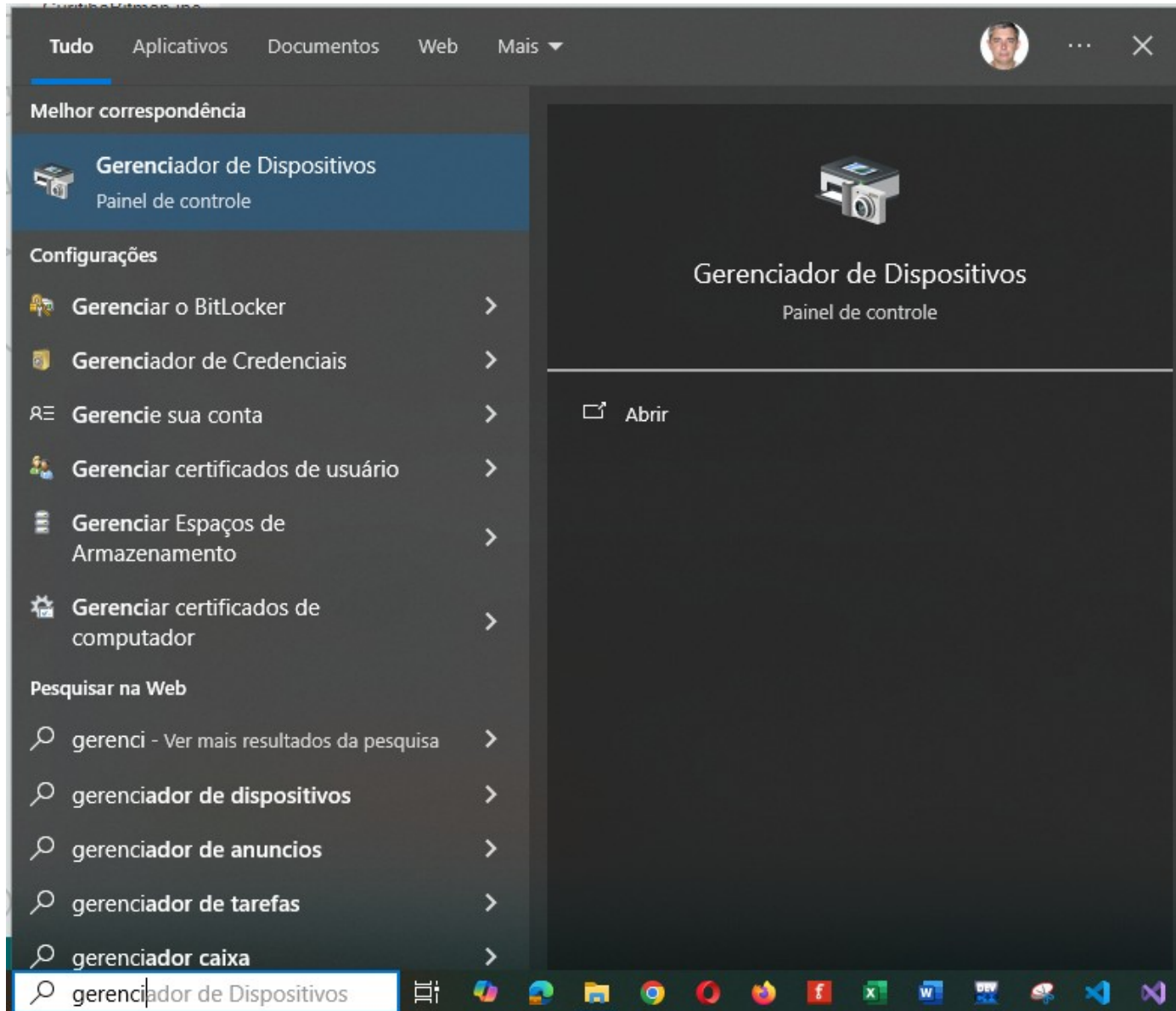
SparkFun SerLCD

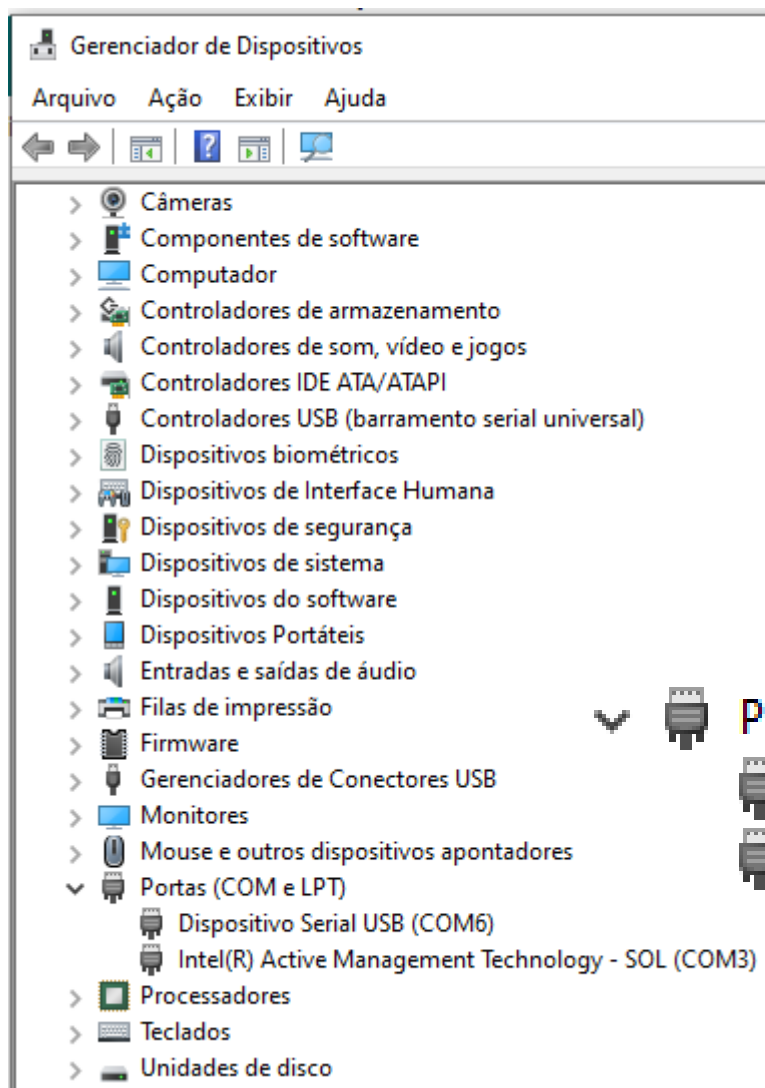
```
#include "U8glib.h" //biblioteca padrão do Google
```

```
GLIB SSD1306 128X64 u8g(U8G_I2C_OPT_NONE); // conexão do display
```

Arduino UNO SDA = A4 SCL = A5

```
efine curitiba_img_width 128
```





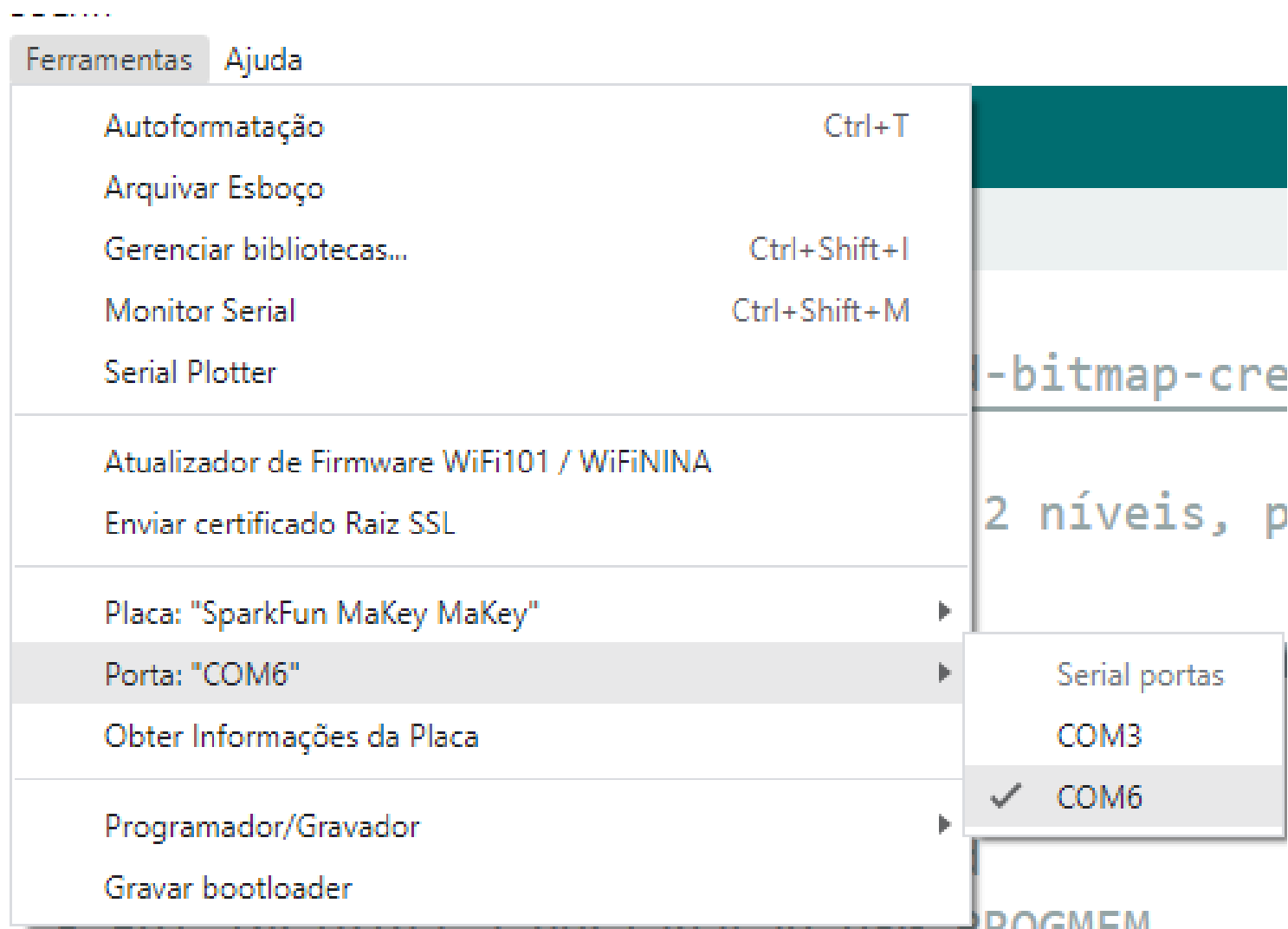
Portas (COM e LPT)



Dispositivo Serial USB (COM6)



Intel(R) Active Management Technology - SOL (COM3)



Instalar Microbit

Instalando placas – Microbit/ IdeaBit



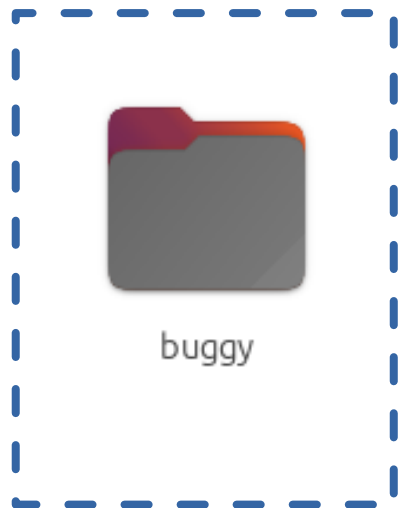
(Interface antiga)



Piscar LEDs
da base *Buggy*



Arquivo com o código



buggy



CuritibaBitmap



faixaDeLEDS



temperaturaDisplay



temperaturaDisplayPaginasImagem

Arquivo com o código





SetColorLED

/*

cores predefinidas

#define COLOR_NONE

#define COLOR_BLACK

#define COLOR_WHITE

#define COLOR_RED

#define COLOR_ORANGE

#define COLOR_YELLOW

#define COLOR_GREEN

#define COLOR_CYAN

#define COLOR_BLUE

#define COLOR_PURPLE

#define COLOR_WARM

#define COLOR_COLD

#define BLACK 0

#define RED 1

#define GREEN 2

#define BLUE 3

*/

Autoformatação

Ctrl+T

Arquivar Sketch

Corrigir codificação e recarregar

Monitor serial

Ctrl+Shift+M

Plotter serial

Ctrl+Shift+L

WiFi101 Firmware Updater

ArduBlock

Placa: "Microduino/mCookie-device"

Processador: "Microduino/mCookie-Core+ (644pa)@16M,5V"

Porta

Obter informações da Placa

Programador: "BusPirate as ISP"

Gravar Bootloader

Microduino/mCookie-Core (328p)@16M,5V

Microduino/mCookie-Core (328p)@8M,3.3V

• Microduino/mCookie-Core+ (644pa)@16M,5V

Microduino/mCookie-Core+ (644pa)@8M,3.3V

Microduino/mCookie-Core+ (1284pa)@16M,5V

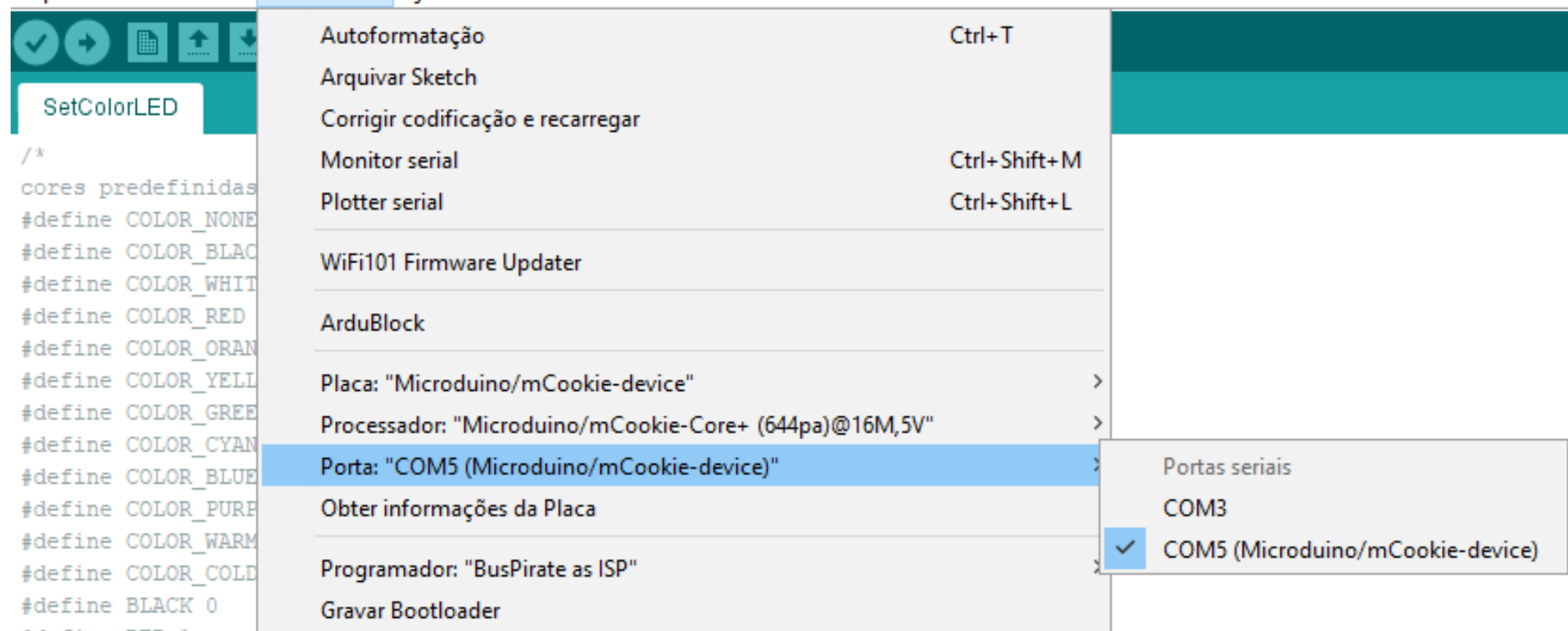
Microduino/mCookie-Core+ (1284pa)@8M,3.3V

Microduino/mCookie-Core USB (32u4)

Microduino/mCookie-Core RF (128rfa1)

SetColorLED | Arduino 1.8.4

Arquivo Editar Sketch Ferramentas Ajuda



The screenshot shows the Arduino IDE interface with the 'Ferramentas' (Tools) menu open. The menu items are as follows:

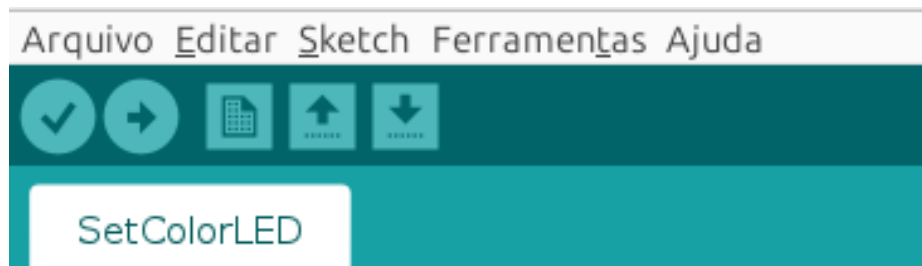
- Autoformatação (Ctrl+T)
- Arquivar Sketch
- Corrigir codificação e recarregar
- Monitor serial (Ctrl+Shift+M)
- Plotter serial (Ctrl+Shift+L)
- WiFi101 Firmware Updater
- ArduBlock
- Placa: "Microduino/mCookie-device" >
- Processador: "Microduino/mCookie-Core+ (644pa)@16M,5V" >
- Porta: "COM5 (Microduino/mCookie-device)" >
- Obter informações da Placa
- Programador: "BusPirate as ISP" >
- Gravar Bootloader

The 'Porta: "COM5 (Microduino/mCookie-device)"' option is selected, and a submenu is open showing the following options:

- Portas seriais
- COM3
- COM5 (Microduino/mCookie-device) (checked)

The code editor on the left shows the following code:

```
/*
cores predefinidas
#define COLOR_NONE
#define COLOR_BLACK
#define COLOR_WHITE
#define COLOR_RED
#define COLOR_ORANGE
#define COLOR_YELLOW
#define COLOR_GREEN
#define COLOR_CYAN
#define COLOR_BLUE
#define COLOR_PURPLE
#define COLOR_WARM
#define COLOR_COLD
#define BLACK 0
#define RED 1
```

```
1 /*
2 cores predefinidas
3 #define COLOR_NONE 0X000000
4 #define COLOR_BLACK 0X000000
5 #define COLOR_WHITE 0XFFFFFF
6 #define COLOR_RED 0XFF0000
7 #define COLOR_ORANGE 0XEA8B17
8 #define COLOR_YELLOW 0XFFFF00
9 #define COLOR_GREEN 0X00FF00
10 #define COLOR_CYAN 0X00FFFF
11 #define COLOR_BLUE 0X0000FF
12 #define COLOR_PURPLE 0X4B0082
13 #define COLOR_WARM 0X9B6400
14 #define COLOR_COLD 0X648264
15 #define BLACK 0
16 #define RED 1
17 #define GREEN 2
18 #define BLUE 3
19 */
```

Nomes / códigos
de cores pré-
definidas

```
21 #include <BittyBuggy.h>  
22 #define mySerial Serial1  
23 BittyBuggy car(&mySerial);
```

SE você usar o compilador do
Microduino funciona direto, pois a
biblioteca “BittyBuggy” já veio junto.

SENÃO, tem que instalar a biblioteca.

fatal error: BittyBuggy.h: No such file or directory

#include <BittyBuggy.h>

^~~~~~

compilation terminated.

exit status 1

BittyBuggy.h: No such file or directory

```
21 #include <BittyBuggy.h>  
22 #define mySerial Serial1  
23 BittyBuggy car(&mySerial);
```



```
#include "BittyBuggy.h"  
#define mySerial Serial1  
BittyBuggy car(&mySerial);
```

(ou, você pode adicionar os arquivos necessários na mesma pasta, mas neste caso deverá trocar “< e >” por aspas duplas “ ”)

SetColorLED §

BittyBuggy.cpp

BittyBuggy.h

```
1 #include "BittyBuggy.h"  
2 #define mySerial Serial1  
3 BittyBuggy car(&mySerial);
```

A vantagem é que não precisa instalar outro ambiente ou outra biblioteca.

A desvantagem é que se a biblioteca for atualizada seu código não será; além disso, a cada nova compilação você terá que colocar vários arquivos na pasta de código – e poderá ter versões diferentes...

```
25 void setup()  
26 {  
27     Serial.begin(57600);  
28     while (!car.begin())  
29         ;  
30     car.setColorLEDA(0X9B6400);  
31     car.setColorLEDB(0X648264);  
32     delay(1000);  
33     car.setColorLEDA(0XEA8B17);  
34     car.setColorLEDB(0XEA8B17);  
35     randomSeed(analogRead(2));  
36     delay(1000);  
37 }
```

```
39 void loop()
40 {
41   car.setColorLEDA(255, 0, 0);
42   car.setColorLEDB(0, 255, 0);
43   delay(800);
44   car.setAllLED(255, 255, 0); //0xFFFF00 = amarelo
45   delay(1500);
46   car.setAllLED(BLACK);
47   delay(800);
48   car.setAllLED(COLOR_PURPLE);
49   delay(1500);
50   car.setAllLED(COLOR_BLUE);
51   delay(500);
52   car.setAllLED(COLOR_YELLOW);
53   delay(500);
54
55   //cores aleatorias
56   car.setColorLEDB(random(0, 256), random(0, 256), random(0, 256));
57   delay(800);
58   car.setColorLEDA(random(0, 256), random(0, 256), random(0, 256));
59   delay(800);
60   car.setAllLED(random(0, 256), random(0, 256), random(0, 256));
61   delay(800);
62 }
```

Alerta – pode prosseguir normalmente

Carregado.

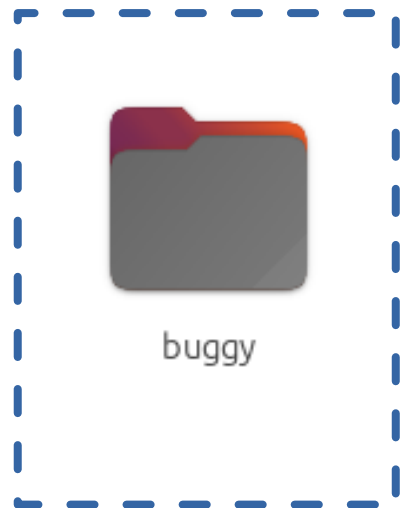
Warning: Board Microduino:avr:mddevice doesn't define a 'build.board' preference. Auto-set to: AVR_MDDEVICE

O sketch usa 5608 bytes (8%) de espaço de armazenamento para programas. O máximo são 64512 bytes.

Variáveis globais usam 443 bytes (10%) de memória dinâmica, deixando 3653 bytes para variáveis locais. O máximo são 4096 bytes.

Seguidor de linha

Arquivo com o código



buggy



CuritibaBitmap



faixaDeLEDS



temperaturaDisplay



temperaturaDisplayPaginasImagem

Arquivo com o código



```
1 #include <BittyBuggy.h>
```

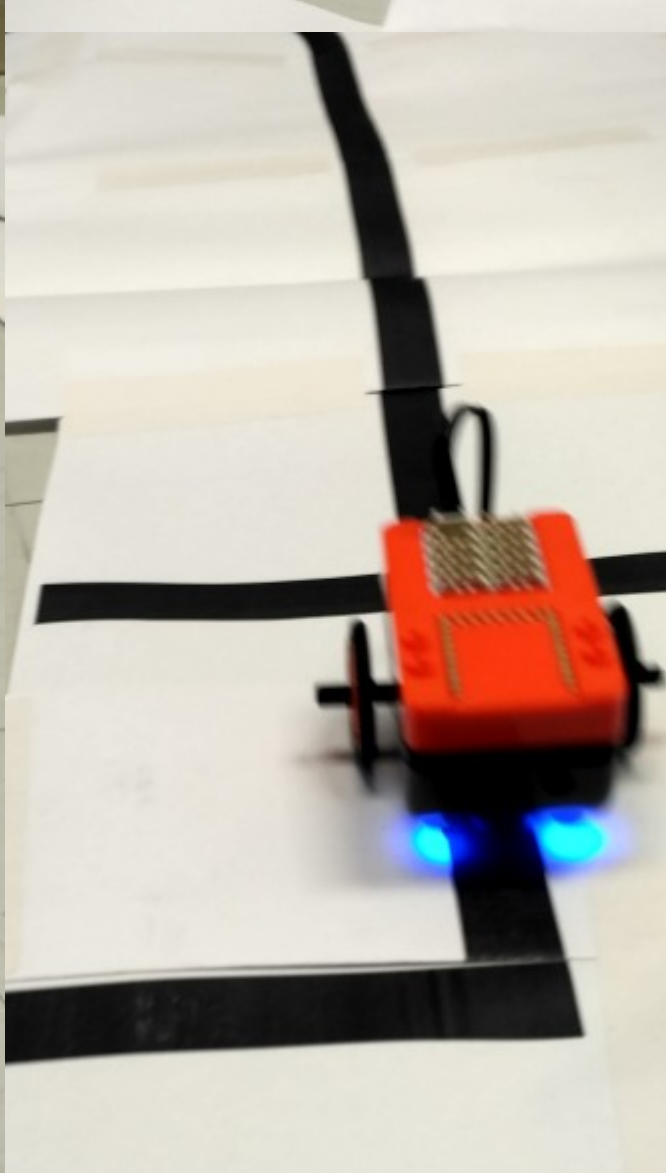
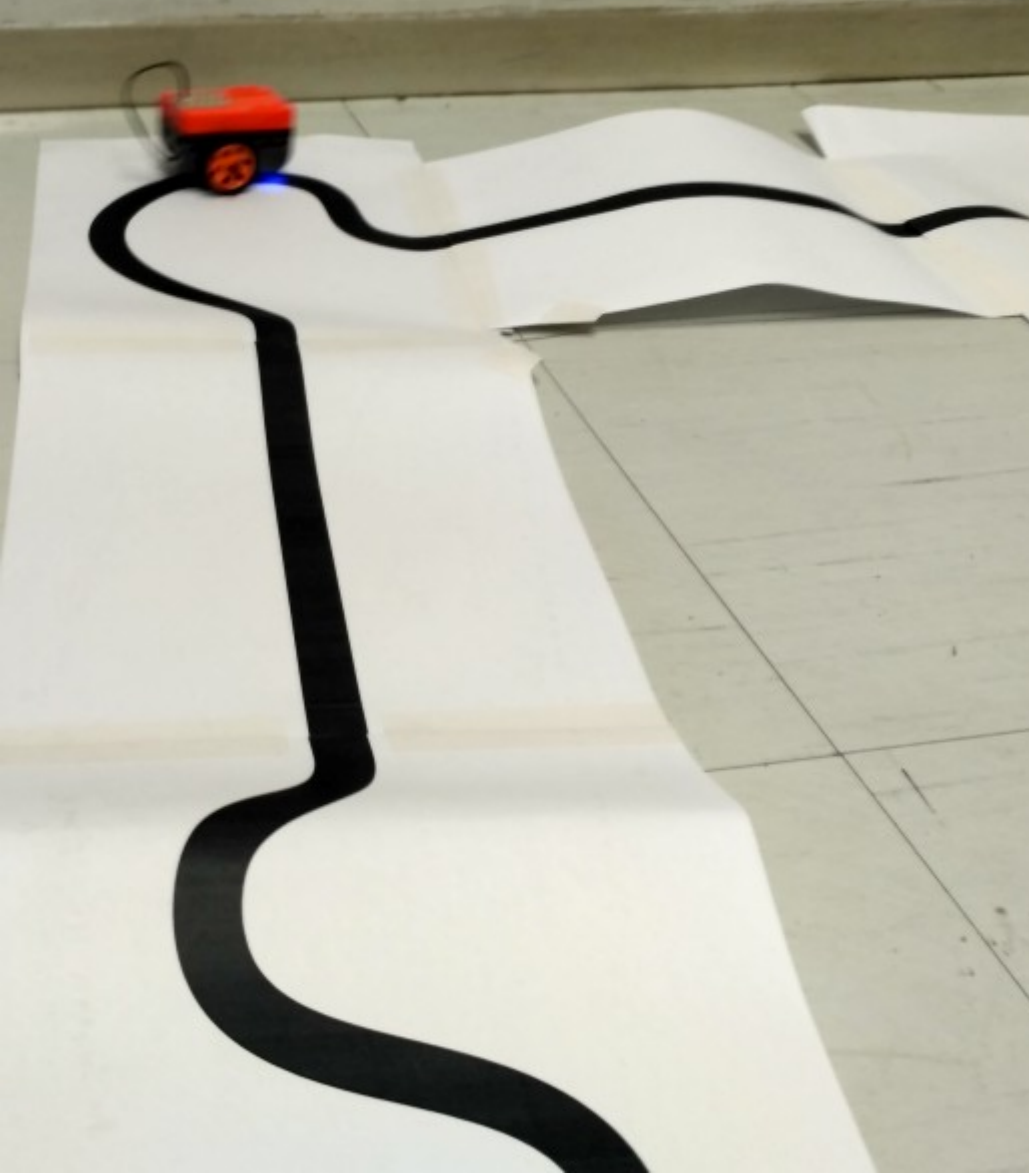
```
1 #include <BittyBuggy.h>
```

CarrinhoSeguidor

BittyBuggy.cpp

BittyBuggy.h

```
1 #include "BittyBuggy.h"
```



```
27 void trackControl(uint8_t trackVal1, uint8_t trackVal2)
28 {
29     if (trackVal1 < 135 && trackVal2 < 135) {
30         carspeed[0] = 0;
31         carspeed[1] = 0;
32     }
33     else {
34         carspeed[0] = 220 + (trackVal1 - 240) * 2; //白248 黒 130 branco, preto
35         carspeed[1] = 220 + (trackVal2 - 240) * 2;
36     }
37     /*
38     if (trackVal1 > 200 && trackVal2 > 200) {
39         car.setSpeed(0, 0);
40         delay(300);
41         carspeed[0] = 0;
42         carspeed[1] = 0;
43     }
44     else {
45         carspeed[0] = 180 - (trackVal1 - 150) * 2;
46         carspeed[1] = 180 - (trackVal2 - 150) * 2;
47     }
48     */
49 }
```

Cortesia do Google...

Experimente comentar e
bloco de cima e
'descomentar' o de
baixo →

```
27 void trackControl(uint8_t trackVal1, uint8_t trackVal2)
28 {
29     /*
30     if (trackVal1 < 135 && trackVal2 < 135) {
31         carspeed[0] = 0;
32         carspeed[1] = 0;
33     }
34     else {
35         carspeed[0] = 220 + (trackVal1 - 240) * 2; //白248 黒 130 branco, preto
36         carspeed[1] = 220 + (trackVal2 - 240) * 2;
37     }
38     */
39     if (trackVal1 > 200 && trackVal2 > 200) {
40         car.setSpeed(0, 0);
41         delay(300);
42         carspeed[0] = 0;
43         carspeed[1] = 0;
44     }
45     else {
46         carspeed[0] = 180 - (trackVal1 - 150) * 2;
47         carspeed[1] = 180 - (trackVal2 - 150) * 2;
48     }
```

Cortesia do Google...

Teste e veja a diferença

Até a próxima

