



Projeto 02

Coisas Replicadas – Teoria

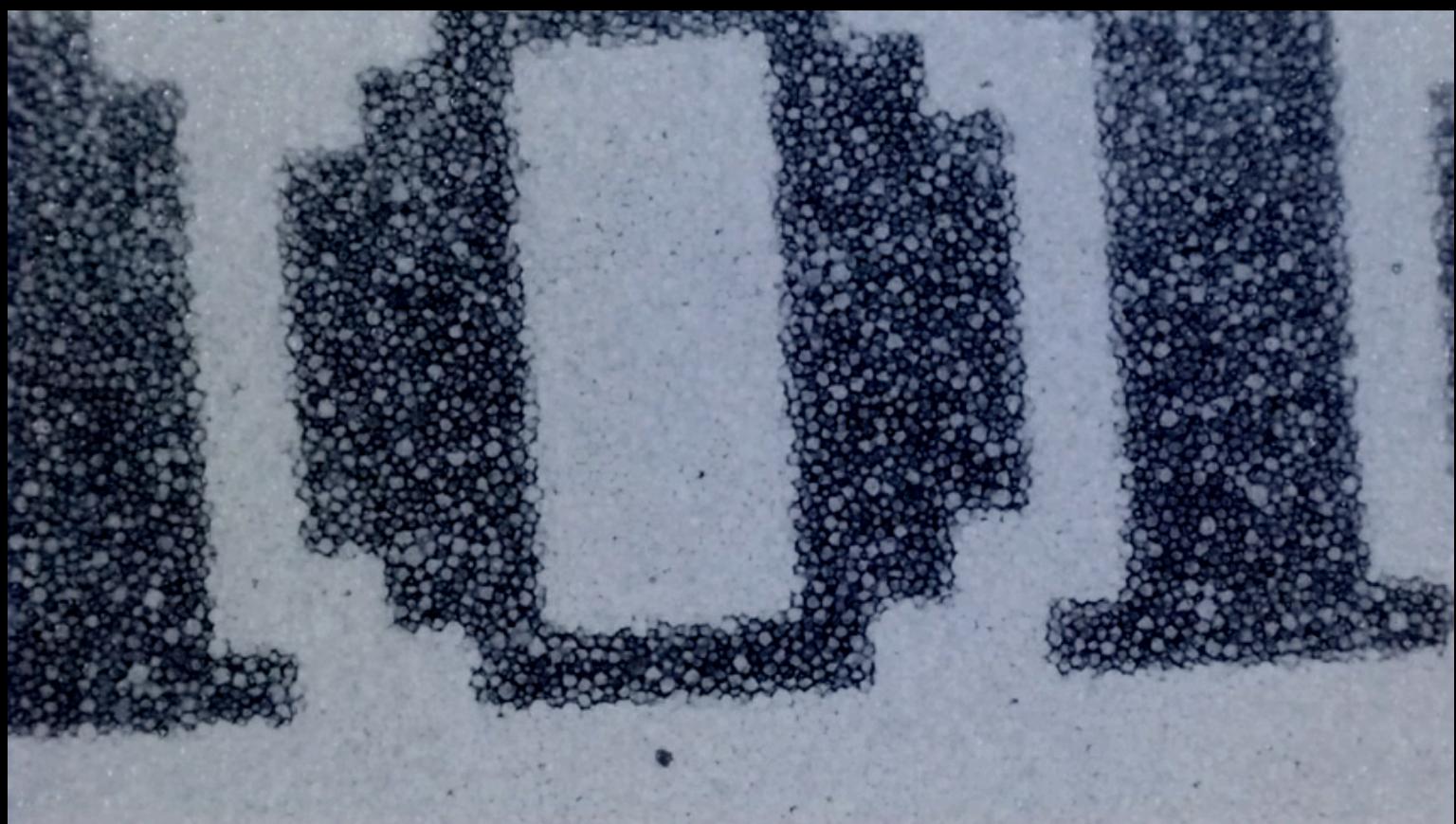
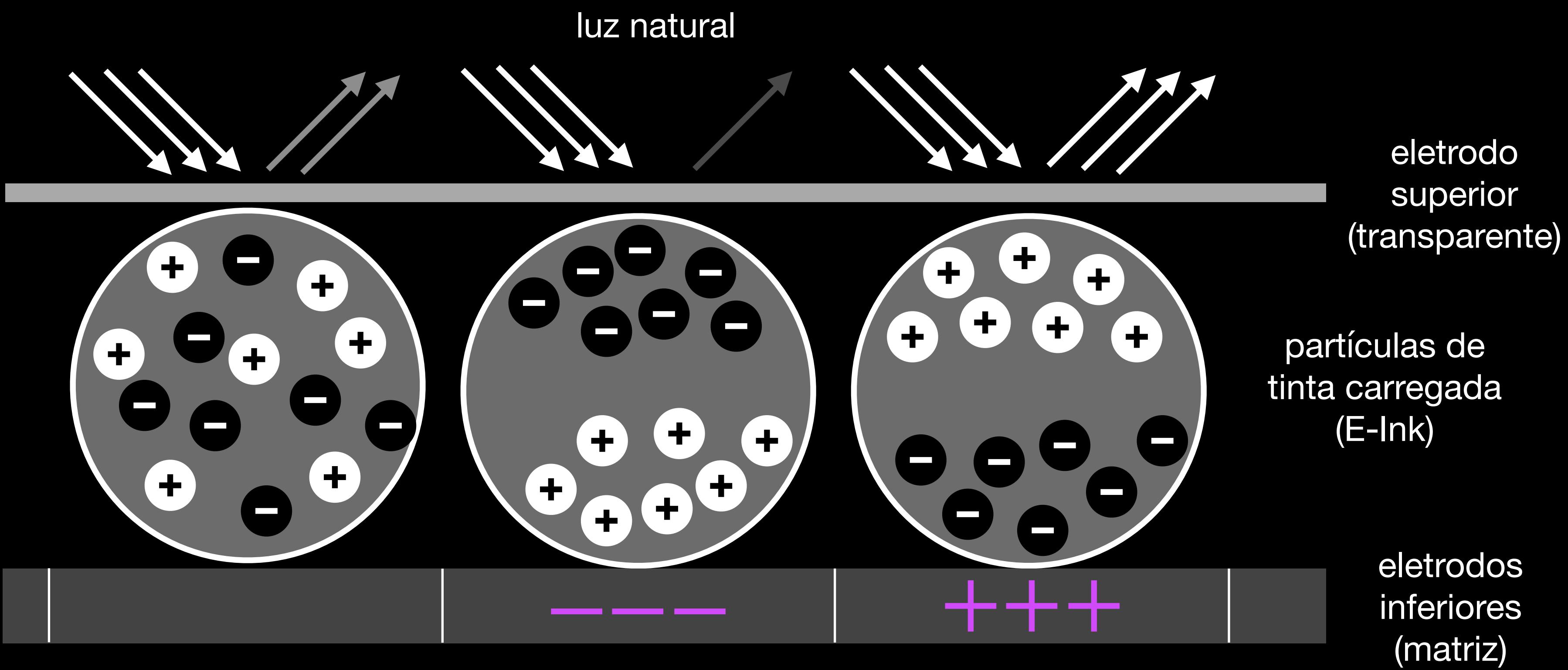
Jan K. S. – janks@puc-rio.br

ENG4051 – Projeto Internet das Coisas

Coisas







Projeto de IoT
2024

atualização
PARCIAL

Projeto de IoT
Olá!
2024

rápida
deixa vestígios

Projeto de IoT
2024

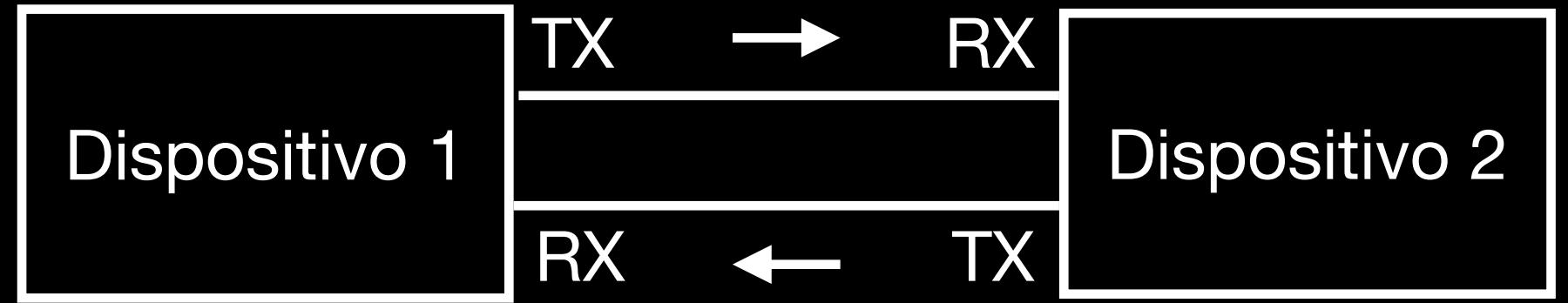
atualização
TOTAL

Olá!

lenta
pisca a tela
limpa tudo

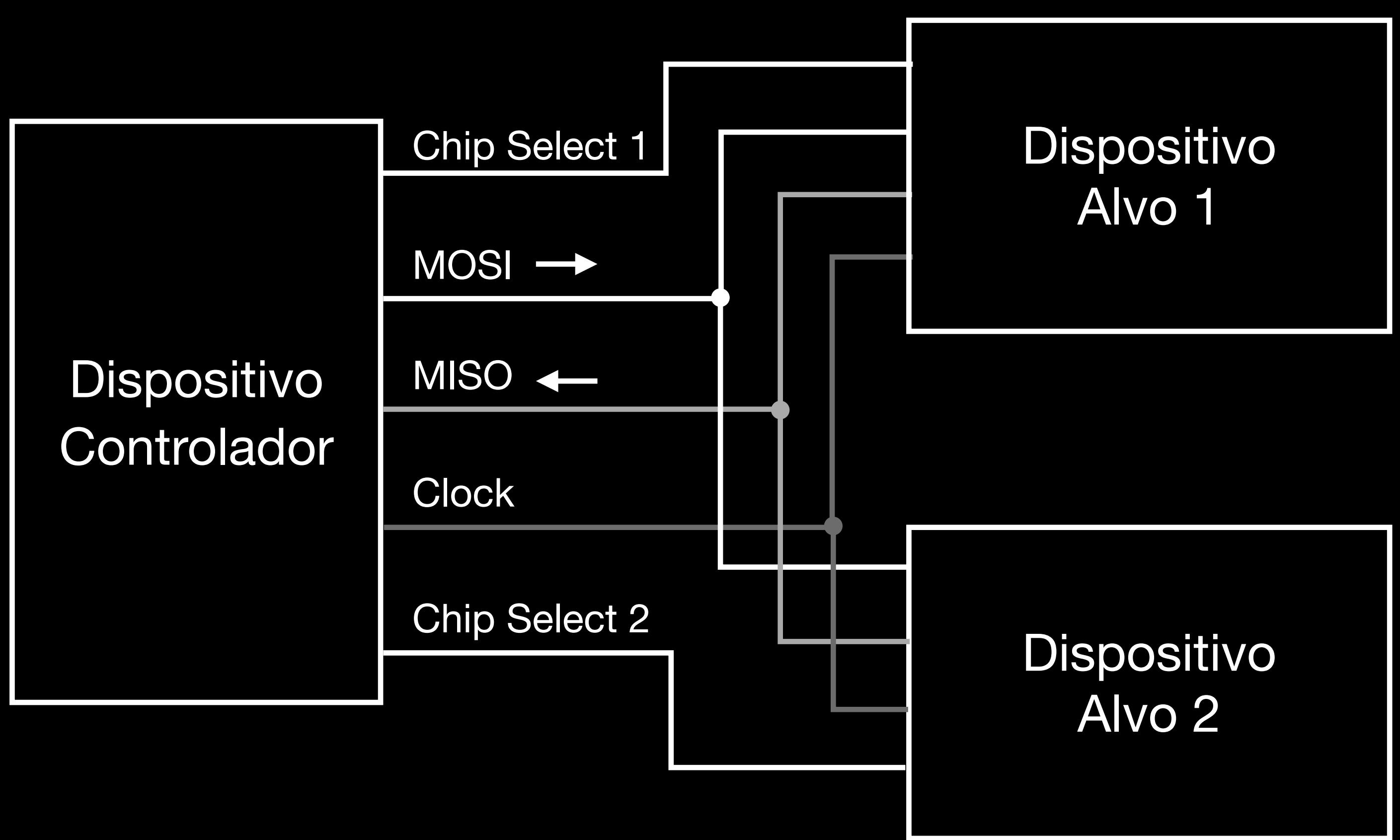
UART

Universal Asynchronous
Receiver/Transmitter



SPI

Serial
Peripheral
Interface

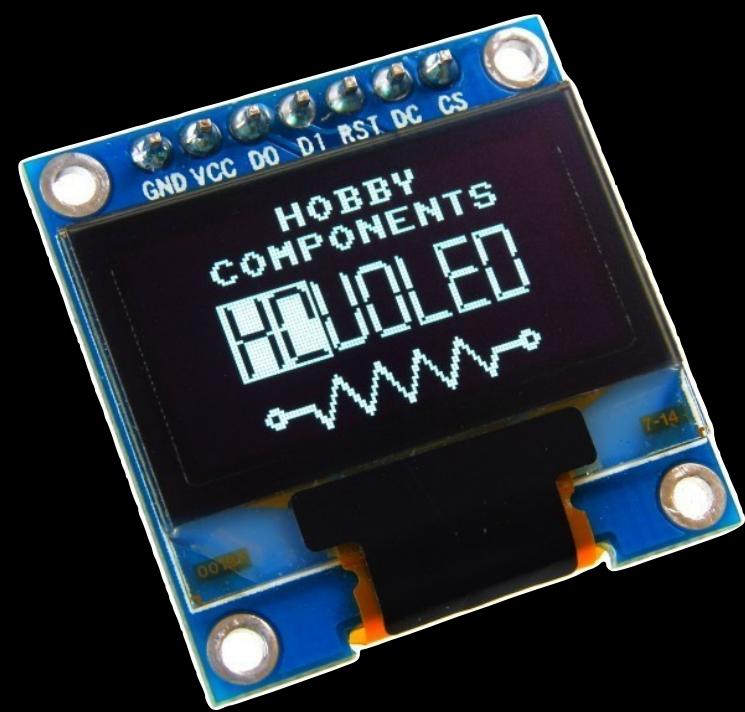




Pino Display WeAct E-Paper

VCC	3.3V
GND	GND
SDA (MOSI)	11
SCL (Clock)	12
CS (Chip Select)	10
D/C (Data/Command)	14
RES (Reset)	15
BUSY	16

} pinos padrão
do SPI no
ESP32 S3 CAM



Adafruit_SSD1306

Adafruit_GFX

TFT_eSPI

Adafruit_ILI9341

MCUFRIEND_kbv



github.com/ZinggJM/GxEPD2

GxEPD2

Arduino Display Library for SPI E-Paper Displays

- With full Graphics and Text support using Adafruit_GFX
- For SPI e-paper displays from Dalian Good Display
- and SPI e-paper boards from Waveshare

important note:

- the display panels are for 3.3V supply and 3.3V data lines
- never connect data lines directly to 5V Arduino data pins, 4k7/10k resistor divider
- series resistor only is not enough for reliable operation (back effect through protection diodes)
- 4k7/10k resistor divider may not work with flat cable extension Waveshare 4.2 board, use level converter then
- do not forget to connect GND

+ 59 releases

Packages

No packages published

Contributors 3

ZinaaJM

sketch_mar2a | Arduino IDE 2.3.4

LIBRARY MANAGER

gxepd2

Type: All

Topic: All

GxEPD2 by Jean-Marc Zingg

1.6.2 installed

Arduino Display Library for SPI E-Paper displays from Dalian Good Display and Waveshare. Requires HW SPI and...

More info

1.6.2

INSTALL

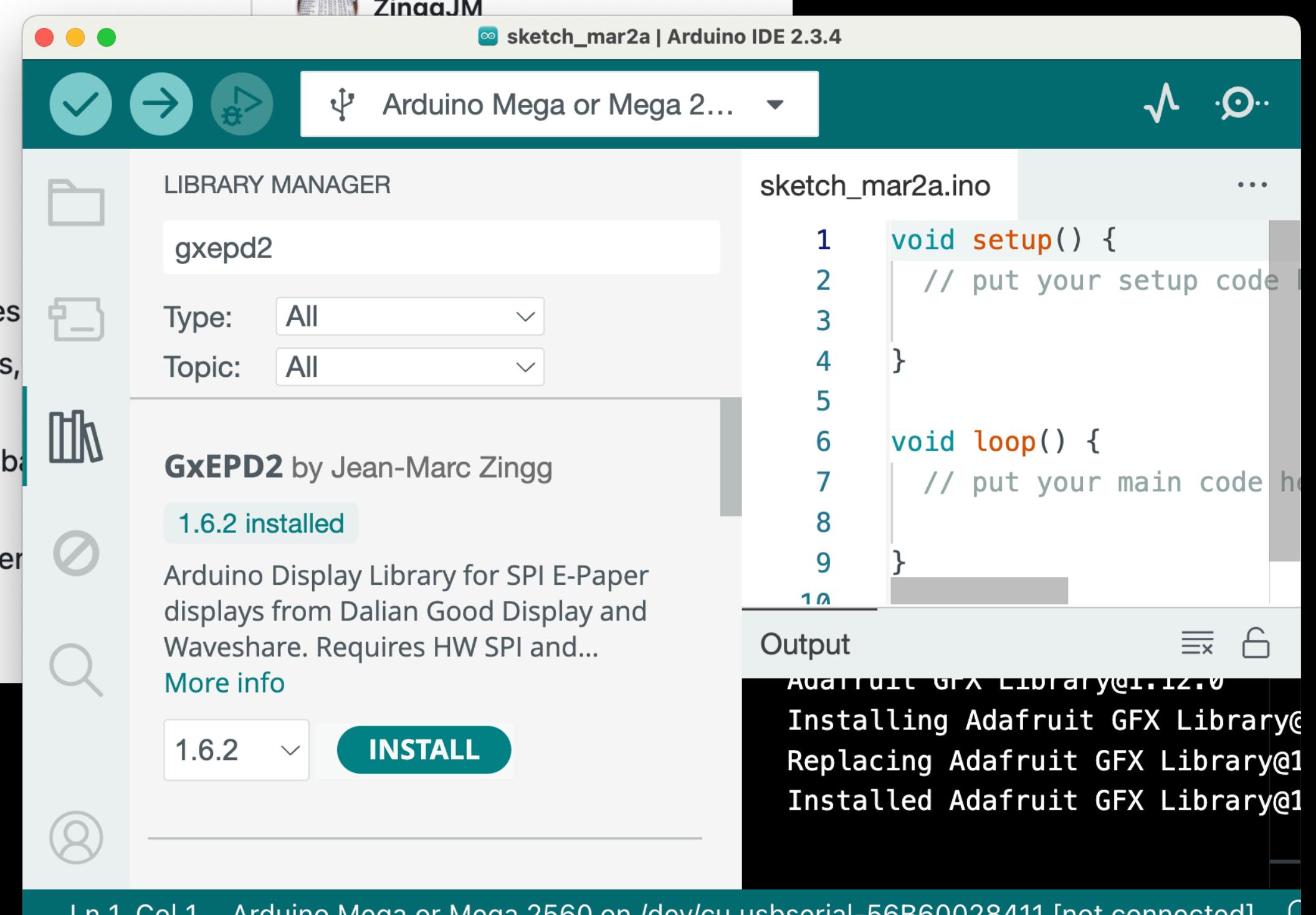
sketch_mar2a.ino

```
void setup() {  
    // put your setup code here  
}  
  
void loop() {  
    // put your main code here  
}
```

Output

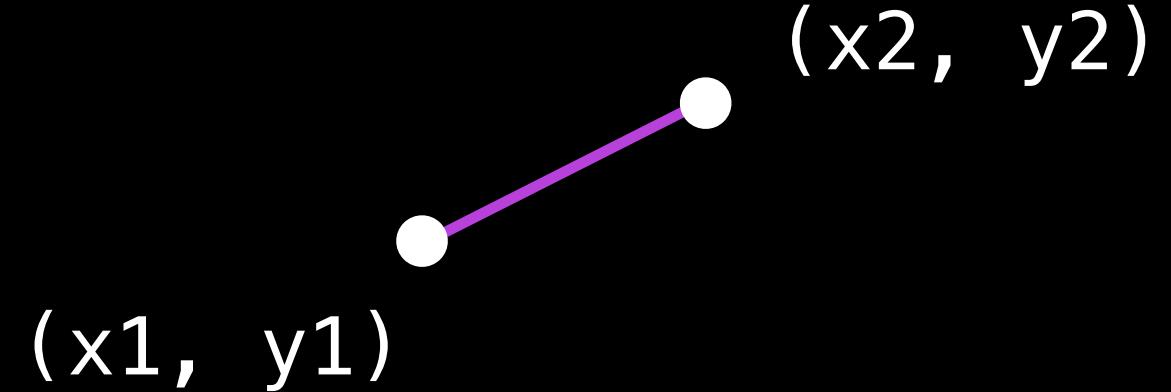
```
Adafruit_GFX Library@1.12.0  
Installing Adafruit GFX Library@1.12.0  
Replacing Adafruit GFX Library@1.12.0  
Installed Adafruit GFX Library@1.12.0
```

Ln 1, Col 1 Arduino Mega or Mega 2560 on /dev/cu.usbserial-56B60028411 [not connected]

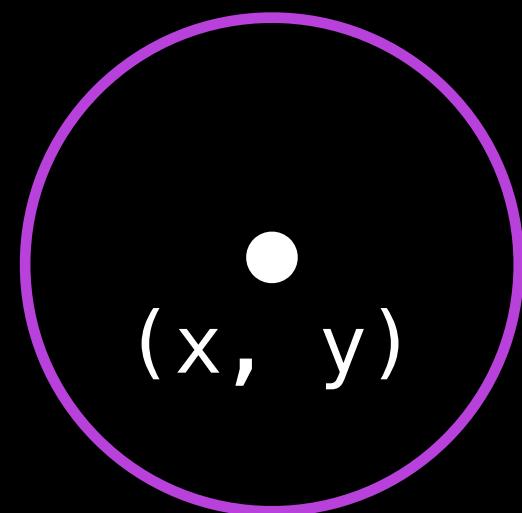


Adafruit_GFX

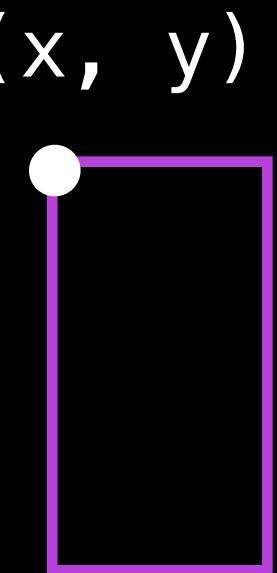
```
tela.drawLine(x1, y1, x2, y2, cor);
```



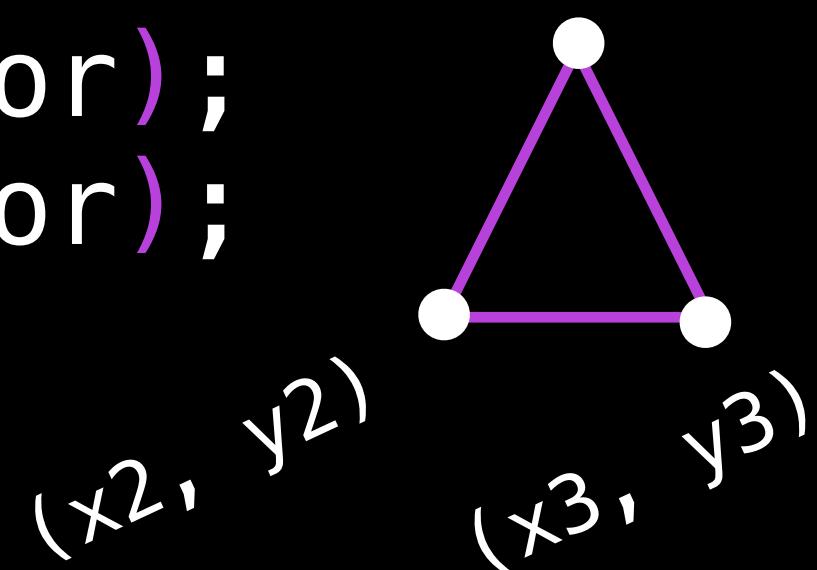
```
tela.fillCircle(x, y, raio, cor);  
tela.drawCircle(x, y, raio, cor);
```



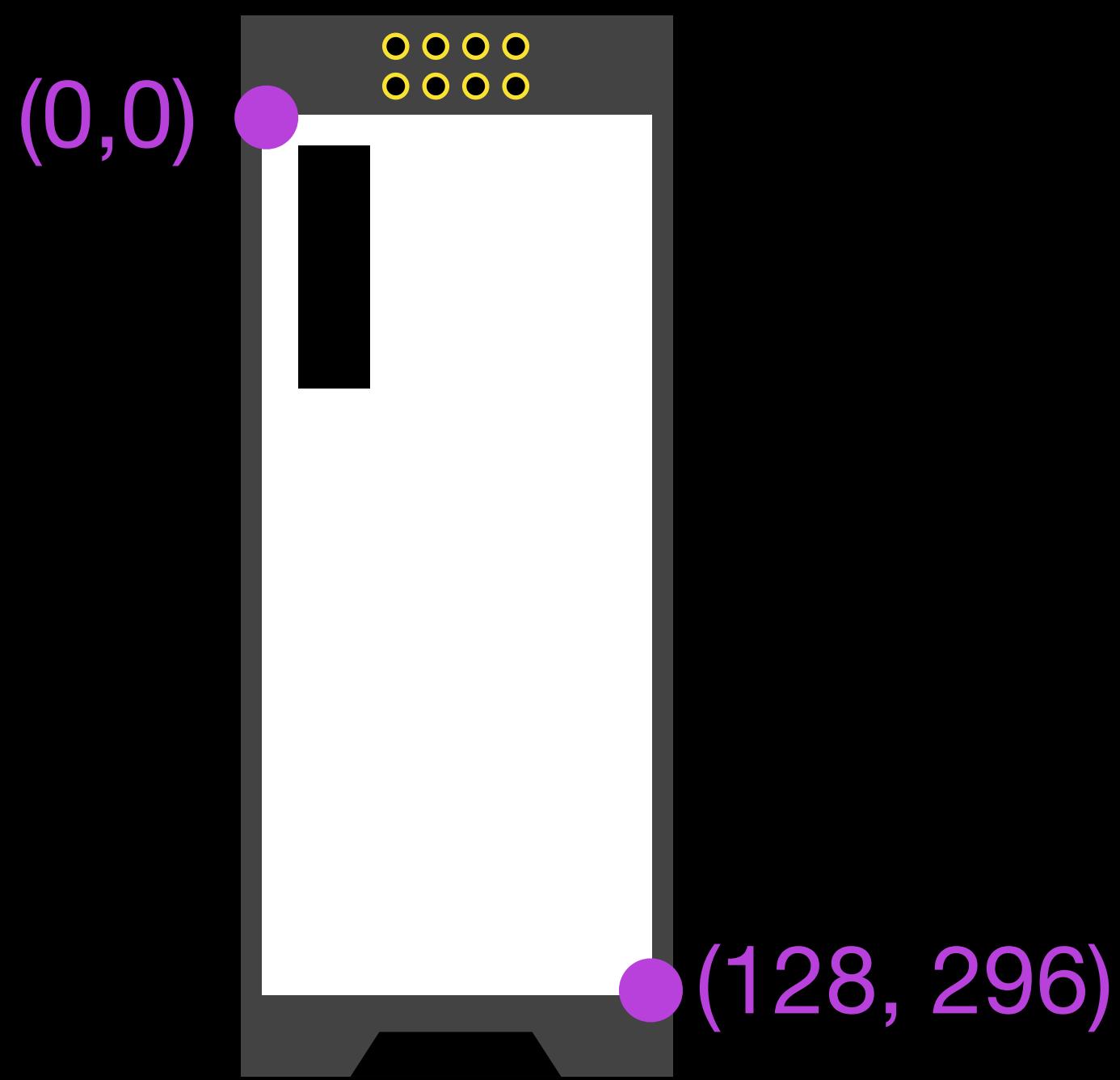
```
tela.fillRect(x, y, comprimento, altura, cor);  
tela.drawRect(x, y, comprimento, altura, cor);
```



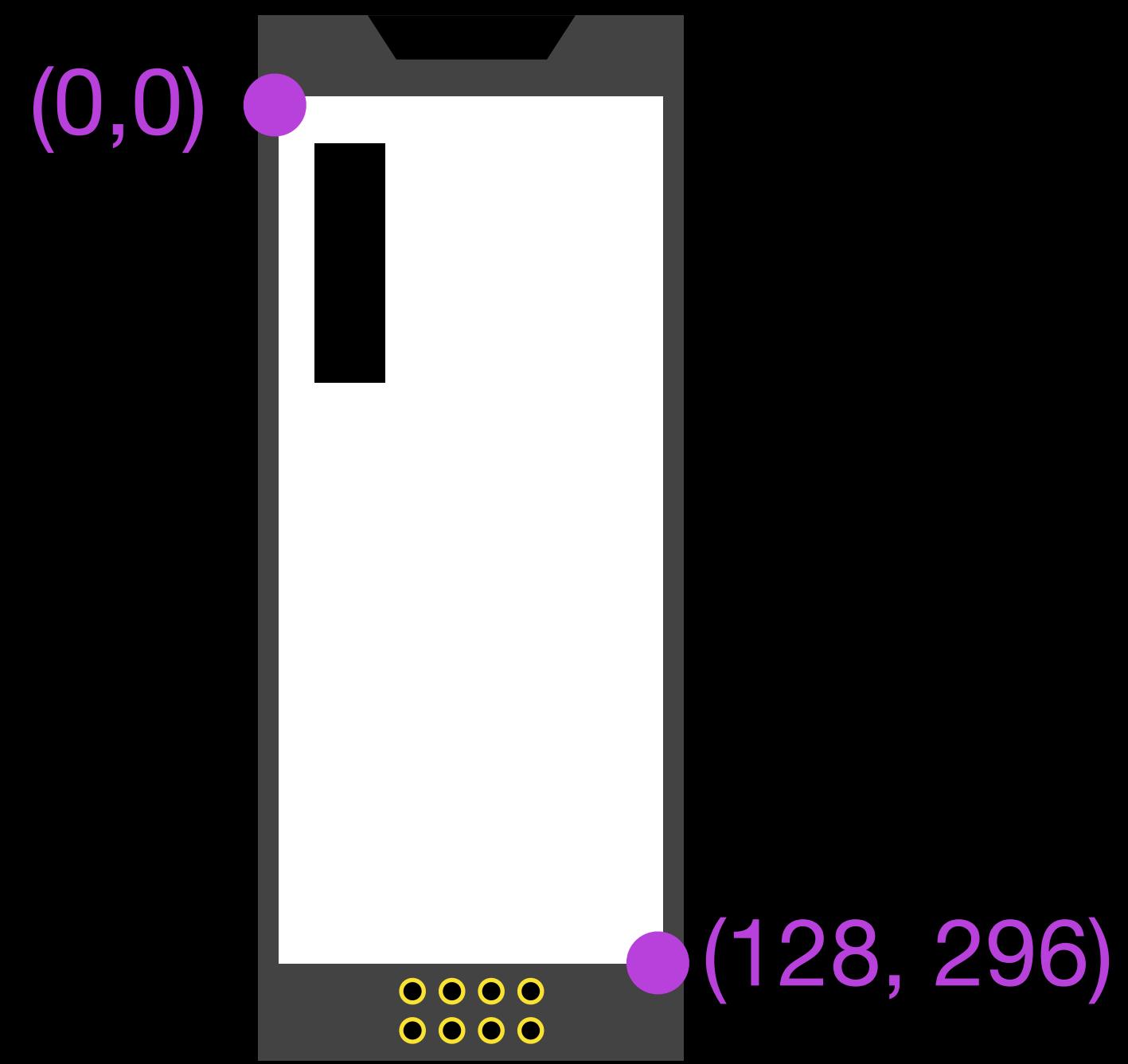
```
tela.fillTriangle(x1, y1, x2, y2, x3, y3, cor);  
tela.drawTriangle(x1, y1, x2, y2, x3, y3, cor);
```



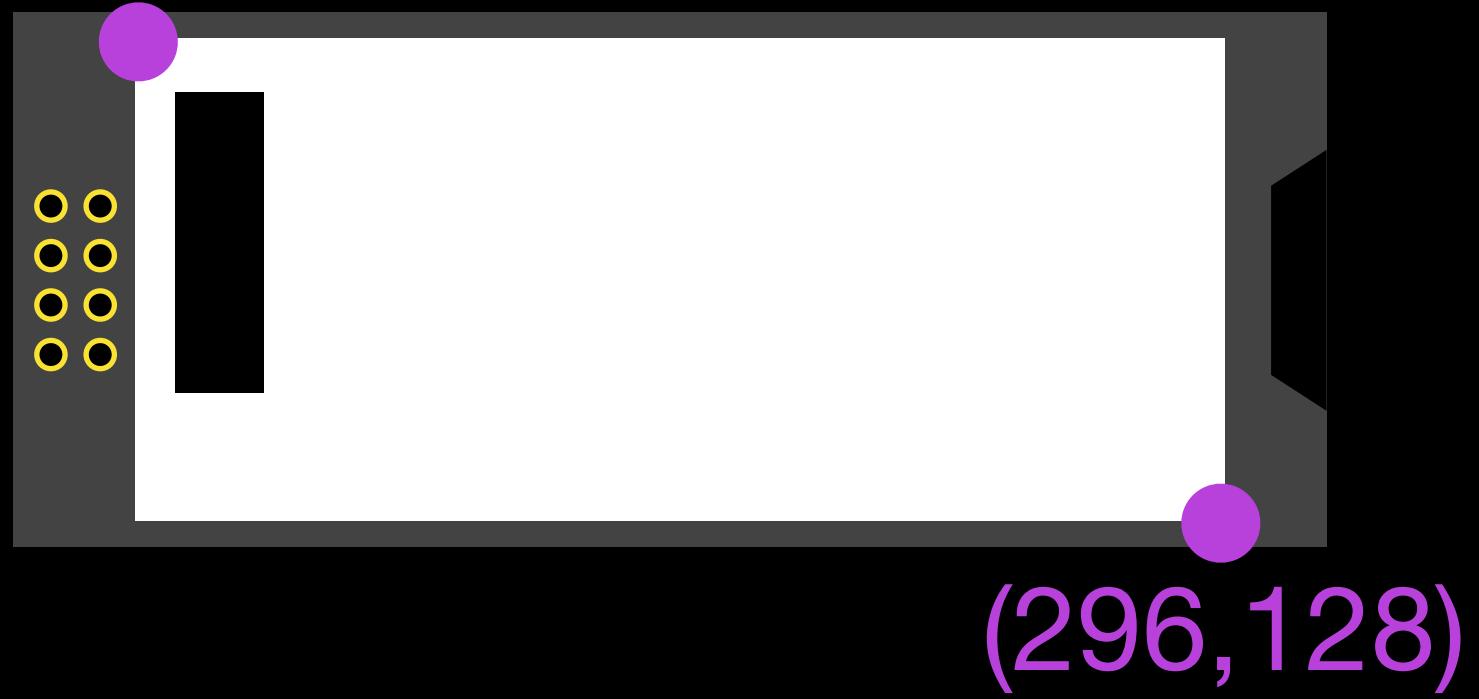
`tela.setRotation(0);`



`tela.setRotation(2);`



`(0,0) tela.setRotation(1);`



`(0,0) tela.setRotation(3);`



```

#include <GxEPD2_BW.h>

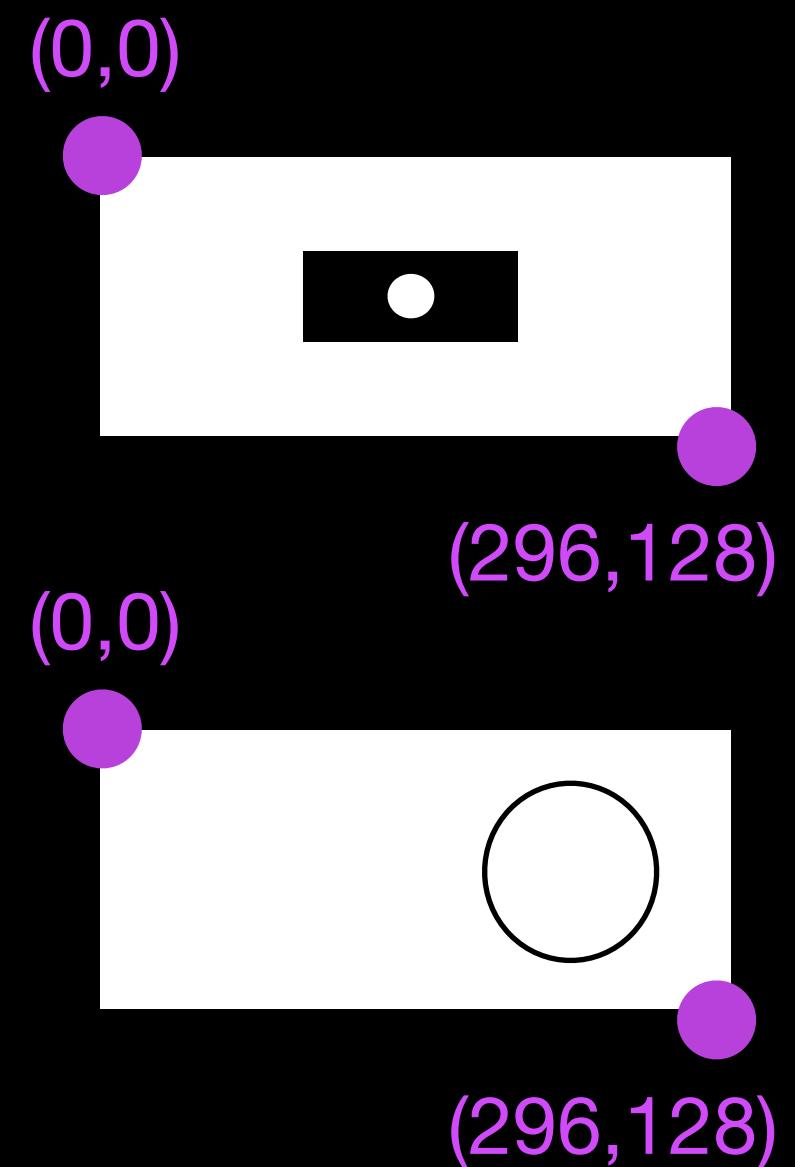
GxEPD2_290_T94_V2 modeloTela(10, 14, 15, 16);
GxEPD2_BW<GxEPD2_290_T94_V2, GxEPD2_290_T94_V2::HEIGHT>
tela(modeloTela);

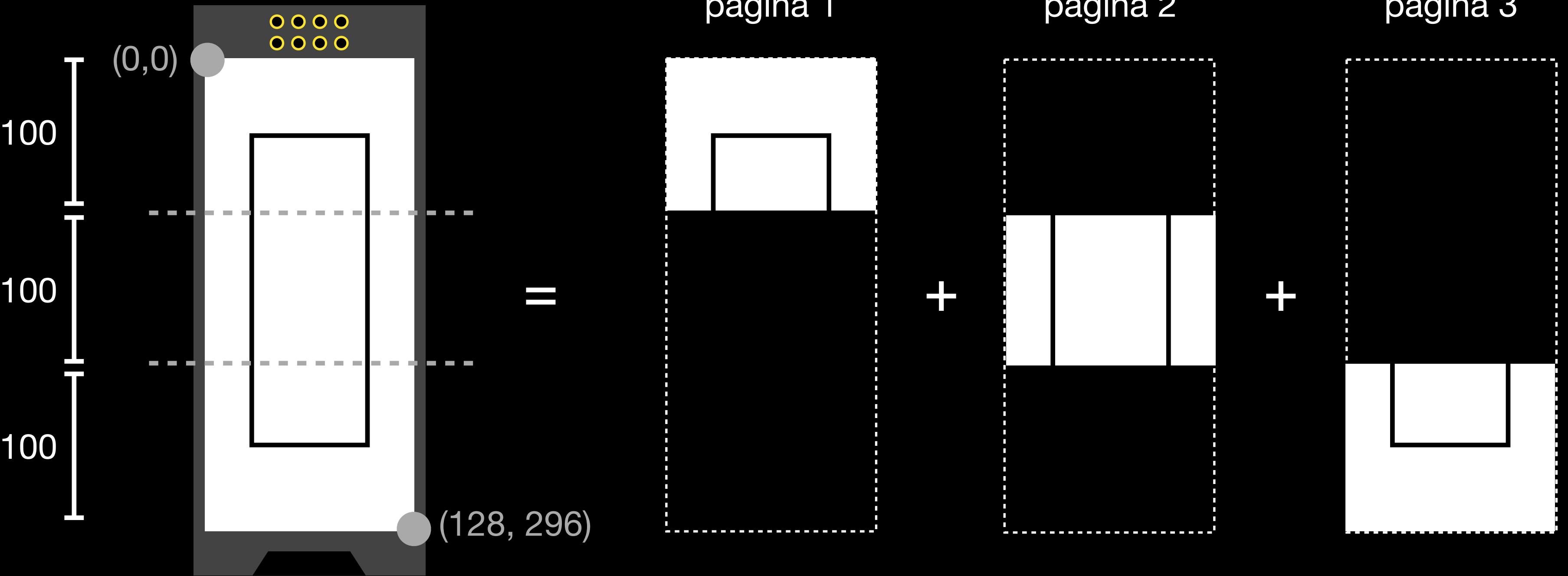
void setup() {
    tela.init();
    tela.setRotation(3); // tela deitada (modo paisagem)
    tela.fillScreen(GxEPD_WHITE);
    tela.display(true); // atualiza tela
}

        // em alguma função do código...
    tela.fillRect(100, 50, 96, 28, GxEPD_BLACK);
    tela.fillCircle(148, 64, 10, GxEPD_WHITE);
    tela.display(true); // atualiza tela

    tela.fillScreen(GxEPD_WHITE);
    tela.drawCircle(220, 64, 40, GxEPD_BLACK);
    tela.display(true); // atualiza tela

```





```
GxEPD2_BW<GxEPD2_290_T94_V2, 100> tela(modeloTela);
// ...
tela.firstPage();
do {
    tela.fillRect(30, 50, 70, 200, GxEPD_BLACK);
}
while (tela.nextPage());
```

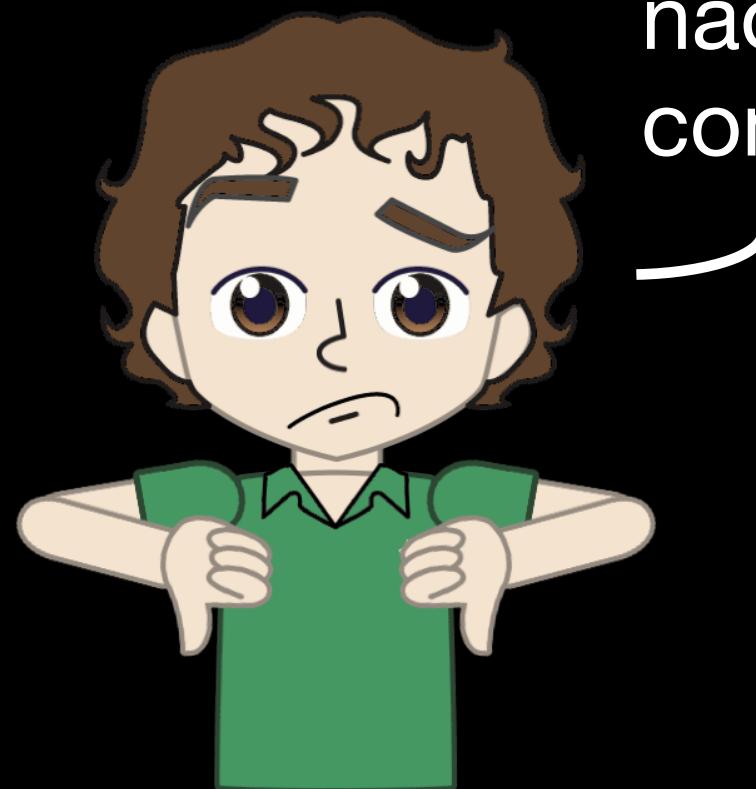
Dá para desenhar e enviar os pixels à tela aos poucos, para economizar memória RAM.

Mas **não precisamos fazer isso** porque a tela é pequena e o ESP32 tem bastante RAM.



```
tela.setCursor(20, 40);  
tela.setTextColor(CXEPD_BLACK);  
tela.setTextSize(5);  
tela.print("Acentuação");
```

Acentua-ção



Fontes da Adafruit_GFX
não aceitam caracteres
com acento.

github.com/olikraus/U8g2_for_Adafruit_GFX

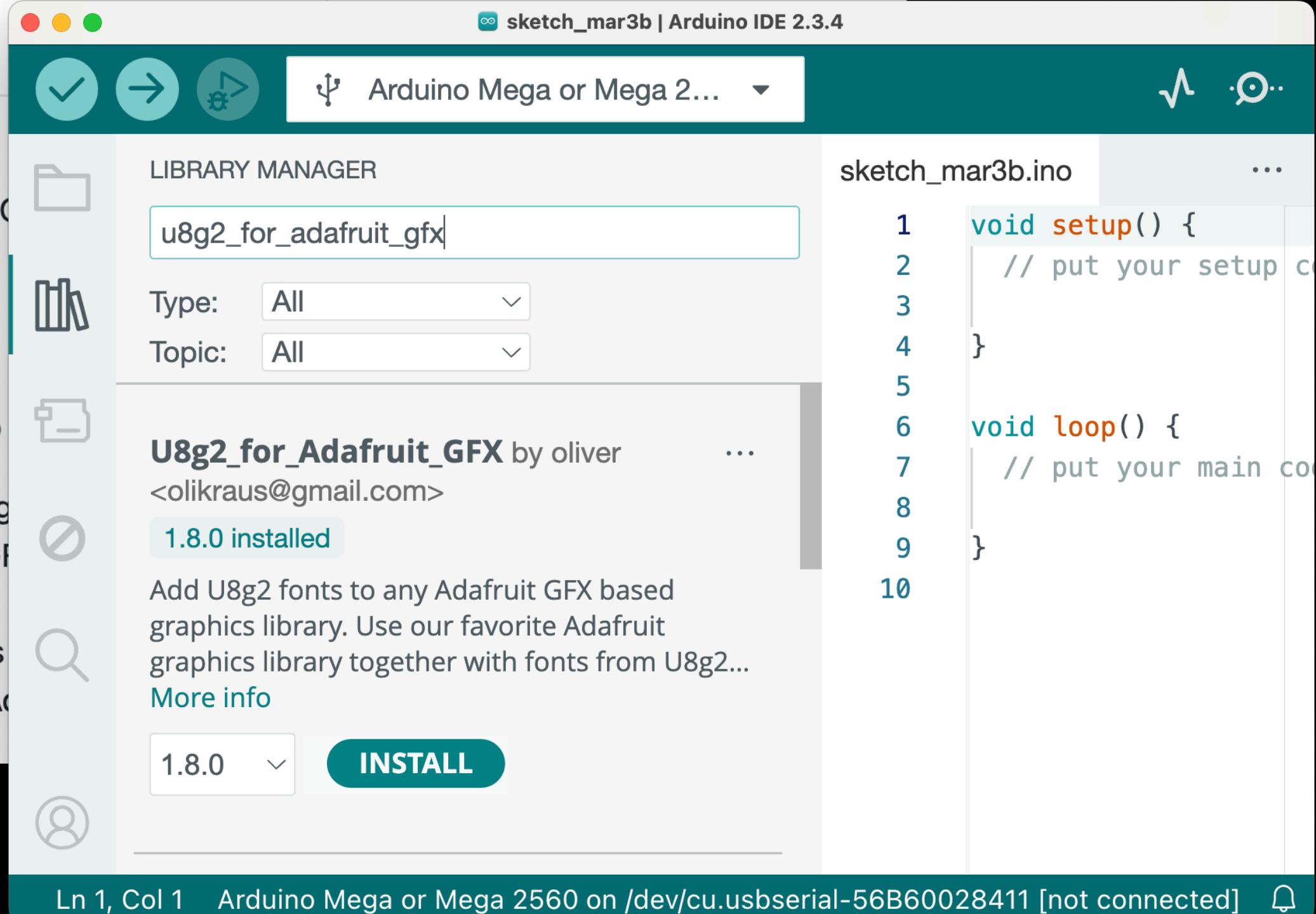
U8g2_for_Adafruit_GFX

Download:
https://github.com/olikraus/U8g2_for_Adafruit_GFX/archive/master.zip

What is U8g2_for_Adafruit_GFX?

- Arduino Library
- Adds a the [U8g2](#) text drawing engine to all Adafruit C based libraries.
- All [U8g2 fonts](#) can be used
- Support for UTF-8 and Unicode
- Support for Arduino print() command and F() Macro

U8g2 is a graphics library for monochrome displays. U8g2 displays, some of them are also supported by Adafruit GFX libraries. Others are supported by Adafruit GFX libraries, supported by U8g2. `U8g2_for_Adafruit_GFX` connects Adafruit Library and adds support for U8g2 fonts to all Adafruit based libraries.



The screenshot shows the Arduino IDE Library Manager window. The title bar says "sketch_mar3b | Arduino IDE 2.3.4". The main area is titled "LIBRARY MANAGER" with a search bar containing "u8g2_for_adafruit_gfx". Below the search bar are dropdown menus for "Type: All" and "Topic: All". A list item for "U8g2_for_Adafruit_GFX" by oliver <olikraus@gmail.com> is shown, with the status "1.8.0 installed". A description below the list states: "Add U8g2 fonts to any Adafruit GFX based graphics library. Use our favorite Adafruit graphics library together with fonts from U8g2...". At the bottom of the list is a "More info" link. To the right of the list, a code editor window shows a sketch named "sketch_mar3b.ino" with the following code:

```
void setup() {  
    // put your setup code here  
}  
  
void loop() {  
    // put your main code here  
}
```

At the bottom of the Arduino IDE window, a status bar displays: "Ln 1, Col 1 Arduino Mega or Mega 2560 on /dev/cu.usbserial-56B60028411 [not connected]" and a bell icon.

Bold (Negrito)

u8g2_font_helvB24_te
u8g2_font_helvB18_te
u8g2_font_helvB14_te
u8g2_font_helvB12_te
u8g2_font_helvB10_te
u8g2_font_helvB08_te

Helvética

Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**

Regular

u8g2_font_helvR24_te
u8g2_font_helvR18_te
u8g2_font_helvR14_te
u8g2_font_helvR12_te
u8g2_font_helvR10_te
u8g2_font_helvR08_te

Bold (Negrito)

u8g2_font_ncenB24_te
u8g2_font_ncenB18_te
u8g2_font_ncenB14_te
u8g2_font_ncenB12_te
u8g2_font_ncenB10_te
u8g2_font_ncenB08_te

New Century Schoolbook

Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**
Bênção **Bênção**

Regular

u8g2_font_ncenR24_te
u8g2_font_ncenR18_te
u8g2_font_ncenR14_te
u8g2_font_ncenR12_te
u8g2_font_ncenR10_te
u8g2_font_ncenR08_te

```

#include <GxEPD2_BW.h>
#include <U8g2_for_Adafruit_GFX.h>

GxEPD2_290_T94_V2 modeloTela(10, 14, 15, 16);
GxEPD2_BW<GxEPD2_290_T94_V2, GxEPD2_290_T94_V2::HEIGHT> tela(modeloTela);
U8G2_FOR_ADAFRUIT_GFX fontes;

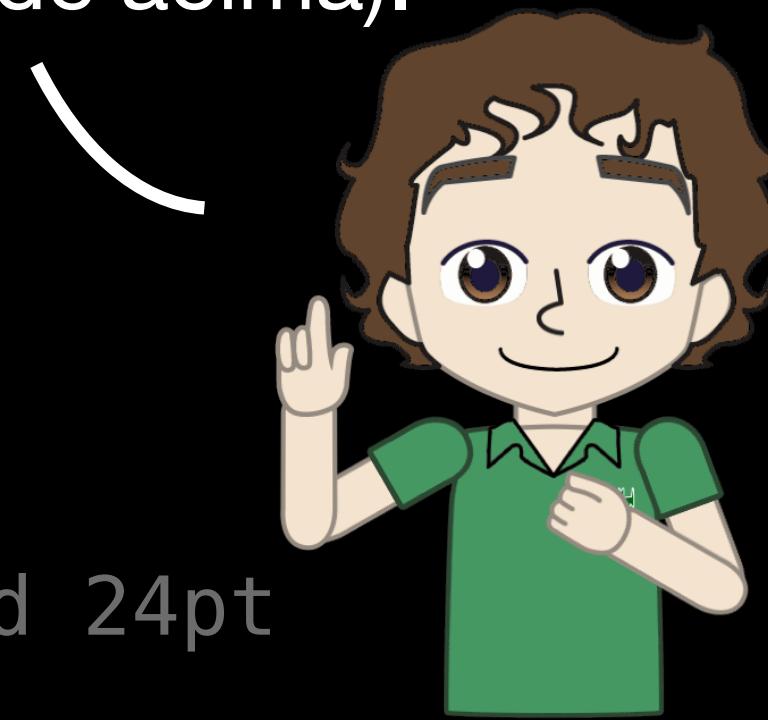
void setup() {
    tela.init();
    tela.setRotation(3);
    tela.fillScreen(GxEPD_WHITE);
    tela.display(true);

    fontes.begin(tela);
    fontes.setForegroundColor(GxEPD_BLACK);
}

                                // em alguma função do código...
    fontes.setFont( u8g2_font_helvB24_te ); // Helvetica Bold 24pt
    fontes.setFontMode(1); // modo transparente
    fontes.setCursor(0, 50); // x, y
    fontes.print("Acentuação ok!");
    tela.display(true);

```

Diferentemente da Adafruit_GFX, a U8g2 define o x,y da setCursor abaixo do texto (em vez de acima).



texto
(x, y)

Acentuação ok!

u8g2_font_open_iconic_all_1x_t

u8g2_font_open_iconic_all_1x_t
BBX Width 8, Height 8, Capital A 8
Font Data Size: 2929 Bytes

64/0040	↔	↔	↔	↔	↔	↔	↔
80/0050	↔	↑	↑	↑	↑	↑	↑
96/0060	↑	↑	↑	↑	↑	↑	↑
112/0070	△	▼	▼	▼	▼	▼	▼
128/0080	↙	↖	↖	↖	↖	↖	↖
144/0090	↙	↖	↖	↖	↖	↖	↖
160/00a0	↙	↖	↖	↖	↖	↖	↖
176/00b0	↙	↖	↖	↖	↖	↖	↖
192/00c0	↙	↖	↖	↖	↖	↖	↖
208/00d0	↙	↖	↖	↖	↖	↖	↖
224/00e0	↙	↖	↖	↖	↖	↖	↖
240/00f0	↙	↖	↖	↖	↖	↖	↖
256/0100	↙	↖	↖	↖	↖	↖	↖
272/0110	↙	↖	↖	↖	↖	↖	↖

u8g2_font_open_iconic_all_2x_t

u8g2_font_open_iconic_all_2x_t
BBX Width 16, Height 16, Capital A 16
Font Data Size: 5617 Bytes

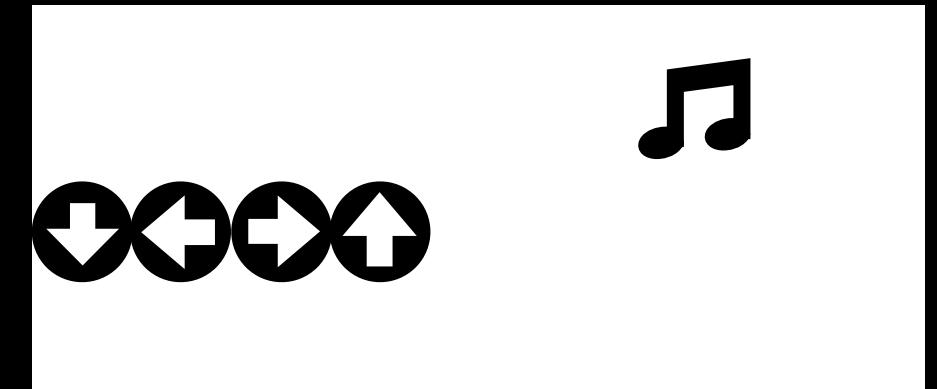
64/0040	↔	↔	↔	↔	↔	↔	↔
80/0050	↔	↑	↑	↑	↑	↑	↑
96/0060	↑	↑	↑	↑	↑	↑	↑
112/0070	△	▼	▼	▼	▼	▼	▼
128/0080	↙	↖	↖	↖	↖	↖	↖
144/0090	↙	↖	↖	↖	↖	↖	↖
160/00a0	↙	↖	↖	↖	↖	↖	↖
176/00b0	↙	↖	↖	↖	↖	↖	↖
192/00c0	↙	↖	↖	↖	↖	↖	↖
208/00d0	↙	↖	↖	↖	↖	↖	↖
224/00e0	↙	↖	↖	↖	↖	↖	↖
240/00f0	↙	↖	↖	↖	↖	↖	↖
256/0100	↙	↖	↖	↖	↖	↖	↖
272/0110	↙	↖	↖	↖	↖	↖	↖

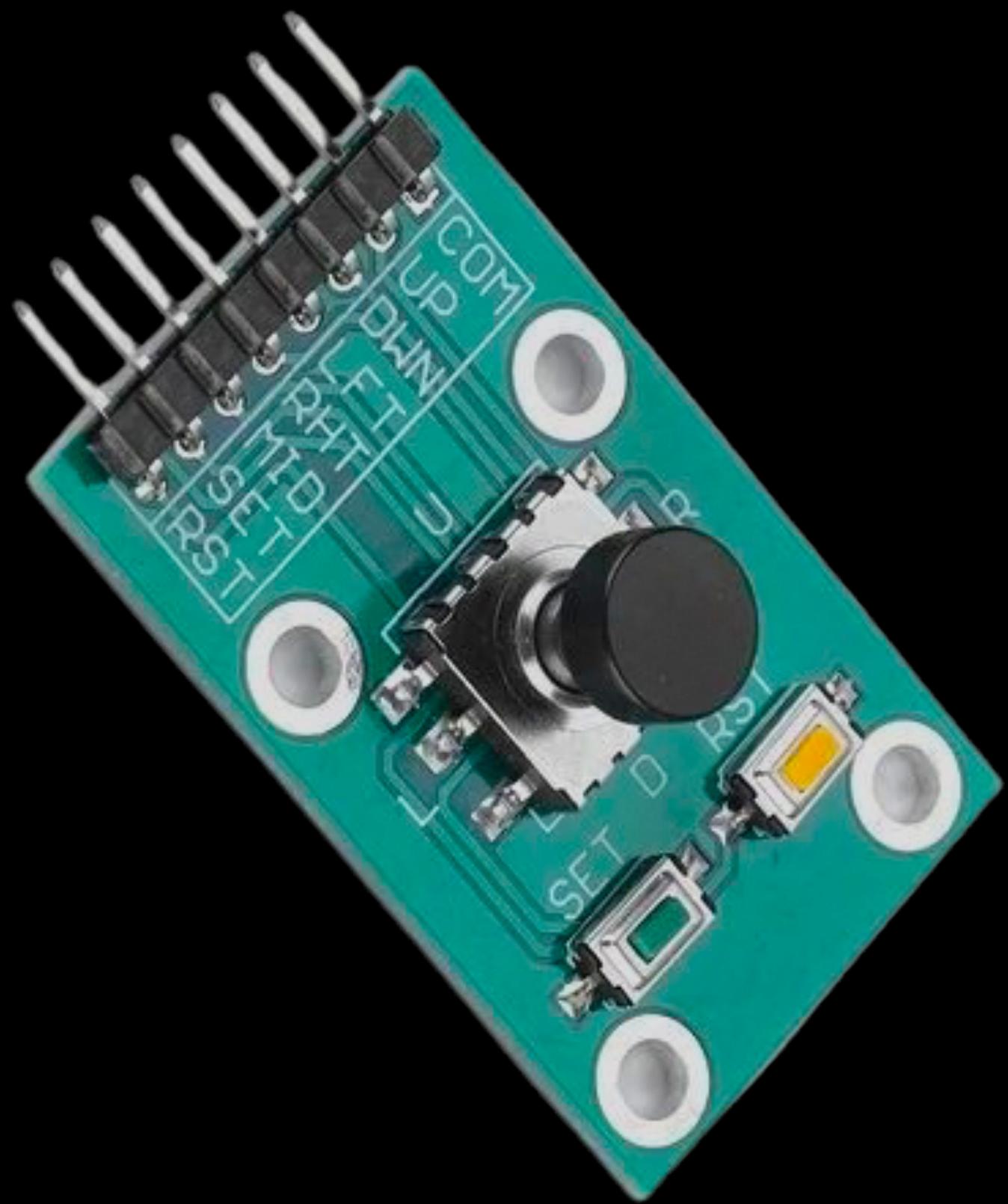
u8g2_font_open_iconic_all_4x_t

u8g2_font_open_iconic_all_4x_t
BBX Width 32, Height 32, Capital A 32
Font Data Size: 11460 Bytes

64/0040	↔	↔	↔	↔	↔	↔	↔
80/0050	↔	↑	↑	↑	↑	↑	↑
96/0060	↑	↑	↑	↑	↑	↑	↑
112/0070	△	▼	▼	▼	▼	▼	▼
128/0080	↙	↖	↖	↖	↖	↖	↖
144/0090	↙	↖	↖	↖	↖	↖	↖
160/00a0	↙	↖	↖	↖	↖	↖	↖
176/00b0	↙	↖	↖	↖	↖	↖	↖
192/00c0	↙	↖	↖	↖	↖	↖	↖
208/00d0	↙	↖	↖	↖	↖	↖	↖
224/00e0	↙	↖	↖	↖	↖	↖	↖
240/00f0	↙	↖	↖	↖	↖	↖	↖
256/0100	↙	↖	↖	↖	↖	↖	↖
272/0110	↙	↖	↖	↖	↖	↖	↖

```
// em alguma função do código...  
  
fontes.setFont( u8g2_font_open_iconic_all_4x_t ); //símbolos  
fontes.setFontMode(1);    // modo transparente  
  
fontes.drawGlyph(200, 50, 0x00e1); // x, y, código hexa  
  
fontes.setCursor(0, 90); // x, y  
fontes.print("IJKL");  
  
tela.display(true);
```

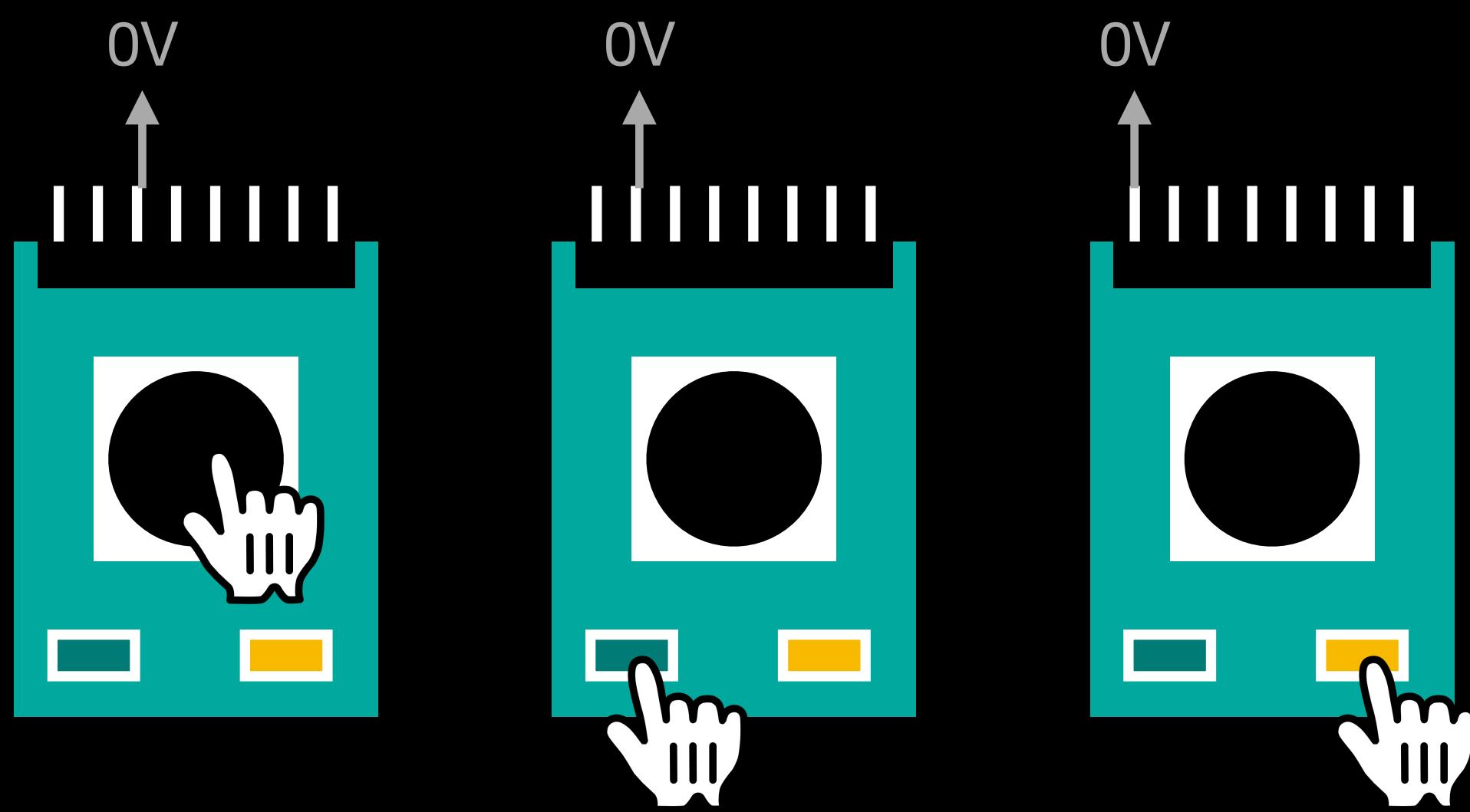
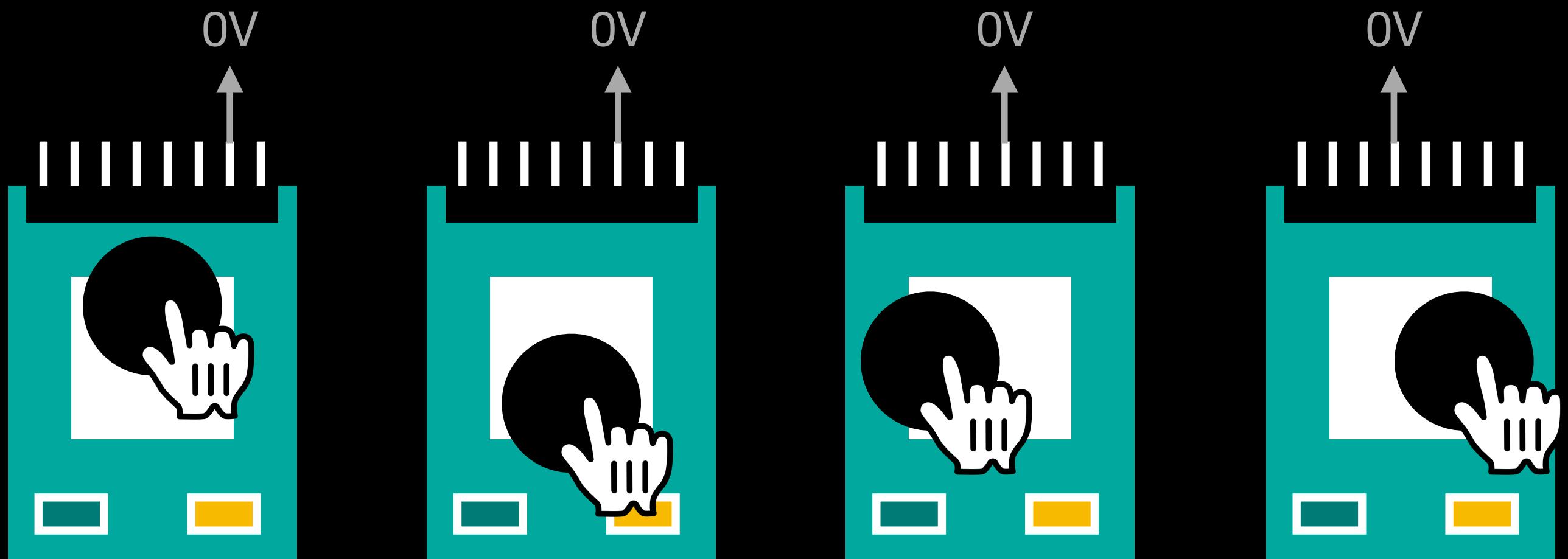
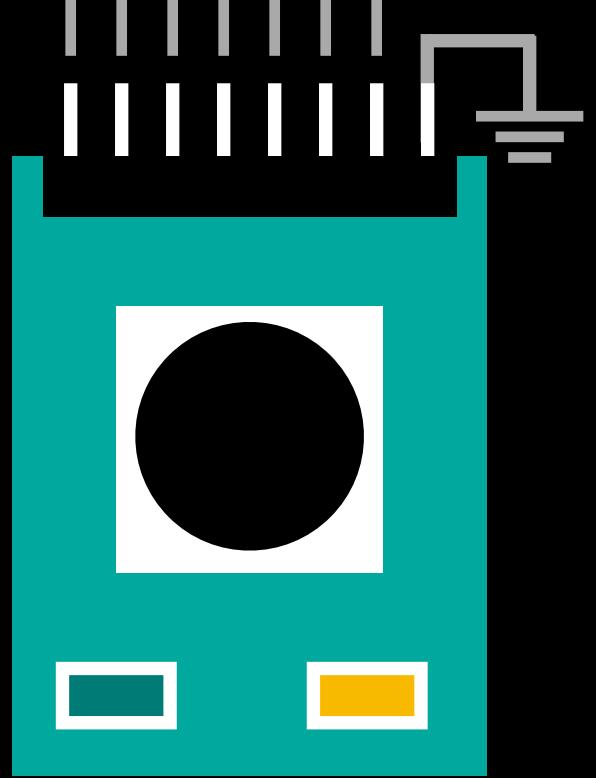






ESP32

7 6 5 4 3 2 1



```
#include <GButton.h>

GButton botaoEsquerda(3);
GButton botaoDireita(4);
int x = 50;

void andaParaEsquerda(GButton &botao) {
    x -= 5;
}

void andaParaDireita(GButton &botao) {
    x += 5;
}

void setup() {
    botaoEsquerda.setPressHandler(andaParaEsquerda);
    botaoDireita.setPressHandler(andaParaDireita);
}

void loop() {
    botaoEsquerda.process();
    botaoDireita.process();
}
```

Teria como os dois botões
chamarem uma mesma função
passando parâmetros diferentes?



```
#include <GFButton.h>

GFButton botaoEsquerda(3);
GFButton botaoDireita(4);
int x = 50;

void atualiza(int variacao) {
    x += variacao;
}

void setup() {
    botaoEsquerda.setPressHandler([](GFButton &b){ atualiza(-5); });
    botaoDireita.setPressHandler([](GFButton &b){ atualiza(+5); });
}

void loop() {
    botaoEsquerda.process();
    botaoDireita.process();
}
```

Internet



MQTT

(Message Queuing Telemetry Transport)



Eu entendi a
referência!



A gente tem
um Hulk.



Vim aqui para
barganhar...



Amor é para
crianças...



Ele é adotado.



Não estou me
sentindo bem...



Você tirou tudo
de mim!



Eu estou
sempre irritado.

Discord

Orchard eSports

welcome

Wave to say hi!

November 7, 2023

Buffy just boosted the server! 11/07/2023 11:49 AM

December 4, 2023

LemonJuice just boosted the server! 12/04/2023 2:57 PM

LemonJuice just boosted the server! Orchard eSports ha...

December 5, 2023

LemonJuice just landed. 12/05/2023 12:16 PM

Wave to say hi!

December 6, 2023

LemonJuice just showed up! 12/06/2023 12:11 PM

Wave to say hi!

January 23, 2024

WELCOME

announcements

welcome

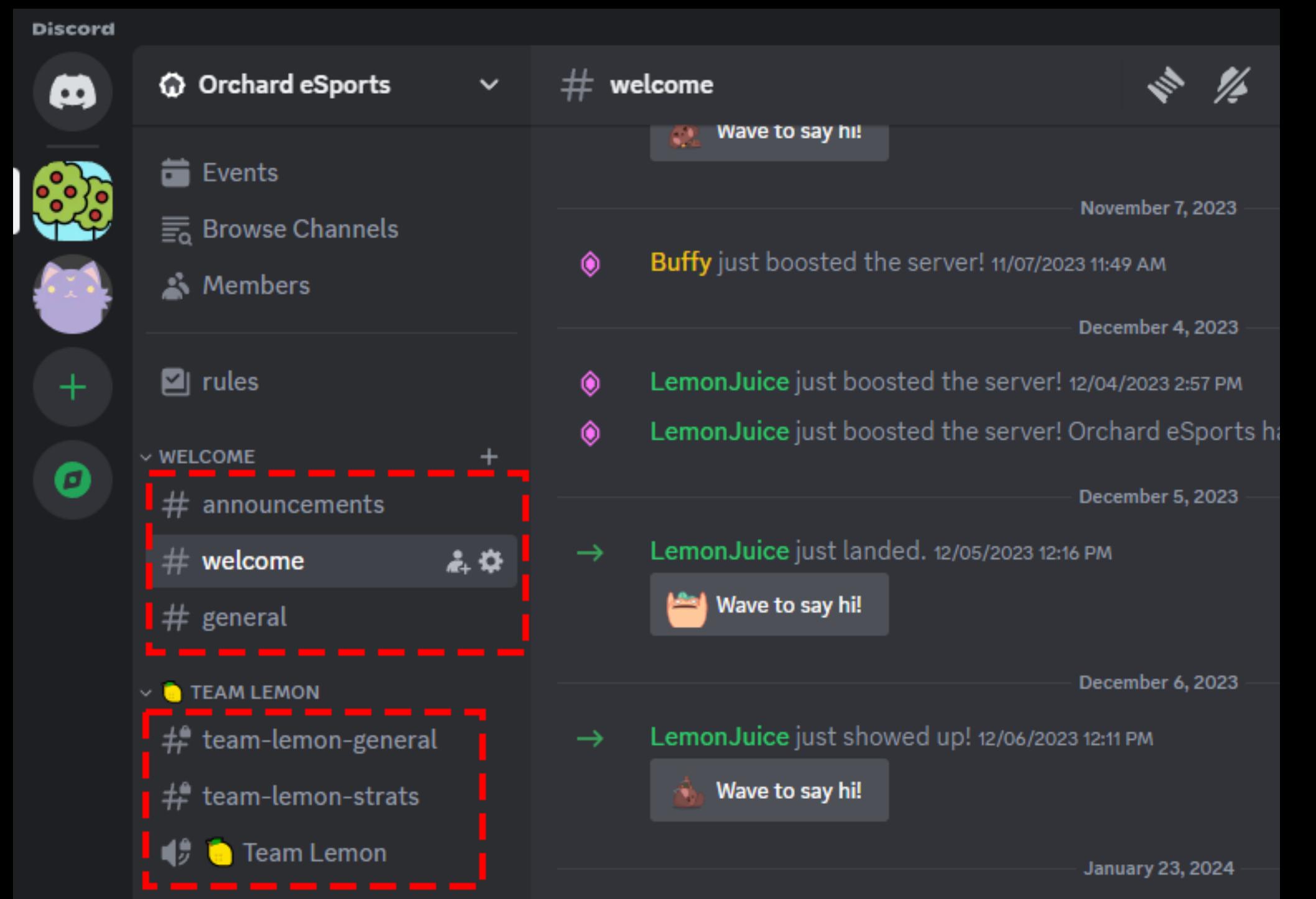
general

TEAM LEMON

team-lemon-general

team-lemon-strats

Team Lemon



10:30

Apartamentos do condo...

Comunidade

Avisos 10:21

Carla: Os cookies famosos de Lucas sã...

Grupos dos quais você participa

Geral 10:17

Rafa: Sejam bem-vindos aos apartame...

Novos inquilinos 09:43

João: Olá a todos!

Grupos nos quais você pode entrar

Pedidos de manutenção Novo

Pais e responsáveis

Troca de receitas

+ Adicionar grupo





Inscrições nos Tópicos

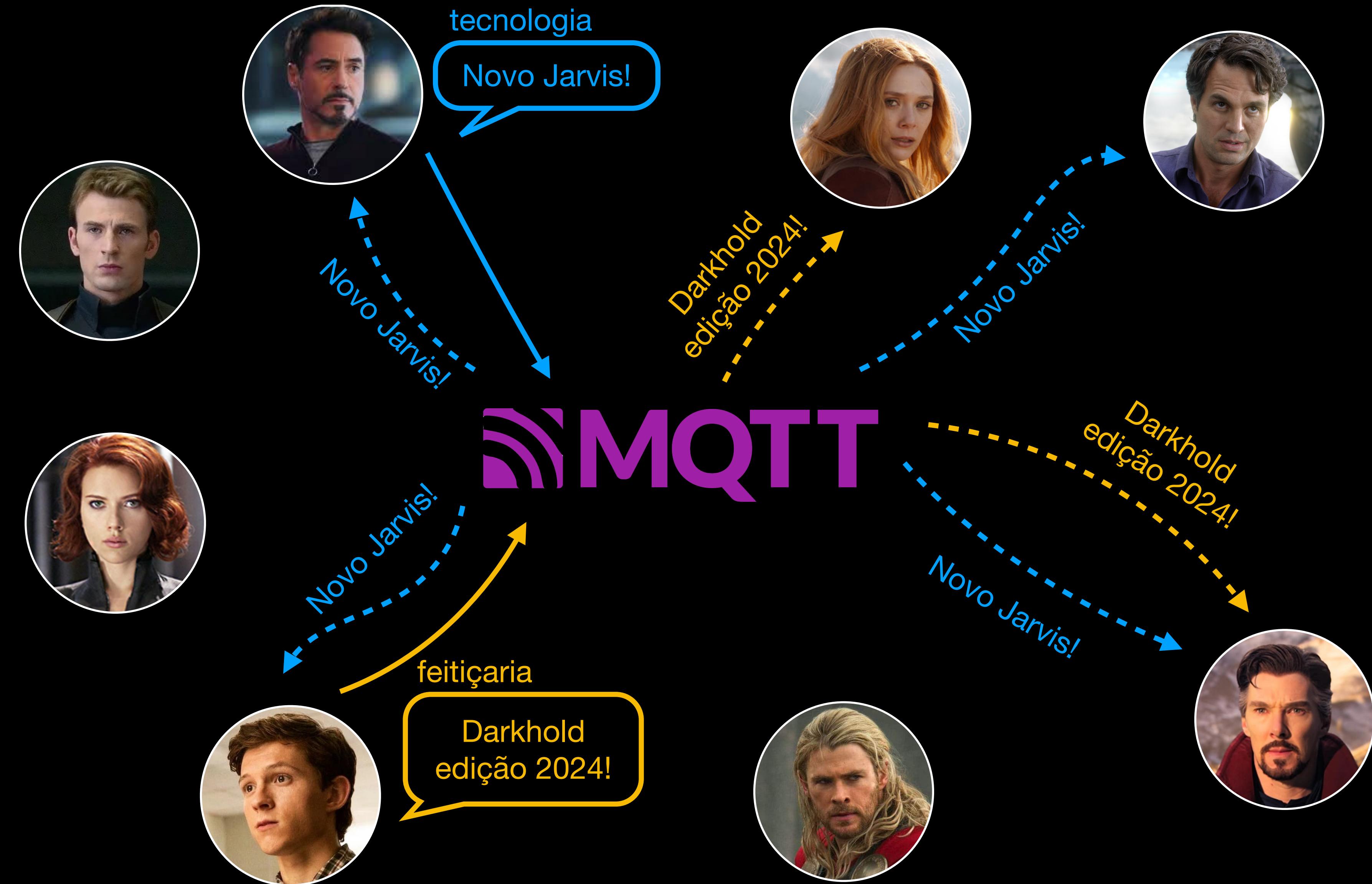
feitiçaria	
tecnologia	
joias	

Me inscreve também
no tópico **tecnologia**

Me inscreve também
no tópico **feitiçaria**

MQTT







joias/2012/terra

Joia da Mente

joias/1970/terra

Joia do Espaço



joias/2012/terra

Joia do Tempo



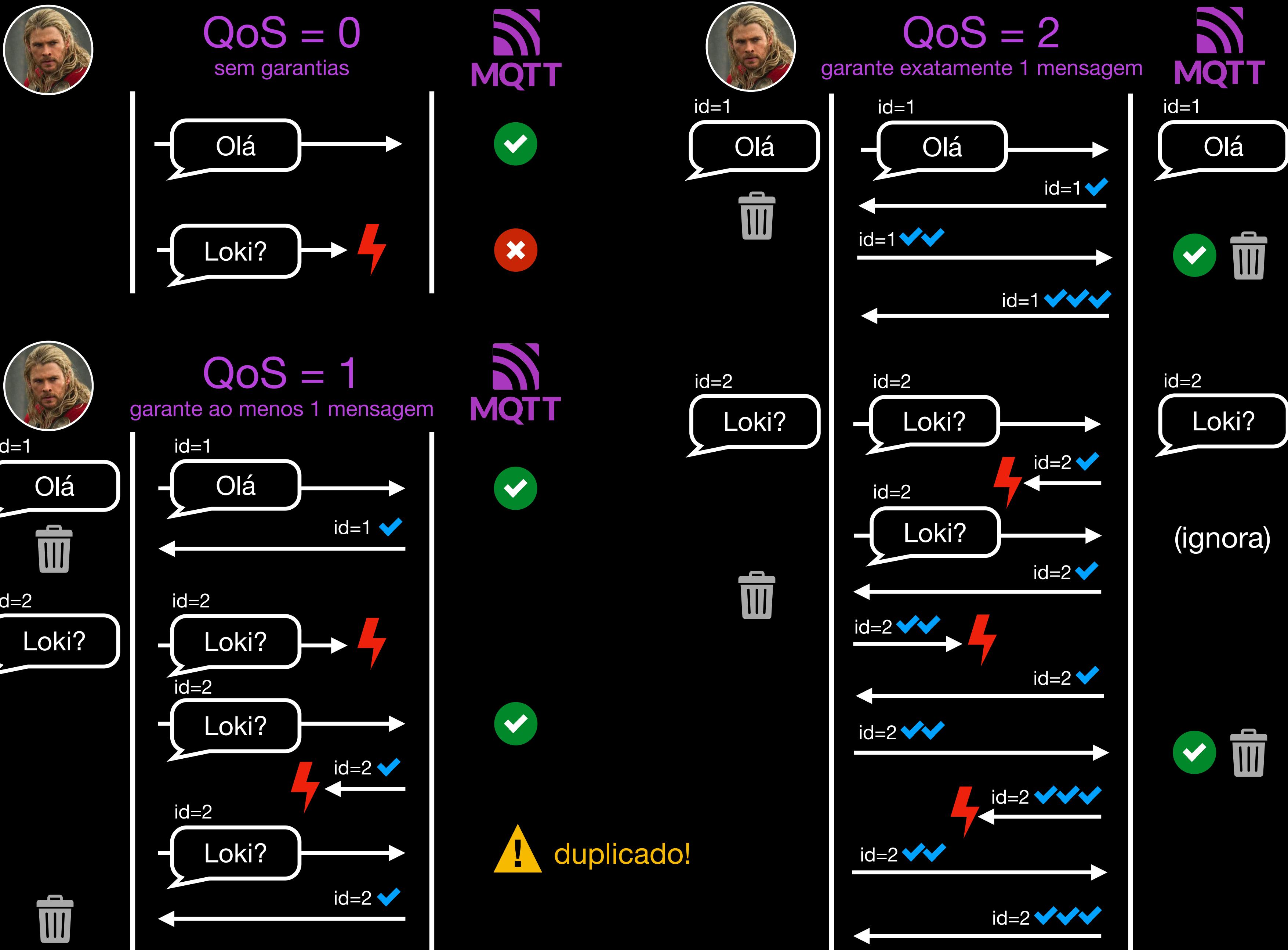
joias/2013/asgard

Joia da Realidade



Tópicos Recebidos de Acordo com a Inscrição

joias/1970/terra	joias/+/terra	joias/#
✓ joias/1970/terra	✓ joias/1970/terra	✓ joias/1970/terra
✗ joias/2012/terra	✓ joias/2012/terra	✓ joias/2012/terra
✗ joias/2013/asgard	✗ joias/2013/asgard	✓ joias/2013/asgard



Sinal de Vida na Conexão (Keep Alive)



Se eu não der sinal de vida por **10s**,
assuma que eu desconectei



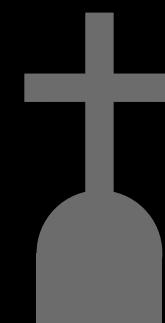
Despedida

❤️x3000



Testamento (Last Will)

❤️x3000



(desconectou)



Feitiçaria

Dica: os avisos
vêm depois do
feitiço.



Reter!



Reter Mensagem

Me inscreve no
tópico **Feitiçaria**



Dica: os avisos vêm
depois do feitiço.

github.com/256dpi/arduino-mqtt?tab=release

No packages published

Contributors 16

Test failing release v2.5.2

This library bundles the [lwmqtt](#) MQTT 3.1.1 client and adds a thin wrapper to get an Arduino like API.

Download the latest version from the [release](#) section or install it via the built-in Library Manager in the Arduino IDE and select the sketch below.

The library is also available on [PlatformIO](#). You can install it via the command:

```
pio lib install "256dpi/MQTT"
```

Compatibility

The following examples show how you can use the library with various Arduino compatible hardware:

- [Arduino Yun & Yun-Shield \(Secure\)](#)
- [Arduino Ethernet Shield](#)
- [Arduino WiFi Shield](#)

sketch_mar2a | Arduino IDE 2.3.4

sketch_mar2a.ino

```
void setup() {  
    // put your setup code here  
}  
  
void loop() {  
    // put your main code here  
}
```

LIBRARY MANAGER

matt

Type: All

Topic: All

MQTT by Joel Gaehwiler
<joe.l.gaehwiler@gmail.com>
2.5.2 installed

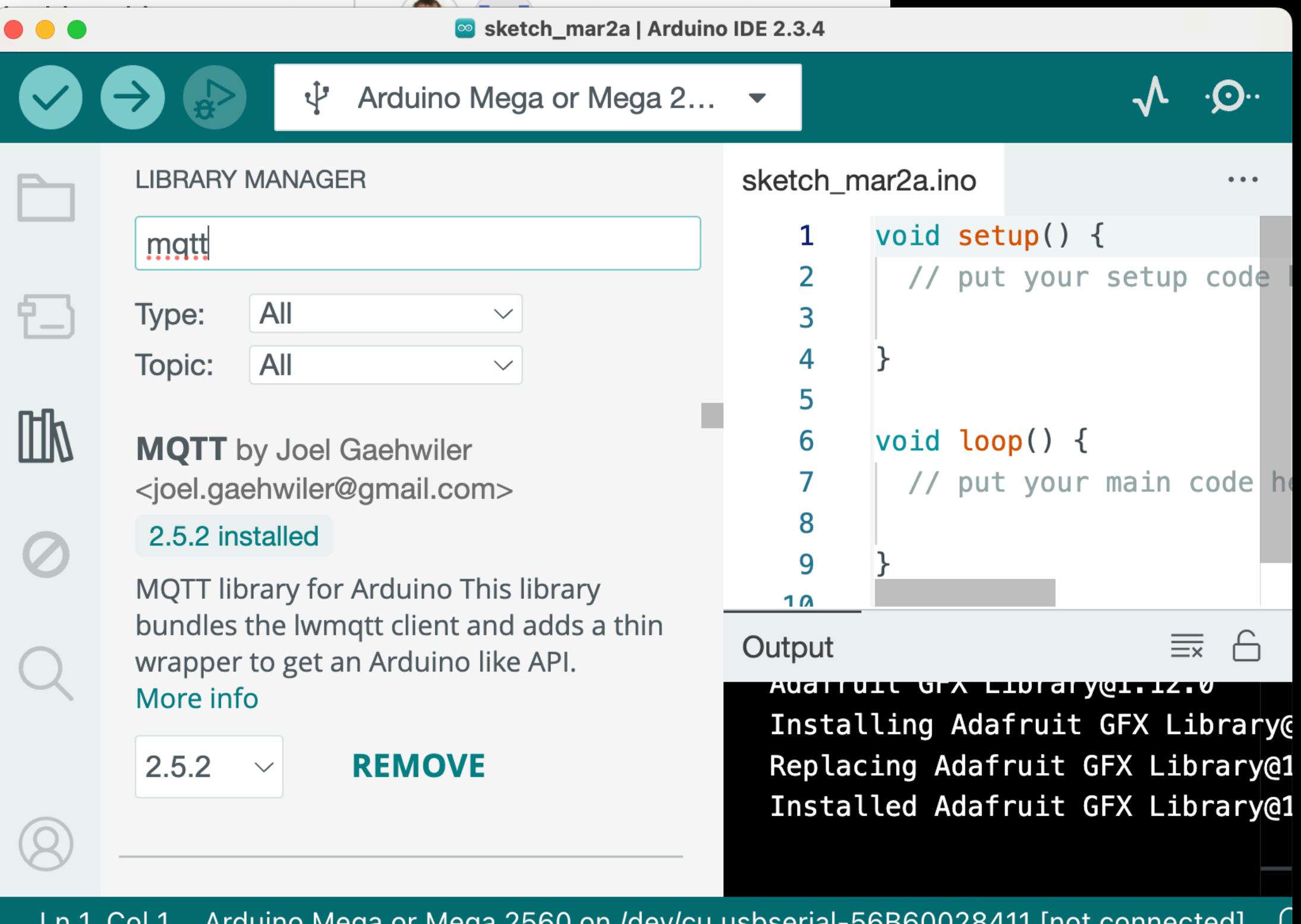
MQTT library for Arduino This library bundles the lwmqtt client and adds a thin wrapper to get an Arduino like API.
[More info](#)

2.5.2 REMOVE

Output

```
Adafruit GFX Library@1.12.0  
Installing Adafruit GFX Library@1.12.0  
Replacing Adafruit GFX Library@1.12.0  
Installed Adafruit GFX Library@1.12.0
```

Ln 1, Col 1 Arduino Mega or Mega 2560 on /dev/cu.usbserial-56B60028411 [not connected]



```

#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "certificados.h"
#include <MQTT.h>

WiFiClientSecure conexaoSegura;
MQTTClient mqtt(1000); // tamanho máximo das mensagens (1000 bytes)

void reconnectMQTT() {
    if (!mqtt.connected()) {
        Serial.print("Conectando MQTT...");
        while(!mqtt.connected()) {
            mqtt.connect("IDENTIFICADOR ÚNICO SEU", "LOGIN", "SENHA");
            Serial.print(".");
            delay(1000);
        }
        Serial.println(" conectado!");
    }
    mqtt.subscribe("topic01"); // qos = 0
    mqtt.subscribe("topico2+/parametro", 1); // qos = 1
}
// código continua a seguir...

```

Conectando MQTT..... conectado!

```
// continuação do código anterior...

void recebeuMensagem(String topico, String conteudo) {
    Serial.println(topico + ":" + conteudo);
}

void setup() {
    Serial.begin(115200); delay(500);

    reconnectarWiFi();
    conexaoSegura.setCACert(certificado1);

    mqtt.begin("ENDERECO_DO_SERVIDOR_MQTT.COM", 8883, conexaoSegura);
    mqtt.onMessage(recebeuMensagem);

    reconnectarMQTT();
}

void loop() {
    reconnectarWiFi();

    reconnectarMQTT();
    mqtt.loop();
}
```

```
void setup() {  
    // Serial, WiFi, certificado...  
  
    mqtt.begin("ENDERECO_DO_SERVIDOR_MQTT.COM", 8883, conexaoSegura);  
    mqtt.onMessage(recebeuMensagem);  
  
    mqtt.setKeepAlive(10);  
    mqtt.setWill("tópico de despedida", "3000"); // testamento  
  
    reconnectarMQTT();  
}  
    // em alguma função do código...  
  
    mqtt.publish("topico1", "conteúdo"); // retain = false, qos = 0  
    mqtt.publish("topico2/123/abc", "conteúdo 2", false, 1); // qos = 1
```

```
import paho.mqtt.client
import ssl
import certifi

def on_connect(mqtt, dados_usuario, flags, codigo):
    mqtt.subscribe("topico1")
    mqtt.subscribe("topico2/+/parametro")

def on_message(cliente, dados_usuario, mensagem):
    texto_recebido = mensagem.payload.decode("UTF-8")
    print("tópico " + msg.topic + ": " + texto_recebido)

mqtt = paho.mqtt.client.Client()
mqtt.tls_set(certifi.where())
mqtt.username_pw_set(username="LOGIN", password="SENHA")

mqtt.on_connect = on_connect
mqtt.on_message = on_message

mqtt.connect("ENDERECO_DO_SERVIDOR_MQTT.COM", port=8883, keepalive=10)

mqtt.publish("tópico", payload="mensagem", qos=2)
```



MQTT X

The screenshot shows the MQTT X application interface. At the top, it displays the connection details: **mqtt.janks.dev.br** with a dropdown arrow and a red notification badge showing **2**. To the right are icons for power (**○**), refresh (**⟳**), edit (**✎**), and more options (**...**).

On the left sidebar, there are several icons: a green circle with a white 'X' (MQTT X logo), a square with four smaller squares (grid or publish), a plus sign (+), a double slash (publish or subscribe), a document with a gear (configuration), a gear (settings), and a Wi-Fi signal icon.

The main area shows two subscriptions:

- topic01** QoS 1 (dark blue background)
- topic02+/teste** QoS 0 (light gray background)

Underneath the subscriptions, there are two received messages:

- Topic: topic01 QoS: 2**
Teste!
2024-09-01 22:00:27:797
- Topic: topic01 QoS: 1**
Teste!
2024-09-01 22:00:27:771

At the bottom of the message list, there are settings for the next message: **Plaintext**, **QoS 2**, **Retain** (unchecked), **Meta**, and a small up arrow icon. Below these settings, the message content is pre-filled: **topic01** and **Teste!**. Navigation buttons (**<**, **-**, **+**, **>**) are also present.