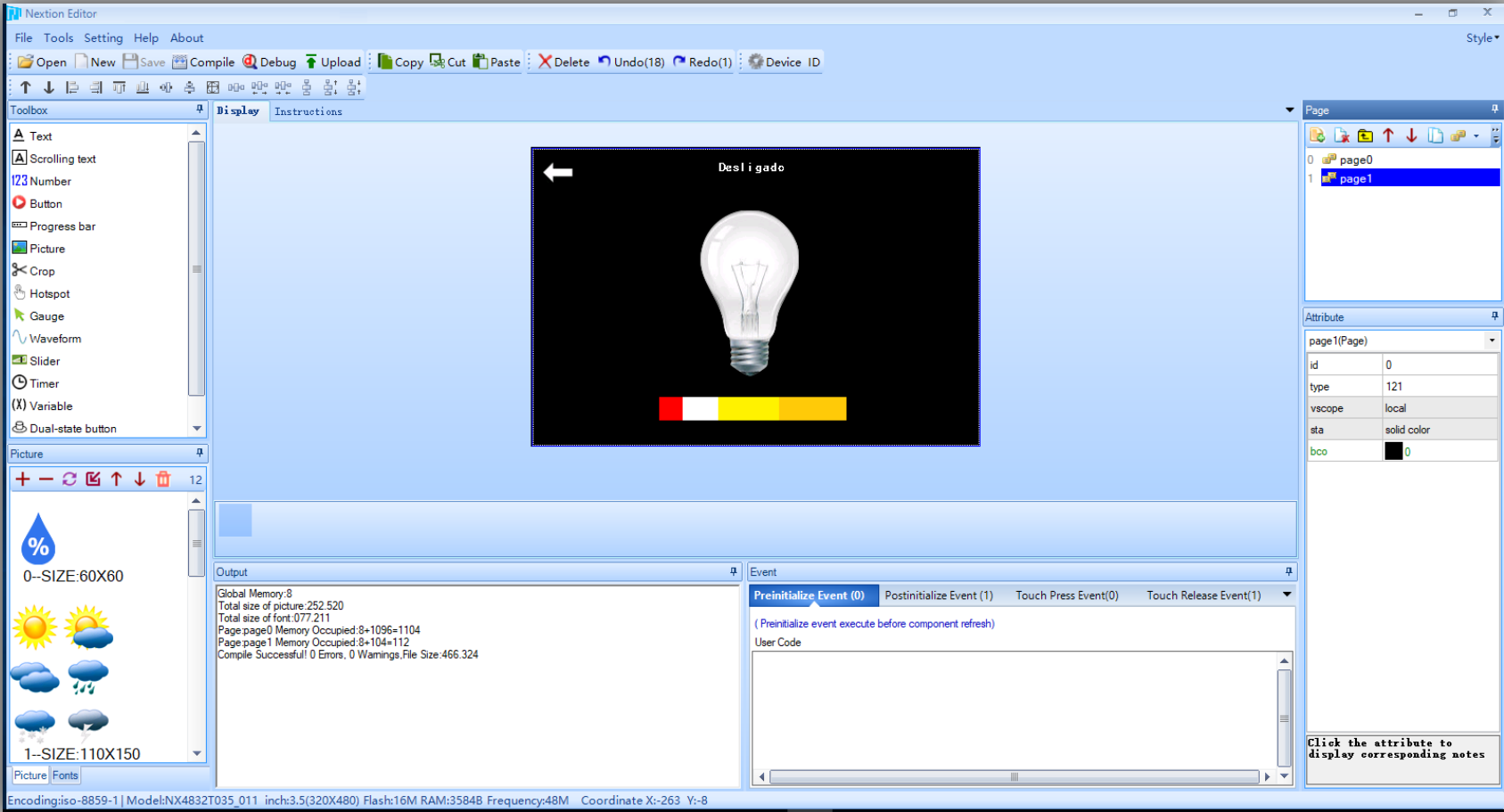


Nextion Editor

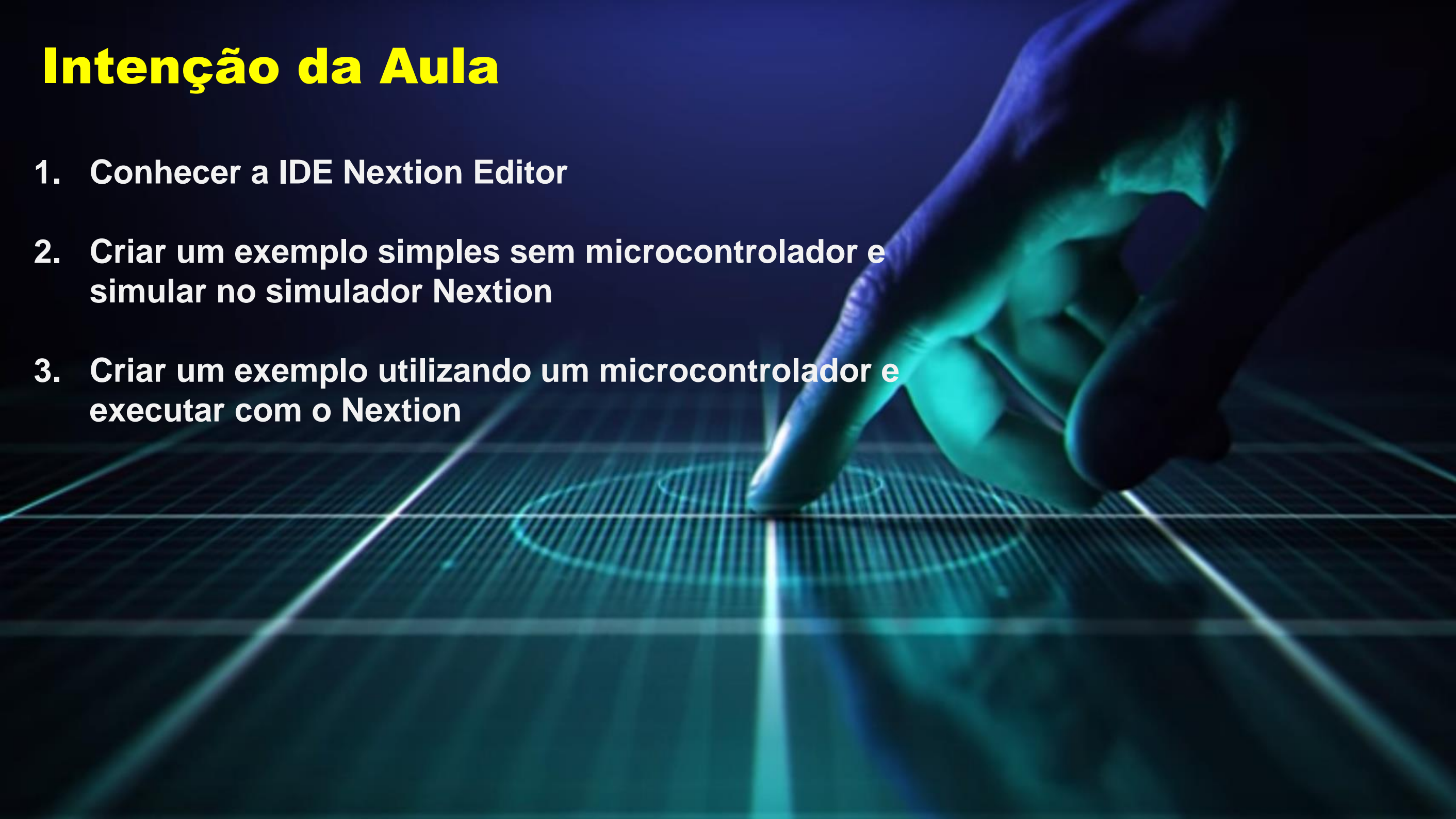


Por Fernando Koyanagi

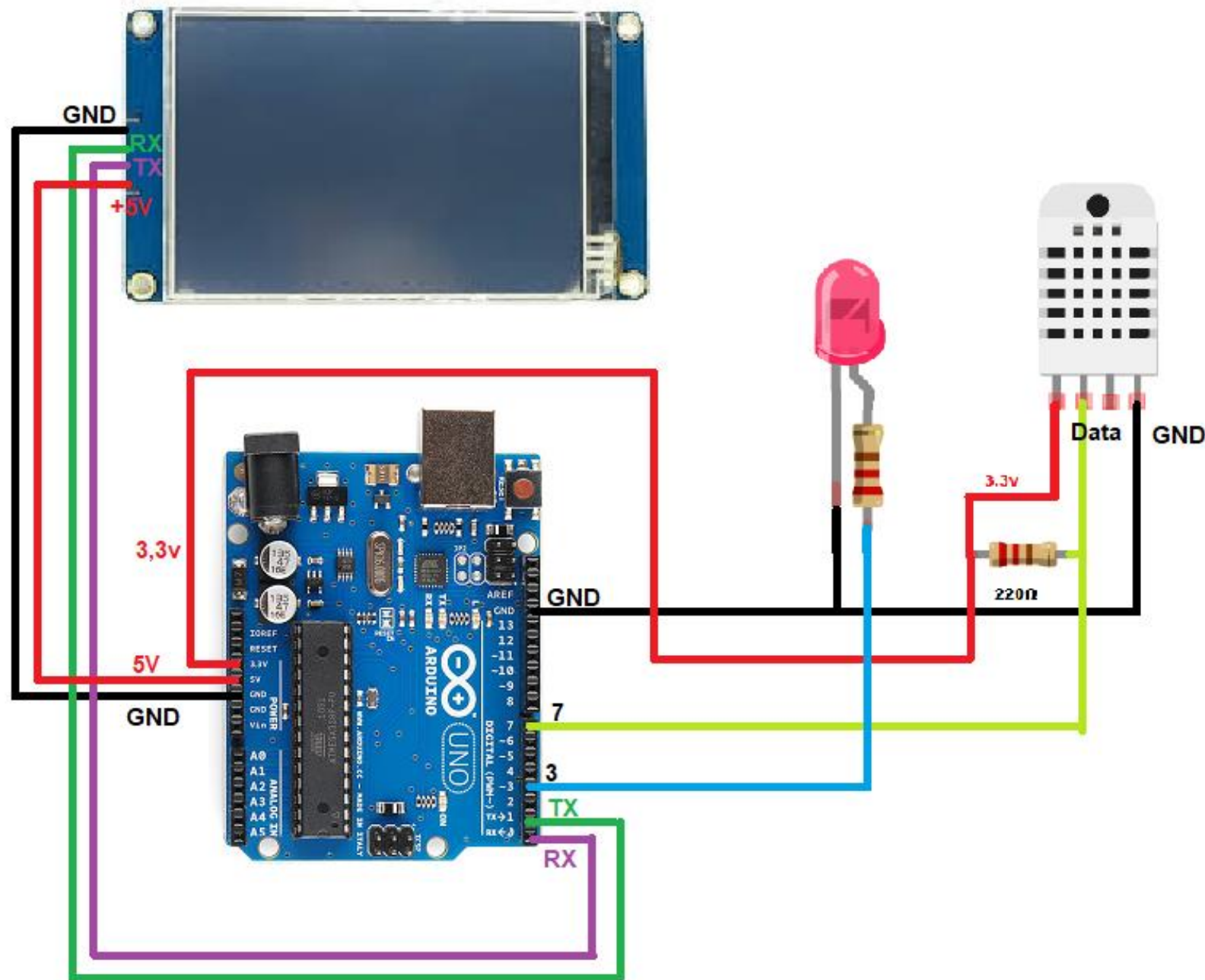


Intenção da Aula

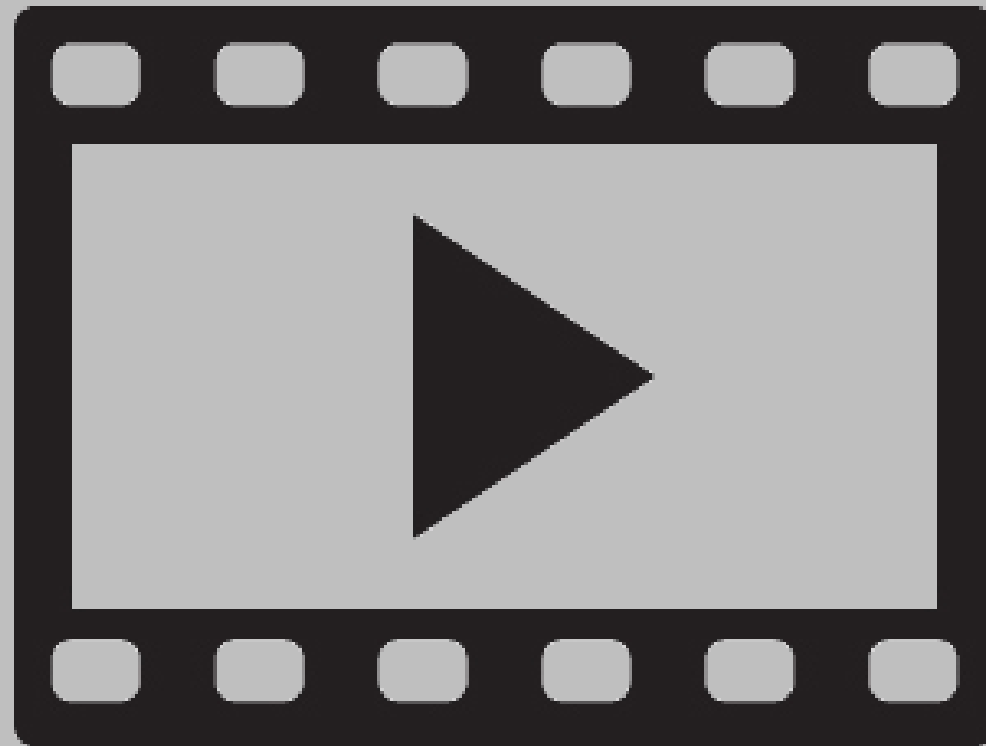
1. Conhecer a IDE Nextion Editor
2. Criar um exemplo simples sem microcontrolador e simular no simulador Nextion
3. Criar um exemplo utilizando um microcontrolador e executar com o Nextion



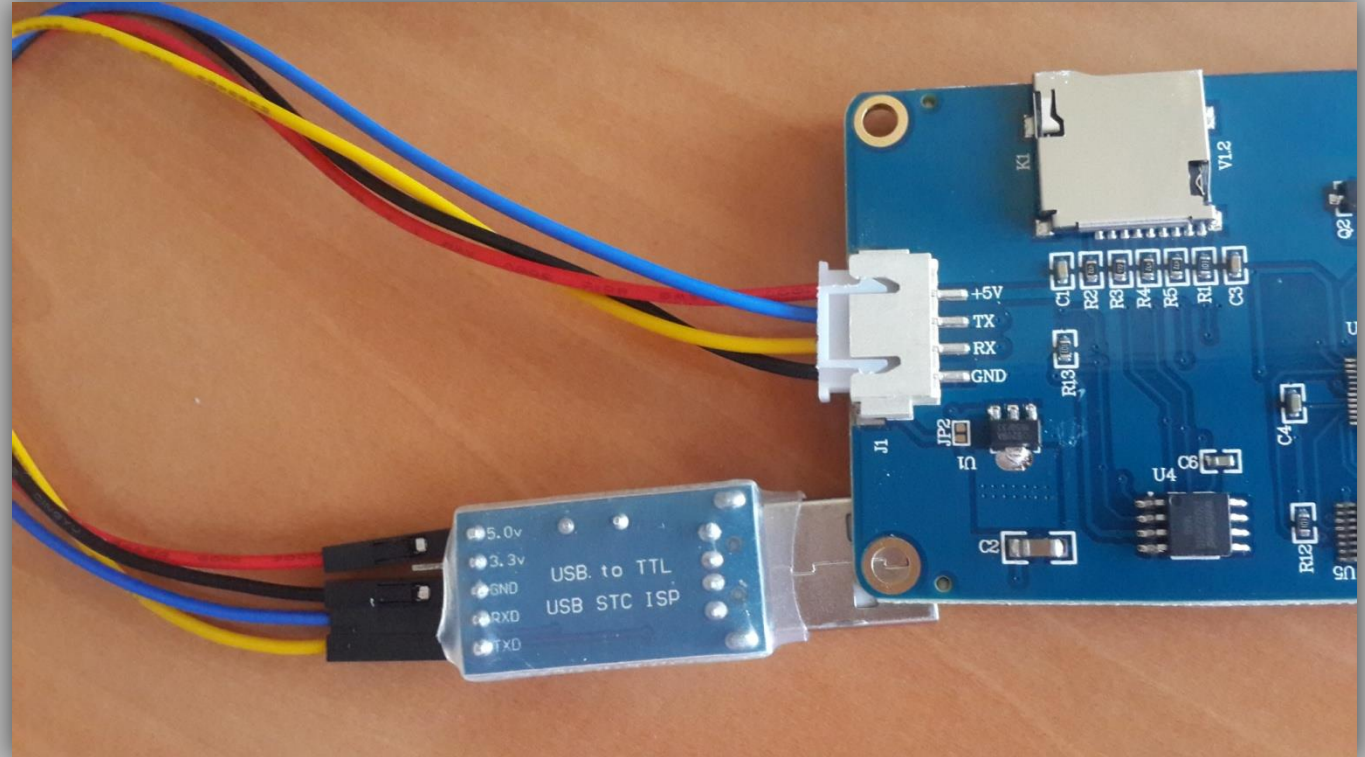
Montagem



Demonstração



Nextion




← → ↻ 🏠 www.fernandok.com

Apps 📱 Fórmula de Lançamei 📄 translator - Pesquisa 📄 Google 📄 Google Maps 📄 DownSub.com | Down 📄 (3) Seus alunos | Khar 📄 YouTube 📄 New Tab 📄 Google DOCS

📅 Domingo, Janeiro 14 2018

📄 f 📄 You Tube 📄




Em www.fernandok.com

PRINCIPAL SOBRE FERNANDO K ARDUINO ESP8266 ESP32 LORAWAN MOTOR DISPLAY MATERIAIS DOWNLOAD

Receba o meu conteúdo GRATUITAMENTE

Insira aqui seu melhor email...

QUERO RECEBER GRÁTIS




Motor de Passo Nema 23 com Driver TB6600 e Arduino Due

by **Fernando K Tecnologia** - 2:44 PM

Hoje vamos voltar a falar de Motor de Passo. Vamos utilizar um Nema 23 que será controlado por um Driver TB6600 e um Arduino Due. É p...

Leia mais




ESP32 Longa Distância - LoRaWan

by **Fernando K Tecnologia** - 9:46 AM

Neste artigo vamos tratar da LoRaWAN, uma rede que vai longe gastando pouca energia. Mas, o quanto "longe"? Com o chip que uso no vídeo...

Leia mais



Motor de HD com Arduino

by **Fernando K Tecnologia** - 2:00 PM

QUAL ASSUNTO VOCÊ TEM


- ☐ Arduino
- ☐ ESP8266
- ☐ ESP32
- ☐ Motor
- ☐ Display
- ☐ Sensor

You may select multiple answers.

Votar Exibir resultados

Votos até o momento: 32
Dias restantes para votar: 49

FACEBOOK



Fernando K Tec
16.235 curtidas

Seu e-mail



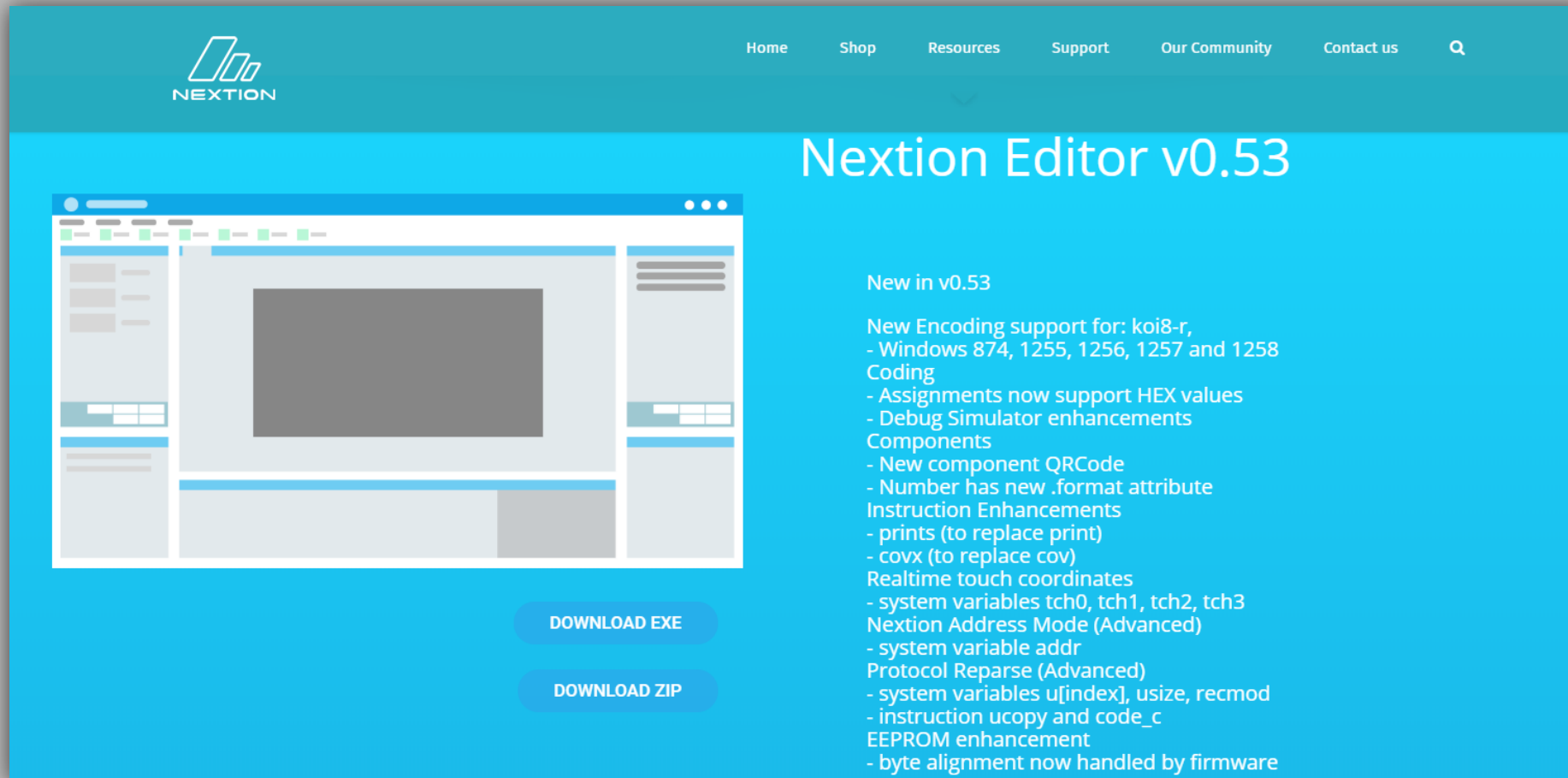
 Inscreva-se



Nextion Editor

Você pode fazer o download do Nextion Editor através do link abaixo:

[download: Nextion Editor](#)

The image is a screenshot of the Nextion Editor v0.53 website. The website has a teal header with the Nextion logo on the left and a navigation menu on the right containing links for Home, Shop, Resources, Support, Our Community, and Contact us, along with a search icon. The main content area has a light blue background. On the left, there is a screenshot of the Nextion Editor software interface, which shows a central workspace with a dark grey rectangle, surrounded by various toolbars and panels. To the right of the screenshot, the text 'Nextion Editor v0.53' is displayed in a large, white, sans-serif font. Below this, there is a section titled 'New in v0.53' followed by a list of updates. At the bottom left, there are two blue buttons with white text: 'DOWNLOAD EXE' and 'DOWNLOAD ZIP'.

Nextion

Home Shop Resources Support Our Community Contact us

Nextion Editor v0.53

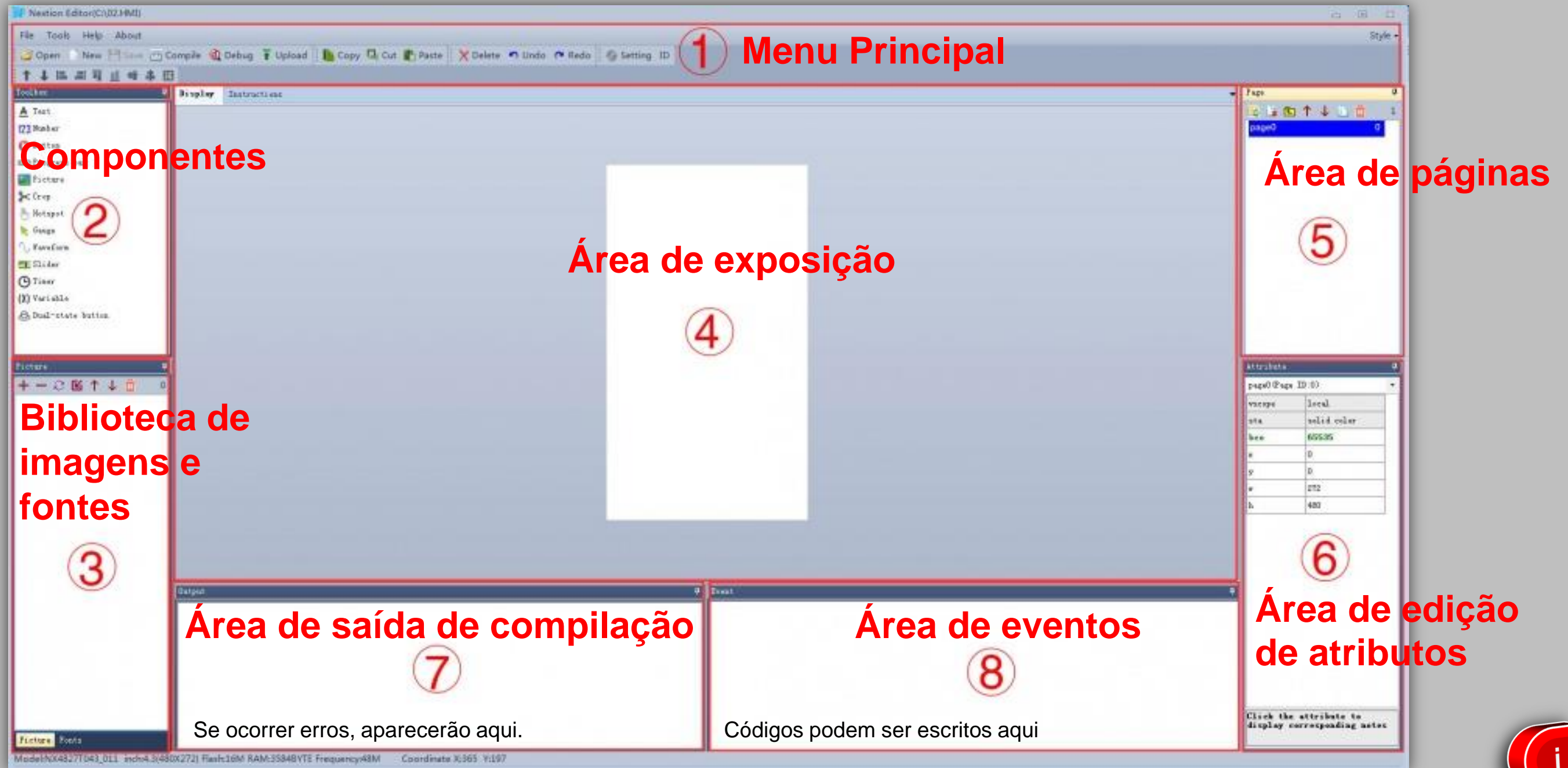
New in v0.53

- New Encoding support for: koi8-r,
 - Windows 874, 1255, 1256, 1257 and 1258
- Coding
 - Assignments now support HEX values
 - Debug Simulator enhancements
- Components
 - New component QRCode
 - Number has new .format attribute
- Instruction Enhancements
 - prints (to replace print)
 - covx (to replace cov)
- Realtime touch coordinates
 - system variables tch0, tch1, tch2, tch3
- Nextion Address Mode (Advanced)
 - system variable addr
- Protocol Reparse (Advanced)
 - system variables u[index], usize, recmod
 - instruction ucopy and code_c
- EEPROM enhancement
 - byte alignment now handled by firmware

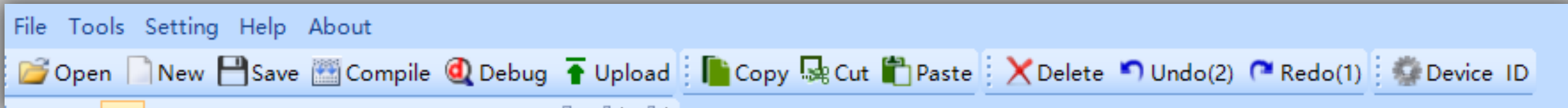
DOWNLOAD EXE

DOWNLOAD ZIP

Nextion Editor



Menu Principal

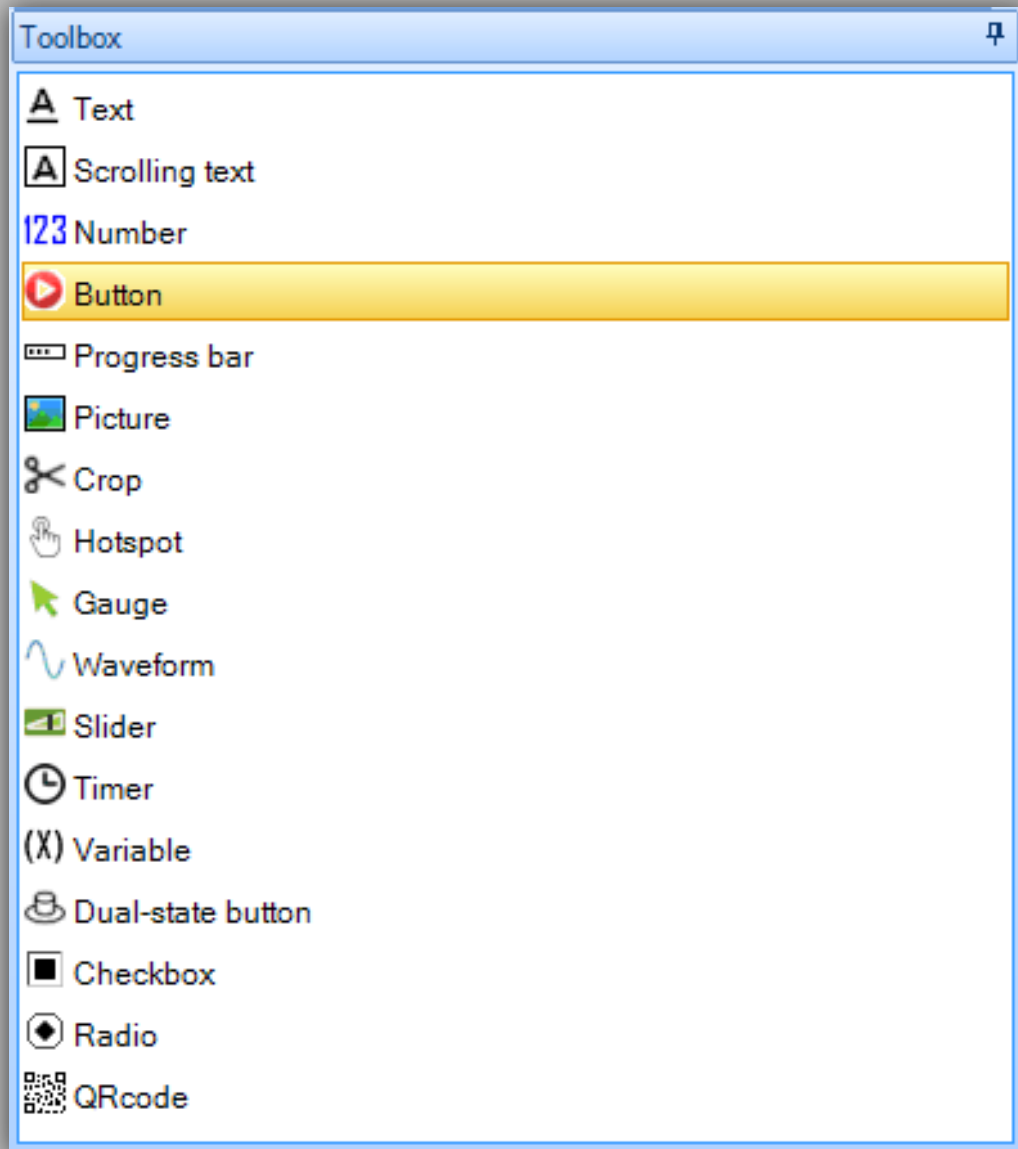


- **File** (criar/abrir/salvar/importar projeto, abrir pasta de construção)
- **Tools** (gerador de fontes)
- **Setting** (configurações, resetar layout)
- **Compile** (compila o projeto)
- **Debug** (abre o simulador para testar o programa) *Não é necessário o Nextion Físico para testar o projeto.
- **Upload** (abre uma janela para fazer o upload do projeto para o Nextion) *pode demorar
- **Device** (abre as configurações de modelo de Nextion e orientação)
- **ID** (exibir/esconder os IDs dos componentes na área de exposição)

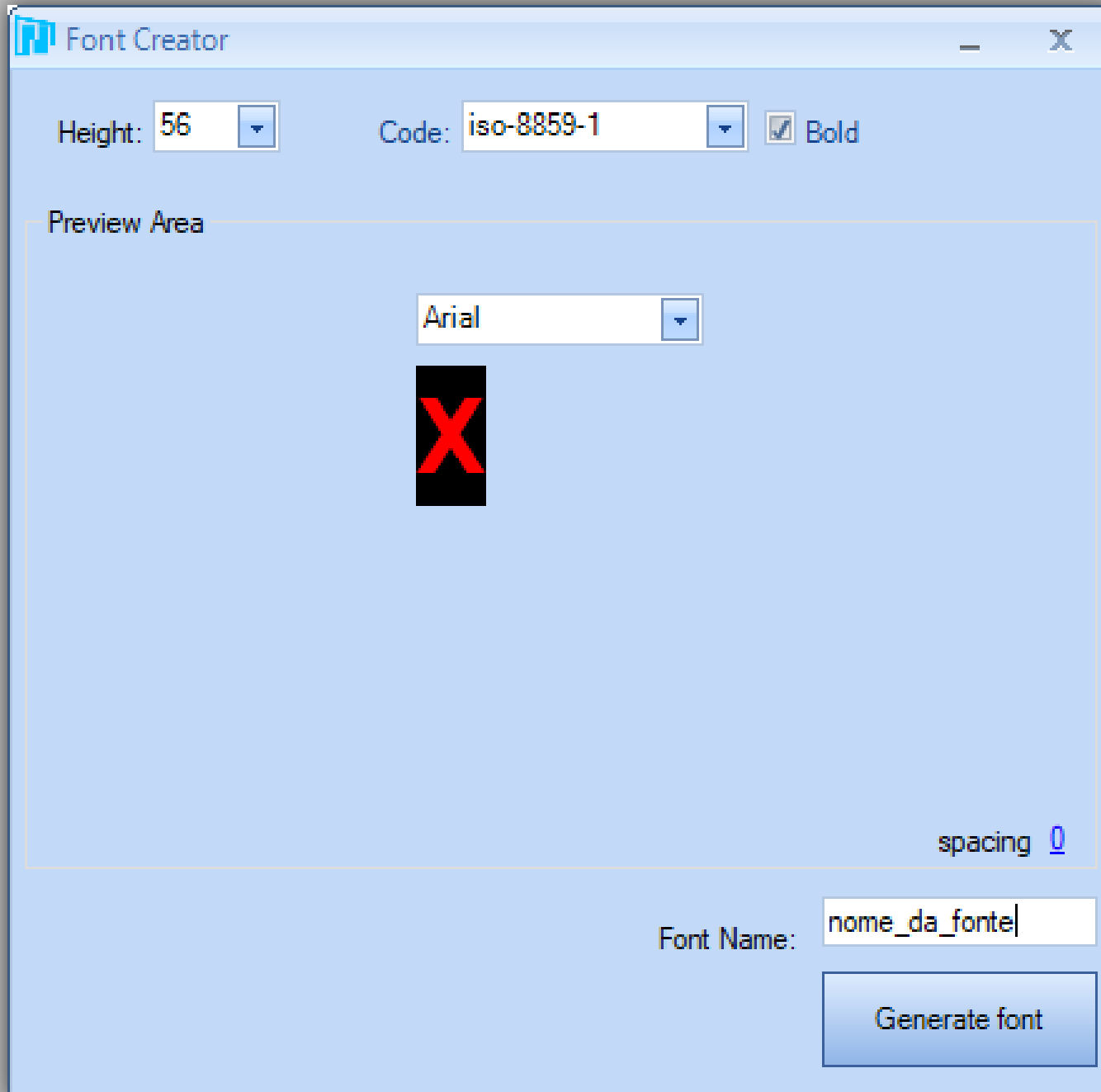


Componentes

Os componentes listados abaixo poderão ser utilizados no projeto, basta clicar sobre o componente desejado, que irá direto para a área de exposição.



Font Generator



Para utilizar componentes de texto, precisamos criar alguma fonte.

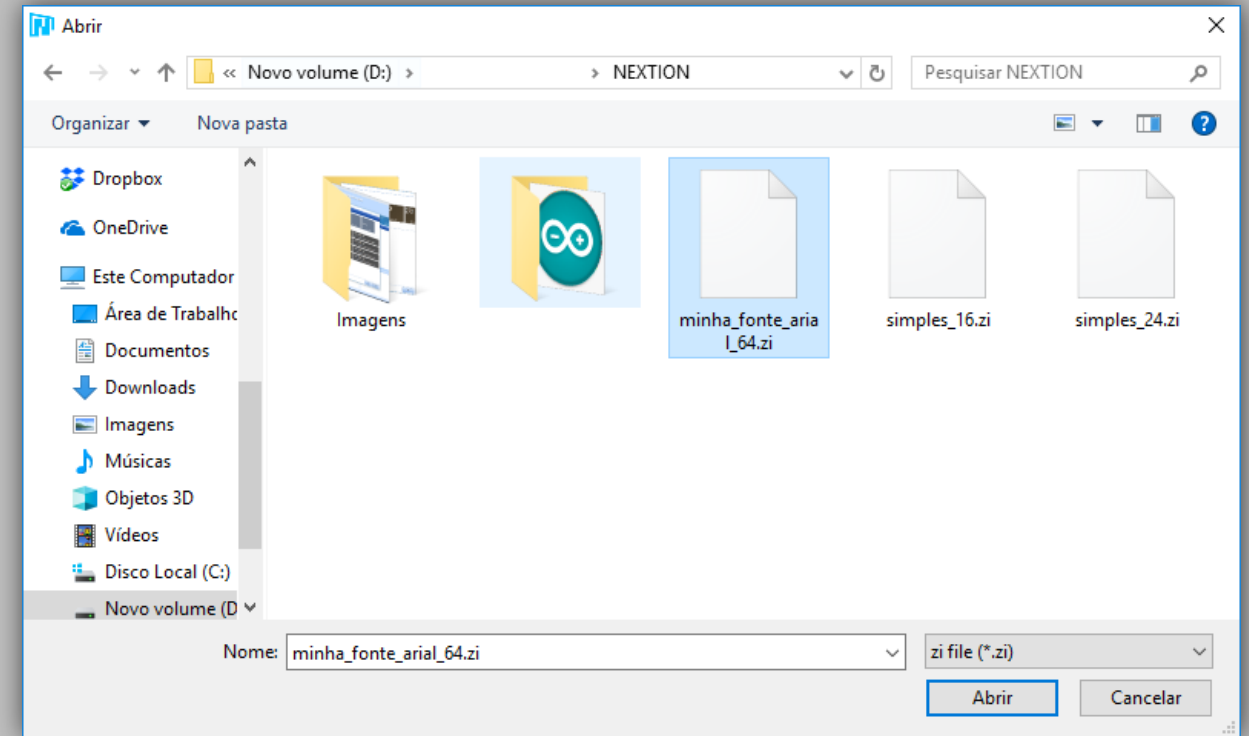
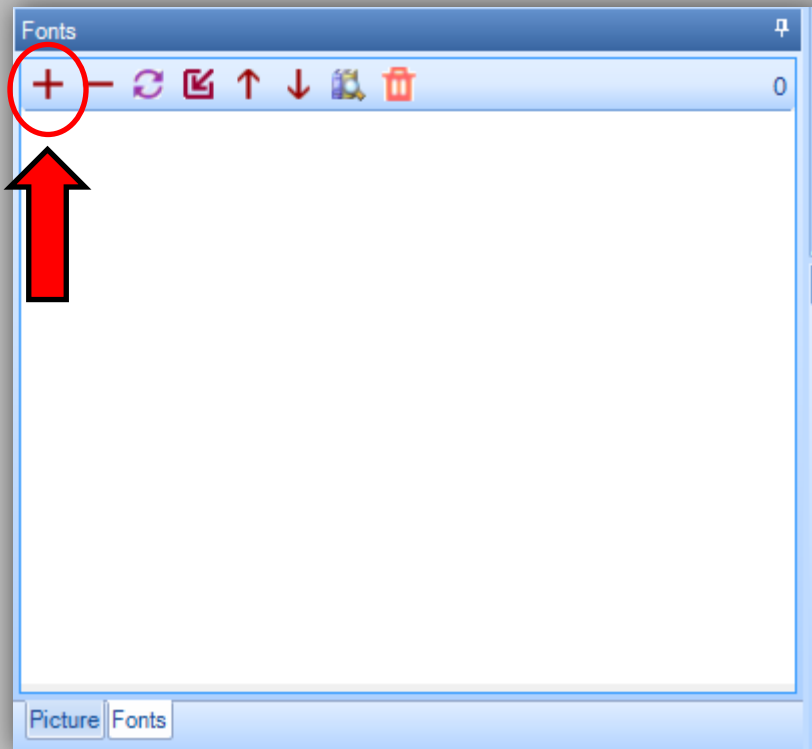
Acesse o menu **Tools >> Font Generator**

Escolha o tipo de fonte, tamanho e estilo, dê um nome a ela.

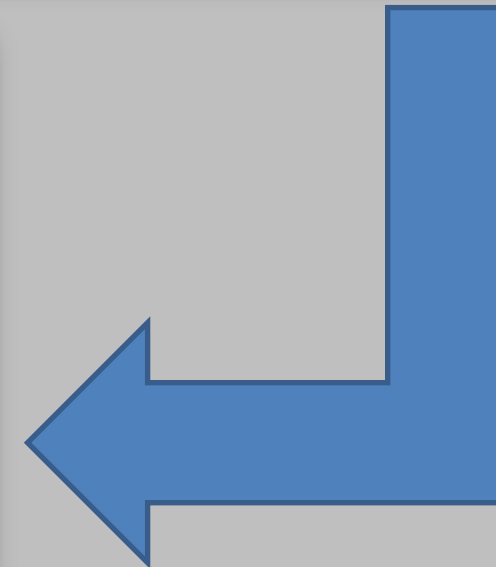
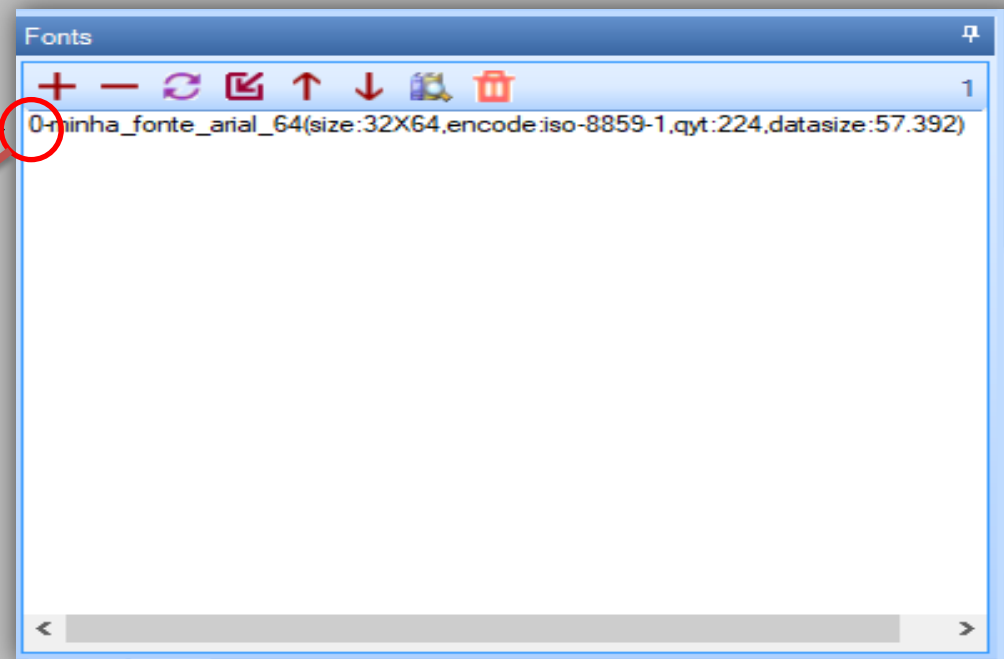
Ao clicar em **Generate Font**, abrirá uma janela para salvar um arquivo com extensão **.zi**. Esse arquivo será utilizado para importar a fonte para o projeto.



Font Generator

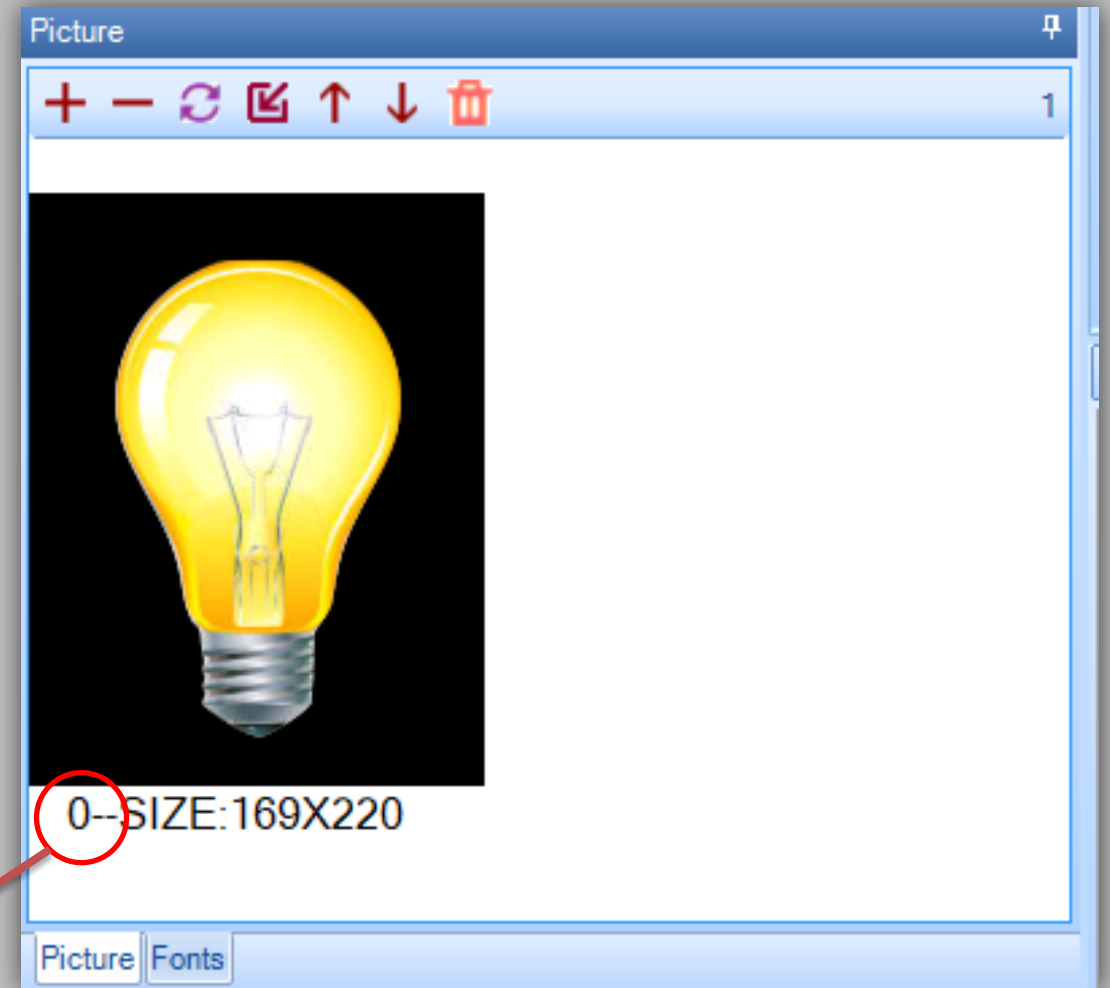
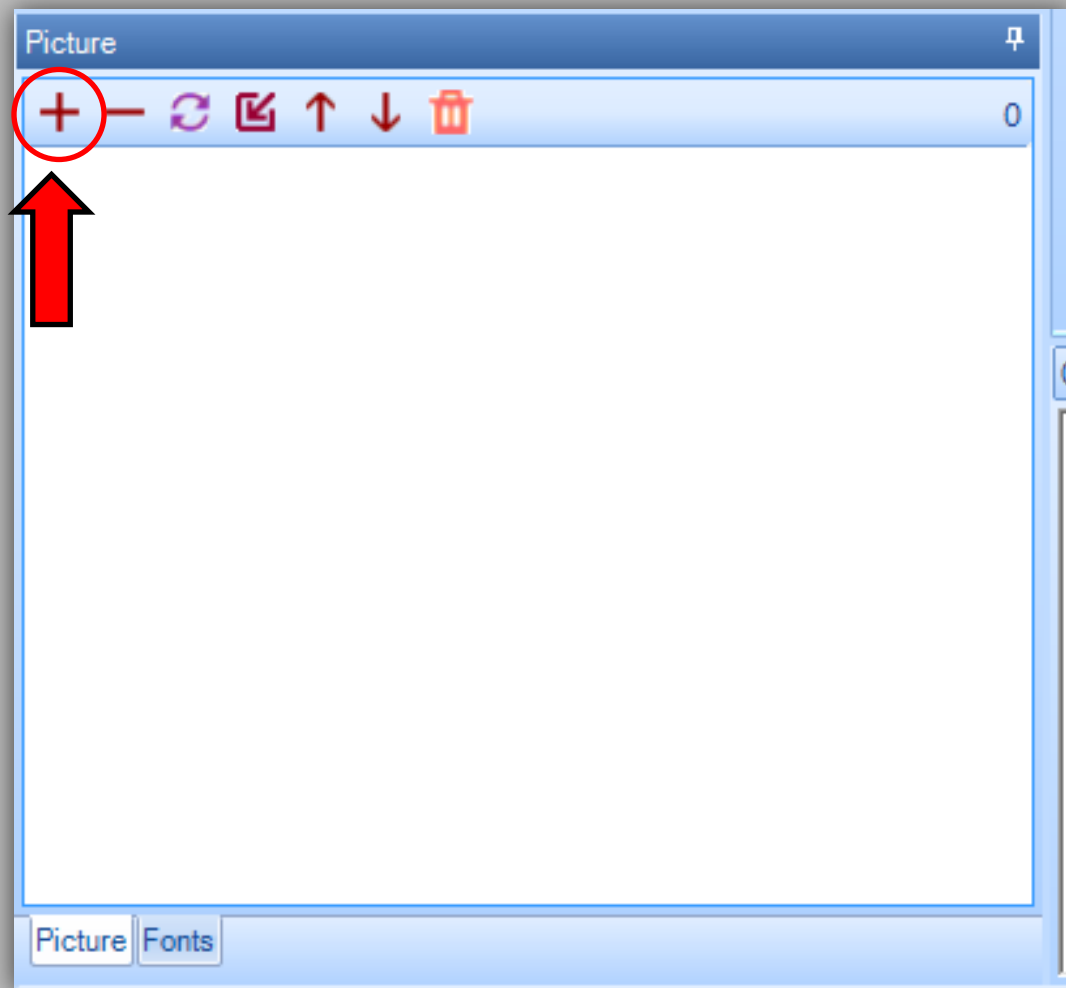


ID da fonte



Imagem

Para importar uma imagem para seu inventário, basta clicar no menu **add**, e escolher a imagem.

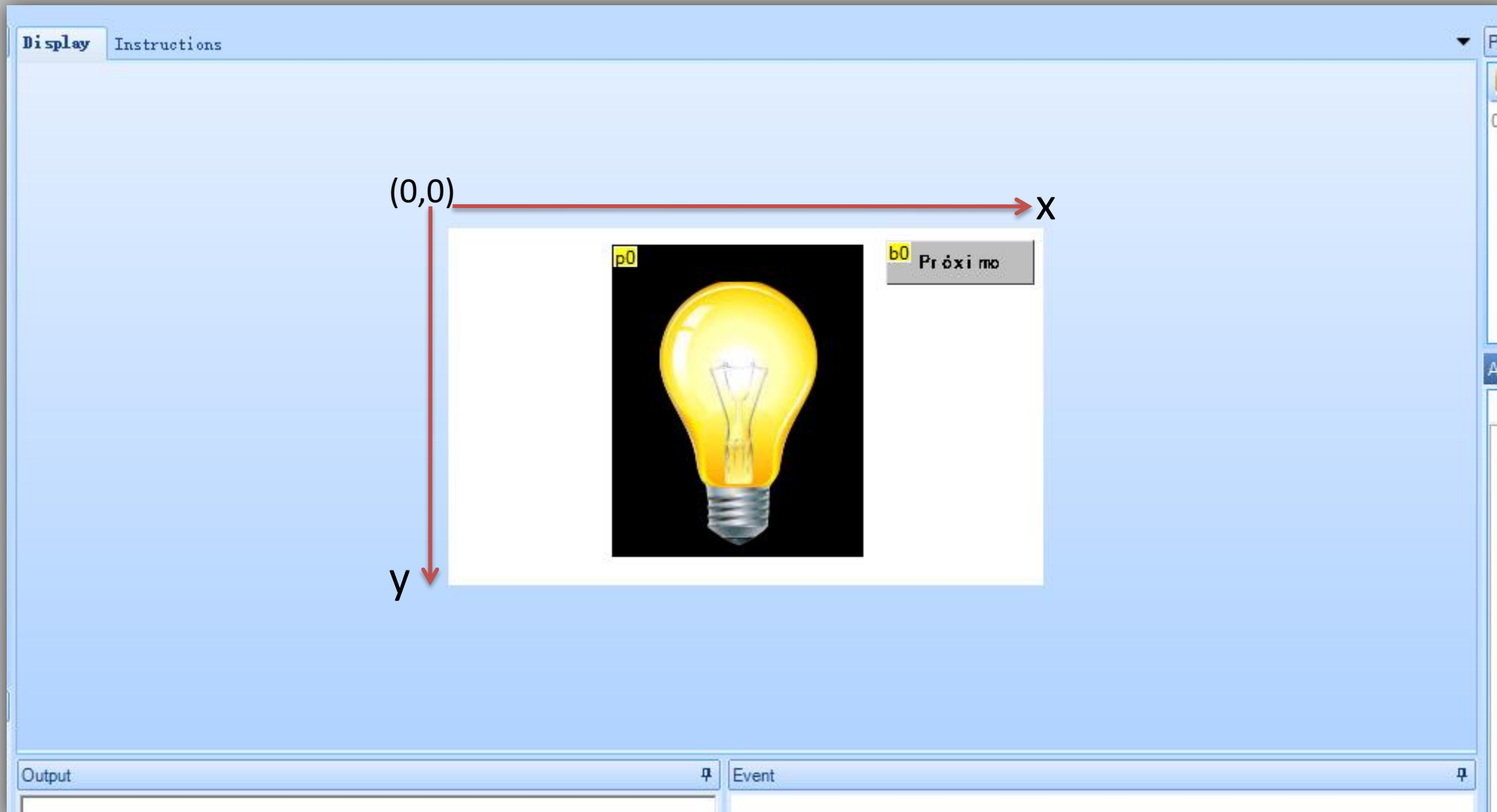


ID da imagem



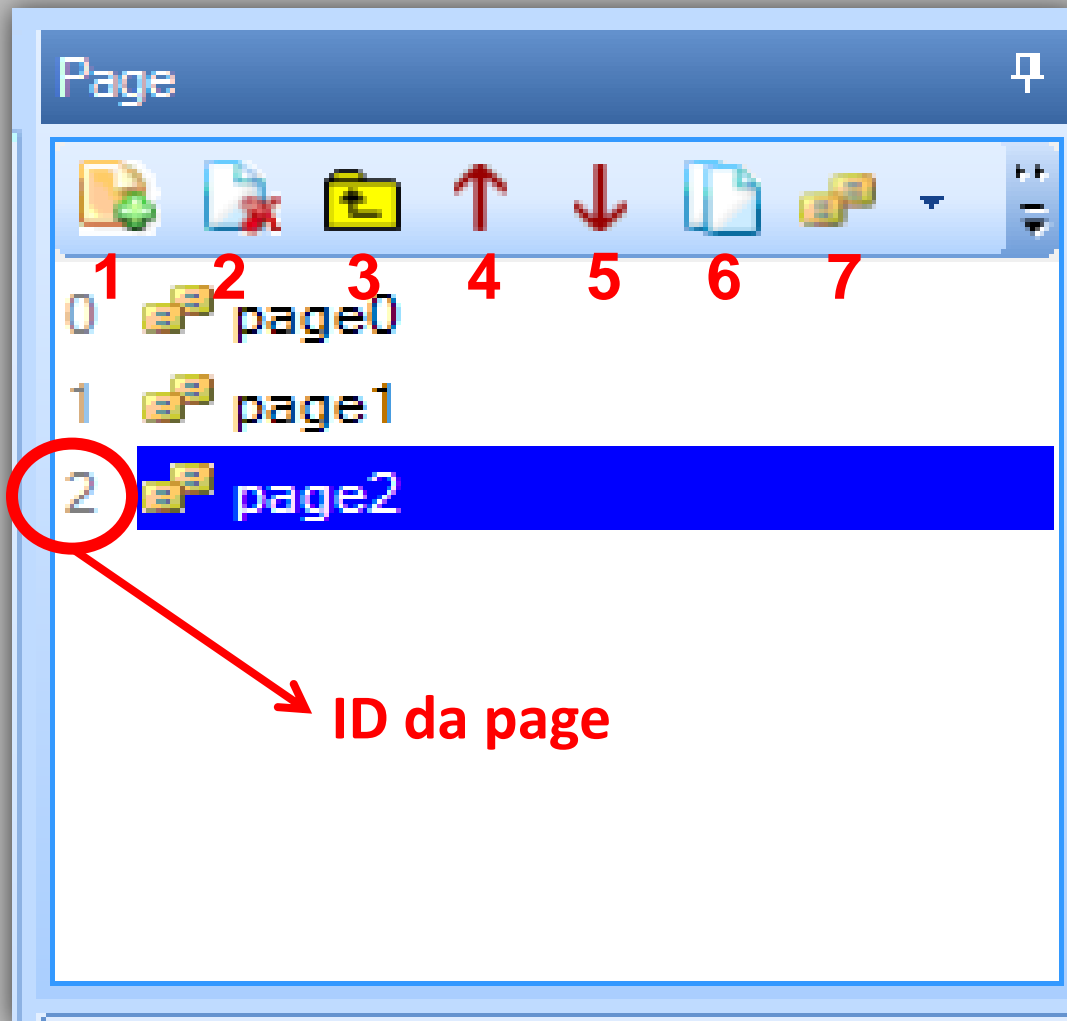
Área de Exposição

Abaixo a imagem da área de exposição com os componentes **PICTURE** e **BUTTON** inseridos.



Área de Páginas

PAGE é o equivalente a cada “tela” que teremos em nosso projeto.



1. Adicionar nova página ao final da lista
2. Remover página
3. Inserir página (entre páginas)
4. Mover página ordem acima
5. Mover página ordem abaixo
6. Copiar página
7. Importar/exportar página



Área de Edição de Atributos

Ao clicar em um componente, podemos editar seus atributos. Abaixo vemos os atributos da “picture” e do “button” que adicionamos na área de exposição.

Attribute

b0(Button)

id	2
objname	b0
type	98
vscope	local
sta	solid color
style	3D_Auto
bco	48631
bco2	1024
pco	0
pco2	0
font	1
xcen	Center
ycen	Center
txt	Próximo
txt_maxl	10
isbr	False
spax	0
spay	0
x	294
y	8
w	100
h	30

Click the attribute to display corresponding notes

ID do objeto

Nome do objeto

ID da fonte

Attribute

p0(Picture)

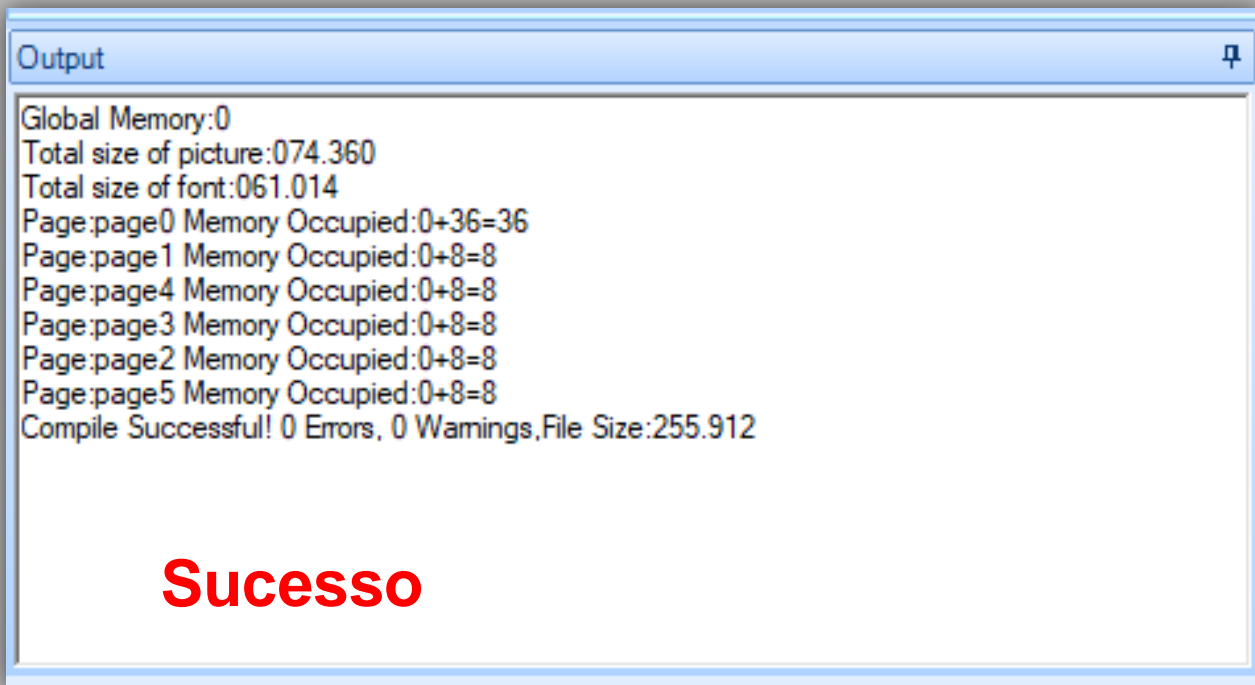
id	1
objname	p0
type	112
vscope	local
pic	0
x	110
y	11
w	169
h	210

ID da imagem



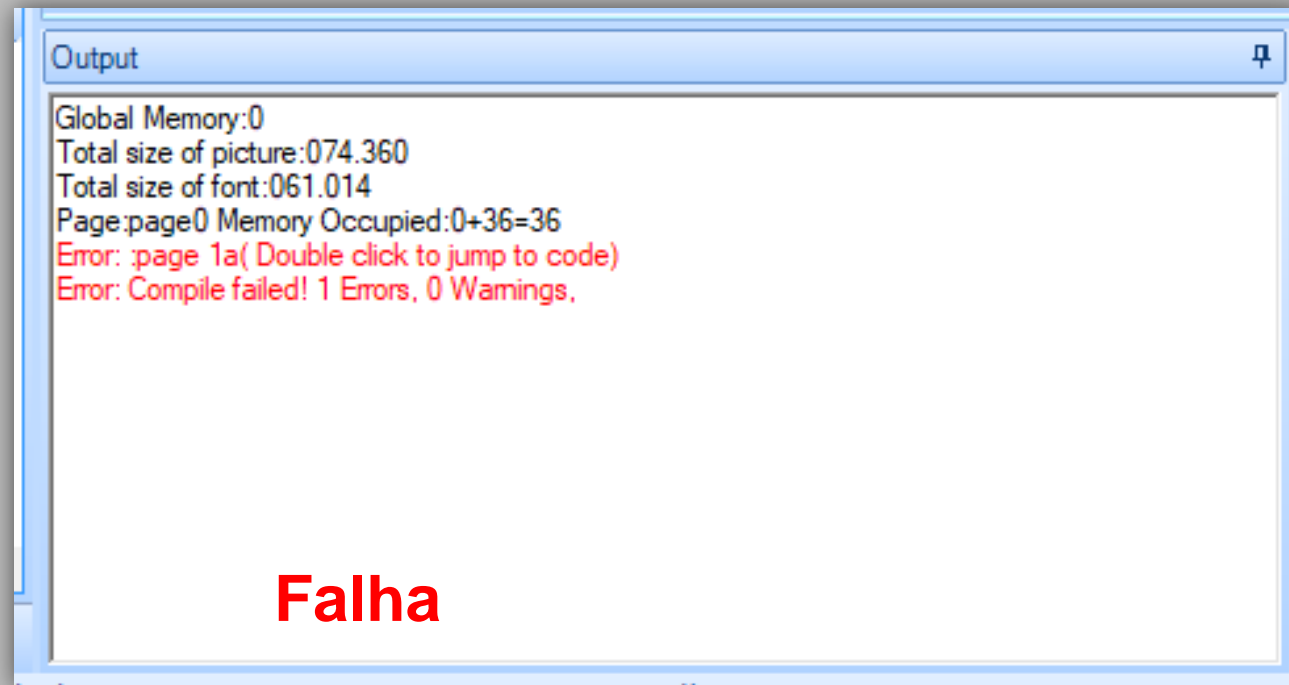
Área de Saída de Compilação

Ao clicar no botão de compilação, teremos os resultados na área de saída.

A screenshot of a software window titled "Output" with a pin icon in the top right corner. The window contains the following text: "Global Memory:0", "Total size of picture:074.360", "Total size of font:061.014", "Page:page0 Memory Occupied:0+36=36", "Page:page1 Memory Occupied:0+8=8", "Page:page4 Memory Occupied:0+8=8", "Page:page3 Memory Occupied:0+8=8", "Page:page2 Memory Occupied:0+8=8", "Page:page5 Memory Occupied:0+8=8", and "Compile Successful! 0 Errors, 0 Warnings,File Size:255.912". At the bottom of the window, the word "Sucesso" is written in large red font.

```
Output
Global Memory:0
Total size of picture:074.360
Total size of font:061.014
Page:page0 Memory Occupied:0+36=36
Page:page1 Memory Occupied:0+8=8
Page:page4 Memory Occupied:0+8=8
Page:page3 Memory Occupied:0+8=8
Page:page2 Memory Occupied:0+8=8
Page:page5 Memory Occupied:0+8=8
Compile Successful! 0 Errors, 0 Warnings,File Size:255.912

Sucesso
```

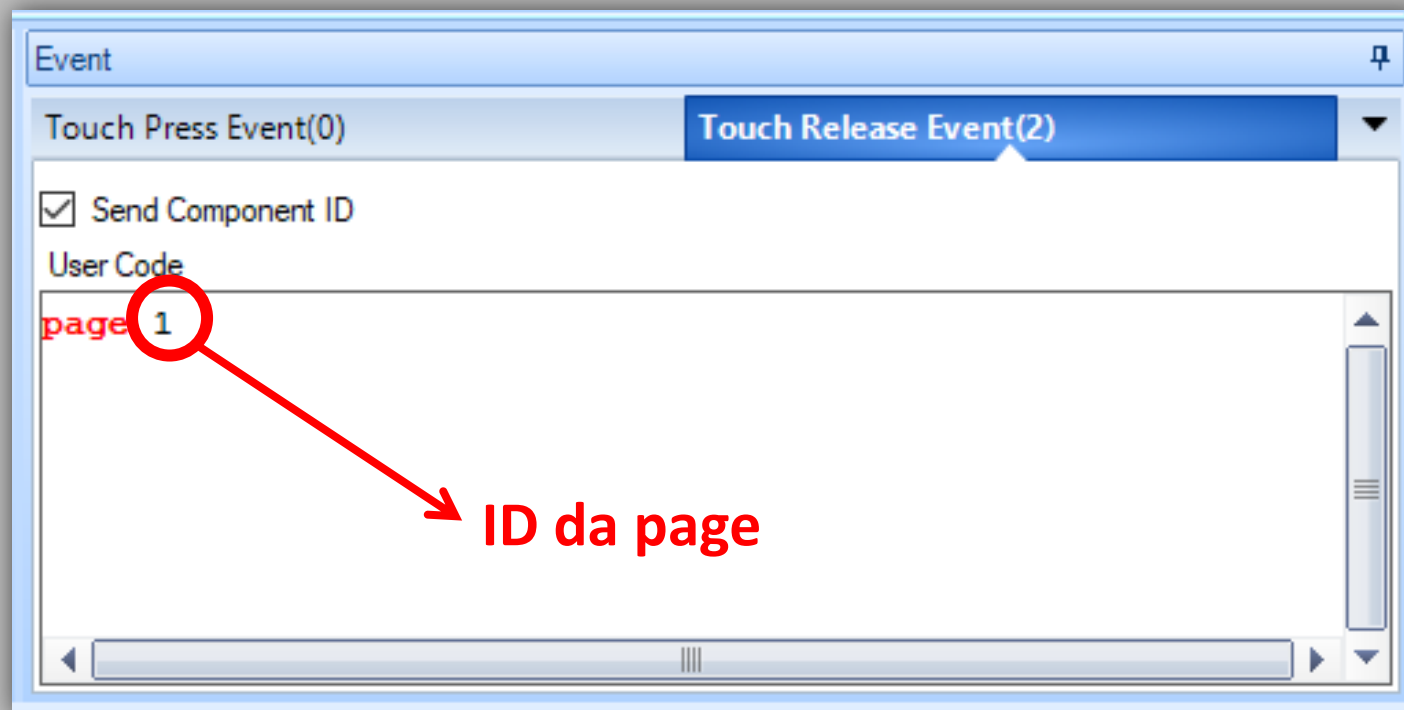
A screenshot of a software window titled "Output" with a pin icon in the top right corner. The window contains the following text: "Global Memory:0", "Total size of picture:074.360", "Total size of font:061.014", "Page:page0 Memory Occupied:0+36=36", "Error: :page 1a(Double click to jump to code)", and "Error: Compile failed! 1 Errors, 0 Warnings,". At the bottom of the window, the word "Falha" is written in large red font.

```
Output
Global Memory:0
Total size of picture:074.360
Total size of font:061.014
Page:page0 Memory Occupied:0+36=36
Error: :page 1a( Double click to jump to code)
Error: Compile failed! 1 Errors, 0 Warnings,

Falha
```


Área de Eventos

Na área de eventos, podemos configurar os eventos dos componentes (touch press, touch release,...) e também adicionar códigos para execução ao ocorrer o evento.



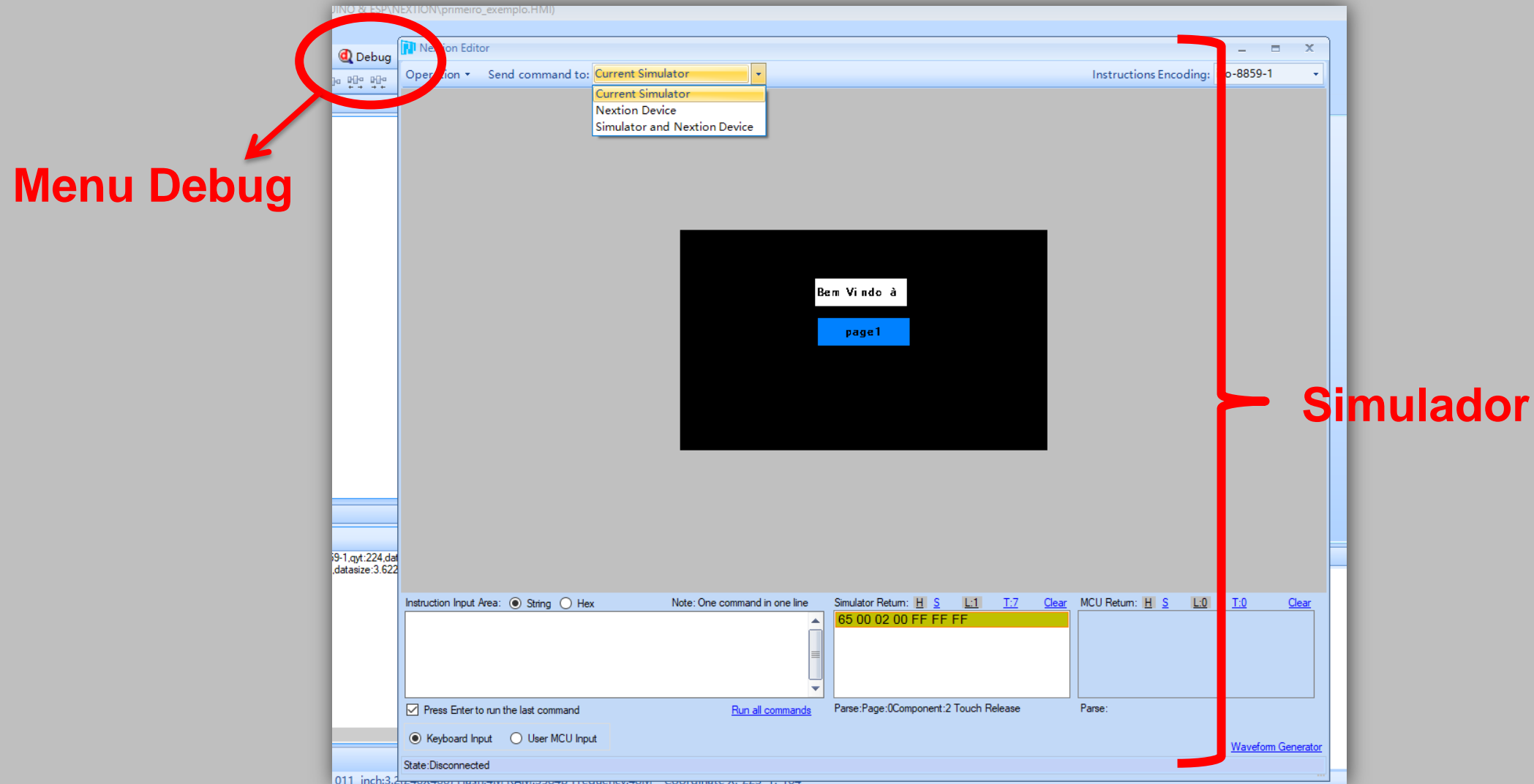
Evento release do botão

A imagem acima, mostra que ao ocorrer o evento release a “**página 1**” será chamada, ou seja, a **page1** então será mostrada em tela. Importante notar que o número **1** refere-se ao ID da página destino.



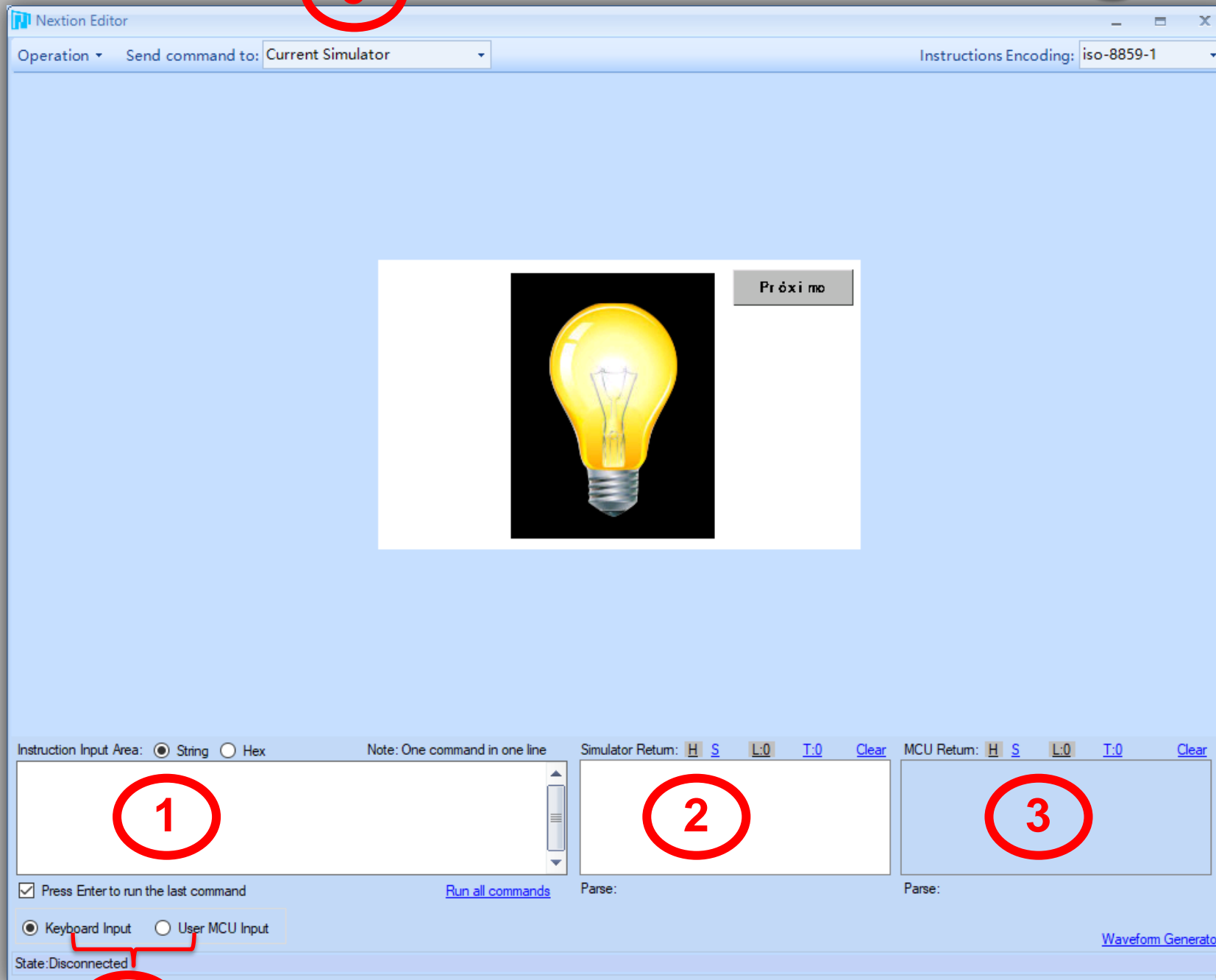
Debug

Ao clicar no menu **Debug** uma janela para simularmos nosso projeto será aberta. Podemos utilizar o microcontrolador com o código e esse simulador por exemplo. Não precisamos necessariamente de um dispositivo físico do Nextion.



Debug

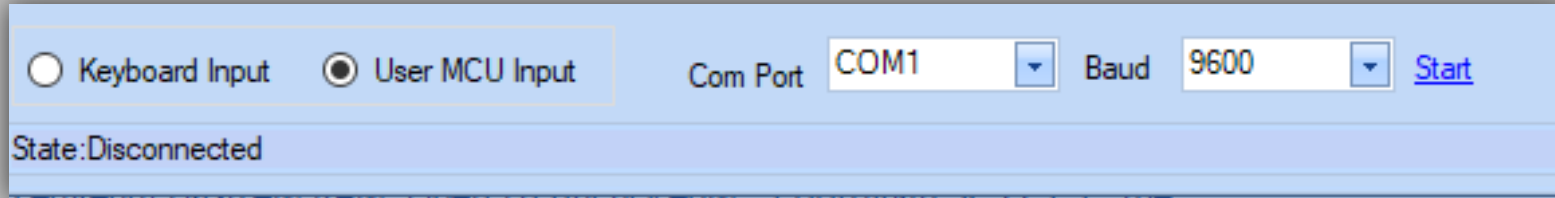
5



1. Área para envio de comandos para o simulador
2. Área de retorno do simulador
3. Área de comandos do microcontrolador
4. Modo de entrada de comandos (keyboard Input ou User MCU Input)
5. Destino do comando (Simulador, Nextion, Simulador e Nextion)



Debug com Microcontrolador

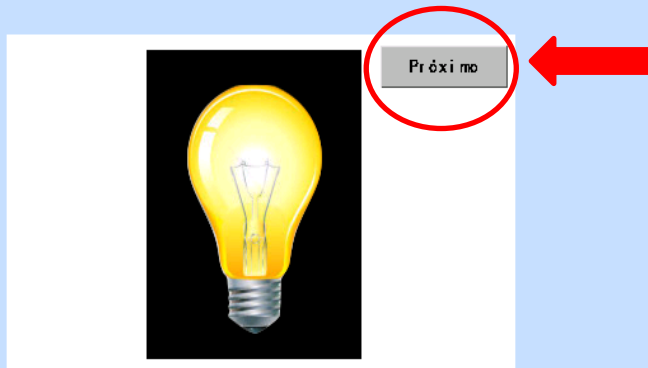


The screenshot shows a software interface for debugging a microcontroller. It features two radio buttons: 'Keyboard Input' (unselected) and 'User MCU Input' (selected). To the right, there are two dropdown menus: 'Com Port' set to 'COM1' and 'Baud' set to '9600'. A 'Start' button is located to the right of the baud rate dropdown. Below these controls, a status bar displays 'State:Disconnected'.

Ao clicar na opção **User MCU Input**, dois novos campos de configurações são abertos, um para definir a porta **COM** do **microcontrolador** e outro para definir a **velocidade de comunicação**. A partir daí os comandos serão enviados pelo microcontrolador, e o simulador receberá.

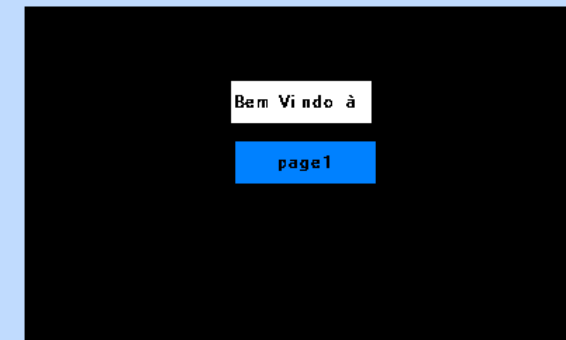
Simulando com o Debug

Ao clicar no botão próximo, a page1 será chamada. (Lembrando que definimos no evento release do botão o código “**page 1**”)



page0

Então o que foi adicionado a “**page1**”, será exibido.

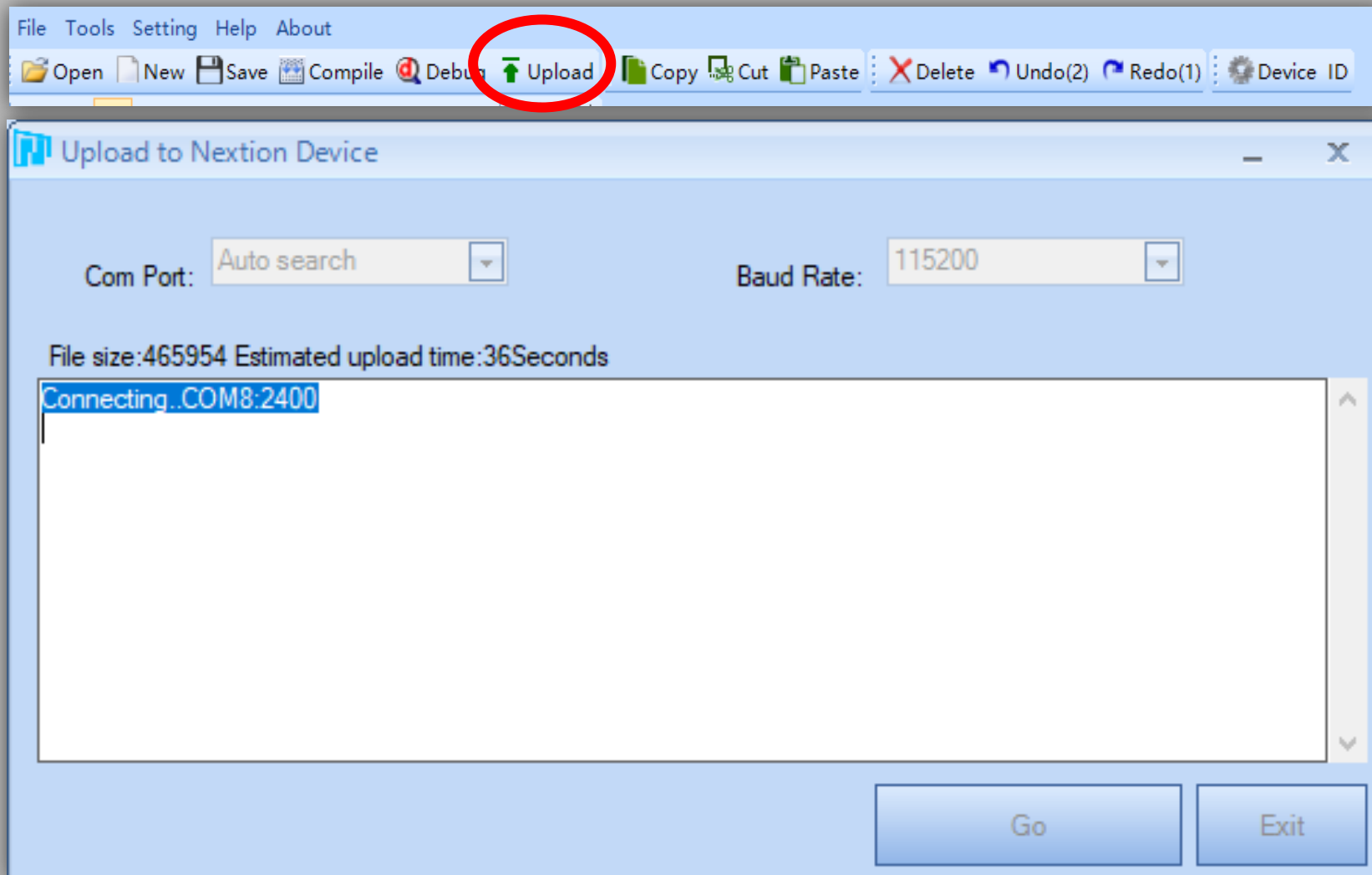


page1

Upload

Para gravar o projeto no Nextion, podemos fazer de duas maneiras:

1. **Menu Upload**
2. Upload através do SD Card



Com Port: ou você escolhe a porta COM do seu Nextion, ou deixa em **Auto search**, que o programa se encarrega de testar todas e em diferentes velocidades.

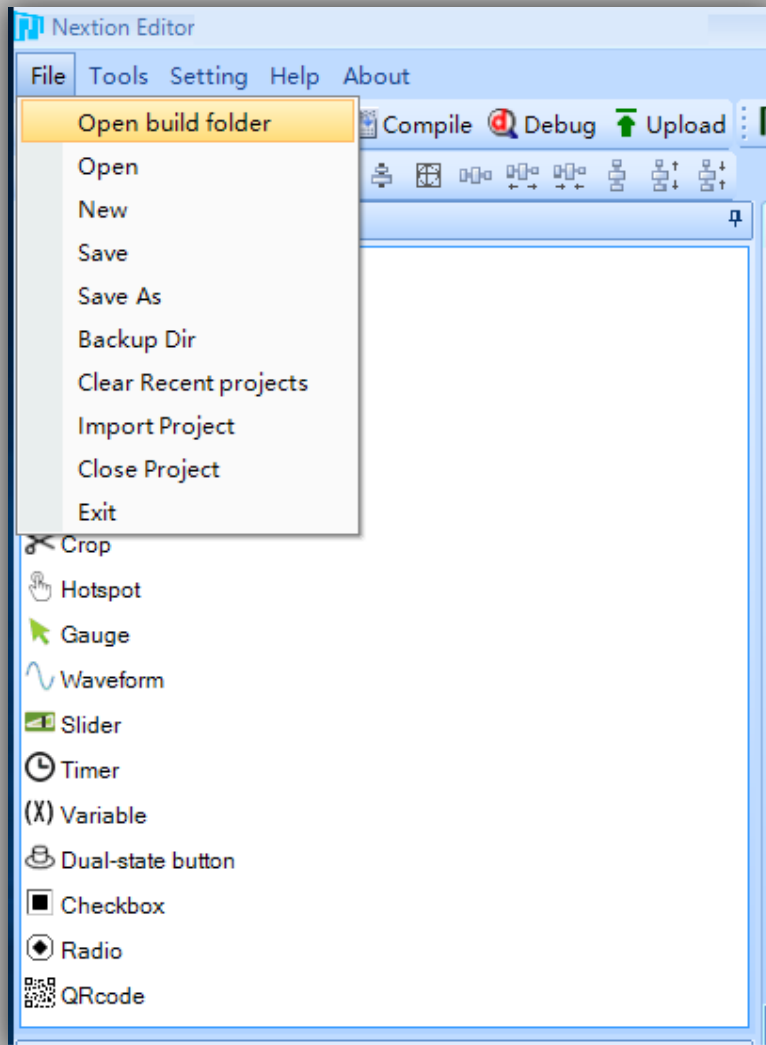
Baud Rate: escolha a velocidade de upload. Caso a escolha da **Com Port** tenha sido **Auto search**, o programa tentará todas as velocidades disponíveis até encontrar a correta.



Upload

1. Menu Upload

2. Upload através do SD Card

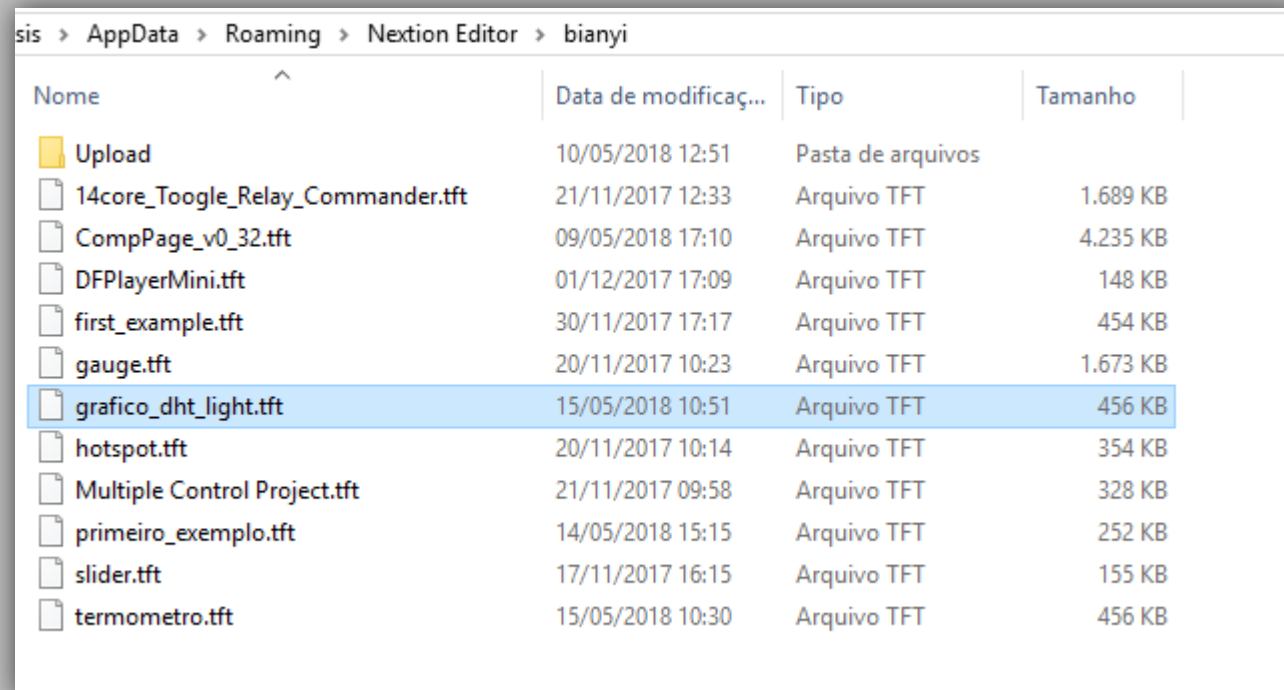


2.1: Acesse o menu **File >> Open build folder**

2.2: Copie o arquivo do seu projeto (extensão **.tft**) e então cole dentro do seu SD Card que utilizará no Nextion.

Observação 1: no SD deve estar apenas o arquivo do seu projeto.

Observação 2: depois de copiado o arquivo dentro do SD Card, coloque-o no Nextion. Ao ser alimentado, ele identificará o arquivo e então fará o “upload” para sua memória. Feito isso, aguarde a mensagem de sucesso e então retire o SD Card.

A screenshot of a Windows File Explorer window showing the contents of the Nextion Editor's build folder. The path is 'This PC > AppData > Roaming > Nextion Editor > bianyi'. The window displays a list of files and folders with columns for 'Nome', 'Data de modificação', 'Tipo', and 'Tamanho'. The file 'grafico_dht_light.tft' is selected.

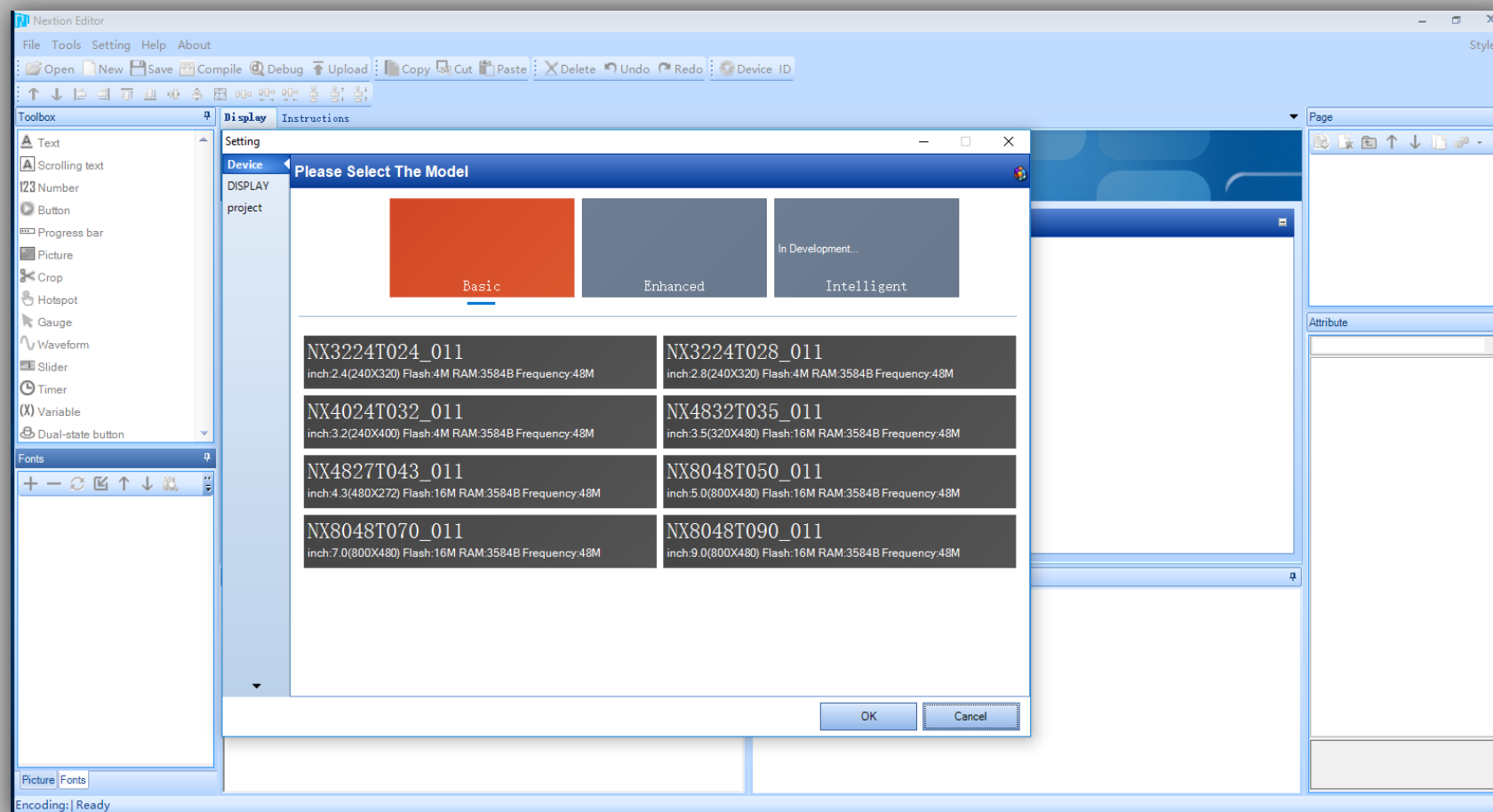
Nome	Data de modificação	Tipo	Tamanho
Upload	10/05/2018 12:51	Pasta de arquivos	
14core_Toogole_Relay_Commander.tft	21/11/2017 12:33	Arquivo TFT	1.689 KB
CompPage_v0_32.tft	09/05/2018 17:10	Arquivo TFT	4.235 KB
DFPlayerMini.tft	01/12/2017 17:09	Arquivo TFT	148 KB
first_example.tft	30/11/2017 17:17	Arquivo TFT	454 KB
gauge.tft	20/11/2017 10:23	Arquivo TFT	1.673 KB
grafico_dht_light.tft	15/05/2018 10:51	Arquivo TFT	456 KB
hotspot.tft	20/11/2017 10:14	Arquivo TFT	354 KB
Multiple Control Project.tft	21/11/2017 09:58	Arquivo TFT	328 KB
primeiro_exemplo.tft	14/05/2018 15:15	Arquivo TFT	252 KB
slider.tft	17/11/2017 16:15	Arquivo TFT	155 KB
termometro.tft	15/05/2018 10:30	Arquivo TFT	456 KB



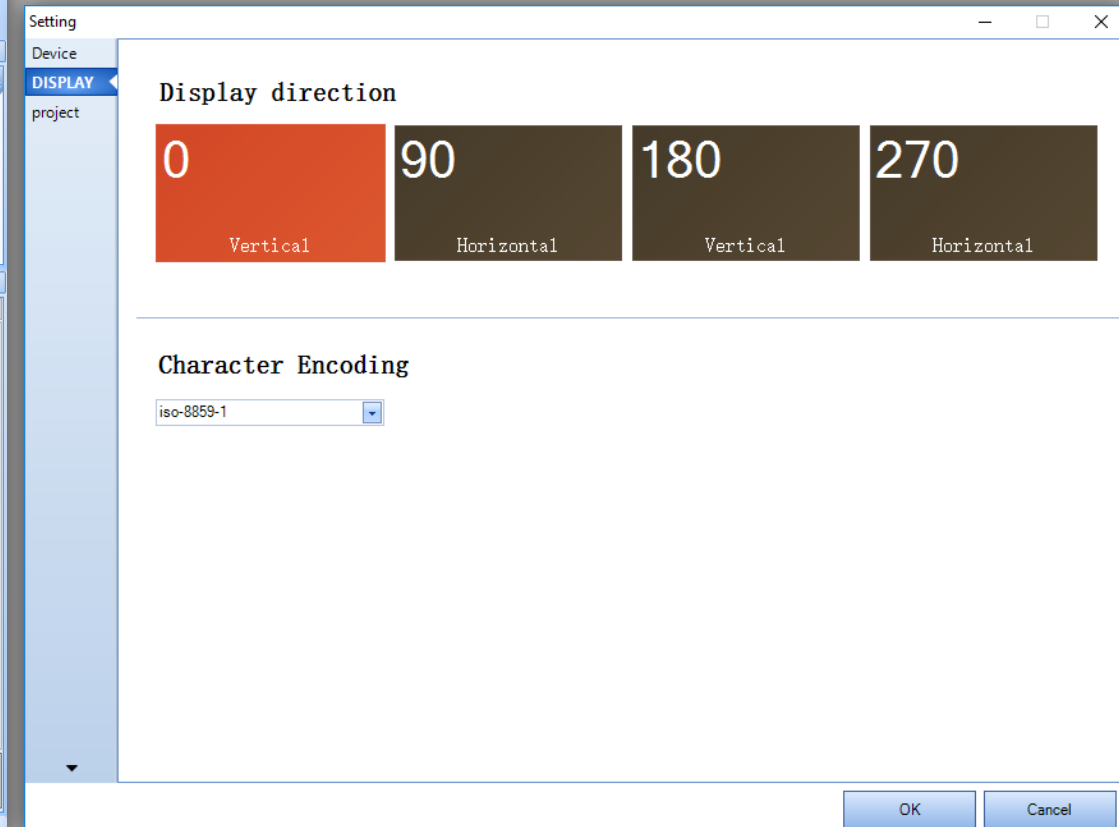
Novo Projeto

Ao criar um novo projeto as seguintes janelas aparecerão, na primeira (1) você escolhe o modelo de Nxtion e na segunda (2) a orientação das telas.

(1)

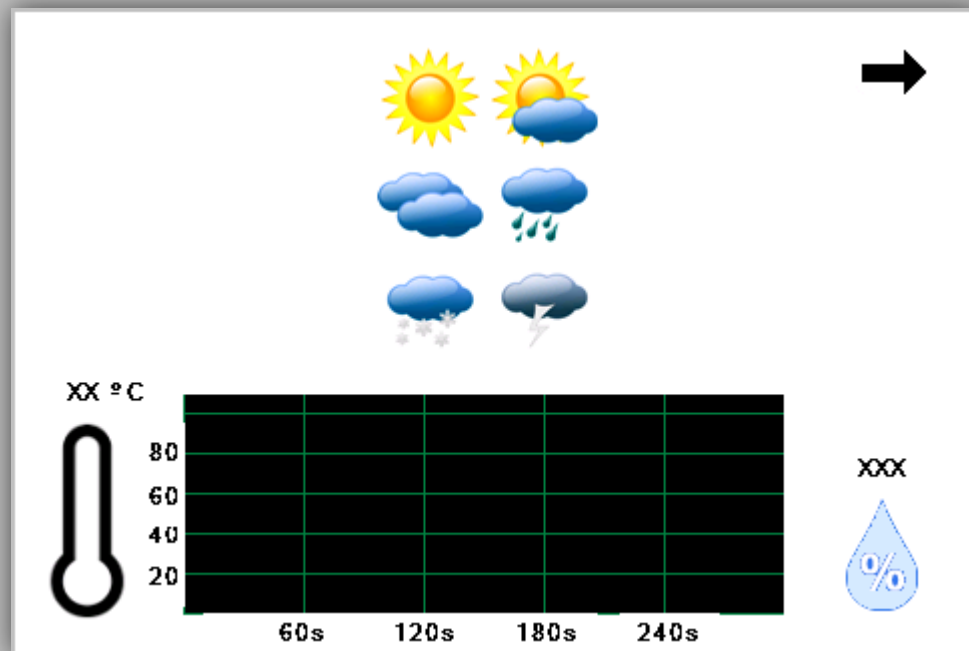


(2)



Exemplo

Faremos agora um exemplo com duas telas, na qual uma mostrará os dados de um sensor DHT22 e a outra para controlarmos um LED e seu PWM.

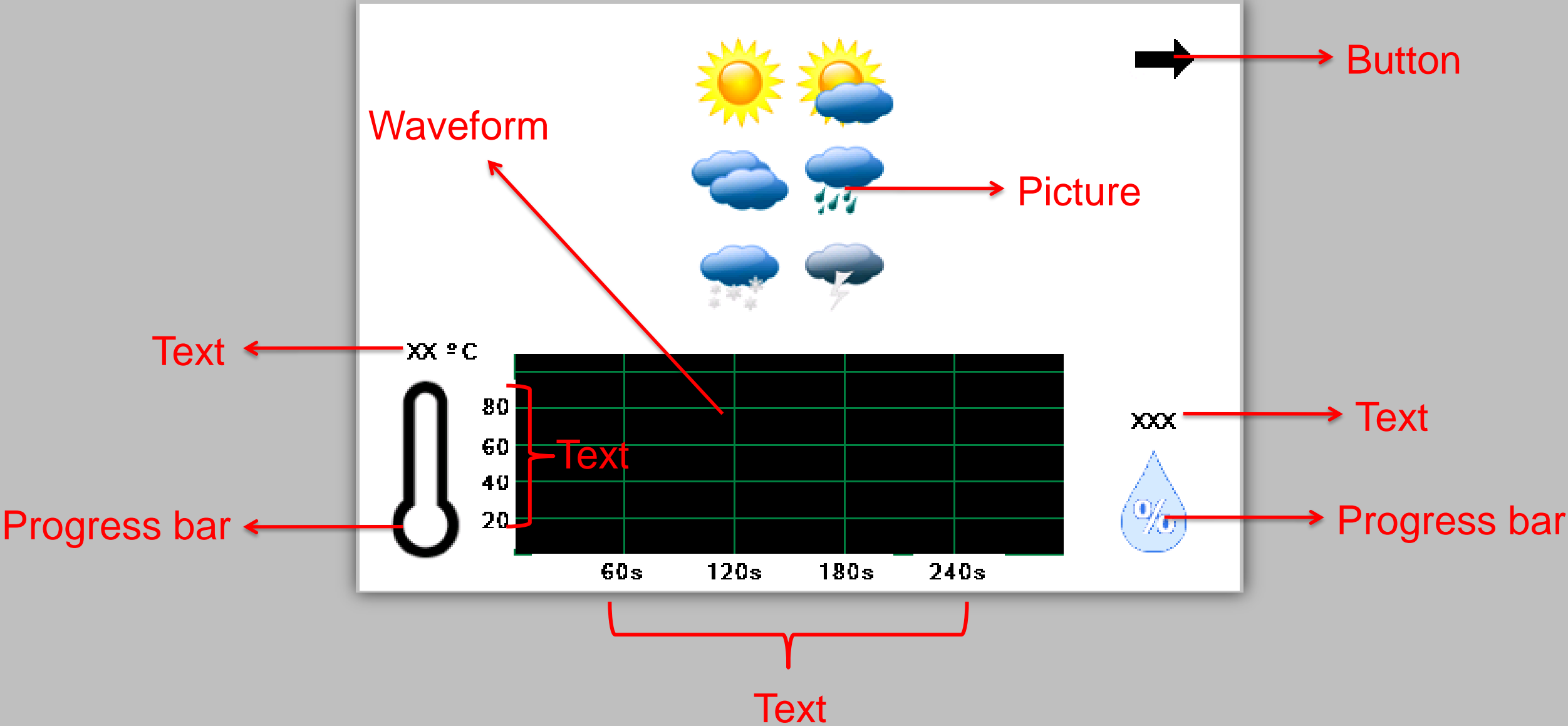


page0

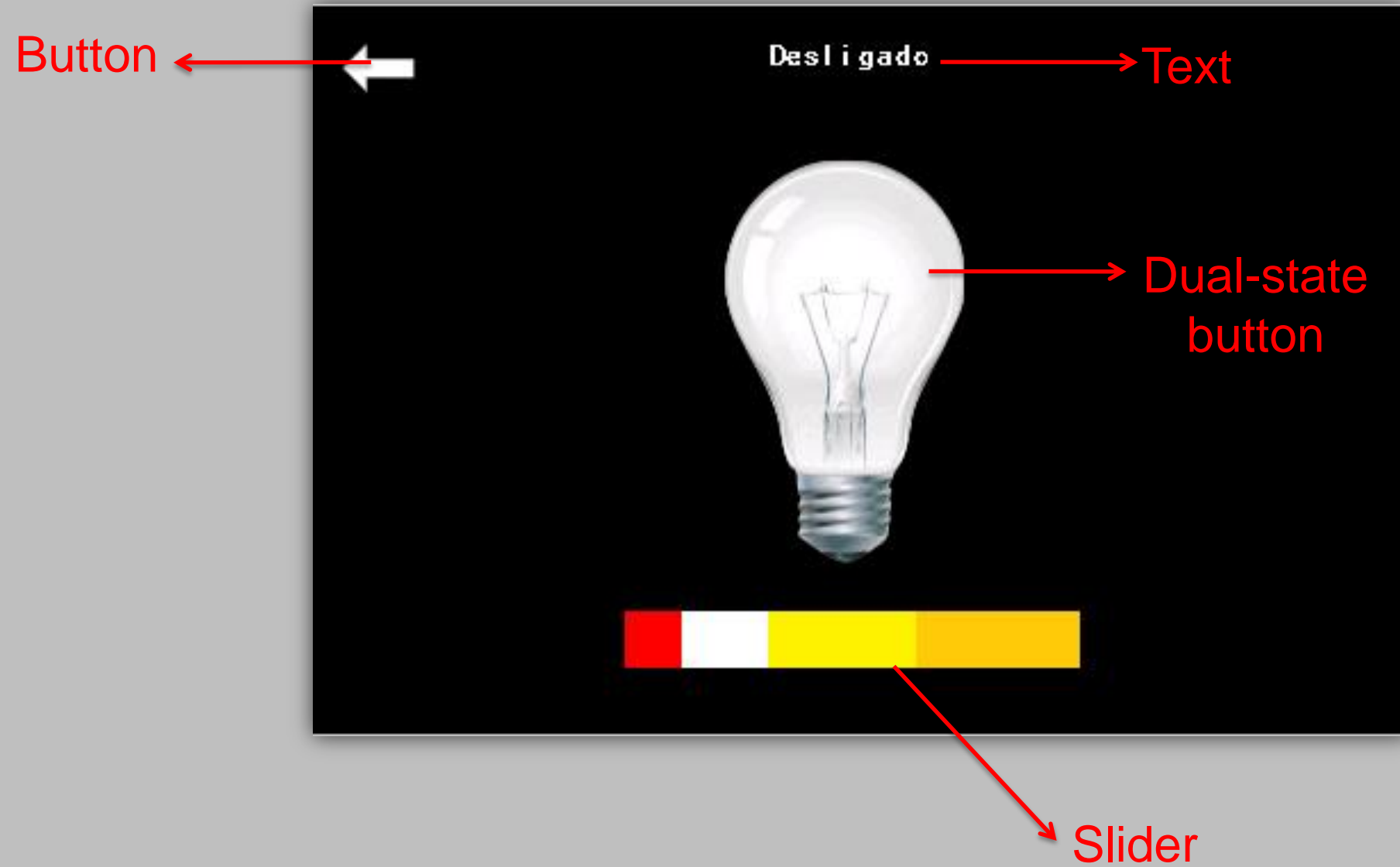


page1

page0



page1

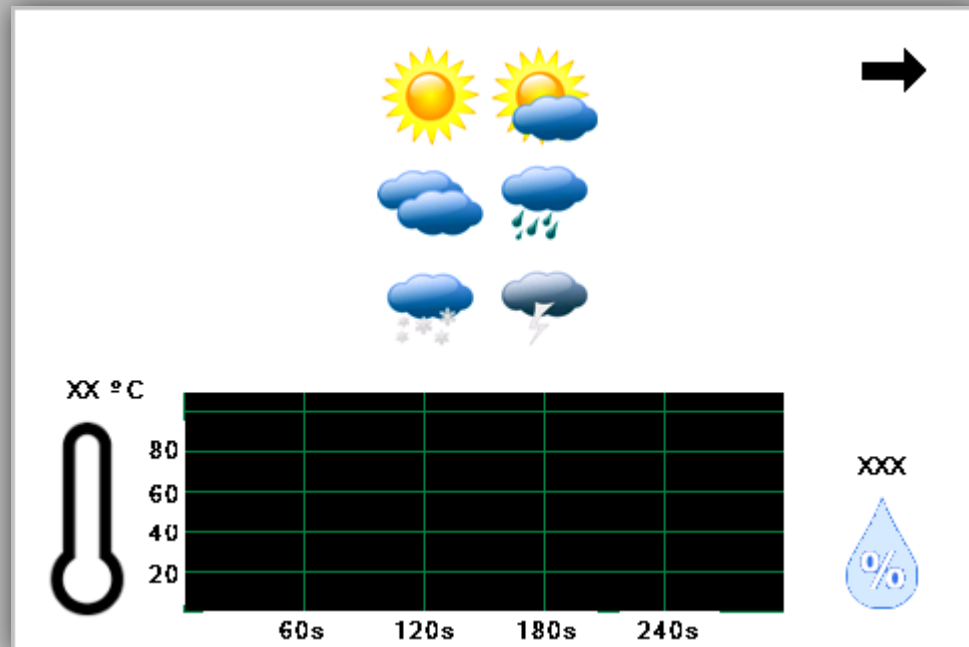


Projeto

O arquivo do projeto pode ser baixado através do link abaixo:

[download: grafico_dht_light.HMI](#)

Basta abri-lo pelo Nextion Editor para visualizá-lo.



Bibliotecas

Adicione biblioteca “ITEADLIB_Arduino_Nextion-master”.

Acesse o [link](#) e faça download da biblioteca.

Descompacte o arquivo e cole na pasta de bibliotecas da IDE do arduino.

C:/Program Files (x86)/Arduino/libraries



Observação

Antes de iniciarmos a codificação através da IDE Arduino, devemos realizar uma pequena alteração no arquivo **NexConfig.h** da biblioteca.

- * **Comente o debug serial**

`#define DEBUG_SERIAL_ENABLE` → `///define DEBUG_SERIAL_ENABLE`

- * **Modifique a definição de Serial2 como padrão Serial**

`#define nexSerial Serial2` → `#define nexSerial Serial`

Antes

```
/* enable debug serial */
#define DEBUG_SERIAL_ENABLE

/* define serial for debug */
#define dbSerial Serial

#ifdef DEBUG_SERIAL_ENABLE
#define dbSerialPrint(a)    dbSerial.print(a)
#define dbSerialPrintln(a) dbSerial.println(a)
#define dbSerialBegin(a)   dbSerial.begin(a)
#else
#define dbSerialPrint(a)    do{}while(0)
#define dbSerialPrintln(a) do{}while(0)
#define dbSerialBegin(a)   do{}while(0)
#endif

/* define serial for communicate with Nextion screen */
#define nexSerial Serial2
```

Depois

```
/* enable debug serial */
///define DEBUG_SERIAL_ENABLE

/* define serial for debug */
#define dbSerial Serial

#ifdef DEBUG_SERIAL_ENABLE
#define dbSerialPrint(a)    dbSerial.print(a)
#define dbSerialPrintln(a) dbSerial.println(a)
#define dbSerialBegin(a)   dbSerial.begin(a)
#else
#define dbSerialPrint(a)    do{}while(0)
#define dbSerialPrintln(a) do{}while(0)
#define dbSerialBegin(a)   do{}while(0)
#endif

/* define serial for communicate with Nextion screen */
#define nexSerial Serial
```

Código para o Microcontrolador

```
#include "Nextion.h" //biblioteca para comunicação com o display NEXTION
#include <SimpleDHT.h> //biblioteca para comunicação com o DHT22

SimpleDHT22 dht22;
static int pinDHT22 = 7; //pino que ligamos o DATA do DHT22 no arduino

//variáveis que armazenam os dados do DHT22
float temperature = 0;
float humidity = 0;

uint8_t pinLED = 3; //pino que ligamos o LED no arduino

bool isPage0 = true; //variável de controle para saber se é a página ZERO que está em
tela
bool isLedOn = false; //variável de controle para saber se o LED está aceso

char buffer[100] = {0}; //buffer para armazenar dados string

//variáveis de controle de tempo, para funcionar como um cronômetro
long previousMillis = 0;
const int maxTime = 1000;
```



Código para o Microcontrolador

```
/* DECLARAÇÕES DOS OBJETOS DO NEXTION: [pageID, componentID, componentName] */
NexProgressBar progTemp    = NexProgressBar(0, 1, "prog_temp"); //Progress Bar temperatura
NexProgressBar progHumid   = NexProgressBar(0, 16, "prog_humid"); //Progress Bar umidade
NexButton btnNext          = NexButton(0, 3, "btn_next"); //botão da página 0
NexText txtTemp            = NexText(0, 2, "txt_temp"); //texto com o valor da temperatura
NexText txtHumid           = NexText(0, 5, "txt_humid"); //texto com o valor da umidade
NexWaveform waveform       = NexWaveform(0, 6, "waveform"); //componente do gráfico de temperatura e umidade

NexDSButton btnLight       = NexDSButton(1, 1, "btn_light"); //botão de duplo estado para controlar o LED
NexSlider slider           = NexSlider(1, 2, "slider_pwm"); //slider que controlará o PWM do LED
NexText statusPWM          = NexText(1, 4, "status_pwm"); //texto com o status do pwm
NexVariable valueSlider    = NexVariable(1, 5, "valueSlider"); //(objeto do nextion) variável que guarda o valor do slider
NexButton btnBack          = NexButton(1, 3, "btn_back"); //botão da página 1

NexPage page0 = NexPage(0, 0, "page0");
NexPage page1 = NexPage(1, 0, "page1");

//Nesse array, declaramos os objetos Nextion que terão interação de eventos touch
NexTouch *nex_listen_list[] =
{
    &btnLight,
    &slider,
    &valueSlider,
    &btnNext,
    &btnBack,
    NULL
};
```



Código para o Microcontrolador

```
void setup() {
    Serial.begin(9600);
    pinMode(pinLED, OUTPUT);

    nexInit(); //inicializa a comunicação com o nextion

    btnNext.attachPop(btnNextPopCallback, &btnNext); //callback para o evento de release do botão btnNext
    btnBack.attachPop(btnBackPopCallback, &btnBack); //callback para o evento de release do botão btnBack
    slider.attachPop(sliderPopCallback); //callback para o evento de release do slider
    btnLight.attachPop(btnLightPopCallback, &btnLight); //callback para o evento de release do botão de dois estados (btnLight)
}

void loop() {
    //essa função trabalha como um listener para os eventos de press e release dos objetos utilizados no NEXTION
    nexLoop(nex_listen_list);
    //verifica se a página atual é a ZERO(inicial)
    //caso seja, verifica se o tempo passado entra a última passagem e a atual, foi de 1 segundo
    //e então chama a atualização do gráfico de temperatura e umidade
    if(isPage0)
    {
        if(millis() - previousMillis >= maxTime)
        {
            previousMillis = millis();
            updateDHTInfo(); //pega os valores do DHT22 e então manda para o NEXTION
        }
    }
}
```



updateDHTInfo

```
//função que recupera os valores de temperatura e umidade do DHT22
//e através dos objetos do NEXTION seta as informações
void updateDHTInfo()
{
    int status = dht22.read2(pinDHT22, &temperature, &humidity, NULL); //faz a leitura dos dados do DHT22
    //se conseguiu ler os dados corretamente
    if (status == SimpleDHTErrSuccess)
    {
        progTemp.setValue(temperature); //progress bar da temperatura
        memset(buffer, 0, sizeof(buffer)); // posiciona o
        itoa(temperature, buffer, 10);
        txtTemp.setText(buffer); //texto que indica a temperatura

        progHumid.setValue(humidity); //progress bar da umidade
        memset(buffer, 0, sizeof(buffer));
        itoa(humidity, buffer, 10);
        txtHumid.setText(buffer); //texto que indica a umidade

        waveform.addValue(0, temperature); //envia o valor da temperatura para o gráfico
        waveform.addValue(1, humidity); //envia o valor da umidade para o gráfico
    }
}
```



CALLBACKS

```
//callback do SLIDER, quando o evento de release acontecer essa função é chamada
void sliderPopCallback(void *ptr)
{
    //se o LED está ligado, busca o valor da variável que guarda o valor do slider e então seta o PWM do LED
    if(isLedOn)
    {
        uint32_t number = 0;
        valueSlider.getValue(&number); //pega o valor atual da variável nextion
        int value = map(number, 0,100,0,255);
        analogWrite(pinLED, value);
        memset(buffer, 0, sizeof(buffer));
        itoa(value, buffer, 10);
        statusPWM.setText(buffer);
    }
}

//callback do botão btnNext
void btnNextPopCallback(void *ptr)
{
    page1.show();
    isPage0 = false;
    sliderPopCallback(ptr); //chama o callback do slider para reconfigura-lo
}

//callback do botão btnBack
void btnBackPopCallback(void *ptr)
{
    page0.show();
    isPage0 = true;
}
```



CALLBACKS

```
/* Callback do botão de dois estados */
void btnLightPopCallback(void *ptr)
{
    uint32_t dual_state;

    /* recupera o valor do estado de botão de dois estados */
    btnLight.getValue(&dual_state);
    if(dual_state)
    {
        isLedOn = true;
        sliderPopCallback(ptr);
    }
    else
    {
        statusPWM.setText("DESLIGADO");
        digitalWrite(pinLED, LOW);
        isLedOn = false;
    }
}
```



Em www.fernandok.com

Download arquivos PDF e **INO** do código fonte

