TouchGFX e STM32

Aplicações gráficas incríveis e fáceis de implementar
Plataforma de baixo custo

KARIMEX

Dennys Ramos

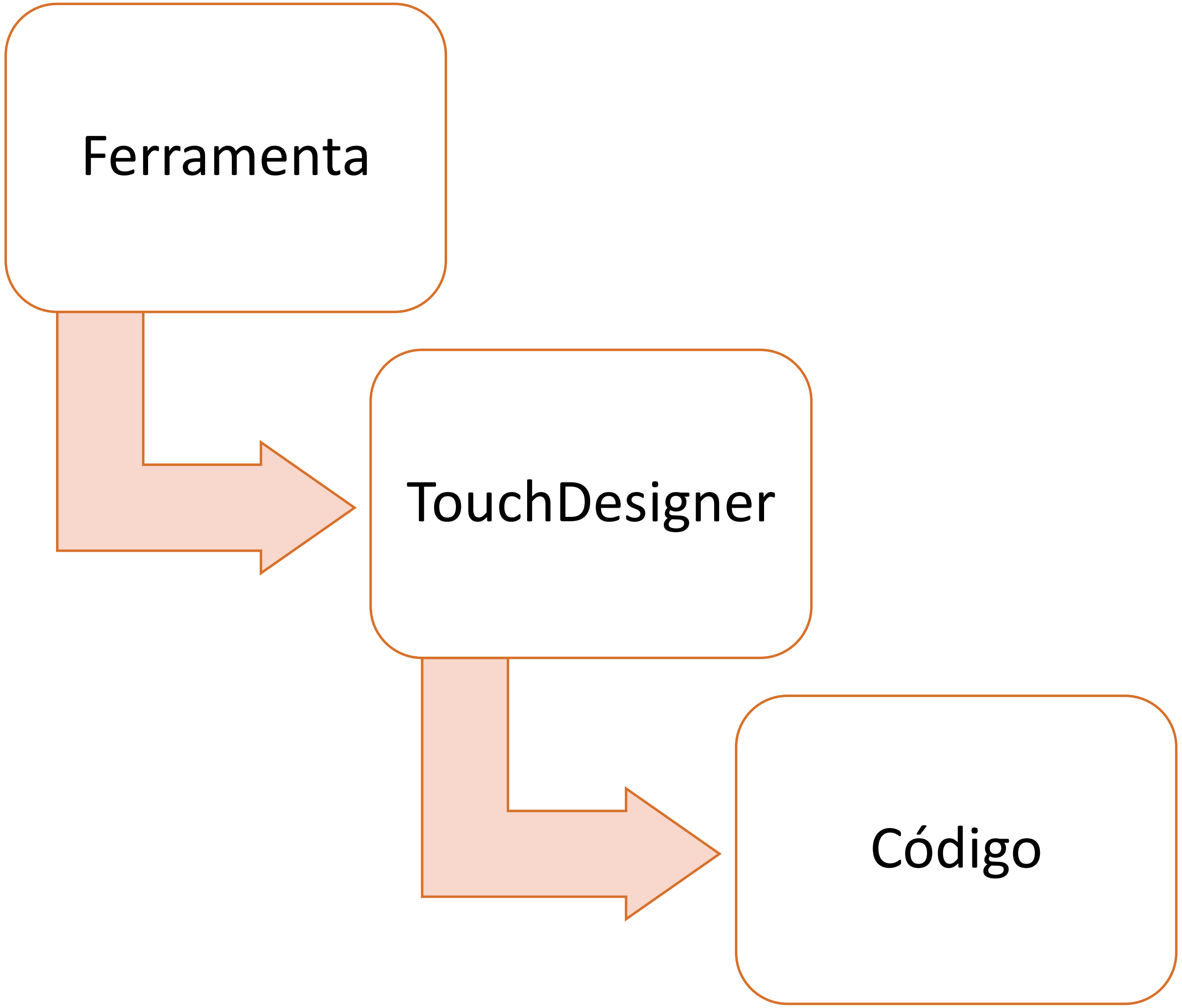
linkedin.com/in/dennysde/

dennys.ramos@karimex.com.br

(11) 98684-9087

Melhores IHMs para os embarcados

Agenda



Ferramenta

TouchDesigner

Código

Ferramenta de desenvolvimento

NUCLEO-G071RB
STM32G071
128K Flash
36K RAM
64MHz

X-NUCLEO-GFX01M1
Display SPI 240x320 2.2’
Memória Flash SPI

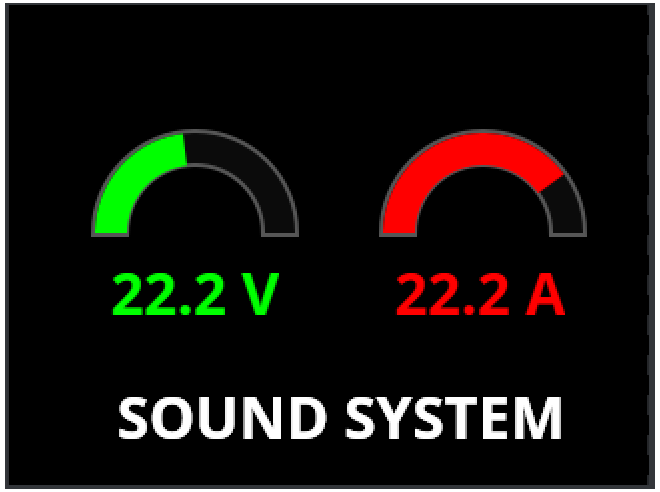
Cálculo de framebuffer

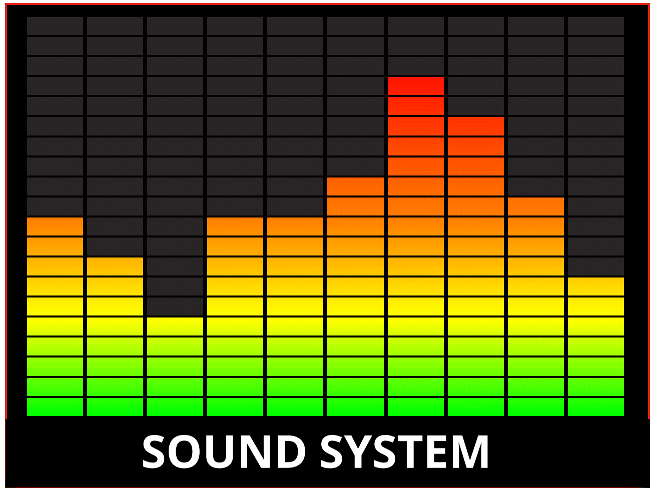
RAM = 240x320 x 16 / 8 = **153,6kBytes**

Framebuffer Parcial

Cost effective GUI Solution with TouchGFX and STM32G0 - YouTube

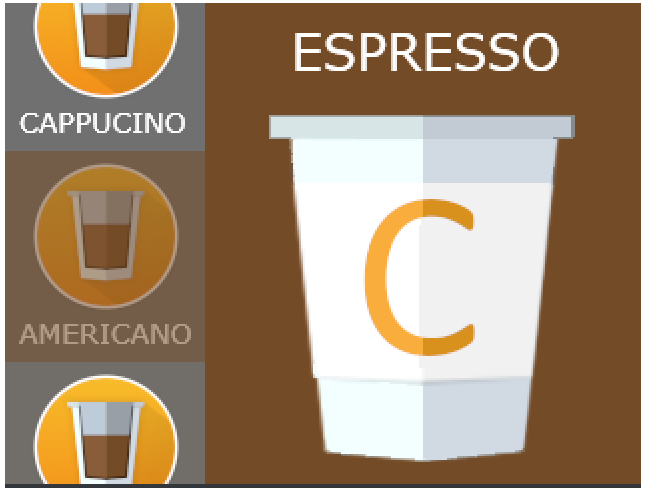
Aplicações com G071

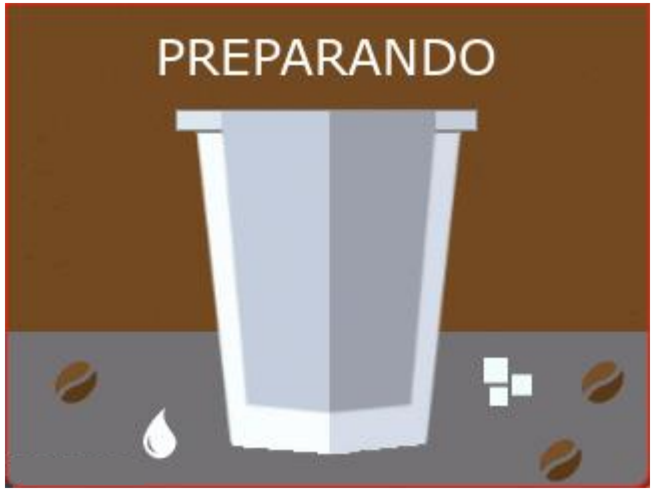


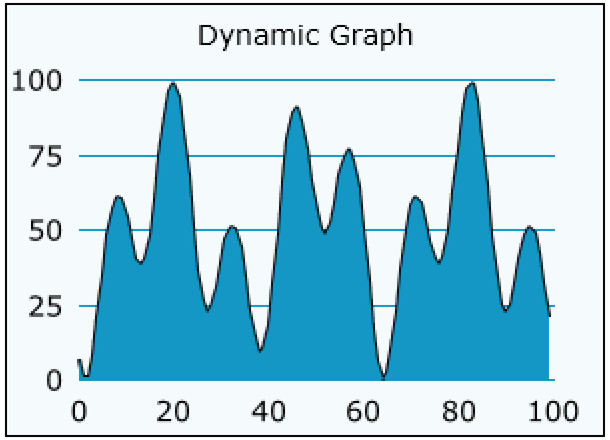


Medidores de energia
Amplificadores de áudio

Aplicações com G071

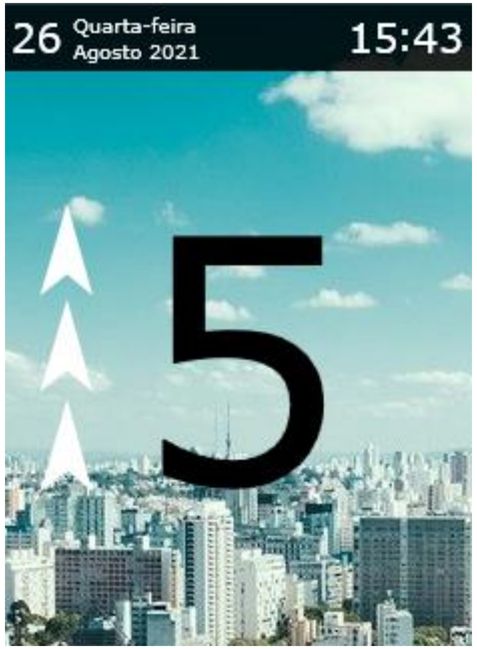


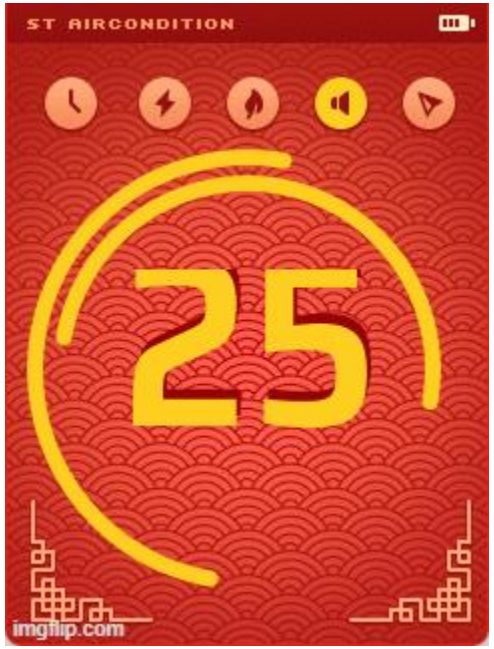


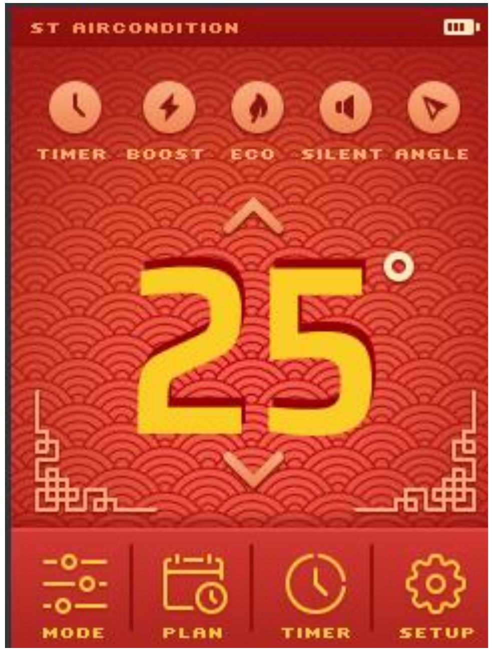


Animações fluidas
Relatórios gráficos

Aplicações com G071







Elevador e Ar condicionado

Ferramenta de desenvolvimento

NUCLEO-G071RB
STM32G071
128K Flash
36K RAM
64MHz

X-NUCLEO-GFX01M1
Display SPI 240x320 2.2'
Memória Flash SPI

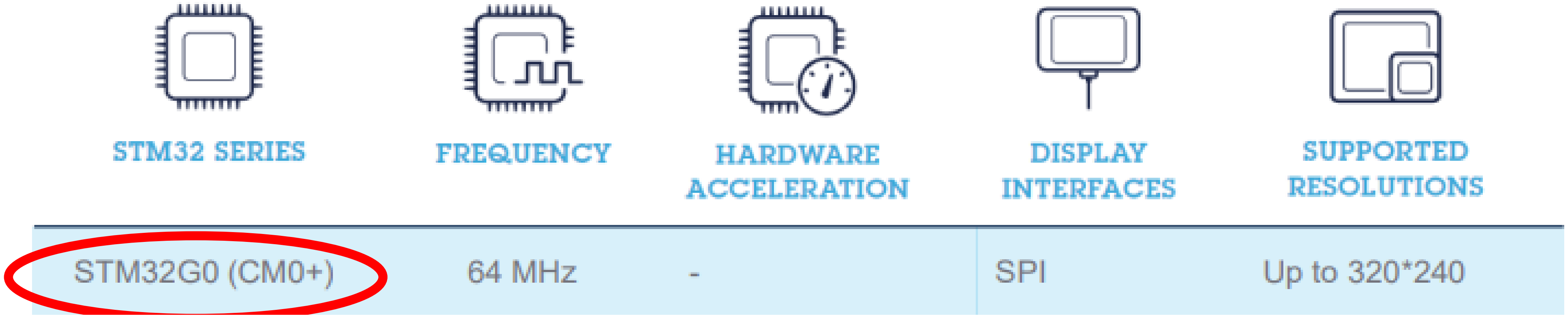
Cálculo de framebuffer

RAM = 240x320 x 16 / 8 = **153,6kBytes**

DMA
Motor Gráfico

Cost effective GUI Solution with TouchGFX and STM32G0 - YouTube

Plataforma de entrada



STM32 SERIES	FREQUENCY	HARDWARE ACCELERATION	DISPLAY INTERFACES	SUPPORTED RESOLUTIONS
STM32G0 (CM0+)	64 MHz	-	SPI	Up to 320*240

TouchGFXDesigner

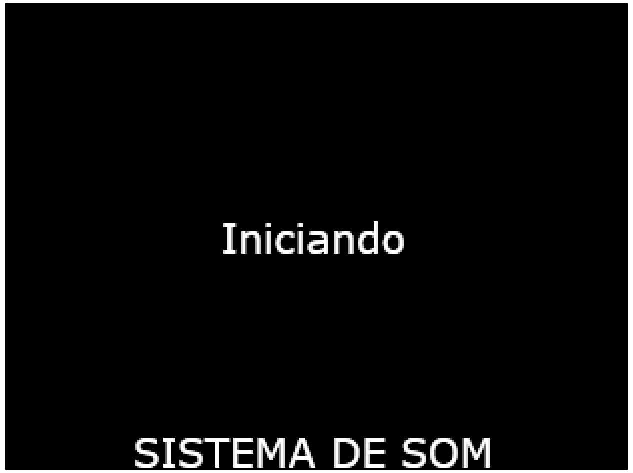
Aplicações gráficas incríveis e fáceis de implementar

Desenvolvendo do ZERO

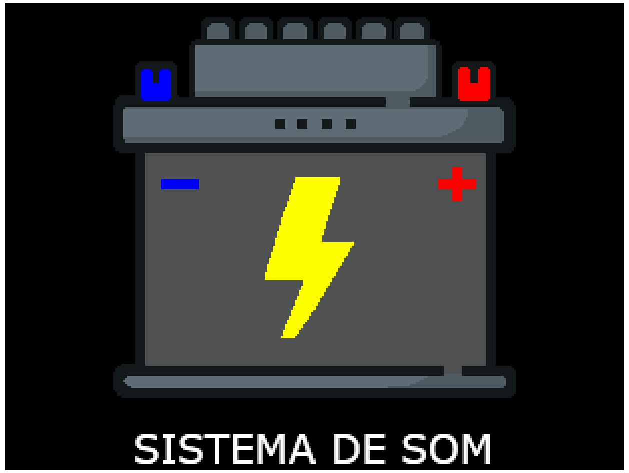
KARIMEX

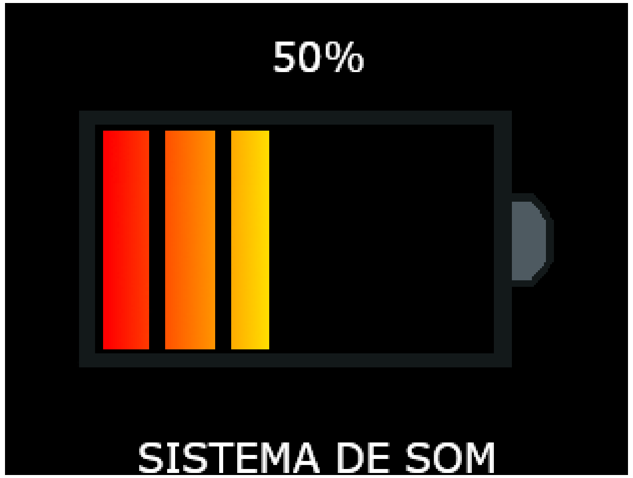
TouchGFX Designer

Tela 1

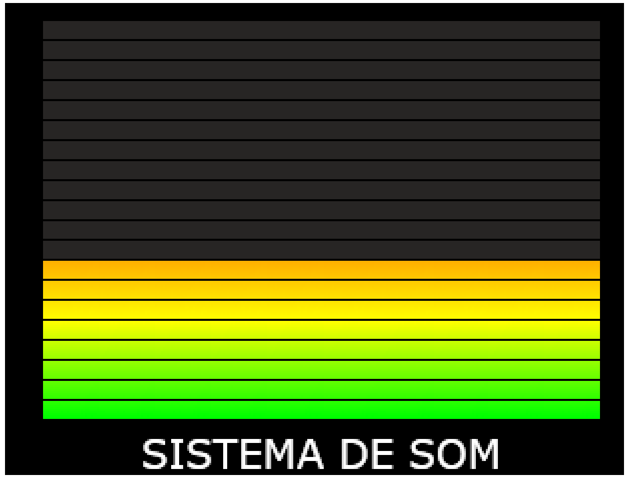


Tela 2





Tela 3



Tela 4

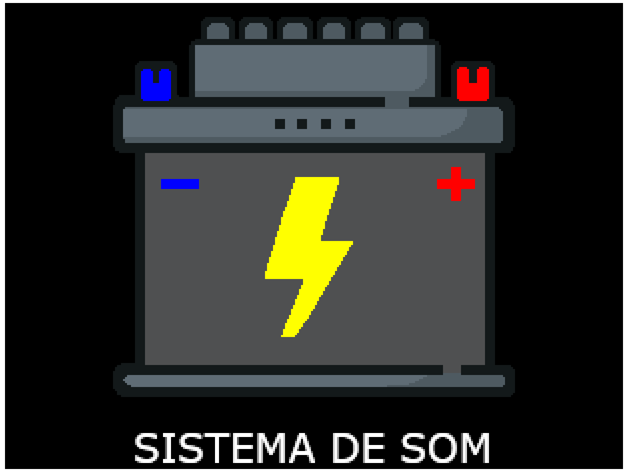
❑ Adicionar *custom container* para plano de fundo

❑ Transições por botão

```
void Screen2View::handleTickEvent()
{
tickCounter++;


if (tickCounter % 3 == 0) {  //Incremento a cada 50ms -> 60^-
1*3= 0,05s


currentValue = imageProgress1.getValue();
max = 65;
min = 20;


if (currentValue == min)
{
increase = true;
}
else if (currentValue == max)
{
increase = false;
}


int nextValue = increase == true ? (int)currentValue + 1 :
(int)currentValue - 1;


imageProgress1.setValue(nextValue);
}
}
```

Screen2View.cpp

Tela 2

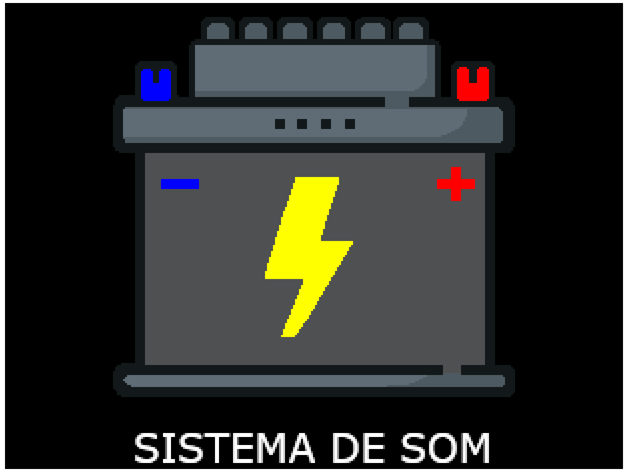


STM32CubeIDE

```
class Screen2View : public Screen2ViewBase
{
public:
    Screen2View();
    virtual ~Screen2View() {}
    virtual void setupScreen();
    virtual void tearDownScreen();
    virtual void handleTickEvent();
protected:
    uint16_t tickCounter =0;
    bool increase = true;
    double currentValue;
    uint8_t max;
    uint8_t min;
};
```

Screen2View.hpp

Tela 2

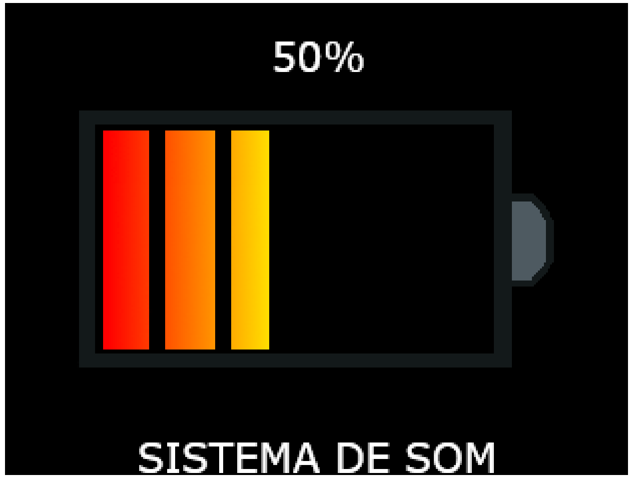


STM32CubeIDE

```
void Screen3View::handleTickEvent()
{
tickCounter++;

if (tickCounter % 6 == 0) {  //Incremento a cada 50ms -> 60^-1*8= 0,1s

currentValue = imageProgress1.getValue();
max = 100;
min = 0;

if (currentValue == min)
{
increase = true;
}
else if (currentValue == max)
{
increase = false;
}

int nextValue = increase == true ? (int)currentValue + 1 :
(int)currentValue - 1;

imageProgress1.setValue(nextValue);

Unicode::snprintf(textArea1Buffer, 4, "%d", nextValue);
textArea1.invalidate();
}
}
```

Screen3View.cpp

Tela 3

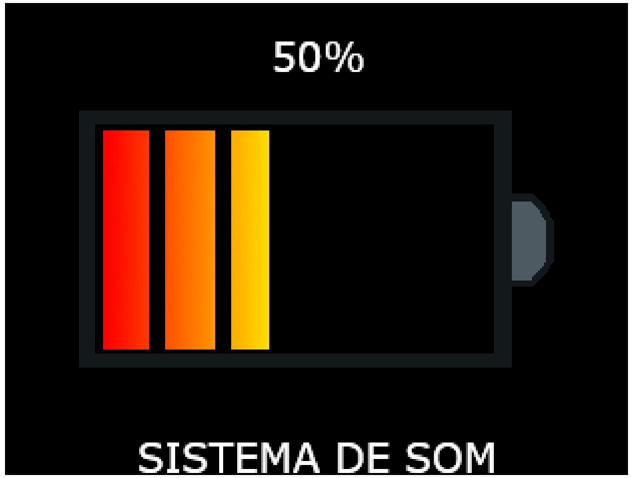


STM32CubeIDE

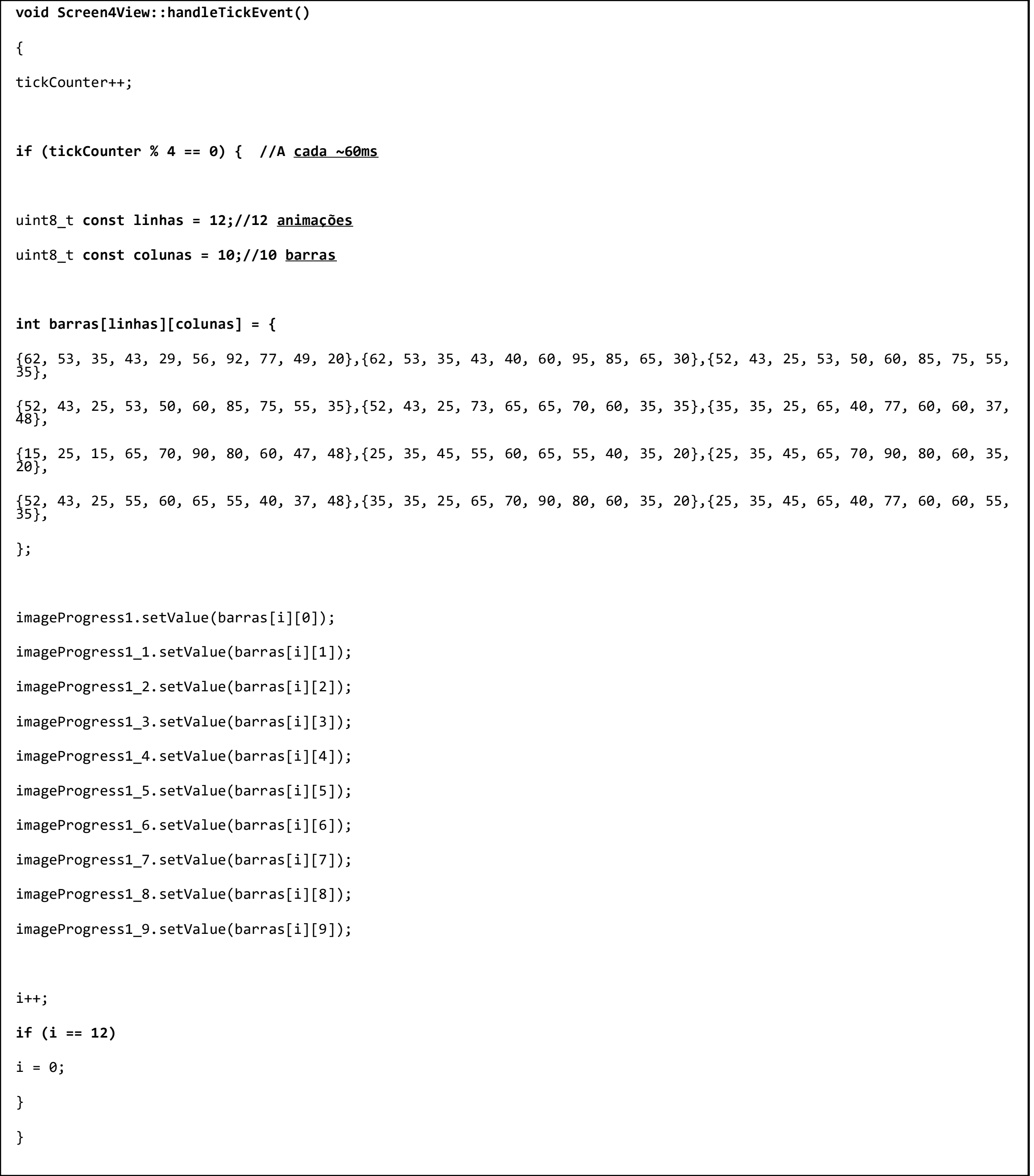
```
class Screen3View : public Screen3ViewBase
{
public:
    Screen3View();
    virtual ~Screen3View() {}
    virtual void setupScreen();
    virtual void tearDownScreen();
    virtual void handleTickEvent();
protected:
    uint16_t tickCounter =0;
    bool increase = true;
    double currentValue;
    uint8_t max;
    uint8_t min;
};
```

Screen3View.hpp

Tela 3



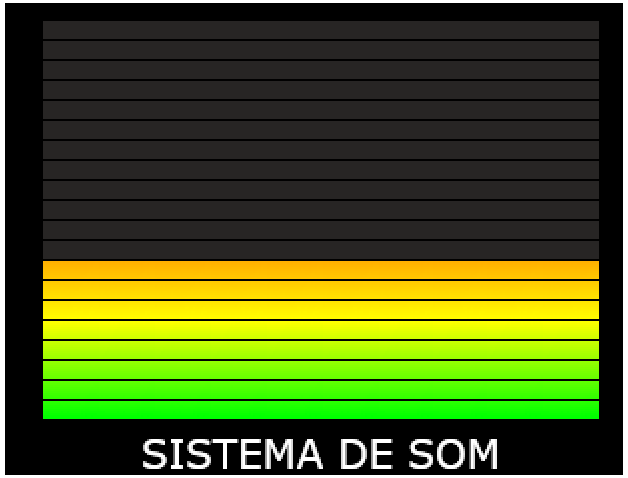
STM32CubeIDE



```
void Screen4View::handleTickEvent()

{

tickCounter++;


if (tickCounter % 4 == 0) {  //A cada ~60ms


uint8_t const linhas = 12;//12 animações

uint8_t const colunas = 10;//10 barras


int barras[linhas][colunas] = {

{62, 53, 35, 43, 29, 56, 92, 77, 49, 20},{62, 53, 35, 43, 40, 60, 95, 85, 65, 30},{52, 43, 25, 53, 50, 60, 85, 75, 55,
35},

{52, 43, 25, 53, 50, 60, 85, 75, 55, 35},{52, 43, 25, 73, 65, 65, 70, 60, 35, 35},{35, 35, 25, 65, 40, 77, 60, 60, 37,
48},

{15, 25, 15, 65, 70, 90, 80, 60, 47, 48},{25, 35, 45, 55, 60, 65, 55, 40, 35, 20},{25, 35, 45, 65, 70, 90, 80, 60, 35,
20},

{52, 43, 25, 55, 60, 65, 55, 40, 37, 48},{35, 35, 25, 65, 70, 90, 80, 60, 35, 20},{25, 35, 45, 65, 40, 77, 60, 60, 55,
35},

};


imageProgress1.setValue(barras[i][0]);

imageProgress1_1.setValue(barras[i][1]);

imageProgress1_2.setValue(barras[i][2]);

imageProgress1_3.setValue(barras[i][3]);

imageProgress1_4.setValue(barras[i][4]);

imageProgress1_5.setValue(barras[i][5]);

imageProgress1_6.setValue(barras[i][6]);

imageProgress1_7.setValue(barras[i][7]);

imageProgress1_8.setValue(barras[i][8]);

imageProgress1_9.setValue(barras[i][9]);


i++;

if (i == 12)

i = 0;

}

}
```

Screen4View.cpp

Tela 4

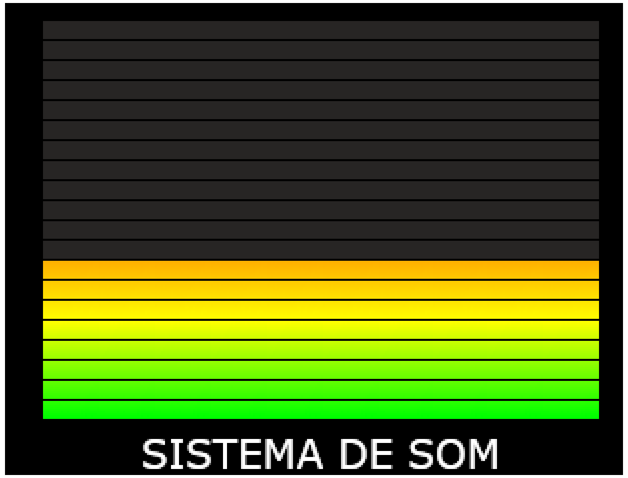


STM32CubeIDE

```
class Screen4View : public Screen4ViewBase
{
public:
    Screen4View();
    virtual ~Screen4View() {}
    virtual void setupScreen();
    virtual void tearDownScreen();
    virtual void handleTickEvent();
protected:
    uint16_t tickCounter =0;
    bool increase = true;
    double currentValue;
    uint8_t max;
    uint8_t min;
    uint8_t i=0;
};
```

Screen4View.hpp

Tela 4



STM32CubeIDE

Dennys Ramos

in linkedin.com/in/dennysde/

✉ dennys.ramos@karimex.com.br

☎ (11) 98684-9087

PT2: Desenvolvendo o próprio Hardware

- Botões
- Software
- Interface gráfica
- Código