



DESENVOLVENDO APLICAÇÕES EMBARCADAS COM QT E TORADEx

Cleiton - Bueno
cleiton.bueno@b2open.com





OLA!

Eu sou Cleiton
Bueno

Engenheiro, ciclista*, FOSS, e tudo isso
junto





1.

TORADIX

Qt

SoM Colibri i.MX6

i.MX6S - Solo Core, 256MB RAM e 4GB eMMC

i.MX6DL - Dual Core, 512MB RAM e 4GB eMMC

ARM Cortex-A9 (800MHz ~ 1GHz)

5x UART's

4x SPI

3x I2C

2x CAN

4x PWM

>150 GPIO's

GPU Vivante GC880

Video Decode (MJPEG, MPEG-4, H.264, H.263, DivX, VC1, MPEG-2)

Video Encode (MJPEG, MPEG-4, H.264, H.263)



BASEBOARD IRIS

Alimentação 6-27V DC

1x USB Host

1x USB OTG

1x I2C

3x UART's RS232

4x PWM

1x Ethernet

1x LVDS

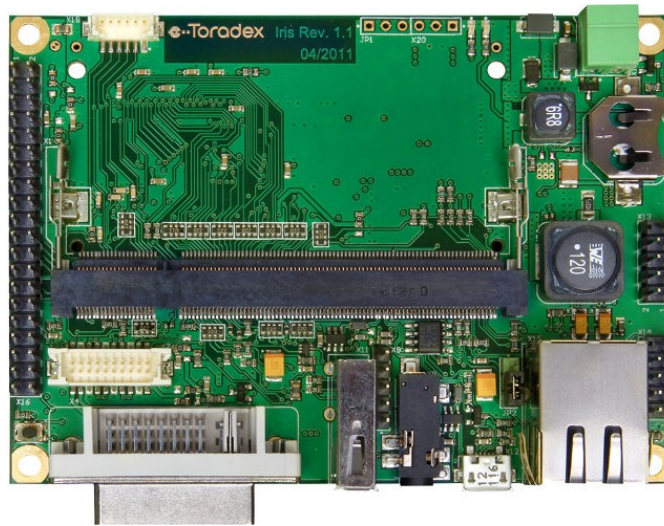
1x HDMI (Conector DVI)

1x VGA (Conector DVI)

1x RTC na placa

1x uSD

>25 GPIO's



Qt

Pinout i.MX6

| SOM Colibri i.MX6 | | | | | |
|-------------------|--------|-------------------|-------------------------|--------------|--------------|
| Placa Base IRIS | | SOM Colibri i.MX6 | Placa 3 Botões e 3 Leds | YPE Board | SO GNU LINUX |
| X16(Pino) | SODIMM | ALT5 | | | NUM. GPIO |
| 13 | 98 | GPIO1_IO15 | INPUT(Botão) | INPUT(Botão) | 15 |
| 14 | 133 | GPIO2_IO03 | INPUT(Botão) | OUTPUT(Led) | 35 |
| 15 | 103 | GPIO2_IO20 | INPUT(Botão) | OUTPUT(Led) | 52 |
| | | | | | |
| 16 | 101 | GPIO2_IO21 | OUTPUT(Led) | OUTPUT(Led) | 53 |
| 17 | 97 | GPIO2_IO18 | OUTPUT(Led) | INPUT(Botão) | 50 |
| 18 | 85 | GPIO6_IO06 | OUTPUT(Led) | INPUT(Botão) | 166 |

Formula: $32 * (\text{controller}-1) + \text{gpio} = \text{number_GPIO}$

$$32 * (1-1) + 15 = 15$$

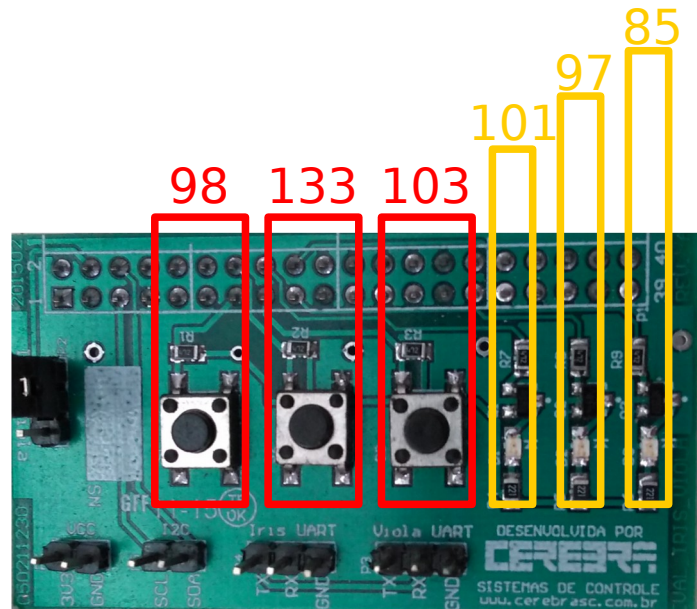
$$32 * (2-1) + 3 = 35$$

Qt

Placa

3 Botoes 3 Leds

| | PINO | SODIM M | GPIO |
|----|------|---------|------|
| S1 | 13 | 98 | 15 |
| S2 | 14 | 133 | 35 |
| S3 | 15 | 103 | 52 |
| D1 | 16 | 101 | 53 |
| D2 | 17 | 97 | 50 |
| D3 | 18 | 85 | 166 |





1.1

LABORATÓRIO

2.

LINUX
EMBARCADO



Seja Linux você também

Se utilizam em roteadores, TV's, Terminal POS, set-top-box, Laser Cutting Machine, IHM's, porque não utilizar na sua empresa?

Linux tudo é “arquivo”!

Kernel modular, com suporte a diversas arquiteturas além de x86, x86-64, ARM e MIPS, suportando drivers em diversos periféricos e barramentos tudo descrito via Device-Tree e carregado na inicialização.

E agora imagine um framework para agregar mais usabilidade, é o Qt!

The Qt logo, consisting of the letters 'Qt' in white on a green square background.

PROCFS

Um pseudo-filesystem onde prove acesso a informações de todos processos em execução ou qualquer file descriptor aberto em nível de usuário, além de outros propósitos, localizado em **/proc**

The Qt logo, consisting of the letters 'Qt' in white on a green square background.

SYSFS

Um pseudo-filesystem projeto para representar uma hierarquia amigável e legível de informações de dispositivos, drivers, barramentos e conexões, estruturado em links-simbólico e diretorios. , localizado em **/sys**



/sys e /proc

Listando os
diretórios **/proc** e
/sys na placa da
Toradex

```
cbueno@nb2-
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@colibri-imx6:/# ls /sys/
block      bus        class     dev       devices  firmware fs        kernel   module  power
root@colibri-imx6:/# cat /sys/firmware/devicetree/base/toradex,product-id
0014root@colibri-imx6:/#
root@colibri-imx6:/# cat /sys/firmware/devicetree/base/toradex,board-rev
V1.0Aroot@colibri-imx6:/#
root@colibri-imx6:/# ls /proc/
1          150        108       26        32        558       79        config.gz  filesystems  kpageflags  self        vmlallocinfo
10         151        2         266       33        559       8         consoles  fs           loadavg     softirqs   vmstat
11         152        20        27        365       6         80        cpu        interrupts  locks       stat        zoneinfo
12         153        208       271       474       625       82        cpufreq   iomem       meminfo     swaps
13         154        21        278       476       665       83        crypto    ioports     misc        sys
14         155        22        29        5         7         9         device-tree  irq         modules     sysvipc
146        16         225       3         512       70        asound    devices     kallsyms    mounts     thread-self
147        17         23        30        514       71        buddyinfo  diskstats  key-users   mtd        timer_list
148        18         24        302       516       70        bus        driver      keys        net
149        182        25        31        519       77        cgroups    execdomains  msg         pagetypeinfo  uptime
15         19         250       317       533       78        cmdline   fb          kpagecount  partitions    version

root@colibri-imx6:/# cat /proc/version
Linux version 4.1.35-00001-g5729297-dirty (raul@localhost.localdomain) (gcc version 4.9.2 (GCC) ) #207 SMP Thu May 18 11:35:31 BRT 2017
root@colibri-imx6:/# cat /proc/mounts
/dev/root / ext3 rw,relatime,data=ordered 0 0
devtmpfs /dev devtmpfs rw,relatime,size=59288k,nr_inodes=14822,mode=755 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,relatime 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
devpts /dev/pts devpts rw,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,nodev,mode=755 0 0
tmpfs /sys/fs/cgroup tmpfs ro,nosuid,nodev,noexec,mode=755 0 0
cgroup /sys/fs/cgroup/systemd cgroup rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd 0
cgroup /sys/fs/cgroup/cpu cpuset cgroup rw,nosuid,nodev,noexec,relatime,cpu,cpusctt 0
cgroup /sys/fs/cgroup/freezer cgroup rw,nosuid,nodev,noexec,relatime,freezer 0 0
cgroup /sys/fs/cgroup/debug cgroup rw,nosuid,nodev,noexec,relatime,debug 0 0
tmpfs /tmp tmpfs rw 0 0
debugfs /sys/kernel/debug debugfs rw,relatime 0 0
fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0
tmpfs /var/volatile tmpfs rw,relatime 0 0
/dev/mmcblk0p1 /media/mmcblk0p1 vfat rw,relatime,gid=6,fsnask=0007,dmknask=0007,allow_utime=0020,codepage=437,icharset=iso8859-1,shortname=mixed,errors=remount-ro 0 0
tmpfs /run/user/0 tmpfs rw,nosuid,nodev,relatime,size=25000k,mode=700 0 0
root@colibri-imx6:/#
```



Via SYSFS iremos exportar e manipular um GPIO

`/sys/class/gpio/export`

`/sys/class/gpio/gpioXX/direction`

`/sys/class/gpio/gpioXX/value`

`/sys/class/gpio/unexport`

```
cbueno@nb2~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
root@colibri-lmx6:/#  
root@colibri-lmx6:/# ls -l /sys/class/gpio/  
--w----- 1 root      root      4096 Aug  8 13:27 export  
lnwxrwxrwx 1 root      root       0 Aug  8 13:27 gpio15 -> ../../devices/soc0/soc/2000000.aips-bus/209c000.gpio/gpio/gpio15  
lnwxrwxrwx 1 root      root       0 Aug  8 13:27 gpio53 -> ../../devices/soc0/soc/2000000.aips-bus/20a0000.gpio/gpio/gpio53  
lnwxrwxrwx 1 root      root       0 Aug  8 10:48 gpiochip0 -> ../../devices/soc0/soc/2000000.aips-bus/209c000.gpio/gpio/gpiochip0  
lnwxrwxrwx 1 root      root       0 Aug  8 10:48 gpiochip128 -> ../../devices/soc0/soc/2000000.aips-bus/20a0000.gpio/gpio/gpiochip128  
lnwxrwxrwx 1 root      root       0 Aug  8 10:48 gpiochip160 -> ../../devices/soc0/soc/2000000.aips-bus/20b0000.gpio/gpio/gpiochip160  
lnwxrwxrwx 1 root      root       0 Aug  8 10:48 gpiochip192 -> ../../devices/soc0/soc/2000000.aips-bus/20b4000.gpio/gpio/gpiochip192  
lnwxrwxrwx 1 root      root       0 Aug  8 10:48 gpiochip32 -> ../../devices/soc0/soc/2000000.aips-bus/20a0000.gpio/gpio/gpiochip32  
lnwxrwxrwx 1 root      root       0 Aug  8 10:48 gpiochip64 -> ../../devices/soc0/soc/2000000.aips-bus/20a4000.gpio/gpio/gpiochip64  
lnwxrwxrwx 1 root      root       0 Aug  8 10:48 gpiochip96 -> ../../devices/soc0/soc/2000000.aips-bus/20a8000.gpio/gpio/gpiochip96  
--w----- 1 root      root      4096 Aug  8 13:26 unexport  
root@colibri-lmx6:/#  
root@colibri-lmx6:/#  
root@colibri-lmx6:/# ls -l /sys/class/gpio/gpio15/  
-rw-r--r-- 1 root      root      4096 Aug  8 14:18 active_low  
lnwxrwxrwx 1 root      root       0 Aug  8 14:18 device -> ../../../209c000.gpio  
-rw-r--r-- 1 root      root      4096 Aug  8 13:27 direction  
-rw-r--r-- 1 root      root      4096 Aug  8 14:18 edge  
dwxr-xr-x  2 root      root       0 Aug  8 14:18 power  
lnwxrwxrwx 1 root      root       0 Aug  8 14:18 subsystem -> ../../../../../../class/gpio  
-rw-r--r-- 1 root      root      4096 Aug  8 13:27 uevent  
-rw-r--r-- 1 root      root      4096 Aug  8 14:18 value  
root@colibri-lmx6:/#  
root@colibri-lmx6:/#  
root@colibri-lmx6:/#
```



Exportando **GPIO**
Configurando **GPIO**
Lendo um **pino**
Escrevendo em um **pino**

Led

```
cbueno@nb2~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
root@colibri-imx6:~# ls /sys/class/gpio/  
export      gpiochip0    gpiochip128  gpiochip160  gpiochip192  gpiochip32   gpiochip64   gpiochip96   unexport  
root@colibri-imx6:~# echo 53 > /sys/class/gpio/export  
root@colibri-imx6:~# ls /sys/class/gpio/  
export      gpio53       gpiochip0    gpiochip128  gpiochip160  gpiochip192  gpiochip32   gpiochip64   gpiochip96   unexport  
root@colibri-imx6:~# echo "out" > /sys/class/gpio/gpio53/direction  
root@colibri-imx6:~# cat /sys/class/gpio/gpio53/value  
0  
root@colibri-imx6:~# echo 1 > /sys/class/gpio/gpio53/value  
root@colibri-imx6:~# cat /sys/class/gpio/gpio53/value  
1  
root@colibri-imx6:~# echo 0 > /sys/class/gpio/gpio53/value  
root@colibri-imx6:~# cat /sys/class/gpio/gpio53/value  
0  
root@colibri-imx6:~# ls -l /sys/class/gpio/gpio53/  
-rw-r--r-- 1 root root 4096 Aug 8 13:22 active_low  
lrwxrwxrwx 1 root root 0 Aug 8 13:22 device -> ../../../../20a0000.gpio  
-rw-r--r-- 1 root root 4096 Aug 8 13:21 direction  
-rw-r--r-- 1 root root 4096 Aug 8 13:22 edge  
drwxr-xr-x 2 root root 0 Aug 8 13:22 power  
lrwxrwxrwx 1 root root 0 Aug 8 13:22 subsystem -> ../../../../class/gpio  
-rw-r--r-- 1 root root 4096 Aug 8 13:21 uevent  
-rw-r--r-- 1 root root 4096 Aug 8 13:22 value  
root@colibri-imx6:~# echo 53 > /sys/class/gpio/unexport  
root@colibri-imx6:~# ls /sys/class/gpio/  
export      gpiochip0    gpiochip128  gpiochip160  gpiochip192  gpiochip32   gpiochip64   gpiochip96   unexport  
root@colibri-imx6:~#
```



Exportando GPIO Configurando GPIO Lendo um pino

Botão

```
chueno@nb2-
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@colibri-imx6:~# echo 15 > /sys/class/gpio/export
root@colibri-imx6:~# echo "in" > /sys/class/gpio/gpio15/direction
root@colibri-imx6:~# ls -l /sys/class/gpio/gpio15/
-rw-r--r-- 1 root root 4096 Aug 8 13:25 active_low
lrwxrwxrwx 1 root root 0 Aug 8 13:25 device -> ../../209c000.gpio
-rw-r--r-- 1 root root 4096 Aug 8 13:24 direction
-rw-r--r-- 1 root root 4096 Aug 8 13:25 edge
drwxr-xr-x 2 root root 0 Aug 8 13:25 power
lrwxrwxrwx 1 root root 0 Aug 8 13:25 subsystem -> ../../../../../../class/gpio
-rw-r--r-- 1 root root 4096 Aug 8 13:24 uevent
-rw-r--r-- 1 root root 4096 Aug 8 13:25 value
root@colibri-imx6:~# cat /sys/class/gpio/gpio15/value
1
root@colibri-imx6:~# cat /sys/class/gpio/gpio15/value
0
root@colibri-imx6:~# cat /sys/class/gpio/gpio15/value
0
root@colibri-imx6:~# cat /sys/class/gpio/gpio15/value
1
root@colibri-imx6:~# cat /sys/class/gpio/gpio15/edge
none
root@colibri-imx6:~# echo 15 > /sys/class/gpio/unexport
root@colibri-imx6:~# ls /sys/class/gpio/
export gpiochip0 gpiochip128 gpiochip160 gpiochip192 gpiochip32 gpiochip64 gpiochip96 unexport
root@colibri-imx6:~#
```

/sys/class/gpio



Um overview geral dos
GPIO's na visão do kernel

DebugFS

```
cbueno@nb2 ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@colibri-imx6:~# echo 15 > /sys/class/gpio/export
root@colibri-imx6:~# echo "in" > /sys/class/gpio/gpio15/direction
root@colibri-imx6:~# echo 53 > /sys/class/gpio/export
root@colibri-imx6:~# echo "out" > /sys/class/gpio/gpio53/direction
root@colibri-imx6:~# cat /sys/kernel/debug/gpio
GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
 gpio-15 (sysfs) ) in hi

GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
 gpio-37 (cd) ) in lo
 gpio-53 (sysfs) ) out lo
 gpio-54 (Wake-Up) ) in hi

GPIOs 64-95, platform/20a4000.gpio, 20a4000.gpio:
 gpio-95 (usb_host_vbus) ) out lo

GPIOs 96-127, platform/20a8000.gpio, 20a8000.gpio:
 gpio-108 (scl) ) in hi
 gpio-109 (sda) ) in hi

GPIOs 128-159, platform/20ac000.gpio, 20ac000.gpio:
 gpio-130 (spi_imx) ) out lo

GPIOs 160-191, platform/20b0000.gpio, 20b0000.gpio:

GPIOs 192-223, platform/20b4000.gpio, 20b4000.gpio:
 gpio-204 (id) ) in lo
root@colibri-imx6:~#
root@colibri-imx6:~#
```

2.1

LABORÁTORIO



3.

Framework QT





Cross-Platform Class Library

One Technology for
All Platforms



Integrated Development Tools

Shorter Time-to-Market

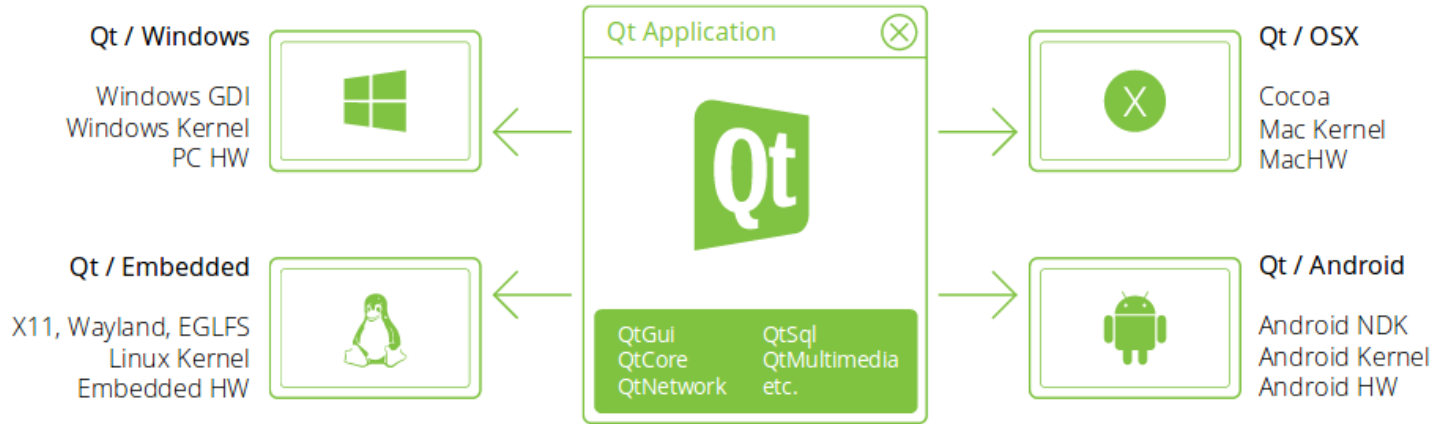


Cross-Platform IDE, Qt Creator

Productive development
environment



Utilizado por mais de 800.000 desenvolvedores em mais de
70 empresas
Tecnologia provada e testada desde 1994





Qt em **Módulos**

Módulos contem bibliotecas, plugins e documentação. Alguns módulos são comuns para todas plataformas, outros, são de proposito especifico e de uma plataforma alvo.

Qt Essentials

Disponíveis para todas plataformas QtCore, QtGui, QtMultimedia, QtNetwork, QtQML, QtQuick, QtSQL, QTest, QtWidgets



Qt Add-ons

Módulos de proposito especifico nem sempre disponível para todas plataformas Qt3D, QtBluetooth, QtConcurrent, QtD-BUS, QtGamePad, QtLocation, QtNFC, QtPositioning, QtPrint, QtQuickControls1/2, QtSensors, QtSerialBus, QtSerialPort e etc...

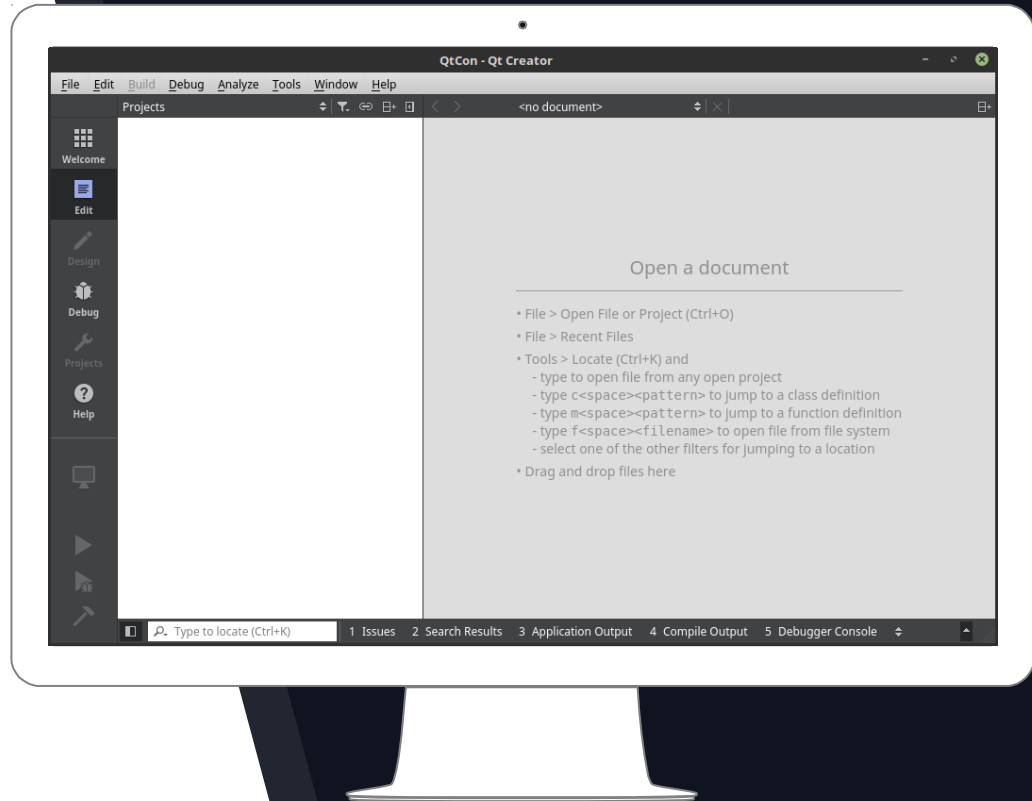
EXTENSÕES

- .qrc** – Qt Resource Collection (armazenar ícones, arquivos de traduções, imagens e etc)
- .cpp** – Código-fonte C++
- .h** – Cabeçalho do Código-fonte C++
- .ui** – User Interface
- .qml** – Qt Modeling Language (linguagem declarativa, fluida, parecida com JSON para descrever GUI)
- .js** – Contendo rotinas e declarações JavaScript
- .pro** – Projeto, descrição do projeto e dependências, utilizado pelo qmake
- .pri** – Includes a serem estendidos no .pro



IDE Qt Creator

Ferramenta open-source fornecida com todos os recursos para o desenvolvimento completo da aplicação

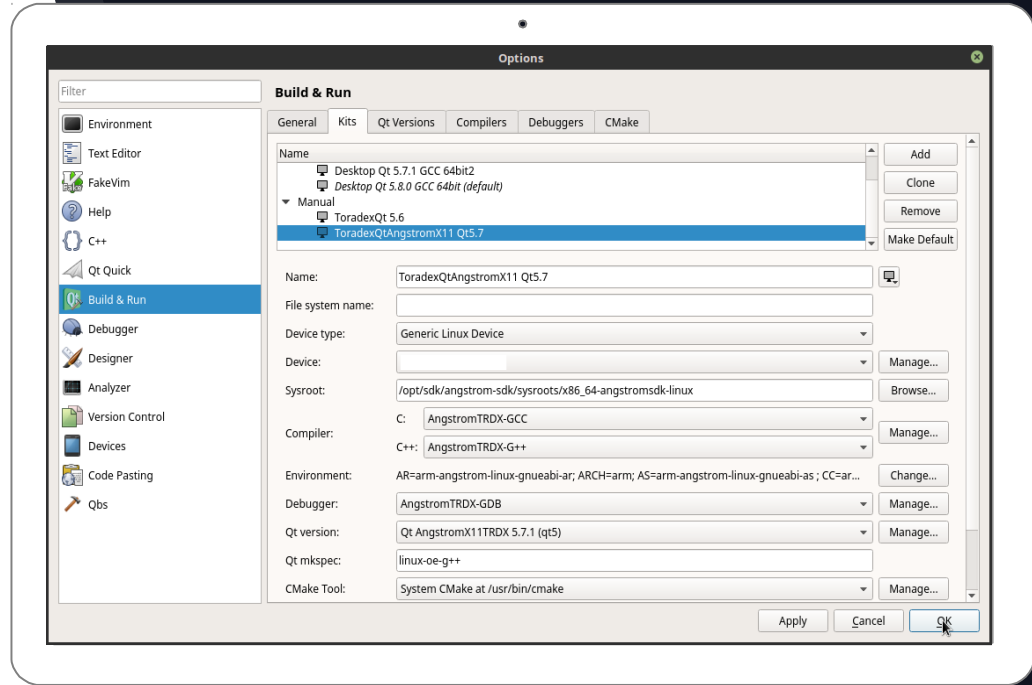




Qt Creator - Kit

O Kit que pode ser criado para cada plataforma e/ou versão, localizado em Build&Run em options

Tools > Options





Qt Quick

Utiliza uma linguagem declarativa conhecida como QML, e imperativa com JavaScript, integração runtime com Qt, API C++ para integração e suporte da IDE Qt Creator para linguagem QML

UI Design

(Designer)

Qt/C++

(Desenvolvedor)



QT **QUICK**



O que tem no QT QUICK

Elements

Elementos são estruturas de marcação da linguagem, podendo ser visuais ou não-visuais (states, models, ...) e elementos possuem propriedades.

Items

Item é um elemento visual, possui posições, dimensões, por exemplo Rectangle, Text e Image.

Properties

Um relação de chave=valor, todos elementos são descritos com propriedades seja (width, height, color, x, y, scale, opacity, ...).

Qt



Qt em **Módulos**

Módulos contem bibliotecas, plugins e documentação. Alguns módulos são comuns para todas plataformas, outros, são de proposito especifico e de uma plataforma alvo.

Rectangle

Utilizando o elemento Rectangle para representar uma retângulo, algumas propriedades:

radius: Arredondamento do retangulo

width: Especificando uma comprimento para a image

height: Especificando uma altura

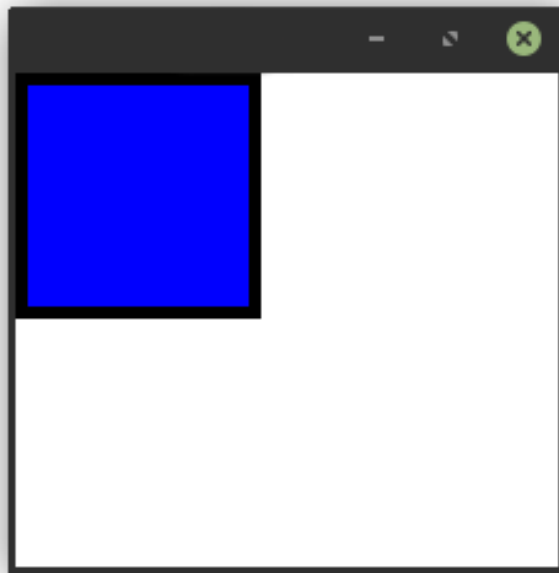
x e y: Especificando coordenadas X e Y

Outras propriedades **border, scale, rotation, opacity, etc ...**



Rectangle

```
Window {  
    visible: true  
    width: 640  
    height: 480  
  
    Rectangle {  
        width: 100; height: 100  
        border.width: 5  
        border.color: "black"  
        color: "blue"  
    }  
}
```



Qt

Text

Utilizando o elemento **Text** para representar um texto, algumas propriedades:

text: Entrada com o texto a ser exibido

width: Especificando uma comprimento

height: Especificando uma altura

x e y: Especificando coordenadas X e Y

Outras propriedades **font, color, etc ...**

Aceita sintaxe HTML na propriedade **text**



Text

```
Window {  
    visible: true  
    width: 640  
    height: 480  
  
    Rectangle {  
        anchors.centerIn: parent  
        width: 150; height: 150  
        border.width: 5  
        border.color: "black"  
        color: "blue"  
  
        Text {  
            anchors.centerIn: parent  
            font.pointSize: 24  
            font.bold: true  
            color: "#FFFFFF"  
            text: "QTCON"  
        }  
    }  
}
```



Image

Utilizando o elemento Image para representar uma imagem, algumas propriedades:

source: Pode-se obter uma imagem por caminho absoluto ou QRC

width: Especificando uma comprimento para a image

height: Especificando uma altura

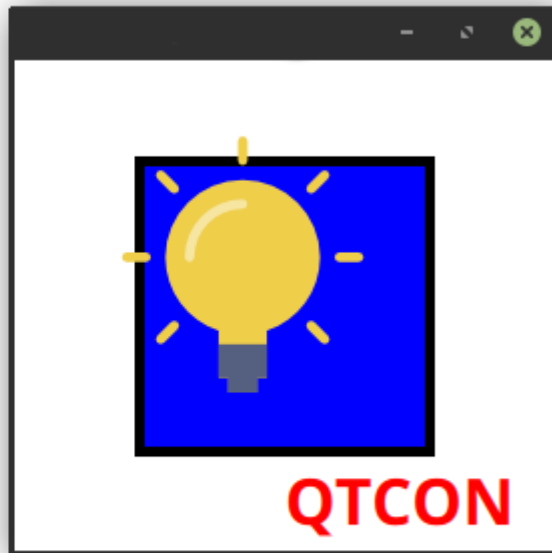
x e y: Especificando coordenadas X e Y

Outras propriedades **scale, rotation, opacity, etc ...**



Image

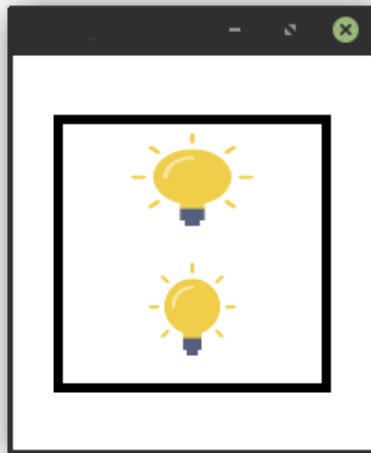
```
Rectangle {  
    anchors.centerIn: parent  
    width: 150; height: 150  
    border.width: 5  
    border.color: "black"  
    color: "blue"  
  
    Image {  
        x: -10  
        y: -10  
        source: "qrc:/IMAGES/led-on.png"  
    }  
  
    Text {  
        y: parent.height  
        x: parent.width/2  
        font.pointSize: 24  
        font.bold: true  
        color: "red"  
        text: "QTCON"  
    }  
}
```



Qt

Image

```
Rectangle {  
    anchors.centerIn: parent  
    width: 150; height: 150  
    border.width: 5  
    border.color: "black"  
    color: "transparent"  
  
    Image {  
        anchors.horizontalCenter: parent.horizontalCenter  
        y: 10  
        width: 70  
        height: 50  
        source: "qrc:/IMAGES/led-on.png"  
    }  
  
    Image {  
        anchors.horizontalCenter: parent.horizontalCenter  
        y: 80  
        width: 70  
        height: 50  
        fillMode: Image.PreserveAspectRatio  
        source: "qrc:/IMAGES/led-on.png"  
    }  
}
```



Qt

Row e Column

Agrupar items em linhas(**Row**) e colunas(**Column**), podendo utilizar Row dentro de Column, Columns dentro Row.

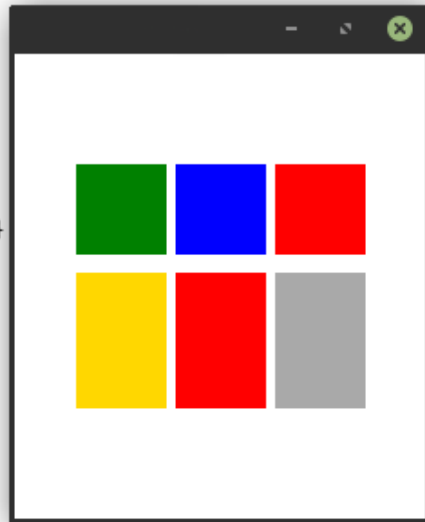
spacing: Espaçamento entre os items dentro do Row ou Column

Outras propriedades **move, add, populate, etc ...**



Row e Column

```
Rectangle {  
    anchors.fill: parent  
  
    Column {  
        spacing: 10  
        anchors.centerIn: parent  
  
        Row {  
            spacing: 5  
  
            Rectangle { width: 50; height: 50; color: "green" }  
            Rectangle { width: 50; height: 50; color: "blue" }  
            Rectangle { width: 50; height: 50; color: "red" }  
        }  
  
        Row {  
            spacing: 5  
  
            Rectangle { width: 50; height: 75; color: "gold" }  
            Rectangle { width: 50; height: 75; color: "red" }  
            Rectangle { width: 50; height: 75; color: "darkgray" }  
        }  
    }  
}
```



Qt

MouseArea

Adicionado em um item para monitorar entrada para cliques e Single-Touch, para Multi-Touch utiliza-se `MultiPointTouchArea`

Sinais identificados:

Click
DoubleClick
Pressed
Release
PressAndHold

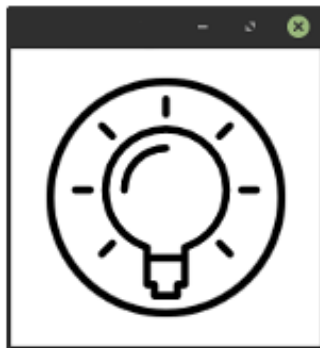
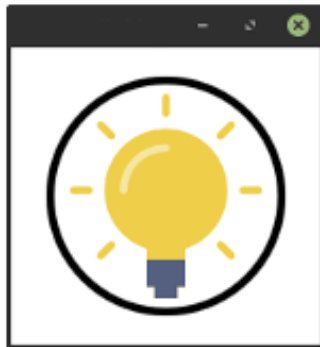
Qt



MouseArea

```
property bool ledOn: true
```

```
Rectangle {  
    anchors.centerIn: parent  
    width: 150; height: 150  
    border.width: 5  
    border.color: "black"  
    color: "transparent"  
    radius: width/2  
  
    Image {  
        anchors.centerIn: parent  
        source: ledOn ? "qrc:/IMAGES/led-on.png" :  
                "qrc:/IMAGES/led-off.png"  
  
        MouseArea {  
            anchors.fill: parent  
            onClicked: { ledOn = !ledOn }  
        }  
    }  
}
```



Qt

CORES

Tipos de especificação de cores

SVG Names: “yellow”, “red”, “green”, “gold”

HTML Style Color: “#0000FF”

Qt built-in: `Qt.rgb(0, 0.75, 0, 1)`



ANCORAS

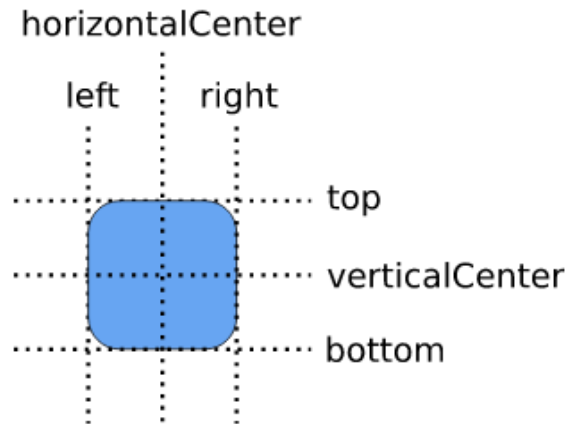
Propriedade anchors para posicionar e alinhar itens

Ancore em outros itens ou ancoras de itens

(left, right, top, bottom, verticalCenter e horizontalCenter)

E margins para adicionar espaços

Qt



PROPRIEDADE **Plus**

Estendendo itens e recursos

```
/* Properties */  
property var version: QtConBrasil2017  
property string productName: "myApp"  
property int year: 2017  
property real factorX: 1.35  
property double factorY: 1.7890  
property url linkUpdate: "http://update.mydomain.com"  
property bool flagActivePopup: false  
property color colorBackgroundMain: "#202020"
```

Qt

3.1

LABORATÓRIO



3.1

LABORATÓRIO





Criando a classe **GPIO**

Criando a classe GPIO

```
#ifndef GPIO_H
#define GPIO_H

#include <QObject>

class GPIO : public QObject
{
    Q_OBJECT
public:
    explicit GPIO(QObject *parent = nullptr);

signals:

public slots:
};

#endif // GPIO_H
```

Qt

Criando a classe GPIO

```
class GPIO : public QObject
{
    Q_OBJECT
public:
    // Direcao do GPIO ENTRADA ou SAIDA
    enum Direction { Input=0, Output };

    // Valor do Pino 0(Desligado) ou 1(Ligado)
    enum Level { Low=0, High };

    explicit GPIO(QObject *parent = nullptr);
    ~GPIO();

    // Funcao para exportar e usar pino do GPIO
    int gpioExport();

    // Funcao para liberar pino do GPIO
    int gpioUnexport();

    Direction direction() const;
    int setDirection(Direction direction);

    qint32 value() const;
    int setValue(Level value);
};
```

Qt

Métodos e Atributos

```
public:  
  
    int gpioExport();    // Funcao para exportar GPIO  
    int gpioUnexport(); // Funcao para liberar GPIO  
    bool gpioExported(); // Retorna se já foi exportado  
  
    gpioNumber();        // Retorna o numero do GPIO utilizado  
    setGpioNumber(quint32 gpioNumber); // Configura o GPIO a ser utilizado pela classe  
  
    direction();        // Retorna a direcao do GPIO  
    setDirection(Direction direction); // Configura a direcao do GPIO  
  
    value();            // Devolve o valor atual lido do GPIO  
    int setValue(Level value); // Escreve no pino 1(High) ou 0(Low)  
  
    static const QString gpioExportFilePath;  
    static const QString gpioUnexportFilePath;  
    static const QString gpioDirectionFilePath;  
    static const QString gpioValueFilePath;  
  
private:  
  
    quint32 m_gpioNumber; // Variavel privada recebe o Num. GPIO da Classe
```

Qt

Sinais e Slots

signals:

```
void outputChanged(); // Sinal emitido GPIO Output modificado  
void inputChanged(); // Sinal emitido GPIO Input modificado
```

public slots:

```
void TimeoutTimer(); // Slot chamado no estouro do QTimer
```



Integrando C++ com QML



As técnicas para expor dados e objetos de C++ para QML.

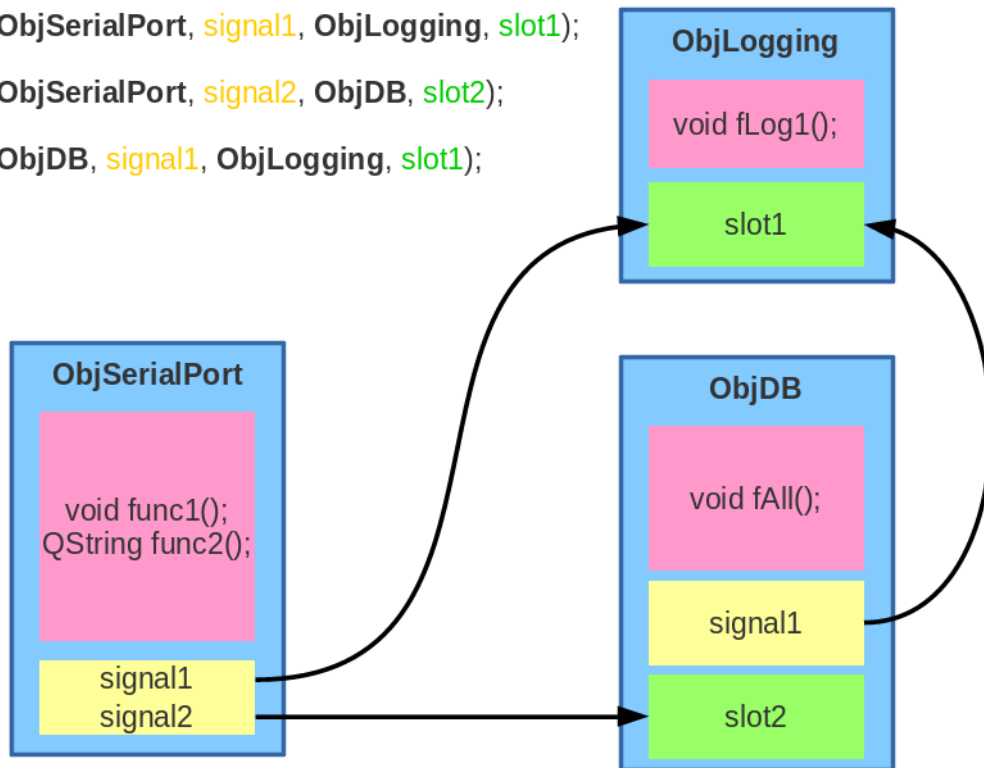
Signals & Slots, Q_PROPERTY e Q_INVOKABLE

SINAIS & SLOTS

```
connect(ObjSerialPort, signal1, ObjLogging, slot1);
```

```
connect(ObjSerialPort, signal2, ObjDB, slot2);
```

```
connect(ObjDB, signal1, ObjLogging, slot1);
```



Q_PROPERTY

É a combinação de funções de gets, sets e notificar sinais.

Ao alterar a variável declarada, um sinal é emitido e a função de leitura pode ser chamada para obter o valor.



```
class GPIO : public QObject
{
    Q_OBJECT
    Q_PROPERTY(int led READ getLed WRITE setLed NOTIFY ledChanged)
public:
    int getLed() const;
    void setLed(int gpio);

signals:
    void ledChanged();
};
```

Q_INVOKABLE

Métodos que podem ser chamados direto do QML, e que podem ou não retornar algum dado, o mesmo valor para slots.

```
class GPIO : public QObject
{
    Q_OBJECT
    Q_PROPERTY(int led READ getLed WRITE setLed NOTIFY ledChanged)
public:
    int getLed() const;
    void setLed(int gpio);
    Q_INVOKABLE void runMonitor();

signals:
    void ledChanged();

public slots:
    void killMonitor();
};
```

The Qt logo, consisting of the letters 'Qt' in white on a green square background.



E como expor
objetos?

Uma instancia

```
GPIO gpio;  
// Uma unica instancia para acesso via QML  
engine.rootContext()->setContextProperty("myGPIO", &gpio);
```

The Qt logo, consisting of the letters 'Qt' in white on a green square background.

Precisa de mais?

```
// Para ter mais de uma instancia, mais de 1 GPIO  
qmlRegisterType<GPIO>("qtcon.toradex.gpio", 1, 0, "GPIO");
```

Uma instancia

```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    GPIO gpio;

    QQmlApplicationEngine engine;

    // Uma unica instancia para acesso via QML
    engine.rootContext()->setContextProperty("myGPIO", &gpio);

    engine.load(QUrl(QStringLiteral("qrc:/main.qml")));

    return app.exec();
}
```

Qt

Uma **instancia**

```
Window {  
    visible: true  
    width: 640  
    height: 480  
  
    MouseArea {  
        anchors.fill: parent  
        onClicked: {  
            myGPIO.led = 15  
            myGPIO.runMonitor()  
            print("Led: "+myGPIO.led)  
        }  
    }  
}
```

Qt

Precisa de mais?

```
int main(int argc, char *argv[])
{
    QGuiApplication app(argc, argv);

    QmlApplicationEngine engine;

    // Para ter mais de uma instancia, mais de 1 GPIO
    qmlRegisterType<GPIO>("qtcon.toradex.gpio",1, 0, "GPIO");

    engine.load(QUrl(QStringLiteral("qrc:/main.qml")));

    return app.exec();
}
```

Qt

Precisa de mais?

```
import qtcon.toradex.gpio 1.0

Window {
    visible: true
    width: 640
    height: 480

    GPIO { id: gpio15; }
    GPIO { id: gpio25; }
    GPIO { id: gpio35; }

    MouseArea {
        anchors.fill: parent
        onClicked: {
            gpio15.led = 15
            gpio15.runMonitor()
            print("Led: "+gpio15.led)
        }
    }
}
}
```

Qt



QFile

Se tudo no Linux é arquivo com QFile podemos abrir, ler e escrever

QFile

```
QFile gpioExportFile("/sys/class/gpio/export");
if (!gpioExportFile.open(QIODevice::Append))
{
    qCritical() << "Cannot open the gpio export";
    return -1;
}

gpioExportFile.seek(0);

if (gpioExportFile.write(15) == -1)
{
    qCritical() << Q_FUNC_INFO << "Error while writing";
    gpioExportFile.close();
    return -1;
}
```

Qt



QTextStream

Ler de e escrever para um QFile aberto

QTextStream

```
QString gpioValueFilePath = "/sys/class/gpio/gpio%1/value";
qint32 m_gpioNumber = 15;
qint32 valueGPIO;

 QFile gpioValueFile(gpioValueFilePath.arg(m_gpioNumber));
 if (!gpioValueFile.open(QIODevice::ReadOnly)) {
     qCritical() << "Cannot open the relevant gpio value file:" <<
         gpioValueFilePath.arg(m_gpioNumber);
     gpioValueFile.close();
     return -1;
 }

 // Lendo o conteudo do arquivo e enviado para variavel valueGPIO
 QTextStream stream(&gpioValueFile);
 stream >> valueGPIO;

 // Caso tivesse que enviar o valor de valueGPIO para o arquivo aberto
 stream << valueGPIO;
```



QDebug

Mais pratico, direto e completa forma de utilizar logs no Qt, podendo agregar ao SysLog e Journald

QDebug

```
13 qDebug("Executando Funcao 1");
14 qDebug() << "Executando Funcao" << "1" << endl;
15 qDebug().nospace() << "Executando Funcao" << "1" << endl;
16
17 qDebugInfo() << "Note: Permission deny";
18
19 qDebugWarning() << "Warning: Check hardware available" << endl;
20
21 qDebugCritical() << "Error: Fail open fd /dev/aio" << endl;
22
```

Application Output



Starting /tmp/build-~~...~~-Desktop_Qt_5_8_0_GCC_64bit-Debug/...
QML debugging is enabled. Only use this in a safe environment.

Executando Funcao 1
Executando Funcao 1

Executando Funcao1

Note: Permission deny
Warning: Check hardware available

Error: Fail open fd /dev/aio

Qt



QTimer

Excelente forma para executar rotina periódica, disparar uma rotina após um tempo, sem loop, sem rotina bloqueante e sem travar GUI.

QTimer

Criando um **QTimer** para a cada 1000ms expirar e emitir o sinal **timeout()** e chamar o slot **procData()**.

```
QTimer *timer = new QTimer(this);  
connect(timer, SIGNAL(timeout()), this, SLOT(procData()));  
timer->start(1000);
```

Pode chamar função estática **singleShot()** do **QTimer** para após 10000ms emitir sinal e executar o slot **procUpdate()**.

```
QTimer::singleShot(10000, this, SLOT(procUpdate()));
```



QSysInfo

Prove diversas informações sobre o sistema em execução

QSysInfo

```
qDebug() << QSysInfo::buildAbi();  
qDebug() << QSysInfo::buildCpuArchitecture();  
qDebug() << QSysInfo::kernelType();  
qDebug() << QSysInfo::kernelVersion();  
qDebug() << QSysInfo::machineHostName();  
qDebug() << QSysInfo::prettyProductName();  
qDebug() << QSysInfo::productType();  
qDebug() << QSysInfo::productVersion();
```



QLibraryInfo

Prove informações sobre a biblioteca Qt

QLibraryInfo

```
qDebug() << QLibraryInfo::build();  
qDebug() << QLibraryInfo::buildDate();  
qDebug() << QLibraryInfo::licensedProducts();  
qDebug() << QLibraryInfo::version();  
qDebug() << QLibraryInfo::location(QLibraryInfo::BinariesPath);  
qDebug() << QLibraryInfo::location(QLibraryInfo::ImportsPath);  
qDebug() << QLibraryInfo::location(QLibraryInfo::LibrariesPath);
```

Qt

3.2

LABORATÓRIO



Não pare **por aqui!**

QML

Row, Column, Grid, ListModel, ListElement, Repeater, Video, Audio, Camera, GridView, ListView, StackView, ...

The Qt logo, consisting of the letters 'Qt' in white on a green square background.

Qt/C++

QProcess, QtSerialPort, QtSerialBus(Can e ModBUS), QThread, QRunnable, QtConcurrent, QDebug, QTextStream, QByteArray, QString, QPA(Qt Platform Abstraction)

Novidades!

QtMqtt - 14/08/2017

QtOTA - 05/2016



Referências

Pinout Designer Toradex

AGT Display [1] [2]

Qt Creator to Cross-Compile for Embedded Linux

Manipulando Logs com Qt5

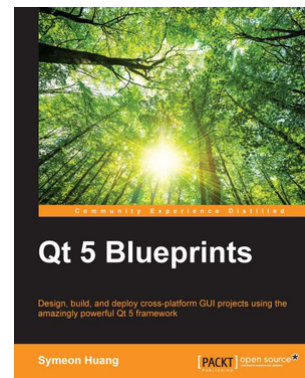
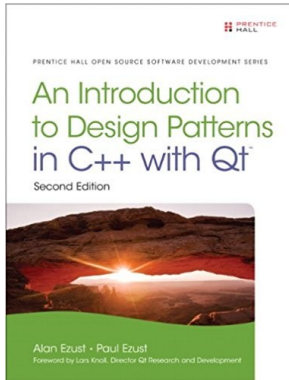
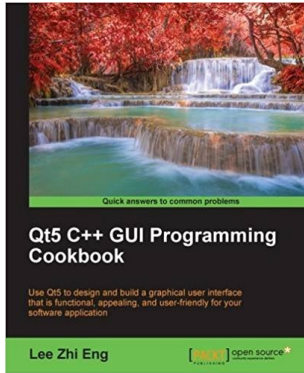
Embarcados sobre Qt

Embarcados sobre Qt5





PARA **APRENDER** MAIS SOBRE QT



Qt



OBRIGADO!

Duvidas?

Pode me encontrar em @cleitonrbueno &
cleiton.bueno@b2open.com