



QtCon Brasil 2018

Desenvolvendo aplicações com Qt em Linux embarcado

Exercícios (Parte 1)



Índice

Sobre este documento.....	3
Orientações gerais.....	3
Convenções.....	3
Comandos de substituição.....	3
Laboratório 1 – Gerando uma distribuição GNU/Linux com Qt5.....	4
1.1 Gerando a distribuição GNU/Linux.....	4
1.2 Gravando a imagem no cartão SD.....	5
1.3 Testando a distribuição GNU/Linux.....	6
1.4 Testando os leds do target.....	6
1.5 Configurando a rede.....	6
1.6 Configurando o touchscreen.....	7
1.7 Carregando o módulo da GPU.....	7
1.8 Testando as aplicações de exemplo do Qt5.....	8
Laboratório 2 – Configurando o Qt Creator.....	9
2.1 Configurando o Qt Creator.....	9
2.1.1 Devices.....	9
2.1.2 Compilers.....	9
2.1.3 Debuggers.....	10
2.1.4 Qt Versions.....	10
2.1.5 Kits.....	10
2.2 Testando o Qt Creator.....	11



Sobre este documento

Este documento contém uma série de exercícios para complementar o conteúdo do treinamento “Desenvolvendo aplicações com Qt em Linux embarcado” apresentado na QtCon Brasil 2018.

Orientações gerais

Não hesite em perguntar e tirar todas as suas dúvidas, bem como em trocar experiências e sugerir soluções diferentes das apresentadas nas aulas de laboratório. Sua participação pode deixar o conteúdo apresentado mais rico e o hands-on mais interativo.

Se você terminar suas tarefas antes de seu colega, ajude-o a investigar os problemas que ele possa estar enfrentando. Quanto mais rápido progredirmos como um grupo, mais chances teremos de ver tópicos adicionais e aprender mais.

Não hesite em reportar problemas ou erros ao instrutor.

Convenções

Durante os exercícios, chamaremos a máquina de desenvolvimento (PC, desktop, notebook) de *host* e o kit de desenvolvimento (hardware) de *target*.

Comandos que devem ser executados no terminal do Linux no *host* começam com \$.

```
$ ls
```

Comandos que devem ser executados no terminal do Linux no *target* começam com #.

```
# ls
```

Comandos de substituição

Expressões entre os sinais de menor e maior em vermelho devem ser substituídas de acordo com as instruções documentadas nos exercícios, incluindo-se os próprios sinais.

```
$ sudo dd if=arquivo.img of=/dev/<device> bs=1M
```

Mensagens em azul devem ser utilizadas de forma literal, e servem para facilitar a visualização de alterações em opções de configuração.

```
Linux kernel version (3.2.72)
```



Laboratório 1 – Gerando uma distribuição GNU/Linux com Qt5

Nesta atividade iremos utilizar o Buildroot para gerar uma distribuição GNU/Linux com suporte ao Qt5 e testar seu funcionamento no kit de desenvolvimento da Toradex.

1.1 Gerando a distribuição GNU/Linux

Entre no diretório do Buildroot:

```
$ cd /opt/labs/ex/buildroot/
```

Carregue a configuração padrão do Buildroot para o kit de desenvolvimento da Toradex:

```
$ make toradex_colibri_imx6_defconfig
```

Agora vamos alterar a configuração do Buildroot para habilitar o suporte ao Qt5.

Abra o menu de configuração do Buildroot:

```
$ make menuconfig
```

Habilite o suporte aos componentes básicos do Qt5 que utilizaremos no treinamento, selecionando as opções abaixo:

```
Target packages --->
  Graphic libraries and applications (graphic/text) --->
    [*] Qt5 --->
      [*] Compile and install examples (with code)
      [*] qt5charts
      [*] qt5declarative
      [*] qt5imageformats
      [*] qt5multimedia
      [*] qt5quickcontrols
      [*] qt5quickcontrols2
      [*] qt5sensors
      [*] qt5serialport
      [*] qt5svg
      [*] qt5tools
      [*] qt5virtualkeyboard
      [*] qt5websockets
```

Vamos agora configurar as bibliotecas do Qt5 para acessar diretamente a camada gráfica via OpenGL através do plugin *eglfs*.

Primeiro selecione as opções abaixo para habilitar o suporte à OpenGL e ao driver da GPU:



```
Target packages --->
  Hardware handling --->
    [*] Freescale i.MX libraries --->
        i.MX platform (imx6q/imx6dl)
    [*] imx-gpu-g2d
    [*] kernel-module-imx-gpu-viv
```

Agora retorne ao menu de configuração do Qt5 para habilitar mais algumas opções, incluindo o suporte ao plugin *eglfs*:

```
Target packages --->
  Graphic libraries and applications (graphic/text) --->
    [*] Qt5 --->
        [*] eglfs support
        (eglfs) Default graphical platform
        [*] fontconfig support
        [*] PNG support
        [*] Enable Tslib support
        [*] qt5graphicaleffects
```

O Qt5 não vem com nenhuma fonte integrada em seu código-fonte. Portanto, para que uma aplicação possa exibir textos, é necessário instalar um pacote de fontes. Para isso, habilite o suporte às fontes *DejaVu*:

```
Target packages --->
  Fonts, cursors, icons, sounds and themes --->
    [*] DejaVu fonts
```

Salve e feche o menu de configuração.

Agora compile a imagem do sistema:

```
$ make
```

No final da compilação, as imagens geradas estarão disponíveis no diretório `output/images`:

```
$ ls output/images
```

1.2 Gravando a imagem no cartão SD

Entre no diretório das imagens geradas pelo Buildroot:

```
$ cd /opt/labs/ex/buildroot/output/images
```

Insira o cartão SD no leitor e conecte-o na sua máquina de desenvolvimento.

Execute o comando `lsblk` para identificar o nome do arquivo de dispositivo criado para o cartão SD (deverá aparecer um dispositivo chamado `sdb`).

```
$ lsblk
```

Execute o comando abaixo para gravar a imagem da distribuição GNU/Linux no cartão SD, substituindo `<device>` pelo nome do arquivo de dispositivo identificado no comando anterior (Ex: `/dev/sdb`). Se uma senha for requisitada, digite `qtcon2018`.

```
$ sudo dd if=sdcard.img of=<device> bs=4M && sync
```

1.3 Testando a distribuição GNU/Linux

Insira o cartão SD no kit de desenvolvimento.

Conecte o target ao host com o cabo USB/serial e execute a aplicação de comunicação serial:

```
$ picocom
```

Ligue o target e verifique se você consegue visualizar as mensagens do boot do sistema.

Ao final do boot, faça o login com o usuário `root` (sem senha).

1.4 Testando os leds do target

Execute o comando abaixo para ligar um dos leds do kit de desenvolvimento:

```
# echo 1 > /sys/class/leds/ipe:green:usr1/brightness
```

Da mesma forma, o led pode ser desligado com o comando abaixo:

```
# echo 0 > /sys/class/leds/ipe:green:usr1/brightness
```

Teste os outros leds disponíveis em `/sys/class/leds/`.

1.5 Configurando a rede

Vamos agora configurar a comunicação entre o host e o target através de uma rede ponto-a-ponto com endereços IP fixos.

O host já está configurado com o endereço IP `192.168.0.1`. Abra um terminal na máquina de desenvolvimento e execute o comando `ifconfig` para confirmar:

```
$ ifconfig
```



Agora vamos configurar o target com o endereço IP `192.168.0.2`. Para isso, acesse o terminal do target e execute os comandos abaixo:

```
# echo auto eth0 >> /etc/network/interfaces
# echo iface eth0 inet static >> /etc/network/interfaces
# echo address 192.168.0.2 >> /etc/network/interfaces
# echo netmask 255.255.255.0 >> /etc/network/interfaces
# echo gateway 192.168.0.1 >> /etc/network/interfaces
# ifup -a
```

Execute o comando `ping` para testar a comunicação com o host:

```
# ping 192.168.0.1
```

Vamos também configurar a comunicação do host com o target via SSH.

Primeiro execute os comandos abaixo no target para que o servidor de SSH permita o login do usuário `root` sem senha:

```
# echo 'PermitRootLogin yes' >> /etc/ssh/sshd_config
# echo 'PermitEmptyPasswords yes' >> /etc/ssh/sshd_config
# /etc/init.d/S50sshd restart
```

Agora abra um terminal no host e teste a conexão SSH com o target:

```
$ ssh root@192.168.0.2
```

1.6 Configurando o touchscreen

Com o display conectado ao target, execute a aplicação `ts_calibrate` para realizar a calibração do display, tocando nos pontos indicados pela aplicação:

```
# export TSLIB_TSDEVICE=/dev/input/event0
# ts_calibrate
```

Agora teste a calibração com o comando `ts_test`. Se perceber que a calibração não ficou boa, repita o processo.

```
# ts_test
```

1.7 Carregando o módulo da GPU

O driver da GPU do SoC utilizado no target (i.MX6) é disponibilizado pelo fabricante através de um módulo externo do kernel.



Vamos então criar um arquivo chamado `/etc/init.d/S60gpu` para carregar este módulo automaticamente no boot do sistema:

```
# echo "#!/bin/sh" >> /etc/init.d/S60gpu
# echo modprobe galcore >> /etc/init.d/S60gpu
# chmod +x /etc/init.d/S60gpu
# reboot
```

Ao final do reboot, faça o login e verifique se o módulo foi carregado corretamente (um módulo chamado `galcore` deverá ser exibido na saída do comando abaixo):

```
# lsmod
```

1.8 Testando as aplicações de exemplo do Qt5

Teste o Qt5 executando algumas aplicações de exemplo:

```
# /usr/lib/qt/examples/charts/qmlchart/qmlchart
# /usr/lib/qt/examples/opengl/cube/cube
# /usr/lib/qt/examples/quick/demos/clocks/clocks
# /usr/lib/qt/examples/quick/demos/maroon/maroon
# /usr/lib/qt/examples/quick/demos/stocqt/stocqt
# /usr/lib/qt/examples/quickcontrols/extras/dashboard/dashboard
# /usr/lib/qt/examples/quickcontrols2/wearable/wearable
# /usr/lib/qt/examples/virtualkeyboard/basic/basic
# /usr/lib/qt/examples/widgets/animation/animatedtiles/animatedtiles
```

Sinta-se à vontade para testar outras aplicações disponíveis em `/usr/lib/qt/examples`.



Laboratório 2 – Configurando o Qt Creator

Nesta atividade iremos configurar o Qt Creator para desenvolver e depurar aplicações em Qt5 no kit de desenvolvimento da Toradex.

2.1 Configurando o Qt Creator

O Qt Creator pode ser baixado diretamente do site do projeto em <https://www.qt.io/download>.

Para facilitar o nosso trabalho, a versão Open Source do Qt Creator já foi baixada e instalada no ambiente de laboratório do treinamento.

Abra um terminal no host e execute o comando abaixo para iniciar o Qt Creator:

```
$ ~/Qt/Tools/QtCreator/bin/qtcreator
```

Acesse o menu “Tools” → “Options” para iniciarmos o procedimento de configuração.

2.1.1 Devices

Selecione a opção “Devices” à esquerda, acesse a guia “Devices” e clique em “Add...”.

Selecione “Generic Linux Device”, clique em “Start Wizard” e preencha os seguintes campos:

The name to identify this configuration	Target
The device's host name or IP address	192.168.0.2
The user name to log into the device	root

Certifique-se de que o target esteja ligado e conectado ao host pelo cabo de rede. Clique então em “Next” e depois em “Finish”.

O Qt Creator irá realizar um teste de conexão com o target via SSH. Se der algum erro, revise as configurações de rede do host e do target.

2.1.2 Compilers

Selecione “Kits” à esquerda, abra a guia “Compilers”, clique em “Add...” → “GCC” → “C” e preencha os seguintes campos:

Name	GCC-ARM
Compiler path	/opt/labs/ex/buildroot/output/host/bin/arm-linux-gnueabi-hf-gcc



Ainda na mesma tela de configuração, clique em “Add...” → “GCC” → “C++” e preencha os seguintes campos:

Name	G++ - ARM
Compiler path	/opt/labs/ex/buildroot/output/host/bin/arm-linux-gnueabi-hf-g++

Clique em “Apply”.

2.1.3 Debuggers

Selecione “Kits” à esquerda, abra a guia “Debuggers”, clique em “Add...” e preencha os seguintes campos:

Name	GDB - ARM
Path	/opt/labs/ex/buildroot/output/host/bin/arm-linux-gnueabi-hf-gdb

Clique em “Apply”.

2.1.4 Qt Versions

Selecione “Kits” à esquerda, abra a guia “Qt Versions”, clique em “Add...” e selecione o arquivo /opt/labs/ex/buildroot/output/host/bin/qmake.

Preencha os seguintes campos:

Version name	Qt 5.11.1 (ARM)
--------------	-----------------

Clique em “Apply”.

2.1.5 Kits

Selecione “Kits” à esquerda, abra a guia “Kits”, clique em “Add...” e preencha os seguintes campos:

Name	Toradex Colibri i.MX6
Device type	Generic Linux Device
Sysroot	/opt/labs/ex/buildroot/output/host/arm-buildroot-linux-gnueabi-hf/sysroot
Compiler C	GCC - ARM
Compiler C++	G++ - ARM



Debugger	GDB-ARM
Qt version	Qt 5.11.1 (ARM)
Qt mkspec	/opt/labs/ex/buildroot/output/host/mkspecs/devices/linux-buildroot-g++

Clique em “Apply” e depois em “OK”.

2.2 Testando o Qt Creator

Vamos agora testar a configuração do Qt Creator.

Selecione “Projects” à esquerda, clique em “Open Project” e abra o projeto `digitalclock.pro` em `/home/qtcon/Qt/Examples/Qt-5.11.1/widgets/widgets/digitalclock/`.

Remova a seleção do kit “Desktop”, selecione apenas o kit “Toradex Colibri i.MX6” e clique em “Configure Project”.

Selecione “Debug” → “Start Debugging” ou pressione F5 para iniciar a depuração da aplicação no target.

Verifique se a aplicação foi exibida corretamente no display do target.

Abra o arquivo `digitalclock.cpp`, coloque um breakpoint no início do método `showTime()` e confirme se o debugger interrompe a execução no breakpoint configurado. Remova o breakpoint e execute a função passo-a-passo.

Sinta-se livre para abrir e depurar outros projetos de exemplo disponíveis no diretório de instalação do Qt em `/home/qtcon/Qt/Examples/`.