关于 QML、Obigo、SDL

2014/08/05	Init	Kevin Tang
2014/08/05	1. QT5 Fix Compile Error 2. 字体放大到小四	Kevin Tang

一、关于 QML

很不幸，系统安装的 QT4 没有 qmlscene，QT5 才有...

为了使用 QML，我们自行编译 QT 和 Qt Creator

【当然你也可以安装系统的 QT5】

【系统安装的 QT4 和 QT5 貌似有部分冲突（qt4-default 和 qt5-default）】

【QT4（libqt4-XXX）和 QT5（libqt5XXX）的包名还稍微有点差异】

【索性那就手动编译 QT4 和 QT5 吧，也可以当作练习编译】

由于换座位了，所以我的 IP 变更为：172.26.14.181

（也许你可以通过 ping 或者 traceroute 或者 nslookup 查找下 tangwenkai 的最新 IP 地址）

很不幸由于编译了 apr 和 apr-util 导致我的 apache 挂了，暂时没有时间修复，用 nginx 替换，端口改为 8080。

Apache Fixed! 80 端口恢复。

QT 下载地址

http://172.26.14.181:8080/nfs/Downloads/qt-everywhere-opensource-src-4.8.3.tar.gz
http://172.26.14.181:8080/nfs/Downloads/qt-everywhere-opensource-src-5.3.0.tar.gz

Qt Creator 下载地址

http://172.26.14.181:8080/nfs/Downloads/qt-creator-2.8.1-src.tar.gz
http://172.26.14.181:8080/nfs/Downloads/qt-creator-opensource-src-3.1.2.tar.gz

1. 编译 Qt4

mkdir -p /opt/qt4.8
tar -zxvf qt-everywhere-opensource-src-4.8.3.tar.gz -C /opt/qt4.8
cd /opt/qt4.8/qt-everywhere-opensource-src-4.8.3

./configure -developer-build -opensource -no-qt3support -confirm-license #。#

make -j4 #。#

DO NOT MAKE INSTALL
sudo update-alternatives --install /usr/bin/qmake qmake

/opt/qt4.8/qt-everywhere-opensource-src-4.8.3/bin/qmake 30 #。#

sudo update-alternatives --config qmake

#ERR Basic XLib functionality test failed!
#FIX sudo apt-get install libX11-dev libXext-dev libXtst-dev

* 对应目录自行替换

* 编译 QT4 的话问题不大，自行解决

2. 编译 Qt Creator

tar -zxvf qt-creator-2.8.1-src.tar.gz -C /opt/qt4.8
cd /opt/qt4.8/qt-creator-2.8.1-src
qmake -r
make -j4
DO NOT MAKE INSTALL

* make 过程会有点漫长 -j4 会稍微快一点

* ./configure 不通过的时候可能需要指定一些参数 【TODO】

3. 编译 QT5

* QT5 是刚入职的时候编译的，也遇到过几个小问题，当时没有记录，抱歉

sudo apt-get install "^libxcb.*" libx11-xcb-dev libglu1-mesa-dev libxrender-dev libxi-dev #。#

sudo apt-get install libssl-dev
mkdir /opt/qt5.3
cd /opt/qt5.3//opt/qt5.3/qt-everywhere-opensource-src-5.3.0
./configure -developer-build -opensource -nomake tests -confirm-license

* 以下是我的 configure 输出，当然不一定完全和你的一样

Configure summary

Build type: linux-g++ (x86_64, CPU features: mmx sse sse2)
Platform notes:

- Also available for Linux: linux-kcc linux-icc linux-cxx

Build options:
Configuration ..... accessibility accessibility-atspi-bridge alsa audio-backend avx c++11 clock-gettime clock-monotonic compile_examples concurrent dbus debug directfb evdev eventfd fontconfig full-config getaddrinfo getifaddrs glib iconv icu inotify ipv6ifname large-config largefile libudev linuxfb medium-config minimal-config mremap nis no-harfbuzz openssl pcre png posix_fallocate precompile_header private_tests pulseaudio qpa qpa reduce_exports reduce_relocations rpath shared small-config sse2 sse3 sse4_1 sse4_2 ssse3 system-freetype system-jpeg system-png system-zlib warnings_are_errors xcb xcb-glx xcb-plugin xcb-render xcb-sm xinput2 xkbcommon-qt xlib xrender
Build parts ..... libs tools examples tests
Mode ..... debug
Using C++11 ..... yes
Using PCH ..... yes
Target compiler supports:
SSE2/SSE3/SSSE3 ..... yes/yes/yes
SSE4.1/SSE4.2 ..... yes/yes
AVX/AVX2 ..... yes/no

Qt modules and options:
- Qt D-Bus ..... yes (loading dbus-1 at runtime)
- Qt Concurrent ..... yes
- Qt GUI ..... yes
- Qt Widgets ..... yes
- Large File ..... yes
- QML debugging ..... yes
- Use system proxies ..... no

Support enabled for:
- Accessibility ..... yes
- ALSA ..... yes
- CUPS ..... no
- Evdev ..... yes
- FontConfig ..... yes
- FreeType ..... yes (system library)
- Glib ..... yes
- GTK theme ..... no
- HarfBuzz ..... no
- Iconv ..... yes
- ICU ..... yes
- Image formats:
  - GIF ..... yes (plugin, using bundled copy)
  - JPEG ..... yes (plugin, using system library)
  - PNG ..... yes (in QtGui, using system library)
- journald ..... no
- mtdev ..... no
- Networking:
  - getaddrinfo ..... yes
  - getifaddrs ..... yes
  - IPv6 ifname ..... yes
  - OpenSSL ..... yes (loading libraries at run-time)
- NIS ..... yes
- OpenGL / OpenVG:
  - EGL ..... no
  - OpenGL ..... no
  - OpenVG ..... no
- PCRE ..... yes (bundled copy)
- pkg-config ..... yes
- PulseAudio ..... yes
- QPA backends:
  - DirectFB ..... yes
  - EGLFS ..... no
  - KMS ..... no

```
        LinuxFB ..... yes
        XCB ..... yes (system library)
            EGL on X ..... no
            GLX ..... yes
            MIT-SHM ..... yes
            Xcb-Xlib ..... no
            Xcursor ..... yes (loaded at runtime)
            Xfixes ..... yes (loaded at runtime)
            Xi ..... no
            Xi2 ..... yes
            Xinerama ..... yes (loaded at runtime)
            Xrandr ..... yes (loaded at runtime)
            Xrender ..... yes
            XKB ..... no
            XShape ..... yes
            XSync ..... yes
            XVideo ..... yes
    Session management ..... yes
    SQL drivers:
        DB2 ..... no
        InterBase ..... no
        MySQL ..... yes (plugin)
        OCI ..... no
        ODBC ..... no
        PostgreSQL ..... yes (plugin)
        SQLite 2 ..... no
        SQLite ..... yes (plugin, using bundled copy)
        TDS ..... no
    udev ..... yes
    xkbcommon ..... yes (bundled copy, XKB config root: /usr/share/X11/xkb)
    zlib ..... yes (system library)

NOTE: libxkbcommon and libxkbcommon-x11 0.4.1 or higher not found on the system, will use
the bundled version from 3rd party directory.
Info: creating super cache file /opt/qt5.3/temp/qt-everywhere-opensource-src-5.3.0/.qmake.super

Qt is now configured for building. Just run 'make'.
Once everything is built, Qt is installed.
You should not run 'make install'.

Prior to reconfiguration, make sure you remove any leftovers from
the previous build.
```

make -j4 #。#还是一个漫长的编译

NO_NEED_TO_INSTALL
sudo update-alternatives --install /usr/bin/qmake qmake

/opt/qt5.3//opt/qt5.3/qt-everywhere-opensource-src-5.3.0/qtbase/bin/qmake 40 #。#

```
tangwenkai@tangwenkai:qt-everywhere-opensource-src-5.3.0$ ls qtbase/bin/
assistant          qcollectiongenerator  qml                qmlviewer
designer           qdbus                 qml1plugindump     qt.conf
findtr             qdbuscpp2xml          qmlbundle          qtdiag
fixqt4headers.pl   qdbusviewer          qmlimportscanner   qtpaths
lconvert           qdbusxml2cpp          qmljs              rcc
linguist           qdoc                  qmlmin             syncqt.pl
lrelease           qhelpconverter        qmlplugindump      uic
lupdate            qhelpgenerator        qmlprofiler        xmlpatterns
moc                qlalr                 qmlscene           xmlpatternsvalidator
pixeltool          qmake                 qmltestrunner
```

* /opt/qt5.3/qt-everywhere-opensource-src-5.3.0/qtbase/bin 路径稍微有些不同

#ERR Project ERROR: Unknown module(s) in QT: quick-private
#FIX sudo apt-get install flex bison gperf libicu-dev libxslt-dev ruby

#EXT 因为这些库和 32 位 GL 库有冲突导致你们的编译失败

* 这里也有一篇参考文档 http://qt-project.org/wiki/Building_Qt_5_from_Git

4. 编译 Qt Creator3.1.2

编译 Qt Creator 之前需要选择 QT5 的 qmake

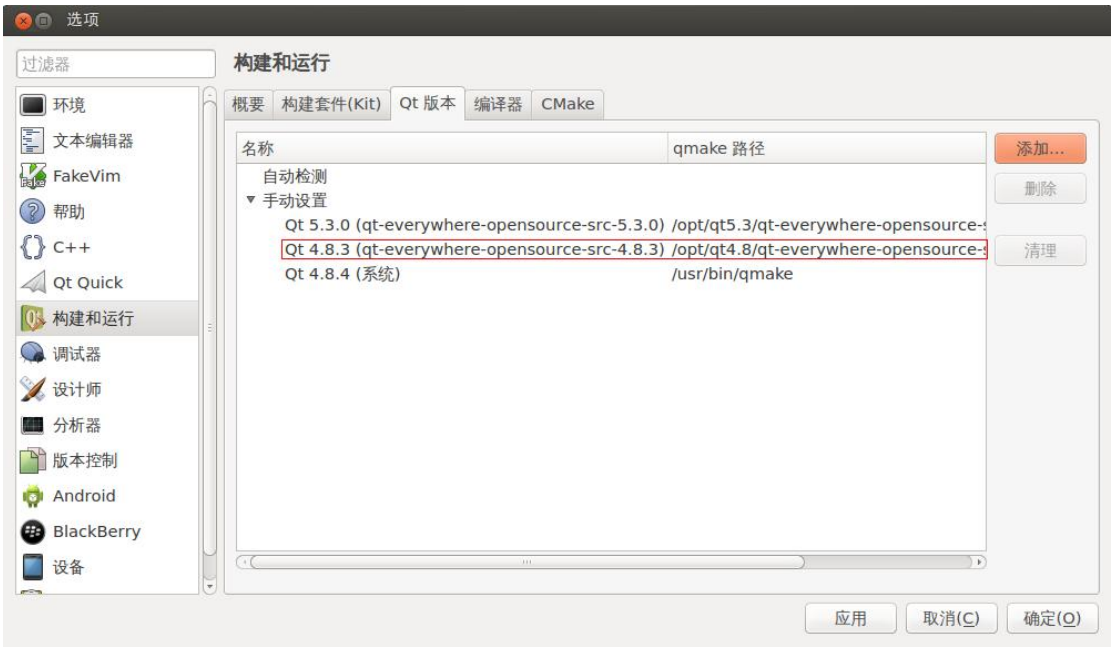
* 后续随时关注 qmake 版本以防出错

sudo update-alternatives --config qmake
qmake
make -j4

5. QtCreator 配置

工具->选项->添加 【Qt4 qmke 的路径：

/opt/qt4.8/qt-everywhere-opensource-src-4.8.3/bin】



名称：自定义

Qt 版本：选择刚刚添加的 Qt 版本

然后打开工程->项目->添加构建套件->选择刚刚添加的构建套件

* 添加 Qt 和构建套件以 4.8.3 为例，QT5 和交叉环境的 qmake 也是如此配置

二、关于 **Obgio**

之前的项目，使用 SPAC 协议？

项目仓库：

repo init -u ssh://git@nutshell.suntec.net/nutshell/develop/platform/manifest.git -b nutshell/iAuto_For_SPAC

代码路径：packages/NativeApp/AV/SPAC*

SDL 分支代码从此拉出，所以这边就可以略过。

三、关于 **SDL**

新的项目，使用 SDL 协议？（3.7 #。#）

全称：smartdevicelink

源码仓库：git://git.projects.genivi.org/smartdevicelink.git

项目仓库：

repo init -u ssh://git@nutshell.suntec.net/nutshell/develop/platform/manifest.git -b nutshell/iAuto_For_SDL -m AppFramework.xml

环境依赖：cmake3.0 （cmake 至少 2.8.11）

下载地址：官网

（或者 http://172.26.14.181:8080/nfs/Downloads/cmake-3.0.1.tar.gz）

编译 cmake：

```
# ./configure
# make
```

sudo make install

如果需要 GUI

./configure --qt-gui
make
make install

依赖 APR APR-UTIL

http://172.26.14.181/nfs/Downloads/apr-1.5.1.tar.gz
http://172.26.14.181/nfs/Downloads/apr-util-1.5.3.tar.gz

编译 APR：

cd /opt/temp/apr-1.5.1
./configure
make
sudo make install

* ./configure 没有指定 PREFIX 的默认安装到/usr/local

* 这里 apr 的话默认安装到了/usr/local/apr 下面

编译 APR-UTIL

cd /opt/temp/apr-util-1.5.3
./configure --with-apr=/usr/local/apr --with-ldap --with-crypto

* 之前没有指定这两个参数导致 Apache 挂了

make
sudo make install

* 由于 so 安装到了 /usr/local/apr/bin 下，需要额外指定软连接（当然拷贝也行）

cd /usr/local/lib
sudo ln -s ../apr/bin/*.so* .

依赖 log4cxx:

http://172.26.14.181/nfs/Downloads/apache-log4cxx-0.10.0.tar.gz

tar -zxvf apache-log4cxx-0.10.0.tar.gz -C /opt/temp
cd /opt/temp/apache-log4cxx-0.10.0

```
# ./configure
# make
# sudo make install
```

SDL 有几种模式 1. WEB 2. QML 3. QNX

编译 SDL：

1. WEB

```
# git checkout dev/merge_3.6
# mkdir build
# cd build
# cmake -DEXTENDED_MEDIA_MODE=ON -DHMI2=ON
-DGSTREAMER_gst_INCLUDE_DIR=/opt/obigo/sdl/src/components/media_manag
er
-DGSTREAMER_gstreamer_LIBRARY=/opt/obigo/sdl/src/components/media_mana
ger ..
# make
# make install
```

* 照理-DEXTENDED_MEDIA_MODE=ON -DHMI2=ON 是编译 QML 的 #。#

* 由于没有指定 HMI=qt -#。#

* 可能缺少的依赖

```
vi ../src/components/media_manager/CMakeLists.txt
vi ../src/appMain/CMakeLists.txt
 48     /usr/include/gstreamer-0.10
 49     /usr/include/libxml2

vi ../src/appMain/CMakeLists.txt
@LINE 49 gstreamer-0.10
```

* 可能缺少的组建

```
# sudo apt-get install libbluetooth-dev
```

2. QML 探索中

```
# cmake 参数 -DHMI=qt
```

#ERR 未找到 QT

#FIX
export CUSTOM_QT_DIR=/opt/qt5.3

#ERR 提示缺少 log4cxx

#FIX
vi ../src/components/qt_hmi/qml_plugins/dbus_adapter/CMakeLists.txt
vi ../src/components/qt_hmi/qml_plugins/log4cxx/CMakeLists.txt
@LINK_DIRECTORIES(/usr/local/lib)

运行

cd /opt/obigo/sdl/build-3.6/bin
./start_hmi.sh

3. QNX 略

暂不需要 QNX 环境

编译 iAuto_For_SDL

参考 NativeApp 编译

* 重点查看 SDL 中 QML 的实现

* 重点查看 iAuto_For_SDL 中 SPAC 相关的实现代码

*** 先自行编译 QT4 和 QT5

*** 二、三章节待完善

*** 写的有点仓促，错误必定很多，待反馈