Qt 4.8.5 Immigration Manual

Ver:1.0

2014/05/05

QIYANG TECHNOLOGY Co., Ltd

Catalogue

Preface

Welcome to use the board from **Zhejiang Qiyang Intelligent Technology Co.,Ltd** , this manual mainly introduce the immigration on QT4.8.5.

Ⅰ.Summary

Hardware Resource: QY-IMX6S Development Board

Host Machine: Ubuntu 12.04

Qt source code: qt-everywhere-opensource-src-4.8.5.tar.gz

Touch panel library: tslib-master.tar.gz

Cross-compile tool: fsl-linaro-toolchain.tar.gz

Ⅱ. Add library to support touch panel

1 .Make sure the cross-compiler is installed correctly.

2 . Compile library which supports touch panel

2.1 .Upgrade the following command in ubuntu.

```
sudo apt-get install libtool
sudo apt-get install autoconf
sudo apt-get install automake
```

2.2 Extract [tar xzvf tslib-master.tar.gz]

```
cd tslib-master
```

2.3 Then, execute the command

```
./autogen.sh
./configure --host=arm-linux ac_cv_func_malloc_0_nonnull=yes CC=
CXX=arm-fsl-linux-gnueabi-g++ -prefix=/home/work/qtenv/tslib
```

Note: Cross-compiler tool is correct. [-prefix] is the installation directory.

Compile the installed [tslib] in [/home/work/qtenv/tslib] folder.

2.4 Compile and install

```
make && make install
```

2.5 Configure

After compilation, the [tslib] will be in [/home/work/qtenv/] directory. Enter into [tslib]directory, edit['ts.conf']file.

```
cd tslib/etc
gedit ts.conf
```

Edit this file, remove[#] and[Spacing] before [module_raw input].

If there is [Spacing] before [module raw input], it will hint [Segmentation fault] when runs test software.

Ⅲ.Install Qt library

1 . Extract Qt source code

```
tar zxvf qt-everywhere-opensource-src-4.8.5.tar.gz
```

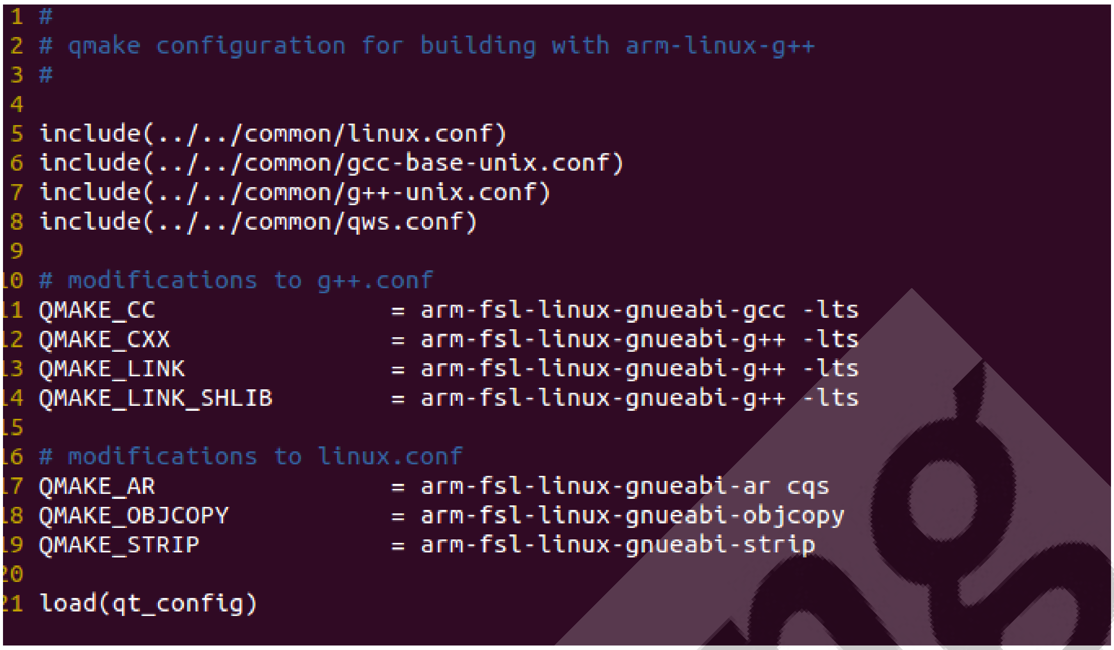
2 . Compile

(1)`cd qt-everywhere-opensource-src-4.8.5`

(2) Assign cross-compiler

```
vi mkspecs/qws/linux-arm-g++/qmake.conf
```

Edit [qmak.conf], modify[arm-linux] to [arm-fsl-linuc-gnueabi], add parameter [lts] to the four options, as shown:



```
#
# qmake configuration for building with arm-linux-g++
#

include(../../common/linux.conf)
include(../../common/gcc-base-unix.conf)
include(../../common/g++-unix.conf)
include(../../common/qws.conf)

# modifications to g++.conf
QMAKE_CC                = arm-fsl-linux-gnueabi-gcc -lts
QMAKE_CXX               = arm-fsl-linux-gnueabi-g++ -lts
QMAKE_LINK              = arm-fsl-linux-gnueabi-g++ -lts
QMAKE_LINK_SHLIB        = arm-fsl-linux-gnueabi-g++ -lts

# modifications to linux.conf
QMAKE_AR                = arm-fsl-linux-gnueabi-ar cqs
QMAKE_OBJCOPY           = arm-fsl-linux-gnueabi-objcopy
QMAKE_STRIP             = arm-fsl-linux-gnueabi-strip

load(qt_config)
```

(3) Create [build_qt.sh]script file, add the following configuration content:

```
# !bin/bash
./configure     -prefix /home/work/qtenv/qt-4.8.5-arm \
 -opensource \
 -confirm-license    \
 -embedded arm \
 -xplatform qws/linux-arm-g++ \
 -platform /qws/linux-x86-g++ \
 -little-endian \
 -host-little-endian \
 -shared \
 -no-qt3support \
 -no-phonon -no-phonon-backend \
 -qt-zlib \
 -no-gif \
 -no-libtiff \
 -no-qvfb \
 -qt-libjpeg \
 -no-nis \
 -no-opengl \
 -no-cups \
 -no-webkit \
 -no-glib \
 -no-dbus \
 -no-rpath \
 -no-mmx -no-3dnow \
 -no-sse -no-sse2 -no-sse3 -no-ssse3 -no-sse4.1 -no-sse4.2 \
 -no-avx -no-neon \
 -no-audio-backend \
 -no-svg \
 -no-javascript-jit \
 -no-script \
 -no-scripttools \
 -no-multimedia \
 -no-openssl \
 -nomake tools \
 -qt-mouse-tslib \
 -I/home/work/qtenv/tslib/include \
 -L/home/work/qtenv/tslib/lib
```

Note:

What's the meaning of this command, especially, the last two commands:

```
I/home/work/qtenv/tslib/include
L/home/work/qtenv/tslib/lib
```

That corresponds to the [tslib] installation path.

[-prefix $HOME/qtenv/qt-4.8.5-arm] corresponds to the installation path for the afterward path. If there is no such directory ,the users should create firstly.

(4) Execute [build_qt.sh] file, configure [qt] compilation rules.

```
sh   build_qt.sh
 “Which edition of Qt do you want to use?”
 Type 'c' if you want to use the Commercial Edition.
 Type 'o' if you want to use the Open Source Edition.
```

Then, input [o], select [Yes]. The setting method is as follows:

```
sh build_qt.sh
Which edition of Qt do you want to use ?

Type 'c' if you want to use the Commercial Edition.
Type 'o' if you want to use the Open Source Edition.

o


This is the   Open Source Edition.

You are licensed to use this software under the terms of
the Lesser GNU General Public License (LGPL) versions 2.1.
You are also licensed to use this software under the terms of
the GNU General Public License (GPL) versions 3.

Type '3' to view the GNU General Public License version 3.
Type 'L' to view the Lesser GNU General Public License version 2.1.
Type 'yes' to accept this license offer.
Type 'no' to decline this license offer.

Do you accept the terms of either license? yes
```

(5)make

(6)make install

After installation , it will generate qt library and demo program in installation directory.

3、 After compilation and installation , you should set on the system environmental variables, then you can compile your own program.

```
vim   setARMenv.sh
#!/bin/sh
export   QTEDIR=/home/work/qtenv/qt-4.8.5-arm:$QTEDIR
export   PATH=/home/work/qtenv/qt-4.8.5-arm/bin:$PATH
export   LD_LIBRARY_PATH=/home/work/qtenv/qt-4.8.5-arm/lib: LD_LIBRARY_PATH
```

Note: The above directory is the path for installing qt library.

Ⅳ.Compile Qt program

Upper computer compiling program, take example of Qiyang QT , qiyang QT test code:

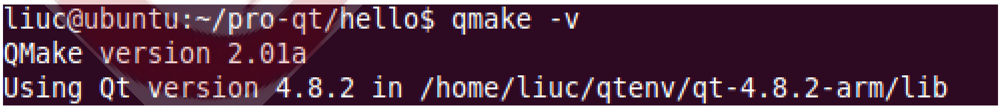
```
Imx6_qt_test.tar.gz
```

Coding in ubuntu, position to the code directory.

Execute [source   setARMenv.sh] to modify environmental variables.

Then check [qmake]'path whether it is correct.

```
qmake -v
```



```
liuc@ubuntu:~/pro-qt/hello$ qmake -v
QMake version 2.01a
Using Qt version 4.8.2 in /home/liuc/qtenv/qt-4.8.2-arm/lib
```

Exectue commands:

```
$ qmake -project       //     Generate enginnering file.pro
$ qmake                       //   Generate makefile
$ make                         //   Imx6_qt_test   Generate executable file [IMX6_QT_TEST].
```

Or use qtcreat to compile.

Ⅴ.Immigrate library to board

1 ） Compile and install arm QT library to your installation directory in Virtual Machine, such as:[/home/work/qtenv/qt-4.8.5-arm/lib], execute the below command in [qt-4.8.5-arm/] directory, it will be used in future.

```
tar   zcvf   lib.tar.gz   lib/
```

2 ） Immigrate to board

Extract [lib.tar.gz] to [/usr/lib] directory on board, [lib] directory includes library and font.

After installation , this step is very important, you must set the environmental variables.

Execute below command in root directory of board:

```
$ vi /etc/profile
```

Then add contents:

```
export LD_LIBRARY_PATH=/usr/lib:$LD_LIBRARY_PATH
export QT_QWS_FONTDIR=/usr/lib/fonts
```

Save and exit!

Prepare [tslib] file, enable touch panel to work

Copy file from [virtual machine/home/work/qtenv/tslib] directory to the relative file location in board.

[tslib/bin] to the board’s file in [/usr/bin]directory.

[tslib/bin] to the board’s file in [/usr/bin]directory.

[tslib/include] to the board’s file in [/usr/include]directory.

[tslib/etc] to the board’s file in [/etc]directory

Modify touch panel configuration file

3.1 Environmental Variables

To achieve Tslib correct running, you need to configure to the Tslib environmental variables.

Set environmental variables as follows: (Or in Shell format)

```
export TSLIB_FBDEVICE=/dev/fb0
export TSLIB_PLUGINDIR=/usr/lib/ts
export TSLIB_TSDEVICE=/dev/input/event0
export POINTERCAL_FILE=/etc/pointercal
export TSLIB_CALIBFILE=/etc/pointercal
export TSLIB_CONFFILE=/etc/ts.conf
export QWS_MOUSE_PROTO=tslib:/dev/input/event0
```

Or you can write into [/etc/profile].

3.2

（1） Support touch panel

After calibrating touch panel, you can execute the following command to support touch panel:

```
export QWS_MOUSE_PROTO=Tslib:/dev/input/event0
```

（2） Support Mouser

```
export QWS_MOUSE_PROTO=Intellimouse:/dev/mouse1
```

（3）

```
 export QWS_MOUSE_PROTO='tslib:/dev/input/event0 Intellimouse:/dev/input/mouse1'
```

Then exectue test program [./Imx6_qt_test -qws]

Then, you can see the Qt interface.

Zhejiang Qiyang Intelligent Technology Co., Ltd

Tel：0571-87858811 / 87858822

Fax：0571-89935912

Technical support：0571-87858811-805

E-MAIL：supports@qiyangtech.com

Website： http://www.qiyangtech.com

Add：5/F, Building 3A, 8 Xiyuanyi Road, West Lake Science Park, Hangzhou,Zhejiang,China

Postcode：310012