kontron

Qtopia Core Application Note

January 29, 2009

Contents

1 Qtopia core application note

This document explains how Qtopia Core Opensource from Trolltech can be compiled for a Kontron Linux 2.6.X BSP. It was tested with kernel version 2.6.21 including realtime patches on an ThinkIO-Duo, but it should be possible to use it on higher kernel version and on different hardware, too.

1.1 Linux BSP

Kontron offers an embedded linux distribution which can be used for several Kontron hardware. The distribution includes the a cross compiler, the Linux kernel, Linux tools for the root file system, HW specific drivers and some shell scripts. After installing these packages on a desktop computer running linux (red hat, suse, ubuntu, ...) the shell scripts can be used to cross compile the kernel and root-fs. Afterwards it can be installed on an CF card and bootet on the target HW.

However, this distribution does not include a graphical user interface like X11 with KDE or Gnome as desktop Linux distributions do.

To be able to run Qtopia programs the BSP must provide two device nodes

- /dev/input/mice for mouse inputs
- /dev/fb0 for graphical output

1.2 Qtopia

Qtopia is a set of libraries for developing a graphical user interface for embedded applications. It does not include a window manager like KDE, but an user interface of a machine control software running on a touchscreen display does not need a window manager. The footprint of Qtopia is relatively small which is advantageous if the target has only an small flash disk.

Trolltech offers Qtopia core under two different licenses: the GNU General Public License (GPL) and a commercial license. The GPL version can only be used to develop open source software following the terms of the GPL. If commercial software must be developed, a commercial license for Qtopia must be ordered from Trolltech. This How-To was only tested with the opensource variant of Qtopia, but it should work with the commercial version also.

Kontron does not sell or distribute the open source or commercial version of Qtopia. It is the responsibility of the application developer to get the Qtopia sources with the appropriate license from Trolltech.

1.3 Versions

This procedure was tested with the following versions:

Qtopia Core 4.4.0 from http://trolltech.com:

- qt-embedded-linux-opensource-src-4.4.0.tar.gz

ThinkIO-Duo Linux BSP R20 from Kontron Modular Computers

- lin26-swp-thinkiod.r20

Crosscompiler

- crosstool-x86-i686-linux-0102.noarch.rpm

1.4 Build procedure

1.4.1 Extract the source tarball

First all files must be extracted from the tar archive.

```
tar xvzf qt-embedded-linux-opensource-src-4.4.0.tar.gz
```

1.4.2 Configure for cross build

Open the file mkspecs/qws/linux-x86-g++/qmake.conf with a text editor, and modify the following the following declarations to point to your cross toolchain.

```
QMAKE_CC            = /opt/crosstool/i686-pentium-linux-gnu/
                          gcc-3.4.5-glibc-2.3.6/bin/i686-pentium-linux-gnu-gcc
QMAKE_CXX           = /opt/crosstool/i686-pentium-linux-gnu/
                          gcc-3.4.5-glibc-2.3.6/bin/i686-pentium-linux-gnu-g++
QMAKE_INCDIR        = /opt/crosstool/i686-pentium-linux-gnu/
                           gcc-3.4.5-glibc-2.3.6/i686-pentium-linux-gnu/include
QMAKE_LIBDIR        = /opt/crosstool/i686-pentium-linux-gnu/
                           gcc-3.4.5-glibc-2.3.6/i686-pentium-linux-gnu/lib
QMAKE_LINK          = /opt/crosstool/i686-pentium-linux-gnu/
                           gcc-3.4.5-glibc-2.3.6/bin/i686-pentium-linux-gnu-g++
QMAKE_LINK_SHLIB    = /opt/crosstool/i686-pentium-linux-gnu/
                           gcc-3.4.5-glibc-2.3.6/bin/i686-pentium-linux-gnu-g++
```

Ensure that the paths for X11 and OpenGL are empty:

```
QMAKE_INCDIR_X11 =
QMAKE_LIBDIR_X11 =
QMAKE_INCDIR_OPENGL =
QMAKE_LIBDIR_OPENGL =
```

1.4.3 Build package

The variable INSTALLPREFIX holds the name of the directory where the Qtopia runtime files will be installed to. It makes life simplier, if the directory is identical for the development host and the target, and in the following the directory /qtopia was chosen. Create this directory on your development host (as root user), and chown it to your normal user. Then, enter the directory qt-embedded-linux-opensource-src-4.4.0 and execute the following commands:

```
export QPEDIR=$PWD
export INSTALLPREFIX=/qtopia
./configure -prefix ${INSTALLPREFIX} -embedded x86 -xplatform qws/linux-x86-g++
    -qt-gfx-linuxfb -little-endian
make
make install
```

1.5 Add Touchscreen support with Tslib

The library tslib is used to support touchscreen input devices for QT applications. Tslib can be obtained from http://tslib.berlios.de/

If touchscreen support is needed, tslib must be crosscompiled first and installed to the Qtopia installation directory /qtopia. This can be achived with the following commands:

```
export CROSS_COMPILE=/opt/crosstool/i686-pentium-linux-gnu/
                          gcc-3.4.5-glibc-2.3.6/bin/i686-pentium-linux-gnu-

export AS="${CROSS_COMPILE}as"
export LD="${CROSS_COMPILE}ld"
export CC="${CROSS_COMPILE}gcc"
export CPP="${CC} -E"
export AR="${CROSS_COMPILE}ar"
export NM="${CROSS_COMPILE}nm"
export STRIP="${CROSS_COMPILE}strip"
export OBJCOPY="${CROSS_COMPILE}objcopy"
export OBJDUMP="${CROSS_COMPILE}objdump"

export INSTALLPREFIX=/qtopia
export LIBPREFIX=${INSTALLPREFIX}/lib
export INCLUDEPREFIX=${INSTALLPREFIX}/include

autoreconf -f -i -I `pwd`/m4
libtoolize --force --copy
aclocal
autoheader
automake --add-missing --copy
autoconf
./configure --prefix=${INSTALLPREFIX} --libdir=${LIBPREFIX} --includedir=${INCLUDEPREFIX}

make
make install
```

Then the following parameters must be used for configure instead of the those shown in the last section.

```
cd qtopia-core-opensource-src-4.3.3
export QPEDIR=$PWD
export INSTALLPREFIX=/qtopia
./configure -prefix ${INSTALLPREFIX} -L${INSTALLPREFIX}/lib -I${INSTALLPREFIX}/include
    -qt-mouse-tslib-embedded x86 -xplatform qws/linux-x86-g++
    -qt-gfx-linuxfb -little-endian
make
make install
```

On the target a touchscreen driver must be running that offers the touch events on the device /dev/input/tsX. The used format must be configured in the file ts.conf. The location of ts.conf and the input device must be defined in environment variables:

```
export TSLIB_CONFFILE=/root/ts.conf
export TSLIB_TSDEVICE=/dev/input/ts0
```

tslib offers some helper programs:

'ts_calibrate' can be used to calibrate the touchscreen. /etc must be writable when tc_calibrate is run, because the result of the calibration will be stored there.

'ts_test' can be used to test the touchscreen driver.

If this works, an additional environment variable must be set to tell Qtopia the input device.

```
export QWS_MOUSE_PROTO=tslib:/dev/input/ts0
```

Then the Qtopia application can be started.

1.6 Installation on target

Now all files needed to run Qtopia have been copied to the directory '/qtopia'. Copy the directory to the target filesystem (CF or NFS-directory). Now you can log in to the target and call one of the example programs. It is necessary to add the parameter '-qws'.

```
/qtopia/examples/mainwindows/application/application -qws
```

1.7 Optimization hints

The build process described above compiles all host tools and examples. Therefore the contents of the '/qtopia' directory may be to much for a small storage device:

```
58M     /qtopia/bin
18M     /qtopia/demos
63M     /qtopia/doc
102M    /qtopia/examples
12M     /qtopia/include
184M    /qtopia/lib
1,4M    /qtopia/mkspecs
284K    /qtopia/phrasebooks
16M     /qtopia/plugins
328K    /qtopia/q3porting.xml
1,6M    /qtopia/translations
```

The only directory needed is the 'lib' subdirectory. All files with extension .debug can also be deleted. Then there are only 42MB left.

```
42M     /qtopia/lib
```