

## Qt库移植--基于海思平台，应该通用的。

Qt编译过程可以参看GoodMan大神的编译过程

地址：<http://www.ebaina.com/bbs/forum.php?mod=viewthread&tid=8086&extra=page%3D1>

我的Qt库的编译安装路径为/opt/Qt4.8

就是在编译的时候指定 --prefix /opt/Qt4.8

编译完成后的目录大概是这样的

```
[root@bogon:1 opt]# tree Qt4.8/ -L 1
```

```
Qt4.8/
├── bin
├── include
├── lib
├── mkspecs
├── plugins
└── translations
```

之后将海思sample目录下的东西拷贝到这里

```
[root@bogon:1 opt]# tree Qt4.8/ -L 1
```

```
Qt4.8/
├── bin
├── hilib      // mpp 目录下的lib文件夹的内容都放在这里
├── hinclude   // mpp 目录下的include下的所有文件都放在这里
├── include
├── lib
├── mkspecs
├── plugins
└── translations
```

因为我们的目的是在代码中复用sample下的基础代码，然而这部分代码与内核代码，以及外部驱动代码都有一定的联系

我之前试着将这部分代码抽离出来，后来没成功。考虑到可能要做的移植工作很多，最后还是决定将qt工程嵌入到sample

目录下，直接使用其各个目录下的Makefile文件，因为他们都已经定义好了编译规则，我们只需要参考其他的工程文件的Makefile即可定义好我们的规则。

开动：

Qt库我就不说了，按照自己的配置也可以，只要将相应的库文件放进去就行了，库文件移植到板子的时候，只需要留下so动态库文件即可，其余的静态库.a文件什么的，以及工具都删除即可，将库放到板子的同目录就可以了，在指定库搜索路径添加你的动态库路径。

我之前写了个qt的测试程序，拖了几个控件，在windows上编译通过。之后我们将该工程代码放到sample目录下，大概是这样的：

```
[root@bogon:1 sample]# tree -L 1
```

```
.
├── audio
├── common
├── guiclick    //该目录为从测试工程目录，其中存放着代码
├── hifb
├── LIB
├── Makefile
├── Makefile.param
├── QtBuild     //该目录为新建的编译目录。
├── region
├── tde
├── vda
├── vdec
├── venc
├── vio
└── work
```

然后在~/.bashrc 中添加一行PATH=\$PATH:/opt/Qt4.8.5

运行source ~/.bashrc 使变量生效，之后运行qmake如果有正常的输出，就是配置好了，若没有检查相应的错误。

之后就开始我们的移植之路

我们将工程文件放到sample目录下。工程目录大概这样的：

```
[root@bogon:1 sample]# tree -L 1 guiclick/
```

```
guiclick/
├── guiclick.pro
├── guiclick.pro.user
├── main.cpp
├── mainwindow.cpp
└── mainwindow.h
```

└─mainwindow.ui

然后我们在sample下新建一个编译目录: mkdir Qtbuild,

进入Qtbuild,

之后使用命令[root@bogon:1 QtBuild]# qmak -r ../guiclick/guiclick.pro生成Makefile

然后修改Makefile文件内容

修改Makefile内容 变更:

```
9 ##### Compiler, tools and options
10 include ../Makefile.param // 添加一行: 添加sample目录编译控制文件。
11 CC      = arm-hisiv100nptl-linux-gcc

14 #CFLAGS      := -pipe -O2 -Wall -W -D_REENTRANT $(DEFINES)
//因为在上层的Makefile.parram中定义了CFALGS, 而我们的编译需要依赖该
//选项, 所以这里修改为+=
15 CFLAGS      += -pipe -O2 -Wall -W -D_REENTRANT $(DEFINES)
16 CXXFLAGS     = -pipe -O2 -Wall -W -D_REENTRANT $(DEFINES)

16 CXXFLAGS     = -pipe -O2 -Wall -W -D_REENTRANT $(DEFINES)
17 INCPATH      := -I/opt/Qt4.8/mkspecs/qws/linux-arm-hisiv100nptl-g++ -I../guiclick -I/opt/Qt4.8/include/QtCore -
I/opt/Qt4.8/include/QtNetwork -I/ opt/Qt4.8/include/QtGui -I/opt/Qt4.8/include -I. -I. -I../guiclick -I.
//这里是添加海思库的头文件支持。
18 INCPATH      += -I/opt/Qt4.8/hinclud -I.

21 LIBS        := $(SUBLIBS) -L/opt/Qt4.8/lib -lQtGui -L/opt/Qt4.8/lib -lQtNetwork -lQtCore -lpthread
//添加海思库文件依赖, 为了方便, 我们把所有的库文件都附后
22 LIBS        += -L/opt/Qt4.8/hilib -lmpi -lhdm -ltde -lanr -laec -ljpeg -lresampler -lVoiceEngine
23 AR          = arm-hisiv100nptl-linux-ar cqs

86 .SUFFIXES: .o .c .cpp .cc .cxx .C
87
88 .cpp.o:
89 $(CXX) -c $(CXXFLAGS) $(INCPATH) -o "$@" "$<" $(MPI_LIBS) $(AUDIO_LIBA) $(JPEGD_LIBA)
// 添加这部分主要是在编译main.cpp的时候需要到这些库的支持, 这些库在上层的Makefile.param中定义了
90

104
105 all: Makefile $(TARGET)
106 // 编译的目标文件依赖于公共代码, 就是上层的common文件夹下的代码, 使用该部分代码可以快速开发
107 $(TARGET): ui_mainwindow.h $(OBJECTS) $(COMM_OBJ)
// 添加连接参数, 加入海思库文件以及公共代码的编译o文件
108 $(LINK) $(LFLAGS) -o $(TARGET) $(OBJECTS) $(OBJCOMP) $(COMM_OBJ) $(LIBS)

219
220 main.o: ../guiclick/main.cpp
// CFLAG指定了common文件夹的头文件, 而我们使用的sample的实例代码主要还是这部分, 如果不加CFLAG
//会有hi_common.h, 以及某些驱动头文件找不到的错误, 在编译的时候也需要指定库
221 $(CXX) -c $(CXXFLAGS) $(CFLAGS) $(INCPATH) -o main.o ../guiclick/main.cpp
222
```

之后是guiclick/main.cpp文件内容的更改, 主要是在该文件中对相关内容进行初始化。代码在二楼给出

之后就是可以编译了。运行make即可编译出来了。编译出来放到板子上运行就好咯。

帖子地址<http://www.ebaina.com/bbs/forum.php?mod=post&action=edit&fid=140&tid=8217&pid=18505&page=1>