



杭州启扬智能科技有限公司

## Qt 4.8.2 的移植及应用

版本号 v1.0

2012/09/05

杭州启扬智能有限公司版权所有  
**QIYANG TECHNOLOGY Co., Ltd**  
**Copyright Reserved**

## 目 录

前言.....	1
一、概述.....	2
二、添加触摸屏库的支持.....	2
三、安装Qt库.....	3
四、编译Qt程序.....	5
五、移植库到开发板.....	6
六、am335x Qt主界面相关功能.....	8
6.1 串口通信.....	9
6.2 音频测试.....	10
6.3 设置.....	11
6.3.1 时间设置.....	11
6.3.2 语言设置.....	11
6.4 屏幕校准.....	12
6.5 屏幕字体.....	12

## 前言

欢迎使用杭州启扬智能科技有限公司产品工控开发板，本手册对 Qt4.8.2 的移植进行了说明。

## 一、概述

硬件资源: IAC-335X-Kit 开发板

宿主机: Ubuntu 10.10

Qt 源码: qt-everywhere-opensource-src-4.8.2.tar.gz

触摸屏库: tslib1.4.tar.gz

交叉编译工具:

arm-arago-linux-gnueabi.tar.gz

## 二、添加触摸屏库的支持

1、先确定交叉编译工具是否安装正确。

2、编译触摸屏支持的库

2.1 在 Ubuntu 命令行中分别执行以下几个升级命令。

```
sudo apt-get install libtool
```

```
sudo apt-get install autoconf
```

```
sudo apt-get install automake
```

2.2 解压 tslib1.4.tar.gz

```
cd tslib
```

2.3 然后执行命令,

```
./configure --host=arm-linux ac_cv_func_malloc_0_nonnull=yes
```

```
CC=arm-arago-linux-gnueabi-gcc CXX=arm-arago-linux-gnueabi-g++
```

```
-prefix=/home/liuc/qtdev/tslib
```

注意: 交叉编译工具要正确, -prefix 为安装目录, 编译安装后的

tslib 在/home/liuc/qtdev/tslib 文件夹下面。

2.4 编译并安装

```
make && make install
```

2.5 配置

编译后会在/home/liuc/qtdev/目录下出现 tslib。进入 tslib 目录, 编辑'ts.conf'文件

```
cd tslib/etc
```

```
gedit ts.conf
```

编辑该文件, 去掉 'module\_raw input'前面的 # 号和空格。

如果'module\_raw input' 前面留有空格的话,运行测试软件时会提示 'Segmentation fault'。

## 三、安装 Qt 库

### 1、解压 Qt 源码

```
tar zxvf qt-everywhere-opensource-src-4.8.2.tar.gz
```

### 2、编译

(1)cd qt-everywhere-opensource-src-4.8.2

(2)指定交叉编译器

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

编辑 qmak.conf, 将 “arm-linux ” 更改为

“arm-arago-linux-gnueabi “, 并增加参数 “lts ”

到四个选项, 如下所示:

```
#
# 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-arago-linux-gnueabi-gcc -lts
QMAKE_CXX          = arm-arago-linux-gnueabi-g++ -lts
QMAKE_LINK         = arm-arago-linux-gnueabi-g++ -lts
QMAKE_LINK_SHLIB  = arm-arago-linux-gnueabi-g++ -lts

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

load(qt_config)
```

(3)新建 build\_qt.sh 脚本文件，增加以下配置内容：

```

# !bin/bash

./configure -prefix $HOME/qtenv/qt-4.8.2-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/liuc/qtdev/tslib/include \
-L/home/liuc/qtdev/tslib/lib
```

注意：这些命令都是什么意思，尤其是最后两个-I/home/liuc/qtdev/tslib/include

-L/home/liuc/qtdev/tslib/lib，是对应相应的 tslib 的安装路径

-prefix \$HOME/qtdev/qt-4.8.2-arm 是对应以后这个程序的安装路径

(4) 执行 build\_qt.sh 文件，配置 qt 编译规则

```
sh build_qt.sh
```

(5)make

(6)make install

安装完成后，在安装目录下生成了 qt 库以及 demo 程序。

3、完成编译和安装后需要对系统的环境变量进行设置，才能编译自己的程序

```
vim setARMenv.sh
```

```
#!/bin/sh
```

```
export QTEDIR=/home/liuc/qtdev/qt-4.8.2-arm:$QTEDIR
```

```
export PATH=/home/liuc/qtdev/qt-4.8.2-arm/bin:$PATH
```

```
export LD_LIBRARY_PATH=/home/liuc/qtdev/qt-4.8.2-arm/lib: LD_LIBRARY_PATH
```

注意：以上目录是你安装 qt 库的路径。

## 四、编译 Qt 程序

1. 上位机编译程序，以 am3359 Qt 界面程序为例

在虚拟机的 ubuntu 下，写好代码之后，定位到代码目录，

先执行 source setARMenv.sh 修改环境变量

然后查看 qmake 的路径是否正确

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

执行命令

```
$ qmake -project // 生成工程文件.pro
```

```
$ qmake // 生成 makefile
```

```
$ make //生成可执行文件 am3359
```

2.或者使用 qtcreat 编译。

## 五、移植库到开发板

1 ) 虚拟机里编译安装 arm 版的 Qt 库在你的安装目录下, 如:

/home/liuc/qtenv/qt-4.8.2-arm/lib 下, 在 qt-4.8.2-arm/目录执行下面的命令打包, 方便以后移植使用

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

2 ) 移植到开发板

最好是跟你在虚拟机里面的 arm 版 Qt 所在目录一样, 方便操作!

在开发板下新建目录: /home/liuc/qtenv/qt-4.8.2-arm/

把 lib.tar.gz 解压到此目录, lib 目录下包含库和字体。

2. 安装成功后, 这步非常关键, 一定要设置环境变量

在开发板的根目录下执行:

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

然后输入

```
export QTDIR=/home/liuc/qtenv/qt-4.8.2-arm:$QTDIR
```

```
export LD_LIBRARY_PATH=/home/liuc/qtenv/qt-4.8.2-arm/lib:$LD_LIBRARY_PATH
```

```
export QT_QWS_FONTDIR=/home/liuc/qtenv/qt-4.8.2-arm/lib/fonts
```

保存退出!

3. 准备 tslib 文件, 使触摸屏工作

把虚拟机/home/liuc/qtenv/tslib 目录下的文件拷贝到开发板上的

/home/liuc/qtenv/tslib, 没有的话自己建个文件。

修改触摸屏配置文件

3.1 环境变量

为了实现 Tslib 的正确运行, 需要对如下的 Tslib 的环境变量进行配置:

设置环境变量如下: (或者写成 shell 形式)

```
export T_ROOT=/home/liuc/qtenv/tslib/
```

```
export LD_LIBRARY_PATH=/home/liuc/qtenv/tslib/lib:$LD_LIBRARY_PATH
```

```
export TSLIB_CONSOLEDEVICE=none
```

```
export TSLIB_FBDEVICE=/dev/fb0
```

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

```
export TSLIB_PLUGINDIR=$T_ROOT/lib/ts
```

```
export TSLIB_CONFFILE=$T_ROOT/etc/ts.conf
```

```
export POINTERCAL_FILE=/etc/pointercal
```

```
export TSLIB_CALIBFILE=/etc/pointercal
```

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

或者也写在/etc/profile 里面.

### 3.2

#### (1) 支持触摸屏

在对触摸屏校准以后，可以执行以下命令来支持触摸屏：

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

#### (2) 支持鼠标

```
export QWS_MOUSE_PROTO=MouseMan:/dev/input/mice
```

#### (3) 同时支持触摸屏和鼠标

```
export set QWS_MOUSE_PROTO="TSLIB:/dev/input/event0 Intellimouse:/dev/input/mice"
```

然后执行测试程序 `./am3359 -qws`

就可以看到 Qt 界面出现了。

## 六、am3359 Qt 主界面相关功能

系统以自启动方式运行 Qt 程序，启动开发板后即可进入 Qt 主界面，如下图 6.1 所示，该界面主要包括串口通信、音频测试、设置和屏幕校准等桌面图标，以及屏幕底部的状态栏包括左边的启扬公司的 logo 和右方的时间显示。

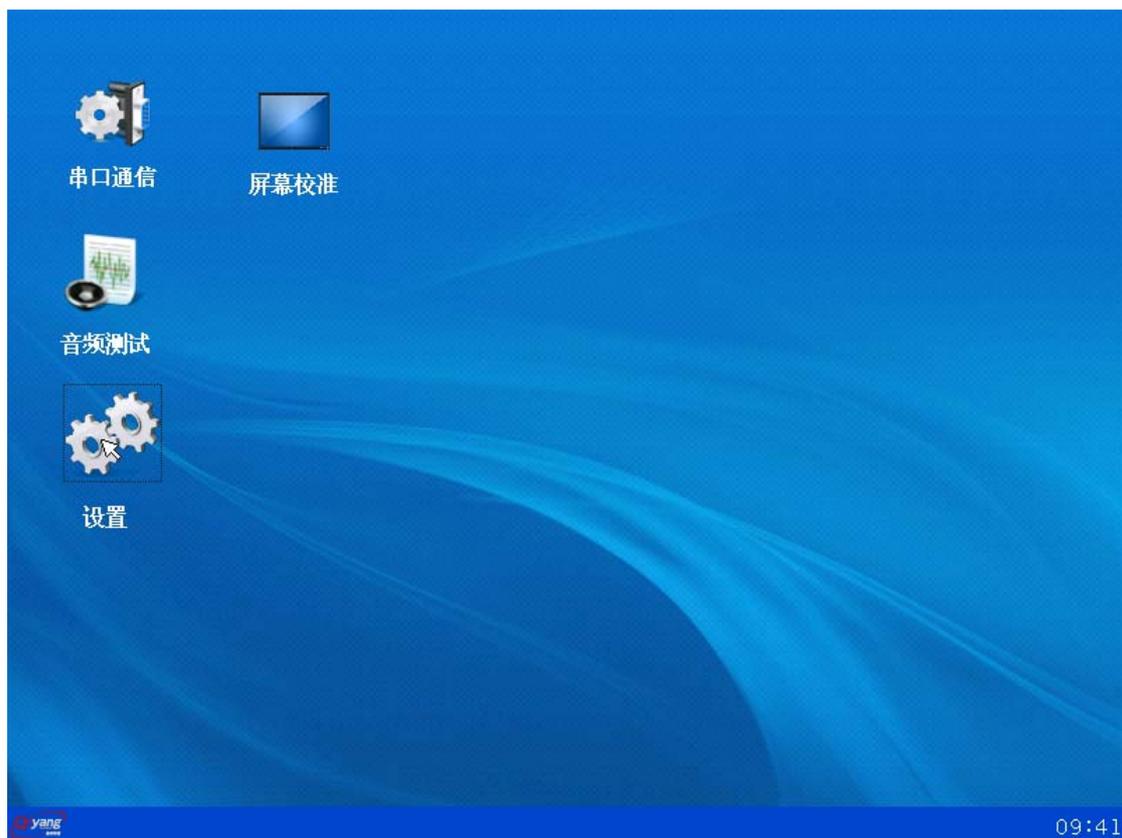


图 6.1

## 6.1 串口通信

串口通信需要先设置端口、波特率、校验位、数据位、停止位，然后再点击打开串口，按钮变成关闭串口之后，波特率等参数不能再进行设置，如果需要设置，需要关闭串口之后，重新设置。

打开成功之后就能接收和发送数据了。

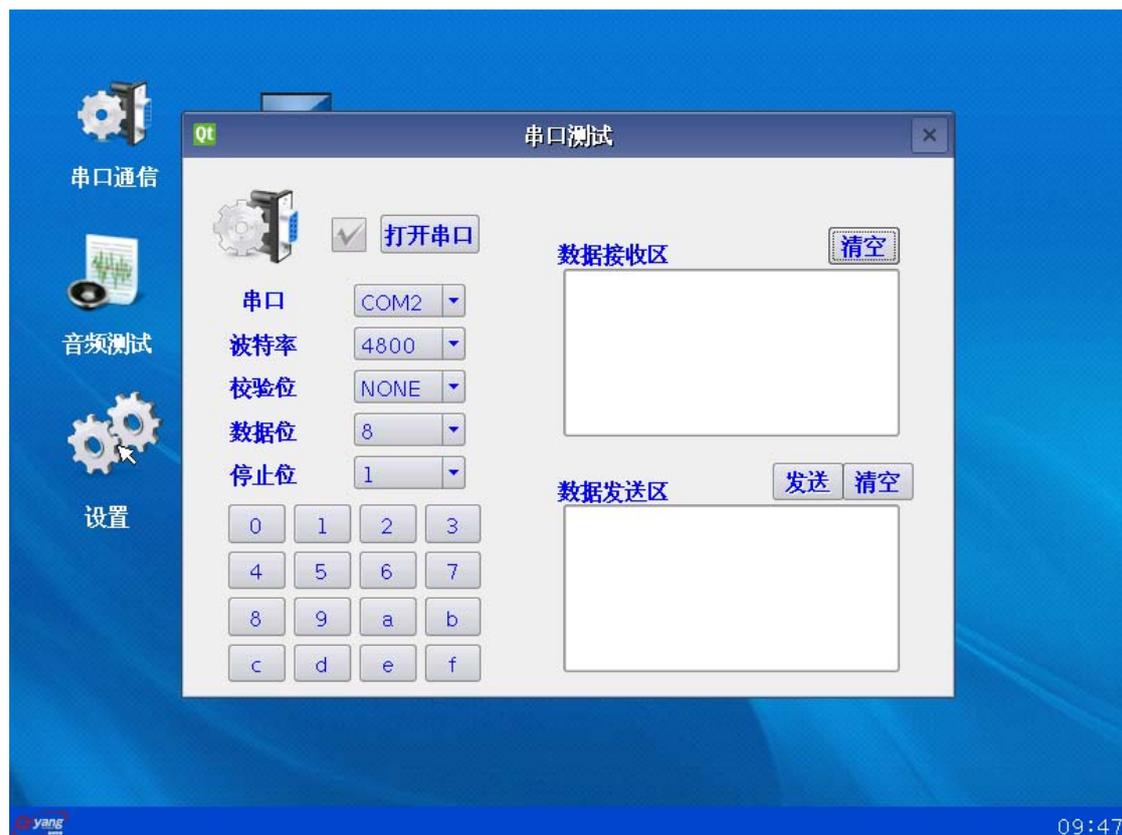


图 6.2

## 6.2 音频测试

音频测试功能主要包括播放、停止按钮和音量控制滑块，及显示提示画面。



图 6.3

## 6.3 设置

设置中包括时间设置和语言设置功能，如下图 6.4 所示：



图 6.4

### 6.3.1 时间设置

设置画面操作很简单，设置完成后，如果不保存，点击取消即可。点击保存后，系统时间改成你所设置的时间。

### 6.3.2 语言设置

包括中文和英文，选择所要显示的语言，点击保存后，画面全部转换成你所设置的语言，如下图 6.4 为设置成的英文界面：

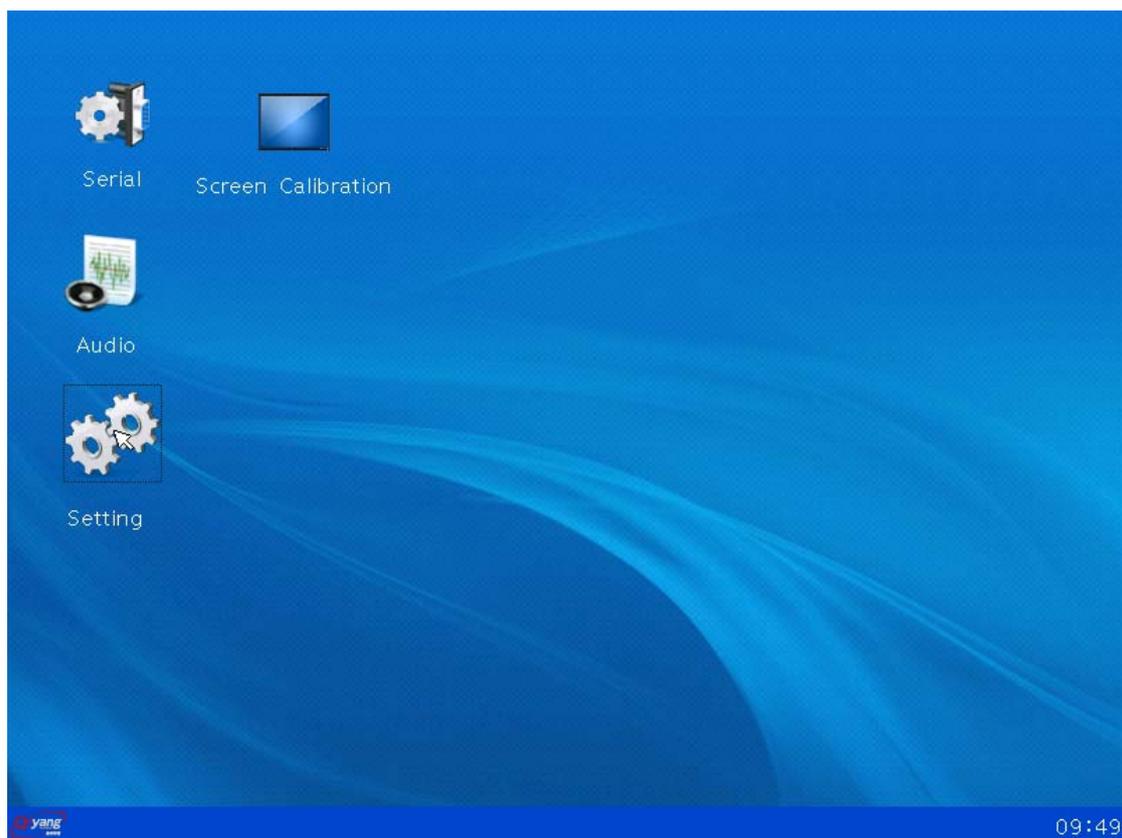


图 6.5

## 6.4 屏幕校准

点击屏幕校准之后，直接进入校准界面，校准之后出现一个对话框，显示是否要重新校准，重启后校准生效，如果你刚才校准不准确可以重新校准一次，然后重启或者取消。

## 6.5 屏幕字体

使用的是 wenquanyi\_160\_75. qpf 字体，开发包里提供，可以选择下载自己喜欢的字体，放在/home/liuc/qtENV/qt-4.8.2-arm/lib/fonts，如果只需要一种字体可以把其他字体全部删除，只保留所需要的字体。

## 杭州启扬智能科技有限公司

电话：0571-87858811 / 87858822

传真：0571-89935912

支持：0571-89935913

E-MAIL: [supports@qiyangtech.com](mailto:supports@qiyangtech.com)

网址： <http://www.qiyangtech.com>

地址：杭州市西湖科技园西园一路 8 号 3A 幢 5 层

邮编：310012