



IAC-335X-KIT Embedded Development Board qt-4.8.2

Immigration Manual

Version:1.0
2012/09/05

HANGZHOU QIYANG INTELLIGENT TECHNOLOGY Co., Ltd

Copyright Reserved

Any question, please send E-mail:supports@qiyangtech.com

Page 1 of 20

Sales E-mail:trade@qiyangtech.com_sales@qiyangtech.com

Website:http://www.qiytech.com

©2014 Qiyangtech Copyright

Catalogue

PREFACE	2
I .SUMMARY	2
II .ADD THE SUPPORT FOR TOUCH PANEL LIBRARY	2
III.INSTALL QT LIBRARY	4
IV.COMPILE QT PROGRAM	8
V.IMMIGRATE LIBRARY TO DEVELOPMENT BOARD	9
VI. EZ335X-EVB QT INTERFACE FUNCTION	12
THE INTERFACE INCLUDES SERIAL, AUDIO, SETTINGS AND SCREEN CALIBRATION, AND THE QIYANG’S LOGO ON THE LEFT BOTTOM AND THE TIME ON THE RIGHT BOTTOM IN STATUS BA	13
6.1 SERIAL COMMUNICATION	13
6.2 AUDIO TEST.....	14
6.3 SETTING	15
6.3.1 Time Setting	16
6.3.2 Language Settings	16
6.4 SCREEN CALIBRATION	17
6.5 SCREEN FONT	17

Preface

Welcome to use the development board from Hangzhou Qiyang Intelligent Technology Co.,Ltd. This manual is used for illustrating the qt-4.8.2.

I .Summary

Hardware Source:[IAC-335X-KIT development Board]

Host Machine: [ubuntu 10.10]

Qt source code: [qt-everywhere-opensource-src-4.8.2.tar.gz]

Touch Panel Library:[tslib1.4.tar.gz]

Cross Compiler Tool: [arm-arago-linux-gnueabi.tar.gz]

II .Add the support for touch panel library

1. Ensure the cross compiler tool is installed correctly.
 2. Compile the library for supporting touch panel
- ①. Execute the following commands in ubuntu command line.

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

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

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

②.Extract [tslib.1.4.tar.gz].

```
$ cd tslib
```

③.Execute the following command:

```
$/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/qtenv/tslib
```

Notice: The cross compiler tool must be right. [-prefix] is the installation directory.

The finished [tslib] will be in [/home/liuc/qtenv/tslib] folder.

④. Compile and install

```
$ make
```

```
$ make install
```

⑤.Configure

After compiling, there will be [tslib] in [/home/liuc/qtenv/]directory. Enter into [tslib]directory, edit ['ts.conf']file.

```
$ cd tslib/etc
```

```
$ gedit ts.conf
```

Remove [#] and spacing before [module_raw input], the default is all closed. If there

is any [spacing] before[module_raw input],it will hint [Segmentation fault] when you testing the software.

III.Install qt library

1.Extract [qt-4.8.2] Qt source code

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

2.Compile

①Enter into qt directory

```
$ cd qt-everywhere-opensource-src-4.8.2
```

②. Appoint the cross compiler

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

Edit [qmak.conf], modify [arm-linux] as [arm-arago-linux-gnueabi], and add parameter[Its].

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-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. Make new script file[build_qt.sh]. Add the following configuration content:

```
# !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 \

-qt-libjpeg \

-no-nis \

-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 \

Any question, please send E-mail:supports@qiyangtech.com

Page 6 of 20

Sales E-mail:trade@qiyangtech.com,sales@qiyangtech.com

Website:<http://www.qiytech.com>

©2014 Qiyangtech Copyright

```
-no-audio-backend \
```

```
-no-svg \
```

```
-no-javascript-jit \
```

```
-no-script \
```

```
-no-scripttools \
```

```
-no-multimedia \
```

```
-no-openssl \
```

```
-nomake tools \
```

```
-qt-mouse-pc -qt-mouse-tslib \
```

```
-I/home/liuc/qtenv/tslib/include \
```

```
-L/home/liuc/qtenv/tslib/lib
```

Notice: The last two [-I/home/liuc/qtenv/tslib/include] and [-L/home/liuc/qtenv/tslib/lib] which are corresponding to [tslib] installation path. [-prefix \$HOME/qtenv/qt-4.8.2-arm] which is corresponding to [qt]program installation path.

④.Execute [build_qt.sh]file, configure [qt] compilation rule.

```
$ sh build_qt.sh
```


⑤. make

⑥. make install

Finish installation, generate [qt] library and [demo] program.

3. After finishing compilation and installation , we should set on the system environment variables. Then you could finish compiling on your own program.

```
vim setARMenv.sh
```

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

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

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

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

```
LD_LIBRARY_PATH
```

Notice: The above directory is the path which is used for installing [qt]library.

IV. Compile qt program

Upper computer compile program, example: am3359 qt Interface Program.

In ubuntu of virtual machine, after writing the code, locate it into code directory.

At first, execute

```
$ source setARMenv.sh
```

then modify environment variables.

Then check whether the [qmake]path 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
```

Execute comand

```
$ qmake -project //Generate project file [.pro]
```

```
$ qmake // Generate [makefile]
```

```
$ make //Generate executable file am3359
```

2.Or compile by [qtcreeat].

V.Immigrate library to development board

1. Compile Qt library in arm version at your installation directory in Virtual

Machine,such as:

```
/home/liuc/qtenv/qt-4.8.2-arm/lib
```

The comand be packed in [qt-4.8.2-arm]directory. Easy for immigration in later.

```
$ tar -zcvf lib.tar.gz lib/
```

2. Immigrate it to development board

For convenient operation, create a directory in development board as virtual

Any question, please send E-mail:supports@qiyangtech.com

Page 9 of 20

Sales E-mail:trade@qiyangtech.com,sales@qiyangtech.com

Website:<http://www.qiytech.com>

©2014 Qiyangtech Copyright

machine:

Creat a new directory in development board.

```
/home/liuc/qtenv/qt-4.8.2-arm/
```

Extract [lib.tar.gz] into this directory, [lib]directory includes the libraries and fonts.

3. After immigrating and installing library file, we need to set the environment variables.

Execute in the root directory of the development:

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

Then input some environment variables in the file.

```
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_P
```

```
ATH
```

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

Save & Exit

Please check the path of the enviornment variables and the path of actual corresponding path.The above paths could be setted according to the user's actual path. It just needs to let the enviornment variables accord to the atctual library files.

4.Add [tstlib]file, make the touch panel work normally

Any question, please send E-mail:supports@qiyangtech.com

Page 10 of 20

Sales E-mail:trade@qiyangtech.com,sales@qiyangtech.com

Website:<http://www.qiytech.com>

©2014 Qiyangtech Copyright

Copy the file in [virtual machine/home/liuc/qtenv/tslib]directory to
[/home/liuc/qtenv/tslib] in development board. If there is no such directory, please
create it.

5.Modify touch panel configuration file

In order to let [tslib] working correctly, it needs to configure the [tslib] environment
variables:

Set environment variables, as follows: (Or in [shell] format)

```
$ 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
```

Or write them into [/etc/profile] directly.

6. Make qt program to support touch panel and mouse

①. Support touch panel

After calibration on the touch panel, execute the following command to support the touch panel:

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

② Support mouse

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

③. Support touch panel and mouse at the same time

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

```
Intellimouse:/dev/input/mice"
```

Then execute test program

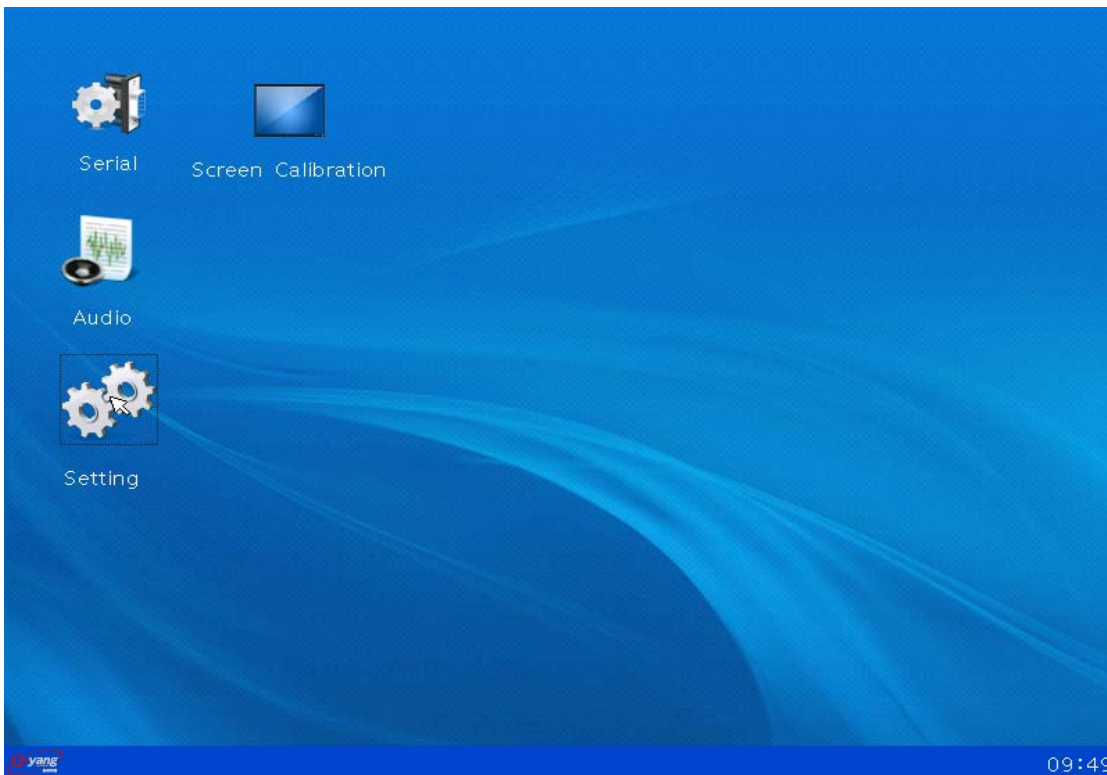
Then execute test program [./am3359 -qws], then it shows the qt interface.

VI. Am3359 qt Interface Function

System use the auto-start mode to run Qt program, after running development board,

then enter into Qt main interface, as the following picture shown:

Picture 6.1

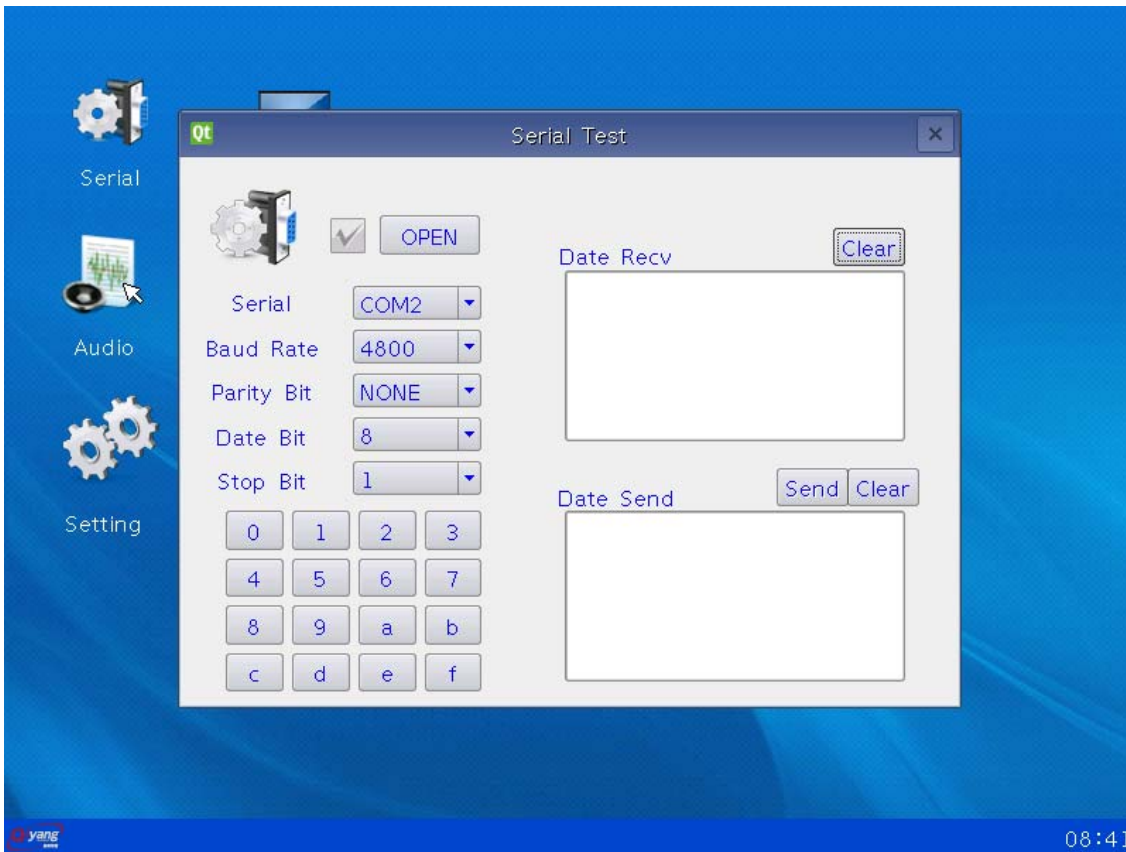


The interface includes Serial, Audio, Settings and Screen Calibration, and the Qiyang's logo on the left bottom and the time on the right bottom in status bar.

6.1 Serial Communication

Set serial port, baud rate, parity, data bit and stop bit. Then click [open], and the button is changed to be [close]. Do not set baud rate again; if you need to set, close the serial port first then set again.

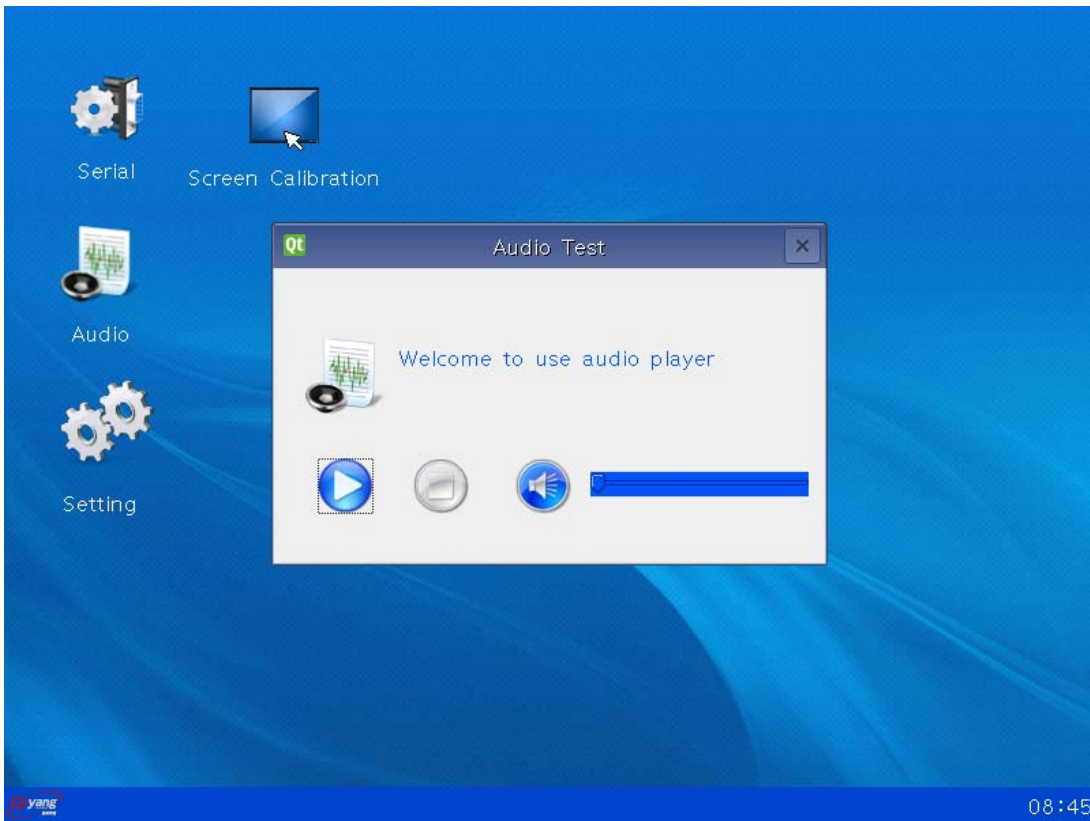
Can send and receive data after opening successfully. Input at the left keyboard at the corresponding dispatching area, receive the data at the receiving area.



Picture 6.2

6.2 Audio Test

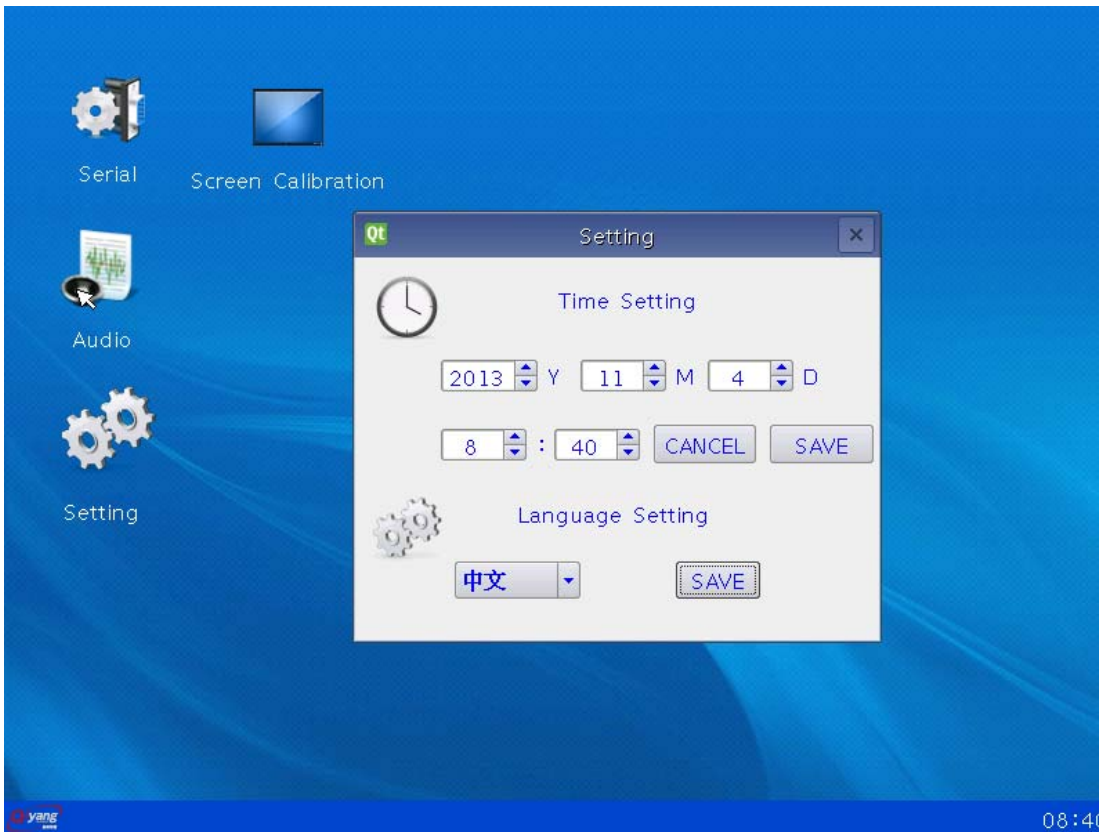
Audio test function includes play, stop, volume control, hint picture.



Picture 6.3

6.3 Setting

Setting includes time setting and language setting function, as the picture 6.4 shown:



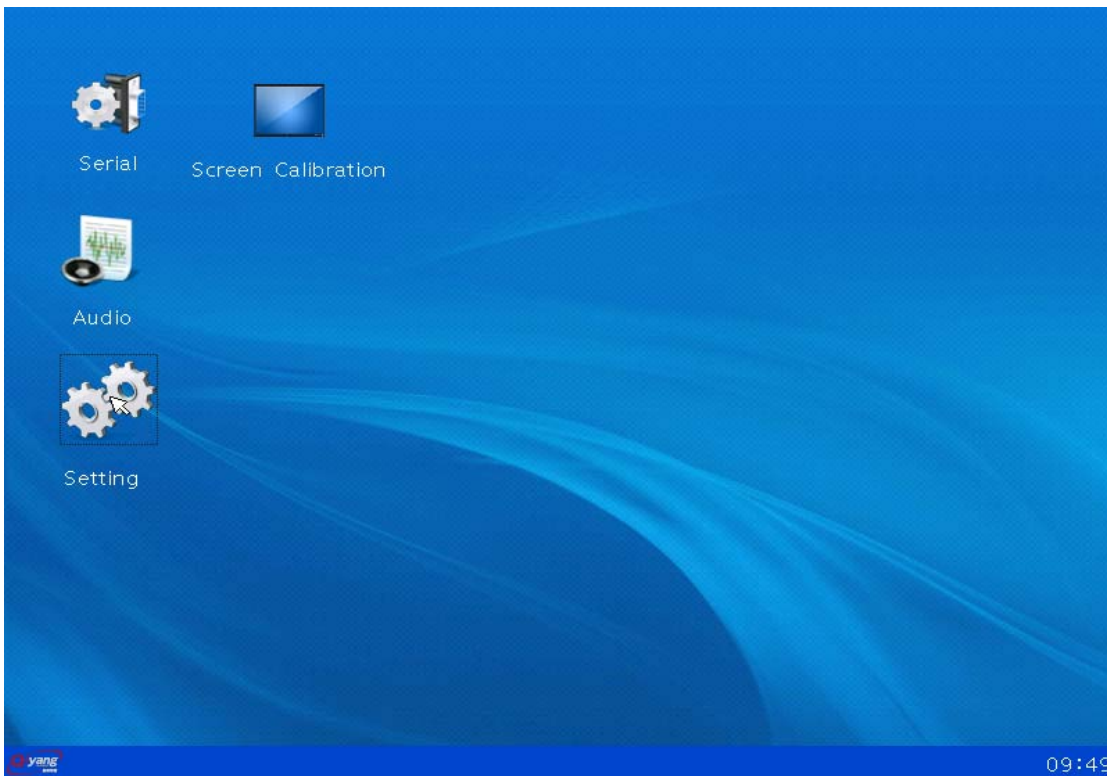
Picture 6.4

6.3.1 Time Setting

Setting screen operation is very easy, setup finished, if you don't save, click [cancel]; after click [save], the system time will change to the time you set.

6.3.2 Language Settings

Select the language, click [save] ,English interface is as shown:



Picture 6.5

6.4 Screen Calibration

Click screen calibration, enter the calibration interface directly, appear a dialog after calibrating, display whether to recalibrate, calibration takes effect after restarting. If calibration is not accurate, you can calibrate again, then restart or cancel.

6.5 Screen Font

The default font is [wenquanyi_160_75.qpf], user can also download fond font, correct time automatically and put it into the

Any question, please send E-mail:supports@qiyangtech.com

Page 17 of 20

Sales E-mail:trade@qiyangtech.com_sales@qiyangtech.com

Website:http://www.qiytech.com

©2014 Qiyangtech Copyright

[/home/liuc/qtENV/qt-4.8.2-arm/lib/fonts] directory.



Hangzhou Qiyang Intelligent Technology Co., Ltd

Any question, please send E-mail:supports@qiyangtech.com

Page 18 of 20

Sales E-mail:trade@qiyangtech.com_sales@qiyangtech.com

Website:<http://www.qiytech.com>

©2014 Qiyangtech Copyright

Tel: 86 -571-87858811 / 87858822

Fax: 86-571-89935912

Technology Support: 86-571-89935913

E-MAIL: supports@qiyangtech.com

Website: <http://www.qiytech.com>

Address: 5F, Building 3A, NO.8 Xiyuanyi Road, West Lake

Science Park, Hangzhou, China

Post Code: 310030

