



# **A40i 系列 & T3 系列 \_Linux-5.10\_QT\_ 配置指南**

**版本号: 1.0**  
**发布日期: 2022.10.18**

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2022.10.18	AWA1586	初始版本

# 目 录

<b>1 前言</b>	<b>1</b>
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
<b>2 QT 适配介绍</b>	<b>2</b>
2.1 前提	2
2.2 主要介绍	2
2.3 具体代码	2
2.4 模块配置介绍	3
2.4.1 内核配置	3
2.4.2 QT 动态库配置	3
2.4.3 qt_demo-launcher 配置	6
<b>3 模块使用范例</b>	<b>9</b>
<b>4 测试验证</b>	<b>11</b>
<b>5 FAQ</b>	<b>12</b>
5.1 Q1: 样机无 qt 画面显示	12
5.2 Q2: 样子显示异常花屏	12

## 插图

图 2-1	target-packages	3
图 2-2	select-graphic-library	4
图 2-3	select-qt5	4
图 2-4	qt5-select-1	5
图 2-5	qt5-select-2	5
图 2-6	target-packages	6
图 2-7	select-allwinner-package	7
图 2-8	select-qt_demo	7
图 2-9	select-launcher	8
图 4-1	launcher	11

# 1 前言

## 1.1 文档简介

介绍 QT 模块的使用方法，方便开发人员使用。

## 1.2 目标读者

QT 模块的驱动开发/维护人员。

## 1.3 适用范围

表 1-1: 适用产品列表

buildroot 版本	配置文件
buildroot202205	package/qt5

## 2 QT 适配介绍

### 2.1 前提

由于 sdk 对 buildroot 版本进行了升级，qt 的编译也需要进行一定程度的适配。

本次环境依赖如下：

软件	版本
buildroot	buildroot-202205
qt5	qt-5.15（默认）
qt_demo	demo-v1.0

### 2.2 主要介绍

本文主要介绍 TQ 系统以下几项

- 如何在 buildroot 工具里编译 QT 动态库；
- 编译及运行 qtdemo 应用程序；
- 适配过程遇到的问题。

### 2.3 具体代码

```
# buildroot/config仓库
./buildroot/qt_demo/launcher // qt_demo配置
./buildroot/allwinner/system/busybox-init-base-files/etc/init.d //launcher运行

# buildroot/buildroot-202205仓库
./dl/qt5base # 源代码文件
./dl/qt5multimedia # 源代码文件
./package/qt5 # 编译配置

# buildroot/package/auto/qt_demo仓库
./qt_demo # qt_demo源码
```

## 2.4 模块配置介绍

### 2.4.1 内核配置

QT 依赖显示驱动，需要把显示驱动加载进来，具体配置请参考显示驱动指导指南说明。

### 2.4.2 QT 动态库配置

在项目根路径执行 `./build.sh buildroot_menuconfig`（默认使用 `buildroot-202205` 配置）进入配置页面。

- 首先，选择 `Target packages` 选项进入下一级配置，如下图所示。



图2-1: target-packages

- 在 `Target packages` 配置项下选择 `Graphic libraries and applications`，如下图所示。

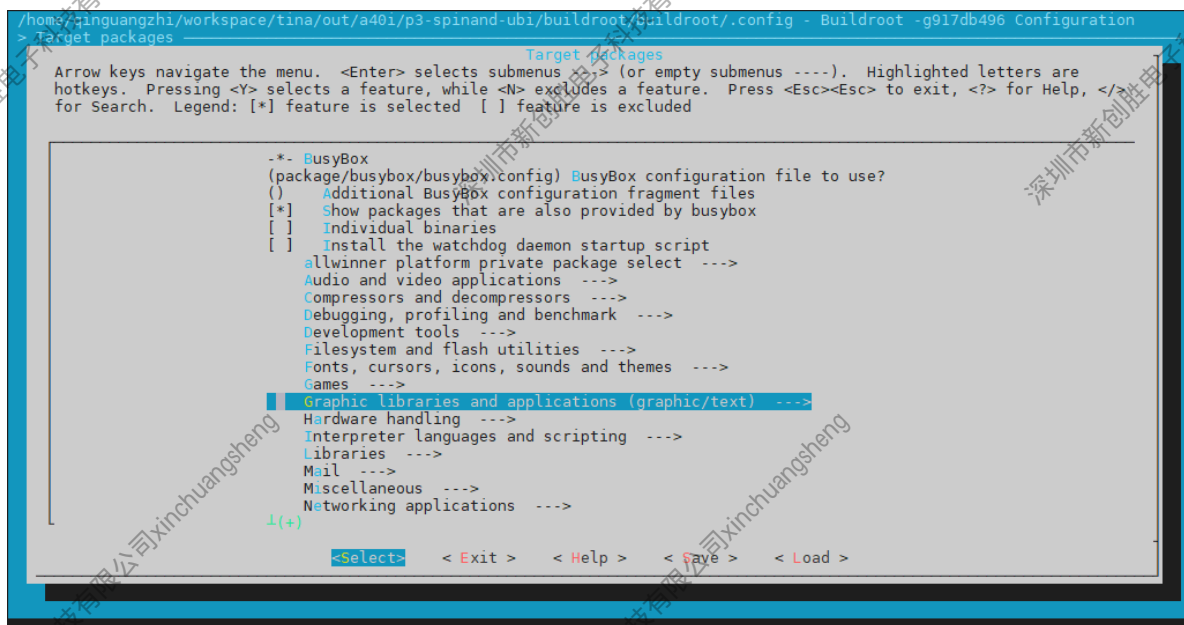


图 2-2: select-graphic-library

- 在 Graphic libraries and applications 配置项中选择 Qt5, 如下图所示。

注意，如果服务器的 gcc 版本较低，是不会显示 Qt5 的，需按照提示更新到要求版本。

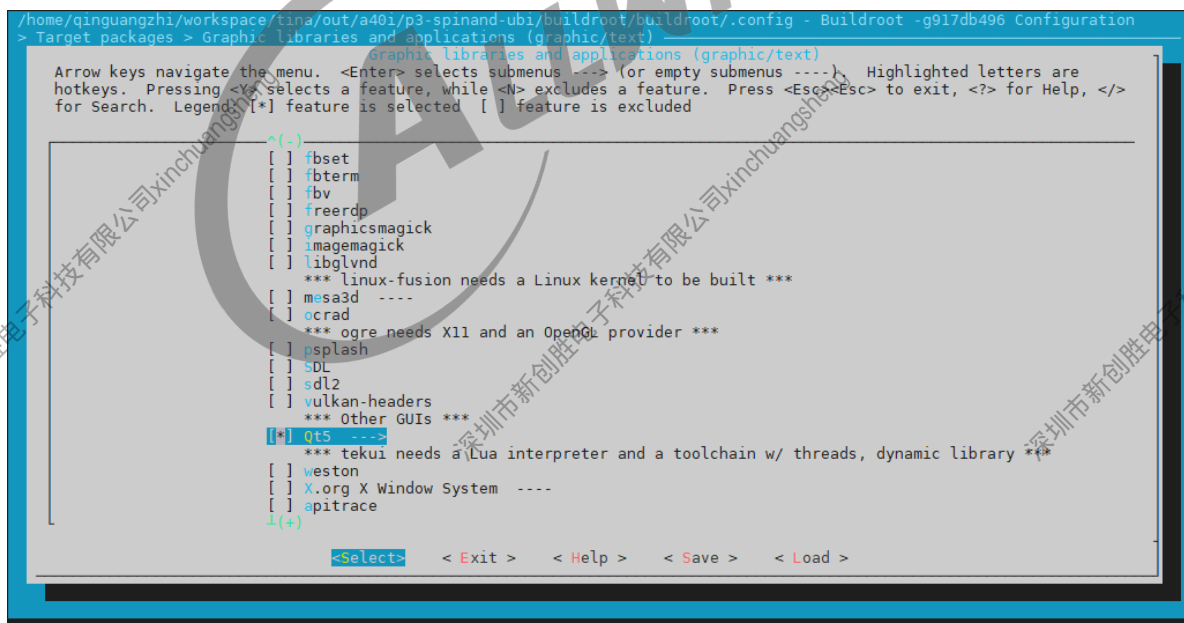


图 2-3: select-qt5

- 在 Qt5 中配置项下选择以下这些选项, 如下图所示。



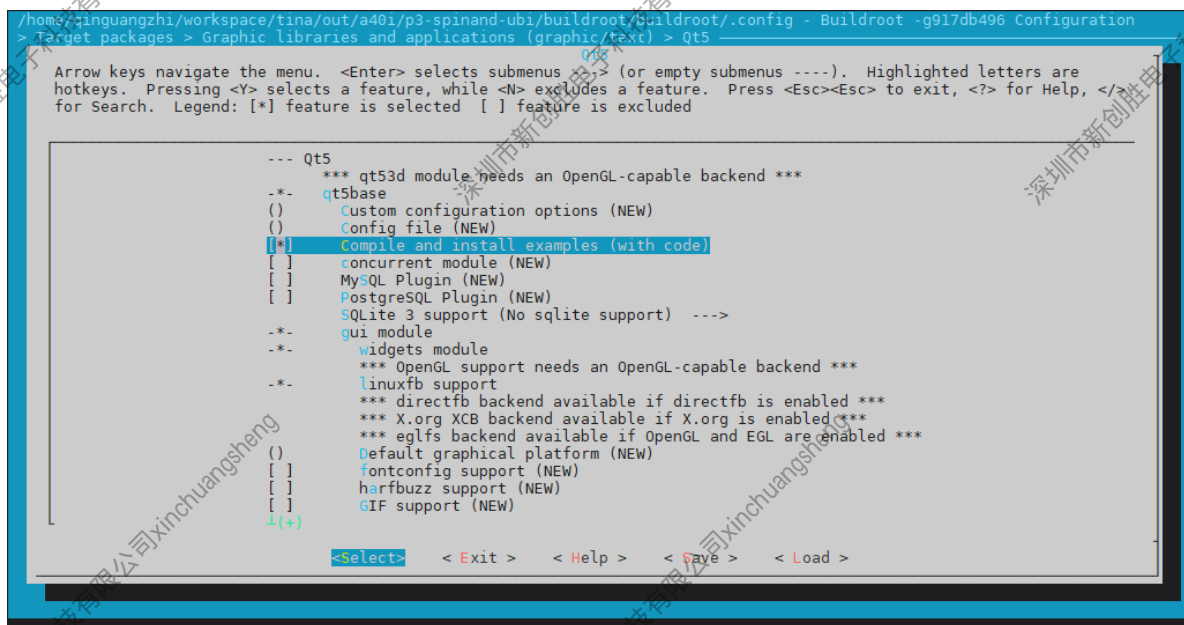


图 2-4: qt5-select-1

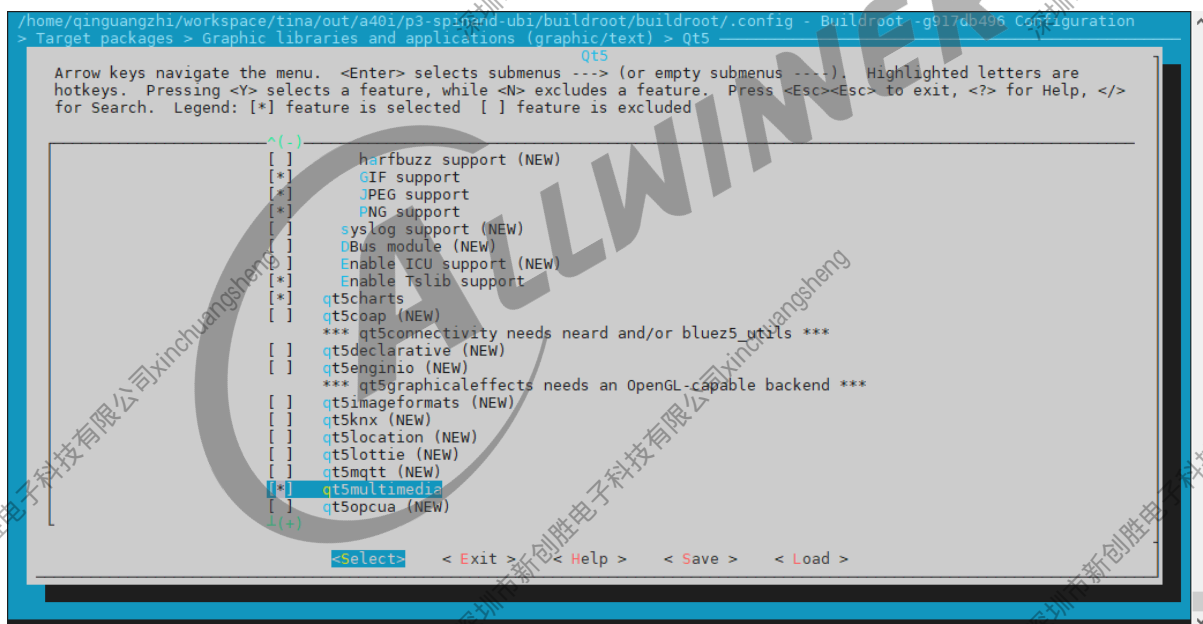


图 2-5: qt5-select-2

以上操作会把一些基本能验证的 QT 动态库编译出来，下面的动作可以把修改的配置保存到配置文件。

```
# 以下路径均基于sdk所在路径为依据
cd buildroot/buildroot-202205/
make savedefconfig
```

### 2.4.3 qt\_demo-launcher 配置

在项目根路径执行./build.sh buildroot menuconfig（默认使用 buildroot-202205 配置）进入配置页面。

- 首先，选择 Target packages 选项进入下一级配置，如下图所示。

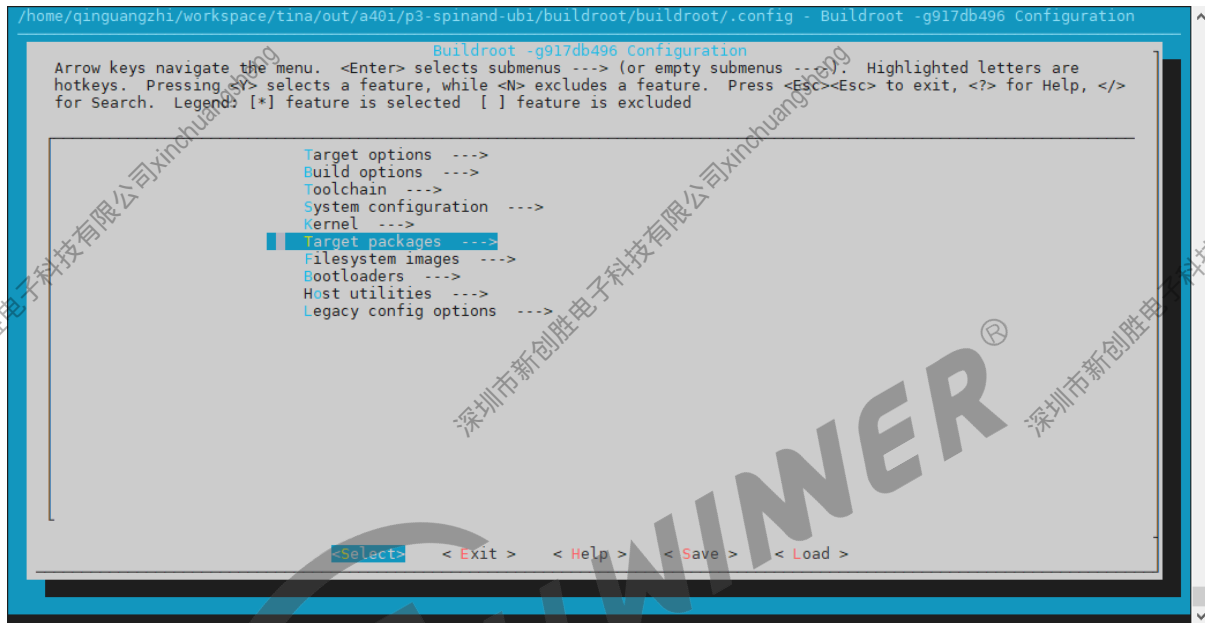


图 2-6: target-packages

- 在 Target packages 配置项下选择 allwinner platform private package select，如下图所示。

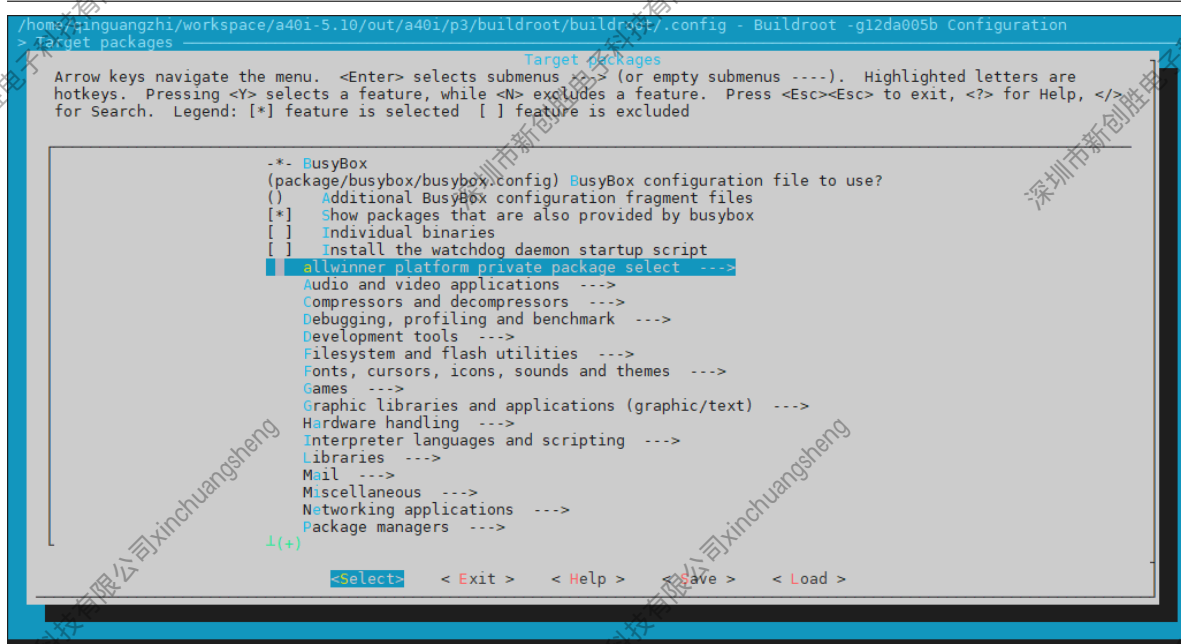


图 2-7: select-allwinner-package

- 在 allwinner platform private package select 配置项中选择 qt\_demo, 如下图所示。

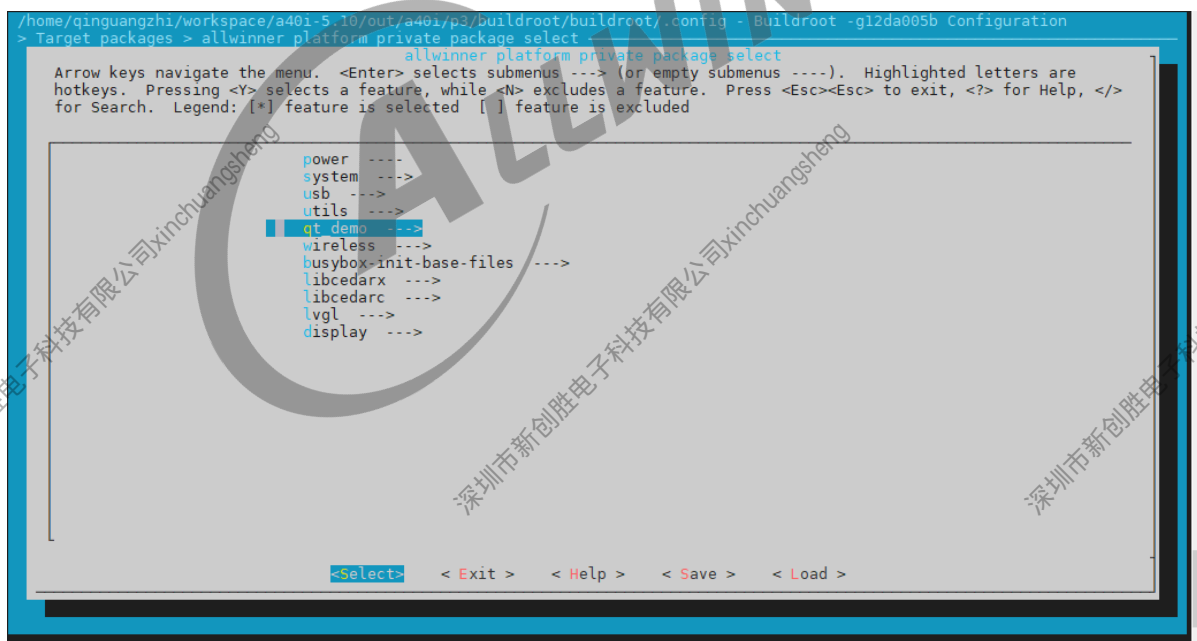


图 2-8: select-qt\_demo

- 在 qt\_demo 配置项中选择 Launcher for qt, 如下图所示。

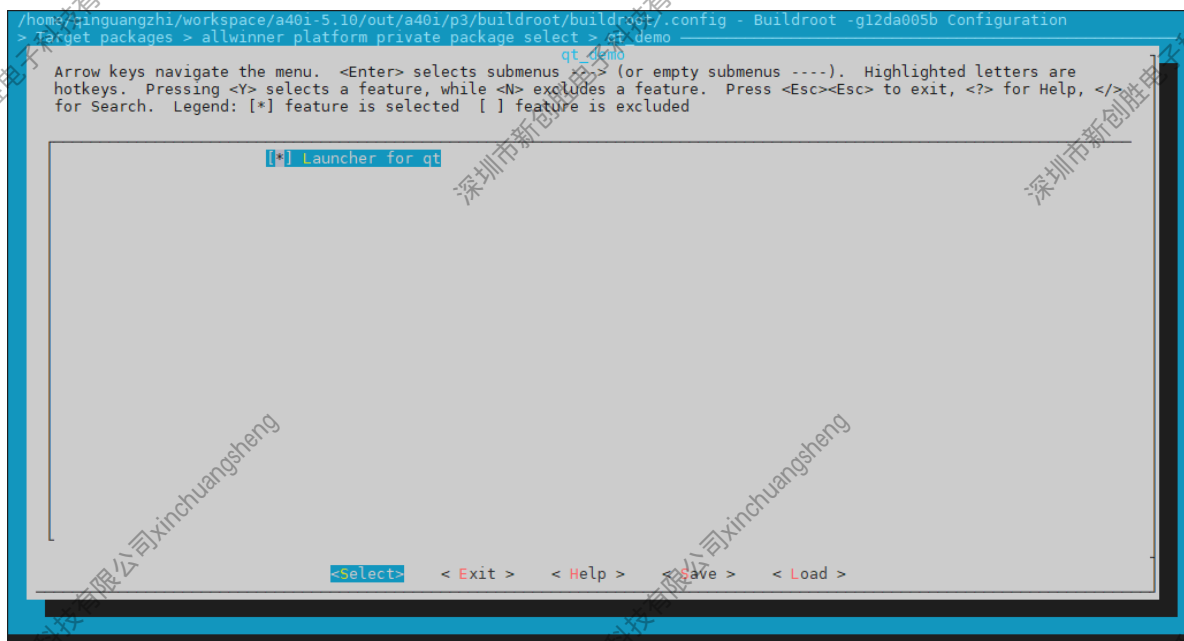


图 2-9: select-launcher

以上操作会把一些基本能验证的 QT demo (launcher) 编译出来，下面的动作可以把修改的配置保存到配置文件。

```
# 以下路径均基于sdk所在路径为依据
cd buildroot/buildroot-202205/
make savedefconfig
```

### 3 模块使用范例

如下是在 busybox 上增加一个开机自启动任务启动 QTdemo。

```
#
# buildroot/config/buildroot/allwinner/system/busybox-init-base-files/etc/init.d/
# S70launcher
#
case "$1" in
  start)
    if [ ! -f "/usr/bin/Launcher" ];then
      exit 1
    fi
    export QTDIR=/usr/lib
    if [ -d $QTDIR ];then

      export QT_ROOT=$QTDIR
      export PATH=$QTDIR/bin:$PATH
      export LD_LIBRARY_PATH=$QTDIR/lib:/usr/lib/cedarx/:$LD_LIBRARY_PATH

      export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_ROOT/plugins
      export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
      export QT_QPA_FONTDIR=$QT_ROOT/fonts

      TouchDevice=gt9xxnew_ts
      for InputDevices in /sys/class/input/input*
      do
        DeviceName=`cat $InputDevices/name`
        if [ $DeviceName == $TouchDevice ];then
          TouchDeviceNum=${InputDevices##*input}
          export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/
event$TouchDeviceNum
          echo "add "/dev/input/event$TouchDeviceNum "to Qt Application."
          break
        fi
      done
      if [ ! -n "$TouchDeviceNum" ]; then
        echo "Error:Input device $TouchDevice can not be found,plz check it!"
      fi

      export QT_QPA_FONTDIR=/usr/lib/fonts
      export QT_QPA_GENERIC_PLUGINS=tslib
      #export QT_QPA_GENERIC_PLUGINS=evdevmouse:/dev/input/event4
      export TSLIB_FBDEVICE=/dev/fb0
      export TSLIB_CONSOLEDEVICE=none
      export TSLIB_TSDEVICE=/dev/input/event4
      export TSLIB_CONFFILE=/etc/ts.conf
      export TSLIB_CALIBFILE=/etc/poointercal
      export TSLIB_PLUGINDIR=/usr/lib/ts

      export QWS_MOUSE_PROTO=
      mkdir -p /dev/shm
      ulimit -c unlimited
    fi
  esac
```

```
Launcher &  
fi  
;;  
stop)]  
;;  
*)  
    echo "Usage: $0 {start}"  
    exit 1  
    ;;  
esac  
  
exit 0
```

## 4 测试验证

编译固件并烧录到样机中，系统启动后，如屏幕显示如图案，这表示 qt 应用程序及动态库运行正常。



图 4-1: launcher



## 5 FAQ

### 5.1 Q1：样机无 qt 画面显示

优先考虑 launcher 是否都正常运行，通过查看 ps 查看 launcher 任务正常运行，

```
# ps -a | grep Luancher
1656 root      grep Luancher
# 这样表示qt demo有运行
```

如果后台没有运行 launcher 没有运行，查看程序是否编译到样机上，或者运行的相关路径是否正确。

如果程序有在运行，但是无显示，可以查看显示驱动是否正常运行，或者 qt 启动的配置是否与该平台符合（如是否使能 gpu，显示节点是否匹配等）

### 5.2 Q2：样子显示异常花屏

优先考虑是否硬件出现问题，如屏幕排线或者是否连接不稳等，使用同样的 sdk 在其他样机查看，是否必现。

如果必现，同样优先让显示的同事使用显示自带接口调试驱动是否异常，若不行，就看 qt 程序和动态库使用端的问题。






## 著作权声明

版权所有 © 2022 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。