

5SSD201 & SSD202

Qt5 环境搭建

© 2020 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.

REVISION HISTORY

Revision No.	Description	Date
0.1	<ul style="list-style-type: none">Initial release	05/29/2020
	<ul style="list-style-type: none">	
	<ul style="list-style-type: none">	

TABLE OF CONTENTS

REVISION HISTORY i

TABLE OF CONTENTS..... ii

Qt5 环境搭建..... 3

 1.1. 环境准备 3

 1.2. Qt5 交叉编译..... 4

 1.3. Qt example 6

QT5 环境搭建

1.1. 环境准备

Qt5 下载地址: <http://download.qt.io/>

Name	Last modified	Size	Metadata
■ snapshots/	16-May-2019 14:07	-	
■ online/	13-Mar-2014 08:45	-	
■ official_releases/	03-Jun-2019 11:59	-	
■ new_archive/	20-Jan-2020 20:34	-	
■ ministro/	20-Feb-2017 10:32	-	
■ linguist_releases/	26-Mar-2019 07:49	-	
■ learning/	22-May-2013 16:20	-	
■ development_releases/	17-May-2019 17:44	-	
■ community_releases/	23-Feb-2017 07:29	-	
■ archive/	18-Dec-2019 07:30	-	
■ timestamp.txt	26-May-2020 07:00	11	Details

目录结构说明:

目录	说明
archive	各种 Qt 开发工具安装包, 包含新旧版本, 可以下载开发工具和源码
community_releases	社区定制的 Qt 库, Tizen 版 Qt 以及 Qt 附加源码包
development_releases	开发版, 包含新的和旧的不稳定版本, 在 Qt 开发过程中的非正式版本
learning	有学习 Qt 的文档教程和示范视频
ministro	Mini 版本, 目前是针对 Android 的版本
official_releases	正式发布版, 是与开发板相对的稳定版 Qt 开发库和开发开发工具, 可以下载开发工具和源码
online	Qt 在线安装源
snapshots	预览版, 最新的开发测试中的 Qt 库和开发工具

当前环境使用 archive Qt5.15.0 版本。

<http://download.qt.io/archive/qt/5.15/5.15.0/single/qt-everywhere-src-5.15.0.tar.xz.mirrorlist>

1.2. Qt5 交叉编译

本例使用 toolchain: gcc-arm-8.2-2018.08-x86_64-arm-linux-gnueabihf 编译 qt5。编译依赖 perl5.8 或以上版本, python2.7 或以上版本。要支持使用触摸还需要链接 tslib 库。

下载 toolchain gcc-arm-8.2-2018.08-x86_64-arm-linux-gnueabihf.tar.xz:
<https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-a/downloads>

设置环境变量:

vim /etc/profile 添加 toolchain 路径:

```
export PATH=/opt/toolchain/gcc-arm-8.2-2018.08-x86_64-arm-linux-gnueabihf/bin:$PATH
```

安装 perl, python:

```
sudo apt-get install perl Python2.7
```

安装 git 工具:

```
sudo apt-get install git
```

下载 tslib:

```
git clone https://github.com/libts/tslib.git
```

编译 tslib:

```
sudo apt-get install automake
sudo apt-get install autogen
sudo apt-get install libtool
./autogen.sh
./configure --prefix=/home/koda.xu/workspace/tslib_1.21/arm_tslib --host=arm-linux ac_cv_func_malloc_0_nonnull=yes CC=arm-linux-gnueabihf-gcc
make -j4
make install
```

--prefix 为 tslib 导出的头文件和 lib 存放路径。

编译 Qt5:

下载 Qt5.15.0 opensource code 包 qt-everywhere-src-5.15.0.tar.xz

1. xz -d qt-everywhere-src-5.15.0.tar.xz
2. tar xf qt-everywhere-src-5.15.0.tar
3. cd qt-everywhere-src-5.15.0

4. vi qtbase/mkspecs/linux-arm-gnueabi-g++/qmake.conf

修改如下:

```
1 #
2 # qmake configuration for building with arm-linux-gnueabi-g++
3 #
4
5 MAKEFILE_GENERATOR      = UNIX
6 CONFIG                  += incremental
7 QMAKE_INCREMENTAL_STYLE = sublib
8
9 include(../common/linux.conf)
10 include(../common/gcc-base-unix.conf)
11 include(../common/g++-unix.conf)
12
13 QT_QPA_PLATFORM=linuxfb:fb=/dev/fb0:rotation=0
14
15 # modifications to g++.conf
16 QMAKE_CC                = arm-linux-gnueabi-gcc
17 QMAKE_CXX                = arm-linux-gnueabi-g++
18 QMAKE_LINK               = arm-linux-gnueabi-g++
19 QMAKE_LINK_SHLIB         = arm-linux-gnueabi-g++
20
21 # modifications to linux.conf
22 QMAKE_AR                 = arm-linux-gnueabi-ar cqs
23 QMAKE_OBJCOPY             = arm-linux-gnueabi-objcopy
24 QMAKE_NM                 = arm-linux-gnueabi-nm -P
25 QMAKE_STRIP              = arm-linux-gnueabi-strip
26 load(qt_config)
```

5. 设置 configure

```
./configure \
-prefix /home/koda.xu/Qt/Qt5.15.0/qt5.15_lib \
-static \
-release \
-opensource \
-make libs \
-xplatform linux-arm-gnueabi-g++ \
-optimized-qmake -pch \
-qt-libjpeg \
-qt-libpng \
-qt-zlib \
-no-opengl \
-skip qt3d \
-skip qtcanvas3d \
-skip qtpurchasing \
-skip qtlocation \
-skip qttools \
-no-sse2 \
-no-openssl \
-no-cups \
-no-glib \
-no-iconv \
-nomake examples \
```

```
-tslib \  
-linuxfb \  
-I /home/koda.xu/Qt/tslib_1.21/arm_tslib/include \  
-L /home/koda.xu/Qt/tslib_1.21/arm_tslib/lib
```

```
-prefix 设置部署路径  
-I XXX 设置引用头文件路径  
-I XXX 设置链接库路径
```

6. 编译

```
make -j4  
make install
```

7. 编译完成后将导出的 qt 库 bin 路径添加到 PATH 环境变量中

```
vim ~/.bashrc  
export PATH=/home/koda.xu/workspace/Qt5.15.0/qt5.15.0_lib/bin:$PATH
```

1.3. Qt example

ftp /SSD20X/Demo_Release/UI_DEMO/Qt5.15.0 目录下有 qt example, 以 HelloWorld 为例, 编译 Qt app 方法如下:

1. 进入 HelloWorld 目录, 执行 `qmake -project`, 生成 .pro 文件
编辑 .pro 文件:

```
koda.xu@sigmastar:~/Qt/Qt5.15.0/examples/HelloWorld$ cat HelloWorld.pro  
#####  
# Automatically generated by qmake (3.1) Wed May 27 20:24:33 2020  
#####  
QT += core gui  
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets  
  
CONFIG += c++11  
  
TEMPLATE = app  
TARGET = HelloWorld  
INCLUDEPATH += .  
  
# You can make your code fail to compile if you use deprecated APIs.  
# In order to do so, uncomment the following line.  
# Please consult the documentation of the deprecated API in order to know  
# how to port your code away from it.  
# You can also select to disable deprecated APIs only up to a certain version of Qt.  
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0  
  
# Input  
HEADERS += mainwindow.h  
FORMS += mainwindow.ui  
SOURCES += main.cpp mainwindow.cpp  
TRANSLATIONS += HelloWorld_zh_CN.ts  
QTPLUGIN += qlinuxfb
```

2. `qmake`

3. `make`

在 HelloWorld 目录生成 HelloWorld bin 档

4. 运行 demo 前需要先初始化 disp 和 panel, 见 dispInit, 在该目录执行 `make` 生成档。

运行 qt app:

1. 先运行 dispInit 初始化 disp 和 panel, 可参考 dispInit/run.sh
2. 运行 qt app, 可参考 HelloWorld/run.sh