

昉·惊鸿-7110 Linux Qt开发指南

版本: 1.0 日期: 2025/02/21 Doc ID: JH7110-DGCH-023

法律声明

阅读本文件前的重要法律告知。

版权注释

版权 ©上海赛昉半导体科技有限公司, 2025。版权所有。

本文档中的说明均基于"视为正确"提供,可能包含部分错误。内容可能因产品开发而定期更新或修订。上海赛昉半导体 科技有限公司 (以下简称"赛昉科技")保留对本协议中的任何内容进行更改的权利,恕不另行通知。

赛昉科技明确否认任何形式的担保、解释和条件,无论是明示的还是默示的,包括但不限于适销性、特定用途适用性和 非侵权的担保或条件。

赛昉科技无需承担因应用或使用任何产品或电路而产生的任何责任,并明确表示无需承担任何及所有连带责任,包括但 不限于间接、偶然、特殊、惩戒性或由此造成的损害。

本文件中的所有材料受版权保护,为赛昉科技所有。不得以任何方式修改、编辑或断章取义本文件中的说明,本文件或 其任何部分仅限用于内部使用或教育培训。

联系我们:

地址: 中国(上海)自由贸易试验区盛夏路61弄张润大厦2号电梯楼层5层(实际楼层4层)06室

网站: <u>http://www.starfivetech.com</u>

邮箱:

- <u>sales@starfivetech.com</u>(销售)
- <u>support@starfivetech.com</u>(支持)

Contents

List of Tables	4
List of Figures	5
法律声明	2
	6
1	0 7
1. J 泊	····· / o
2. 编庠和运行SDK	0
2.1. 获取代码和编译	8
2.2. 运行Qt应用程序	8
2.3. 运行Qt Demo程序	9
3. 基于QT库开发UI程序	10
3.1. Qt Creator环境搭建	10
3.1.1. 所需环境	10
3.1.2. 安装Qt Creator	10
3.2. Qt Creator开发UI代码工程	16
3.3. Qt UI代码工程编译	17
3.3.1. 基于Qt Creator工具编译	17
3.3.2. 基于Buildroot环境编译	25
3.3.3. 基于昉·惊鸿-7110 DevKit板上编译	26
4. QT+TP触摸配置	28

List of Tables

目录

Table 0-1 修	多订历史	3
Table 1-1 支	支持的Qt模块	7
Table 3-1 编	 肩译选项说明	5

List of Figures

Figure 3-1 下载Qt	
Figure 3-2 安装界面	
Figure 3-3 点击Next	
Figure 3-4 Setting	13
Figure 3-5 选择组件	
Figure 3-6 开始安装	15
Figure 3-7 完成安装	16
Figure 3-8 UI Design界面	
Figure 3-9 选择Options	
Figure 3-10 选择C++	
Figure 3-11 添加G++编译器	20
Figure 3-12 添加GCC编译器	
Figure 3-13 配置Debuggers	
Figure 3-14 配置Qt Version	
Figure 3-15 配置Kits	
Figure 3-16 Qt Creator首页	
Figure 3-17 配置界面	23
Figure 3-18 项目名称和路径	24
Figure 3-19 创建成功	24
Figure 3-20 文件	25

前言

关于本指南和技术支持信息

关于本手册

本手册主要为开发者提供赛昉科技新一代SoC平台——昉·惊鸿-7110 Qt模块的编程基础,生成可在昉·惊鸿-7110 DevKit上正常运行的RISC-V端Qt镜像。

受众

本手册主要服务于与Qt相关驱动程序的开发人员。如果您正在开发其他模块,请与您的销售或支持顾问联系,获取我 们在昉·惊鸿-7110上的完整文档。

修订历史

Table 0-1 修订历史

版本	发布说明	修订
1.0	2025/02/21	首次发布。

注释和注意事项

•

本指南中可能会出现以下注释和注意事项:

- *i* Tip: 建议如何在某个主题或步骤中应用信息。
- ・ <mark>え</mark> Note: 解释某个特例或阐释一个重要的点。
- Important: 指出与某个主题或步骤有关的重要信息。
- CAUTION: 表明某个操作或步骤可能会导致数据丢失、安全问题或性能问题。
 - Warning:

表明某个操作或步骤可能导致物理伤害或硬件损坏。

1. 介绍

Qt是一种跨平台的C++图形应用程序开发框架,它提供了丰富的工具和库,用于构建高性能、可扩展且美观的应用程序。Qt最初由挪威的Trolltech公司(现在是Qt公司的一部分)开发,并于1995年首次发布。目前Qt支持的操作系统包括Windows、嵌入式Linux、MS/Windows - 95、98、NT4.0、ME、2000、XP、Vista 等,这意味着使用Qt您只需一次性开发应用程序,无须重新编写源代码,便可跨不同桌面和嵌入式操作系统部署这些应用程序。

本手册主要为开发者提供赛昉科技新一代SoC平台——昉·惊鸿-7110 Qt模块的编程基础,生成可在昉·惊鸿-7110 DevKit上正常运行的RISC-V端Qt镜像。

防·惊鸿-7110 DevKit SDK基于buildroot框架,目前支持Qt5.15.2版本,目前支持以下Qt模块:

Table 1-1 支持的Qt模块

QT5BASE	QT5WEBSOCKETS	QT5DECLARATIVE	QT5MULTIMEDIA
QT5SVG	QT5SERIALBUS	QT5SERIALPORT	QT5QUICKCONTROLS
QT5WEBKIT	QT5WAYLAND	QT5WEBCHANNEL	QT5VIRTUALKEYBOARD
QT5CHARTS	QT5XMLPATTERNS	QT5WEBKIT_EXAMPLES	

2. 编译和运行SDK

2.1. 获取代码和编译

按照以下步骤,获取代码:

1. 执行以下命令, 下载赛昉科技昉·惊鸿-7110 DevKit的代码:

\$ git clone git clone https://glab.starfivetech.com/jh7110-devkits/devkits.git

```
2. 执行以下命令, 切换分支:
```

```
$ cd devkits
$ git check --track origin/devkits-jh7110-devel
```



建议切换到分支devkits-jh7110-devel for 5.15 kernel或devkits-6.6.y-devel for 6.6 kernel。

3. 执行以下命令, 下载其他模块:

```
$ git submodule update --init -recursive
```

4. 执行以下命令, 编译代码:

```
$ export HWBOARD_CONFIG=qt
$ make -j$(nproc)
$ make buildroot_rootfs -j$(nproc)
$ make img
```

结果:

编译完成后, 在work/目录下可找到以下两个固件:

- 。sdcard.img: TF卡烧录镜像
- 。imag.fit: U-Boot下TFTP启动镜像



建议使用sdcard.img,可参考《昉·惊鸿-7110 DevKit快速参考手册》查看详细烧录步骤。

2.2. 运行Qt应用程序

按照以下步骤,运行Qt应用程序:

- 1. 按照SDK <u>README.md</u>里*Running on JH7110 Devkits Board via Network*和*APPENDIX I: Generate Booting SD Card*的说明将imag.fit或sdcard.img在防惊鸿-7110 DevKit上运行起来后,进入串口终端或ssh终端:
 - 。账号: root
 - 。密码: starfive
- 2. 执行以下命令,运行weston桌面:

```
$ ssh root@ip
$ cd /root/ && ./run_weston.sh &
```

- \$ export XDG_RUNTIME_DIR=/root
- \$ export WAYLAND_DISPLAY=wayland-1
- \$ export QT_QPA_PLATFORM=wayland-egl

2.3. 运行Qt Demo程序

以下提供了几个运行Qt的demo:

- 模拟时钟和布局相关的demo:
 - \$ /usr/lib/qt/examples/gui/analogclock/analogclock
 - \$ /usr/lib/qt/examples/widgets/layouts/basiclayouts/basiclayouts
- •运行qv4l2的demo:



需要提前在昉·惊鸿-7110 DevKit上接入传感器或USB摄像头,详细操作请查看《昉·惊鸿-7110 Camera开发和移植手册》。

\$ /usr/bin/qv4l2

•运行qtwebkit的demo:

/usr/lib/qt/examples/webkitwidgets/browser/browser

3. 基于QT库开发UI程序

本章主要介绍了以下两个方面:

- <u>Qt Creator环境搭建 (on page 10)</u>
- <u>Qt Creator开发UI代码工程</u> (on page 16)
- <u>Qt UI代码工程编译 (on page 17)</u>

3.1. Qt Creator环境搭建

本节包含以下两个方面:

- <u>所需环境 (on page 10)</u>
- <u>安装Qt Creator</u> (on page 10)

3.1.1. 所需环境

在搭建环境前,请您做好以下准备:

- Ubuntu版本: 18.04及以上版本
- •硬件平台: 昉·惊鸿-7110 DevKit
- 软件: qt-opensource-linux-x64-5.12.9.run

3.1.2. 安装Qt Creator

请按照以下步骤,安装Qt Creator:

1. 点击进入<u>Qt官网</u>,下载Qt安装文件,此处我们使用的Qt版本是5.12.9,如下图所示:

Figure 3-1 下载Qt

16-Jun-2020 07:04	-	
16-Jun-2020 07:04	- <u>-</u>	
16-Jun-2020 18:07	3.7G	Details
16-Jun-2020 14:49	2.7G	Details
16-Jun-2020 14:48	1.3G	Details
16-Jun-2020 18:14	207	Details
	16-Jun-2020 07:04 16-Jun-2020 07:04 16-Jun-2020 18:07 16-Jun-2020 14:49 16-Jun-2020 14:48 16-Jun-2020 14:48 16-Jun-2020 18:14	16-Jun-2020 07:04 - 16-Jun-2020 07:04 - 16-Jun-2020 18:07 3.7G 16-Jun-2020 14:49 2.7G 16-Jun-2020 14:48 1.3G 16-Jun-2020 18:14 207

- 2. 下载完成后,将qt-opensource-linux-x64-5.12.9.run拷贝至虚拟机开发环境的任意目录下,并执行以下命令:
 - \$ chmod 777 qt-opensource-linux-x64-5.12.9.run
 - \$./qt-opensource-linux-x64-5.12.9.run
- 3. 执行完成后, 系统会弹出以下界面:

Figure 3-2 安装界面

	Qt 5.12.9 Setup 🛛 😣
Welcome to the	Qt 5.12.9 installer
Qt	 This installer provides you with the open source version of Qt 5.12.9. Please log in with your Qt Account credentials. If you do not have a Qt Account yet, you can create one free of charge in the next step. The Qt Account will give you access to Qt downloads, exclusive services, bug reports, code review, and forums & wiki. Looking for Squish, the automated GUI testing framework? Click here to download or try it. Network requests completed.
Settings	Next Cancel

4. 在弹窗中一直点击Next进行安装:

Figure 3-3 点击Next



5. 在如下**Settings**界面中,点击**Browse...**选择Qt creator的安装路径,选择完成后,点击**Next**进入下一步:

	ing	
		Settings 🤤
Network	Repositories	
No prox	κy	
O System	proxy settings	
O Manual	proxy configura	tion
HTTP pro	xy:	Port: 0
FTP proxy	/:	Port/ 0
		Cancel 40K

6. 在如下界面中, 勾选Developer and Designer Tools组件, 然后点击Next进入下一步:

Figure 3-5 选择组件	
Qt 5.12.9 Setu	IP 🙁
Select Components Please select the components you want to	o install.
Default Select All Deselect All ✓ Qt > Qt 5.12.9 > ✓ Developer and Designer Tools	Qt
	This component will occupy approximately 664.75 MB on your hard disk drive.
< <u>B</u> ack	k <u>N</u> ext > Cancel
点击Install开始安装:	

7. 点击Install开始安装:

Figure 3-6 开始安装



8. 安装完成后,显示如下界面,取消选项Launch Qt Creator的勾选,点击Finish,成功安装Qt Creator。

Figure 3-7 完成安装



3.2. Qt Creator开发UI代码工程

Qt界面库的一大优点就是跨平台,您可以使用Qt Creator设计出好看的UI,然后移植进目标开发板上,以下内容介绍怎 么使用Qt Creator来设计布局:

Qt Creator的布局支持使用QtWidget文件名为*.ui和QtDeclarative模块使用*.qml,这里介绍QtWidget方式。

直接点击如上的mainwindow.ui可以打开UI Design界面:

⊦ıgu	re 3-8 UI Design界I	町							
			qt_test/mainwindo	w.ui @ qt_test - Qt	Creator			00	×
File E	dit Build Debug Analyze	Tools Window I	Help						
	🗉 🍃 qt_test/mainwindow.ui	\$ X %	🐴 🔖 🛤 III 🗉 M 🗷 🗄	III 🖷 🔣					
	Filter	• Type Here		•			Filter		
Welcome	Layouts						Object	Class	Ľ
E							 MainWindow 	QMaindow	
Edit	Grid Layout			· · · · · · · · · · · · · · · · · · ·			👼 centralw	idget QWidget	
			十四 人词	きまた (44			menubar	QMenuBar	
Design	」 进入设计模式		<u> </u>	计全计			Statusbai	QStatusbai	
ŵ	🕅 Horizontal Spacer								
Nebua	📓 Vertical Spacer								
<i>c</i> .	 Buttons 								
7	Push Button								
Projects	Tool Button								
•	Radio Button								
Help	Check Box								-
							Filtor		1
	Them Views (Model-Based)						MainWindow : ON	lainWindow	
	List View						Property	Value	1
							T OObject	Value	ſ
	Table View		<i></i>	-11			objectName	MainWindow	
	🔲 Column View		Q. Tayb	Filter	Chaskable	TeelTie	• OWidget		L
	📃 Undo View	Name	Used Text	Shortcut	Checkable	Toothp	enabled	V	1
	 Item Widgetem-Based) 						geometry	[(0, 0), 800 x 600]	
· •	List Widget						sizePolicy	[Preferred, Pref	1
Debug	Tree Widget						minimumSize (0 x 0	
	Table Widget						maximumSize	16777215 x 167	
							sizeIncrement	0 x 0	
		•					baseSize	0 x 0	
>		Action Editor	Signals Slots E				palette	Inherited	
	■ R. Type to locate (1 Issues 2 Searc	h Re 3 Applicatio	4 Compile O 5	QML Deb 6	6 General M.	8 Test Results	÷ 🛋 🗆	

在这个界面上,您可以拖拽左边的控件来添加您窗体的控件,具体操作方法可以详见<u>Qt Creatot官网</u>上的丰富教程。

3.3. Qt UI代码工程编译

本节主要包含以下三个方面:

. . .. **.**

.

- <u>基于Qt Creator工具编译 (on page 17)</u>
- <u>基于Buildroot环境编译 (on page 25)</u>
- 基于昉·惊鸿-7110 DevKit板上编译 (on page 26)

3.3.1. 基于Qt Creator工具编译

Qt Creator 是一个用于Qt 开发的轻量级跨平台集成开发环境。与Qt 语言相辅相成, Qt Creator 能够跨平台运行, 目前 支持的系统包括Linux(32 位及64 位), Mac OS X 和Windows, Qt Creator 的设计使得开发人员能够利用Qt 这个应用 程序框架更加快速和轻易的开发务。

3.3.1.1. Qt Creator交叉编译器配置

本节主要介绍了以下几个内容:

- <u>配置交叉编译器 (on page 17)</u>
- 配置Debuggers (on page 20)
- <u>配置Qt Version</u> (on page 21)
- <u>配置Kits (on page 22)</u>

3.3.1.1.1. 配置交叉编译器

按照以下步骤,配置交叉编译器:

1. 执行以下命令, 启动Qt Creater, 会出现Qt开发界面:

sudo ./qtcreator

2. 在Qt开发界面中,点击Tools > Options选项,如下图所示:



3. 进入到Options界面, 按照Kits > Compilers > GCC > C++顺序选择C++, 如图所示:

Figure 3-9 选择Options



4. 下图为弹出的窗口,添加G++编译器。在Compiler path填入/opt/riscv64-buildroot-linuxgnu_sdk-buildroot/bin/riscv64-buildroot-linux-gnu-g++,并将Name修改为G++,如下图所示:



Figure 3-11 添加G++编译器

				Options	— Qt Creator			
er	Kits							
Kits	Kits Qt Versions	Compilers	Debuggers	CMake				
Environment	Name					Туре		Add
Text Editor	 Auto-detected C++ 							Clone
Falsa) Gar	GCC (C++,	x86 64bit in /	usr/bin)			GCC		Remove
Fakevim	GCC (C, x8	6 64bit in /usr/	bin)	12 0/Teols/Otc	reator/liboxoc/atcreator/cla	GCC		Remove All
Help	▼ Manual	80 04Dit III /110	ine/ioninx/qcs.	12.9/10015/QCC	reator/fibexec/qtcreator/cla	ng/bin/ clang		Re-detect
C++	← C++ G++					GCC		Auto detection Cottings
Qt Quick	₹ C GCC					GCC		Auto-detection settings
Build & Run								
Qbs								
Debugger								
Debugger Designer								
Debugger Designer Python	Name:	G++						
Debugger Designer Python Analyzer	Name: <u>C</u> ompiler path:	G++ /opt/risc	v64-buildroot-lii	nux-gnu_sdk-t	uildroot/bin/riscv64-buildroot	ot-linux-gnu-g++	Browse	
Debugger Designer Python Analyzer Version Control	Name: Compiler path: Platform codegen f	G++ /opt/risc lags:	v64-buildroot-lii	nux-gnu_sdk-t	uildroot/bin/riscv64-buildro	ot-linux-gnu-g++	Browse	
Debugger Designer Python Analyzer Version Control Devices	Name: <u>C</u> ompiler path: Platform codegen f Platform linker flag	G++ /opt/risc lags: s:	v64-buildroot-lii	nux-gnu_sdk-t	uildroot/bin/riscv64-buildro	ot-linux-gnu-g++	Browse	
Debugger Designer Python Analyzer Version Control Devices Code Parting	Name: <u>Compiler path:</u> Platform codegen f Platform linker flag <u>A</u> BI:	G++ /opt/risc lags: s: unknow	v64-buildroot-lii n-linux-generi 1	nux-gnu_sdk-t	uildroot/bin/riscv64-buildroo	ot-linux-gnu-g++ c eif	Browse	5
Debugger Designer Python Analyzer Version Control Devices Code Pasting	Name: <u>Compiler path:</u> Platform codegen f Platform linker flag <u>A</u> BI:	G++ /opt/risc lags: s: unknow	v64-buildroot-lii n-linux-generi N	nux-gnu_sdk-t	uildroot/bin/riscv64-buildroo •] - [inux •] - [geneci	ot-linux-gnu-g++ c •]-[eif	Browse	
Debugger Designer Python Analyzer Version Control Devices Code Pasting Language Client	Name: <u>Compiler path:</u> Platform codegen f Platform linker flag <u>A</u> BI:	G++ /opt/risc lags: s: unknow	v64-buildroot-lii n-linux-generi N	nux-gnu_sdk-t	uildroot/bin/riscv64-buildroo •] - [inux •] - [geneci	ot-linux-gnu-g++ c v]-feit	Browse	

5. 按照<u>第3步 (on page 18)</u>描述的同样的方法,添加GCC编译器。点击右侧Add > GCC > C,在Compiler path填入/opt/riscv64-buildroot-linux-gnu_sdk-buildroot/bin/riscv64-buildroot-linux-gnu_gcc,并将Name修改为GCC,如下图所示:

i Tip: 为避免混淆,您也可以将Name修改为其他名称,如jh7110-GCC。

Figure 3-12 添加GCC编译器

			Options — Qt Creator		8
Filter	Kits				
Kits	Kits Qt Versions Co	mpilers Debuggers	CMake		
Environment	Name			Туре	Add *
Text Editor	 Auto-detected C++ CCC (C++ x86 	64bit in (usr/bin)		666	Clone
K. FakeVim	* C	bit in (usr/bin)		ecc	Remove
😮 Help	Clang (C, x86 64	4bit in /home/forlinx/Qt5.	.12.9/Tools/QtCreator/libexec/qtcreator/clang/bin)	Clang	Remove All
{} C++	* C++			600	Re-detect
1 Qt Quick	+ C			GCC	Auto-detection Settings
> Build & Run					
Obs					
1 Debugger					
/ Designer					
Pvthon	Name:	GCC			
Analyzer	Compiler path:	/opt/riscv64-buildroot-l	inux-qnu sdk-buildroot/bin/riscv64-buildroot-linux	-gnu-gcc Browse	
Version Control	Platform codegen flags:	:			
	Platform linker flags:				
	ABI:	unknown-linux-generi	▼ unknown *]-[linux *]-[generic *]	- [elf *]- [64bit *]	
Code Pasting					
UC, lesting					
				<i>₩</i> ок	🗙 Cancel 🛛 🖌 Apply

3.3.1.1.2. 配置Debuggers

配置Debuggers

在弾出的如下窗口,点击**Debuggers**选项卡,在Path中填入/opt/riscv64-buildroot-linux-gnu_sdkbuildroot/bin/riscv64-buildroot-linux-gnu-gdb,选中后点击Open,并将**Name**修改为Debuggerfor-7110,如下图所示:

Figure 3-13 配置Debuggers

			Options — Qt Creator		
lter	Kits				
Kits	Kits Qt Version	compilers	Debuggers CMake		
Environment	Name		Location	Туре	Add
Text Editor	 Auto-detected System GDB at /usr/k System GDB at /bin/k 		/usr/bin/gdb	GDB	Clone
🕻 FakeVim	 Manual Debugger-fi 	5 at /bin/gdb	/pm/gap	on/riscv64-buildroot-linux-anu-adb GDB	Remove
Help	Debuggerin	5177110	fopensevo+ banarooennax-gna_sae banarooen	Shiphsevor Sandrotennax ghurgas Obs	
} C++					
Qt Quick					
Build & Run					
9 Qbs					
🖡 Debugger					
 Designer 					
Python					
Analyzer	Name:	Name: Debugger-for-7110			
Version Control	Path:	ath: /opt/riscv64-buildroot-linux-gnu_sdk-buildroot/bin/riscv64-buildroot-linux-gnu-gdb Browse			se
Devices	Туре:	Type: GDB			
Code Pasting	ABIs:	ABIs: unknown-linux-generic-elf-unknown			
	Version:				
Language Client	version.			Preuv	
Language Client Testing	Working directo	ry:		Brow	se

3.3.1.1.3. 配置Qt Version

配置Qt Version

点击Qt Versions选项卡,点击右侧的Add,在/opt/riscv64-buildroot-linux-gnu_sdk-buildroot/bin/的 目录下找到qmake,选中后点击Open,添加后显示下图,点击Apply。

Figure 3-14 配置Qt Version

	Options — Qt Creator	0
Filter	Kits	
🖬 Kits	Kits Qt Versions Compilers Debuggers CMake	
Lenvironment	Name Type qmake Location	Add
Text Editor	Manual ▲ Ot 5.15.2 (riscy64-huildroot-linux-onu.sdk-huildroot) /opt/riscy64-huildroot-linux-onu.sdk-huildroot/bin/omake	Remove
👫 FakeVim		Link with Ot
Help		Clean Up
{} C++		
1 Qt Quick		
≯ Build & Run		
🖽 Qbs		
🏶 Debugger		
/ Designer		
New Python		
Analyzer		5
Version Control	Version name: Qt %{Qt:Version} (riscv64-buildroot-linux-gnu_sdk-buildroot)	
Devices	gmake location: /opt/riscv64-buildroot-linux-gnu_sdk-buildroot/bin/gmake Browse, Browse,	
Code Pasting	No qmlscene installed.	
🗗 Language Client	Qt version 5.15.2 for Desktop Details	-
🛃 Testing	Register documentation: Highest Version Only 🔹	
	<i>₫</i> ок	ncel 🖌 Apply

3.3.1.1.4. 配置Kits

配置Kits

点击Kits选项卡,点击右侧Add,添加一个新的Kits,按照下图的内容进行修改,点击Apply。

Figure 3-15 配置Kits

		Options — Qt Creator	
ter	Kits		
Kits	Kits Qt Versions Compiler	5 Debuggers CMake	AUU
] Environment	Auto-detected Manual		Clone
Text Editor	Desktop (default)		Remove
FakeVim			Make Default
Help		S	ettings Filter
C++		Defau	ult Settings Filter
Qt Quick	Name:	7110	· ·
Build & Run	File system name:		
Qbs	Device type:	Desktop	•
Debugger	Device:	Local PC (default for Desktop)	 Manage
Designer	Sysroot:		Browse
Python	Compiler:	C: GCC	 Manage
Analyzer	C++: G++ *		
Version Control	Environment: No changes to apply.		
Devices	Debugger: Debugger-for-7110 *		
Code Pasting	Qt version: Qt 5.15.2 (riscv64-buildroot-linux-gnu_sdk-buildroot)		
Language Client	Additional Obs Profile Settings:		Change
Testing	CMake Tool:		Manage
	CMake consister	enonos enonos Diotformi enonos Taoleoti enonos	Change

3.3.1.2. 新建项目

按照以下步骤,新建Qt Creator项目:

1. 执行以下命令, 进入QT Creator, 启动QT Creator:

cd /home/riscvdev/Qt5.12.9/Tools/QtCreator/bin/ sudo ./qtcreator

示例输出:

Figure 3-16 Qt Creator首页

		tst1/mai	inwindow.ui @ tst1 - Qt Creator	
File Ed	dit Build Debug Analyz	e Tools Window Help		
Welcome	Projects Examples	Sessions 🌣 Manage	Projects + New C Open	
Edit Design	Tutorials	进入设计模式	/data/workspace/my_qt/QT_Examples/tst1/tst1.pro 2 ■ recipebrowser /data/opt/Qt5.12.9/Examples/Qt-5.12.9/webengine/recipebrowser/recipebrowser.pro	
û Debug	Marketplace		3 bos-studio /data/workspace/my_qt/my_obs_studio/obs-studio/CMakeLists.txt	
بر			4 v25_video_tool /data/workspace/my_qt/video_player/cv25_video_tool/cv25_video_tool.pro	
Projects	New to Qt?		s 🖿 camera	
? Help	Learn how to develop your own applications and explore Qt Creator.		/data/opt/Qt5.12.9/Examples/Qt-5.12.9/multimediawidgets/camera/camera.pro c ■ qt_2019_ncov /data/workspace/my_qt/qt_2019_ncov/qt_2019_ncov.pro	v
	Get Started Now		7 analogclock /data/opt/Qt5.12.9/Examples/Qt-5.12.9/gui/analogclock/analogclock.pro	
tst1				
,	L Qt Account			
Debug	Online Community			
		行		
	② User Guide			
>				
	■ 🔑 Type to locate (1 Issues 2 Search R 3 A	Applicat 4 Compile 5 QML De 6 General 8 Test Res ÷	= 0 //

2. 点击New后,进入如下界面选择Qt Widgets Application,然后选择Choose,配置工程选项:

Figure 3-17 配置界面



3. 输入您的项目名称,选择项目的保存路径,并点击Next:

Figure 3-18 项目名称和	路径		
	Qt Wid	gets Application — Qt Creator 🛛 😵	
Location	Project	Location	
Build System Details Translation Kits	This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.		
Summary			
	Name:	qt_test	
	Create in:	/data/workspace/my_qt/QT_Examples Browse	
	Use as	default project location	
		<u>N</u> ext > Cancel	

4. 之后一直点击Next即可,项目创建成功后会出现如下界面:

Figure 3-19 创建成功



3.3.1.3. 项目源码结构介绍

在默认新建的项目中, 会包含以下这4类文件:



以下是对这4类文件的介绍:

- •.pro项目文件是qmake编译所需要的,具体编写方法见下文。
- •.cpp文件和.h文件是c++的资源文件。
- •.ui文件是设计师设计界面文件,使用xml来编写。

3.3.2. 基于Buildroot环境编译

本节介绍在Buildroot中添加自己编写的Qt,下面介绍添加进Buildroot的方法,可以参考已有的Qt package: buildroot/package/starfive/qt_test:

0

1. 在buildroot/package/starfive目录下,新建package文件夹,在文件夹中添加两个文件:



2. 准备Qt源码。在创建Buildroot package文件夹后,需要把前面在QT Creator里创建的<u>源码 (on page 24)</u>添加 到SDK中去编译,并将整个源码复制到buildroot/package/starfive/xxx/src/目录下。

Note:

将*.pro文件里的install目录改为其他目录,如target.path = /root/bin。

3. 编译。在 SDK 顶层目录,执行以下命令:

• For image.fit

```
$ make -C ./work/buildroot_initramfs/ O=./work/buildroot_initramfs qt_test-dirclean
$ make -C ./work/buildroot_initramfs/ O=./work/buildroot_initramfs qt_test-rebuild
```

• For sdcard.img

```
$ make -C ./work/buildroot_rootfs/ O=./work/buildroot_rootfs qt_test-dirclean
$ make -C ./work/buildroot_rootfs/ O=./work/buildroot_rootfs qt_test-rebuild
```

3.3.2.1. Buildroot Qt相关编译选项说明

Table 3-1 编译选项说明

编译选项	说明
BR2_PACKAGE_QT5BASE	是否支持QT,添加qtbase(必选)
BR2_PACKAGE_QT5CHARTS	是否支持数据可视化和图表的功能
BR2_PACKAGE_QT5DECLARATIVE	是否支持QML图形设计方式
BR2_PACKAGE_QT5MULTIMEDIA	是否添加多媒体类
BR2_PACKAGE_QT5QUICKCONTROLS	是否引入Qt Quick Controls服务组件
BR2_PACKAGE_QT5SERIALBUS	是否提供串行总线通信的功能,如CAN、Modbus
BR2_PACKAGE_QT5SERIALPORT	是否提供串口通信的功能
BR2_PACKAGE_QT5SVG	是否支持位图格式图片格式
BR2_PACKAGE_QT5VIRTUALKEYBOARD	是否支持虚拟键盘
BR2_PACKAGE_QT5WAYLAND	是否添加wayland显示服务组件
BR2_PACKAGE_QT5WEBCHANNEL	可用于在Web应用程序和Qt应用程序之间进行通信和交互
BR2_PACKAGE_QT5WEBKIT	可提供基于WebKit引擎的Qt模块,可以用于浏览web 页面
BR2_PACKAGE_QT5WEBKIT_EXAMPLES	可提供 Qtwebkit的一些example,如browser
BR2_PACKAGE_QT5WEBSOCKETS	可用于用于在应用程序中实现WebSocket协议的通信功能
BR2_PACKAGE_QT5XMLPATTERNS	基于XQuery和XPath语言,用于在Qt应用程序中进行XML数据 处理和查询

3.3.3. 基于昉·惊鸿-7110 DevKit板上编译

3.3.3.1. Debian OS下编译

按照以下步骤,在Debian OS下编译Qt:

Note: Debian OS下root目录切换命令,请使用su root。

1. 执行以下命令, 安装Perl:

sudo apt install perl gcc g++ make # 更新到最新的即可

2. 执行以下命令,安装python:

sudo apt install python2 # 更新到最新的即可

Tip:

编译过程中有可能会报错,如未知的成员变量,只需要在当前报错文件上引入头文件#include <limits>即可;如果报python相关的错误,请执行cp /usr/bin/python3 /usr/bin/python。

- 3. 在命令行下先切换到工程目录下,编译Qt。
 - 。在命令行下,执行以下命令,生成的qmake文件:

qmake

。执行以下命令, 生成相关二进制可执行文件:

make

4. QT+TP触摸配置

目前SDK里默认已经支持触摸功能,只需要连接带触摸功能的屏幕,步骤同上面即可实现触摸功能,触摸功能准确无偏差。