

# EVM-RK3562 QT用户手册

## 产品名称: EVM-RK3562 评估板

- 核心板: Core-RK3562 V1.0
- 修订人员: 杨伟锋

修订历史:

版本	日期	原因	修订者
V1.00	2024/08/13	初始版本	杨伟锋

Guangzhou TaloWe Electronics Technology Co., Ltd http://www.talowe.com/

## 目录

1. QT 编译环境的搭建	3
1.1. 安装 QT Creator	3
1.2. Qt creator 环境搭建	3
1.3. QT 工程创建	6
2. QT 自启动程序更换	10
3.免责声明	11

## 1. QT 编译环境的搭建

## 1.1. 安装 QT Creator

如需使用自己开发的 QT 程序,我们需要通过 QT Creator 来进行开发。因此需要在 RK3562 的开发环境下安装 QT Creator,即为在 Ubuntu22.04 中安装 QT Creator。

安装 QT Creator 的方式可以通过 QT 官网进行下载安装,并移到虚拟机开发环境中进行安装,下载地址为: <u>Download Qt OSS: Get Qt Online Installer</u>, 下载版本为 Linux X64 版本的 QT Creator,并在自己的开发环境下进行安装。

### 1.2. Qt creator 环境搭建

安装完成后,启动 QT creator 程序,此处需要进行 QT 编译环境的搭建。 启动 QT creator 程序,点击 Edit->Preferences 选项。



图 15 打开 QT Creator

进入 Perferences 界面,点击左侧 Kits,而后点击中部上方的"compliers"选项,进入到如下界面后,点击右侧的"Add---->GCC---->C",如下图所示。

按照下图所示依次选择添加对应的 GCC 编译器,选择选项7时,在路径: [SDK]/buildroot/output/rockchip\_rk3562/host/bin 中找到 arch64-buildroot-linux-gnu-gcc 并进行添加。

	Preferences — Qt Creator			×
ilter 1	Kits 2 Kits Qt Versions Compilers Debuggers	3	3	
Devices Environment Text Editor FakeVim Help C++ Qt Qt Quick Build & Run	Name         Type <ul></ul>	4	Add ICC MinGW CC Clang Custon QCC Emscripten	C C C+++
▲ CMake	选择5后弹出下列推图 7、 Name: CCC(RK3562) ← 6、此名字可进行更改 Compiler path: me/y/linux/buildroot/output/rockchip_rk3562/host/bin/aarch64-buildroot-linux-gnu-gc Platform linker flags: Platform linker flags: ABI: arm-linux-generic-elf * arm * - linux * - generic * - elf * Target triple: Override for code model	选择编译器路径 cc Browse*		
		Apply	Ostancel €	⊘ <u>о</u> к

#### 图 16 GCC 编译器添加

采用同样的方法进行 G++编译器的添加, 在[SDK]/buildroot/output/rockchip\_ rk3562/host/bin 中找到 arch64-buildroot-linux-gnu-g++进行添加。

Kits 2   Kits Qt Versions   Denvices Name   Text Editor * Auto-detected   Text Editor * CCC (x86 64bit in /usr/bin)   FakeVim CCC (x86 64bit in /usr/bin)   Help CCC (x86 64bit in /usr/bin)   Canalyzer   Patform Control   Code Pasting   Language Client   Terminal   Terminal   Terminal   Soft catenories		Preferences — Qt Creator		×
Devices       Name       * Type       Add         Environment       * Auto-detected       * G       GCC (c, x86 64bit in /usr/bin)       GCC       ICC       MinGW         FakeVim       * Ccc (c, x86 64bit in /usr/bin)       GCC       Ccc (c, x86 64bit in /usr/bin)       GCC       Clang       Ccang       Ccc (c, x86 64bit in /usr/bin)       GCC       Ccang       Ccang       Ccc (c, x86 64bit in /usr/bin)       GCC       Ccang       Ccc (c, x86 64bit in /usr/bin)       GCC       Clang       Ccang       Ccc (c, x86 64bit in /usr/bin)       GCC       Ccang       Ccang       Ccc (c, x86 64bit in /usr/bin)       GCC       Clang       Ccang       Ccc (c, x86 64bit in /usr/bin)       GCC       Clang       Ccc (c + in /usr/bin)       GCC       Clang       Custom       GCC       Clang       Custom       GCC       Custom       GCC       Custom       GCC       Custom       GCC       GCC <t< th=""><th>Kits Qt Vers</th><th>2 ions Compilers Debuggers</th><th>3</th><th></th></t<>	Kits Qt Vers	2 ions Compilers Debuggers	3	
	Devices     Name       Environment     ~ Auto-detect       Text Editor     ~ C ( CCC ( CCC ( CHC ( CCC ( CCC ( ChC ( CCC (	Type       ted     Type       C, x86 64bit in /usr/bin)     GCC       C++, x86 64bit in /usr/bin)     GCC       (C, x86 64bit in /home/y/Qt5.12.8/Tools/QtCreator/libexec/qtcreator/clang/bin)     Clang       (C, x86 64bit in /home/y/Qt5.12.8/Tools/QtCreator/libexec/qtcreator/clang/bin)     Clang       (C, x86 64bit in /home/y/Qt5.12.8/Tools/QtCreator/libexec/qtcreator/clang/bin)     Clang       (K3562)     GCC       (K3562)     GCC       6     Ferrification       7     Ferrification       6     Ferrification       7     Ferrification       6     Ferrification       6     Ferrification       7     Ferrification       7     Ferrification	Add ICC MinGW CCang Custom OCC Emscripten	• • • • • • •

#### 图 17 G++编译器添加

点击 QT Versions 选项卡,点击 Add,进行 qmake 的添加。

在[SDK]/buildroot/output/rockchip\_rk3562/host/bin 中找到 qmake,并将其重 命名为 RK3562, 然后点击 Apply。

		Preferences -	- Qt Creator		×
Filter	Kits Qt Versions	Compilers Debuggers	点击后会弹出路	路径选择 选择对应的qma	ake文件 3
<ul> <li>Rts</li> <li>Devices</li> <li>Environment</li> <li>Text Editor</li> <li>FakeVim</li> <li>Help</li> <li>C++</li> <li>Qt Quick</li> <li>Build &amp; Run</li> <li>CMake</li> <li>Qbs</li> </ul>	Name Name Auto-detected Manual Qt 5.15.10 (RK	<pre>complets Debuggers</pre>	chip_rk3562/host/bin/qmake		Add Remove Link with Qt Clean Up
<ul> <li>✤ Debugger</li> <li>✓ Designer</li> <li>◆ Python</li> <li>E Analyzer</li> <li>℮ Version Control</li> <li>ি Code Pasting</li> <li>♂ Language Client</li> <li>▷ Terminal</li> <li>♥ Sort categories</li> </ul>	Name: Qt %{ qmake path /home Qt version 5.15.10 f Register documentat	4 此处可更改为名字 Qt:Version)( <u>(RK3562)</u> ///linux/buildroot/output/rockchip_rk3562/ho or Desktop ion: Highest Version Only *	st/bin/qmake	talis ₹	

图 18 qmake 添加

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。

		Preferences — Qt Creator				
Filter 1	Kits 2 Kits Qt Versions C	ompilers Debuggers	3	3		
Lo Devices	Name		- A	dd		
Environment	<ul> <li>Auto-detected</li> <li>Desktop Qt 6.5</li> </ul>	× Auto-detected     □      □				
Text Editor	A WebAssembly	Qt 6.5.3 (multi-threaded) Ot 6.5.3 (single-threaded)	Ren	nove		
K FakeVim	<ul> <li>Manual</li> <li>PK3562</li> </ul>		Make I	Default		
🕑 Help	I INSSE		Setting	s Filter		
{} C++		4	Default Set	tings Filter		
A Qt Quick	Name:	RK3562		Ψ.		
Build & Run	File system name:					
CMake	Run device type:	Remote Linux Device 5				
🖽 Qbs	Run device:			Manage		
Debugger	Build device:	Desktop (default for Desktop)		Manage		
Designer     Python	Compiler:	C: GCC(RK3562) 6 C++: G++(RK3562)	•	Manage		
Analyzer	Environment:	No changes to apply.		Change		
	Debugger:	System GDB at /usr/bin/gdb		Manage		
Code Pasting	Sysroot:	/		Browse		
	Qt version:	Qt 5.15.10 (RK3562)		Manage		
- Terminal	Qt mkspec:					
	- Additional Qbs Profile Se	ettings:		Change		
Sort categories	CMake Tool:	System CMake at /usr/bin/cmake	*	Manage		

图 19 Kits 添加

到此,QT编译环境已配置完成,可以开始愉快的使用QT了。

## 1.3. QT 工程创建

启动 Qt creator 程序,进入到 qtcreator 界面,点击 files-->New Project 创建新的工程。



图 20 创建新工程

#### 选择"Application"->"Qt Widgets Application", 然后点击右下角的"Choose":



#### 图 21 选择工程类型

在如下界面中为新建的工程修改名字,并选择安装路径选择(此路径由用户 自己决定),然后点击"Next":

	Qt Widgets Application — Qt Creator	×
Location Build System Details Translation Kits Summary	Qt widgets Application — Qt Creator <b>Project Location</b> This wizard generates a Qt Widgets Application project. The application der default from QApplication and includes an empty widget. 修改工程路径以及工程名	ives by
	Name: Test_project Create in: /home/y/TestApp Use as default project location	Browse*
	Next >	Cancel
	图 22 修改工程路径以及工程名	
选择 qmak	e并继续进行下一步。	
	Qt Widgets Application — Qt Creator	>
Location Build System Details Translation	Define Build System Build system: qmake 选择qmake并继续下一步	×

		Qt Widgets Application — Qt Cre	ator
Location	Define Bu	ild System	
<ul> <li>Build System</li> <li>Details</li> <li>Translation</li> <li>Kits</li> <li>Summary</li> </ul>	Build system:	qmake 选择qmake并继续下一步	•
			< Back Next > Cancel
		图 02 选择 amalia	

在如下界面中,修改类名"Class name"为"TsetAPP",将基类"Base cla ss"选择为 MainWindow, 然后点击"next":

Location	Class Info	ormation	
Build System		Specify basic information about the classes for which you want to generate skeleton source code files.	10.1
Translation	Class name:	TestApp	
Kits Summary	Base class:	QMainWindow	•
	Header file:	testapp.h	
	Source file:	testapp.cpp	
		Generate form	
	Form file:	testapp.ui	
		7	

#### 图 24 基类 MainWindow

之后一直 next,当出现如下界面时,选择之前已添加过的"RK3562"作为 当前工程的 kit, 然后点击"next":

		et magets Application Qt creator	
Location Build System	Kit Selecti	<b>ON</b> kits can be used for project <b>Test_project</b> :	
Details	🔲 Select all ki	Type to filter kits by name	
Translation Kits Summary	🗆 🖵 Deskt	op Qt 6.5.3 GCC 64bit	Details 👻
-	🔶 🗹 🔺 🗡	52	Details *
-	✓ Debug	/home/y/TestApp/Test_project/build/RK3562-Debug	Browse
仅选择此选项	✓ Release	/home/y/TestApp/Test_project/build/RK3562-Release	Browse
	✓ Profile	/home/y/TestApp/Test_project/build/RK3562-Profile	Browse
	🗌 🔺 WebA	ssembly Qt 6.5.3 (multi-threaded)	Details 👻
	🗆 🔺 WebA	ssembly Qt 6.5.3 (single-threaded)	Details 👻
		< Back Ne	xt > Canc

最后点击 Finish 完成工程的创建。

$\sim$	
~	

Loc Bui Del	ation ild System	Project Management			
Bui Del	ild System				
	tails	Add as a subproject to project:	<none></none>		•
Tra Kits	nslation s	Add to version control:	<none></none>	•	Configure
Sur	nmary				
		Files to be added in			
		/home/y/TestApp/Test_pro	oject:		
		Test_project.pro			
		main.cpp		1	
		testapp.cpp			
		testapp.ui			
				< <u>B</u> ack <u>F</u> inish	Cancel

图 25 工程创建成功

## 2. QT 自启动程序更换

在 RK3562 烧录固件中,具有本公司所开发的 QT 程序及瑞芯微官方 QT 作为开机启动界面,若用户需进行开机 QT 程序的更换,可查看本小节进行 QT 程序的更换。

以 TestApp 为例,将开发完成的 QT 程序 TestApp 移到开发板之中,存放于 /opt 目录下。

root@rk3562-buildroot:/opt# ls 4g.sh TestApp config.ini myMusic myVideo sound.sh taloweApp unixbench

增加可执行权限

chmod +x TestApp

更改 S49weston 文件,将其中的 TaloweApp 进行注释,并添加自己所开发的 QT 程序。

vi /etc/init.d/S49weston



图 26 更改启动文件

注意:添加过程一定不能忘记后面的 & , 否则会导致系统奔溃, 需重新进行固件的烧录。

添加成功后重启开发板便能成功更换 QT 应用。

## 3. 免责声明

本文档提供有关广州眺望电子科技有限公司产品的信息。本文档并未授予任何知识产权的许可,并未以明示或暗示,或以禁止发言或其它方式授予任何知识 产权许可。除广州眺望电子科技在其产品的销售条款和条件中声明的责任之外, 概不承担任何其它责任。并且,产品的销售和/或使用不作任何明示或暗示的担保,包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产 权的侵权责任等,均不作担保。广州眺望电子科技产品并非设计用于医疗、救生 或维生等用途。广州眺望电子科技可能随时对产品规格及产品描述做出修改,恕 不另行通知。

文档所属产品可能包含某些设计缺陷或错误,一经发现将收入勘误表,并因此可能导致产品与已出版的规格有所差异。如客户索取,可提供最新的勘误表。在订购产品之前,请您与我司销售处或分销商联系,以获取最新的规格说明。本文档中提及的含有订购号的文档以及其它文献可通过访问<u>https://www.talowe.com/</u>获得。

广州眺望电子科技有限公司保留所有权利。