



# PET\_A40I-H\_P01\_Linux 应用编程手册



### 一、虚拟磁盘文件说明

虚拟机磁盘文件已经配置好 QT 应用开发环境和交叉编译器,可以直接运行 qtcreator 软件进行应用开发,无需再执行下面 2~5 步骤。

在开发工具目录下 GZPEITE\_Ubuntu\_22.04.7z ,将其解压出 GZPEITE\_Ubuntu\_22.04.vmdk 文件。 在 PC 端安装 VMware 15.5.6 及以上版本

打开 VMware 软件,创建一个虚拟机,然后设置虚拟磁盘文件为 PET\_RK3288\_Ubuntu18.04\_64.vmdk,硬盘 类型为 SCSI。

## 二、环境准备

在 PC 机端安装 Ubuntu 22.04 64 位系统

## 三、复制文件到 PC 机 Ubuntu 系统并解压

将开发工具目录下的 PET\_A40I\_APP\_DEV 目录及文件复制到 Ubuntu 系统内





# 四、无图形界面应用程序开发

交叉编译器路径: /usr/local/PET\_A40I\_APP\_DEV/out/sun8iw11p1/linux/common/buildroot/host/opt/ext-toolchain/bin/

sysroot 路径: /usr/local/PET\_A40I\_APP\_DEV /out/sun8iw11p1/linux/common/buildroot/host/usr/arm-buildroot-linux-gnueabihf/sysroot

export PATH=/usr/local/PET\_A40I\_APP\_DEV/out/sun8iw11p1/linux/common/buildroot/host/opt/ext-toolchain/bin/:\$PATH

在编译源码时使用 --sysroot=/usr/local/PET\_A40I\_APP\_DEV /out/sun8iw11p1/linux/common/buildroot/host/usr/arm-buildroot-linux-gnueabihf/sysroot

F	gzpeite@gzpeite: ~/PE	T_A40I_APP_DEV	Q =		
gzpeite@gzpeite: ~/PET_A40I_APP_DEV		gzpeite@gzpeite: ~/PET_A40I_APF	_DEV		× ~
<pre>gzpeite@gzpeite:~/PET_A40I_APP_DEV\$ export PATH=/us in/bin/:\$PATH gzpeite@gzpeite:~/PET_A40I_APP_DEV\$ arm-linux-gnuea Using built-in specs. COLLECT_GCC=arm-linux-gnueabihf-gcc COLLECT_LTO_WRAPPER=/usr/local/PET_A40I_APP_DEV/out nux-gnueabihf/5.3.1/lto-wrapper Target: arm-linux-gnueabihf Configured with: /home/tcwg-buildslave/workspace/tc /gcc-linaro-5.3-2016.05/configure SHELL=/bin/bash - 64-tcwg/target/arm-linux-gnueabihf/_build/builds/de e-release/label/docker-trusty-amd64-tcwg/target/arm wg-buildslave/workspace/tcwg-make-release/label/doc wn-linux-gnuwith-gnu-aswith-gnu-lddisable- sable-nlsenable-c99with-tune=cortex-a9with ltilibenable-multiarchwith-build-sysroot=/hom /arm-linux-gnueabihf/_build/sysroots/arm-linux-gnueabihf strapbuild=x86_64-unknown-linux-gnuhost=x86_6 kspace/tcwg-make-release/label/docker-trusty-amd64- Thread model: posix gcc version 5.3.1 20160412 (Linaro GCC 5.3-2016.05) gzpeite@gzpeite:~/PET_A40I_APP_DEV\$</pre>	r/local/PET_A40I_AP bihf-gcc -v /sun8iw11p1/linux/c wg-make-release/lab -with-mpc=/home/tcw stdir/x86_64-unknow -linux-gnueabihf/_b ker-trusty-amd64-tc libstdcxx-pch -dis bker-trusty-amd64-tc libstdcxx-pch -dis -arch=armv7-a -wit e/tcwg-buildslave/w eabihfenable-lto (-e-release/label/doc /libcenable-lang 4-unknown-linux-gnu tcwg/target/arm-lin	P_DEV/out/sun8iw11p1/linux/common/build ommon/buildroot/host/opt/ext-toolchain g-buildslave/workspace/tcwg-make-releas n-linux-gnuwith-mpfr=/home/tcwg-bui uild/builds/destdir/x86_64-unknown-lin wg/target/arm-linux-gnueabihf/_build/b able-libmudflapwith-float=hardwit orkspace/tcwg-make-release/label/docke enable-linker-build-idenable-long ker-trusty-amd64-tcwg/target/arm-linux uages=c,c++,fortran,ltoenable-check target=arm-linux-gnueabihfprefix: ux-gnueabihf/_build/builds/destdir/x86_	droot/host/op /bin//libex se/label/dock ldslave/works ux-gnuwith uilds/destdir -ppl=nowit th-mode=thumb r-trusty-amd6 -longenabl -gnueabihf/_b ing=release - =/home/tcwg-b _64-unknown-l	t/ext-1 ec/gcc/ er-trus pace/to -gmp=/h /x86_64 h-isl=r disa 4-tcwg/ e-share uild/bu -disabl uildsla inux-gr	toolcha /arm-li apshots sty-amd cwg-mak home/tc 4-unkno nodi able-mu /target edwi uilds/d le-boot ave/wor nu



## 五、QT 应用程序开发

## 1、安装并设置 QT-Creator

安装	sudo apt -y install	qtcreator
----	---------------------	-----------

启动 qtcerator

启动 qtcreator 后,选择菜单 Tools->Options->Kits->Qt Versions, 点击 "ADD", 选择

/usr/local/PET\_A40I\_APP\_DEV/buildroot-201611/dl/qt-everywhere-opensource-src-5.9.0/Qt-5.9.0/bin/qmake 再点击右下角 Apply 按钮

	Options — Qt Creator	×
Filter	Kits	
🖬 Kits 🗖	Kits         Qt Versions         Compilers         Debuggers         CMake	
🖵 Environment	Name 🔻 qmake Path	
Text Editor	Auto-detected	
K. FakeVim	Qt 5.9.0 (Qt-5.9.0) /usr/local/PET_A40I_APP_DEV/buildroot-201611/dl/qt-everywhere-opensource-src-5.9.0/Qt-5.9.0/bin/	qn
Help		_
{} C++	Name: Qt %{Qt:Version} (Qt-5.9.0)	
🦪 Qt Quick	qmake path: /usr/local/PET_A40I_APP_DEV/buildroot-201611/dl/qt-everywhere-opensource-src-5.9.0/Qt-5.9.0/bin/qmake	го١
➤ Build & Run	Qt version 5.9.0 for Desktop	De
🖽 Qbs	Register documentation: Highest Version Only	
🕸 Debugger 🚽		Þ
	Apply Scancel ScA	

选择 Kits->Complies 选项页,点击 Add->GCC->C,文件路径如下,

/usr/local/PET\_A40I\_APP\_DEV/out/sun8iw11p1/linux/common/buildroot/host/opt/ext-toolchain/bin/arm-linuxgnueabihf-gcc, 然后点击右下角 Apply 按钮

点击 Add->GCC->C++, 文件路径选择:

/usr/local/PET\_A40I\_APP\_DEV/out/sun8iw11p1/linux/common/buildroot/host/opt/ext-toolchain/bin/arm-linuxgnueabihf-g++,点击右下角 Apply 按钮,设置结果如下图:

		Options -	— Qt Crea	ator		0
Filter	Kits					
🖬 Kits 📍	Kits Qt Versions Comp	oilers Debuggers	CMake			
Environment	Name		Т	уре		Add 👻
Text Editor	<ul> <li>Auto-detected</li> <li>C</li> </ul>					Clone
K FakeVim	GCC (C, x86 64bit a	it /usr/bin/gcc) bit at /usr/bin/clang-	14) (	iCC lang		Remove
🕑 Help	<ul> <li>C++</li> <li>CCC (C++ x86 64bit)</li> </ul>	t at /usr/bip/a++)	, .			Remove All
{} C++	Clang 14 (C++, x86	64bit at /usr/bin/clar	ng++-14) C	lang		Re-detect
🕈 Qt Quick				Auto-detection Settings		
Build & Run	× C++					
Qbs	A40I-G++					
🗰 Debugger						
✔ Designer	Name:	A40I-GCC				
Python	Compiler path:	inux/common/buildro	ot/host/o	pt/ext-toolchain/bin/arm-linux-o	nueabit	
Analyzer	Platform codegen flags:			, , , , , , , , , , , , , , , , , , ,		
Version Control	Platform linker flags:					
Devices	ABI:	arm-linu) 👻 arm	- lin	Jx 👻 - generic 👻 - elf		
Code Pasting				,,,		
P Language Client -					۲.	
					Apply	/ ⊗ <u>C</u> ancel ⊘ <u>O</u> K



#### PET\_A40I-H P01 开发板

选择菜单 Tools->Options->Kits->Kits,点击"ADD"按下图进行相关设置,设置好以后点击 Make Default 按钮和右下角 Apply 按钮,最后点击右下角 OK 按钮,关闭设置页。 名称为 GZPEITE Device type 选择 Generic Linux Device

sysroot 路径选择

/usr/local/PET\_A40I\_APP\_DEV/out/sun8iw11p1/linux/common/buildroot/host/usr/arm-buildroot-linux-gnueabih f/sysroot

C和 C++编译器选择上一步添加的 A40I-GCC 和 A40I-G++

QT version 选择 QT5.9.0

		Options — Qt Creator	×
Filter	Kits		
🖼 Kits	Kits Qt Versions Compi	lers Debuggers CMake	
Environment	Name		Add
Text Editor	Auto-detected Manual		Clone
K FakeVim	Desktop     GZPEITE (default)		Remove
Help		N	1ake Default
{} C++		Se	ttings Filter
🦪 Qt Quick		Defau	lt Settings Filter
➤ Build & Run			
🖽 Qbs			
Debugger	-11 (	GZPEITE	
✓ Designer	File system name:		
Python	Device type:		·
Analyzer	Device:		<ul> <li>Manage</li> </ul>
Version Control	Build device:	Local PC (default for Desktop)	Manage
Devices	Sysroot:	(1) p //inux/common/buildroot/nost/usr/arm-buildroot-linux-gnueabinr/sysr	oot Browse
Code Pasting	Compiler:		Manage
-	Fauireameatu		Chapter
A Testing	Environment:	No changes to apply.	Change
	Ot version:		<ul> <li>Manage</li> </ul>
	Qt version.	Qr 5.9.0 (Qr 5.9.0)	· Manage
	Additional Obs Brofile Settin	2051	Chapte
	CMake Tool:		Mapage
	CMake generator	<pre><none> - <none> Platform: <none> Toolset: <none></none></none></none></none></pre>	Change
	CMake Configuration	CMAKE CXX COMPILEDISTRING-%/CompilerExecutable:CvvV: CMAKE C C	Change
			S Change
		Apply 😣	ancel OK

## 2、创建项目

选择菜单 File->New File or Project,弹出页面点击 choose 按钮



		New Project — Qt Creator	×
Choose a template:			All Templates 👻
Projects Application (Qt) Application (Qt for Py Library Other Project Non-Qt Project Import Project Files and Classes	thon)	Qt Widgets Application         Qt Console Application         Qt Quick Application	Creates a widget-based Qt application that contains a Qt Designer-based main window. Preselects a desktop Qt for building the application if available. <b>Supported Platforms</b> :
			Scancel Schoose
Location Build System Details Translation Kits Summary	Project I This wizard default from Name: Create in:	Location generates a Qt Widgets Application proje n QApplication and includes an empty wid test	ct. The application derives by lget.
	Use as o	default project location	
			<u>N</u> ext > Cancel



		Qt Widgets Application — Qt Creator			×
Location	Define B	uild System			
Build System Details Translation Kits Summary	Build system	: gmake			•
		Qt Widgets Application — Qt Creator	< <u>B</u> ack	<u>N</u> ext >	Cancel
Location Build System Details	Class Info	<b>ormation</b> Specify basic information about the class skeleton source code files.	es for which	you want to	generate
Translation	Class name:	MainWindow			
Kits Summary	Base class:	QMainWindow			¥
	Header file:	mainwindow.h			
	Source file:	mainwindow.cpp ✔ Generate form			
	Form file:	mainwindow.ui			
			< <u>B</u> ack	<u>N</u> ext >	Cancel



	Qt Widgets Application — Qt Creator	×
Location Build System Details Translation Kits Summary	Qt Widgets Application — Qt Creator         Translation File         If you plan to provide translations for your project's user interface via the Qt tool, please select a language here. A corresponding translation (.ts) file will generated for you.         Language: <none>         Translation file:       <none></none></none>	Linguist be
Location Build System Details	< <u>Back</u> <u>Next</u> > Qt Widgets Application — Qt Creator Kit Selection The following kits can be used for project test: Type to filter kits by name	Cancel
Translation Kits Summary	Select all kits	Details • Details •
	< <u>B</u> ack <u>N</u> ext >	Cancel



	Qt Widgets Appl	lication — Qt Creato	г	×
Location	Project Management			
Build System	Add as a subproject to project:	<none></none>		~
Details Translation Kits	Add to <u>v</u> ersion control:	<none></none>		▼ Configure
Summary				
	Files to be added in			
	<pre>/home/gzpeite/test:</pre>			
	main.cpp mainwindow.cpp mainwindow.h mainwindow.ui test.pro			
			< <u>B</u> ack <u>F</u>	inish Cancel

## 3、编译项目

选择菜单 Build->Build Project "test",编译完成后会生产 test 可以执行文件。 将设备烧写成 Linux 系统,然后将 test 文件复制到设备内,即可运行查看效果。

# 六、与主板传输文件

主板移植有 adb 服务,在 PC 机电脑端可以通过 adb 命令行与主板进行文件上传与下载操作,例如:

上传 test\_app 文件到主板/root 目录内: adb push test\_app /root 设置文件可运行属性: adb shell chmod +x /root/test\_app 从主板下载文件: adb pull /root/test\_app

# 七、设置应用开机自动运行

1、无图形界面应用程序可以使用不含 QT 的系统,修改/etc/init.d/rcS 文件,将需要运行的应用程序添加到 文件最后

/root/test\_app

2、有图形界面应用程序需要使用带 QT 的系统,修改/etc/init.d/runqt,注释掉原系统默认开机启动程序,添加新的需要启动应用。

QT 环境变量设置保存在根目录 qt5.9.sh 文件内,可根据需要进行修改(例如鼠标输入设备等)。



#!/bin/sh
# killall Launcher MediaUI CameraUI
. /qt5.9.sh
<pre>echo 0,0 &gt; /sys/class/disp/disp/attr/fb2ui</pre>
<pre># FRAMEBUFFER=/dev/fb0 /Launcher &amp;</pre>
<pre># FRAMEBUFFER=/dev/fb1 /MediaUI &amp;</pre>
<pre># FRAMEBUFFER=/dev/fb2 /CameraUI &amp;</pre>
<pre># echo Launcher, &gt; /sys/inputFocus/write</pre>
/root/test_app &

# 八、联系方式

总公司 : 广州佩特电子科技有限公司

总公司地址: 广州市天河区大观中路新塘大街鑫盛工业园 A1 栋 201 总公司网站: <u>http://www.gzpeite.net</u> SMT 子公司: 广州佩特精密电子科技有限公司(全资子公司) 子公司地址: 广州市白云区人和镇大巷村顺景路 11 号 SMT 网站 : <u>http://www.gzptjm.com</u> 官方淘宝店: <u>https://shop149045251.taobao.com</u>

#### 微信扫描二维码联系支持人员:



广州佩特电子科技有限公司

2021年5月