TW-STM32MP157-EVM 评估板 用户手册

版本: V1.0.1

修订历史

版本	日期	原因	修订者
V1.00	2021/06/21	创建文档	
V1.01	2022/03/07		

目 录

1.	产品介绍	4
	1.1 产品简介	4
	1.2 硬件资源介绍:	4
	1.3 软件资源介绍	5
	1.4 产品布局介绍	6
2.	开发板准备事项	8
	2.1 上电前需要注意的事项	8
	2.2 拨码开关设置及登录开发板	8
	2.3 安装 SecureCRT 软件	8
	2.3.1 安装 SecureCRT	8
	2.3.2 使用 SecureCRT 登录	8
3.	ST 官方软件安装	11
	3.1 Java 环境安装	11
	3.2 STM32CubeProgrammer 下载	12
	3.2.1 STM32CubeProgrammer 的简介	12
	3.2.2 STM32CubeProgrammer 软件下载	13
	3.3 STM32CubeIDE 下载	18
	3.3.1 STM32CubeIDE 的简介	18
	3.3.2 STM32CubeIDE 软件ト载	18
4.	烧与和更新固件	23
	4.1 週刊 STM32CubeProgrammer 烧与固件	23
_	4.2 通过 SD 卡进行烧与固件	25
5.	切能测试	
	5.1 LED 测试	28
	5.2 蜂鸣	28
	5.3 申口测试	28
	5.4 WIFI 测风	31
	5.5 40 例 风	32
	5.61 杏毛玄绘时轴.	54
	5.6.7 宣有永沈时行:	34
	5.6.2 登省 RTC 时钟·	34
	5.6.4 同步系统时钟·	35
	5.7 CAN 测试	35
	5.8 摄像头测试	
	5.9 网络测试	36
	5.10 音频测试	38
	5.11 TF 卡测试	38
	5.12 U 盘使用	39
	5.13 USB 鼠标与 USB 键盘使用	40
	5.14 LCD 背景亮度调节	41

	5.15	USB 接口测试	41
	5.16	触摸屏测试	42
	5.17	CPU 温度	42
	5.18	CPU 主频	42
	5.19	查看 CPU 信息	43
	5.20	查看内存信息	44
6.	环境招	搭建	
	6.1 3	安装虚拟机软件 VMware	45
	6.2 ゔ	交叉工具链	47
	6.3 绯	编译 Helloworld 源程序	48
	6.4 绯	编译 Uboot	
	6.5 绯	编译内核	
	6	6.5.1 内核源码简介	50
	6	6.5.2 内核配置	51
	6	6.5.3 内核编译	54
	6.6 指	搭建 QT 编译环境	55
	6.7 绯	编译 QT 的程序	60
7.	免责声	声明	64

- 1. 产品介绍
- 1.1 产品简介



图 1.1 TW-STM32MP157-EVM 评估板外观

1.2 硬件资源介绍:

表 1.1 TW-STM32MP157 评估板参数表

产品名称	TW-STM32MP157 评估板		
操作系统	Linux		
处理器	STM32MP157A		
主频	A7 主频 650MHz, M4 主频 209MHz		
内存	1GB		
电子硬盘	2~ 8GB		
3D-GPU	支持		
WiFi	RTL8723DS		
摄像头	1 路, CSI, 可扩展模拟摄像头		
LCD 最高分辨率	1366 * 768		

LVDS	一路 LCD 转 LVDS		
触摸屏	电容触摸屏		
音频	wm8960 立体声编解码器		
USB	3 路 USB2.0		
串口	最高8路(复用)		
CAN-Bus	2 路		
以太网	1 路		
ADC	2 通道		
DAC	2 通道		
SD卡接口	最高2路(复用)		
I2C	6 路		
SPI	6 路		
SAI	4 路		
GPIO	176		
电源供电	直流+12V 电源供电		

注:受限于评估底板的尺寸与接口布局,核心板部分资源以插针方式引出。

1.3 软件资源介绍

TW-STM32MP157-EVM 评估板提供完善的 Linux BSP,包括 Linux 内核源码、和开发工具等,具体软件资源如下表所示:

软件资源		说明		
Linux 内核		5.4.31		
文件系统		根文件系统采用 ext4,在根文件系统上可挂载多种文件系统,如:sysfs、 yaffs2、ubifs 等		
交叉编译器(内核)		arm-none-linux-gnueabihf-gcc 9.2.1		
交叉编译器(应用程序)		arm-none-linux-gnueabihf-gcc 9.2.1		
	eMMC	驱动源码: /drivers/mmc		
外设驱	SD/MMC	驱动源码: /drivers/mmc		
动	LCD	驱动源码: /drivers/gpu/drm/panel/panel-simple.c		
	触摸屏	驱动源码: /drivers/input/touchscreen/		
摄像头		驱动源码: /drivers/media/platform/mxc/capture/		

表 1.2 TW-STM32MP157-EVM 评估板软件资源表

	I2C	驱动源码: i2c/busses/i2c-stm32f7.c		
	UART	驱动源码: drivers/tty/serial/stm32-usart.c		
	USB	驱动源码: /drivers/usb/host/xxx-platform.c		
	以太网 驱动源码: drivers/net/ethernet/stmicro/stmmac/dwmac-stm32			
	CAN	驱动源码: drivers/net/can/m_can/m_can.c		
	WIFI	驱动源码: /drivers/net/wireless		
PWM		驱动源码: /drivers/pwm		
	GPIO	驱动源码: /drivers/gpio		
	RTC	驱动源码: /drivers/rtc		
图形界面		使用 QT5.14.2		
工具软件		如系统镜像烧写工具、串口调试工具、网络调试工具、tftp 服务器软件等		

1.4 产品布局介绍



图 1.2 TW-STM32MP157-EVM 评估板布局图 注:图片仅供参考,以实际销售产品为准

表 1.4 评估底板跳线与指示灯说明

序号	评估底板标号	描述
----	--------	----

1	LED1	运行指示灯, GPIO 控制		
2	PWR	电源指示灯		
3	RST	主板复位按键		

2. 开发板准备事项

2.1 上电前需要注意的事项

- 首先检查电源是否插上,如果电源不插上,仅由 USB_TTL 供电可能会导致整个开发 板供电 不足,导致 LCD 或者 HDMI 无法显示和 LCD 无法触摸及 4G 模块无法正 常工作等情况。
- 防止底板接触到金属异物,将底板下的某两个触点短路。在使用久了的情况下也要注 意观察有没有异物落在开发板的上面,以防短路,注意防水,防潮,防尘等

序		U15 拨码顺序	描述			
号	3	2	1	JX Mt		
1	off	on	EMMC 启动			
2	off	off off off		USB 启动		
3	on	off	on	SD/TF 启动		
4	off	off	on	Cotex-M4		

2.2 拨码开关设置及登录开发板

表 2.1 拨码开关启动选择说明

硬件连接如下:

(1) 断开 TW-STM32MP157-EVM 评估板的供电电源;

(2)把TW-STM32MP157-EVM评估板设置为EMMC启动方式(U15拨码开关拨到off on off);

2.3 安装 SecureCRT 软件

2.3.1 安装 SecureCRT

用户可以从"TW-STM32MP157光盘资料\2.软件开发参考资料\5.工具软件 \1.SecureCRT\SecureCRT"下载 SecureCRT 安装程序。直接将 SecureCRT 文件夹放到电脑的任意位置,点击文件夹中的 SecureCRT.exe 执行程序即可使用。

2.3.2 使用 SecureCRT 登录

运行 SecureCRT 软件,弹出"连接"对话框,如下图 2-1 所示:

	× 🖆 🖆 🗊 💡
Quick Connect 25 192.168.2.10 192.168.2.10 192.168.2.11 192.168.2.20 202.100.22.29 202.100.22.29 202.100.22.29 (1) Serial-COM19 Serial-COM20	
	□在一个标签中打开(0)

图 2.1 SecureCRT "连接"对话框

点击"Quick Connect"	按钮,	弹出	"快速连接"	对话框,	如下图所示:
-------------------	-----	----	--------	------	--------

♪议(Ľ):	Serial	~	
端口(@):	COMS	~	数据流控制
皮特率(B):	115200	\sim	DTR/DSR(I)
数据位(D):	8	\sim	$\left(\begin{array}{c} \mathbf{KIS}/\mathbf{CIS}(\mathbf{K}) \\ \mathbf{XON}/\mathbf{XOFF}(\mathbf{X}) \end{array} \right)$
奇偶校验(<u>A</u>):	无	\sim	
停止位(<u>S</u>):	1	~	
停止位(<u>S</u>):	1	~	
一启动时显示	快速连接(2))	☑ 保存会话(Y)
			方 太持繁出打开(*)

图 2.2 SecureCRT "快速连接"对话框

在该对话框中,选择"协议"为"Serial","端口"需根据主机实际使用的串口号进 行选择(可从操作系统的"设备管理器"中获得该信息),并对串口参数进行相应的设置, 值得注意的是在设置串口参数时需关闭所有的"数据流控制"选项。在设置完成后,点击 "连接"按钮,会出现一个用于登录的 SecureCRT 终端窗口。

Serial-COM21 - SecureCRT	
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)	
19 19 G 49 10 10 11 16 18 19 19 19 19 19 19 19 19 19 19 19 19 19	Ŧ
Serial-COM21	4 ⊳
<pre>[root@TWDZ-stm32mp157]:~# [root@TWDZ-stm32mp157]:~# [root@TWDZ-stm32mp157]:~# [root@TWDZ-stm32mp157]:~#</pre>	*
< III	
, Serial: COM21, 115200 4, 27 24行, 80列 VT100	大写数字

图 2.3 SecureCRT 终端登录窗口

3. ST 官方软件安装

3.1 Java 环境安装

在 Window 安装 STM32CubeProgrammer 软件需要安装 Java 环境才能正常使用。如果没 有安装 Java 软件包,在打开的 STM32CubeProgrammer 软件的时候会报以下错误。

This application requires a Java Runtime Environ	ment 1.8.0_66 - 10.99.99

图 3.1 STM32CubeProgrammer 软件打开错误

可以在路径为"TW-STM32MP157 光盘资料\2. 软件开发参考资料\5. 工具软件\固件烧录 软件"中选中 jre-8u291-windows-x64. exe 进行安装。

Java 安装程序 - 欢迎使用	
欢迎使用 Java - 更新的许可证条款	
授权使用此软件版本的条款已更改。 <u>更新的许可协议</u> 此 Java 运行时版本仅授权供您的个人(非商用)台式机和笔记本电脑使用。 要对此软件进行商业使用,需要从 Oracle 或您的软件供应商获得单独的许可证。 单击"安装"可接受许可协议并立即安装 Java,单击"删除"可从您的系统中卸载 Java。 在我们的安装过程中,不会收集个人信息。 <u>有关我们所收集信息的详细信息</u>	
□ 更改目标文件夹 安装 日	刪除(R)

更改目标文件夹后点击安装,如果使用默认安装路径直接点击安装即可

图 3.2 Java 环境的安装



图 3.3 Java 安装过程

安装完成后,直接点击关闭

Java 安装 - 完成	- • •
✔ 您已成功安装 Java	
当有 Java 更新可用时, 系统将会提示您。 请始终安装更新以获取最新的性能和安全改进。 <u>有关更新设置的详细信息</u>	
	关闭(C)

图 3.4 Java 安装完成

安装完 Java 环境后,STM32Cube 系列软件可以正常打开使用。

3.2 STM32CubeProgrammer 下载

3.2.1 STM32CubeProgrammer 的简介

STM32CubeProgrammer 简称 STM32CubeProg, 是一个适用于 STM32 系列产品的跨 平台、多合一的程序烧写工具。有如下特点:

(1) "跨平台" 体现在支持 Windows、macOS 和 Linux 操作系统,软件运行时 需要 Java 环境。

(2) "多合一"体现在支持通过 USB、ST-LINK、UART、OTA 多种方式来烧写 固件。

3.2.2 STM32CubeProgrammer 软件下载

我们需要在 Windows 使用上位机通过 USB Type-C 连接线连接到 TW-STM32MP157 开发板,然后进行烧写固件操作,所以我们就需要这样的上位机软件, STM32CubeProgrammer 是由 ST 官方提供的,适用于烧写 ST 各种平台,现在我们只讲 用 USB Type-C 连接方法烧写烧写固件到 TW-STM32MP157 开发板的方法。路径为 "TW-STM32MP157 光盘资料\2.软件开发参考资料\5.工具软件\固件烧录软件"。



图 3.5 STM32CubeProgrammer 软件

点击 next



图 3.6 STM32CubeProgrammer 欢迎界面

STM32CubeProgram	nmer Installation Wizard	
Information Step 2 of 8	STM32 CubeProgra	mmer
Ite augmented	Please read the following information: SIM32Cube programming software for SIM32 microcontrollers	•
STM32	Features ======= • Supports Motorola S19, Intel HEX, ELF, and binary formats • Erases, programs, views and verifies the	111
Cube	 content of the device Flash memory Supports debug and bootloader interfaces: SI-LINK debug probe (JIAG/SWD) UARI and USB DFU bootloader interfaces Programs, erases and verifies external 	
STMicroelectronics	memories, with examples of external Flash	+ Luit

图 3.7 STM32CubeProgrammeer 安装信息

选择第1选项,然后 next



图 3.8 STM32CubeProgrammer 软件条款同意

选择软件的安装路径,然后 next

Pre STM32CubeProgramm	er Installation Wizard	
STM32CubeProgrammer : Step 4 of 8	Installation path	1 STM32 CubeProgrammer
LT1	Select the installation path: ram Files\SIMicroelectronics\SIM32Cube\SIM	132CubeProgrammer <u>B</u> rowse
life.augmented		
STMicroelectronics	Erev	i ous Pext Quit

图 3.9 STM32CubeProgrammer 安装路径



图 3.10 STM32CubeProgrammer 默认安装选项

Pro STM32CubeProgramm	ner Installation Wizard	
STM32CubeProgrammer Step 6 of 8	Package installation	STM32 CubeProgrammer
	Pack installation progress: E:\Program Files\util\populateApi.bat	
life.augmented	Overall installation progress:	
STM32 Cube		
STMicroelectronics	Previous] 🗭 Mext 🖉 Quit

图 3.11 STM32CubeProgrammer 安装过程

在安装的时候,需要安装相关的驱动软件,选择"始终安装此驱动程序软件"



图 3.12 驱动软件的安装



图 3.13 驱动安装界面

Device Driver Installation Wizard



图 3.14 驱动安装完成

3.3 STM32CubeIDE 下载

3.3.1 STM32CubeIDE 的简介

1. STM32CubeIDE 是一个先进的 C/C++开发平台,具有 STM32 微控制器的 IP 配置, 代码生成,代码编译和调试功能。

2. 它基于 ECLIPSE™/ CDT 框架和用于开发的 GCC 工具链,以及用于调试的 GDB。 它允许集成数百个现有插件,完成 ECLIPSE™IDE 的功能。

3.3.2 STM32CubeIDE 软件下载

光盘资料提供了 STM32CubeIDE 软件的安装包,路径为"TW-STM32MP157 光盘资料\2. 软件开发参考资料\5.工具软件\固件烧录软件"的 st-stm32cubeide_1.4.0_7511_20200720_0928_x86_64.exe。STM32CubeIDE 的安装比较简单, 基本一路 Next 下去就行,该勾选的勾选上即可。注意:(STM32CubeIDE 安装包不能放在带 中文路径下,否则会报以下图 3.15 的错误)



图 3.15 安装包有中文路径错误

点击 next



图 3.16 STM32CubeIDE 欢迎界面

点击 I Agree

Please review the license terms before ins STMicroelectronics STM32CubeIDE.	stalling	
Press Page Down to see the rest of the a	greement.	
STMicroelectronics Software License Agre	eement	
SLA0048 Rev4/March 2018		
BY INSTALLING COPYING, DOWNLOADIN SOFTWARE PACKAGE OR ANY PART THE FROM STMICROELECTRONICS INTERNA AFFILIATED COMPANIES (STMICROELEC HIMSELF OR HERSELF, OR ON BEHALF C EMPLOYED AND/OR ENGAGED AGREES T	NG, ACCESSING OR OTHERWISE USING THIS EREOF (AND THE RELATED DOCUMENTATION) TIONAL N.V, SWISS BRANCH AND/OR ITS CTRONICS), THE RECIPIENT, ON BEHALF OF OF ANY ENTITY BY WHICH SUCH RECIPIENT IS TO BE BOUND BY THIS SOFTWARE PACKAGE	÷
LICENSE AGREEMENT.		
LICENSE AGREEMENT. If you accept the terms of the agreement agreement to install STMicroelectronics ST	t, dick I Agree to continue. You must accept th IM32CubeIDE.	e

图 3.17 STM32CubeIDE 安装协议

选择安装路径后,点击 next

STMicroelectronics STM32CubeIDE	
Choose Install Location	
Choose the folder in which to install STMicroelectronics STM32CubeIDE.	
Setup will install STMicroelectronics STM32CubeIDE in the following folde different folder, click Browse and select another folder. Click Next to cor	r. To install in a ntinue.
Destination Folder	
G:\jde\STM32CubeIDE_1.4.0	Browse
Space required: 2.2 GB Space available: 94.3 GB	
< <u>B</u> ack	:> Cancel

图 3.18 STM32CubeIDE 安装路径

点击 install

Choose which features of STM want to install.	icroelectronics STM32CubeIDE y	ou
Check the components you wa nstall. Click Install to start the	nt to install and uncheck the con installation.	nponents you don't want to
Select components to install:	SEGGER J-Link drivers	Description Position your mouse over a component to see its description.
Space required: 2.2 GB		

图 3.19 STM32CubeIDE 安装配置选项

STMicroelectronics STM32CubeIDE	
Installing	
Please wait while STMicroelectronics STM32CubeIDE is being installed.	
Execute: "G:\ide\STM32CubeIDE_1.4.0\STM32CubeIDE\drivers	\JLinkUSBDriver\InstDrivers.ex
Show <u>d</u> etails	
(Pode	Nevt > Cancel
	Mext > Cancel

图 3.20 STM32CubeIDE 正在安装中

点击 next

etup was completed successfully.		
ompleted		
Extract: winXPTSFrame-rotated.png		-
Extract: winXPTSFrame.png		
Output folder: G:\ide\STM32CubeIDE_	1.4.0\STM32CubeIDE\readme	
Extract: readme_eclipse.html		
Output folder: G:\ide\STM32CubeIDE_	_1.4.0	
Output folder: G:\ide\STM32CubeIDE_	1.4.0\STM32CubeIDE	
Created uninstaller: G:\jde\STM32Cub	eIDE_1.4.0\STM32CubeIDE\uninstall.exe	
Create folder: C:\ProgramData\Micros	oft\Windows\Start Menu\Programs\STMicroele	
Create shortcut: C:\ProgramData\Micr	rosoft\Windows\Start Menu\Programs\STMicroe	h
Completed		

图 3.21 STM32CubeIDE 安装完成



图 3.22 STM32CubeIDE 创建桌面快捷方式

4. 烧写和更新固件

4.1 通过 STM32CubeProgrammer 烧写固件

注意:在此之前,需要完成 3.ST 官方软件安装章节才能进行烧写固件。

1. 双击打开 STM32CubeProgrammer 软件

Pre STM3	2CubeProgrammer	
STM32 Cube	Frogrammer (1996)	FI 🕨 💆 🛧 🖅
	Memory & File edition	Not connected
	Device memory Open file +	USB Connect
.	Addre V Sk Data wi 32-bit V Find Data Ox Read V	USB configuration Port No DFU d 0
OB		Serial number
CPU		
swv		
	No data to display	
	Log Verbosity level	
	09:57:15 : STM32CubeProgrammer API v2.5.0	Target information
		Baard _ Device _ Type _ Device ID _ Revision ID _ Fish size _
(?)	0% 🛞	CPU

图 4.1 STM32CubeProgrammer 软件界面

2. 将启动模式设为 USB 模式,查看(2.2 <u>拨码开关设置及登录开发板</u>),然后点击 Connect 键

Pro STM3	2CubeProgrammer	
STM32 Cube	Programmer (1997)	🖪 区 У 🛧 🏹
	Memory & File edition	Not connected
	Device memory Open file +	USB Connect
	Ad • Data 32 • Find 0x Read •	USB contiguration Port USB1 C Serial number 001F00263030510332313
СРИ		Read Unprotect (MCU)
swv	No data to display	
	Log Verbosity level 1 2 3	
()) () () () () () () () () ()	09:57:15 : 5TM32CubeProgrammer API v2.5.0	Target information Board _ Device _ Type _ Device ID _ Revision ID _ Flash size _ CPU _

图 4.2 STM32CubeProgrammer 连接 USB

3. 选中下载的固件 emmc_stm32mp157_tw.tsv 配置文件, 路径为 "TW-STM32MP157 光盘资料\4.固件更新与烧写方法\1.USB 烧写\2.标准固件\MP1_IMAGE"

Pro STM32Cu	DubeProgrammer		
STM32 CubePro	ogrammer	(19)	F 🕒 🦻 🔆 😽
Me	lemory & File edition		Not connected
	Device memory Open file + 1		USB Connect
Ad	d ♥ Open file	▼ 検 搜索 MP1 IMAGE	USB configuration
	142 ▼ 新建文件表	2 ⊯ • □	USB1
OB	國 视频 ▲ 名称 ▲		d Unprotect (MCU)
CPU	■ 图片 ■ 文档	2021/6/18 14:00 TSV 文件	
SWIV	□ ×13 ♪ 音乐	3	
(and	₩ 计算机		
	🏭 本地磁盘 (C:)		
	本地越盐 (D:) 软件 (E:)		
	🕞 本地磁盘 (F:)		
	□ 本地燃量 (G:) = □ My Passport (H		
	99		Taractinformation
		1	rd
	文件名(N): emmc_stm32mp157_tw.tsv	 4Firmware files (*.bin;*.binary; 	
\bigcirc		打开(Q) 取消	ce ID
0			n size
		0% 🛞	-

图 4.3 STM32CubeProgrammer 导入配置文件

4. Binaries path 的路径是下载的固件路径,选择正确后,按 Download 键进行下载。

Pre STM	32CubePr	rogran	nmer							
STM32 Cube	Program	nmer							🕸 f 🤇	> 🖌 🔀
Ξ	Memo	ory &	ı File e	dition						🛑 Not connected
	Device	e mem	nory e	mmc_stm32mp157_t	w.tsv × +				USB	Connect
	Addres	s		* Size		Data w	idth 32-bit	Find Data Ox Download	USB	configuration
	Select	Opt	Id	Name	Туре	IP	Offset	Binary	Port USB1	- Ø
OB		-	0x1	fsbl1-boot	Binary	none	0x00000000	tf-a-stm32mp157d-tw-serialboot.stm32	Serial number	001F00263030510332313933
H		-	0x3	ssbl-boot	Binary	none	0x00000000	u-boot.stm32	Read Unprotect (MCU)	-
CPU		P	0x4	fsbl1	Binary	mmc1	0x000000B	tf-a-stm32mp157d-tw-trusted.stm32		
swy		Р	0x5	fsb12	Binary	mmc1	0x000000B	tf-a-stm32mp157d-tw-trusted.stm32		
	\checkmark	PD	0x6	ssbl	Binary	mmc1	0x00080000	u-boot.stm32		
		P	0x21	boot	System	mmc1	0x00280000	bootfs.ext4		
		P	0x22	rootfs	FileSystem	mmc1	0x04280000	rootfs.ext4		
	20									
	Rinarie	c nath	0	Lisers\ Administrator\ De	sktop\TW_STM32MP1	57米舟湾	814 因任事新与做军	方法\11/SPI构石\2 标准图性\MD1 TMAGE Browse		
		s paul	C.	oscis (danning dator (de	acceptine of mozining	St / Line Set	1044112-01-024-0			
	Log							Verbosity level 1 2 3		
	09:57	:15 : :53 :	STM32 Read	CubeProgrammer API TSV File: C:\Users	V2.5.0 Administrator\	Desktop	TW-STM32MP157	(盘资料\4.固件更新与赎写方法\1.USB赎写\2.标准固件	Tarc	et information
	10:13	IMAGE	Number	r of partitions: 7				P	Board Device	-
$\overline{\circ}$									Туре	.
Ø									Revision ID	-
0									Flash size	-
÷								0% 🛞	CPO	

图 4.4 STM32CubeProgrammer 烧写固件

有时候在烧写固件的时候,并不想将所有的固件重新烧写一次,然后可以通过修改烧 写脚本来进行部分烧写。

1. 打开烧写脚本 emmc_stm32mp157_tw.tsv,路径为 "TW-STM32MP157 光盘资料\4.固 件更新与烧写方法\1.USB 烧写\2.标准固件\MP1 IMAGE"

1	#Op	t Id	Name	Type	Device	Offset	Binary
2	- 1	0x01	fsbl1-b	oot Bir	nary non	e 0x0	tf-a-stm32mp157d-tw-serialboot.stm32
3		0x03	ssbl-bo	ot Bir	nary non	e 0x0	u-boot.stm32
4	P	0x04	fsbl1	Binary	mmc1	boot1	tf-a-stm32mp157d-tw-trusted.stm32
5	P	0x05	fsbl2	Binary	mmc1	boot2	tf-a-stm32mp157d-tw-trusted.stm32
6	PD	0x06	ssbl	Binary	mmc1	0x00080	000 u-boot.stm32
7	Ρ	0x21	boot	System	mmc1	0x00280	000 bootfs.ext4
8	P	0x22	rootfs	FileSys	stem mmc	1 0x0	1280000 twdz rootfs emmc.ext4

图 4.5 emmc_stm32mp157_tw.tsv 内容

域	作用
Opt	选项字段,可以设置为"-"、"P"、"D"或"E"
Id	会根据这个 id 来决定烧写分区
Name	分区名字
Туре	制定烧写的类型, 仅 uboot 使用。
Device	指定烧写的设备类型与编号,比如 emmc0、emmc1、nand0 等,如果 opt 为 '-',那么此字段就为 none
Offset	分区的起始位置,如果为"boot1"表示 EMMC 的第一个分区,如果为 "boot2"就表示 EMMC 第二个分区。如果是数字就表示需要偏移的字节 数。
Binary	要烧录的文件

表 4.1.1 emmc_stm32mp157_tw.tsv 的配置解析

2. 通过配置 Opt 项,可以选择烧写指定固件

'-':空选型。

'P':更新分区或设备,也就是向分区或设备烧写固件。

'PE':不更新,也就是指定某个分区或者设备不需要烧写固件,这样我们就可以单独只更新 tf-a、uboot、kernel 或者 rootfs。

'PD':删除并更新,也可以写作 DP。

'PDE': 删除并且保持为空,也可以写作 PED/DPE/DEP/EPD/EDP。

4.2 通过 SD 卡进行烧写固件

通过 SD 卡烧写固件的实际速度要比 USB 烧写速度要快 3 分钟左右,所以,我们也可 以通过 SD 卡去烧写或者更新 EMMC。

1. 将光盘资料路径为"TW-STM32MP157光盘资料\4.固件更新与烧写方法\2.TF卡烧 写\1.TF 卡制作\"中的文件夹 SDupdate 拷贝到虚拟机上。

2. 准备一张 SD 卡烧写固件,存储大小至少 4GB (SD 卡需要格式化,所以在格式化前,将里面的文件备份好,以免丢失),/dev/sdb 是该 SD 卡的节点。



3. 执行 SDupdate 文件中的 sd_update.sh 脚本,执行指令:./sd_update.sh /dev/sdb。执行 过程中会打印以下信息,看到"SD 启动卡制作完成"的字样即完成 SD 卡制作。

```
Disk identifier (GUID): 7C1CA90B-1281-4E1F-82D9-A6500EA5BE92
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 15554526
Partitions will be aligned on 1-sector boundaries
Total free space is 0 sectors (0 bytes)
Number Start (sector)
                                  End (sector)
                                                   Size
                                                                   Code
                                                                          Name
                                                   256.0 KiB
                                                                          fsbl1
                                                                   8301
                       34
                                           545
   1
   2
                      546
                                          1057
                                                   256.0 KiB
                                                                           fsbl2
                                                                   8301
                     1058
                                          5153
   3
                                                   2.0 MiB
                                                                   8301
                                                                           ssbl
   4
                     5154
                                       136225
                                                   64.0 MiB
                                                                   8300
                                                                           bootfs
                                                   100.0 MiB
                  136226
                                       341027
                                                                   8300
                                                                          rootfs
   5
                                                   7.3 GiB
   б
                  341028
                                    15554526
                                                                   8300
                                                                          emmc_rootfs
The operation has completed successfully.
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 65536 1k blocks and 16384 inodes
Filesystem UUID: fc37a255-57f6-455b-854a-b6e5bd5f50d5
Superblock backups stored on blocks:
          8193, 24577, 40961, 57345
Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
   😑 🗉 twdz@ubuntu:~/MP1/twdz源码包/SDupdate
usr/share/udhcpc/
usr/share/udhcpc/default.script
usr/share/dict/
usr/games/
usr/src/
var/
var/volatile/
var/run
var/spool/
var/spool/mail/
var/log
var/cache/
var/cache/ldconfig/
var/cache/ldconfig/aux-cache
var/cache/rpm/
var/lib/misc/
var/lib/misc/
var/lib/urandom/
var/lock
var/backups/
var/local/
var/tmp
文件系统写入成功
sp启动卡制作完成..
 b后动下前TF元成......
:wdz@ubuntu:~/MP1/twdz源码包/SDupdate$
```

4. 将评估板拨码到 TF 卡启动模式,插入烧写过固件的 TF 卡,上电,当打印出以下 信息并且伴随蜂鸣器响时,说明 TF 卡烧写完成。

Populating dev cache
Fri Aug 25 07:15:10 UTC 2017
INIT: Éntering runlevel: 5
check file ok
write TE-A file
write TE-A file finish
Write oboot file
write u-boot file finish
write Bootfs file
write bootfs file finish
write Rootfs file
write Rootfs file finish
Undate complete
opdate comprete

5. 断电,将拨码拨回 EMMC 启动即可。

5. 功能测试

5.1 LED 测试

对开发板的 LED0、LED1 进行测试,串口终端执行相应的指令去控制对应的 IO 来 控制对应的继电器和 LED 灯的亮灭,如下图所示。



图 5.1 LED0 原理图

当执行以下指令后,开发板中 LED0 会亮起, COM1 和 NO1 连通

[root@TWDZ-stm32mp157]:~# echo 82 > /sys/class/gpio/export//申请 IO

[root@TWDZ-stm32mp157]:~# echo out > /sys/class/gpio/gpio82/direction//设置方向

[root@TWDZ-stm32mp157]:~# echo 1 > /sys/class/gpio/gpio82/value//设定输出值

同理可以测试 LED1 是否能正常使用。

5.2 蜂鸣器测试

通过指令对开发板的 Beep 进行测试, Beep 配置为普通 IO 口,通过控制 IO 口的高低 电平可以驱动蜂鸣器,可在命令行下运行如下命令后,蜂鸣器会鸣叫:

[root@TWDZ-stm32mp157]:~#echo 1 > /sys/class/pwm/pwmchip8/export [root@TWDZ-stm32mp157]:~# echo 1000000 > /sys/class/pwm/pwmchip8/pwm1/period [root@TWDZ-stm32mp157]:~# echo 500000 > /sys/class/pwm/pwmchip8/pwm1/duty_cycle [root@TWDZ-stm32mp157]:~# echo 1 > /sys/class/pwm/pwmchip8/pwm1/enable

5.3 串口测试

TW-STM32MP157-EVM的底板上总共有三个串口。

串口名	对应串口	对应设备
UC4(调试串口)	Uart4	/dev/ttySTM0
UC3	Uart3	/dev/ttySTM1
UC5	Uart5	/dev/ttySTM2

1.通过下面的指令配置串口

[root@TWDZ-stm32mp157]:~# stty -F /dev/ttySTM1 ispeed 115200 ospeed 115200 cs8 stty 指令解释:

(1) -F(--file): 打开指定的设备,并用此设备作为输入来代替标准输入

- (2) ispeed N: 设置输入速率为 N
- (3) ospeed N: 设置输出速率为 N
- (4) csN: 把字符长度设为 N

2.在 PC 端打开串口调试工具,该工具可接收开发板串口发送的数据,并可发送数据到 开发板。Windows 下的串口调试软件比较多,此处以 sscom32.exe 为例介绍串口调试工具 的使用,该工具可在"TW-STM32MP157光盘资料/2、软件开发参考资料/5、开发工具软 件\SSCOM32"中直接获取。

打开 sscom32.exe 软件,并进行相应的配置,对话框如下图所示:

▲ SSCOM V5.13.1 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502449(最新版本)
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者
Settings Port Baud rate 115200 Data bits 8 Stop bits 1 Parity None Flow control None OK Cancel
端口号 COM27 Silicon Labs CP210x 🔽 🗆 HEX显示 保存数据 🖂 接收数据到文件 🖂 HEX发送 🗆 定时发送: 1000 ms/次 🗂 加回车换行,
● 关闭串口 2 厘多串口设置 ▼ 加时间戳和分包显示,超时时间: 20 ms 第1 字节 至 末尾 ▼ 加校验 None ▼ FD 89 4
HISIDIH 波特率: 115200 ▼ Weicome to InDL
为了更好地发展SSU的软件 发送 请您注册嘉立创作结尾客户 发送 ·
【升级到SSCOM5.13.1】★2. ★RT-Thread中国人的开源免费操作系统 ★新一代WiFi芯片兼容8266支持RT-Thread ★8KM远距离WiFi可自组网
www.daxia.com S:0 R:0 COM27 已打开 115200bps,8,1,None,None //

在该对话框中设置主机侧的串口号,并设置串口通讯参数为: 波特率 115200,8 位数 据位,1 位停止位,无数据校验位。勾选"HEX 显示"选项。设置完成后,点击"打开 串口"按钮。

测试开发板发送数据是否正常。方法如下:

1. 在开发板执行如下命令:

[root@TWDZ-stm32mp157]:~# echo "twdz test ttySTM1" > /dev/ttySTM1

2. 此时 SSCOM..exe 会收到/dev/ttySTM1 发过来的信息,如下图所示

▲ SSCOM V5.13.1 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502449(最新版本)	3
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者	
[17:12:57.682]₩x ← ♦twdz test ttySTM1	*
 	-
	?
▼ KIPHI C ZZPHIKE VARANCE TYDZ	-
为了更好地发展SSCOM软件 请您注册嘉立创作结尾客户 发送	-
【升级到SSCOM5.13.1】★2. ★RT-Thread中国人的开源免费操作系统 ★新一代WiFi芯片兼容8266支持RT-Thread ★8KM远距离WiFi可自组网	
www.daxia.com S:0 R:19 COM27 已打开 115200bps,8,1,None,None	11.

接下来测试开发板串口接收数据是否正常。方法如下:

1. 执行以下指令

[root@TWDZ-stm32mp157]:~# microcom -t 10000 -s 115200 /dev/ttySTM1

####microcom 读数据, 10 秒无数据, 退出

2. 在 SSCOM.exe 上,输入需要发送的数据,如果数据是二进制字节数据(即非 ASCII 字符)需勾选 "HEX 发送"选项。设置完成后,点击"发送"按钮。如下图所示:

↓ SSCOM V5.13.1 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502449(最新版本)	3
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者	
[17:12:57.682]收←◆twdz test ttySTM1 [17:17:08.908]发→◇welcome to TWDZ□	*
	÷
清除窗口 打开文件 C:\Users\Administrator\Desktop\hyb\uu.dat 发送文件 停止 清发送区 □ 最前 □ English 保存参数 扩展 -	-
端口号 COM27 Silicon Labs CP210x ▼ FHEX显示 保存数据 F 接收数据到文件 FHEX发送 F 定时发送: 1000 ms/次 F 加回车换行	?
● 美闭串口 さ 更多串口设置 ▼ 加时间戳和分包显示,超时时间: 20 ms 第1 字节 至 末尾 ▼ 加校验 None ▼FD 89 ■ BIS □ DIE 波转案: 115200 ▼ welcome to TWDZ	-
为了更好地发展SSCOM软件 请你注册真立创作程序空白	
【升级到SSCOM5.13.1】★2. ★RT-Thread中国人的开源免费操作系统 ★新一代WiFi芯片兼容8266支持RT-Thread ★8KM远距离WiFi可自组网	-
www.daxia.com S:15 R:19 COM27 已打开 115200bps,8,1,None,None	1.

3. 查看串口接收信息,会出现

[root@TWDZ-stm32mp157]:~# microcom -t 10000 -s 115200 /dev/ttySTM1

welcome to TWDZ

5.4 WIFI 测试

查看是否加在到 wifi 驱动

[root@T can0	<pre>wDZ-stm32mp157]:~# ifconfig -a Link encap:UNSPEC Hwaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-</pre>
eth0	Link encap:Ethernet HWaddr ae:18:4d:a0:6b:0d inet addr:192.168.1.10 Bcast:192.168.1.255 Mask:255.255.255.0 UP BROADCAST MULTICAST DYNAMIC MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B) Interrupt:55 Base address:0xa000
10	Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 UP LOOPBACK RUNNING MTU:65536 Metric:1 RX packets:2 errors:0 dropped:0 overruns:0 frame:0 TX packets:2 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:140 (140.0 B) TX bytes:140 (140.0 B)
vlan0	Link encap:Ethernet HWaddr 68:b9:d3:cf:ca:b2 UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

```
1、使能 wifi
```

[root@TWDZ-stm32mp157]:~# rfkill unblock all

2、打开 wlan0

[root@TWDZ-stm32mp157]:~# ifconfig wlan0 up

3、配置 wifi 连接文件

根文件系统的/etc 目录下存在 "wpa_supplicant.conf" 的配置文件,此文件用于配置要 连接的 WIFI 热点以及 WIFI 密码,比如我要连接到 "TWDZ" 这个热点上。

因此 wpa_supplicant.conf 修改部分文件内容如下所示:

[root@TWDZ-stm32mp157]:~#vi /etc/wpa_supplicant.conf

network={

```
ssid="TWDZ"
psk="twdz123456"
```

4、连接 wifi

}

准备好以后就可以使用 wpa_supplicant 工具让 RTL8723DS USB WIFI 连接到热点上,输入如下命令:

[root@TWDZ-stm32mp157]:~#wpa_supplicant -D wext -B -i wlan0 -c /etc/wpa_supplicant.conf

使用 udhcpc 命令从路由器申请 IP 地址, 输入如下命令: udhcpc -i wlan0

```
[root@TWDZ-stm32mp157]:~/wifi# udhcpc -i wlan0
udhcp client (v0.9.8) started
Sending discover...
sending select for 192.168.43.63...
Lease of 192.168.43.63 obtained, lease time 3599
deleting routers
SIOCDELRT: No such process
adding dns 192.168.43.1
[root@TWDZ-stm32mp157]:~/wifi#_____
```

可知分配到地址为 192.168.43.63

5、测试 wifi 能否连接外网

[root@TWDZ-stm32mp157]:~#ping baidu.com

```
[root@TWDZ-stm32mp157]:~/wifi# ping baidu.com
PING baidu.com (39.156.69.79) 56(84) bytes of data.
64 bytes from 39.156.69.79: icmp_seq=1 ttl=48 time=98.8 ms
64 bytes from 39.156.69.79: icmp_seq=2 ttl=48 time=134 ms
^c64 bytes from 39.156.69.79: icmp_seq=3 ttl=48 time=424 ms
--- baidu.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 10354r
rtt min/avg/max/mdev = 98.850/219.320/424.342/145.712 ms
```

5.5 4G 测试

1. 首先进入\home下的 4g 目录,直接执行./4g.sh

[root@TWDZ-stm32mp157]:/# cd home/4g/

[root@TWDZ-stm32mp157]:/home/4g# ./4g.sh

会出现以下打印信息。

Guangzhou TaloWe Electronics Technology Co., Ltd.

send (ATDT*99***1#^M) send (ATDT*99***1#^M) expect (CONNECT) ٨M ٨M CONNECT -- got it Script chat -s -v -f /etc/ppp/gprs-connect-chat finished (pid 1076), status = 0x0
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB2
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xc25a1a30> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0x7c10622> <pcomp> <accomp
>] Sent [LCP ConfReq id=0x1 <asyncmap 0x0> <mag1c 0xC25aLa30> <pcomp> <accomp>]
No auth is possible
sent [LCP ConfReq id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0x7c10622> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x25aLa30> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x25aLa30> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x7c10622> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x7c10622> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x7c10622> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x7c10622> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <compress v] of 01> <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0
.0.0.0>]
rcvd [LCP DiscReq id=0x2 magic=0x7c10622]
rcvd [LCP DiscReq id=0x3 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03 2f]
Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfReq id=0x1 <compress v] of 01>]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfReq id=0x1]
sent [IPCP ConfReq id=0x1]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
rcvd [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0]
sent [IPCP ConfReq id=0x1]
sent [IPCP ConfReq id=0x1]
sent [IPCP ConfReq id=0x1]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms 7>1 sent [IPCP ConfReq id=0x3 <addr 10.38.26.11> <ms-dns1 221.179.38.7> <ms-dns2 120.196.165. 7>] rcvd [IPCP ConfAck id=0x3 <addr 10.38.26.11> <ms-dns1 221.179.38.7> <ms-dns2 120.196.165. 7>] Could not determine remote IP address: defaulting to 10.64.64.64 local IP address 10.38.26.11 remote IP address 10.64.64.64 primary DNS address 221.179.38.7 secondary DNS address 120.196.165.7

-- got it

```
send (ATSO=0^M)
expect (OK)
^M
ATS0=0^M^M
ок
-- got it
send (ATEOV1^M)
expect (OK)
^M
ATEOV1^M^M
OK.
 -- got it
send (AT+CSQ^M)
expect (OK)
^M
٨M
+C5Q: 26,99^M
AM
OK
 -- got it
send (AT+CGDCONT=1,"IP","CMNET"^M)
expect (OK)
OK
```

Talowe。广州眺望电子科技有限公司

abort on (\nBUSY\r) abort on (\nNO ANSWER\r) abort on (\nRINGING\r\n\r\nRINGING\r) timeout set to 40 seconds send (AT^M) expect (OK) ATAMAM OK

OK

-- got it

33

2. 使用指令 if config 指令查看是否生成 ppp0 节点

eth0	Link encap:Ethernet Hwaddr 9E:11:2F:FF:9D:01 UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B) Interrupt:55
10	Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 UP LOOPBACK RUNNING MTU:65536 Metric:1 Rx packets:82 errors:0 dropped:0 overruns:0 frame:0 TX packets:82 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:6220 (6.0 KiB) TX bytes:6220 (6.0 KiB)
орр0	Link encap:Point-to-Point Protocol inet addr:10.38.26.11 P-t-P:10.64.64.64 Mask:255.255.255.255 UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1 Rx packets:6 errors:0 dropped:0 overruns:0 frame:0 TX packets:14 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:3 RX bytes:192 (192.0 B) TX bytes:545 (545.0 B)
usb0	Link encap:Ethernet HWaddr 7A:0A:9C:41:4D:2D UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

3. 通过指令尝试连接外网 Ping baidu.com

PING baidu.com (39.156.69.79) 56(84) bytes of data.
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=5 ttl=50 time=76.2 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=6 ttl=50 time=80.4 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=7 ttl=50 time=78.7 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=8 ttl=50 time=76.7 ms
AC .
baidu.com ping statistics
8 packets transmitted, 4 received, 50% packet loss, time 11283ms
rtt min/avg/max/mdev = 76.177/77.981/80.383/1.667 ms

5.6 时钟设置

Linux 将时钟分为系统时钟(System Clock)和硬件时钟(Real Time Clock,简称 RTC)两种。系统时钟是由 Linux 内核所维护的时钟,用户一般使用和看到的都是系统时钟。而硬件时钟则是由主板上的电池供电的主板硬件时钟。系统时钟在系统断电后即会消失,但 RTC 时钟在主板电池有电的情况下会长期运行。因此每次上电时,Linux 内核都会读取主板上的 RTC 时钟,并将它同步到系统时钟。下面列出一些与时钟相关的命令:

5.6.1 查看系统时钟:

使用 date 命令可以查看系统时钟:

[root@TWDZ-stm32mp157]:~# date

Sat Jan 4 21:09:19 CST 2020

5.6.2 查看 RTC 时钟:

使用 hwclock 命令可以查看 RTC 时钟:

[root@TWDZ-stm32mp157]:~#hwclock

Mon Dec 7 10:01:30 2020 0.000000 seconds

5.6.3 设置 RTC 时钟:

使用 hwclock -w, 可以将系统时钟写入 RTC 时钟:

[root@TWDZ-stm32mp157]:~# hwclock -w

5.6.4 同步系统时钟:

使用 hwclock -s, 可以将 RTC 时钟写入系统时钟:

[root@TWDZ-stm32mp157]:~# hwclock -s

通过上面的叙述可以看出,如果想要改变当前的系统时间,且希望系统重启后改变依 然生效,需要执行如下两步操作:

● 使用 date -s 命令修改当前的系统时钟;

● 使用 hwclock -w 命令将修改后的系统时钟写入 RTC 时钟。

例如需要将当前时钟设置为 2020-12-07 10:03:10,并希望该改变在系统重启后依然 有效,应执行如下命令:

[root@TWDZ-stm32mp157]:~# date -s "2020-12-07 10:03:10"

Mon Dec 7 10:03:10 UTC 2020

[root@TWDZ-stm32mp157]:~# hwclock -w

重启计算机后,如果 RTC 正常的话,使用以下命令可以查看到刚刚设定的时间值: [root@TWDZ-stm32mp157]:~# hwclock

5.7 CAN 测试

TW-STM32MP157-EVM 底板上只有一路 CAN 接口,如果想要测试 CAN 接口,用户 需要找另外的 CAN 口进行对接测试。

1. 设置 CAN 的速率

[root@TWDZ-stm32mp157]:~# ip link set can0 up type can bitrate 50000

2. 发送 CAN 数据

[root@TWDZ-stm32mp157]:~# cansend can0 123#01.02.03.04.05.06

3. 接收 CAN 数据

[root@TWDZ-stm32mp157]:~#candump -ta can0

5.8 摄像头测试

注: 在测试之前请接好相应的摄像头,注意引脚顺序,执行命令后就可以在液晶上看 到相应的摄像头画面,TW-STM32MP157-EVM 开发板提供 CSI 摄像头接口:

在控制端输入 v4l2-ctl --device=/dev/video0 --list-formats-ext 查看摄像头支持格式、分辨率及帧率。

[root@īwbz-stm32mp157]:~# v4l2-ctldevice=/dev/video0list-formats-ext
ioctl: VIDIOC_ENUM_FMT
Type: Video Capture
[0]: 'JPEG' (JETE JPEG, compressed)
Size: Discrete 176x144
Interval: Discrete 0.067s (15.000 fps)
Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 320x240
Interval: Discrete 0.067s (15.000 fps)
Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 640x480
Interval: Discrete 0.067s (15.000 fps)
Interval: Discrete 0.033s (30.000 tps)
Interval: Discrete 0.017s (60.000 fps)
Size: Discrete /20x480
Interval: Discrete 0.06/s (15.000 fps)
Interval: Discrete 0.0335 (30.000 Tps)
Size: Discrete / 2005/6
Interval: Discrete 0.0275 (15.000 fps)
Size: Discrete 1.04/2768
T_{12}
Interval: Discrete 0.023 (20.000 fps)
Size: Discrete 1280y720
Interval: Discrete 0.067s (15.000 fps)
Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 1920x1080
Interval: Discrete 0.067s (15.000 fps)
Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 2592x1944
Interval: Discrete 0.067s (15.000 fps)
[1]: 'UYVY' (UYVY 4:2:2)
Size: Discrete 176x144
Interval: Discrete 0.067s (15.000 fps)
Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 320x240
Interval: Discrete 0.06/s (15.000 fps)
Cine Discrete 0.033s (30.000 fps)
STZE: DISCRETE 640X480
interval: Discrete 0.00/s (15.000 lps)

为了测试 CSI 接口,需要打开 weston 服务,在命令行下执行如下命令,可以将采集图 像显示于 LCD 屏:

[root@TWDZ-stm32mp157]:/#systemctl stop qtdesktop

[root@TWDZ-stm32mp157]:/#systemctl start weston@root.service

[root@TWDZ-stm32mp157]:/# gst-launch-1.0 v4l2src device=/dev/video0 ! "video/x-raw, format=YUY2, width=640, height=480, framerate=(fraction)60/1" ! waylandsink

5.9 网络测试

TW-STM32MP157-EVM 评估板有一路百兆以太网接口,使用标准的 RJ45 网口插座, 插座内带状态指示灯。

可使用 ifconfig 指令来显示或者配置网络, 查看网络信息。

[root@TWDZ-stm32mp157]:/# ifconfig



插上网线到以太网接口处可以看到如下信息,系统自动获取了 ip 同理。

[root@TWDZ-stm32mp157]:~# [13952.809184] stm32-dwmac 5800a000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx

[root@TWDZ-stm32mp157]:/# udhcpc -i eth0

[root@TWD	DZ-stm32mp157]:/# ifconfig
eth0	Link encap:Ethernet HWaddr 66:a9:22:26:bb:3b
S2507 E 3207 S	inet addr:192.168.0.123 Bcast:192.168.0.255 Mask:255.255.255.0
	UP BROADCAST RUNNING MULTICAST DYNAMIC MTU:1500 Metric:1
	RX packets:4256 errors:0 dropped:181 overruns:0 frame:0
	TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:1000
	RX bytes:353277 (344.9 KiB) TX bytes:9767 (9.5 KiB)
	Interrupt:55 Base address:0xa000
10	Link encap:Local Loopback
	inet addr:127.0.0.1 Mask:255.0.0.0
	UP LOOPBACK RUNNING MTU:65536 Metric:1
	RX packets:8 errors:0 dropped:0 overruns:0 frame:0
	TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:1000
	RX bytes:732 (732.0 B) TX bytes:732 (732.0 B)

如果对应网卡没有自动获取到 IP,请使用下面的指令获取。"-i"是指定网卡名称,如不指定,会使用默认会使用 eth0。

[root@TWDZ-stm32mp157]:/# ifconfig eth0 down // 关闭网口,网卡名字请根据实际情况 修改, down 表示关闭网口

[root@TWDZ-stm32mp157]:/# ifconfig eth0 up // 打开网口,网卡名字请根据实际情况 修改,up 表示打开网口

测试网口是否能上网,以访问 www.baidu.com 为例,执行如下命令,"-I"代表指定 网口, 不加"-I"则使用默认网卡(默认网卡指的是有网络接入的一端,如果两个网口都 有网络接入,则使用 eth0 作为默认网卡)。按"Ctrl+c"终止 ping 指令。百度的实际地 址根据网络运营商不同,访问的地址会不同。

[root@TWDZ-stm32mp157]:/# ping baidu.com

[root@TWDZ-stm32mp157]:/# ping baidu.com PING baidu.com (220.181.38.148) 56(84) bytes of data. 64 bytes from 220.181.38.148: icmp_seq=1 tt]=53 time=42.3 ms 64 bytes from 220.181.38.148: icmp_seq=2 tt]=53 time=42.4 ms 64 bytes from 220.181.38.148: icmp_seq=3 tt]=53 time=42.4 ms ^C64 bytes from 220.181.38.148: icmp_seq=4 tt]=53 time=42.2 ms --- baidu.com ping statistics ---4 packets transmitted, 4 received, 0% packet loss, time 15154ms rtt min/avg/max/mdev = 42.266/42.360/42.452/0.164 ms [root@TWDZ-stm32mp157]:/#

使用 route 命令查看网关后, 并 ping 网关。

[root@TWDZ-stm32mp157]:/# route

[root@TWDZ-str Kernel IP rout	n32mp157]:/# rou ting table	te		-42	30.4.9		
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	192.168.0.1	0.0.0.0	UG	0	0	0	eth0
192.168.0.0	×	255.255.255.0	U	0	0	0	eth0

由上可知网关为 gao. ke, 根据路由器不同, 网关可能不同。ping 网关可测试内网与 开发板连接是否正常。下面指令不加 "-I" 参数, 使用默认网卡。

[root@TWDZ-stm32mp157]:/# ping gao.ke

```
[root@TWDZ-stm32mp157]:/# ping gao.ke
PING g.17986.net (104.160.174.162) 56(84) bytes of data.
64 bytes from MAYALINZTIME.NET (104.160.174.162): icmp_seq=2 tt]=54 time=159 ms
64 bytes from 162.174.160.104.in-addr.arpa (104.160.174.162): icmp_seq=3 tt]=54
time=160 ms
64 bytes from 162.174.160.104.in-addr.arpa (104.160.174.162): icmp_seq=4 tt]=54
time=159 ms
^c
--- g.17986.net ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3019ms
rtt min/avg/max/mdev = 159.871/159.936/160.019/0.465 ms
```

5.10 音频测试

开始录音5秒钟,执行下面的指令:

[root@TWDZ-stm32mp157]:/# arecord -r 44100 -f S16 LE -d 5 record.wav

指令解释:

- (1) -r 44100: 采样率 44.1K
- (2) -f S16 LE: 以 S16 LE 格式采样
- (3) -d 10: 录音长度 10s
- (4) record. wav: 录音存生成的音频文件

播放录制的音频文件,执行下面的指令:

[root@TWDZ-stm32mp157]:/# aplay record.wav

5.11 TF 卡测试

将 TF 卡插入到开发板 TF 卡插槽中,Linux 操作系统会检测到 TF 卡,并在控制台 终端上打印 TF 卡的相关信息,例如:

[root@TWDZ-stm32mp157]:/# [15376.081750] mmc1: host does not support reading read-only

switch, assuming write-enable

[15376.093354] mmc1: new high speed SDHC card at address b368

[15376.101728] mmcblk1: mmc1:b368 SD08G 7.42 GiB

[15376.108947] mmcblk1: p1 p2

[15376.767721] EXT4-fs (mmcblk1p2): recovery complete

[15376.771141] EXT4-fs (mmcblk1p2): mounted filesystem with ordered data mode. Opts: (null)

[15376.799782] FAT-fs (mmcblk1p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.

TF 卡被挂载到/run/media/mmcblk1p1 目录上,为了将 TF 卡根目录上的 test 文件拷贝到/home 目录下,可执行如下命令:

[root@TWDZ-stm32mp157]:/#cp /run/media/mmcblk1p1/test /home

为了将/home 目录下的 test1 文件拷贝到 TF 卡根目录上,可执行如下命令: [root@TWDZ-stm32mp157]:/#cp /home/test1 /run/media/mmcblk1p1

此处假定 TF 卡被挂载在/run/media/mmcblk2p1 目录下。umount 会确保所有缓存的 数据都被正确的写入 TF 卡。在 umount 成功后,即可拔出 TF 卡。在调用 umount 前, 必须确保 TF 卡上的文件没有被其他程序所占用且用户当前的工作目录不在 TF 卡的挂载 目录上,否则调用 umount 会提示失败,如下所示:

umount: /run/media/mmcblk1p1/: target is busy (In some cases useful info about processes that use the device is found by lsof(8) or fuser(1).

5.12 U 盘使用

将格式为 FAT32 的 U 盘插入到开发板 USB HOST 接口上, Linux 操作系统会检测到 U 盘, 并在控制台终端上打印 U 盘的相关信息, 例如:

[root@TWDZ-stm32mp157]:~# usb 1-1.2: new high-speed USB device number 4 using ci_hdrc

usb-storage 1-1.2:1.0: USB Mass Storage device detected

scsi host0: usb-storage 1-1.2:1.0

scsi 0:0:0:0: Direct-Access SMI USB DISK 1100 PQ: 0 ANSI: 4

sd 0:0:0:0: [sda] 15769600 512-byte logical blocks: (8.07 GB/7.51 GiB)

sd 0:0:0:0: [sda] Write Protect is off

sd 0:0:0:0: [sda] No Caching mode page found

sd 0:0:0:0: [sda] Assuming drive cache: write through

sda: sda1

sd 0:0:0:0: [sda] Attached SCSI removable disk

FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.

在此例中,从打印信息可以看出 Linux 操作系统检测到一个容量为 8GB 的 U 盘,其 对应的设备名为 sda。 系统会在/etc 目录下为 U 盘的每个分区都生成一个目录,目录的名字为 sdxn (x 用 于区分不同的 U 盘,n 用于区分不同的分区,x=a、b、c······ n=1、2、3······)。U 盘的每 个分区就挂载在这些目录下。U 盘挂载成功后,即可对 U 盘进行文件查看、文件拷贝等操 作。以下以文件拷贝操作为例,介绍 U 盘的使用。

假定 U 盘被挂载到/run/media/sda1 目录上,为了将 U 盘根目录上的 test 文件拷 贝到/home 目录下,可执行如下命令:

[root@TWDZ-stm32mp157]:/# cp /run/media/sda1/test /home

为了将/home 目录下的 test1 文件拷贝到 U 盘根目录上,可执行如下命令:

[root@TWDZ-stm32mp157]:/# cp /home/test1 /run/media/sda1

此处假定 U 盘被挂载在/run/media/sda1 目录下。umount 会确保所有缓存的数据都 被正确的写入 U 盘。在 umount 成功后,即可拔出 U 盘。在调用 umount 前,必须确保 U 盘上的文件没有被其他程序所占用且用户当前的工作目录不在 U 盘的挂载目录上,否则调 用 umount 会提示失败,如下所示:

umount: /run/media/sda1/: target is busy(In some cases useful info about processes that use the device is found by lsof(8) or fuser(1).)

5.13 USB 鼠标与 USB 键盘使用

将 USB 鼠标插入到开发板 USB HOST 接口上, Linux 操作系统会检测到 USB 鼠标, 并在控制台终端上打印 USB 鼠标的相关信息,例如:

usb 1-1.2: new low-speed USB device number 4 using ci_hdrc

input: USB Optical Mouse as

/devices/soc0/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb1/1-1/1-1.2/1-1.2: 1.0/0003: 1BCF: 0007.0001/input/ input2

hid-generic 0003: 1BCF: 0007.0001: input: USB HID v1.10 Mouse [USB Optical Mouse] on usb-ci_hdrc.1-1.2/input0

将 USB 键盘插入到开发板 USB HOST 接口上, Linux 操作系统会检测到 USB 键盘, 并在控制台终端上打印 USB 键盘的相关信息,例如:

usb 1-1.2: new low-speed USB device number 6 using ci hdrc

input: USB USBKeykoard as

/devices/soc0/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb1/1-1/1-1.2/1-1.2 : 1.0/0003 : 1A2C: 0002.0004/input/ input5

hid-generic 0003: 1A2C: 0002.0004: input: USB HID v1.10 Keyboard [USB USBKeykoard] on usb-ci hdrc.1-1.2/input0

input: USB USBKeykoard as

/devices/soc0/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb1/1-1/1-1.2/1-1.2 : 1.1/0003 : 1A2C: 0002.0005/input/ input6

hid-generic 0003: 1A2C: 0002.0005: input: USB HID v1.10 Device [USB USBKeykoard] on usb-ci_hdrc.1-1.2/input1

5.14 LCD 背景亮度调节

LCD 屏幕的背光支持 8 级变化,亮度级数为 0~7。可以通过以下命令查看等级: [root@TWDZ-stm32mp157]:~#cat /sys/class/backlight/panel-backlight/max_brightness // 查 看 lcd 最大亮度等级

修改当前屏幕背光亮度等级,执行命令如下:

[root@TWDZ-stm32mp157]:~# echo 5 > /sys/class/backlight/panel-backlight/brightness

5.15 USB 接口测试

TW-AC6G-EVM 评估板有 2 路 USB2.0 HOST 接口 CN20,使用标准的双层 USB-A 插座。1 路 USB2.0 HOST 接口 CZ1 和 CN21 二选一使用,CZ1 使用标准的 PH2.0-4A 插座, CN21 使用侧插 USB-AF 插座, 方便用户扩展接口。

TW-AC6G-EVM 评估板有 1 路 Micro_USB 接口 CN38,使用标准的 Micro_USB 插座,该接口默认为 Device 接口。

TW-AC6G-EVM 评估板 USB/OTG 接口说明:

- ◆ USB_HOST1~USB_HOST3 为 HOST 模式, 默认为 HOST 模式;
- ◆ USB1 HOST 支持 HOST 模式; USB OTG1 支持切换为 HOST/DEVICE

以下测试根据使用的 U 盘的不同和实验环境不同,测试结果会有所差异。将 FAT32 格式 U 盘插到开发板 USB_HOST1~USB_HOST3/USB1_HOST 其中一个接口。 插入后会打印如下信息,可以从中看到 U 盘大小和挂载名,如下图所示:

i-platform
[16133.888659] usb-storage 2-1.5:1.0: USB Mass Storage device detected
[16133.905029] scsi host0: usb-storage 2-1.5:1.0
[16134.968808] scsi 0:0:0:0: Direct-Áccess Kingston DataTraveler 3.0 PQ: 0 ANSI: 6
[16134.978595] sd 0:0:0:0: Attached scsi generic sg0 type 0
[16134.985979] sd 0:0:0:0: [sda] 60437492 512-byte logical blocks: (30.9 GB/28.8 GiB)
[16134.998463] sd 0:0:0:0: [sda] Write Protect is off
[16135.002664] sd 0:0:0:0: [sda] write cache: disabled, read cache: enabled, doesn't support DF
OR FUA
[16135.055013] sda: sda1
[16135.069957] sd 0:0:0:0: [sda] Attached SCSI removable disk
<pre>[16135.359988] udevd[3499]: failed to execute '/etc/mount-usb.sh' '/etc/mount-usb.sh sda1': No</pre>
uch file or directory
[16135_611909] FAT-fs (sda1): volume was not properly unmounted. some data may be corrupt. Plea
e run fsck.

输入 df-h 查看 U 盘的挂载路径,可以看到下图 U 盘已经挂载在/run/media/下,挂载名为 sda1。

[root@TWDZ-stm]	32mp157]:~+	# df			
Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/root	1277484	1000592	193948	84%	/
devtmpfs	379092	4	379088	1%	/dev
tmpfs	445140	188	444952	1%	/run
tmpfs	445140	420	444720	1%	/var/volatile
/dev/mmcblk2p2	8887	8282	0	100%	/run/media/mmcblk2p2
/dev/mmcblk1p2	1426656	1410272	0	100%	/run/media/mmcblk1p2
/dev/mmcblk1p1	511720	406296	105424	80%	/run/media/mmcblk1p1
/dev/sda1	30203008	5567568	24635440	19%	/run/media/sda1

写速度测试:

输入 time dd if=/dev/zero of=/run/media/sda1/test bs=1024k count=100 conv=fdatasync



本次写 100MiB, 速度为 8.8MB/s。

5.16 触摸屏测试

测试触摸是否有反应

[root@TWDZ-stm32mp157]:~# hexdump /dev/input/event0

- ·			-	-		-		-
[root@T	NDZ-ST	m32m	0157]	:~# he	exdum	o /dev	/inpu	ut/ever
0000000	9966	5e10	d1f3	000d	0003	0035	018f	0000
0000010	9966	5e10	d1f3	000d	0003	0036	0100	0000
0000020	9966	5e10	d1f3	000d	0003	0000	018f	0000
0000030	9966	5e10	d1f3	000d	0003	0001	0100	0000
0000040	9966	5e10	d1f3	000d	0000	0000	0000	0000
0000050	9966	5e10	ca36	000e	0003	0035	018b	0000
0000060	9966	5e10	ca36	000e	0003	0036	00ff	0000
0000070	9966	5e10	ca36	000e	0003	0000	018b	0000
0000080	9966	5e10	ca36	000e	0003	0001	00ff	0000
0000090	9966	5e10	ca36	000e	0000	0000	0000	0000
00000a0	9966	5e10	f331	000e	0003	0035	0186	0000
00000b0	9966	5e10	f331	000e	0003	0000	0186	0000
00000c0	9966	5e10	f331	000e	0000	0000	0000	0000
00000d0	9966	5e10	1c39	000f	0003	0035	017f	0000
00000e0	9966	5e10	1c39	000f	0003	0036	00fe	0000
00000f0	9966	5e10	1c39	000f	0003	0000	017f	0000
0000100	9966	5e10	1c39	000f	0003	0001	00fe	0000
0000110	9966	5e10	1c39	000f	0000	0000	0000	0000
0000120	9967	5e10	0642	0000	0003	0035	0178	0000

5.17 CPU 温度

CPU 温度的高低与环境温度及 CPU 运行状况有关,例如跑一些大的应用,跑视频音频等,温度会高一些,建议不要用手去触摸 CPU,以免造成损坏。使用以下命令可以读出 CPU 温度。

cat /sys/class/hwmon/hwmon0/temp1_input

[root@TWDZ-stm32mp157]:~# cat /sys/class/hwmon/hwmon0/temp1_input 53127

5.18 CPU 主频

使用以下命令 cpufreq-info 测试主频。



由上图可得知,CPU 主频工作在 650MHZ, 调频模式为 "ondemand", 也可以通过以

下指令查看当前模式和当前的频率。

[root@TWDZ-stm32mp157]:~# cd /sys/devices/system/cpu/cpufreq/policy0/

[root@TWDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0# cat scaling cur freq 650000

[root@TWDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0# cat scaling max freq 650000

[root@TWDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0# cat scaling min freq 650000

[root@TWDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0# cat scaling governor

ondemand

root@TwDZ-stm32mp157]:/sys/devices/system/cpu/cpuFreq/policy0# cat scaling_cur_freq 00000 root@TWDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0# cat scaling_max_freq 300000 root@TWDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0# cat scaling_min_freq 00000 [root@TWDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0# cat scaling_governor ndemand oot@TwDZ-stm32mp157]:/sys/devices/system/cpu/cpufreq/policy0#

5.19 查看 CPU 信息

[root@TWDZ-stm32mp157]:~# cat /proc/cpuinfo

Ē.		
[root@TWDZ-stm3]	2m	pls/j:~# cat /proc/cpuinto
processor		0
model name		ARMV7 Processor rev 5 (V/T)
BOGOMIPS		
Features		half thumb fastmult vfp edsp thumbee neon vfpv3 tis vfpv4 idiva idivt vfpd32 ipae evtstr
m		
CPU implementer	:	0x41
CPU architectur	e:	1
CPU variant		0x0
CPU part		0xc07
CPU revision		5
processor		
model name		ARMV7 Processor rev 5 (v71)
BOGOMIPS		24.00
Features	-	half thumb fastmult vfp edsp thumbee neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstr
m		
CPU implementer	15	0x41
CPU architectur	e:	7 100
CPU variant		0x0
CPU part		0xc07
CPU revision		5
Hardware		STM32 (Device Tree Support)
Revision		0000
Serial	:	001E00263030510332313933
[root@TWDZ-stm3]	2 m	p157]:~#

5.20 查看内存信息

[root@TWDZ-stm32mp157]:~# cat /proc/meminfo

Iroot@TwD2-stm32mp157]:-# cat /proc/meminfo MemTotal: 953304 kB MemEree: 747736 kB MemErree: 747736 kB MemErree: 747736 kB MemErree: 747736 kB MemErree: 747736 kB SwapCached: 0 kB Active: 67064 kB Inactive: 83512 kB Active(anon): 85352 kB Inactive(anon): 85352 kB Inactive(anon): 8536 kB Active(file): 74976 kB Unevictable: 0 kB Mlocked: 0 kB Unevictable: 0 kB Unevictable: 0 kB SwapTotal: 691164 kB LowTotal: 691164 kB LowTotal: 691164 kB SwapFree: 0 kB Dirty: 4 kB Writeback: 0 kB Shmen: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB Sreclaimable: 17548 kB Surrelaim: 242766 kB Writeback: 0 k						
MemTotal: 953304 kg MemFree: 747736 kg MemAvailable: 850444 kg Buffers: 14304 kg Cached: 98708 kg Swapcached: 0 kg Active: 67064 kg Inactive: 83512 kg Active: 67064 kg Inactive: 83512 kg Active(file): 74976 kg Unevictable: 0 kg Mlocked: 0 kg Morotal: 621164 kg Lowrotal: 691164 kg SwapTotal: 0 kg SwapTotal: 0 kg SwapFree: 0 kg Dirty: 48 kg Writeback: 0 kg AnonPages: 37560 kg Shmem: 9228 kg KReclaimable: 17548 kg Slab: 42304 kg Surreclaim: 24756 kg Kernelstack: 1000 kg Pagerables: 1264 kg Vmalloctotal: 245760 kg Vmalloctotal: 245760 kg Vmalloctounk: 0 kg <	[root@TWDZ-stm32	mp157]:~-	# cat	/proc/meminfo		
MemEree: 747736 kB MemAvailable: 850444 kB Buffers: 14304 kB Cached: 98708 kB SwapCached: 0 kB Active: 67064 kB Inactive: 83512 kB Active(anon): 38252 kB Inactive(anon): 8356 kB Active(file): 74976 kB Unevictable: 0 kB Molcked: 0 kB Highrotal: 262140 kB Highree: 119300 kB LowFreet: 628436 kB SwapTotal: 0 kB Dirty: 48 kB Writeback: 0 kB Symped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Supreclaim: 24756 kB Vernetale: 0 kB PageTables: 1264 kB Supreclaim: 24756 kB WritebackTmp: 0 kB CommitLimit: 47652 kB CommitLimit: 47652 kB CommitLimit: 47652 kB CommitLimit: 47652 kB </td <td>MemTotal:</td> <td>953304</td> <td>kв</td> <td></td> <td></td> <td></td>	MemTotal:	953304	kв			
Memavailable: 850444 KB Buffers: 14304 KB Swapcached: 0 KB Swapcached: 0 KB Active: 67064 KB Inactive: 83512 KB Active: 67064 KB Inactive: 83512 KB Active(anon): 38252 KB Inactive(file): 28812 KB Unevictable: 0 KB Mlocked: 0 KB Mlocked: 0 KB LowTotal: 691164 KB LowFree: 628436 KB SwapTotal: 0 KB SwapTotal: 0 KB SwapFree: 0 KB Shmem: 9228 KB Shmem: 9228 KB Superable: 17548 KB Superable: 1264 KB Superable: 0 KB Sounce: 0 KB Bounce: 0 KB <	MemFree:	747736	kв			
Buffers: 14304 kB Cached: 98708 kB SwapCached: 0 kB Active: 67064 kB Inactive: 83512 kB Active(anon): 38522 kB Inactive(anon): 38522 kB Inactive(file): 74976 kB Unevictable: 0 kB HighFreal: 262140 kB HighFree: 119300 kB LowFree: 628436 kB SwapTotal: 0 kB SwapFree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Supreclaim: 24756 kB Sup	MemAvailable:	850444	kв			
Cached: 98708 kB SwapCached: 0 kB Active: 67064 kB Inactive: 83512 kB Active(anon): 38252 kB Inactive(anon): 8536 kB Active(file): 74976 kB Unevictable: 0 kB Mlocked: 0 kB HighTotal: 262140 kB HighTree: 119300 kB LowFree: 628436 kB SwapTotal: 0 kB SwapTotal: 0 kB SwapFree: 0 kB SwapFree: 0 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB Sunreclaim: 24756 kB Sunreclaim:	Buffers:	14304	kв			
SwapCached: 0 kB Active: 67064 kB Inactive: 83512 kB Active(anon): 38252 kB Inactive(file): 2812 kB Inactive(file): 74976 kB Unevictable: 0 kB Mlocked: 0 kB LowTotal: 691164 kB LowTotal: 691164 kB SwapTotal: 0 kB SwapTotal: 0 kB SwapTotal: 0 kB Shmem: 9228 kB KReclaimable: 17548 kB Shaperel: 0 kB SpaceTaimable: 17548 kB Supreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Committed_AS: 378448 kB VmallocTotal: 245760 kB Vmalloctouck: 0 kB Committed_AS: 378448 kB Vmal	Cached:	98708	kв			
Active: 67064 kB Inactive: 83512 kB Active(anon): 38252 kB Inactive(anon): 8536 kB Active(file): 28812 kB Inactive(file): 74976 kB Unevictable: 0 kB Mlocked: 0 kB Highrotal: 262140 kB Highrotal: 262140 kB LowTree: 628436 kB SwapTree: 0 kB SwapTree: 0 kB Dirty: 48 kB Writeback: 0 kB Shmen: 9228 kB Shmen: 9228 kB Slab: 42304 kB Slab: 42304 kB Slab: 42304 kB Slab: 42304 kB Slab: 17548 kB Slab: 42304 kB Slab: 1256 kB Kerolaimable: 17548 kB Slab: 42304 kB Slab: 42304 kB Slowreclaim: 24756 kB Kernelstack: 1000 kB Pagerables: 1264 kB VritebackTmp: 0 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmalloctotal: 131072 kB Cmarotal: 115352 kB Cmarotal: 115352 kB	SwapCached:	0	kв			
<pre>Inactive: 83512 kB Active(anon): 38252 kB Inactive(anon): 83536 kB Active(file): 28812 kB Inactive(file): 74976 kB Unevictable: 0 kB HighTotal: 262140 kB HighTotal: 262140 kB HighTotal: 262140 kB LowTree: 628436 kB SwapTotal: 0 kB SwapTotal: 0 kB SwapTree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmen: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Sunreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB SFS_Unstable: 0 kB Bounce: 0 kB Bounce: 0 kB Bounce: 0 kB MoritebacKTmp: 0 kB Committed_AS: 378448 kB VmallocTotal: 245760 kB VmallocChunk: 0 kB Percpu: 400 kB Percpu: 400 kB Percpu: 400 kB Percpu: 400 kB PageTables: 1264 kB</pre>	Active:	67064	kв			
Active(anon): 38252 kB Inactive(anon): 8536 kB Active(file): 28812 kB Inactive(file): 74976 kB Unevictable: 0 kB Hightrotal: 262140 kB Hightree: 119300 kB LowTotal: 691164 kB LowTotal: 691164 kB LowTree: 628436 kB SwapTrotal: 0 kB SwapTrotal: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Sureclaimable: 17548 kB Sureclaimable: 17548 kB Sureclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB WritebacKTmp: 0 kB Bounce: 0 kB WritebacKTmp: 0 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB Cmartotal: 131072 kB Cmartotal: 131072 kB	Inactive:	83512	kв			
<pre>Inactive(anon): 8536 kB Active(file): 28812 kB Inactive(file): 74976 kB Unevictable: 0 kB Mlocked: 0 kB Highrotal: 262140 kB Highrotal: 262140 kB LowTotal: 691164 kB LowTotal: 691164 kB LowFree: 628436 kB SwapFree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB Slab: 42304 kB Slab: 42304 kB SReclaimable: 17548 kB Slab: 12548 kB Slab: 42304 kB SReclaimable: 17548 kB Slab: 42304 kB Slab: 42304 kB SReclaimable: 17548 kB Slab: 42304 k</pre>	Active(anon):	38252	kв			
Active(file): 28812 kB Inactive(file): 74976 kB Unevictable: 0 kB Mlocked: 0 kB HighTotal: 262140 kB HighTotal: 262140 kB LowTotal: 691164 kB LowFree: 628436 kB SwapTotal: 0 kB SwapTotal: 0 kB Joirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Sunreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB Bounce: 0 kB Bounce: 0 kB Committed_As: 378448 kB Vmalloctotal: 245760 kB Vmalloctotal: 245760 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTotal: 131072 kB	Inactive(anon):	8536	kв			
Inactive(file): 74976 kB Unevictable: 0 kB Mlocked: 0 kB HighTotal: 262140 kB HighFree: 119300 kB LowTotal: 691164 kB LowTotal: 691164 kB LowTotal: 691164 kB LowTotal: 691164 kB SwapTotal: 0 kB SwapTotal: 0 kB Writeback: 0 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 4230 kB Seclaimable: 17548 kB SUnreclaim: 24736 kB Kernelstack: 1000 kB Pagetables: 1264 kB NFS_Unstable: 0 kB Gommitted_AS: 378448 kB Vmalloctotal: 245760 kB Vmalloctotal: 245760 kB Vmalloctotal: 245760 kB Vmalloctotal: 2604 kB Vmalloctotal: 245760 kB Vmalloctotal: 2604 kB Vmalloctourk: 0 kB Committed_AS	Active(file):	28812	kв			
Unevictable: 0 kB Mlocked: 0 kB Highrotal: 262140 kB Highrotal: 262140 kB LowTree: 119300 kB LowTree: 628436 kB SwapTree: 0 kB SwapTree: 0 kB Dirty: 48 kB writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmen: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB Slab: 42304 kB SReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Slurreclaim: 24756 kB KernelStack: 1000 kB Pagerables: 1264 kB NF5_Unstable: 0 kB Bounce: 0 kB Bounce: 0 kB Committed_AS: 378448 kB VmallocUsed: 2604 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTotal: 131072 kB	Inactive(file):	74976	kв			
Mlocked: 0 kB Highrotal: 262140 kB Highree: 119300 kB LowTotal: 691164 kB LowFree: 628436 kB Swaprotal: 0 kB Swaprotal: 0 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Sunreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB WritebacKTmp: 0 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTree: 12852 kB	Unevictable:	0	kв			
HighTotal: 262140 kB HighFree: 119300 kB LowTotal: 691164 kB LowFree: 628436 kB SwapFree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB Slab: 42304 kB Sseclaimable: 17548 kB Surreclaim: 24756 kB Kernelstack: 1000 kB PageTables: 1264 kB NF5_unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB Committed_AS: 378448 kB Vmalloctual: 245760 kB Vmalloctual: 245760 kB Vmalloctual: 2604 kB Percpu: 400 kB Cmarotal: 131072 kB Cmarree: 115352 kB	Mlocked:	0	kв			
HiğhFree: 119300 kB LowTrotal: 691164 kB LowTrotal: 691164 kB SwapTree: 0 kB SwapFree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmen: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Surreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NF5_Unstable: 0 kB Bounce: 0 kB Bounce: 0 kB WritebacKTmp: 0 kB Committed_AS: 378448 kB VmallocUsed: 2604 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTree: 115352 kB	HighTotal:	262140	kв			
LowTotal: 691164 kB LowTree: 628436 kB SwapTotal: 0 kB SwapTree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Surreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NF5_Unstable: 0 kB Bounce: 0 kB Bounce: 0 kB WritebacKTmp: 0 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTree: 115352 kB CmaTree: 115352 kB	HighFree:	119300	kв			
LowFree: 628436 kB SwapTrea: 0 kB Dirty: 48 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmen: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB Slab: 42304 kB Slab: 1256 kB Kernelstack: 1000 kB PageTables: 1264 kB NF5_unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB Committimit: 476652 kB Committed_AS: 378448 kB Vmalloctual: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTree: 115352 kB CmaTree: 115352 kB	LowTotal:	691164	kв			
SwapTotal: 0 kB SwapFree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB AnonPages: 44240 kB Shmen: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Sunreclaim: 24756 kB Kernelstack: 1000 kB PageTables: 1264 kB Sounce: 0 kB Bounce: 0 kB Committed_AS: 378448 kB Vmallocused: 2604 kB Vmallocchunk: 0 kB Percpu: 400 kB Cmartal: 131072 kB Cmartal: 131072 kB	LowFree:	628436	kв			
SwapFree: 0 kB Dirty: 48 kB Writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmen: 928 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Surreclaim: 24756 kB Kernelstack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Gommitted_AS: 378448 kB Vmalloctotal: 245760 kB Vmalloctotal: 245760 kB Vmalloctotal: 0 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocthunk: 0 kB Cmartal: 131072 kB Cmartal: 131072 kB Cmartal: 131072 kB	SwapTotal:	0	kв			
Dirty: 48 kB writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmen: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB SUnreclain: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTree: 115352 kB CmaTree: 115352 kB	SwapFree:	0	kв			
<pre>writeback: 0 kB AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Sunreclaim: 24756 kB Kernelstack: 1000 kB Pagerables: 1264 kB NF5_unstable: 0 kB Bounce: 0 kB Bounce: 0 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTree: 115352 kB</pre>	Dirty:	48	kв			
AnonPages: 37560 kB Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB Sunreclaim: 24756 kB Sunreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB CommitLimit: 476652 kB Committed_AS: 378448 kB Vmallocused: 2604 kB Vmallocused: 2604 kB Vmallocchunk: 0 kB CmaTotal: 131072 kB CmaTotal: 131072 kB CmaTotal: 15352 kB	Writeback:	0	kв			
Mapped: 44240 kB Shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB SUnreclaim: 24756 kB Kernelstack: 1000 kB PageTables: 1264 kB NFS_unstable: 0 kB Bounce: 0 kB Committed_AS: 378448 kB Vmalloctback: 0 kB Vmalloccbunk: 0 kB Percpu: 400 kB Cmartal: 131072 kB Cmarte: 115352 kB	AnonPages:	37560	kв			
shmem: 9228 kB KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB Sunreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTrtal: 131072 kB CmaTrtal: 131532 kB	Mapped:	44240	kв			
KReclaimable: 17548 kB Slab: 42304 kB SReclaimable: 17548 kB SUnreclaim: 24756 kB Surreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NF5_Unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB CommitLimit: 476652 kB CommitLed_AS: 378448 kB Vmallocused: 2604 kB Vmallocchunk: 0 kB Comartotal: 131072 kB Cmarree: 115352 kB Cmarree: 115352 kB	Shmem:	9228	kв			
Slab: 42304 kB SReclaimable: 17548 kB SUnreclaim: 24756 kB Kernelstack: 1000 kB PageTables: 1264 kB NF5_unstable: 0 kB Bounce: 0 kB CommitLimit: 476652 kB Committed_AS: 378448 kB Vmalloctback 0 kB Vmallocchunk: 0 kB Percpu: 400 kB Cmarree: 115352 kB Cmarree: 115352 kB	KReclaimable:	17548	kв			
SReclaimable: 17548 kB SUnreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB commitLimit: 476652 kB CommitLed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTree: 115352 kB EncordFWDZ-sTm32mp1571: *#	slab:	42304	kв			
SUnreclaim: 24756 kB KernelStack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB CommitLimit: 476652 kB Committed_AS: 378448 kB VmallocChal: 245760 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTotal: 115352 kB	SReclaimable:	17548	kв			
Kernelstack: 1000 kB PageTables: 1264 kB NFS_Unstable: 0 kB Bounce: 0 kB CommitLimit: 476652 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB Cmartal: 113072 kB Cmarte: 113352 kB	SUnreclaim:	24756	kв			
PageTables: 1264 kB NF5_Unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB CommitLimit: 476652 kB CommitLed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTotal: 1131072 kB CmaTotal: 115352 kB	KernelStack:	1000	kв			
NFŠ_Unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB CommitLimit: 476652 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTotal: 115352 kB	PageTables:	1264	kВ			
Bounce: 0 kB WritebackTmp: 0 kB CommitLimit: 476652 kB Committed_AS: 378448 kB Vmalloctotal: 245760 kB Vmallocchunk: 0 kB Percpu: 400 kB Cmartal: 131072 kB Cmarte: 115352 kB	NFS_Unstable:	0	kв			
<pre>writebackTmp: 0 kB CommitLimit: 476652 kB CommitLed_AS: 378448 kB VmallocTotal: 245760 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTree: 115352 kB Froot@WDZ=Stm32mm1571:c##</pre>	Bounce:	0	kв			
CommitLimit: 476652 kB Committed_AS: 378448 kB VmallocUsed: 245760 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTree: 115352 kB [root@WDZ=Stm32mn157]:~#	WritebackTmp:	0	kв			
Committed_AS: 378448 kB VmallocTotal: 245760 kB VmallocCused: 2604 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTrdal: 131072 kB CmaFree: 115352 kB [root@WDZ=stm32mm157]:~#	CommitLimit:	476652	kв			
VmallocTotal: 245760 kB VmallocUsed: 2604 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTree: 115352 kB [root@WDZ=stm32mm157]:~#	Committed_A5:	378448	kв			
VmallocUsed: 2604 kB VmallocChunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaTree: 115352 kB [root@WDZ-stm32mp157]:~#	VmallocTotal:	245760	kв			
Vmallocchunk: 0 kB Percpu: 400 kB CmaTotal: 131072 kB CmaFree: 115352 kB [root@WDZ-stm32mp157]:~#	VmallocUsed:	2604	kв			
Percpu: 400 kB CmaTotal: 131072 kB CmaFree: 115352 kB [rootGTWDZ-stm32mp157]:~#	VmallocChunk:	0	kв			
CmaTotal: 131072 kB CmaFree: 115352 kB [root@vDZ-stm32mp157]:~#	Percpu:	400	kВ			
CmaFree: 115352 kB [root@TWDZ-stm32mp157];~#	CmaTotal:	131072	kв			
[root@TWDZ-stm32mp157]:~#	CmaFree:	115352	kв			
	[root@TWDZ-stm32	mp157]:~-	¥.			

6. 环境搭建

6.1 安装虚拟机软件 VMware

安装 Ubuntu 的前提是什么? 需要一个虚拟机 VMware 去启动它,虚拟机顾名思义就 是虚拟出来的一个机器,然后在这个机器上安装任何你想要的系统,相当于在克隆出一个 你的电脑,这样的话,主机上运行 Window 系统,当我们需要用到 Ubuntu 的话,打开安装 有 Ubuntu 系统的虚拟机就可以了。

Vmware Workstation 软件可以在 Wmware 官网下载,下载地址: <u>https:</u>//www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html。下载最新版的 Vmware Workstation Pro 15。

VMware 虚拟机的版本高低可能会导致 Ubuntu 无法正常开启。需要去修改光盘资料所提供 Ubuntu 的 Ubuntu 64 位.vmx 文件。

```
▶ 本地磁盘 (G:) ▶ 压缩包 ▶ Ubuntu16.04 64 位
```

称	修改日期	类型	大小
autoinst.flp	2020/5/7 8:31	FLP 文件	1,440 KB
🛃 autoinst.iso	2020/5/7 8:31	光盘映像文件	44,256 KB
📄 Ubuntu 64 位.nvram	2020/6/2 17:28	VMware 虚拟机	9 KB
불 Ubuntu 64 位.vmdk	2020/6/2 17:28	VMware <u>虚</u> 拟磁	24,497,40
👌 Ubuntu 64 位.vmsd	2020/5/7 8:31	VMware 快照元	0 KB
i Ubuntu 64 位.vmx	2020/6/2 17:28	VMX 文件	3 KB
👌 Ubuntu 64 位.vn	2020/5/7 8:31	VMware 组成员	1 KB
vmware.log	2020/6/2 17:28	文本文档	236 KB
vmware-0.log	2020/6/1 8:39	文本文档	227 KB
🗋 vmware-1.log	2020/5/30 13:42	文本文档	236 KB
] vmware-2.log	2020/5/27 17:04 主他软件打开修改	文本文档	282 KB

通过修改.vmx 文件里面的相应参数,如下图所示。



安装完毕后,打开 Vmware Workstation 软件,打开虚拟机,找到虚拟机解压的路径。

文件①	编辑(E)	查看(⊻)	虚拟机(<u>M</u>)	选项卡(1)	帮助(出)	•	ф,	₽	4	₽		R	>_	
命主页	x													

WORKSTATION 15.5 PRO



Talowe。广州眺望电子科技有限公司

用户手册

						ř.		
							•	٥
名称		修改日期	类型	大小				
) 564d6be8-4903-99	c0-fee4-6089b98b	2020/6/2 15:48	文件夹					
📙 Ubuntu 64 位.vmdk	c.lck	2020/6/2 15:48	文件夹					
Ubuntu 64 位.vmx.l	ck	2020/6/2 17:23	文件夹					
🖞 Ubuntu 64 位.vmx		2020/6/2 15:48	VMware 虚拟植	Л	3 KE	3		
ป): Ubuntu 64 位.vmx					•	所有支持的文	4	
						打开(0)		取消
打开虚拟机,这	进入系统							
打开虚拟机,这 Ubuntu 64 伯 开启此虚拟机 编辑虚拟机设置	进入系统 立							
打开虚拟机, Ubuntu 64 (开启此虚拟机 编辑虚拟机设置	进入系统 立							
打开虚拟机, Ubuntu 64 (1 · 开启此虚拟机 · 编辑虚拟机设置 · 编辑	进入系统 立							
打开虚拟机, Ubuntu 64 (1 · · · · · · · · · ·	进入系统 立 4 GB 2							
打开虚拟机, Ubuntu 64 位 开启此虚拟机 编辑虚拟机设置 译备 国内存 处理器 通母 (SCSD	进入系统 立 4 GB 2 80 GB							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 2备 回内存 处理器 2硬盘 (SCSI) 2 (CD/DVD 2 (SATA)	进入系统 ↓ <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 2 4 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	进入系统 ↓ <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 2备 目内存 处理器 2硬盘 (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA)	 进入系统 立 4 GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 A 							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 编辑虚拟机设置 经备 目内存 处理器 2硬盘 (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA)	 掛入系统 ☆ 4 GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 a 桥接模式(自动) 							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 备 同内存 处理器 @硬盘 (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA) 0 St 控制器	 							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 经备 内存 处理器 强硬盘 (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA)	 井入系统 立 4 GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 D: 正在使用文件 a 桥接模式(自动) 存在 自动検测 							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 经备 同内存 处理器 @硬盘 (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA) 0 St 控制器 0 声卡	 井入系统 立 4 GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 A 桥接模式(自动) 存在 自动检测 存在 							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编编虚拟机设置 经备 回内存 处理器 @硬盘 (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA) 1 软盘 0 USB 控制器 0 声卡	掛入系统 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓							
打开虚拟机, 打开虚拟机, 研启此虚拟机 编辑虚拟机设置 2 2 3 2 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4	进入系统 ↓ 人系统 ↓ GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 D: 正在使用文件 A 桥接模式 (自动) 存在 自动检测 存在 自动检测							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 经备 同内存 处理器 @硬盘 (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA) 0 St 控制器 0 声卡 0 JID 机 0 显示器	掛入系统 4 GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 D: 正在使用文件 a 桥接模式(自动) 存在 自动检测 存在 自动检测							
打开虚拟机, 打开虚拟机, 研启此虚拟机 编辑虚拟机设置 2 2 3 2 4 9 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1	进入系统 ↓ 人系统 ↓ 4 GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 D: 市在使用文件 a 桥接模式 (自动) 存在 自动检测 存在 自动检测 若法。							
打开虚拟机, 该 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 2 4 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	掛入系统 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓							
打开虚拟机, 近 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 译备 同内存 处理器 @ (SCSI) 0 CD/DVD 2 (SATA) 0 CD/DVD 2 (SATA) 0 CD/DVD (SATA) 0 CD/DVD (SATA) 0 SB 控制器 0 声卡 0 J显示器 插述 此处键入对该虚拟机的描	掛入系统 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	▼ 虚拟机详细	信息					
打开虚拟机, 该 Ubuntu 64 (开启此虚拟机 编辑虚拟机设置 译备 9内存 处理器 3硬盘 (SCSI)) CD/DVD 2 (SATA)) CD/DVD 2 (SATA)) CD/DVD (SATA)] 软盘 9 网络适配器 1 USB 控制器 9 声卡 9 打印机 1 显示器 1 述 此处键入对该虚拟机的描	 井入系统 立 4 GB 2 80 GB 正在使用文件 D: 正在使用文件 D: 正在使用文件 A 市接模式(自动) 存在 自动检测 存在 自动检测 述。 	▼ 虚拟机详细 状态: E 配雪女件: G	信息	6 04 64 (X)		64 ft umu		

6.2 交叉工具链

光盘资料中提供的 Ubuntu 已经把交叉工具链的环境搭建好了,用户可以直接进行使用。

Ubuntu 版本	16.04
交叉工具链版本	arm-none-linux-gnueabihf-gcc 9.2.1
交叉工具链的所在	/home/twdz/MP1/arm-linux/gcc-arm-9.2-2019.12-x86_64-arm-n
路径	one-linux-gnueabihf/bin

TW-STM32MP157 开发板使用 arm-none-linux-gnueabihf-gcc 9.2.1 版本交叉编译器,

其 安 装 包 为 gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf.tar.xz 可 从 **"TW-STM32MP157** 光盘资料\2.软件开发参考资料\1.编译工具"中直接获取。

可以简单的测试一下交叉编译器是否可以正常运行。方法如下:

在命令行下执行:

\$source

/home/twdz/MP1/arm-linux/gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf/bin

\$arm-none-linux-gnueabihf-gcc -v

如果能显示编译器的版本信息,则表明编译器已经可以正常运行了,如下图所示:

Using built-in specs.
COLLECT GCC=arm-none-linux-anueabihf-acc
COLLECT LTO WRAPPER=/home/twdz/MP1/arm-linux/acc-arm-9.2-2019.12-x86 64-arm-none-linux-an
ueabihf/bin//libexec/gc/arm-none-linux-gnueabihf/9.2.1/lto-wrapper
Target: arm-none-linux-gnueabihf
Configured with: /tmp/dgboter/bbs/rhev-vm7rhe6x86 64/buildbot/rhe6x86 64arm-none-linu
x-gnueabihf/build/src/gcc/configuretarget=arm-none-linux-gnueabihfprefix=with-sy
sroot=/arm-none-linux-gnueabihf/libcwith-build-sysroot=/tmp/dgboter/bbs/rhev-vm7rhe6
x86_64/buildbot/rhe6x86_64arm-none-linux-gnueabihf/build/build-arm-none-linux-gnueabihf
/install//arm-none-linux-gnueabihf/libcwith-bugurl=https://bugs.linaro.org/enable-g
nu-indirect-functionenable-shareddisable-libsspdisable-libmudflapenable-check
<pre>ing=releaseenable-languages=c,c++,fortranwith-gmp=/tmp/dgboter/bbs/rhev-vm7rhe6x8</pre>
6_64/buildbot/rhe6x86_64arm-none-linux-gnueabihf/build/build-arm-none-linux-gnueabihf/h
ost-tools with-mpfr=/tmp/dgboter/bbs/rhev-vm7rhe6x86_64/buildbot/rhe6x86_64arm-none
-linux-gnueabihf/build/build-arm-none-linux-gnueabihf/host-toolswith-mpc=/tmp/dgboter/
bbs/rhev-vm7rhe6x86_64/buildbot/rhe6x86_64arm-none-linux-gnueabihf/build/build-arm-no
ne-linux-gnueabihf/host-toolswith-isl=/tmp/dgboter/bbs/rhev-vm7rhe6x86_64/buildbot/r
he6x86_64arm-none-linux-gnueabihf/build/build-arm-none-linux-gnueabihf/host-toolswit
h-arch=armv7-awith-fpu=neonwith-float=hardwith-mode=thumbwith-arch=armv7-a
with-pkgversion='GNU Toolchain for the A-profile Architecture 9.2-2019.12 (arm-9.10)'
Thread model: posix
gcc version 9.2.1 20191025 (GNU Toolchain for the A-profile Architecture 9.2-2019.12 (arm

6.3 编译 Helloworld 源程序

在 Linux 宿主机任意目录下创建一个 HelloWorld 文件夹,用于存放 helloworld.c 源 文件。此处假定在/home/twdz/demo 下创建 HelloWorld 文件夹。执行的命令如下:

\$mkdir /home/twdz/demo

\$mkdir /home/twdz/demo/HelloWorld

\$cd /home/twdz/demo/HelloWorld

```
在该目录下新建一个 helloworld.c 的源文件,其内容如下:
```

#include<stdio.h>

int main()

{

```
Printf( "helloworld!\n" );
```

Return 0;

}

保存退出后,进入源码所在目录,在命令行下执行如下命令编译程序:

arm-none-linux-gnueabihf-gcc – o helloworld helloworld.c

将交叉编译出来的执行文件 helloworld 下载到开发板上运行。

~#chmod a+x helloworld

~# ./helloworld

```
helloworld
```

从执行结果可以看出,在终端上已经打印出了"Hello World"字符串。

6.4 编译 Uboot

uboot 的顶层 Makefile 中我们已经定义了 ARCH 和 CROSS_COMPILE 的值,所以后面 输入指令的时候可以简化。

通过输入"make menuconfig"来打开图形化配置界面

```
😑 🗊 twdz@ubuntu: ~/MP1/Uboot
.config - U-Boot 2020.01-stm32mp-r1 Configuration
                      U-Boot 2020.01-stm32mp-r1 Configuration
    Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
    submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in []
              Architecture select (ARM architecture) --->
            ARM architecture --->
              General setup --->
              Boot images
                             --->
              API
                    --->
              Boot timing --->
              Boot media --->
         (2) delay in seconds before automatically booting
          *] Enable boot arguments
         ()
                Boot arguments
           <Select>
                          < Exit >
                                         < Help >
                                                        < Save >
                                                                      < Load >
```

通过输入"./build all.sh"执行完后源根目录会出现以下新编译出来的文件。

twdz@ubuntu:~/MP	1/Uboot\$ l		(heat of a
api arch board build_all.sh build_config.sh cmd common config.mk configs CONTRIBUTING.md disk twdz@ubuntu:~/MP	doc drivers dts env examples fs include Kbuild Kconfig lib Licenses 1/Uboot\$	MAINTAINERS Makefile net post README scripts System.map test tools u-boot u-boot.bin	u-boot.cfg u-boot.cfg.configs u-boot.dtb u-boot.dtb.bin u-boot.lds u-boot.map u-boot.nodtb.bin u-boot.srec u-boot.stm32 u-boot.stm32.log u-boot.sym

通过输入"./build_config.sh"指令可以恢复默认配置。

6.5 编译内核

内核源码的顶层提供了各种编译执行脚本,用户可以根据不同的需求去执行相应的脚本,得到目标固件。

shell 脚本	说明	固件路径				
	护汉山坊吨佈	编译出来的目标文件				
built-all-tw.sh	新年内核联修、 反奋 动	源码顶层的 arch/arm/boot/uImage				
	ניאר	源码顶层的 arch/arm/boot/dts/stm32mp157a-tw.dtb				
		需根据在内核配置编译的模块的路径,在源码顶层的				
built-modules.sh	编译内核模块	driver/目录下相应的目录,可参照 "错误!未找到引用				
		源。"这章节				
huilt manuagefie sh	使用 menuconfig 配					
built-menuconiig.sn	置内核					

6.5.1 内核源码简介

Linux 内核源码很复杂,包含多级目录,形成一个庞大的树状结构,通常称为 Linux 源码目录树。TW-STM32MP157-EVM 开发板使用 Linux5.4.31 内核版本,本节以该版本 为例介绍 Linux 源码的目录结构:

目录	说明
arch	包含各体系结构特定的代码,如 arm、x86、ia64、mips 等,在每个体系结构目录下通常都有:
	-boot 内核需要的特定平台代码
	一kernel 体系结构特有的代码
	-lib 通用函数在特定体系结构的实现
	一math-emu 模拟 FPU 的代码,在 ARM 中,使用 mach-xxx 代替
	一mm 特定体系结构的内存管理实现

	一include 特定体系的头文件
block	存放块设备相关代码
crypto	存放加密、压缩、CRC 校验等算法相关代码
Docu	存放相关说明文档,很多实用文档,包括驱动编写等
menta	
tion	
driver	存放 Linux 内核设备驱动程序源码。驱动源码在 Linux 内核源码中站了很大比例,常见外设
s	几乎都有可参考源码,对驱动开发而言,该目录非常重要。该目录包含众多驱动,目录按照设
	备类别进行分类,如 char、block、input、i2c、spi、pci、usb 等
firmw	存放处理器相关的一些特殊固件
are	
fs	存放所有文件系统代码,如 fat、ext2、ext3、ext4、ubifs、nfs、sysfs 等
includ	存放内核所需、与平台无关的头文件,与平台相关的头文件已经被移动到 arch 平台的 include
e	目录,如 ARM 的头文件目录 <arch arm="" asm="" include=""></arch>
init	包含内核初始化代码
ipc	存放进程间通信代码
kernel	包含 Linux 内核管理代码
lib	库文件代码实现
mm	存放内存管理代码
net	存放网络相关代码
sampl	存放提供的一些内核编程范例
es	
srcipts	存放一些脚本文件,如 menuconfig 脚本
securi	存放系统安全性相关代码
ty	
sound	存放声音、声卡相关驱动
tools	编译过程中一些主机必要工具
usr	cpio 相关实现
virt	内核虚拟机 KVM

6.5.2 内核配置

1.内核 Makefile 文件

源码目录树顶层 Makefile 是整个内核源码管理的入口,对整个内核的源码编译起着 决定性作用。编译内核时,顶层 Makefile 会按规则递归遍历内核源码的所有子目录下的 Makefile 文件,完成各子目录下内核模块的编译。

在内核源码的子目录中,几乎每个子目录都有相应的 Makefile 文件,管理着对应目录下的代码,对该目录的文件或者子目录的编译进行控制。Makefile 中有两种表示方式来 控制某个文件是否编译,一种是默认选择编译,用 obj-y 表示,如:

```
obj-y += usb-host.o #默认编译 usb-host.c 文件
```

obj-y += gpio/ #默认编译 gpio 目录

另一种表示则与内核配置选项相关联,编译与否以及编译方式取决于内核配置,例如: obj-\$(CONFIG WDT) += wdt.o # wdt.c 编译控制

obj-\$(CONFIG_PCI) += pci/ # pci 目录编译控制

是否编译 wdt.c 文件,或者以何种方式编译,取决于内核配置后的变量 CONFIG_WDT 值:如果在配置中设置为[*],则静态编译到内核,如果配置为[M],则编译为 wdt.ko 模块, 否则不编译。

2.Kconfig 文件

内核源码树每个目录下都还包含一个 Kconfig 文件,用于描述所在目录源代码相关的 内核配置菜单,各个目录的 Kconfig 文件构成了一个分布式的内核配置数据库。通过 make menuconfig (make xconfig 或者 make gconfig)命令配置内核的时候,从 Kconfig 文件读 取菜单,配置完毕保存到文件名为.config 的内核配置文件中,供 Makefile 文件在编译内 核时使用。

3.内核配置工具

用户可以使用如下命令对内核进行配置:

- 1) make config: 基于文本模式的交互式配置;
- 2) make menuconfig: 基于文本模式的菜单型配置;
- 3) make oldconfig: 使用已有的配置文件(.config),但是会询问新增的配置选项;
- 4) make xconfig:图形化的配置(需安装图形化系统);

其中 make menuconfig 是最为常用的内核配置方式。下面将主要介绍 make menuconfig 的使用。

在运行 make menuconfig 前, 主机必须先安装 ncurses 相关的库。在 Ubuntu 下执行 如下命令完成 ncurses 的安装:

\$sudo apt-get install libncurses5-dev

在 Linux 内核源码顶层目录下执行脚本文件:

./build-menuconfig.sh, 可进入 Linux 内核配置主界面, 如下图所示:

```
config - Linux/arm 3.10.101 Kernel Configuration
                  Linux/arm 3.10.101 Kernel Configuration
   Arrow keys navigate the menu. <Enter> selects submenus --->.
  Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
  <M> modularizes features. Press <Esc> to exit, <?> for Help, </>
   for Search. Legend: [*] built-in [ ] excluded <M> module
                                                               < >
         General setup --->
       [*] Enable loadable module support --->
         Enable the block layer
          System Type --->
          Bus support
                       --->
          Kernel Features --->
          Boot options --->
          CPU Power Management
                                --->
          Floating point emulation --
          Userspace binary formats --->
         <Select>
                    < Exit >
                                < Help >
                                                        < Load >
                                            < Save >
```

图 4.1 Linux 内核配置主界面

4.内核配置主界面由三部分组成:

- 1) 最上面部分为基本操作的简要介绍;
- 2) 中间部分为内核配置的各个菜单项;
- 3) 最下面部分为功能菜单

5.基于 Ncurses 的 Linux 内核配置界面不支持鼠标操作,必须用键盘操作。基本操作方法如下:

1) 使用上、下箭头可以选择内核配置菜单项;使用左、右箭头可以选择下排的功能 菜单项;

2) 当功能菜单中的"Select"项被选中时,在某个具有子菜单的内核菜单项上按回车 键可进入该菜单的子菜单;

3) 对于某个菜单项,可以直接按"?"键查看该菜单项的帮助,或者用左右箭头键 将功能菜单移到"Help"上,并按回车键查看帮助;

- 4) 根据菜单项的类型,其配置方式略有不同。可以分成三种情况,具体说明如下:
- 对于以[]开头的配置项,[*]表示选中,[]表示未选中。可以用空格键进行切换
 或者使用键盘快捷键(Y-选中,N-未选中)进行切换;
- 对于以<>开头的配置项,<*>表示静态编译,<M>表示编译为模块,<>表示未选中。可以使用空格键进行切换或者使用键盘快捷键(Y-静态编译,M-编译为模块,N-未选中)进行切换;
- 对于()开头的配置项,表明该配置是数值或者字符串,可以按回车键直接编辑。
- 5) 按斜线(/)可启用搜索功能,填入关键字后可搜索全部菜单内容;

6) 连续按两次 ESC 键或者用左右箭头键将功能菜单移到"Exit"上,并按回车键, 将退回到上一级菜单。如果当前已经位于顶层菜单界面(即内核配置主界面),此时如果 用户对内核配置进行了修改,则将弹出确认修改的提示画面,如下图所示:



选择"Yes"将保存对内核配置的修改并退出,选择"No"将不保存修改直接退出, 连续按两次 ESC 键将重新返回内核配置画面,允许用户继续对内核配置进行修改。

内核配置完成后,其配置信息保存在内核源码顶层目录的.config 文件中。用户可以直接在命令行下将.config 文件备份为其他的文件名,以备日后使用,例如:

\$cp .config .config_bak

为了使用以前备份的内核配置文件,可以直接将备份的内核配置文件覆盖内核源码顶 层目录的.config 文件,例如:

\$cp .cofnig_bak .config

或者执行 make menuconfig 进入内核配置主界面,并用左右箭头选中功能菜单上的 "Load"项,并按回车键,弹出配置文件加载画面,如下图所示:

configura abort.	tion you last	retrieve	ed. Leave	blank to
.config_b	ak]
1011223				

图 6.3 内核配置加载画面

在该画面中输入需要加载的备份文件的名字,选择"Ok"按钮后按回车键,此时会加载用户选择的备份文件,用户可对其进行修改并保存设置产生新的.config 文件。

6.5.3 内核编译

在"TW-STM32MP157光盘资料\2.软件开发参考资料\3.软件源码\1.内核"目录中,包含了一个已经配置好的 Linux 内核源码文件 tw-stm32mp157_kernel_0621.tar.gz。编译 Linux 操作系统的步骤如下:

为了方便用户开发调试,提供了相对应的编译脚本文件,尽量使用脚本文件进行编译:

./build-menuconfig.sh	对内核进行配置
./build-modules.sh	编译模块 ko
./build-all.sh	编译设备树和内核

1. 将 Linux 内核源码文件 tw-stm32mp157_kernel_0621.tar.gz 拷贝到 Ubuntu 虚拟 机中,此处假定将这该文件拷贝到/home/tw/imx6dl 中,并对其进行解压:

\$cd /home/twdz/MP1/kernel

\$tar -xvf tw-stm32mp157_kernel_0621.tar.gz

2. 设置编译环境。每次编译前都需要把编译环境加载到当前 shell 来,执行如下命令:

\$export

PATH=\$PATH:/home/twdz/MP1/arm-linux/gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnuea bihf/bin

3. 清除前一次的编译结果。执行如下命令:

\$cd /home/twdz/MP1/kernel/

\$ make distclean

4. 使用开发板默认的配置文件生成编译内核时所需要的.config 文件。执行如下命令:

\$ make stm32mp1_tw_deconfig

注:如果已经生成了.config 文件,或已通过 make menuconfig 对配置进行了修改,则 可跳过该步骤。

5. 使用 make menuconfig 对内核进行配置。执行如下命令:

\$ make menuconfig

或者执行脚本 ./build-menuconfig.sh

注:如果不需要对内核配置进行了修改,则可跳过该步骤。

6. 编译内核或者设备树。执行如下命令或执行脚本文件: ./built-all.sh

\$./built-all.sh

编译完成后,会在内核源码顶层目录(即 kernel 目录)的 arch/arm/boot 子目录下生 成 uImage 文件。

6.6 搭建 QT 编译环境

如果用户使用光盘资料里提供的虚拟机,路径为"TW-STM32MP157光盘资料\2.软件 开发参考资料\4.开发环境\ubuntu16.04",可以直接跳过此章节。否则,需要重新搭建编译 QT的环境。

1. 安装交叉工具链。

将光盘资料中"TW-STM32MP157光盘资料\2.软件开发参考资料\1.编译工具"的 st-example-image-qtwayland-openstlinux-weston-stm32mp1-x86_64-toolchain-3.1-snapshot.sh 脚本文件拷贝到虚拟机。并执行以下命令安装。

twdz@ubuntu:~\$ chmod 777 st-example-image-qtwayland-openstlinux-weston-stm32mp1-x86_64-toolchain-3.1-snapshot.sh twdz@ubuntu:~\$./st-example-image-qtwayland-openstlinux-weston-stm32mp1-x86_64-toolchai n-3.1-snapshot.sh

安装过程中需要按 y 选择默认选项,完成后/opt/st 路径下会出现 stm32mp1 文件夹。

2. 安装 linux QtCreator5.14.2 版本。

可以将光盘路径为"TW-STM32MP157光盘资料\2.软件开发参考资料\1.编译工具 \qt-opensource-linux-x64-5.14.2.run"安装包拷贝到虚拟机中。也可以通过命令来下载安装 包,执行指令:

wget -c http://download.qt.io/archive/qt/5.14/5.14.2/qt-opensource-linux-x64-5.14.2.run

Connecting to mirrors.ustc.edu.cn (mirrors.ustc. 0 connected. HTTP request sent, awaiting response 200 OK Length: 1335706944 (1.2G) [application/x-makese] Saving to: 'qt-opensource-linux-x64-5.14.2.run'	.edu.cn) Lf]	2001:da8:d	800:9	5::110	:8
qt-opensource-linux 100%[=================>]	1.24G	74.6MB/s	in	27s	
2021-08-11 09:25:23 (47.7 MB/s) - 'qt-opensource 35706944/1335706944]	e-linux-	x64-5.14.2.	run'	saved	[13

- 3. 修改安装包权限,执行指令: sudo chmod 777 qt-opensource-linux-x64-5.14.2.run
- 4. 执行指令:sudo ./qt-opensource-linux-x64-5.14.2.run 后,会弹出QT5.14.2的安装界面。

😣 🗊 Qt 5.14.2 S	etup
Welcome to the 0	Qt 5.14.2 installer
01	This installer provides you with the open source version of Qt 5.14.2.
	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.
	Network requests completed.
Settings	Next Cancel

图 6.4 QtCreator 安装界面

需要输入账号和密码,如果没有,则去Qt 官网 https://www.qt.io/注册一个账号。

Qt	Pl Login	ease log in to Qt Account
		Need a Qt Account?
	Sign-up	Valid email address
		Password
		Confirm Password
		I accept the <u>service terms</u> .

图 6.5 输入用户密码界面



图 6.6 同意使用条款

Default Select All Deselect All Qt Qt Qt ✓ Qt 5.14.2 ✓ ✓ Desktop gcc 64-bit Android ✓ Sources ✓ ✓ Qt Charts ✓ ✓ Qt Data Visualization ✓ ✓ Qt Data Visualization ✓ ✓ Qt Data Visualization ✓ ✓ Qt Urtual Keyboard ✓ ✓ Qt Wayland Compositor ✓ ✓ Qt WebEngine ✓ ✓ Qt Network Authorization ✓ ✓ Qt Quick Streaming Plugin ✓ ✓ Qt Quick Timeline (Technolo) ✓ ✓ Qt Quick Timeline (Technolo) ✓ ✓ Developer and Designer Tools ✓	Qt
	This component will occupy approximately 4.28 GB on your hard disk drive.

图 6.7 选择安装的选项

5. 安装完成后, 打开 QtCreator 软件(按 window+F 键)。

800	
0	⊗ qt
	A Applications
>	Qt Creator



6. 配置 Qt 版本的 ARM 交叉编译。Tools->Options

<u>F</u> ile <u>E</u> dit	<u>Build Debug</u> Analyze	<u>T</u> ools <u>W</u> indow <u>H</u> elp	
			Ctrl+K
		<u>C</u> ++	
Welcome	Projects	QML/JS	•
E		Tests	
Edit	Examples	Code Pasting	
1		<u>B</u> ookmarks	
Design	Tutorials	Git	2
лîк.	75	Text Editing Macros	2
Debug		Me <u>r</u> curial	2
	New to Ot?	For <u>m</u> Editor	
Projects	New to Qt:	Parse Build Output	
6	own applications and	External	
Help	explore Qt Creator.	Diff 2	•
	Get Started Now	Options	

图 6.9 打开配置选项

	Kits
Kits	Kits Qt Versions Compilers Debuggers Qbs CMake
Environment	Name • qmake Location
Text Editor	 Auto-detected Qt 5.14.2 GCC 64bit /opt/Qt5.14.2/5.14.2/gcc_64/bin/qmake
FakeVim	Qt (qt5) /home/twdz/Qtyuanma/arm-qt/bin/qmake
Help	Qt 5.14.2 /opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/qmake
C++	
Qt Quick	
Build & Run	
D <mark>ebugg</mark> er	
Designer	
	Version name: Qt %{Qt:Version}
Python	
Python	qmake location: /opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/qmake Browse
Python Analyzer	qmake location: /opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/qmake Browse No qmlscene installed. No qmlscene installed.
Python Analyzer Version Control	qmake location: /opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/qmake Browse No qmlscene installed. Doe not state for the bit of the b
Python Analyzer Version Control Devices	qmake location: /opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/qmake Browse No qmlscene installed. Qt version 5.14.2 for Desktop Details *

Qt5.14.2 的 交 叉 编 译 工 具 链 qmake , 路 径 在 /opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/qmake

图 6.10 配置 qmake

G++交叉工具链路径为:

/opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/arm-ostl-linux-

gnueabi/arm-ostl-linux-gnueabi-g++

er	Kits										
Kits	Kits	Qt Versions	Compilers	Debuggers	Qbs	CMake					
Environment	Name	2							Туре	1	Add
Text Editor	* Au	to-detected C++									Clone
FakeVim		GCC (C++, x	86 64bit in /u	sr/bin)					GCC		Remove
Help		GCC (C, x86	64bit in /usr/	bin)		astar/liber	ac lateraat	or/clang/hip)	GCC		Remove All
C++	- M	anual	so oquit in Jop	1/Q13.14.2/100	JIS/QLCI	eator/libex	ecyquieau	or/ctarig/bill,	Cialig		Re-detect
Qt Quick		GCC++							GCC		Auto-detection Settings.
Build & Run Debugger Designer		Clang (C, x8 GCC	36 64bit in /op	t/Qt5.12.8/Too	ols/QtCr	eator/libex	ec/qtcreat	or/clang/bin)	GCC		
Python	Nar	ne:	GCC++					2			
Version Control	Cor	npiler path:	/x86_6	4-ostl_sdk-linu	x/usr/bi	n/arm-ostl-l	inux-gnue	abi/arm-ostl	linux-gnueabi-g+	+ Browse	2
Devices Code Pasting Language Client	Pla Pla <u>A</u> BI	tform codegen tform linker fla I:	flags: gs: arm-lii	nux-ger 👻 🛛 arn	n	- linux	~]• [g	eneric *	elf -	32bit *	5

图 6.11 配置交叉编译工具链 G++

ter	Kits										
Kits	Kits Qt	Versions	Compilers	Debuggers	Qbs	CMake					
Environment	Name								1		Add
Text Editor	* Auto-d	letected Desktop Qt	5.14.2 GCC 64	bit (default)					Ŧ	C	lone
FakeVim	 Manua 	l QT5.14.2								Re	move
Help	0.000									Make	Default
C++										Settin	gs Filter
Qt Quick										Default Se	ttings Filter
Build & Run	Name:		ſ	QT5.14.2	٦.	,					Q
Debugger	File syste	em name:	[
Designer	Device by	/pe:		Desktop						*	
Python	Device:			Local PC (defa	ult for C	esktop)				•	Manage
Analyzer	Sysroot:		(Browse
Version Control	Compilor		-	GCC						*	Manage
Devices	compile		c	C++: GCC++						•	Manage
Code Pasting	Environn	nent:		No changes to a	apply.			- 3			Change
Language Client	Debugge	er:		System GDB at	: /usr/bi	n/gdb				*	Manage
Testing	Qt versio	on:	(Qt 5.14.2			4			•	Manage
	Qt mksp	ec:		linux-oe-g++			5				
	Addition	al Qbs Prof	ile Settings:						6	;	Change.7

图 6.12 配置 ARM 板 Kits

6.7 编译 QT 的程序

Ubuntu 虚拟机提供了 5.14.2 版本的 QT creator, 交叉工具 qmake 安装在路径 /opt/st/stm32mp1/3.1-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/bin/qmake。

1. 打开 Qt Creator 应用软件, 创建新的 Qt 程序, 工程名为 helloworld, 点击 next。

Location	Project Location										
Build System Details Translation Kits Summary	This wizard by default f	generates a Qt Widgets Application project. The a from QApplication and includes an empty widget.	pplication deri								
	Name:	helloworld									
	Create in:	/home/twdz	Browse.								
	🗌 Use as	default project location									
		N	ext > Can								
		图 6.13 新建文件名									
Qt Widgets Ap	oplication — (Qt Creator									
	Class In	formation									
Location	closs in										

> Details		generate skeleton source code files.
Translation	Class name:	MainWindow
Summary	Base class:	QMainWindow •
	Header file:	mainwindow.h
	Source file:	mainwindow.cpp
		✓ Generate form
	Form file:	mainwindow.ui

Cancel

2. 选择编译工具, Desktop Qt 5.14.2 GCC 64bit 是 linux 下的编译器, QT5.14.2 是 ARM 版本的交叉工具。

Location	Kit Selection						
Build System	The following kits can be used for project helloworld: Type to filter kits by name ✓ Select all kits						
Details							
Translation Kits							
Summary	✓ 🖵 Desktop Qt 5.14.2 GCC 64bit	Details *					
	☑ 및 QT5.14.2	Details *					

图 6.15 选择 Kit

3. 选择 ARM 版本的编译方式。

	Project: helloworld Deploy: Deploy Configuration Run: Custom Executable	
helloworld	Kit	Build
_ ,	Desktop Qt 5.14.2 GCC 64bit	Debug
Debug	QT5.14.2	Profile
		Release
h		
7		

图 6.16 选择对应的 kit 编译

4. 如 果 编 译 成 功 , 可 执 行 文 件 helloworld 在 /home/twdz/build-helloworld-QT5.14.2-Debug 路径下,通过命令: file helloworld 可以查 看文件信息。



图 6.17 查看可执行文件

5. 将 helloworld 可执行文件放到开发板上运行。

7. 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>http://www.iot-tw.com/</u>获得。

广州眺望电子科技有限公司保留所有权利。