



FCU1201 嵌入式控制单元

Embedded Control Unit

Product Manual_Linux3.0.35
Rev. 1.2
2021/04/12

注意事项与维护



1、注意事项

- **请勿带电插拔核心板及外围模块！**
- 请遵循所有标注在产品上的警示和指引信息。
- 请保持本产品干燥。如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干。
- 使用中注意本产品的通风散热，避免温度过高造成元器件损坏。
- 请勿在无尘、脏乱的环境中使用或存放本产品。
- 请勿将本产品应用在冷热交替环境中，避免结露损坏元器件。
- 请勿粗暴对待本产品，跌落、敲打或剧烈晃动都可能损坏线路及元器件。
- 请勿使用有机溶剂或腐蚀性液体清洗本产品。
- 请勿自行修理、拆卸本公司产品，如产品出现故障请及时联系本公司进行维修。
- 擅自修改或使用未经授权的配件可能损坏本产品，由此造成的损坏将不予以保修。

2、售后维修

如产品使用过程中出现硬件故障可根据售后服务政策进行维修

服务政策：参见官方网站 www.forlinx.com 售后服务说明：

地 址：河北省保定市高开区向阳北大街 2699 号保定飞凌嵌入式新楼 5 层售后维修部

联 系 人：售后维修部

电 话：0312-3102650-952/953 邮编：071000

邮寄须知：建议使用顺丰、圆通或韵达，且不接收任何到付

技术支持与定制

1、技术支持方式

- 1.1 电话: 0312-3119192
- 1.2 论坛: bbs.witech.com.cn
- 1.3 邮箱:

Linux 技术支持:	linux@forlinx.com
Android 技术支持:	android@forlinx.com
硬件技术支持:	hardware@forlinx.com

- 1.4 知识库: bbs.witech.com.cn/kb

2、技术支持时间

周一至周五: 上午 9:00—11:30, 下午 13:30—17:00;

公司按照国家法定节假日安排休息, 在此期间无法提供技术支持, 请将问题发送至邮箱或论坛技术支持区, 我们会在工作日尽快给您回复。

3、定制开发服务

我公司提供嵌入式操作系统底层驱动、硬件板卡的有偿定制开发服务, 以缩短您的产品开发周期。

了解定制流程: <http://www.forlinx.com/OEM.htm>

填写需求文档: <http://www.forlinx.com/docs/PR.docx>

发至项目邮箱: project@forlinx.com

资料更新与获取

1、资料的更新

产品相关资料会不断的完善更新，本手册内容亦然如此；当您在使用这些内容时，请确保其为最新状态。

2、更新后如何通知

飞凌嵌入式产品资料更新通知采用微信公众号推送，敬请关注！



服务号

3、资料如何获取

3.1 网络下载：

请注册并登陆“bbs.witech.com.cn”找到“[开发板资料下载](#)”选择对应平台下载；

下载前请阅读《[资料下载说明](http://bbs.witech.com.cn/thread-67932-1-1.html)》：<http://bbs.witech.com.cn/thread-67932-1-1.html>。

3.2 光盘：

请联系我公司销售人员购买。

版权声明

本手册版权归保定飞凌嵌入式技术有限公司所有。未经本公司的书面许可，任何单位和个人无权以任何形式复制、传播、转载本手册的任何部分，违者将被追究法律责任。

更新记录

日期	版本	更新内容
2019.03.19	V1.0	软件手册第一版
2019.12.02	V1.1	1、增加对 NXP i.MX6Quad 芯片支持 2、更改 Qt 程序中串口实现方式 3、增加 DO 和 DI 的命令行测试程序 do_test、di_test
2021.04.12	V1.2	1、增加了源码编译章节 2、增加了 4G 模块 EC20 支持 3、增加 wifi 模块 8723du 支持

目录

注意事项与维护	- 1 -
技术支持与定制	2
资料更新与获取	3
版权声明	3
更新记录	4
目录	5
第一章 FCU1201 产品介绍	7
1.1 产品简介	7
1.2 应用领域	7
1.3 硬件参数	8
1.4 软件参数	9
第二章 FCU1201 功能介绍	10
2.1 接口示意图	10
2.2 电源供电	10
2.3 用户登录	10
2.3.1 网络登录	10
2.3.2 调试串口	11
2.3.3 查看系统 log	13
2.4 DI、DO	13
2.4.1 接口说明	13
2.4.2 软硬件对应关系	13
2.4.3 命令行测试	14
2.4.4 图形界面测试	14
2.5 串口	14
2.5.1 线序说明	15
2.5.2 软硬件对应关系	15
2.5.3 RS485 测试	15
2.5.4 RS232 接口测试	16
2.5.5 图形界面测试	16
2.6 FlexCAN 测试	17
2.6.1 FlexCAN 线序说明	17
2.6.2 软硬件对应关系	17
2.6.2 命令行测试	17
2.6.3 图形界面测试	18
2.7 显示	18
2.7.1 LVDS 接口测试	18
2.7.2 HDMI 接口测试	18
2.8 背光	18
2.8.1 命令行测试	19
2.8.2 Qt 界面测试	19
2.9 系统复位	19
2.9.1 Reset 按键手动复位	19
2.9.2 看门狗自动复位	20
2.10 USB 接口测试	20
2.10.1 USB HOST 接口存储测试	20
2.10.2 OTG 转 HOST 接口测试	20
2.11 有线网卡测试	21
2.11.1 软硬件对应关系	21

2.11.2 网卡测试	21
2.11.3 网络的相关服务测试	22
2.12 4G 模块测试	23
2.12.1 查看已连接的模块类型	24
2.12.2 4G 拨号	24
2.12.3 4G 上网测试	25
2.12.4 4G-AP	25
2.13 WiFi 测试	26
2.13.1 板载 WiFi 测试	26
2.13.2 板载 AP 测试	28
2.14 RTC	28
2.15 TF 卡测试	29
2.16 ESAM、PSAM 测试	29
2.17 录音/放音	30
2.17.1 命令行测试	30
2.17.2 图形界面测试	30
第三章 系统固件更新	32
3.1 使用烧写工具烧写镜像	32
3.1.1 烧写 Qt 4.8.6GUI linux	32
3.2 TF 卡更新固件	32
3.2.1 制作 TF 卡	32
3.2.2 TF 卡更新系统	35
第四章 FCU1201 Linux 编译	36
4.1 版本说明	36
4.2 编译环境说明	36
4.3 编译环境搭建	36
4.3.1 安装交叉编译器	36
4.3.2 安装依赖包	37
4.4 编译	38
4.4.1 编译 u-boot	38
4.4.2 编译 Linux-3.0.35	38
4.4.3 制作 rootfs qt4.8.6	39
附录一 外壳尺寸图	41

第一章 FCU1201 产品介绍

1.1 产品简介

FCU1201 嵌入式控制单元采用 NXP i.MX6Dual Lite 双核处理器（或者选配 NXP i.MX6Quad 四核处理器）开发设计，具有超高效、高性能、接口丰富等优势。主频高达 1GHZ，1GB DDR3，8GB eMMC，内部集成 RS485、CAN、ESAM、PSAM、USB、以太网口、4G、WiFi、LVDS 屏、HDMI、DI、DO、音频功能接口和模块，以满足不同场合的需求。



产品特点：

- 采用 NXP 的 i.MX6Dual Lite 处理器（或者选配 NXP i.MX6Quad 处理器），高性能、低功耗、高可靠性
- 核心模块所有元器件达到工业级-40 至 85℃温度范围
- 支持 ISO7816 协议，可直接与国家电网 ESAM/PSAM 模块通信
- 内置超级电容，断电后至少可维持系统正常运行 15 秒，确保信息不丢失
- 采用模块化设计，可迅速针对客户的个性化需求提供私人定制服务
- 4 路 DI、4 路 DO，2 路 485、2 路 CAN，均采用电气隔离和接口保护，安全可靠
- 可通过 TF 卡升级系统，无需拆卸外壳，简单方便
- 内置超级电容，外部电源掉电后可维持系统运行 15s（只接 LVDS 屏）
- 采用铝合金型材外壳，带耳体积仅 100×147.5×41.8mm，体积小巧、外形美观、安装方便
- 采用 7 吋 LVDS 触摸屏（深圳拓普微提供），具备友好的人机交互界面
- 标准 MINI HDMI 接口，支持 1080P、720P 高清显示屏
- 全面的状态指示灯，令系统运行、网络通讯、接口连接……所有状况一目了然
- 通讯方式多样，板载千兆网口、Wi-Fi&蓝牙、4G 模块
- 标准 DB9 调试串口
- 标准 TF 卡插槽，方便扩充本地存储空间
- 标准 3.5mm 立体声耳机接口，内置话筒，亦可增设 1W×2 喇叭或 3.5mm 单声道话筒接口

1.2 应用领域

FCU1201 嵌入式控制单元适用于充电桩、广告牌、新零售、安防、车载、电力通讯等领域。

1.3 硬件参数

设备	描述	
CPU	NXP i.MX6Dual Lite ARM Cortex-A9 双核 1GHz	NXP i.MX6Quad ARM Cortex-A9 四核 1.2GHz
RAM	DDR3 1GB	
ROM	eMMC 8GB	
外扩存储	标准 TF 卡接口，最大支持 64GB（实测）	
移动通信	华为 ME909S 模块； 支持中国移动 4G/3G/2G、中国联通 4G/3G/2G、中国电信 4G； 采用标准 SIM 卡槽（卡槽上方标有 4G 标识）	
ESAM	支持 ESAM 芯片，ISO7816 协议； 飞凌提供读写驱动，用户自行购买板载芯片	
PSAM	支持 PSAM 卡； 采用抽屉式 Mini SIM 卡槽（卡槽下方有“SIM”标识） 飞凌提供读写驱动，用户自行买卡	
开关量输出	4 路，电磁继电器隔离； 触点容量：1A 30VDC / 0.5A 125VAC / 0.3A 60VDC 接口：3.81mm 间距绿端子	
开关量输入	4 路，光耦隔离，采用 3.81mm 间距绿端子； 默认配置为：直流电压输入，3V 至 24VDC 被判定为高电平，1VDC 以下被判定为低电平； 亦可配置为：内部提供隔离的 5V 电源，外部仅提供干接点。	
触摸彩屏接口	采用标准 DVI-I 插座； 默认 LMT070DICFWD-AKA 液晶显示器	
Mini HDMI接口	采用标准 mini HDMI 插座	
断电应对措施	CPU 有一路 GPIO 专门用于监测外部电源状态； 超级电容至少可维持系统运行 15 秒； 监测整机内部的 5V 主电源电压，当该电压跌落超过 10%时，整机断电， 以免系统电压过低导致软件异常。	
串口（包含RS-485）	UART1：三线调试串口，DB9 插座，非隔离，机壳标识是 Console UART3：在机内转换为 RS-485-1，隔离电压 1.5KV，静电四级防护 UART4：在机内转换为 RS-485-2，隔离电压 1.5KV，静电四级防护 UART5：接读卡器，三线，绿端子引出，非隔离	
USB	1 个 USB OTG 接口，采用标准 Micro USB 插座 1 个 USB 主口，采用标准 USB A 型插座	
CAN BUS	CAN1：CAN2.0 B，1Mbps，隔离电压 1.5KV，静电四级防护 CAN2：CAN2.0 B，1Mbps，隔离电压 1.5KV，静电四级防护	
以太网路	标准 RJ-45 插座，10M/100/1000M 自适应	
蓝牙和Wi-Fi	采用 F23BUUM13-W2 模块，支持 IEEE 802.11b/g/n 1T1R WLAN and Bluetooth 2.1/3.0/4.0	
实时钟	采用专用 RTC 芯片 RX8010SJ； 板载 CR2032 电池，可至少维持 1 年走时	
音频	3.5mm 标准立体声耳机插座，内置单声道话筒； 可增设 1W×2 喇叭插座或 3.5mm 单声道话筒接口（PCB 有预留插座焊盘，但挡板未开孔）	
复位按键	1 个，用于系统复位	
Boot按键	1 个，与复位按键同时使用，用于系统固件更新	
电源与功耗	额定电压：12V，具备反接保护； 配拓普微 LMT070DICFWD-AKA 液晶显示器时的总功耗为 5.1W	

尺寸	100mm*147.5mm*41.8mm(长*宽*高)
安装	8 只 Φ3mm 螺钉
工作环境	湿度：5%~95%，无凝露。 工作温度：-40℃~70℃（注：WiFi 模块工作温度为 0℃~70℃） 存储温度：-40℃~85℃（注：WiFi 模块存储温度为-40℃~80℃）

1.4 软件参数

软件支持	详细描述
操作系统	Linux-3.0.35
文件系统	Ext3
GCC	4.6.2
RS485	提供 485 测试示例
网络协议	TCP/IP、UDP、DHCP、TFTP、FTP、Telnet、SSH、Web、HTTP、IPtables、SQLITE
以太网	10M/100M/1000 自适应以太网，支持静态/动态分配 IP 地址，支持修改 MAC 地址
WiFi	支持 STA、AP 功能
4G 网络	ME909 模块，支持移动、联通、电信 4G 上网功能
RTC	支持 NTP 自动校时
看门狗	支持复位时间设置

第二章 FCU1201 功能介绍

2.1 接口示意图

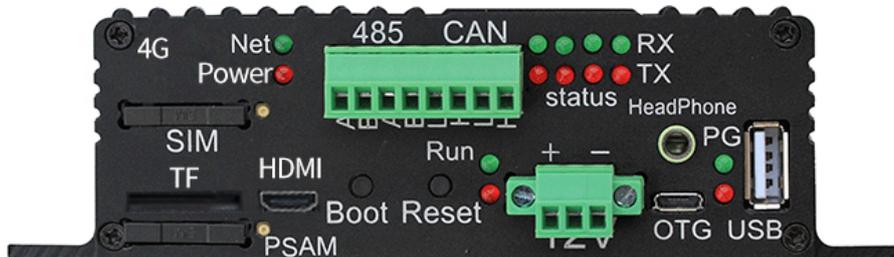


图 1 左侧接口图



图 2 右侧接口图

2.2 电源供电

本产品为 DC12V 供电，飞凌提供适配器为 12V 2A。

实物图	说明	
	引脚序号	引脚说明
	1	电源正极，DC 12V
	2	保护地
	3	电源负极，GND

电源供电状态可通过电源指示灯显示，如下图：



CPU 有一路 GPIO 专门用于监测外部电源状态，当外部电源电压高于 8V 时，该 IO 为高电平，外壳上的 PG 绿灯点亮；反之为低电平，同时 PG 绿灯熄灭。

当外部电源中断后，系统自动切换为内置的超级电容供电，同时面板上的 PG 红灯点亮。超级电容至少可维持系统运行 15 秒，同时监测整机内部的 5V 主电源电压，当该电压跌落超过 10% 时，整机断电，以免系统电压过低导致软件异常。

2.3 用户登录

FCU1201 支持调试串口与网络两种登录系统的控制方法，本手册以串口登录方式说明。

2.3.1 网络登录

将 PC 机或笔记本通过网线连接 FCU1201，FCU1201 的 eth0 开机默认 IP 地址为 192.168.0.232，将 PC 的 IP 地址设置为 192.168.2.xx 网段，正常上电之后，通过 telnet 登录，输入命令测试基本功能与应用。

在 Windows 的命令窗口下输入 telnet 192.168.0.232，用户名输入 root，密码为 root。

注：开始->运行，输入 cmd 回车，即可进入 Windows 的命令窗口

```
Welcome to Freescale Semiconductor Embedded Linux Environment
```

```

freescale login: root
Password:
root@freescale ~$
    
```

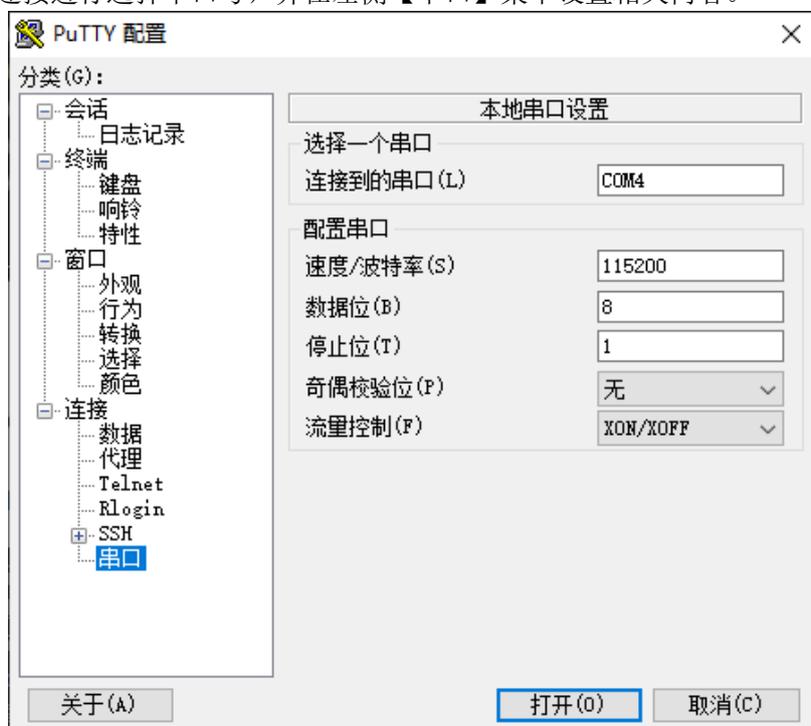
2.3.2 调试串口

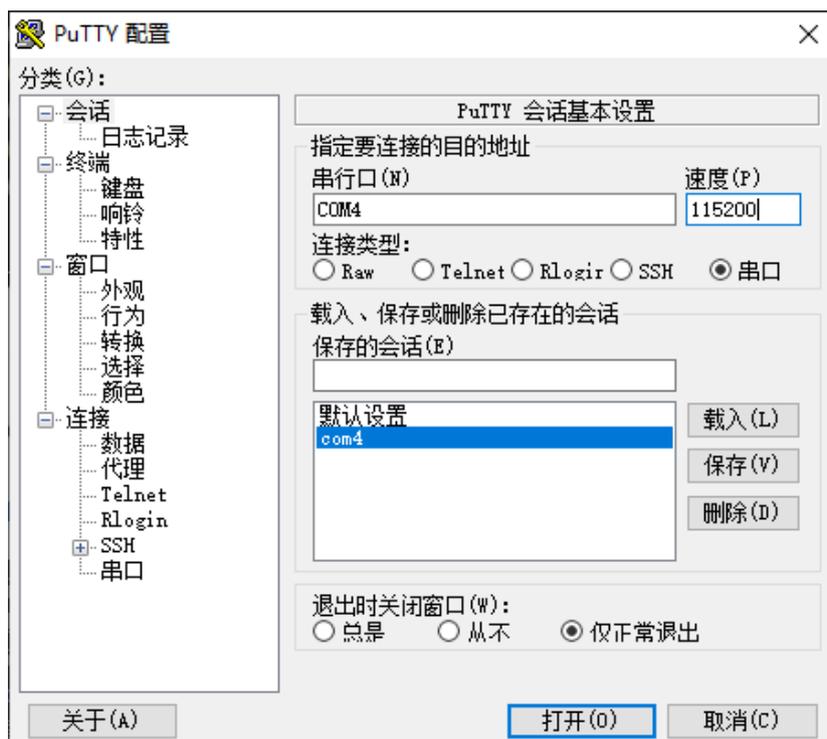
FCU1201 的调试串口在面板上标识符为 Console，如图所示：



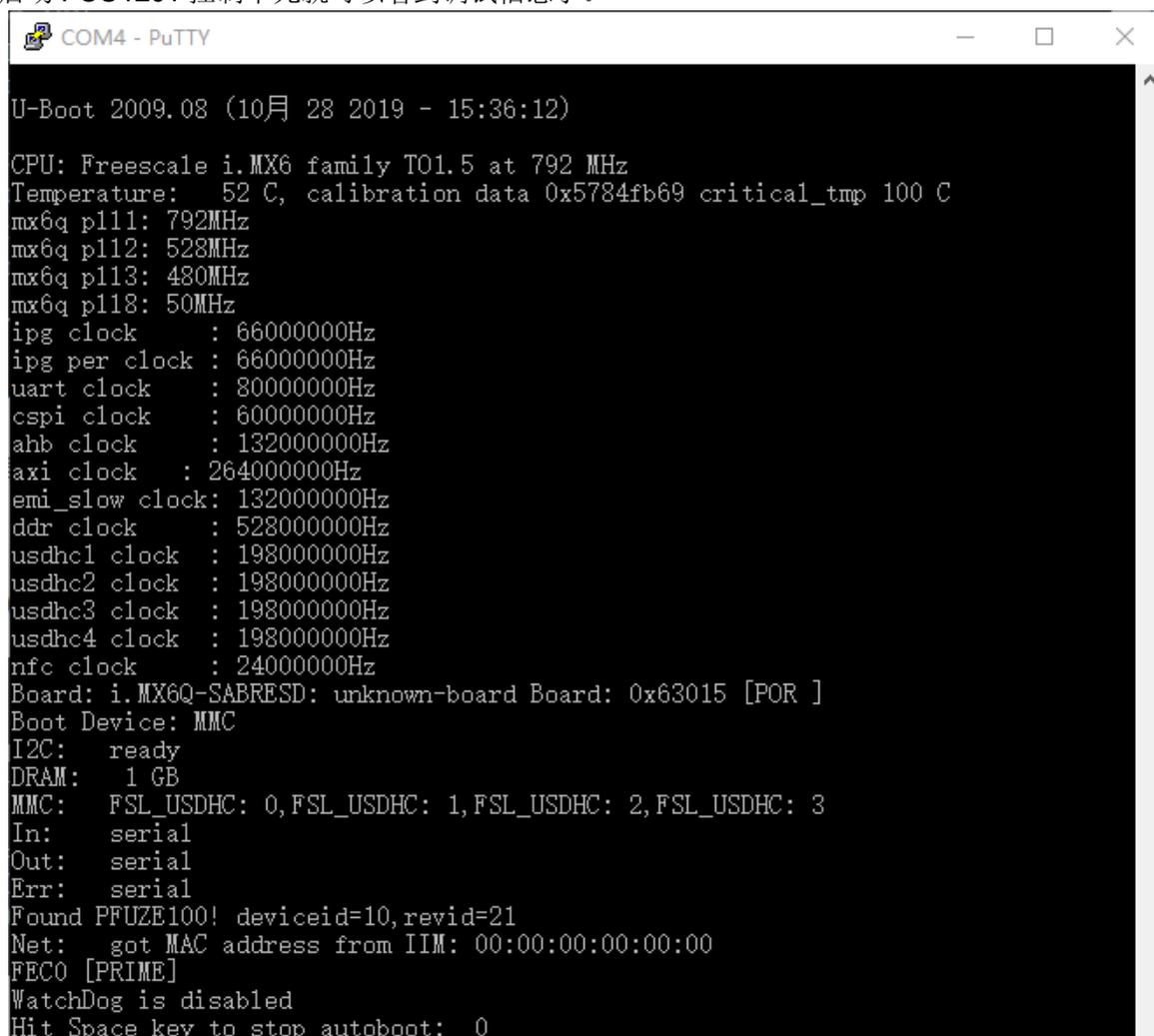
调试串口 Console 用于查看板子运行信息，作为调试口，不做它用。

- 1、将 PC 和 FCU1201 控制单元的 Console 通过双母头交叉的串口线连接。
- 2、打开超级终端（Win7 系统的可以使用“用户资料\工具\putty.exe”）进行如下的设置：
根据您的串口连接进行选择串口号，并在左侧【串口】菜单设置相关内容。





注意:此时采用的波特率为 115200，8 位数据位，无奇偶校验，1 位停止位，无数据流控制。设置完成后，启动 FCU1201 控制单元就可以看到调试信息了。



3、只需输入“root”并敲击回车，密码也是”root”，即可进入命令行；

```

arm-none-linux-gnueabi-gcc (Freescale MAD -- Linaro 2011.07 -- Built at 2011/08/
10 09:20) 4.6.2 20110630 (prerelease)
root filesystem built on Thu, 23 Apr 2015 13:54:31 +0800
Freescale Semiconductor, Inc.

freescale login: root
Password:
login[2853]: root login on 'ttymxc0'
calibrated!
root@freescale /$
    
```

2.3.3 查看系统 log

系统启动之后，可使用 `dmesg` 命令查看内核启动的信息。

采用 `dmesg`：

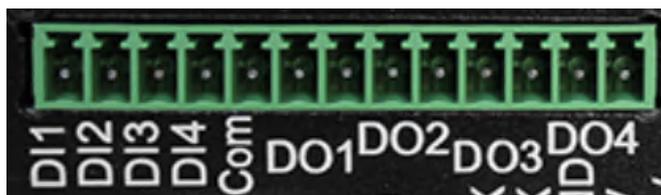
```

root@freescale /$ dmesg
0x80ba3e80 (456 kB)
.bss : 0x80ba3ea4 - 0x80bfb3c (352 kB)
SLUB: Genslabs=13, HWalign=32, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
Preemptible hierarchical RCU implementation.
NR_IRQS:624
MXC GPIO hardware
sched_clock: 32 bits at 3000kHz, resolution 333ns, wraps every 1431655ms
arm_max_freq=1GHz
MXC_Early serial console at MMIO 0x2020000 (options '115200')
bootconsole [ttymxc0] enabled
Console: colour dummy device 80x30
Calibrating delay loop... 1581.05 BogoMIPS (lpj=7905280)
pid_max: default: 32768 minimum: 301
    
```

2.4 DI、DO

2.4.1 接口说明

DI、DO 接口如下图：



其中数字量输入接口能承受直流 24V 之内的开关量电压信号，3V 以上为高电平，1V 以下为低电平。这些输入口内部采用兼容设计，亦可根据用户需求，配置为机内 5V 供电，外部仅提供干接点的形式。

2.4.2 软硬件对应关系

设备节点名称	硬件标识名称
/dev/gpio_input_0	DI1
/dev/gpio_input_1	DI2
/dev/gpio_input_2	DI3
/dev/gpio_input_3	DI4
/sys/class/leds/do1/brightness	DO1
/sys/class/leds/do2/brightness	DO2
/sys/class/leds/do3/brightness	DO3
/sys/class/leds/do4/brightness	DO4
/sys/class/leds/run-red/brightness	RUN RED
/sys/class/leds/run-green/brightness	RUN GREEN

2.4.3 命令行测试

背光的亮度设置范围为（0-248），248 表示亮度最高。0 表示关闭背光亮度。进入系统后在终端下输入如下命令进行背光测试。

1. DO 输出测试

```
root@freescale/$ do_test
do1
do2
do3
do4
red
gre
do1
do2
do3
do4
red
gre
```

串口终端返回对应 DO1~DO4 依次导通，RUN RED 和 RUN GREEN 依次灭掉；然后 DO1~DO4 依次断开，RUN RED 和 RUN GREEN 依次点亮；通过在串口终端输入快捷键：Ctrl+C 结束测试。

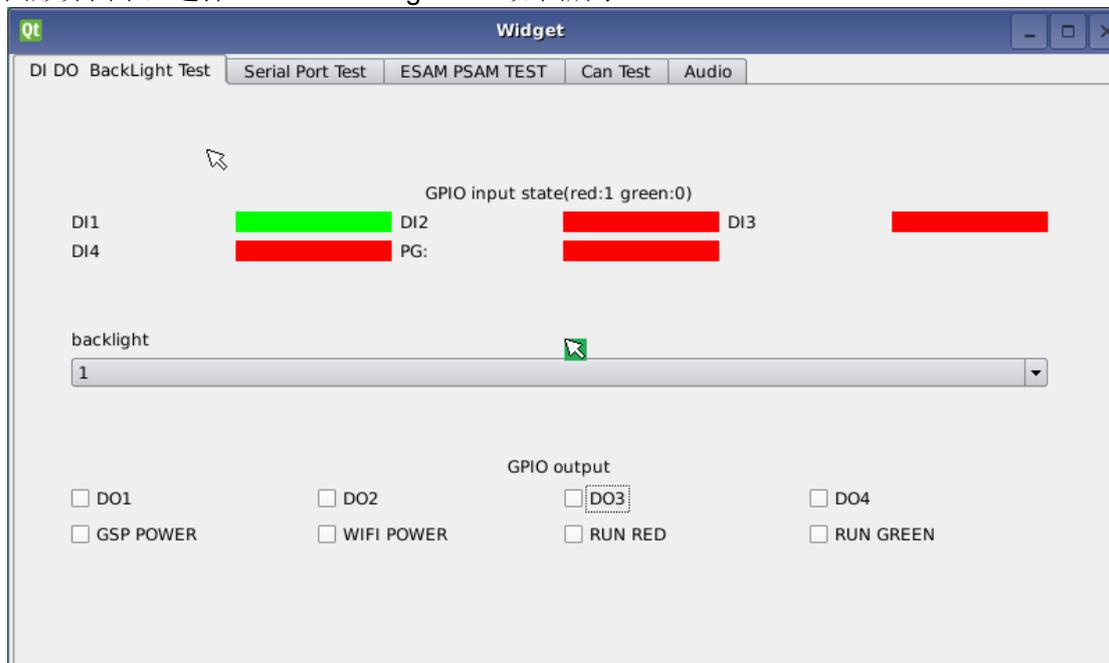
2. DI 输入测试

```
root@freescale/$ di_test
DI1= 0 DI2= 0 DI3= 0 DI4= 0 PW= 1
```

串口终端返回对应 DI1~DI4 和 PG 的当前状态，高电平用 1 表示，低电平用 0 表示；通过在串口终端输入快捷键：Ctrl+C 结束测试。

2.4.4 图形界面测试

在图形界面中，选择 DI DO BackLight Test 如图所示：



DI DO 测试：

改变 DI 的输入状态，界面自动监控 DI 的状态，红色代表 1，绿色代表 0

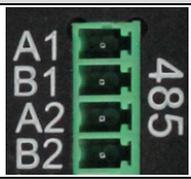
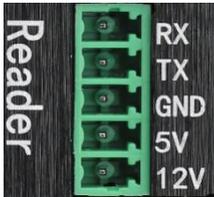
选择 DO 状态，DO 输出随着复选框的状态改变（每次刷新所有的 DO）

运行指示灯 RUN 测试：

选择 RUN RED 或者 RUN GREEN 的状态，运行指示灯输出随着复选框的状态改变

2.5 串口

2.5.1 线序说明

实物图	说明	
	引脚	引脚说明
	A1	RS485_1 Data+
	B1	RS485_1 Data-
	A2	RS485_2 Data+
	B2	RS485_2 Data-
	引脚	引脚说明
	RX	串口数据接收
	TX	串口数据发送
	GND	数字地
	5V	5V 电源输出
	12V	12V 电源输出

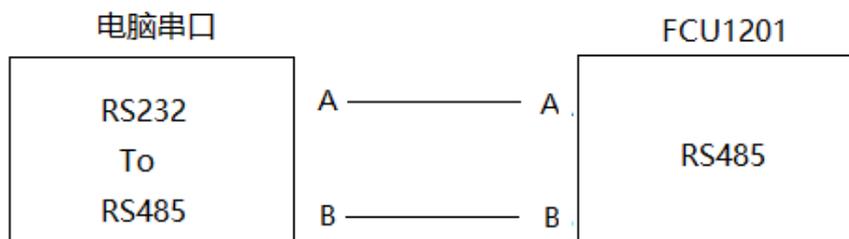
2.5.2 软硬件对应关系

设备节点名称	硬件标识名称
/dev/ttyxc2	RS485_1
/dev/ttyxc3	RS485_2
/dev/ttyxc4	Reader

2.5.3 RS485 测试

以 RS485_1 为例，如需测试其他 RS485 接口，只需修改对应接口的设备节点名称即可；

本测试方法以与电脑通讯为例，将 RS232 转 RS485 模块和 FCU1201 的 RS485_1 接口连接，连接方式如下所示：



输入如下命令：

```
root@freescale /$tty_test /dev/ttyxc2 9600
```

RS485 会以 9600 波特率向 PC 机发送 “forlinx uart test.....bz”，同时设置 PC 端串口助手软件定时发送 “abcdefg”。调试信息如下：

```
root@freescale /$ tty_test /dev/ttyxc2 9600
```

```
Welcome to TTYtest! Press Ctrl + 'c' to stop.
```

```
/dev/ttyxc2,creat thread 727356528 sucess
```

```
/dev/ttyxc2,creat thread 735790192 sucess
```

```
abcdefgabcdefgabcdefgabcdefgabcdefgabcdefgabcdefgabcdefgabcdefg
```

```
/dev/ttyxc2,Send: 448 ,Receive: 63
```

PC 端的串口助手软件（用户资料/工具/sscom32.exe），发送与接收数据如下：



2.5.4 RS232 接口测试

FCU1201 嵌入式控制单元中除了调试串口 Console 之外还有一个串口，在 FCU1201 嵌入式控制单元面板上的标识符为 Reader，并提供对外 5V 与 12V 供电接口。

Reader 外设接口测试方法如下：

- 1、将 RS232 电平的 Reader 外设接口的 RX 和 TX 短接；
- 2、终端中输入如下命令，tty_test 启动后会以 9600 波特率不断的向外发送“forlinx uart test.....bz”。

```
root@freescale ~$ tty_test /dev/ttymx4
```

串口信息为：

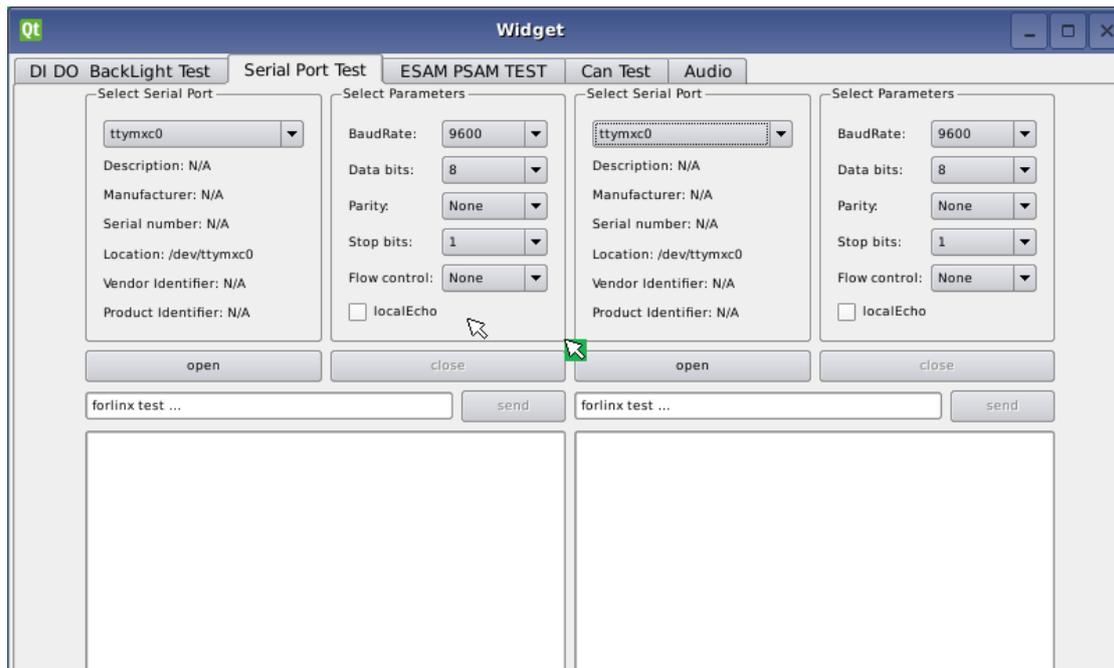
```
Welcome to TTYtest! Press Ctrl + 'c' to stop.

/dev/ttymx4,creat thread 726602864 sucess
/dev/ttymx4,creat thread 735302768 sucess
forlinx uart test.....bzforlinx uart test.....bzforlinx uart test.....bzforlinx uart test.....bzforlinx
uart test.....bzforlinx uart test.....bzforlinx uart test.....bz
/dev/ttymx4,Send: 224 ,Receive: 168
```

2.5.5 图形界面测试

选择 Serial Port Test 页面，此测试页面可同时测试 2 个串口，以 Reader 接口测试为例，测试步骤如下：

- 1、将 Reader 外设接口的 RX 和 TX 短接；
- 2、串口名称选择 ttymx4
- 3、点击 open 按钮打开串口
- 4、点击 send 按钮发送字符，就可以下面的数据接收区接收到串口不断发送的 forlinx test...



2.6 FlexCAN 测试

2.6.1 FlexCAN 线序说明

实物图	说明	
	引脚	引脚说明
	L1	CAN1_L
	H1	CAN1_H
	L2	CAN2_L
	H2	CAN2_H

2.6.2 软硬件对应关系

备节点名称	硬件标识名称
can0	CAN0
Can1	CAN1

2.6.2 命令行测试

1、FlexCAN 连线方式：CAN 总线有两个物理连线，CAN 的 H 端子与其它 CAN 设备 H 端子连接；CAN 的 L 端子与其它 CAN 设备 L 端子连接。默认 CAN 口已经启动。

2、以 can0 为例进行测试，客户端发送数据，服务端接收数据（可用两个 FCU1201 嵌入式控制单元来模拟或者购买 PC 端的 USB 转 CAN 模块测试）

服务端接收数据（服务端先执行以下命令）：

```
root@freescale ~$candump can0
interface = can0, family = 29, type = 3, proto = 1
```

客户端发送数据（客户端发送数据）：

```
root@freescale ~$cansend can0 -i 0x10 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
interface = can0, family = 29, type = 3, proto = 1
```

此时服务端会接收到以下数据：

```
<0x010> [8] 11 22 33 44 55 66 77 88
```

3、修改 tx_queue_len

修改方法如下：

```
root@freescale ~$echo 1000 > /sys/class/net/can0/tx_queue_len 修改的数值酌情选择
```

4、修改波特率

以修改 can0 的波特率为 125000 为例，修改方法如下：

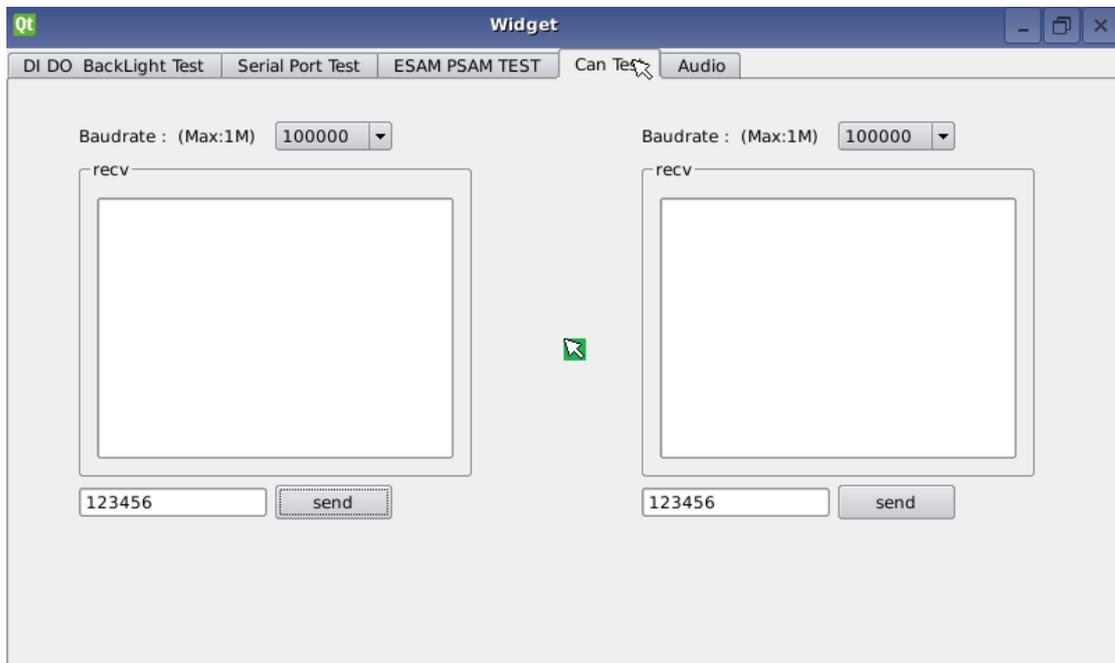
```

root@freescale ~$ifconfig can0 down
root@freescale ~$ip link set can0 up type can bitrate 125000 triple-sampling on
root@freescale ~$ifconfig can0 up
    
```

2.6.3 图形界面测试

在图形界面中，选择 Can Test 页面。

测试程序能同时测试 2 路 can，can 接口在程序启动时已打开，选择不同的波特率或重新按新的波特率打开接口。



2.7 显示

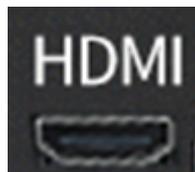
2.7.1 LVDS 接口测试

关闭电源，将 LCD 屏幕接至 LVDS 接口，上电即可正常显示，触摸屏可以使用。现阶段支持 LMT070DICFWD-AKA 液晶显示器。



2.7.2 HDMI 接口测试

本机采用标准 Mini HDMI 接口座，如下图：



第一步、使用 mini HDMI 转 HDMI 接口线连接嵌入式控制单元与显示器

第二步、在命令行中输入以下命令进行 HDMI 测试

```

root@freescale ~$ cd /root/
root@freescale ~$ ./hdmi.sh
    
```

可以看到 HDMI 屏幕上有视频播放。

2.8 背光

2.8.1 命令行测试

背光的亮度设置范围为（0-248），248 表示亮度最高。0 表示关闭背光亮度。进入系统后在终端下输入如下命令进行背光测试。

1. 查看当前屏幕背光最大值

```
root@freescale/$ cat /sys/devices/platform/pwm-backlight.0/backlight/pwm-backlight.0/max_brightness
```

串口终端返回当前屏幕背光最大值：

248

2. 查看当前屏幕背光值

```
root@freescale /$ cat /sys/devices/platform/pwm-backlight.0/backlight/pwm-backlight.0/brightness
```

串口终端返回当前屏幕背光值：

128

3. 设置当前屏幕背光值

```
root@freescale/$ echo 248 > /sys/devices/platform/pwm-backlight.0/backlight/pwm-backlight.0/brightness
```

设置成功之后，查看：

```
root@freescale /$ cat /sys/devices/platform/pwm-backlight.0/backlight/pwm-backlight.0/brightness
```

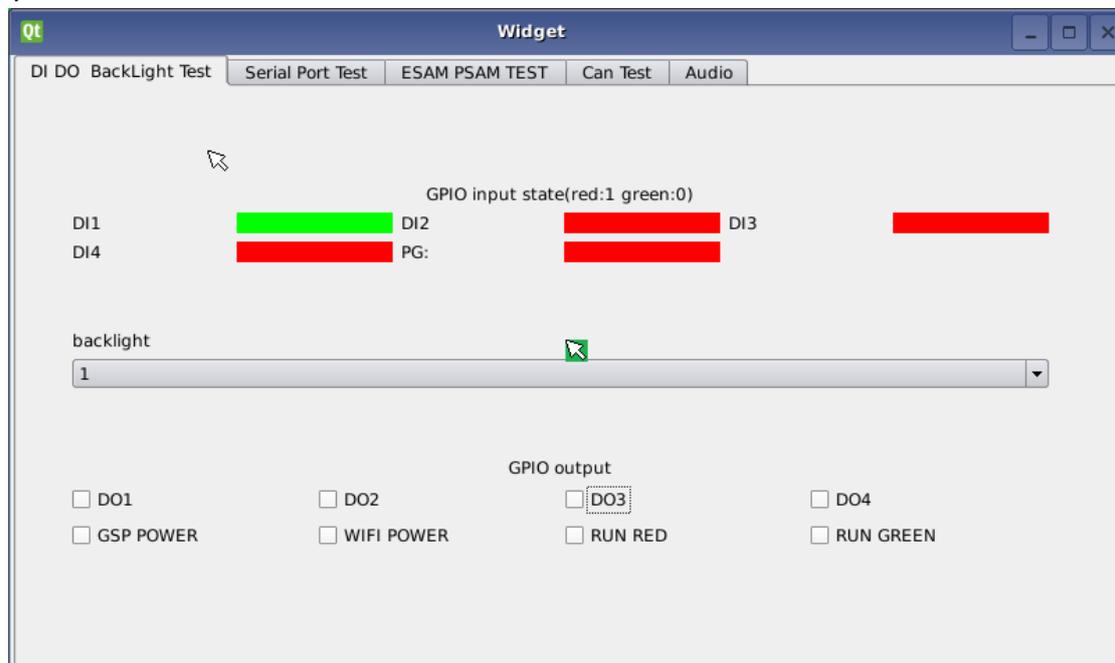
串口终端返回当前屏幕背光值：

248

设置成功。

2.8.2 Qt 界面测试

选择 Qt 图形界面中“DI DO BackLight T”，选择“backlight”可进行 LCD 背光调节如图所示：

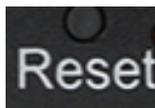


2.9 系统复位

2.9.1 Reset 按键手动复位

按 Reset 按钮，系统立即重启。同时调试串口将打印系统启动时的信息。

Reset 按钮如下图：



2.9.2 看门狗自动复位

看门狗是嵌入式系统中经常用到的功能。

1. 设置看门狗自动复位时间为 5S:

```
root@freescale /#$ watchdogrestart 5
driver set outtime is 60
user set outtime is 5
```

2. 执行 watdogtest 命令，每秒钟喂狗一次，不发生重启。

```
root@freescale ~$ watdogtest
Watchdog Ticking Away!
```

2.10 USB 接口测试

2.10.1 USB HOST 接口存储测试

USB 端子如图:



我们使用 U 盘来测试 USB 功能。U 盘插入到 FCU1201 嵌入式控制单元的 USB 口，超级终端会打印如下信息:

```
root@freescale /#$ usb 2-1.4: new high speed USB device number 3 using fsl-ehci
scsi0 : usb-storage 2-1.4:1.0
scsi 0:0:0:0: Direct-Access      Generic  STORAGE DEVICE   1402 PQ: 0 ANSI: 6
sd 0:0:0:0: [sda] 31116288 512-byte logical blocks: (15.9 GB/14.8 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk

root@freescale /#$ ls -l /media/
drwxr-xr-x  24 root    root          4096 Jan  1 00:43 mmcblk0p1
drwxr-xr-x  10 root    root          8192 Jan  1 00:00 sda1
```

往U盘卡中写入文件，命令如下，写1到test.txt文件中:

```
root@freescale /#$ echo 1 > /media/sda1/test.txt
root@freescale /#$ sync
```

读取U盘卡中test.txt 文件，命令如下:

```
root@freescale /#$ cat /media/sda1/test.txt
1
```

会读到刚才我们写入的 1。

如果 U 盘能自动挂载到/media/sdxx 目录下，并且可以读写到 U 盘内容，就可以确定 USB 口是正常的。

2.10.2 OTG 转 HOST 接口测试

OTG 接口如下图:



使用 OTG 转 HOST 线连接到 OTG 口，插入 U 盘能读取 U 盘内容。



查看 U 盘中内容:

```
root@freescale /$ usb 1-1: new high speed USB device number 5 using fsl-ehci
usb 1-1: device v05e3 p0751 is not supported
scsi3 : usb-storage 1-1:1.0
scsi 3:0:0:0: Direct-Access      Generic  STORAGE DEVICE   1402 PQ: 0 ANSI: 6
sd 3:0:0:0: [sda] 31116288 512-byte logical blocks: (15.9 GB/14.8 GiB)
sd 3:0:0:0: [sda] Write Protect is off
sd 3:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1
sd 3:0:0:0: [sda] Attached SCSI removable disk

root@freescale /$ ls -l /media/sda1/
drwxr-xr-x  2 root  root           8192 Mar  8  2019 2019?3???
-rwxr-xr-x  1 root  root       508311 Feb 27  2019 CanTest
-rwxr-xr-x  1 root  root       582941 Feb 18  2019 EsamTest
drwxr-xr-x  2 root  root           8192 Feb 18  2019 LOST.DIR
-rwxr-xr-x  1 root  root     6433036 Mar  2  2019 NVNBC-SERA
drwxr-xr-x  3 root  root           8192 Jan  1  1980 Project
drwxr-xr-x  2 root  root           8192 Feb  1  2019 System Volume
Information
```

2.11 有线网卡测试

FCU1201 支持 1 路 10M/100M/1000M 自适应以太网，接口方式为 RJ45。如下图所示:



2.11.1 软硬件对应关系

设备节点名称	硬件标识名称
eth0	eth0

2.11.2 网卡测试

将 PC 与 FCU1201 的网口使用网线连接，并设定 PC 机 IP 为固定 IP（此处以 192.168.2.87 为例），FCU1201 的 IP 地址为 192.168.0.232，默认系统启动后，已经打开网络。

在 Windows 的命令窗口下输入 ping 192.168.0.232，注：开始->运行，输入 cmd 回车，即可进入 Windows 的命令窗口。

```
C:\Users\Administrator>ping -n 3 192.168.0.232

正在 Ping 192.168.0.232 具有 32 字节的数据:
来自 192.168.0.232 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.0.232 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.0.232 的回复: 字节=32 时间<1ms TTL=64

192.168.0.232 的 Ping 统计信息:
    数据包: 已发送 = 3, 已接收 = 3, 丢失 = 0 (0% 丢失),
    往返行程的估计时间<以毫秒为单位>:
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

PC 与 FCU1201 之间可 ping 通，使用 Telnet 登录成功，即说明网卡工作正常。

2.11.3 网络的相关服务测试

2.11.3.1 Telnet 服务

FCU1201 在/etc/inetd.conf 脚本文件中已经启动了 telnet 服务，设置好 IP 地址后就可以作为一台 telnet 服务器了。

在 Windows 的命令窗口下输入 telnet 192.168.0.232，用户名输入 root，密码为 root。注：开始->运行，输入 cmd 回车，即可进入 Windows 的命令窗口

```
Trying 192.168.0.232...
Connected to 192.168.0.232
0.232.
Escape character is '^]'.

Welcome to Freescale Semiconductor Embedded Linux Environment

freescale login: root
Password:
root@freescale ~$
```

2.11.3.2 Web 服务

FCU1201 默认移植了一个 webserver: boa ;

boa webserver 是一个小巧高效的 web 服务器，可运行在 Unix 或 Linux 平台，支持 CGI，源代码开放；是一个非常适合于嵌入式系统的单任务 http 服务器。

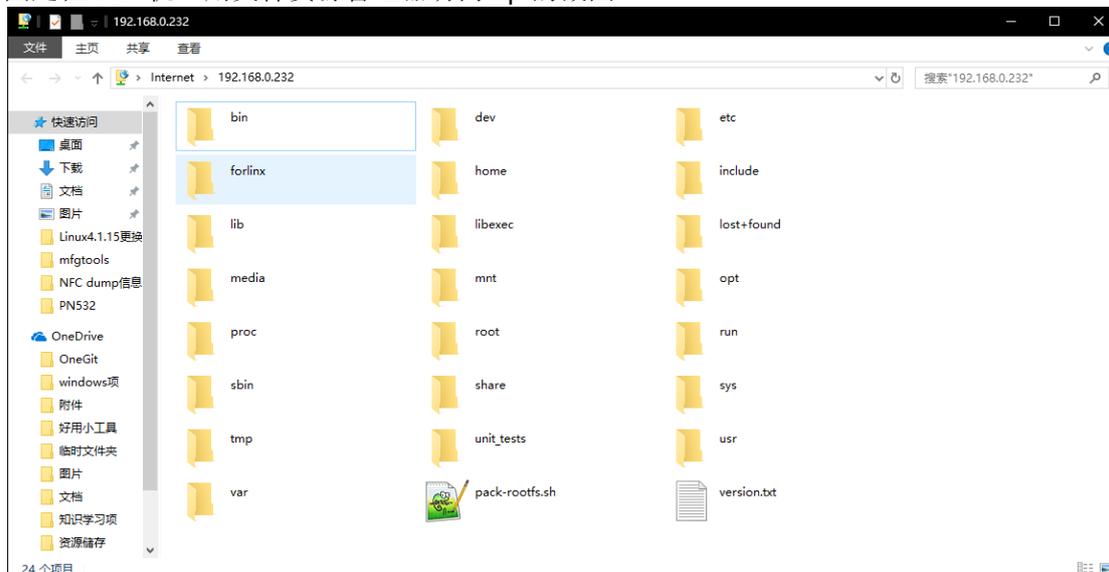
系统启动时已经自动启动了 boa 服务，在 IE 中输入 FCU1201 的 IP 地址即可浏览开发板 webserver 中的网页。下图是在 IE 中浏览的截图：



2.11.3.3 FTP

系统启动时已经自动启动了 vsftpd 服务，可在电脑的文件资源管理器中的地址栏输入 <ftp://192.168.0.232/> 进行访问，用户名 root，密码是 root：

下图是在 PC 机上用文件资源管理器访问 ftp 的截图：



2.12 4G 模块测试

FCU1201 支持 ME909 和 EC20 两种模块，开机之后，默认 4G 模块已供电，驱动已经加载。

注意：4G 电源控制引脚默认是使能的，但是在测试 DO 的时候可能会被关闭，可以通过如下命令测试引脚状态，返回非 0 值为使能。

```
root@freescale /$ cat /sys/class/leds/gprs/brightness
255
```

如果电源被关闭执行如下命令打开电源

```
root@freescale /$ echo 1 > /sys/class/leds/gprs/brightness
```

开机之后，默认 4G 模块已供电，驱动已经加载。

4G 模块配备两支天线，其中 M 为主天线，A 为副天线，主天线是必须的，副天线是用来增强接收效果的，并非必须，但仍建议接上。如下图：



4G 模块 SIM 卡使用标准抽屉式 mini SIM 卡槽，按压抽屉右边的黄色按钮可弹出抽屉，如下图：



2.12.1 查看已连接的模块类型

可以通过 lsusb 命令查看板子连接的模块类型

EC20 的 pid 和 vid 如下：

```
root@freescale /$ lsusb
Bus 001 Device 001: ID 1d6b:0002
Bus 002 Device 001: ID 1d6b:0002
Bus 002 Device 002: ID 0424:2514
Bus 002 Device 003: ID 14e1:6000
Bus 002 Device 004: ID 2c7c:0125
```

ME909 的 pid 和 vid 如下：

```
root@freescale /$ lsusb
Bus 001 Device 001: ID 1d6b:0002
Bus 002 Device 001: ID 1d6b:0002
Bus 002 Device 002: ID 0424:2514
Bus 002 Device 003: ID 14e1:6000
Bus 002 Device 004: ID 12d1:15c1
```

2.12.2 4G 拨号

➤ ME909S

保证模块正常供电之后，拨号，动态分配 IP：

```
root@freescale ~$ cd /root/
root@freescale /root$ ./me909s.sh
udhcpc (v1.20.2) started
Sending discover...
Sending discover...
Sending select for 10.1.157.183...
Lease of 10.1.157.183 obtained, lease time 518400
Deleting routers
adding dns 111.11.1.3
adding dns 111.11.11.3
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=1:01, irq=-1)
```

➤ EC20

保证模块正常供电之后，拨号，动态分配 IP

```
root@freescale /$ quectel-CM &
[1] 3222
[01-01_00:19:47:994] WCDMA&LTE_QConnectManager_Linux&Android_V1.1.34
[01-01_00:19:47:995] quectel-CM profile[1] = (null)/(null)/(null)/0, pincode = (null)
[01-01_00:19:47:998] Find /sys/bus/usb/devices/2-1.4 idVendor=2c7c idProduct=0125
[01-01_00:19:47:998] Find /sys/bus/usb/devices/2-1.4:1.4/net/eth1
[01-01_00:19:47:998] Find usbnet_adapter = eth1
```

```
[01-01_00:19:47:999] Find /sys/bus/usb/devices/2-1.4:1.4/GobiQMI/qcqm1
[01-01_00:19:47:999] Find qmichannel = /dev/qcqm1
[01-01_00:19:48:031] Get clientWDS = 7
root@freescale /$ [01-01_00:19:48:063] Get clientDMS = 8
[01-01_00:19:48:095] Get clientNAS = 9
[01-01_00:19:48:127] Get clientUIM = 10
[01-01_00:19:48:159] Get clientWDA = 11
[01-01_00:19:48:191] requestBaseBandVersion EC20CEHCR06A02M1G
[01-01_00:19:48:287] requestGetSIMStatus SIMStatus: SIM_READY
[01-01_00:19:48:319] requestGetProfile[1] cmnet//0
[01-01_00:19:48:351] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached,
DataCap: LTE
[01-01_00:19:48:383] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[01-01_00:19:48:447] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached,
DataCap: LTE
[01-01_00:19:48:480] requestSetupDataCall WdsConnectionIPv4Handle: 0xe178f700
[01-01_00:19:48:575] requestQueryDataCall IPv4ConnectionStatus: CONNECTED
[01-01_00:19:48:607] ifconfig eth1 up
[01-01_00:19:48:635] busybox udhcpc -f -n -q -t 5 -i eth1
[01-01_00:19:48:652] udhcpc (v1.20.2) started
[01-01_00:19:48:697] Sending discover...
[01-01_00:19:48:817] Sending select for 10.32.204.124...
[01-01_00:19:49:017] Lease of 10.32.204.124 obtained, lease time 7200
[01-01_00:19:49:040] Deleting routers
[01-01_00:19:49:058] adding dns 111.11.1.3
[01-01_00:19:49:059] adding dns 111.11.11.3
```

2.12.3 4G 上网测试

测试是否连接外网:

```
root@freescale /root$ ping www.baidu.com -l usb0 -c 3
PING www.baidu.com (39.156.66.18): 56 data bytes
64 bytes from 39.156.66.18: seq=0 ttl=51 time=50.539 ms
64 bytes from 39.156.66.18: seq=1 ttl=51 time=50.531 ms
64 bytes from 39.156.66.18: seq=2 ttl=51 time=50.469 ms

--- www.baidu.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 50.469/50.513/50.539 ms
```

测试 ping 百度, 发送接收 3 包数据, 丢包率为 0, 4G 可上外网。如果 4G 模块不能连接外网, 请检查 4G 拨号设置或 SIM 卡是否欠费。

2.12.4 4G-AP

1. 4G 模块拨号成功并分配 IP, 可连接外网。设置转发规则:

```
root@freescale ~$ cd /root/
root@freescale ~$ ./me909s.sh & /*拨号*/
root@freescale ~$ ifconfig eth0 down
root@freescale ~$ echo 1 > /proc/sys/net/ipv4/ip_forward /* 打开 IP 转发 */
root@freescale ~$ iptables -t nat -A POSTROUTING -o usb0 -j MASQUERADE /*usb0
为 4G 模块识别出的网卡, 设置转发规则 */
```

2. 设置 WiFi 的模式与 IP

确保模块 8723bu 已经加载。

```
root@freescale ~$ ifconfig wlan0 up /*打开 WiFi*/
root@freescale ~$ ifconfig wlan0 192.168.18.1 netmask 255.255.255.0 /*设置 IP 与子网
掩码*/
root@freescale ~$ ifconfig wlan0 promisc /*设置 wlan0 为混杂模式 */
```

3. 开启 AP

```
root@freescale ~$udhcpd /etc/udhcpd.conf & /*WiFi 地址、网关等配置信息*/
root@freescale ~$hostapd -d /etc/hostapd.conf & /* 加密方式、用户名、密码等设置，此
时用户名为 imx6qap，密码为 1234567890 */
```

4. 手机等移动终端可以通过 WiFi 连接到 FCU1201 的 AP 热点，访问外网。

2.13 WiFi 测试

WiFi 天线如下图：



2.13.1 板载 WiFi 测试

系统启动之后，默认板载 WiFi 模块已使能，驱动已加载。

注意：wifi 电源控制引脚默认是使能的，但是在测试 DO 的时候可能会被关闭，可以通过如下命令测试引脚状态，返回非 0 值为使能

```
root@freescale /$ cat /sys/class/leds/wifi/brightness
255
```

如果电源被关闭执行如下命令打开电源

```
root@freescale /$ echo 1 > /sys/class/leds/wifi/brightness
```

2.13.1.1 查看已连接的模块类型

板载模块支持 8723bu 和 8723du，可以通过 lsusb 命令查看板子连接的模块类型

8723du 的 pid 和 vid 如下：

```
root@freescale /$ lsusb
Bus 001 Device 001: ID 1d6b:0002
Bus 002 Device 001: ID 1d6b:0002
Bus 002 Device 002: ID 0424:2514
Bus 002 Device 003: ID 14e1:6000
Bus 002 Device 004: ID 0bda:d723
```

8723bu 的 pid 和 vid 如下：

```
root@freescale /$ lsusb
Bus 001 Device 001: ID 1d6b:0002
Bus 002 Device 001: ID 1d6b:0002
Bus 002 Device 002: ID 0424:2514
Bus 002 Device 003: ID 14e1:6000
Bus 002 Device 004: ID 0bda:b720
```

2.13.1.2 查看驱动是否加载

8723du:

```
root@freescale /$ lsmod
Module                Size  Used by
8723du                1522301  0
```

注：出现 8723du 即模块加载。无 8723du，则模块没加载，检查内核版本与文件系统中模块版本是否一致。

8723bu:

```
root@freescale /$ lsmod
Module                Size  Used by
8723bu                1184409  0
```

注：出现 8723bu 即模块加载。无 8723bu，则模块没加载，检查内核版本与文件系统中模块版本是否一致。

2.13.1.3 手动加载 WiFi 模块

如果模块没自动加载，请确保已经卸载之后，手动加载模块，例如 8723bu:

```
root@freescale /lib/modules/wifi$ insmod
/lib/modules/3.0.35-2666-gbdde708/kernel/drivers/net/wireless/rtl8723bu/8723bu.ko
```

串口信息:

```
RTL871X: module init start
RTL871X: rtl8723bu v4.3.16_14189.20150519_BTCOEX20150119-5844
RTL871X: build time: Nov 27 2018 10:02:58
RTL871X: rtl8723bu BT-Coex version = BTCOEX20150119-5844
RTL871X: rtw_ndev_init(wlan0)
RTL871X: rtw_ndev_init(wlan1)
usbcore: registered new interface driver rtl8723bu
RTL871X: module init ret=0
```

2.13.1.4 配置 WiFi

```
root@freescale /root$ ./wifi.sh -i 8723 -s H3C -p NONE
wifi 8723
ssid H3C
pasw NONE
RTL871X: indicate disassoc
ps: invalid option -- 'f'
BusyBox v1.20.2 (2015-07-01 11:39:37 CST) multi-call binary.

Usage: ps

Show list of processes

This version of ps accepts no options
  l   Long output
  T   Show threads

Successfully initialized wpa_supplicant
RTL871X: set bssid:00:00:00:00:00:00
ioctl[SIOCSIWAP]: Operation not permitted
RTL871X: set ssid [güsQÿJ«|T]
fw_state=0x00000008
ioctl[SIOCGIWSCAN]: Resource temporarily unavailable
ioctl[SIOCGIWSCAN]: Resource temporarily unavailable
ioctl[SIOCGIWSCAN]: Resource temporarily unavailable
RTL871X: indicate disassoc
wlan0: Trying to associate with 04:d7:a5:84:fa:40 (SSID='H3C' freq=2462 MHz)
RTL871X: set bssid:04:d7:a5:84:fa:40
RTL871X: start auth
udhcpc (v1.20.2) started
Sending discover...
RTL871X: auth success, start assoc
RTL871X: assoc success
wlan0: Associated with 04:d7:a5:84:fa:40
wlan0: CTRL-EVENT-CONNECTED - Connection to 04:d7:a5:84:fa:40 completed [id=0
id_str=]
Sending discover...
Sending select for 192.168.4.251...
Lease of 192.168.4.251 obtained, lease time 7200
Deleting routers
adding dns 222.222.202.202
adding dns 222.222.222.222
Finished!
```

2.13.1.5 测试 WiFi

测试 WiFi, ping 域名或者 IP。

```
root@freescale /root$ ping www.baidu.com -c 3 -I wlan0
PING www.baidu.com (220.181.38.150): 56 data bytes
64 bytes from 220.181.38.150: seq=0 ttl=51 time=12.525 ms
64 bytes from 220.181.38.150: seq=1 ttl=51 time=12.728 ms
64 bytes from 220.181.38.150: seq=2 ttl=51 time=13.851 ms

--- www.baidu.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 12.525/13.034/13.851 ms
```

2.13.1.6 卸载模块

在实际使用时,有可能会需要卸载模块,卸载模块的指令如下:

```
root@freescale /root$ rmmod 8723bu
RTL871X: module exit start
usbcore: deregistering interface driver rtl8723bu
RTL871X: indicate disassoc
RTL871X: rtw_cmd_thread: DriverStopped(1) SurpriseRemoved(0) break at line 564
wlan0: CTRL-EVENT-DISCONNECTED bssRTL871X: rtw_ndev_uninit(wlan0)
sid=04:d7:a5:84:fa:40 reason=3 locally_generated=1
RTL871X: rtw_ndev_uninit(wlan1)
RTL871X: rtw_dev_unload: driver not in IPS
RTL871X: module exit success
```

8723du 执行 rmmod 8723du

2.13.2 板载 AP 测试

默认生成的 WiFi 热点的 IP 地址段为 192.168.18.30-192.168.18.100, 默认网关为 192.168.18.1, 采用 WPA-PSK 加密方式。用户可通过修改/etc/udhcpd.conf 修改地址段与网关等配置信息。

通过配置文件/etc/hostapd.conf 查看或者修改热点名称 imx6qap 和 WPA 密码 1234567890。

2.13.2.1 以太网-AP

1、设置以太网 IP, 配置网络防火墙:

注: 本例为以太网连接路由器说明, 配置完以太网后, 需要测试是否可以连接外网, 如果可以连接外网(方法参考“有线网卡”章节), 请继续按照操作步骤执行, 如果不可以请检查以太网或者路由器连接是否正常。

2、确保模块 8723bu 已经加载。

```
root@freescale /$ ifconfig wlan0 up /*打开 WiFi*/
```

3、开启 AP

```
root@freescale /$ wifiap.sh /*输入命令开启 HostAp*/
```

4、手机等移动终端可以通过 WiFi 连接到 FCU1201 的 AP 热点, 访问外网。

2.13.2.2 4G-AP

参考“4G 模块测试”章节中“4G-AP”部分。

2.14 RTC

RTC测试, 主要通过使用date和hwclock 工具设置软、硬件时间, 测试当操作系统重启的时候, 软件时钟读取RTC时钟是否同步。默认FCU1201已经安装了纽扣电池。

1. 查看系统时间

```
root@freescale ~$ date -u
```

2. 设置时间如下命令

```
root@freescale ~$ date -s 2018.09.11-15:01:00
```

```
Tue Sep 11 15:01:00 HKT 2018
```

3. 查看硬件时间

```
root@freescale ~$ hwclock -w
```

hwclock -w 将系统时间设置到时钟芯片里面,此时rtc 就可以使用了。如果没有这步,下次启动时,系统时间是不会更新的。

4. 断电重启系统,此时,软、硬件时间已经同步,说明RTC工作正常。

```
root@freescale ~$ date -u
Tue Sep 11 15:02:05 UTC 2018
```

5. 网络对时

```
root@freescale ~$ ntpdate cn.pool.ntp.org
4 Aug 09:25:24 ntpdate[957]: step time server 182.92.12.11 offset 1533345862.993804
sec
root@freescale ~$ date -u
Sat Aug 4 09:25:30 HKT 2018
```

注意: /etc/localtime 采用的是 Hong_Kong 的时区, /etc/timezone 为 Asia/Hong_Kong

2.15 TF 卡测试

TF 卡接口如下图:



插上 TF 卡后系统会自动将其挂载到/media 目录。同时终端会打印关于 TF 卡的信息,由于存在很多种卡,显示的信息可能会有差别。

以 16G TF 卡为例,插入 TF 卡打印信息如下:

```
root@freescale /$ mmc1: new high speed SDHC card at address aaaa
mmcblk1: mmc1:aaaa SL08G 7.40 GiB (ro)
mmcblk1: p1
```

查看 TF 卡中的文件,命令如下:

```
root@freescale /$ ls -l /media/mmcblk1p1/
-rwxr-xr-x 1 root root 8010372 Oct 29 12:30 bin
-rwxr-xr-x 1 root root 19705002 Nov 24 2018 IMX6DLRM.pdf
```

例如拷贝 TF 卡中的内容 bin 到 forlinx 路径下:

```
root@freescale /$ cp /media/mmcblk1p1/bin /forlinx/
root@freescale /$ sync
```

注意: 拷贝完之后, sync 同步, 否则拷贝不成功。

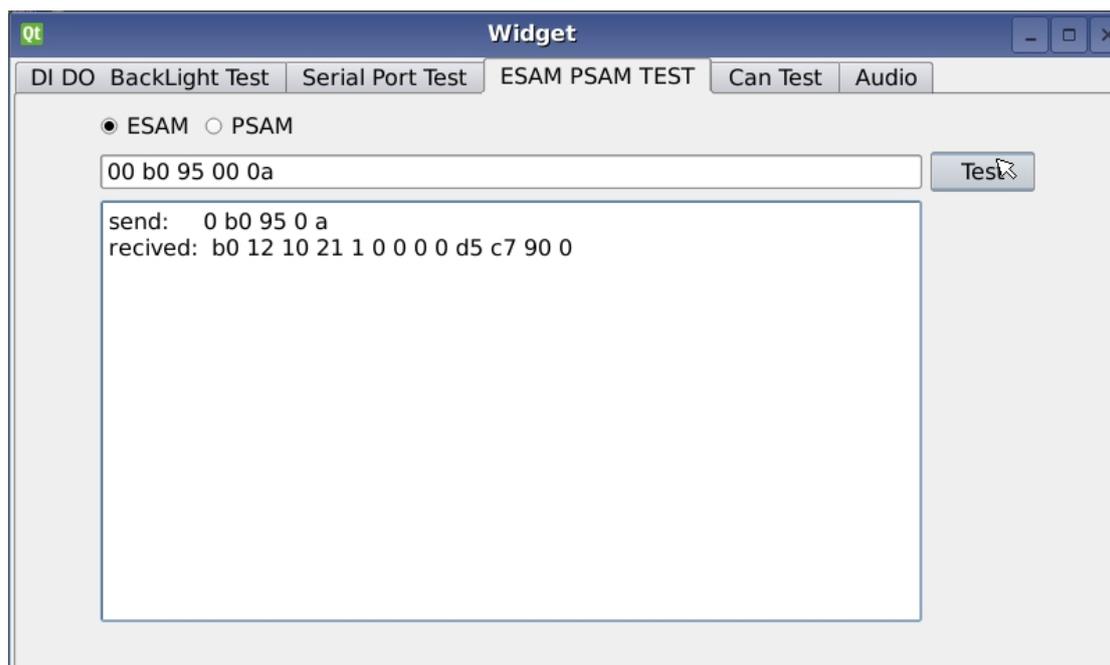
2.16 ESAM、PSAM 测试

ESAM 模块在机壳内,采用 DIP-8 封装座子, PSAM 采用标准抽屉式 mini SIM 卡槽, 按压抽屉右边的黄色按钮可弹出抽屉, 如下图:



在图形界面中,选择 ESAM PSAM Test 页面

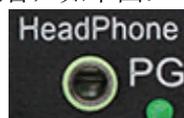
选择 ESAM 或 PSAM, 点击 Test, 在文本框内输出读取的结果



2.17 录音/放音

本机内置单声道话筒，可录音。

外置 3.5mm 标准立体声耳机插座，可放音，如下图：



2.17.1 命令行测试

用户可以使用文件系统内带的 ALSA 音频录制、播放、配置工具进行测试。

1、设置参数

```
root@freescale ~$ amixer sset Headphone 101,101           设置放音音量
root@freescale ~$ amixer sset 'Left Output Mixer PCM' on
root@freescale ~$ amixer sset 'Right Output Mixer PCM' on
root@freescale /usr/bin$ amixer cset name='Capture Volume' 63,63   设置录音音量
```

2、放音测试

```
root@freescale /usr/bin$ aplay /forlinx/sound/wo.wav       播放音频文件
Playing WAVE '/forlinx/sound/wo.wav' : Signed 16 bit Little Endian, Rate 22050 Hz, Stereo
```

3、录音测试

```
root@freescale /usr/bin$ arecord -r 44100 -f S16_LE -c 2 -d 10 record.wav   录音
Recording WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

4、播放录音

```
root@freescale /usr/bin$ aplay record.wav                 播放录音
音
Playing WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@freescale /usr/bin$
```

2.17.2 图形界面测试

本节介绍的是音频的测试，选择 Audio 页面。

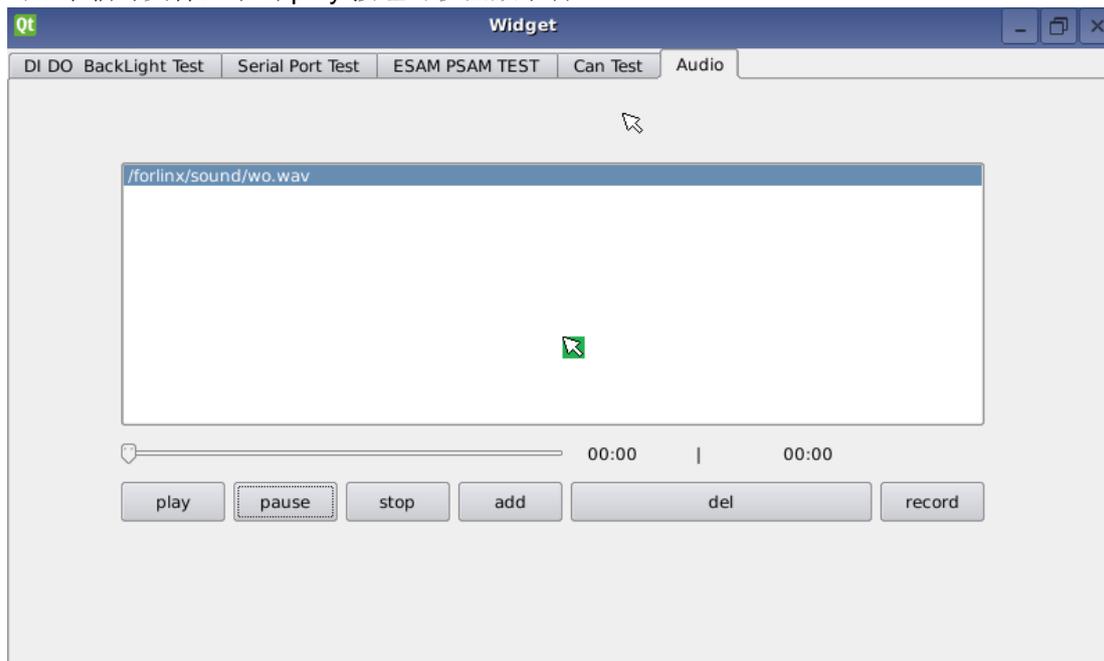
放音测试：

将耳机插入开发板的 PHONE 插孔，点击 play 按钮，若能听到音乐声音，则放音正常

录音测试：

单击 record 按钮，然后对着 MIC 说话，持续数秒后单击 stop 按钮，之后会看到歌曲列表里

多出一个临时文件，单击 play 按钮可以回放录音



第三章 系统固件更新

本产品更新固件方法有：OTG 更新固件，TF 卡更新固件。

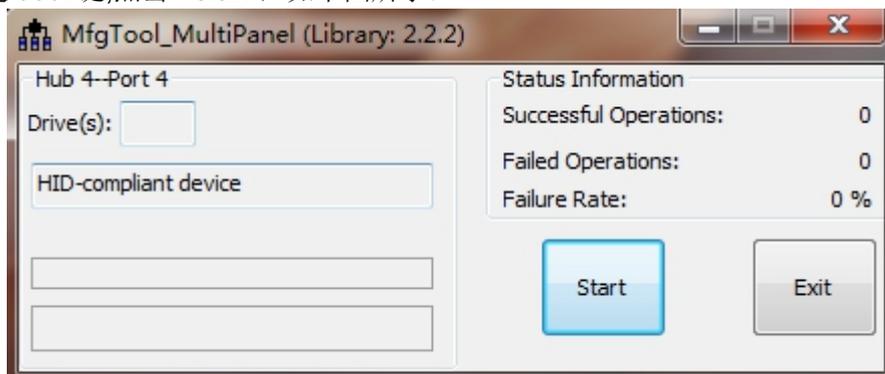
3.1 使用烧写工具烧写镜像

- 1、把 micro usb 数据线连接到开发板的 otg 接口，数据线的另一端连接到 pc 的 usb 接口。
- 2、打开 Linux\工具\USB OTG\mfgtools (win7 用户请使用管理员权限打开)
mfgtool2-qt-OKMX6-C-emmc.vbs (用于 1g 内存 qt 的烧写)

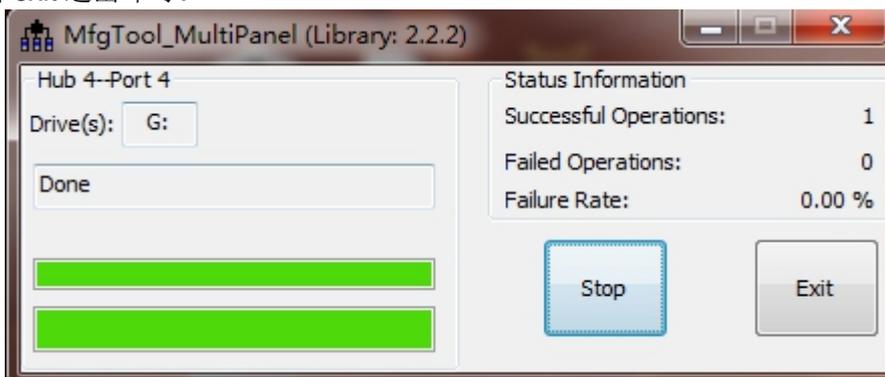
Linux\工具\USB OTG\mfgtools。

3.1.1 烧写 Qt 4.8.6 GUI linux

- 1、烧写时，确保其烧写工具目录下\Profiles\Linux\OS Firmware\files \linux\下有 ulmage, logo.bmp, u-boot-q.bin (四核)，u-boot-dl.bin (双核) 文件,同时还要求\...\OS Firmware\files\linux\qt4\下有 rootfs.qt4.tar.bz2 文件。
- 2、1g 内存使用 mfgtool2-qt-OKMX6-C-emmc.vbs 进行烧写，进行烧写。此时已经配置好烧写文本
- 3、给设备上电，按住 boot 键的同时按下 reset 键，松开 reset 后等烧写工具识别 HID 设备后可以松开 BOOT 键。
- 4、首次升级过程中，会通过网络自动安装驱动，等安装完成后，烧写工具里出现设备的名称(HID-compliant device)后，抬起 boot 键,点击 "start", 如下图所示：



- 5、中间弹出格式化对话框，点击“取消”格式化选项，或者**不管它**，直到烧写完成，看见 **DONE** 之后先点击 stop，再点击 exit 退出即可。



- 6、烧写完成后，复位或者重新上电启动即可。

注意：

1. 多次烧写时，为保证烧写稳定性，每次烧写完后都需要关闭烧写软件再重新打开。
2. 最好在 win7 系统下烧写，调试串口是 **DEBUG (UART1)** 口。

3.2 TF 卡更新固件

3.2.1 制作 TF 卡

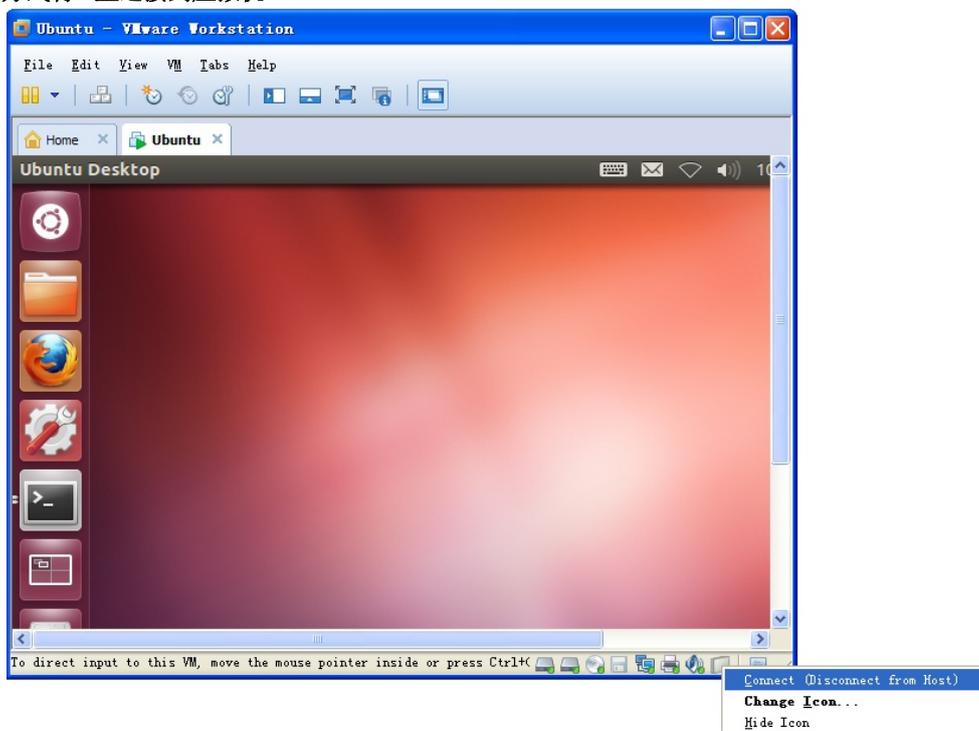
如果处理器使用 i.MX6Dual Lite 则使用 imx6dl-sd-linux-tools.tar.bz2 脚本，若处理器为 i.MX6Quad

则使用 imx6q-sd-linux-tools.tar.bz2。以 imx6dl-sd-linux-tools.tar.bz2 为例说明，将 imx6dl-sd-linux-tools.tar.bz2 拷贝到到 ubuntu 系统的任一目录，假设为/home/forlinx/work；
Linux\工具\TF。

步骤 1: 解压 imx6dl-sd-linux-tools.tar.bz2，命令如下。

```
cd /home/forlinx/work/  
tar xvf imx6dl-sd-linux-tools.tar.bz2
```

步骤 2: 使用 USB 读卡器把TF卡插入到电脑的USB 端口（VMware 虚拟机用户如果U盘没有被虚拟机识别，可以使用如下方式将U盘连接到虚拟机）。



步骤 3: 进入 imx6dl-sd-linux-tools 目录

```
cd /home/forlinx/work/imx6dl-sd-linux-tools
```

执行脚本：

```
./createSdcard.sh
```

执行上述命令后，终端会列出电脑的硬盘或优盘，对应选择自己的TF 卡，然后回车。

注意：判定自己的U盘是 sda/sdb/sdc 可以根据容量进行判断，比如自己的U盘容量为4G，则其size 为 3872256 字节 \approx 4G，建议用户执行此操作时不要同时插入多个优盘，以免混淆。

这里以我们的操作为例：

选择1，回车

```
#####  
  
This script will create a bootable SD card from custom or pre-built binaries.  
  
The script must be run with root permissions and from the bin directory of  
the SDK  
  
Example:  
$ sudo ./create-sdcard.sh  
  
Formatting can be skipped if the SD card is already formatted and  
partitioned properly.  
  
#####  
  
Available Drives to write images to:  
  
# major minor size name  
1: 8 16 3872256 sdb  
  
Enter Device Number: █
```

格式化，选择y，回车，完成格式化。

```
# major minor size name  
1: 8 16 3872256 sdb  
  
Enter Device Number: 1  
  
sdb was selected  
  
Checking the device is unmounted  
unmounted /dev/sdb1  
  
sdb1 sdb2 sdb3  
3775275  
  
#####  
  
Detected device has 1 partitions already  
  
Re-partitioning will allow the choice of 1 partitions  
  
#####  
Would you like to re-partition the drive anyways [y/n] : █
```

格式化完成，如图：

```
Copying boot partition
untar update.tar.bz2 to boot partition
sdrun/
sdrun/uImage
sdrun/logo.bmp
sdrun/u-boot.bin
sdrun/ramdisk.img.u
target/
target/uImage
target/rootfs.tar.bz2
target/logo.bmp
target/u-boot.bin
Buring th u-boot.bin to sdcard
500+0 records in
500+0 records out
512000 bytes (512 kB, 500 KiB) copied, 1.29715 s, 395 kB/s
201+1 records in
201+1 records out
206064 bytes (206 kB, 201 KiB) copied, 1.23928 s, 166 kB/s
0+1 records in
0+1 records out
768072 bytes (768 kB, 750 KiB) copied, 2.55958 s, 300 kB/s

Syncing....

Un-mount the partitions

Remove created temp directories

Operation Finished
```

卡制作完成后可以看到 boot 分区包含 sdrun 和 target 两个目录。sdrun 文件夹内容用于引导系统烧写，无需修改；target 目录内容会烧写到 flash 芯片。如果需要替换镜像文件，只需将 target 目录中的对应文件替换掉，并保持同样的命名，之后再重新进行系统烧写。

3.2.2 TF 卡更新系统

将上一节中制作好的 TF 卡插入 FCU1201 的 TF 卡槽中，给设备上电，按住 boot 键的同时按下 reset 键，松开 reset 后再抬起 boot 键。TF 卡中新的固件会自动更新到 FCU1201 中。更新时间较长。可以从屏幕上看到更新过程。

FCU1201 重新上电或者复位。即可正常使用。

第四章 FCU1201 Linux 编译

请不要跳过这段话:

开发环境是开发人员在开发过程当中,所需的软硬件平台。开发环境并不是一个固定的样式,在这里,我们详细讲解一个嵌入式 Linux 开发环境搭建的方法。您已经对嵌入式开发非常了解的话,可以按照自己的需求来搭建环境。如果和本手册环境不一样而产生报错,您可以从国内一些大 Linux 论坛和网站搜索相关的信息来解决。本册介绍的环境经过飞凌的测试,如果对嵌入式开发不是非常熟悉的朋友,希望您按照飞凌提供的方法来搭建环境。Ubuntu 的安装与设置工作请参考[虚拟机 Ubuntu 的安装说明 Rev1.1](#)手册,该手册可在用户资料与论坛资料下载区获得。

4.1 版本说明

- 操作系统: Ubuntu12.04 64 位版
- 交叉工具链: arm-fsl-linux-gnueabi-gcc-4.6.2
- Bootloader 版本: u-boot-2009.08
- 内核版本: linux3.0.35

4.2 编译环境说明

- Linux PC: 在 PC 上直接安装 ubuntu12.04 作为编译的主机,这样做的好处是编译速度快、稳定。
- Vmware8.0+ubuntu12.04: 在 XP 下安装虚拟机,并在虚拟机下进行编译。这种方式的好处是 1 台电脑就可以搭建开发环境,编译、烧写都在同一台电脑完成,缺点是编译速度太慢,各种网络服务不稳定。
- 另外,本公司的网盘中提供了带编译器的虚拟机,可作为参考。

4.3 编译环境搭建

在按照[虚拟机 Ubuntu 的安装说明](#)手册完成虚拟机的安装与设置后,我们还需要配置 Linux 的开发环境。

4.3.1 安装交叉编译器

步骤 1: 将文件 [gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12.tar.gz](#) 拷贝到 Ubuntu 主目录下

 [FCU1201 嵌入式控制单元 Linux3.0.35 用户资料\工具\gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12.tar.gz](#)

步骤 2: 在 Ubuntu 中新建一个终端,输入下面的命令安装交叉编译器:

```
#cd (进入主目录)
#mkdir -p /opt/freescale/usr/local (创建目录,若目录已存在会提示错误,跳过即可)
#tar zxvf gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12.tar.gz -C /opt/freescale/usr/local (编译器解压到/opt/freescale/usr/local)
```

步骤 3: 查看 gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12 是否解压成功

```
#ls -l /opt/freescale/usr/local
```

如可以看到 gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12 文件夹即解压成功

步骤 4: 添加环境变量到 Profile,执行命令打开编辑 Profile 后,重启电脑

```
#gedit /etc/profile
```

最后一行添加以下内容

```
export ARCH=arm
export CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-
export PATH=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin:$PATH
```

之后执行命令:

```
#source /etc/profile
```

来使得刚配置的环境变量生效。

步骤 5: 在终端里面执行以下命令,验证交叉编译器安装是否成功

```
#arm-fsl-linux-gnueabi-gcc -v
```

下图为安装成功提示

4.4 编译

本章说明了 U-boot 和 Linux 内核在 PC Linux 的编译方法。在进行本章实验之前，搭建好开发环境。

4.4.1 编译 u-boot

编译 u-boot

步骤 1: 四核 u-boot 源码在用户资料中路径：

 FCU1201 嵌入式控制单元 Linux3.0.35 用户资料\Linux\源码\uboot2009-08.tar.gz

步骤 2: 解压缩 u-boot 源码，并进入 u-boot 源码根目录，命令如下。

```
#tar zxvf uboot2009-08.tar.gz
```

```
#cd uboot2009-08
```

执行以下命令设置环境变量：

```
#export ARCH=arm
```

```
#export CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-
```

```
#export PATH=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin:$PATH
```

清除编译产生的中间文件

```
make distclean
```

四核设置配置参数（在编译四核 1G 内存 u-boot 时执行）

```
make okmx6q_c_config
```

双核设置配置参数（在编译双核 1G 内存 u-boot 时执行）

```
make okmx6dl_c_config
```

编译生成 u-boot

```
make -j8
```

步骤 3: 编译烧写到 emmc 所用 u-boot 映像 u-boot.bin。

编译成功后，将在 ‘uboot2009-08’ 目录下产生名为 ‘u-boot.bin’ 的二进制文件。该文件即我们需要文件夹下 u-boot。

4.4.2 编译 Linux-3.0.35

➤ 解压源码

内核源码在用户资料中路径如下：

 FCU1201 嵌入式控制单元 Linux3.0.35 用户资料\Linux\源码\linux-3.0.35.tar.gz

将压缩包 ‘linux-3.0.35.tar.gz’ 拷贝到你的工作目录下，解压缩：

```
#tar zxvf linux-3.0.35.tar.gz
```

然后进入内核源码根目录：

```
#cd linux-3.0.35
```

➤ 设置环境变量

执行以下命令设置环境变量：

```
#export ARCH=arm
```

```
#export CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-
```

```
#export PATH=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin:$PATH
```

注：以上是分别以 “#” 开头的三条命令，命令内部没有换行。

➤ 配置内核

步骤 1: 首先你需要复制 imx6_defconfig 到 .config 以方便你配置你的内核，命令如下。

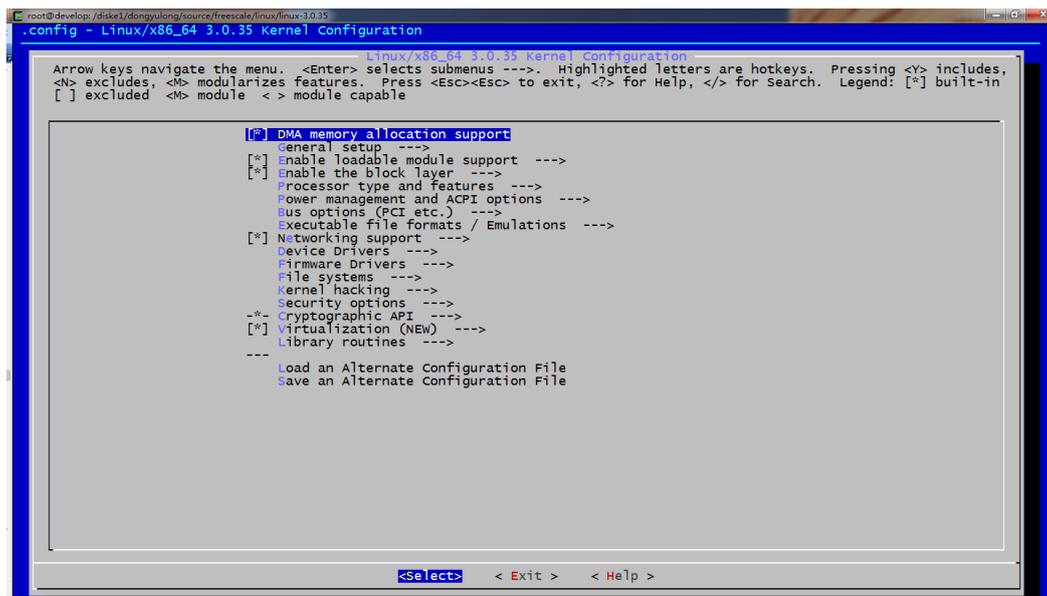
```
#make distclean
```

```
#cp arch/arm/configs/imx6_c_defconfig .config
```

步骤 2: 开始配置内核，命令如下。

```
#make menuconfig
```

步骤 3: 完成配置内核，选择 exit 保存退出



➤ 编译内核

编译内核命令如下：

```
#make ulmage
```

```
Image Name: Linux-3.0.35-2666-gbdde708-g65ca
Created: Thu May 14 14:08:22 2015
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 4006912 Bytes = 3913.00 kB = 3.82 MB
Load Address: 10008000
Entry Point: 10008000
Image arch/arm/boot/uImage is ready
```

编译结束后将在内核源码目录的 `arch/arm/boot` 中得到 Linux 内核映像文件 `ulmage`。

```
root@develop:/disk1/dongyulong/source/freescale/linux/linux-3.0.35# ls arch/arm/boot/
bootp compressed Image install.sh Makefile uImage zImage
root@develop:/disk1/dongyulong/source/freescale/linux/linux-3.0.35#
```

➤ 编译模块

```
#make modules
```

➤ 安装模块

将模块导出到目录，如目录 `/home/forlinux/work`

```
#make modules_install INSTALL_MOD_PATH=/home/forlinux/work
```

将导出的模块压缩打包

```
# tar jcvf lib.tar.bz2 lib
```

制作文件系统时，将其放入文件系统，参考文件系统的制作。

4.4.3 制作 rootfs qt4.8.6

步骤 1: 将文件系统源文件 ‘`rootfs.qt4.tar.bz2`’ 拷贝到 Ubuntu 的根目录下，源文件在用户资料中的路径如下：

 `FCU1201 嵌入式控制单元 Linux3.0.35 用户资料\Linux\镜像\filesystem\qt4\rootfs.qt4.tar.bz2`

步骤 2: 解压文件系统源文件

```
#mkdir rootfs
```

```
#tar jxvf rootfs.qt4.tar.bz2 -C rootfs
```

```
#cd rootfs
```

步骤 3: 开始制作 `rootfs` 文件系统，制作命令：

```
#rm -fr lib/modules
```

```
#tar xvf /home/forlinux/work/lib.tar.bz2 -C /
```

```
#rm rootfs.tar.bz2
```

```
#./pack-rootfs.sh
```

```
root@develop:/disk1/dongyulong/source/freescale/rootfs/rootfs-cmdline#  
root@develop:/disk1/dongyulong/source/freescale/rootfs/rootfs-cmdline#  
root@develop:/disk1/dongyulong/source/freescale/rootfs/rootfs-cmdline#  
root@develop:/disk1/dongyulong/source/freescale/rootfs/rootfs-cmdline#  
root@develop:/disk1/dongyulong/source/freescale/rootfs/rootfs-cmdline#  
root@develop:/disk1/dongyulong/source/freescale/rootfs/rootfs-cmdline#  
root@develop:/disk1/dongyulong/source/freescale/rootfs/rootfs-cmdline# ./pack-rootfs.sh
```

生成 `rootfs.tar.bz2`,最后修改名称为 `rootfs.qt4.tar.bz2`,此文件是可以下载到开发板 `emmc` 中的文件系统映像。

Qt 开机自启动脚本在 `etc/rc.d/apps.sh`。

附录一 外壳尺寸图

下图为 FCU1201 外壳尺寸图单位 mm:

