



IAC-IMX8MP-Kit 功能说明与测试手册

版本号：V2.1
2022年02月

浙江启扬智能科技有限公司版权所有

QIYANG TECHNOLOGY Co., Ltd

Copyright Reserved

版本更新记录

版本	硬件平台	描述	日期	修订人
1.0	IAC-IMX8MM-MB V1.00	内部版本	2021-09	wangtg
2.0	IAC-IMX8MM-MB V1.20	更新图片	2021-11	wangtg
2.1	IAC-IMX8MM-MB V2.00	更新图片	2022-02	wwx

目 录

目 录.....	3
阅读前须知：本手册主要介绍接口功能测试.....	4
一、前言.....	4
公司简介.....	4
二、准备工作.....	5
三、主板测试.....	7
2.1、显示测试.....	7
2.1.1 HDMI 显示.....	7
2.1.2 MIPI-DSI 显示.....	7
2.1.3 LVDS 显示.....	10
2.2、触摸测试.....	13
2.3、串口测试.....	14
2.4、CAN 测试.....	17
2.5、USB 测试.....	23
2.6、TF 测试.....	25
2.7、音频测试.....	27
2.8、以太网测试.....	28
2.9、WIFI 测试.....	30
2.10、5G 测试.....	32
2.11、SSD 测试.....	36
2.12、时钟测试.....	39
2.13、看门狗测试.....	41
2.14、GPIO 测试.....	错误！未定义书签。
2.15、摄像头测试.....	42
四、测试小结.....	44

阅读前须知：本手册主要介绍接口功能测试

一、前言

公司简介

浙江启扬智能科技有限公司 2007 年成立于杭州，是一家专注于 ARM 嵌入式产品研发、生产与销售的国家高新技术企业。10 余年的积累与沉淀，成功构建了产品从开发到量产的服务链。

作为公司的核心，启扬研发团队由 30 余位嵌入式工程师组成，致力于为用户提供简单易用的嵌入式硬件、软件工具以及定制化的产品解决方案。已广泛应用于工控、物联网、新零售、医疗、电力、环境监测、充电桩等领域。

设立于诸暨的生产基地为启扬提供了强有力的保障，占地面积 5000 平米，拥有 2 条 SMT 产线，通过并严格遵循 ISO9001 质量管理体系认证指导生产。依托雄厚的生产实力，年产能可达 100 万套，保证用户交期，解决后顾之忧。

启扬拥有完善的销售市场网络，专业的销售和售后团队为用户提供全方位的技术支持与服务。业务已遍及 120 多个国家和地区，成功帮助 2000 多家用户将产品快速高效地推向市场。

研发、产能、市场的结合与延伸，为启扬智能成为专业化、全球化的嵌入式软硬件供应商奠定了坚实的基础。

我们为您提供：

- **多平台软/硬件产品**

NXP、Rockchip、MTK、Renesas、TI、Atmel、Cirrus Logic 等多平台 ARM 开发板/核心板/工控板和周边硬件产品以及支持用户快速二次开发的配套工具与软件资源。

- **定制服务**

充分发挥在 ARM 平台及 Linux、Android、Ubuntu 操作系统上的技术累积，为用户提供量身定制嵌入式产品服务（OEM/ODM）。

感谢您使用启扬智能的产品，我们会尽最大努力为您提供技术协助！祝愿您工作顺利！

二、准备工作

测试前可先查看《IAC-IMX8MP-Kit 硬件说明书》、《IAC-IMX8MP-Kit Linux 用户手册》。

出厂前已烧写 Linux 操作系统（Linux5.10.35 系统），可直接使用测试。

串口调试：

可参照《IAC-IMX8MP-Kit Linux 用户手册》串口调试。

开发板上电，正常连接调试串口，可通过调试串口进入到板子的文件系统中。

```

Starting user database manager...
[ OK ] Started WPA supplicant.
[ OK ] Finished File System Check on /dev/mmcblk2p1.
Mounting /run/media/mmcblk2p1...
[ OK ] Mounted /run/media/mmcblk2p1.
[ OK ] Started User Database Manager.
[ OK ] Started Hostname Service.
[ OK ] Created slice User Slice of UID 0.
Starting User Runtime Directory /run/user/0...
[ OK ] Finished User Runtime Directory /run/user/0.
Starting User Manager for UID 0...
[ 8.914032] audit: type=1006 audit(1616581522.092:2): pid=493 uid=0 old-auid=4294967295 a
es=1 res=1
[ OK ] Started User Manager for UID 0.
[ OK ] Started Session c1 of user root.
[ 9.328925] audit: type=1006 audit(1616581522.504:3): pid=484 uid=0 old-auid=4294967295 a
=2 res=1
[ 9.540227] alloc_contig_range: [66600, 66858) PFNs busy
[ 9.547454] alloc_contig_range: [66700, 66958) PFNs busy
[ OK ] Started weston, a wayland 欵positor, as a system service.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Finished Update UTMP about System Runlevel Changes.
[ 12.793526] random: crng init done
[ 12.796956] random: 7 urandom warning(s) missed due to ratelimiting
[ OK ] Finished Load/Save Random Seed.
[ OK ] Stopped ISP i.MX 8Mplus daemon.
[ OK ] Started ISP i.MX 8Mplus daemon.

NXP i.MX Release Distro 5.10-hardknott imx8mpevk ttyMX1
imx8mpevk login: root
root@imx8mpevk:~#
    
```

login: root 登录

主板测试程序位于/usr/test 目录下，进入该目录，后续测试都在该目录下进行。

```
cd /usr/test/
```

```
ls
```

```

imx8mpevk login: root
root@imx8mpevk:~# cd /usr/test/
root@imx8mpevk:/usr/test# ls
1.mp4          UART_scan.sh  rs485_test  test.wav      watchdog_notfeed_test
Imx8_qt_test  rs232_test   rtc_test    watchdog_feed_test  xyn44.wav
root@imx8mpevk:/usr/test#
    
```

Uboot 环境:

在前期开发测试时，需要在 uboot 环境测试设置相关参数，如液晶屏参数等。

主板启动，在 3 秒内按回车键（电脑），可进入 uboot 设置环境，输入 print 打印基本参数:

```
U-Boot 2021.04-00004-gcb364d6 (Sep 08 2021 - 16:10:49 +0800)
```

```
CPU: i.MX8MP[8] rev1.1 1600 MHz (running at 1200 MHz)
CPU: Industrial temperature grade (-40C to 105C) at 38C
Reset cause: POR
Model: QIYANG i.MX8MPplus LPDDR4 board
DRAM: 2 GiB
MMC: FSL_SDHC: 1, FSL_SDHC: 2
Loading Environment from MMC... OK
In: serial
Out: serial
Err: serial
SEC0: RNG instantiated
```

```
BuildInfo:
- ATF ec35fef
```

```
switch to partitions #0, OK
mmc2(part 0) is current device
flash target is MMC:2
Net: Could not get PHY for FEC0: addr 3
Could not get PHY for FEC0: addr 3
eqos_probe(dev=00000000bdf1f610):
eth1: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot: 0
u-boot=>
u-boot=>
```

三、主板测试

2.1、显示测试

IAC-IMX8MP-Kit 开发板支持 HDMI、MIPI-DSI、LVDS 多种显示接口；系统默认移植有 Qt5.15 相关库，该程序将带您走进 Qt 的世界。

2.1.1 HDMI 显示

测试 HDMI 显示前，您还需要准备 HDMI 显示器及 HDMI 连接线；

测试步骤和结果：

开发板默认启动为 HDMI 显示；

运行 qt 测试程序

```
#!/mx8_qt_test
```

此时 HDMI 显示器会显示 Qt 界面。

设备树文件：

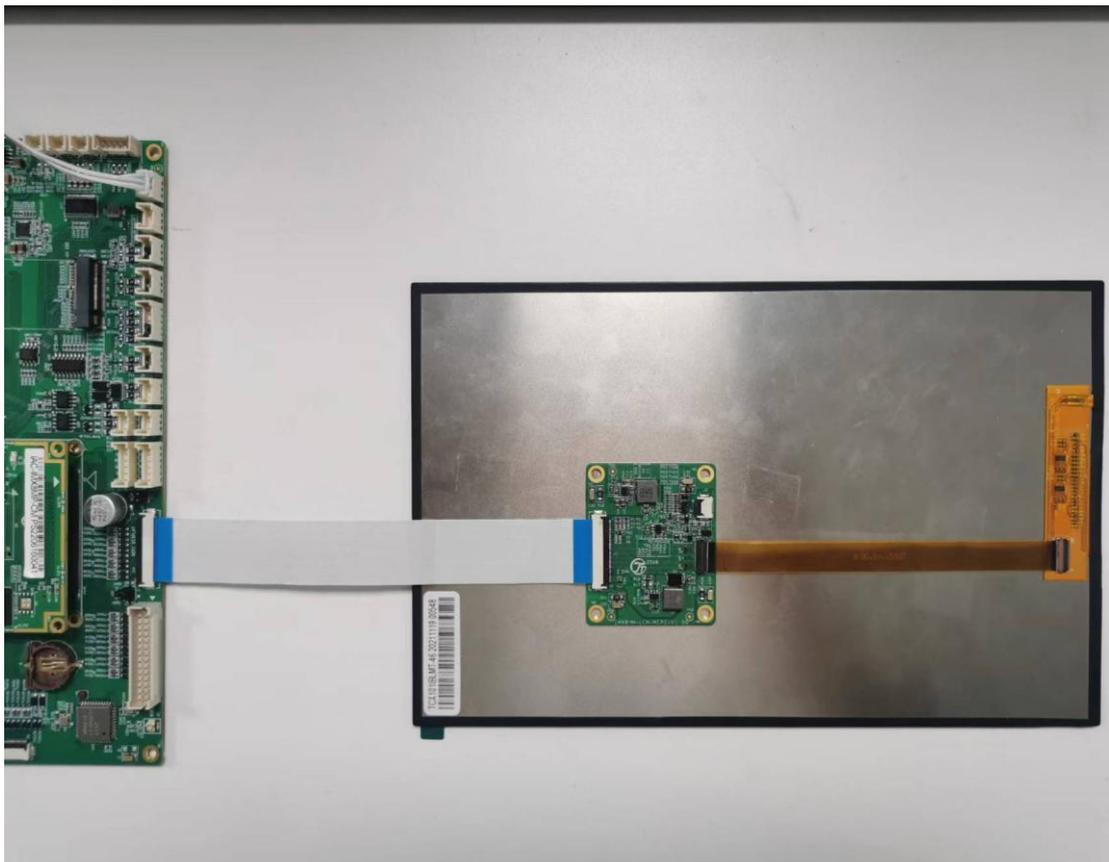
```
arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts
```

2.1.2 MIPI-DSI 显示

支持启扬配套 10.1 寸 MIPI 显示屏（电容触摸），型号 GN101IB27811433，分辨率 800x1280。需要单独采购。

MIPI-DSI 显示接口-J14（40Pin0.5mm 间距 FPC 插座），接口包含 MIPI-DSI 信号、I2C 信号；还需要通过转接版本与液晶屏进行连接。

主板与液晶屏连接如下图：（注意 FPC 线连接方向）



有任何技术问题或需要帮助，请联系：supports@qiyangtech.com

第 8 页 共 45 页

购买产品，请联系销售：sales@qiyangtech.com

更多信息请访问：<http://www.qiytech.com>

©2021 Qiyangtech 版权所有

测试步骤和结果:

1.开发板启动时候在 u-boot 输入如下命令

u-boot=> **setenv fdtfile imx8mp-qiyang-gnl.dtb**

u-boot=> **saveenv**

u-boot=> **boot**

```
Hit any key to stop autoboot: 0
u-boot=> setenv fdtfile imx8mp-qiyang-gnl.dtb
u-boot=> boot
```

2.运行 qt 测试程序

./Imx8_qt_test

此时液晶屏会显示 Qt 界面，如下图：补图



3.播放视频

gplay-1.0 /usr/test/1.mp4

设备树文件:

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

2.1.3 LVDS 显示

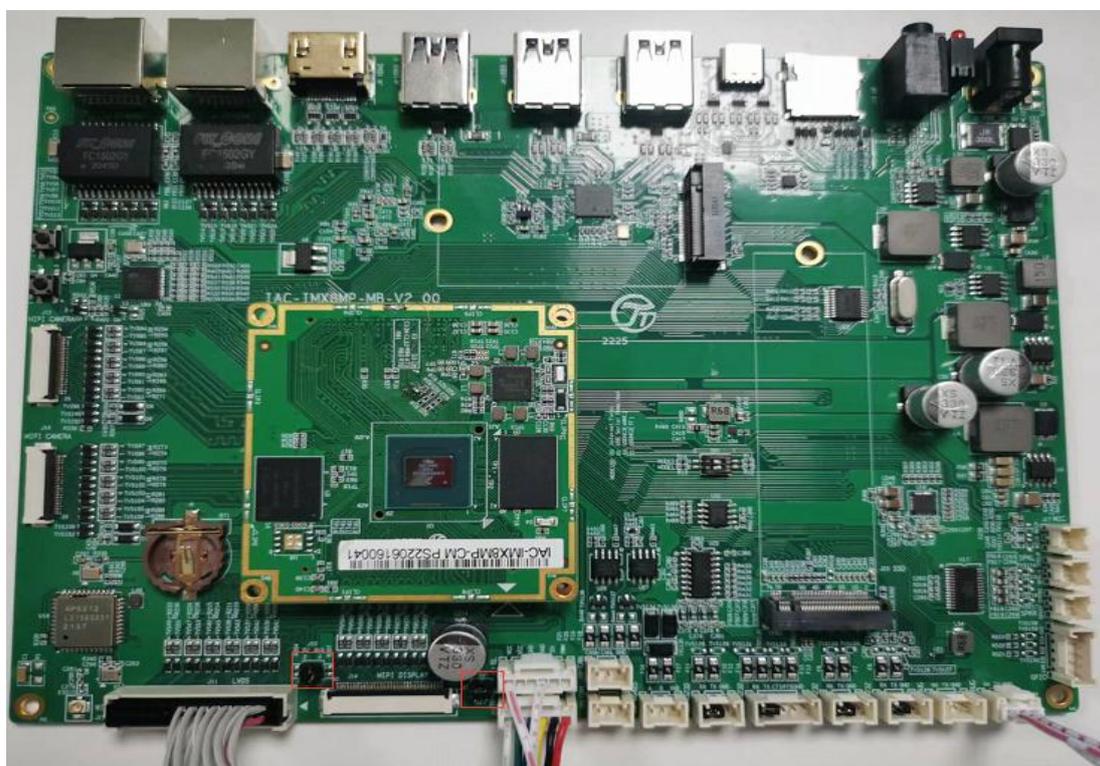
支持启扬配套 7 寸 LVDS 显示屏(电容触摸), 型号 QY-HJ070NA-V1.2, 分辨率 1024x600。需要单独采购。

LVDS 显示屏 LVDS 信号接口-J11, (注意 J10 供电接口跳线, 跳 3.3V)

LVDS 显示屏背光供电接口-J13, (注意 J37 供电接口跳线, 跳 5V)

I2C 电容触摸屏接口-J12。

主板与液晶屏连接如下图:





测试步骤和结果:

1.开发板启动时候在 u-boot 输入如下命令

```
u-boot=> setenv fdtfile imx8mp-qiyang-HJ070NA.dtb
```

```
u-boot=> saveenv
```

```
u-boot=> boot
```

```
Hit any key to stop autoboot: 0
u-boot=> setenv fdtfile imx8mp-qiyang-gnl.dtb
u-boot=> boot
```

2.运行 qt 测试程序

```
# ./Imx8_qt_test
```

此时液晶屏会显示 Qt 界面，如下图:



3.播放视频

`gplay-1.0 /usr/test/1.mp4`

设备树文件:

`arch/arm64/boot/dts/freescale/imx8mp-qiyang-HJ070NA.dts`

2.2、触摸测试

IAC-IMX8MP-Kit 开发板支持 MIP-DSI 显示和 LVDS 显示支持电容屏触摸(I2C 接口)。

测试原理:

打开 QT 测试程序, 点击屏幕、屏幕显示正常; 因为 HDMI 显示没有外接触摸屏。
本测试针对 MIPI-DSI 显示。

测试步骤和结果:

1.运行 qt 测试程序

```
# ./Imx8_qt_test
```

触摸正常, 无明显偏差

设备树文件:

```
arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts
```

驱动代码:

```
drivers/input/touchscreen/goodix.c
```

内核对应选项:

```
CONFIG_TOUCHSCREEN_GOODIX=y
```

2.3、串口测试

IAC-IMX8MP-Kit 开发板对外提供 7 路串口，其中 2 路调试串口 (RS232)，4 路 RS232，1 路 RS485。

对应接口说明：

串口	对应接口说明	设备节点
DEBUG (A53 调试串口)	J24	/dev/ttymx1
DEBUG (M4 调试串口)	J25	
COM1 (485 串口)	J30	/dev/ttymx0
COM1 (232 串口)	J26 (三线制)	/dev/ttyWK0
COM2 (232 串口)	J27 (三线制)	/dev/ttyWK1
COM3 (232 串口)	J28 (三线制)	/dev/ttyWK2
COM4 (232 串口)	J29 (五线制)	/dev/ttymx2

其中 J26~J27 为 SPI 总线扩展串口。

测试原理：

测试程序实现了一个串口每隔 1s 发送字符数据 `"/dev/ttyXXXX" test string!`，其中 x 为实际测试的设备节点，同时通过多线程方式，阻塞读取串口数据并打印

测试步骤和结果：

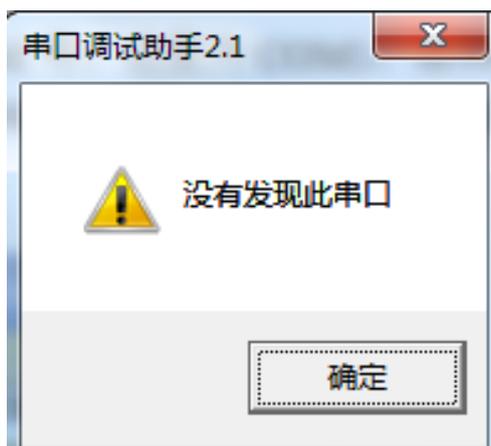
串口测试时 PC 机需要用到两个串口：

- ①、一个接调试串口，用于交互
- ②、一个接待测串口，用于测试串口收发数据

根据串口和硬件关系表，选择要测试的串口，通过提供的专用串口转接线，连接要测试的串口到 PC 机串口上。

PC 机打开光盘中的串口调试工具。

如果打开之后提示如下：

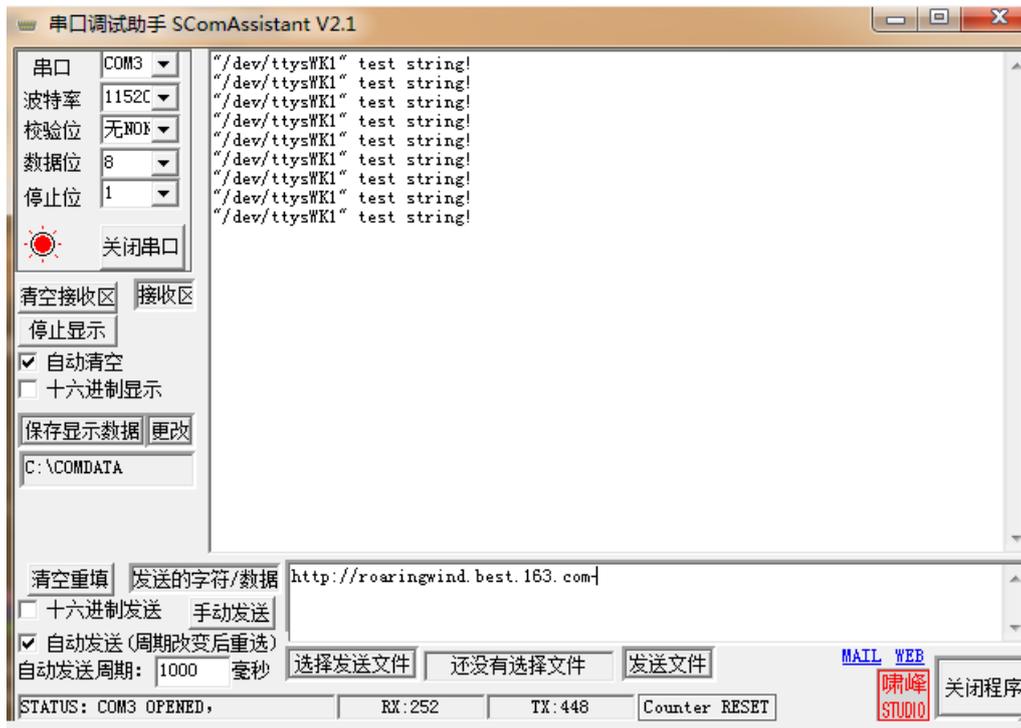


表示默认的 PC 机的 COM 口被终端占用了，关闭被占用的终端，然后重新打开串口调试工具。

设置串口属性，串口对应 PC 机的 COM 号，这里为 COM3，波特率为 115200，数据位为 8 位，停止位为 1，奇偶校验为 NONE。



串口连接好并设置好之后就可以开始测试了
 分别测试 COM1、COM2、COM3、COM4 这 4 个串口
 这里以 COM2 为例作介绍，其他串口测试方法一样
`# ./rs232_test /dev/ttyWK1 115200`



调试串口收到数据

```

root@imx8mmqiyang:/usr/test# ./rs232_test /dev/ttysWK1 115200
receive 32 datas: http://roaringwind.best.163.com
    
```

因为 485 的流控脚由硬件来控制，使用这边 485 测试的方法和 232 是一样的。

设备树文件:

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

2.4、CAN 测试

IAC-IMX8MP-Kit 开发板共有 2 路 CAN 接口。

对应接口说明：

CAN 接口	对应接口说明	设备节点
CAN0	J31(CANH、CANL、GND 对应 1、2、3 脚)	CAN0
CAN1	J32(CANH、CANL、GND 对应 1、2、3 脚)	CAN1

测试原理：

利用文件系统中提供的测试 CAN 的工具进行 CAN 测试。

测试步骤和结果：

这里需要准备 CAN 分析仪，下面测试以开发板与 CAN 分析仪测试为例。

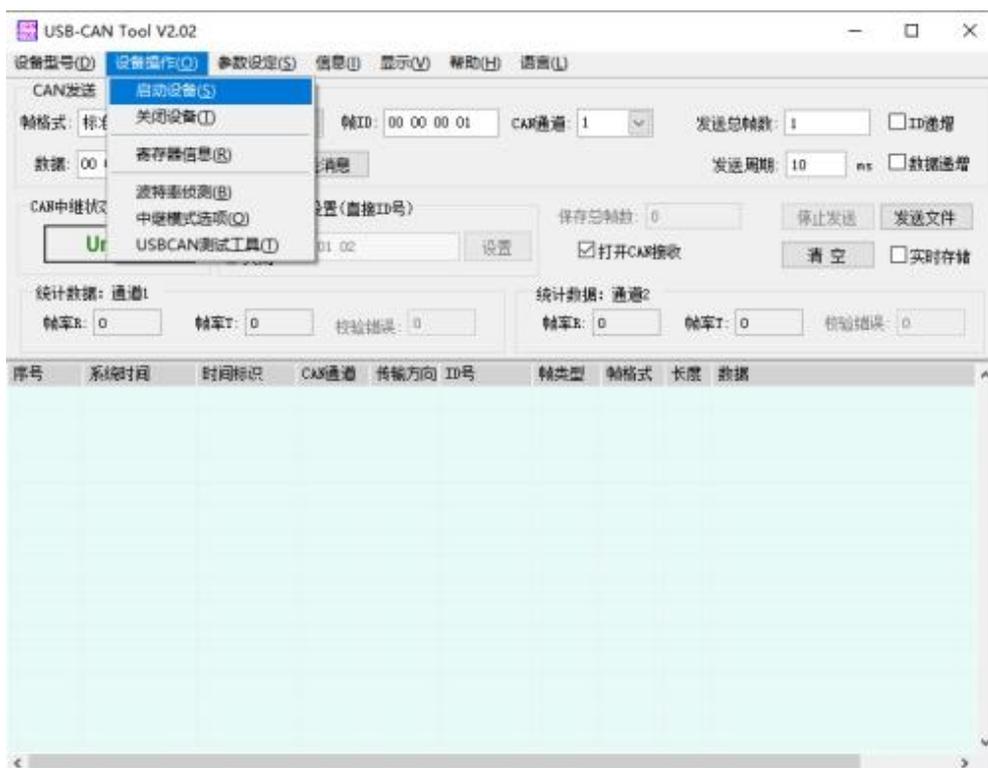


其它 CAN 分析仪设备均可。

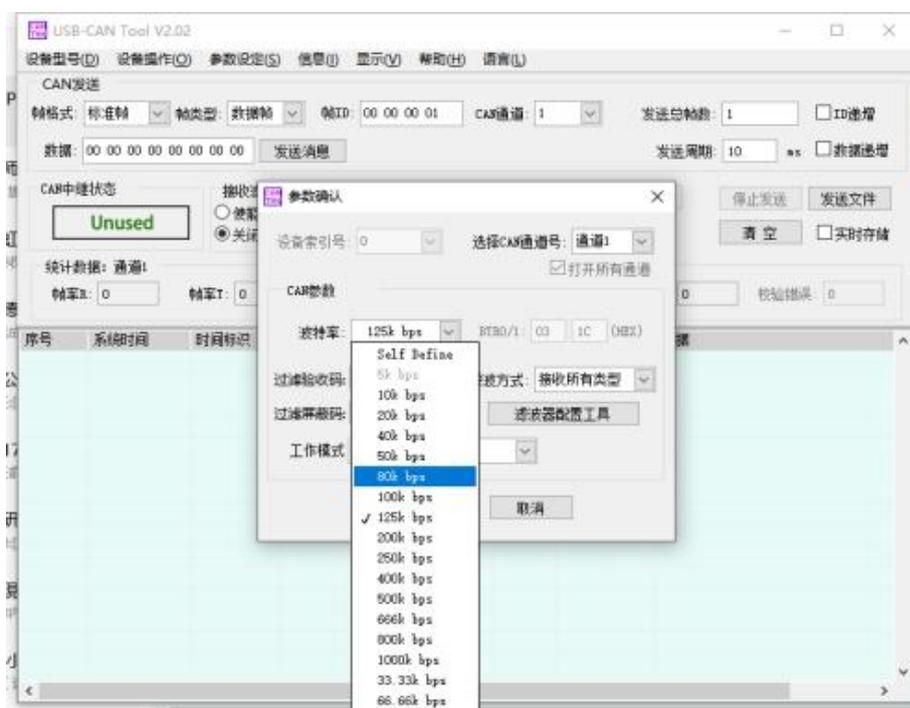
1.将开发板上的 CAN 接口 (CAN0-J 31), 连接到 CAN 分析仪及其它板子上的 CAN 接口 (**H 对 H, L 对 L**)

2. 利用 CAN 分析接收板子传过来的数据

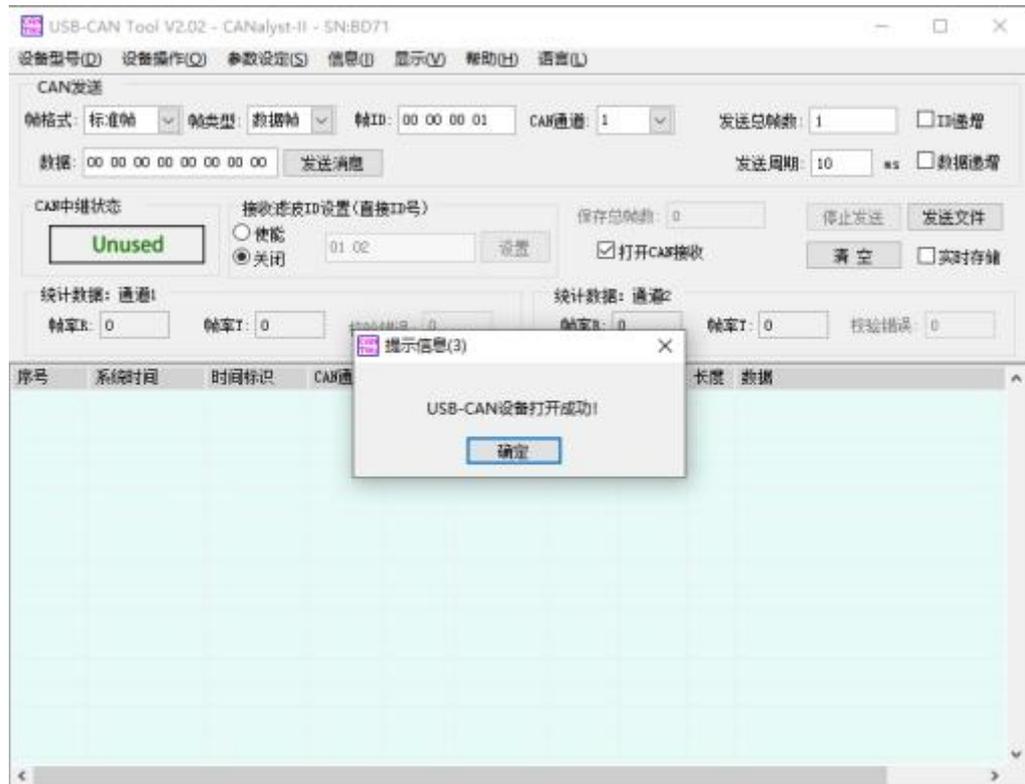
2.1 打开 CAN 分析仪的软件并设置 CAN 的参数



将波特率设置为 125k bps，并选择通道 1（根据实际情况选择）



弹出以下窗口即可



2.2 设置板子上 CAN 的参数并启动

```
# ip link set can0 type can bitrate 125000
# ifconfig can0 up
# ifconfig
```

```
root@qiyang:~# ip link set can0 type can bitrate 125000
root@qiyang:~# ifconfig can0 up
flexcan 2090000.flexcan can0: writing ctrl=0x0e312055
root@qiyang:~# ifconfig
can0      Link encap:UNSPEC  Hwaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP MTU:16 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B) TX bytes:6 (6.0 B)
          Interrupt:34
```

2.3 通过板子的 CAN，将数据传输到 CAN 分析仪中

```
# cansend can0 111#112233445566
# cansend can0 111#112233445577
```

```
root@qiyang:~# cansend can0 11#112233445566
root@qiyang:~# cansend can0 11#112233445577
root@qiyang:~# █
```

接收到以下数据，说明传输成功



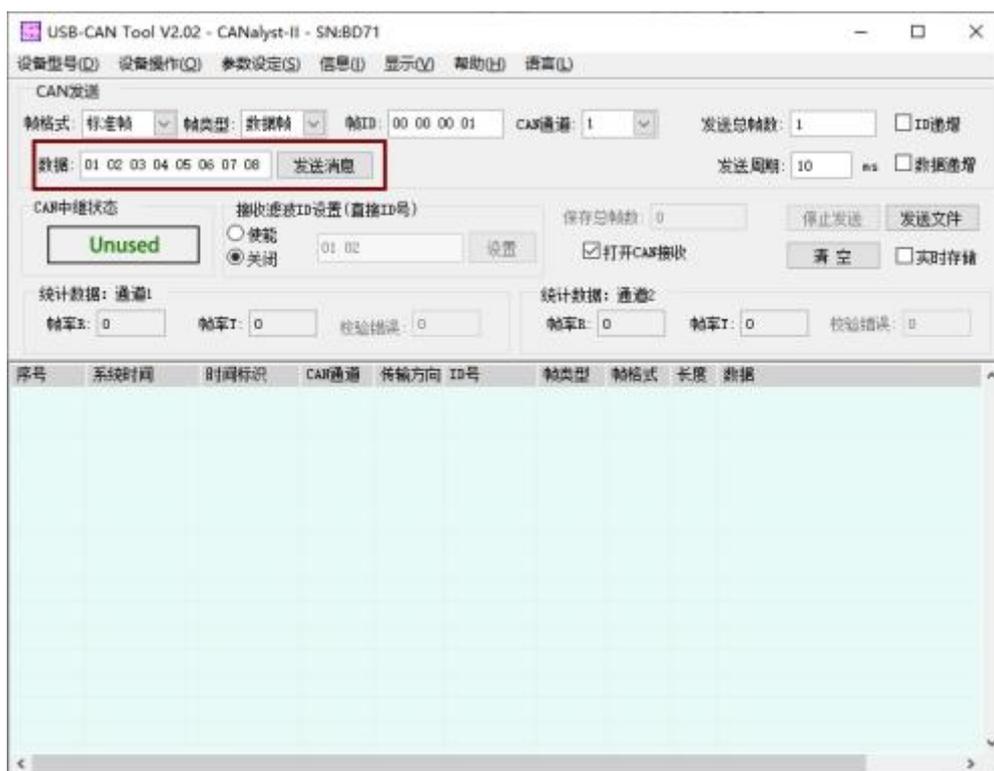
3. 利用 CAN 分析发送数据到目标板

3.1 通过命令使板子上的 CAN 阻塞接收

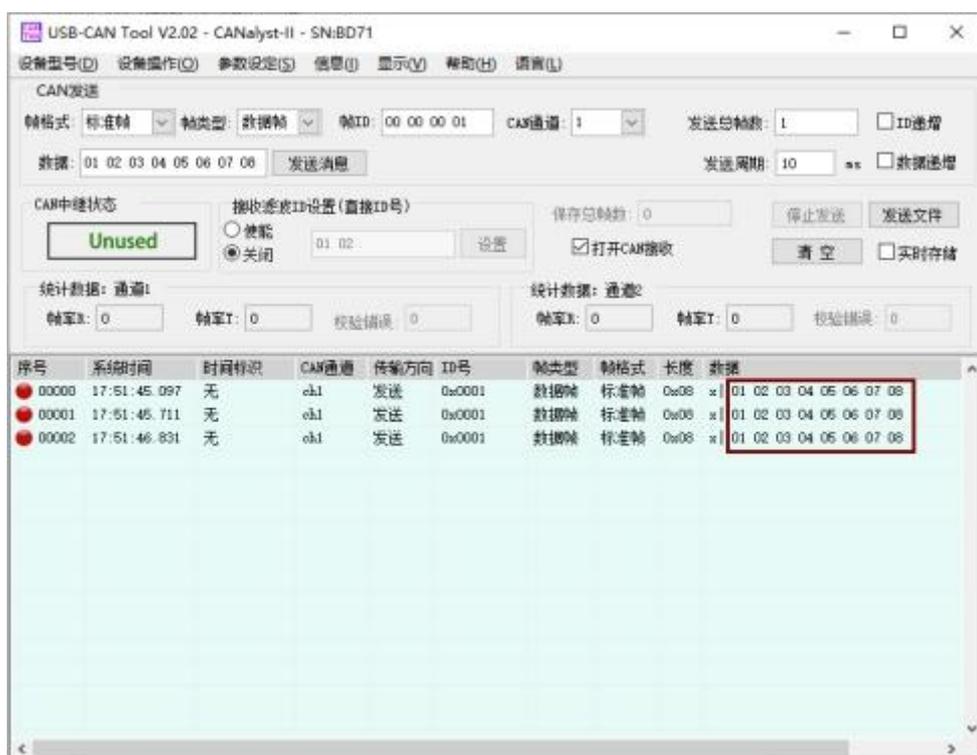
```
# candump can0
```

```
root@qiyang:~# candump can0
█
```

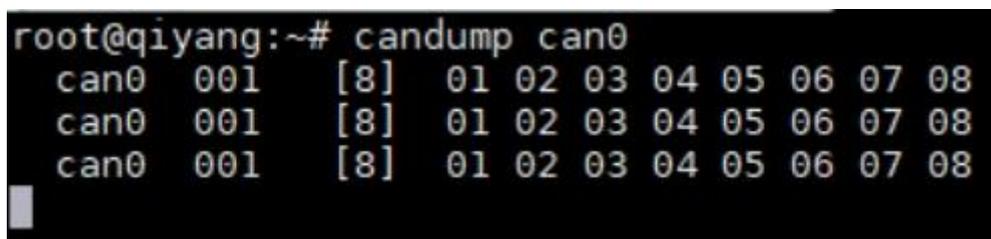
3.2 利用 CAN 分析仪的软件发送数据



3.3 最后的正确结果



板子的数据显示



至此 CAN0 接口的测试结束，可按照此方法测试 CAN1 接口。

设备树文件：

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

2.5、USB 测试

IAC-IMX8MP-Kit 开发板共有 5 路 USB 接口。

对应接口说明：

USB	对应接口说明	功能说明
USB-Type-C	J5 (Type-C 接口)	镜像烧写
USB-HOST (USB3.0)	J6、J7、J8 (USB3.0 插座)	USB-HOST 接口使用，外接 USB 设备
USB-HOST (USB3.0)	J22 (M.2 插座)	外接 5G 模块 (USB3.0 通讯)

测试原理：

开发板 usb 支持热插拔，将 U 盘插入后系统会自动识别并打印出 U 盘相关信息

识别后在/dev 目录下生成该设备节点/dev/sda 及分区节点/dev/sda1 (若有多个分区，数字部分依次增加)

系统会将所有分区挂载到/run/media/目录下，通过读写对应目录下的文件来判断该接口是否正常。

测试步骤和结果：

测试以只有一个分区的 U 盘为例

1. 将正常使用的 U 盘插入，调试串口打印如下信息：

```

root@imx8mpevk:~# [ 21.886848] usb 3-1.2: new high-speed USB device number 5 using xhci-hcd
[ 21.996755] usb-storage 3-1.2:1.0: USB Mass Storage device detected
[ 22.004680] scsi host0: usb-storage 3-1.2:1.0
[ 23.041563] scsi 0:0:0:0: Direct-Access      USB          SanDisk 3.2Gen1 1.00 PQ: 0 ANSI: 6
[ 23.051813] sd 0:0:0:0: [sda] 60088320 512-byte logical blocks: (30.8 GB/28.7 GiB)
[ 23.060087] sd 0:0:0:0: [sda] Write Protect is off
[ 23.065238] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 23.106217] sda: sda1
[ 23.110727] sd 0:0:0:0: [sda] Attached SCSI removable disk
    
```

如上图所示，显示 U 盘的一些基本信息，U 盘识别的设备节点为 sda，子节点为 sda1

2. 用 fdisk 命令来查看 sda 的信息

```
# fdisk -l /dev/sda
```

```
root@imx8mpevk:~# fdisk -l /dev/sda
Disk /dev/sda: 28.65 GiB, 30765219840 bytes, 60088320 sectors
Disk model: SanDisk 3.2Gen1
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device      Boot Start      End  Sectors  Size Id Type
/dev/sda1                32 60088319 60088288 28.7G  c W95 FAT32 (LBA)
```

3. 查看 U 盘里的内容

```
# ls -l /run/media/sda1/
```

```
root@imx8mpevk:~# ls -l /run/media/sda1/
total 8416
drwxrwx--- 2 root disk 16384 Jul  9 2018 SanDiskSecureAccess
-rwxrwx--- 1 root disk 8600360 Nov  4 2016 SanDiskSecureAccessV3.01_win.exe
```

4. 可以通过创建、拷贝、删除文件来测试 U 盘的读写

5. 用相同的方法来测试 2 个 host usb 口，测试完毕后拔出 U 盘，打印信息如下

```
root@imx8mpevk:~# [ 188.558041] usb 3-1.2: USB disconnect, device number 5
[ 188.640912] FAT-fs (sda1): unable to read boot sector to mark fs as dirty
```

设备树文件:

```
arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts
```

2.6、TF 测试

IAC-IMX8MP-Kit 开发板提供 1 路 TF 卡接口（J33）可供用户使用。

准备 TF 卡容量：可选择 8GB、16GB；支持 TF 卡格式：FAT32。

测试原理：

板载 TF 卡接口支持热插拔，将 TF 卡插入后，系统会识别该 TF 卡，并打印出 TF 卡相关信息。

在/dev 目录下生成该设备节点及分区节点，之后系统会自动将所有分区挂载到 /run/media/目录下，通过读写对应目录下文件，可判断该接口是否正常。

测试步骤和结果：

以下测试步骤以只有一个分区的 TF 卡为例，若有多个分区，则测试方法类似。插入 TF 卡产生的设备节点为/dev/mmcblk1，分区 n 对应的分区设备节点为/dev/mmcblk1pn 在这里插入一张金斯顿 8G 的 TF 卡，打印信息如下：

```
root@imx8mmqiyang:/# [ 1479.217181] mmc1: host does not support reading read-only switch, assuming write-enable
[ 1479.233857] mmc1: new high speed SDHC card at address aaaa
[ 1479.239907] mmcblk1: mmc1:aaaa SC16G 14.8 GiB
[ 1479.249011] mmcblk1: p1
[ 1479.500045] FAT-fs (mmcblk1p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
```

如上图所示，显示 TF 的卡一些基本信息，这里的设备节点为 mmcblk1，分区为 p1 也可以用 fdisk 命令来查看 TF 卡的信息

```
# fdisk -l /dev/mmcblk1
```

```
root@imx8mmqiyang:/# fdisk -l /dev/mmcblk1
Disk /dev/mmcblk1: 14.9 GiB, 15931539456 bytes, 31116288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device          Boot Start      End  Sectors  Size Id Type
/dev/mmcblk1p1  0 31116287 31116288 14.9G  b W95 FAT32
root@imx8mmqiyang:/#
```

这里已经自动将 TF 卡挂载到/run/media/mmcblk1p1/目录下，可以直接查看 TF 卡里的内容

```
root@imx8mmqiyang:/# ls /run/media/mmcblk1p1/  
11.txt Android DCIM LOST.DIR  
root@imx8mmqiyang:/# █
```

可以通过创建、拷贝、删除文件来测试 TF 卡的读写
拔出 TF 卡，打印信息如下

```
root@imx8mmqiyang:/#  
root@imx8mmqiyang:/# [ 1579.299337] mmc1: card aaaa removed  
[ 1579.380679] FAT-fs (mmcblk1p1): FAT read failed (blocknr 32)  
█
```

2.7、音频测试

IAC-IMX8MP-Kit 开发板对外提供 1 路双声道音频输出接口（耳机插座）、2 路功放输出（喇叭）接口-J19/J20、1 路 MIC 录音接口-J17。

测试原理：

可通过 `aplay` 命令播放音频文件，通过 `arecord` 录音，录音接口-J17；

测试步骤和结果：

1.录音测试

使用麦克风接好录音接口 J17，然后在终端中输入 `arecord -c 2 -r 44100 test.wav` 进行录音，录音文件名为 `test.wav`

```
# arecord -c 2 -r 44100 test.wav
```

2.播放音频测试

通过耳机（J18）和喇叭（J20）

通过 `aplay test.wav` 命令播放 `test.wav`

```
# aplay test.wav
```

或者将提前准备好的音频文件（**mp3 等格式**）放入 U 盘中，通过 U 盘拷贝文件到主板系统中进行播放音频测试。

测试工具：

`arecord`、`aplay`

设备树文件：

```
arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts
```

2.8、以太网测试

IAC-IMX8MP-Kit 开发板对外提供 2 路千兆以太网接口，eth0 接口 J4、eth1 接口 J3；

2 路千兆以太网工作在不同网段，测试以 eth0 为例。

测试原理：

设置板子网络，用 ping 工具检查网络是否连通。

测试步骤和结果：

测试前需要准备网线和网络环境（路由器或者交换机）

1.网线正常连接 eth0 网口（J4），另外一端连接到交换机上；确保交换机网络环境可以上网。

网线连接以后调试串口终端会打印以下信息：

```
root@imx8mpevk:~# [ 104.351507] fec 30be0000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[ 104.359296] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

2.IP 地址配置

IP 地址配置支持 DHCP 和静态 IP 设置；

DHCP 自动配置，输入

`udhcpd -i eth0`（注：有时无法自动获取，可利用下面的手动配置）

手动配置，输入

`ifconfig eth0 192.168.1.71`（板子上电已经默认设置为这个）

`echo nameserver 114.114.114.114 > /etc/resolv.conf`

`route add default gw 192.168.1.1 dev eth0`

3. 测试内网，输入

`ping -I eth0 192.168.1.1`

```
root@imx8mpevk:~# ping -I eth0 192.168.1.1
PING 192.168.1.1 (192.168.1.1) from 192.168.1.233 eth0: 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=254 time=10.1 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=254 time=1.35 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=254 time=1.33 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=254 time=1.33 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=254 time=1.33 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=254 time=1.33 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=254 time=1.33 ms
^C
--- 192.168.1.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 1.325/2.580/10.065/3.055 ms
```

4.测试外网，输入

有任何技术问题或需要帮助，请联系：supports@qiyangtech.com

第 28 页 共 45 页

购买产品，请联系销售：sales@qiyangtech.com

更多信息请访问：<http://www.qiyatech.com>

©2021 Qiyangtech 版权所有

```
# ping -I eth0 www.baidu.com
```

```
root@imx8mpevk:~# ping -I eth0 www.baidu.com
PING www.a.shifen.com (180.101.49.11) from 192.168.1.233 eth0: 56(84) bytes of data.
64 bytes from 180.101.49.11 (180.101.49.11): icmp_seq=1 ttl=52 time=15.7 ms
64 bytes from 180.101.49.11 (180.101.49.11): icmp_seq=2 ttl=52 time=15.8 ms
64 bytes from 180.101.49.11 (180.101.49.11): icmp_seq=3 ttl=52 time=15.5 ms
64 bytes from 180.101.49.11 (180.101.49.11): icmp_seq=4 ttl=52 time=15.6 ms
64 bytes from 180.101.49.11 (180.101.49.11): icmp_seq=5 ttl=52 time=16.0 ms
^C
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 15.542/15.714/16.003/0.168 ms
```

MAC 地址:

出厂前已默认对 2 路网口设置 MAC 地址，如果镜像重新烧写，还需要重新设置 MAC 地址；

通过 UBoot 可修改 MAC 地址

```
# setenv ethaddr 00:00:0A:12:34:56
# setenv eth1addr 00:00:0A:12:34:57
# saveenv
```

```
u-boot=> setenv ethaddr 00:00:0A:12:34:56
u-boot=> setenv eth1addr 00:00:0A:12:34:57
u-boot=> save
save saveenv
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(2)... OK
```

2.9、WIFI 测试

IAC-IMX8MP-Kit 开发板板载 WIFI 模块。

测试原理：

使用 wpa_passphrase 与 wpa_supplicant 命令实现 wifi 的连接。

测试步骤和结果：

1. 加载驱动（注：匹配的时间较长，大约一分钟左右）

```
# insmod /lib/modules/brcmutil.ko
```

```
# insmod /lib/modules/brcmfmac.ko
```

```
root@imx8mpevk:~# insmod /lib/modules/brcmutil.ko
root@imx8mpevk:~# insmod /lib/modules/brcmfmac.ko
root@imx8mpevk:~# [ 972.022899] brcmfmac: brcmf_fw_alloc_request: using brcm/brcmfmac4339-sdio for chip BCM4339/2
[ 972.350655] brcmfmac: brcmf_fw_alloc_request: using brcm/brcmfmac4339-sdio for chip BCM4339/2
[ 1033.182655] brcmfmac: brcmf_c_process_clm_blob: no clm_blob available (err=-11), device may have limited channels available
[ 1033.197534] brcmfmac: brcmf_c_preinit_dcmds: Firmware: BCM4339/2 wl0: Apr 6 2015 19:38:13 version 6.37.32.46.1 (A0 Station/P2P)
[ 2718.422598] audit: type=1006 audit(1616585401.708:4): pid=1137 uid=0 old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=3 res=1
```

2. 设置 wifi 的用户名：QYWIFI，密码：QY@2019.com，如果不一样，请修改

```
# wpa_passphrase QYWIFI QY@2019.com >> /etc/wpa_supplicant.conf
```

```
# sync
```

3. 连接 wifi

```
# wpa_supplicant -Dnl80211 -i wlan0 -c /etc/wpa_supplicant.conf -B
```

4. 自动获取 IP

```
# udhcpc -i wlan0
```

```
root@imx8mpevk:~# udhcpc -i wlan0
udhcpc: started, v1.33.0
udhcpc: sending discover
udhcpc: sending discover
udhcpc: sending select for 192.168.3.234
udhcpc: sending select for 192.168.3.234
udhcpc: sending select for 192.168.3.234
udhcpc: lease of 192.168.3.234 obtained, lease time 86400
[ 4773.463690] ieee80211 phy0: brcmf_inetaddr_changed: fail to get arp ip table err:-52
/etc/udhcpc.d/50default: Adding DNS 202.101.172.35
/etc/udhcpc.d/50default: Adding DNS 202.101.172.47
```

5. 静态 IP

如果网段在 192.168.3.1，则设置 ip 的命令替换为

```
ifconfig wlan0 192.168.3.xxx
```

如果此时需要连接外网时，需要添加默认网关

```
route del default
```

```
route add default gw 192.168.3.1 dev wlan0
```

```
echo nameserver 114.114.114.114 > /etc/resolv.conf
```

6. Ping 百度

```
# ifconfig eth0 down
```

```
# ping -I wlan0 www.baidu.com
```

```
root@imx8mpevk:~# echo nameserver 114.114.114.114 > /etc/resolv.conf
root@imx8mpevk:~# ping -I wlan0 www.baidu.com
PING www.a.shifen.com (220.181.38.149) from 192.168.3.234 wlan0: 56(84) bytes of data.
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=1 ttl=52 time=41.2 ms
64 bytes from 220.181.38.149: icmp_seq=4 ttl=52 time=37.3 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=6 ttl=52 time=41.3 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=7 ttl=52 time=31.1 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=8 ttl=52 time=32.4 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=9 ttl=52 time=31.7 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=10 ttl=52 time=34.8 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=11 ttl=52 time=31.0 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=12 ttl=52 time=31.9 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=13 ttl=52 time=32.2 ms
^C
--- www.a.shifen.com ping statistics ---
13 packets transmitted, 10 received, 23.0769% packet loss, time 25196ms
rtt min/avg/max/mdev = 30.956/34.487/41.268/3.841 ms
```

注：如有打量丢包现象，记得接上天线

设备树文件：

```
arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts
```

对外提供 1 路双声道音频输出接口

2.10、5G 测试

IAC-IMX8MP-Kit 开发板板载 M.2 接口（USB 通讯），可以用于外接 4G/5G 模块。

这里测试以测试 5G 模块为例，需要准备 5G 模块（参考型号 Quectel_RM500U）、天线和 SIM 卡。如不需要可跳过此节。

测试原理：

外接 5G 模块，进行拨号上网测试。

测试步骤和结果：

1.关闭有线网络

```
# route del default
```

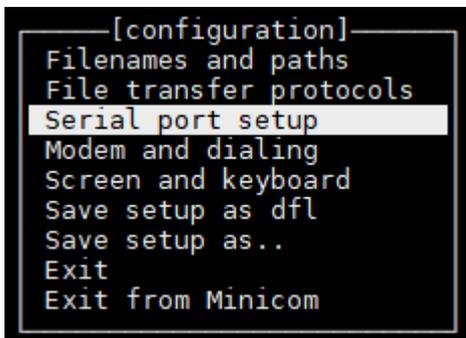
```
# ifconfig eth0 down
```

2.在终端执行：（执行完过 10 几秒左右出现如下打印信息，表示识别到模块）

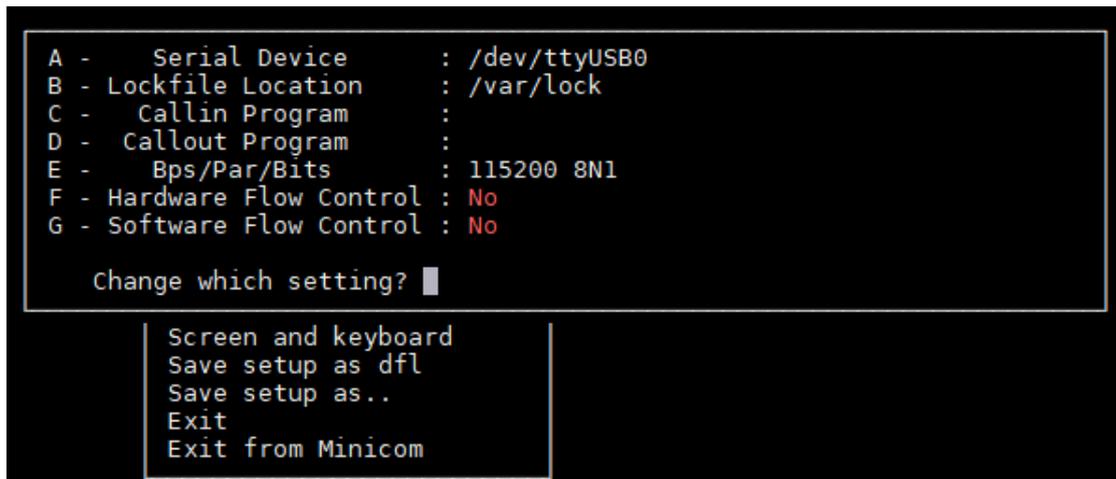
```
imx8mpevk login: [ 11.708439] usb 3-1.1: USB disconnect, device number 4
[ 31.230952] usb 4-1.1: new SuperSpeed Gen 1 USB device number 3 using xhci-hcd
[ 31.340336] cdc_ether 4-1.1:1.0 usb0: register 'cdc_ether' at usb-xhci-hcd.1.auto-1.1, CDC Ethernet Device, d2:8a:2e:97:5c:62
[ 31.354776] option 4-1.1:1.2: GSM modem (1-port) converter detected
[ 31.361677] usb 4-1.1: GSM modem (1-port) converter now attached to ttyUSB0
[ 31.369040] option 4-1.1:1.3: GSM modem (1-port) converter detected
[ 31.375683] usb 4-1.1: GSM modem (1-port) converter now attached to ttyUSB1
[ 31.383024] option 4-1.1:1.4: GSM modem (1-port) converter detected
[ 31.389695] usb 4-1.1: GSM modem (1-port) converter now attached to ttyUSB2
[ 44.280747] IPv6: ADDRCONF(NETDEV_CHANGE): usb0: link becomes ready
```

3.用 AT 指令查看模块信息和卡

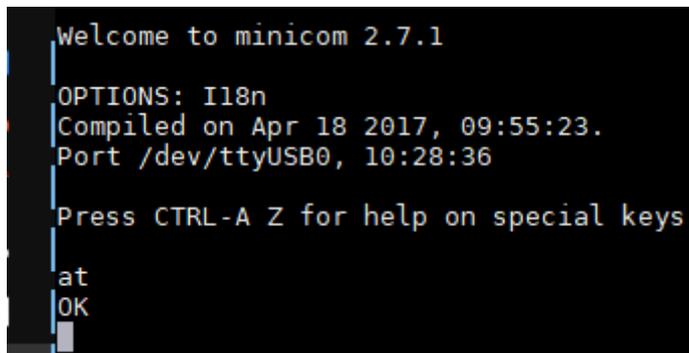
```
# minicom -s
```



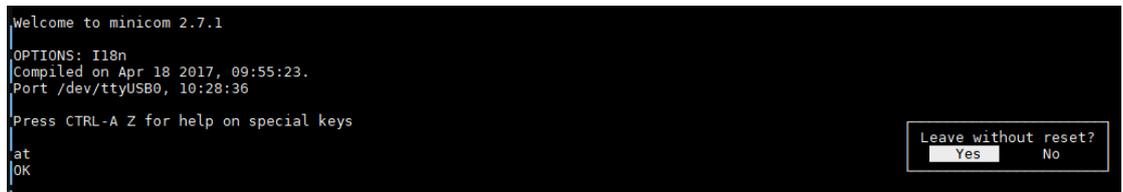
选择 Serial port setup，然后输入 A，将串口修改为/dev/ttyUSB0，如下图



回车，选择 Exit，在回车进入如下界面，输入 AT 出现 ok 打印



再输入 ctrl+A Z，在输入 Q，回车



4.拨号上网:

AT+CGDCONT=1,"IP","3GNET"

AT+NWNESHAREACT=1,1,0

```
# AT+NWNESHAREACT?
```

```
at
OK
AT+CGDCONT=1,"IP","3GNET"
OK
AT+NWNESHAREACT=1,1,0
OK
+CMT: ,156
0891683110302605F06005A00191F80008129092017192238C050003A2020100300
+CMT: ,54
0891683110302605F06405A00191F800081290920171922326050003A20202670D5
AT+NWNESHAREACT?
+NWNESHAREACT: 1,0,IP,ECM
OK
```

退出 minicom ,ctrl+a > z > q 退出

```
# ifconfig (拨号成功 IP 为 192.168.0.2)
```

```
root@imx8mpevk:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:00:0a:12:34:57 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 53

eth1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:00:0a:12:34:56 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 184 bytes 14060 (13.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 184 bytes 14060 (13.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

usb0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC> mtu 1500
    inet 192.168.42.2 netmask 255.255.255.0 broadcast 192.168.42.255
    inet6 fe80::6c76:88ff:febd:baa2 prefixlen 64 scopeid 0x20<link>
    ether 6e:76:88:cd:ba:a2 txqueuelen 1000 (Ethernet)
    RX packets 30 bytes 3980 (3.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 85 bytes 15145 (14.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

5.Ping 百度

ping -I usb0 www.baidu.com

```
root@imx8mpevk:~# ping -I usb0 www.baidu.com
PING www.a.shifen.com (112.80.248.76) from 192.168.42.2 usb0: 56(84) bytes of data.
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=1 ttl=51 time=58.8 ms
[ 288.194820] audit: type=1006 audit(1616581801.576:4): pid=920 uid=0 old-auid=4294
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=2 ttl=51 time=39.9 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=3 ttl=51 time=44.0 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=4 ttl=51 time=32.9 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=5 ttl=51 time=50.7 ms
^C
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 32.921/45.273/58.837/8.911 ms
```

设备节点:

/dev/ttyUSB0

设备树文件:

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

2.11、SSD 测试

IAC-IMX8MP-Kit 开发板板载 M.2 接口（PCIE 通讯），可以用于外接 SSD 硬盘。

需要准备 SSD 硬盘。如不需要可调过此节。

测试原理：

插入 SSD 硬盘，对其可进行文件的写入写出即可

测试步骤和结果：

1.插入 SSD，开机启动会有如下信息显示

```

3.726380] pci 0000:00: root bus resource [bus 00-ff]
3.731877] pci 0000:00: root bus resource [io 0x0000-0xffff]
3.738066] pci 0000:00: root bus resource [mem 0x18000000-0x1fffffff]
3.744969] pci 0000:00:00:0: [16c3:abcd] type 01 class 0x060400
3.750993] pci 0000:00:00:0: reg 0x10: [mem 0x00000000-0x0000ffff]
3.757273] pci 0000:00:00:0: reg 0x28: [mem 0x00000000-0x0000ffff pref]
3.764016] pci 0000:00:00:0: supports D1
3.768833] pci 0000:00:00:0: PME# supported from D0 D1 D3hot D3cold
3.772726] pci 0000:01:00:0: [1987:5013] type 00 class 0x018002
3.783449] pci 0000:01:00:0: reg 0x10: [mem 0x00000000-0x00003fff 64bit]
3.790577] pci 0000:01:00:0: 7.876 Gb/s available PCIe bandwidth, limited by 8.0 GT/s PCIe x1 link at 0000:00:00:0 (capable of 31.504 Gb/s with 8.0 GT/s PCIe x4 link)
3.807485] pps pps0: new PPS source ptp0
3.817187] pci 0000:00:00:0: BAR 0: assigned [mem 0x18000000-0x180fffff]
    
```

2.fdisk -l（可以看到节点已经出现 /dev/nvme0n1）

```

root@imx8mpevk:~# fdisk -l
Disk /dev/mtdblock0: 32 MiB, 33554432 bytes, 65536 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mmcbk2: 14.56 GiB, 15636365312 bytes, 30539776 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6b208ad9

Device            Boot  Start      End  Sectors  Size Id Type
/dev/mmcbk2p1          20480  225279    204800  100M  c W95 FAT32 (LBA)
/dev/mmcbk2p2          307200 30539775 30232576 14.4G  83 Linux

Disk /dev/nvme0n1: 476.94 GiB, 512110190592 bytes, 1000215216 sectors
Disk model: PCIe SSD
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
    
```

3.快速分区并格式化硬盘

```
# fdisk /dev/nvme0n1
```

```
> n
```

有任何技术问题或需要帮助，请联系：supports@qiyangtech.com

第 36 页 共 45 页

购买产品，请联系销售：sales@qiyangtech.com

更多信息请访问：<http://www.qiyatech.com>

©2021 Qiyangtech 版权所有

> p

> 回车 3 次

> w

mkfs.ext4 /dev/nvme0n1

```

root@imx8mpevk:~# fdisk /dev/nvme0n1
Welcome to fdisk (util-linux 2.36.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xa9b2bf8c.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-1000215215, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-1000215215, default 1000215215):

Created a new partition 1 of type 'Linux' and of size 476.9 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@imx8mpevk:~# mkfs.ext4 /dev/nvme0n1
nvme0n1   nvme0n1p1
root@imx8mpevk:~# mkfs.ext4 /dev/nvme0n1p1
mke2fs 1.46.1 (9-Feb-2021)
Discarding device blocks: done
Creating filesystem with 125026646 4k blocks and 31260672 inodes
Filesystem UUID: a15cc75c-d78d-41ea-beea-4a3e9d2a0b61
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): [ 288.386590] audit: type=1006 audit(1616581801.636:
done
Writing superblocks and filesystem accounting information: done

```

4.挂载

mount /dev/nvme0n1p1 /mnt/

5.执行文件读写操作

```
root@imx8mpevk:~# cd /mnt/  
root@imx8mpevk:/mnt# ls  
lost+found  
root@imx8mpevk:/mnt# echo hello > wangtg.txt  
root@imx8mpevk:/mnt# cat wangtg.txt  
hello
```

设备树文件:

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

2.12、时钟测试

IAC-IMX8MP-Kit 开发板使用 I2C 连接底板上的 PCF8563 芯片作为外部硬件时钟，请在测试 RTC 之前，确保已经安上电池。

测试原理：

通过 date 系统命令设置系统时间，用 hwclock 命令把系统时间写入硬件时钟，通过 rtc_test 测试程序读取硬件时钟并打印出来，断电后重启，查看时钟是否准确。

测试步骤和结果：

- 1.连接网线，可同外网

ping cn.ntp.org.cn

```

root@imx8mpevk:~# ping cn.ntp.org.cn
PING cn.ntp.org.cn (120.25.115.20) 56(84) bytes of data.
64 bytes from 120.25.115.20 (120.25.115.20): icmp_seq=1 ttl=50 time=27.0 ms
64 bytes from 120.25.115.20 (120.25.115.20): icmp_seq=2 ttl=50 time=26.8 ms
64 bytes from 120.25.115.20 (120.25.115.20): icmp_seq=3 ttl=50 time=26.7 ms
64 bytes from 120.25.115.20 (120.25.115.20): icmp_seq=4 ttl=50 time=26.5 ms
64 bytes from 120.25.115.20 (120.25.115.20): icmp_seq=5 ttl=50 time=27.0 ms
^C
--- cn.ntp.org.cn ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 8044ms
rtt min/avg/max/mdev = 26.519/26.813/27.031/0.190 ms
    
```

- 2.同步时间

ntpdate cn.ntp.org.cn

```

root@imx8mpevk:~# ntpdate cn.ntp.org.cn
11 Oct 11:57:40 ntpdate[884]: step time server 203.107.6.88 offset +12431760.491077 sec
    
```

- 3.在板子上执行 date 命令，可查看到当前系统时钟。

date

```

root@imx8mpevk:~# date
Mon Oct 11 11:57:58 CST 2021
    
```

- 4.用 hwclock 命令把系统时间写入硬件时钟芯片。

hwclock -w

- 5.分别用 date 和 hwclock 命令来查看系统和硬件时钟。

```
root@imx8mpevk:~# hwclock -w
root@imx8mpevk:~# date
Mon Oct 11 11:58:38 CST 2021
root@imx8mpevk:~# hwclock
2021-10-11 11:58:40.335425+08:00
```

6.设置成功之后，执行 rtc_test 测试程序。（注：此时间为 UTC 时间）

```
# ./rtc_test /dev/rtc0
```

```
root@imx8mpevk:~# ./rtc_test /dev/rtc0

          RTC Driver Test Example.
Current RTC date/time is 2021/10/11, 04:00:01.
Current RTC date/time is 2021/10/11, 04:00:02.
Current RTC date/time is 2021/10/11, 04:00:03.
Current RTC date/time is 2021/10/11, 04:00:04.
Current RTC date/time is 2021/10/11, 04:00:05.
Current RTC date/time is 2021/10/11, 04:00:06.
Current RTC date/time is 2021/10/11, 04:00:07.
Current RTC date/time is 2021/10/11, 04:00:08.
Current RTC date/time is 2021/10/11, 04:00:09.
Current RTC date/time is 2021/10/11, 04:00:10.
```

程序打印 10 条当前硬件时间后退出程序，提前退出程序请按 ctrl+c

查看是否精准走时，查看有无出现丢秒现象。

7.断电，过一会再上电，再次查看系统和硬件时钟，看时间有没有保存，走时是否精准。

8.通过和 PC 时间对比之后，发现时间无误差，若需测试长时间的走时精确度，可以分别断电和上电老化测试几天几周或者几个月来测试时钟误差。

本主板在发布之前都经过 1 个月的老化测试，时间误差不超过 2S

对于发货产品均经过 24 小时以上老化测试，时间误差不超过 1S

设备节点：

/dev/rtc

/dev/rtc0

设备树文件：

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

驱动代码：

drivers rtc/rtc-pcf8563.c

内核对应选项：

CONFIG_RTC_DRV_PCF8563=y

2.13、看门狗测试

IAC-IMX8MP-Kit 开发板硬件设计看门狗电路。

测试原理：

硬件看门狗，GPIO2_IO10 使能看门狗，GPIO2_IO11 执行喂狗操作。

测试步骤和结果：

1.运行 watchdog_feed_test 开发板不重启

./watchdog_feed_test /dev/qiyang_watchdog

```
root@imx8mpevk:/usr/test# ./watchdog_feed_test /dev/qy_watchdog
[ 841.225305] watchdog: enable watchdog
```

2.运行 watchdog_notfeed_test 开发板过 1.6s 马上重启

./watchdog_notfeed_test /dev/qiyang_watchdog

```
root@imx8mpevk:/usr/test# ./watchdog_notfeed_test /dev/qy_watchdog
[ 899.049298] watchdog: enable watchdog

U-Boot SPL 2021.04-00004-gcb364d6 (Sep 08 2021 - 16:10:49 +0800)
DDRINFO: start DRAM init
DDRINFO: DRAM rate 3000MTS
DDRINFO: ddrphy calibration done
DDRINFO: ddrmix config done
SEC0: RNG instantiated
Normal Boot
WDT: Not found!
Trying to boot from BOOTROM
image offset 0x8000, pagesize 0x200, ivt offset 0x0
NOTICE: BL31: v2.4(release):lf-5.10.35-2.0.0-rc2-0-gec35fef
NOTICE: BL31: Built : 07:23:15, Aug 10 2021

U-Boot 2021.04-00004-gcb364d6 (Sep 08 2021 - 16:10:49 +0800)
```

设备节点：

/dev/qiyang_watchdog

测试源码：

watchdog_feed_test.c

watchdog_notfeed_test.c

设备树文件：

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

驱动代码：

drivers/misc/qiyang_watchdog.c

有任何技术问题或需要帮助，请联系：supports@qiyangtech.com

第 41 页 共 45 页

购买产品，请联系销售：sales@qiyangtech.com

更多信息请访问：<http://www.qiytech.com>

©2021 Qiyangtech 版权所有

2.14、摄像头测试

IAC-IMX8MP-Kit 开发板对外提供 2 路 MIPI-CSI (4 通道) 接口, 用于外接摄像头, 支持 2 路摄像头同时输入。

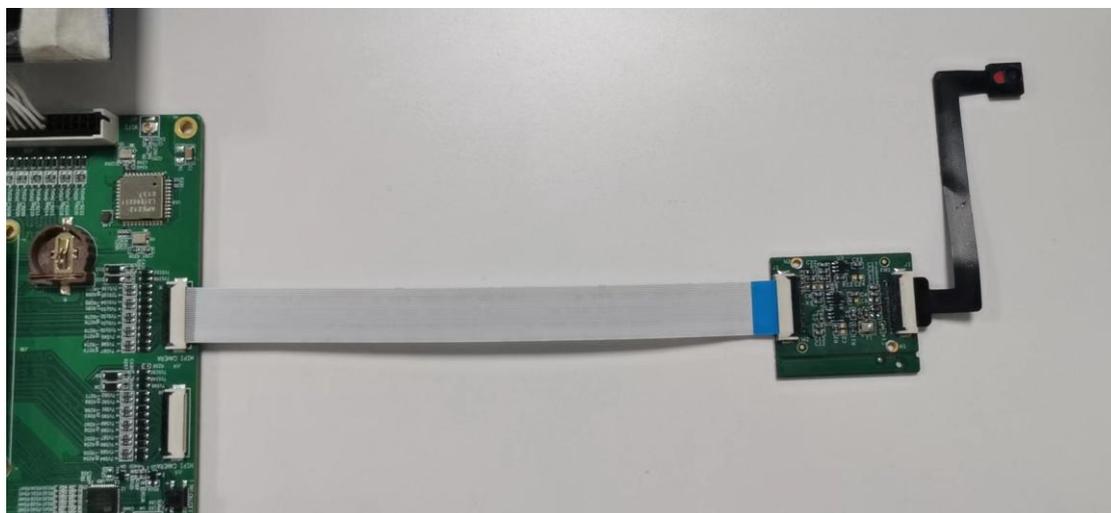
需准备摄像头模块, 默认调试 OV5640 摄像头模组, 需要单独采购。

测试原理:

连接两路 ov5640 摄像头, 查看拍摄内容。

测试步骤和结果:

1. 连接任意一路摄像头, 连接到 J15 或者 J16 上 (注意 FPC 线连接方向)



2. 查看摄像头节点, 以下 cap_devic 就是一个摄像头节点

```
# v4l2-ctl --list-device
```

```

root@imx8mpevk:~# v4l2-ctl --list-device
():
    /dev/v4l-subdev0

():
    /dev/v4l-subdev1

mxc-isi-m2m (platform:32e00000.isi:m2m_devic):
    /dev/video2

mxc-isi-cap (platform:32e02000.isi:cap_devic):
    /dev/video3

FSL Capture Media Device (platform:mxc-md):
    /dev/media0

vsi_v4l2dec (platform:vsi_v4l2dec):
    /dev/video1

vsi_v4l2enc (platform:vsi_v4l2enc):
    /dev/video0
    
```

3.实时录像

```

# gst-launch-1.0 -v v4l2src device=/dev/video3 !
"video/x-raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink
window-width=640 window-height=480
    
```

这里以连接 7 寸屏为例；



设备树文件：

arch/arm64/boot/dts/freescale/imx8mp-qiyang.dts

四、测试小结

开发板的基本功能到此测试完毕，对于测试过程中出现的问题，可根据提供的测试源码进行排查。编写中的笔者本着求真务实的精神对文字和程序进行斟酌和校验，但仍难免存在疏误，敬请读者批评指正和谅解。

浙江启扬智能科技有限公司

电话：0571-87858811 / 87858822

传真：0571-89935912

技术支持：0571-87858811 转 805

E-MAIL: supports@qiyangtech.com

网址： <http://www.qiytech.com>

地址：浙江省杭州市西湖科技园西园八路6号A幢3楼

邮编：310030