MYD-YT113X_Linux 软件评估指南

文件状态： [ ] 草稿 [√] 正式发布	**文件标识：**	MYIR-MYD-YT113X-SW-EG-ZH-L5.4.61
	当前版本：	V1.1[文档]
	作　　者：	Nico
	创建日期：	2023.05.01
	最近更新：	2023.08.30

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

版 本 历 史

版本	作者	参与者	日期	备注
V1.0[文档]	Nico	Licy	2023.05.01	初始版本
V1.1[文档]	Nico	Licy	2023.08.30	新增适用型号 MYD-YT113-I 删除 RS232 测试，新增 RS485 测试

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

目 录

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

1. 概述

Linux 软件评估指南用于介绍在米尔的开发板上运行开源 Linux 系统下的核心资源与外设资源的测试步骤与评估方法。本文可作为前期评估指导使用，也可以作为通用系统开发的测试指导书使用。

1.1. 硬件资源

米尔电子的 MYD-YT113X 板卡由核心板 MYC-YT113X 和底板 MYB-YT113X 组成，核心板与底板采用邮票孔焊接方式。此外 MYIR 提供了丰富的软件资源以及文档资料。有关硬件部分的详细配置参数请查看《MYD-YT113X 产品手册》。同时用户在评估测试过程中会用到一些配件，参见下面的列表。

表 1-1.选配模块

配件	接口方式	说明及链接
LCD 屏	lvds 接口	7 寸 lvds 屏: https://www.myir.cn/shows/106/3.html
拓展板模块	树莓派接口	MY-WiredCom: https://www.myir.cn/index.php/shows/8/9.html

1.2. 软件资源

MYD-YT113X 开发板的 BSP 是基于全志官方开源社区版 Linux BSP 移植与修改而来，系统镜像采用 buildroot 项目进行构建。Bootloader，Kernel 以及文件系统各部分 软件资源全部以源码的形式开放，具体内容请查看《MYD-YT113X_SDK 发布说明》。开发板在出厂时根据核心板型号已经烧录了镜像，您只需要上电即可使用。

1.3. 文档资源

根据用户使用开发板的各个不同阶段，SDK 中包含了各阶段的文档，发布说明，评估 指南，开发指南，应用笔记，常用问答等不同类别的文档和手册。具体的文档列表参见 《MYD-YT113X_SDK 发布说明》表 2-4 中的说明。

1.4. 环境准备

在开始评估开发板软件之前，您需要对开发板做一些必要的准备和配置一些基础环境，包括正确硬件接线，配置调试串口，设置启动等步骤。详细的步骤可以参照《MYD-YT113X 快速入门指南》。接下来的部分重点介绍如何对系统的硬件资源和接口以及软件

功能进行评估和测试。 主要借助一些 Linux 下常用的工具和命令，以及自己开发的应用进行测试。软件评估指南 分为多个部分来描述，包括：核心资源，外设资源，网络应用，多媒体应用，开发支持应用，系统工具等几大类。后面的章节会针对各个部分做全方位的讲解，并详细描述各部分 资源的具体评估方法和步骤。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

2. 核心资源

在 Linux 系统中，提供了 proc 虚拟文件系统来查询各项核心资源的参数以及一些通用工具来评估资源的性能。下面将具体对 CPU，memory，eMMC，RTC 等核心资源的参数进行读取与测试。

2.1. CPU

1) 查看 CPU 信息命令

读取系统中的 CPU 的提供商和参数信息，则可以通过/proc/cpuinfo 文件得到。

```
[root@myir:/]# cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 48.00
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idiv
t vfpd32 lpae
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

processor       : 1
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 48.00
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idiv
t vfpd32 lpae
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5
```

```
Hardware        : Generic DT based system
Revision        : 0000
Serial          : 0000000000000000
```

- processor：系统中逻辑处理核的编号，对于多核处理器则可以是物理核、或者使用超线程技术虚拟的逻辑核
- modelname：CPU 属于的名字及其编号
- BogoMIPS：在系统内核启动时粗略测算的 CPU 每秒运行百万条指令数（MillionInstructionsPerSecond）

2) 查看 CPU 使用率

在 T113 系列芯片执行以下操作可以查看 CPU 使用率:

```
[root@myir:/]# top
Mem: 38180K used, 68796K free, 84K shrd, 1760K buff, 9704K cached
CPU:   4% usr   0% sys   0% nic  95% idle   0% io   0% irq   0% sirq
Load average: 0.64 0.20 0.07 1/93 1954
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
 1954  1468 root     R     2152   2%   2% top
   12     2 root     SW       0   0%   2% [rcuc/0]
 1476     1 root     S    30832  29%   0% adbd
 1953  1483 root     S     4848   5%   0% sleep 1
 1468     1 root     S     2940   3%   0% -sh
 1483     1 root     S     2688   3%   0% {adb_conf.sh} /bin/sh /etc/adb_conf.sh start
 1368     1 root     S     2392   2%   0% dbus-daemon --system
    1     0 root     S     2152   2%   0% init
 1350     1 root     S     2152   2%   0% /sbin/klogd -n
 1346     1 root     S     2152   2%   0% /sbin/syslogd -n
 1402     1 root     S     2152   2%   0% udhcpc -R -n -p /var/run/udhcpc.eth0.pid -i eth0
 1420     1 root     S     2152   2%   0% /usr/sbin/telnetd -F
 1409     1 root     S     1868   2%   0% /usr/sbin/dropbear -R
```

```
 1426     1 root     S     1740   2%   0% /usr/sbin/tftpd -c -l -s /var/lib/tf
tpboot
 1140     2 root     SW       0   0%   0% [irq/45-sunxi-gp]
 1393     2 root     SW       0   0%   0% [irq/172-usb_id]
  979     2 root     SW       0   0%   0% [irq/49-dispaly]
   38     2 root     IW       0   0%   0% [kworker/0:1-eve]
    9     2 root     SW       0   0%   0% [ksoftirqd/0]
   20     2 root     SW       0   0%   0% [ksoftirqd/1]
 1064     2 root     SW       0   0%   0% [irq/58-ehci_hcd]
   19     2 root     SW       0   0%   0% [rcuc/1]
   10     2 root     IW       0   0%   0% [rcu_preempt]
   39     2 root     IW       0   0%   0% [kworker/1:1-eve]
 1131     2 root     SW       0   0%   0% [irq/40-sunxi_tp]
   28     2 root     IW       0   0%   0% [kworker/u4:2-ev]
 1259     2 root     SW       0   0%   0% [irq/43-sunxi-i2]
  751     2 root     IW       0   0%   0% [kworker/u4:4-ev]
 1032     2 root     SW       0   0%   0% [irq/37-4500000.]
 1324     2 root     IW<      0   0%   0% [kworker/1:1H-kb]
    7     2 root     IW       0   0%   0% [kworker/u4:0-ev]
  352     2 root     IW       0   0%   0% [kworker/u4:3-ev]
 1175     2 root     SW       0   0%   0% [irq/38-mmc2]
 1176     2 root     SW       0   0%   0% [irq/38-s-mmc2]
 1246     2 root     IW<      0   0%   0% [kworker/0:1H-mm]
```

- %usr：表示用户空间程序的 cpu 使用率（没有通过 nice 调度）
- %sys：表示系统空间的 cpu 使用率，主要是内核程序
- %nic：表示用户空间且通过 nice 调度过的程序的 cpu 使用率
- %idle：空闲 cpu
- %irq：cpu 处理硬中断的数量
- %sirq：cpu 处理软中断的数量

3）获取 CPU 温度信息

CPU 内置温度传感器作为 CPU 温度采集，可以很方便的获取 CPU 内部温度。

```
[root@myir:/]# cat /sys/class/thermal/thermal_zone0/temp
43282
```

上面显示数字为千分之一度，除以 1000 就是当前温度值。

```
[root@myir:/]# echo "scale=5000;4*a(1)" | bc -l -q &
[1] 2177
3.14159265358979323846264338327950288419716939937510582097494459230
78164062862089986280348253421170679821480865132823066470938446095505
82231725359408128481117450284102701938521105559644622948954930381964
42881097566593344612847564823378678316527120190914564856692346034861
04543266482133936072602491412737245870066063155881748815209209628292
5
.....
[1]+ Done echo "scale=5000;4*a(1)" | bc -l -q
```

上述命令将在后台计算的 PI，并精确到小数点后 5000 位。计算过程需要一段时间。此时，我们可以通过 top 命令检查 CPU 利用率的变化，如下所示：

```
[root@myir:/]# top
Mem: 39348K used, 67628K free, 88K shrd, 1832K buff, 9816K cached
CPU: 84% usr 1% sys 0% nic 45% idle 0% io 0% irq 0% sirq
Load average: 0.88 0.65 0.33 2/91 3809
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
 3654  1468 root     R      1516   1%  50% bc -l -q
```

约 3 分钟后，PI 结果被计算出来。在此期间 CPU 使用率达到 100%，没有发生异常，说明 CPU 压力测试通过。还可以继续增加精确值，可进一步提高测试压力。

2.2. Memory

MYD-YT113X 内存为 128MB 版本，系统会把内存分成设备内存(CMA)和系统内存(MEM)。设备内存是给驱动程序使用的一段连续空间，系统内存是给用户态分配空间。

1) 查看内存信息

读取系统中的内存的参数信息，则可以通过/proc/meminfo 文件得到。

```
[root@myir:/]# cat /proc/meminfo
MemTotal:         106976 kB
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
MemFree:           67332 kB
MemAvailable:      78668 kB
Buffers:            2240 kB
Cached:             9520 kB
SwapCached:            0 kB
Active:            10988 kB
Inactive:           3188 kB
Active(anon):       2464 kB
Inactive(anon):       32 kB
Active(file):       8524 kB
Inactive(file):     3156 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:          2404 kB
Mapped:             4560 kB
Shmem:                84 kB
KReclaimable:       4788 kB
Slab:              15196 kB
SReclaimable:       4788 kB
SUnreclaim:        10408 kB
...
```

- MemTotal:所有可用的 RAM 大小，物理内存减去预留位和内核使用
- MemFree：LowFree+HighFree
- Buffers：用来给块设备做缓存的大小
- Cached：文件的缓冲区大小
- SwapCached：已经被交换出来的内存。与 I/O 相关
- Active：经常（最近）被使用的内存

- Inactive：最近不常使用的内存

2）获取内存使用率

可使用 free 命令来读取内存的使用情况，-m 参数代表单位为 MByte。

```
[root@myir:/]# free -m
             total       used       free     shared    buffers     cached
Mem:           104         38         65          0          2          9
-/+ buffers/cache:         27         77
Swap:            0          0          0
```

- total：内存总量
- used：被使用的内存量
- free：可使用的内存量

3）内存压力测试

通过给定测试内存的大小和次数,可以对系统现有的内存进行压力上的测试。可使用系统工具 memtester 进行测试，如指定内存大小 60MB，测试次数为 10，测试命令为“memtester 60M 10"。

下列以使用 60MB 内存空间，单次测试为例:

```
[root@myir:/]# memtester 60M 1
memtester version 4.3.0 (32-bit)
Copyright (C) 2001-2012 Charles Cazabon.
Licensed under the GNU General Public License version 2 (only).

pagesize is 4096
pagesizemask is 0xfffff000
want 60MB (62914560 bytes)
got 60MB (62914560 bytes), trying mlock ...locked.
Loop 1/1:
  Stuck Address       : ok
  Random Value        : ok
  Compare XOR         : ok
  Compare SUB         : ok
```

```
Compare MUL          : ok
Compare DIV          : ok
Compare OR           : ok
Compare AND          : ok
Sequential Increment: ok
Solid Bits           : ok
Block Sequential     : ok
Checkerboard         : ok
Bit Spread           : ok
Bit Flip             : ok
Walking Ones         : ok
Walking Zeroes       : ok
8-bit Writes         : ok
16-bit Writes        : ok

Done.
```

2.3. eMMC 与 nand 测试

本节主要介绍 eMMC 和 nand 的测试，适用于配置有 eMMC 和 nand 存储器的开发板。eMMC 是一个数据存储设备，包括一个 MultiMediaCard(MMC)接口，一个 NAND 组件。它的成本、体积小、Flash 技术独立性和高数据吞吐量使其成为嵌入式产品的理想选择。

2.3.1. eMMC 测试

1). 查看 eMMC 容量

通过 fdisk-l 命令可以查询到 eMMC 分区信息及容量。

```
[root@myir:/]# fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk0: 7634944 sectors, 3728M
Logical sector size: 512
Disk identifier (GUID): ab6f3888-569a-4926-9668-80941dcb40bc
Partition table holds up to 8 entries
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
First usable sector is 73728, last usable sector is 7634910

Number  Start (sector)    End (sector)  Size  Name
     1           73728          114687  20.0M  boot-resource
     2          114688          116735  1024K  env
     3          116736          118783  1024K  env-redund
     4          118784          180223  30.0M  boot
     5          180224         2277375  1024M  rootfs
     6         2277376         2279423  1024K  dsp0
     7         2279424         2312191  16.0M  private
     8         2312192         7634910  2598M  UDISK
```

2). 查看 eMMC 分区信息

通过 df 命令可以查询到 eMMC 分区信息，使用情况，挂载目录等信息。

```
[root@myir:/]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       992M  232M  745M  24% /
tmpfs            53M   52K   53M   1% /tmp
tmpfs            53M   32K   53M   1% /run
devtmpfs         43M     0   43M   0% /dev
```

- tmpfs: 内存虚拟文件系统，挂载到不同的目录下。
- devtmpfs：用于系统创建 dev。

3). eMMC 的性能测试

性能测试主要测试 eMMC 在 linux 系统下对文件的读写速度，一般结合 time 与 dd 双命令进行测试。

- **写文件测试**

```
[root@myir:/]# time dd if=/dev/zero of=tempfile bs=1M count=100 conv=fdat
async
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 8.11714 s, 12.9 MB/s
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
real    0m8.132s
user    0m0.000s
sys     0m1.564s
```

这里可以得出 eMMC 写速度为 12.9 MB/s。

- **读文件测试**

在嵌入式系统中，经常需要测试系统文件读写性能。读文件时忽略 cache 的影响。这时可以指定参数 iflag=direct，nonblock。

```
[root@myir:/]# time dd if=tempfile of=/dev/null bs=1M count=100 iflag=direct,nonblock

100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2.33601 s, 44.9 MB/s

real    0m2.347s
user    0m0.000s
sys     0m0.077s
```

可知，从磁盘直接读取读速度为 44.9 MB/s。

2.3.2. Nand 测试

1). 查看 nand 容量以及分区信息

通过 df 命令可以查询到 nand 分区信息、容量、使用情况和挂载目录等信息。

```
[root@myir:~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
ubi0_5          147M   49M   99M  33% /
tmpfs            53M   36K   53M   1% /tmp
tmpfs            53M   28K   53M   1% /run
devtmpfs         48M     0   48M   0% /dev
```

2). Nand 的性能测试

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **写文件**

性能测试主要测试 nand 在 linux 系统下对文件的读写速度，一般结合 time 与 dd 双命令进行测试。

```
[root@myir:~]# time dd if=/dev/zero of=write_file bs=5M count=1 conv=fsync
1+0 records in
1+0 records out
5242880 bytes (5.2 MB, 5.0 MiB) copied, 0.878978 s, 6.0 MB/s
real    0m 0.89s
user    0m 0.00s
sys     0m 0.86s
```

这里可以得出 nand 写速度为 6.0 MB/s。

- **读文件**

在嵌入式系统中，经常需要测试系统文件读写性能，读文件时忽略 cache 的影响。先执行下下面的命令清除 cache

```
[root@myir:~]# sync; echo 3 > /proc/sys/vm/drop_caches
```

在嵌入式系统中，经常需要测试系统文件读写性能。

```
[root@myir:~]# time dd if=write_file of=read_file bs=5M count=1
1+0 records in
1+0 records out
5242880 bytes (5.2 MB, 5.0 MiB) copied, 0.0556429 s, 44.2 MB/s
real    0m 0.06s
user    0m 0.00s
sys     0m 0.05s
```

这里可以得出 nand 读速度为 44.2 MB/s。

2.4. RTC

RTC（Real-time clock）本身是一个时钟，用来记录真实时间，当软件系统关机后保留系统时间并继续进行计时，系统重新开启后在将时间同步进软件系统。 MYD-YT113X 拥有内部 RTC,以及外部 RTC（RX8025），如果实际产品对 RTC 功耗要求不是很高，RTC 的测试通常采用 Linux 系统常用的 hwclock 和 date 命令配合进行，下面测试将系统时间写入 RTC，读取 RTC 时间并设置为系统时间并进行时间掉电保持的测试。

1) 查看系统 RTC 设备

```
[root@myir:/]# ls /dev/rtc* -al
crw-rw---- 1 root root 253, 0 Jan  1 00:00 /dev/rtc0
```

2) 设置系统时间

在将系统时间设置为 Mon Feb 7 09:28:00 UTC 2023 ：

```
[root@myir:/]# date 020709282023.00
Mon Feb  7 09:28:00 UTC 2023
```

- **将系统时间写入 RTC**

将上一步 date 命令设置的系统时间写入到 RTC 设备：

```
[root@myir:/]# hwclock -w
```

- **读取 RTC 时间并设置为系统时间**

```
[root@myir:/]# hwclock -r
Mon Feb  7 09:28:24 2023  0.000000 second
```

- **掉电保持 RTC 时间**

将开发板关机断开电源，经过几分钟左右，重新上电开机。查看 RTC 时间和系统时间：

```
[root@myir:/]# hwclock -r
Mon Feb  7 09:32:14 2023  0.000000 second
```

重新开机之后查看的 RTC 时间和系统时间比之前设置的时候增加了大约 20 分钟，说明 RTC 工作正常。如果需要详细测试 RTC 的精度，可以将断电时间延长如 24 小时，测试 RTC 时间与标准时间的差异。

- **将系统时间与 RTC 时间同步**

```
[root@myir:/]# hwclock -s
[root@myir:/]# date
Mon Feb  7 09:35:23 UTC 2023
```

如果将 hwclock-s 命令添加到启动脚本中就可以保证每次启动都可以将系统时间和 RTC 时间保持同步了。

2.5. Watchdog

Linux 内核包含 Watchdog 子系统，硬件设计过程中一般可以利用芯片内部的看门狗定时器或者使用外部看门狗芯片来实现 Watchdog 的功能，用于监测系统的运行。当系统出现异常情况无法喂狗时系统将可以进行自动复位。MYD-YT113X 芯片内部有 1 个看门狗，本章节将讲解 linux 下看门狗的测试方法。

1）应用程序测试看门狗

以下使用应用程序设置看门狗的超时时间和喂狗时间。（测试的应用程序源码在 04_Sources/Example/目录下)

- **设置看门狗超时时间**

通过 ioctl 实现超时时间，具体命令为：WDIOC_SETTIMEOUT，需要一个参数，即超时时间 timeout。使用示例：

```
ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
```

以上为设置看门狗当前超时时间的参考代码。其中 fd 为看门狗设备的文件句柄。

- **看门狗应用程序测试**

编译生产执行文件 watchdog 并拷贝到开发板，如下命令执行：

```
[root@myir:/]# ./watchdog
Usage: wdt_driver_test <timeout> <sleep> <test>
    timeout: value in seconds to cause wdt timeout/reset
    sleep: value in seconds to service the wdt
    test: 0 - Service wdt with ioctl(), 1 - with write()
```

运行看门狗应用,超时时间为 4s，每间隔 1s 喂一次狗：

```
[root@myir:/]# ./watchdog 4 1 0
Starting wdt_driver (timeout: 4, sleep: 1, test: ioctl)
Trying to set timeout value=4 seconds
The actual timeout was set to 4 seconds
Now reading back -- The timeout is 4 seconds
```

如果将上面的 1s 改到大于 4s，则超过了要求的 4s 喂狗时间，开发板会重启。

2.6. PMIC

本章节演示 Linux 电源管理的 Suspend 功能，让开发板睡眠 ，通过外部事件唤醒。Linux 内核一般提供了三种 Suspend：Freeze、Standby 和 STR(Suspend to RAM)，在用

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

户空间向”/sys/power/state”文件分别写入”freeze”、和”mem”，即可触发它们。MYD-YT113X 支持 freeze 和 mem 2 种方式。

1) 查看当前开发板支持的模式

```
[root@myir:/]# cat /sys/power/state
freeze mem
```

2) 在用户空间写入的方法

```
[root@myir:/]# echo "freeze" > /sys/power/state
[root@myir:/]# echo "mem" > /sys/power/state
```

- **mem 休眠**

输入休眠命令后开发板休眠，调试串口无法再输入，此时系统和设备状态保存到内存（处在自刷新模式，已保留其内容），所有设备进入低功耗模式。

```
[root@myir:/]# echo "mem" > /sys/power/state
```

- **freeze 休眠**

输入休眠命令后开发板休眠，调试串口无法再输入，此时用户空间被冻结，所有 I/O 设备进入低功耗状态，处理器进入空闲状态。

```
[root@myir:/]# echo "freeze" > /sys/power/state
```

此时，按下用户按键 S1，即可唤醒系统：

```
[root@myir:/]# [ 1558.783587] libphy: gmac1: probed
** 55 printk messages dropped **
[ 154.942339] 001: [DMIC]Enter sunxi_dmic_resume
```

此时调试串口可以重新输入。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

3. 基本外设接口

3.1. GPIO

GPIO 的测试是通过文件系统 sysfs 接口来实现的，下面内容以 PD20 为例说明 GPIO 的使用过程。

计算 gpio 对应引脚的数值= (n-1)*32+x(设 A 为 1，B 为 2，以此类推，H 对应的是 8)(PH2 中 x 表示 2)如:

PH2 对应的数值为: (8-1) 32+2=226 PH3 对应的数值为: (8-1) x32+3=227

设计底板的扩展接口 PD20

PD20 = (4 - 1) * 32 + 20 = 116

echo 116 > /sys/class/gpio/export

echo out > /sys/class/gpio116/direction

echo 1 > xxx/value (高电平)

echo 0 > xxx/value (低电平)

参数:

- function：复用
- data: 电平数据（1 代表高电平；0 代表低电平）
- plevel：驱动能力
- pull:上下拉情况

1). 导出 GPIO

```
[root@myir:/]# echo 117 > /sys/class/gpio/export
```

导出成功后会在/sys/class/gpio/目录下生成 gpio117 这个目录。

2). 设置/查看 GPIO 方向

- **设置输入**

```
[root@myir:/]# echo "in" > /sys/class/gpio/gpio117/direction
```

- **设置输出**

```
[root@myir:/]# echo "out" > /sys/class/gpio/gpio117/direction
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **查看 gpio 方向**

```
[root@myir:/]# cat /sys/class/gpio/gpio117/direction
out
```

返回 in 表示输入，返回 out 表示输出。

3). 设置/查看 GPIO 的值

- **设置输出低**

```
[root@myir:/]# echo "0" > /sys/class/gpio/gpio117/value
```

- **设置输出高**

```
[root@myir:/]# echo "1" > /sys/class/gpio/gpio117/value
```

- **查看 gpio 值**

```
[root@myir:/]# cat /sys/class/gpio/gpio117/value
1
```

可以看到 PD20 输出高电平，可以用万用表测量 J2 扩展 IO 的 PD20 引脚，可以看到电压为 3.3V 左右

3.2. LED 灯

Linux 系统提供了一个独立的子系统以方便从用户空间操作 LED 设备，该子系统以文 件的形式为 LED 设备提供操作接口。这些接口位于/sys/class/leds 目录下。在硬件资源 列表中，我们已经列出了开发板上所有的 LED。下面通过命令读写 sysfs 的方式对 LED 进行测试。下述命令均为通用命令，也是操控 LED 的通用方法。

1）操作 LED 的目录为/sys/class/leds

```
[root@myir:/sys/class/leds]# ls
led-blue    led-green
```

通过向/sys/class/leds/led-blue/brightness 写入不同的值可以改变心跳灯的亮灭占空比。

2）以心跳灯 led-blue 为例测试 LED

- **关闭心跳灯**

```
[root@myir:/sys/class/leds]# echo none > /sys/class/leds/led-blue/trigger
```

关闭心跳灯，然后可以单独进行 led 关闭和开启操作。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **熄灭 LED**

```
[root@myir:/sys/class/leds]# echo 1 > /sys/class/leds/led-blue/brightness
```

- **点亮 LED**

```
[root@myir:/sys/class/leds]# echo 0 > /sys/class/leds/led-blue/brightness
```

- **开启 LED 触发模式**

开启“heartbeat”模式后，LED 默认以 1Hz 周期闪烁，占空比为 50%：

```
[root@myir:/]# echo heartbeat > /sys/class/leds/led-blue/trigger
```

3.3. Key(按键)

Linux 的/dev/input/eventx 设备可以用来方便地调试鼠标、键盘、触摸板等输入设备。本节主要是测试 key。通过 hexdump 命令以及 dmesg 命令来查看按键是否有反应。MYD-YT113X 有两个按键，

S1 是 User 按键

S2 是复位键

1）设备树配置信息

打开配套的设备树文件 auto-t113-linux/device/config/chips/t113(t113_i)/configs/myir-image-yt113x-xxx/board.dts ，可以看到按键 S1 User 按键的节点：(xxx 代表不同的配置)

```
        gpio-keys {
                compatible = "gpio-keys";
                status = "okay";
                vol-down-key {
                        gpios = <&pio PE 1 GPIO_ACTIVE_LOW>;
                        linux,code = <114>;
                        label = "user key";
                        debounce-interval = <10>;
                        wakeup-source = <0x1>;
                };
        };
```

2）按键测试

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **查看对应的输入 event 信息**

```
[root@myir:/]# cat /proc/bus/input/devices
I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-keyboard"
P: Phys=sunxikbd/input0
S: Sysfs=/devices/virtual/input/input0
U: Uniq=
H: Handlers=kbd event0
B: PROP=0
B: EV=3
B: KEY=1000 800 c0000 0 0 10000000

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-tpadc"
P: Phys=
S: Sysfs=/devices/virtual/input/input1
U: Uniq=
H: Handlers=kbd event1
B: PROP=0
B: EV=100003
B: KEY=200000 0 0 0 0 0 0 0 8c0000 0 0 0

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-gpadc0"
P: Phys=sunxigpadc0/input0
S: Sysfs=/devices/virtual/input/input2
U: Uniq=
H: Handlers=event2
B: PROP=0
B: EV=100003
B: KEY=0

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
N: Name="sunxi-gpadc1"
P: Phys=sunxigpadc1/input0
S: Sysfs=/devices/virtual/input/input3
U: Uniq=
H: Handlers=event3
B: PROP=0
B: EV=11
B: MSC=10

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-ir"
P: Phys=sunxi-ir/input0
S: Sysfs=/devices/platform/soc@3000000/7040000.s_cir/rc/rc0/s_cir_rx
U: Uniq=
H: Handlers=kbd event4
B: PROP=20
B: EV=100017
B: KEY=2
B: REL=3
B: MSC=10

I: Bus=0000 Vendor=0000 Product=0000 Version=0000
N: Name="audiocodec sunxi Audio Jack"
P: Phys=ALSA
S: Sysfs=/devices/platform/soc@3000000/2030340.sound/sound/card0/input5
U: Uniq=
H: Handlers=kbd event5
B: PROP=0
B: EV=23
B: KEY=40 0 0 0 0 0 0 0 0 0 0 4 0 0 0 c0000 0 0 0
B: SW=14

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
```

```
N: Name="gpio-keys"
P: Phys=gpio-keys/input0
S: Sysfs=/devices/platform/gpio-keys/input/input7
U: Uniq=
H: Handlers=kbd event6
B: PROP=0
B: EV=3
B: KEY=40000 0 0 0
```

由上可以知 gpio-keys 的对应设备事件为 event6(用户根据实际信息找到 gpio-keys 名字来判断是 event 几)。

- **evtest 测试按键信息**

执行下面命令，操作按键 S1，串口终端会打印出如下信息：

```
[root@myir:/]# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:        sunxi-keyboard
/dev/input/event1:        sunxi-tpadc
/dev/input/event2:        sunxi-gpadc0
/dev/input/event3:        sunxi-gpadc1
/dev/input/event4:        sunxi-ir
/dev/input/event5:        audiocodec sunxi Audio Jack
/dev/input/event6:        gpio-keys
Select the device event number [0-6]: 6
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
Properties:
Testing ... (interrupt to exit)
```

```
Event: time 1245.584992, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 1
Event: time 1245.584992, ---------- SYN_REPORT ------------
Event: time 1245.724901, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 0
Event: time 1245.724901, ---------- SYN_REPORT ------------
Event: time 1246.034910, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 1
Event: time 1246.034910, ---------- SYN_REPORT ------------
Event: time 1246.174891, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 0
Event: time 1246.174891, ---------- SYN_REPORT ------------
Event: time 1246.324911, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 1
Event: time 1246.324911, ---------- SYN_REPORT ------------
Event: time 1246.454896, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 0
Event: time 1246.454896, ---------- SYN_REPORT ------------
```

每按一次 S1 当前终端会打印出当前事件码值，即按键正常。

3.4. USB

本节通过相关命令或热插拔、USB HUB 验证 USB Host 驱动的可行性，实现读写 U 盘的功能、usb 枚举功能。

1) 查看插入 usb 的打印信息

- **查看 USB 设备信息**

将 U 盘连接到开发板 USB Host 接口(J11)，提示信息如下:

```
[root@myir:~]# create /dev/sda
create /dev/sda1
```

从上述信息可以得出需要挂载的设备为 sda1,而且将会自动挂载/mnt/usb/sda1/目录下。

2) U 盘挂载读写

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **读文件**

需要提前在 U 盘上建立一个 test.txt 文件。

```
[root@myir:/]# cd /mnt/usb/sda1/
[root@myir:/mnt/usb/sda1:~]# ls
test.txt
[root@myir:/mnt/usb/sda1:~]# cat test.txt
hello!!!!
```

- **写文件**

```
[root@myir:/mnt/usb/sda1:~]# touch test
[root@myir:/mnt/usb/sda1:~]# echo " helloworld!!!" > test
[root@myir:/mnt/usb/sda1:~]# cp test /mnt
[root@myir:/mnt/usb/sda1:~]# cat /mnt/test
helloworld!!!
```

写完文件后需要执行下 sync 命令，确保数据完全写入到 U 盘里面之后，才可以卸载 U 盘设备。

3) 卸载 U 盘

- **卸载操作**

卸载 U 盘时要退出挂载目录

```
[root@myir:/]#umount /mnt/usb/sda1
```

3.5. Micro SD 卡

Micro SD Card，原名 Trans-flash Card(TF 卡)，Micro SD 卡是一种极细小的快闪存储器卡。Micro SD 卡相比标准 SD 卡，外形上更加小巧，是 SD 卡类型中尺寸最小的一种 SD 卡。尽管 Micro SD 卡的外形大小及接口形状与原来的 SD 卡有所不同，但接口规范保持不变，确保了兼容性。若将 Micro SD 插入特定的转接卡中，可当作标准 SD 卡来使用，SD 卡已成为目前消费数码设备中应用最广泛的一种存储卡，具有大容量、高性能、安全等多种特点的多功能存储卡。Micro SD 卡背面一般有 9 个引脚，包含 4 根数据线，支持 1bit/4bit 两种数据传输宽度。MYD-YT113X 支持 3 路 1bit 或者 4bit SDMMC 接口，开发板上使用 SDMMC0 连接 Micro SD。

1) 查看 TF 卡容量

通过 fdisk-l 命令可以查询到 TF 卡分区信息及容量：

```
[root@myir:/]# fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/sda: 30930944 sectors, 2815M
Logical sector size: 512
Disk identifier (GUID): d05e42e5-9ac6-4bc7-8c63-6d7c228073cb
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 30930910

Number  Start (sector)    End (sector)  Size  Name
     1            2048        30928895  14.7G  Basic data partitionm
```

2) 查看 TF 卡分区信息

通过 df 命令可以查询到 TF 卡分区信息，使用情况，挂载目录等信息：

EMMC 板型：

```
[root@myir:/]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       992M  332M  645M  34% /
tmpfs            53M  112K   53M   1% /tmp
tmpfs            53M   28K   53M   1% /run
devtmpfs         43M     0   43M   0% /dev
/dev/sda1        15G   32K   15G   1% /mnt/usb/sda1
```

Nand 板型：

```
[root@myir:/]# df -h
Filesystem      Size  Used Avail Use% Mounted on
ubi0_5          147M   53M   95M  36% /
tmpfs            53M  104K   53M   1% /tmp
tmpfs            53M   28K   53M   1% /run
devtmpfs         48M     0   48M   0% /dev
/dev/sda1        15G   32K   15G   1% /mnt/usb/sda1
```

- devtmpfs ： 用于系统创建 dev
- tmpfs：内存虚拟文件系统，挂载到不同的目录下

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

➢ /dev/sda1: TF 卡目录

3）TF 卡的性能测试

性能测试主要测试 TF 卡在 linux 系统下对文件的读写速度，一般结合 time 与 dd 双命令进行测试。挂载需要测试的 TF 卡分区，这里以最后一个分区/dev/mmcblk1p1 为例，挂载目录为/tmp。

- **写文件测试**

```
[root@myir:/tmp]# time dd if=/dev/zero of=test_write bs=1M count=100 conv=fsync
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 13.7005 s, 7.7 MB/s

real    0m13.712s
user    0m0.000s
sys     0m1.891s
```

这里测试出写磁盘速度为 7.7MB/s。

- **读文件测试**

```
[root@myir: /tmp]# time dd if=test_write of=/tmp/test_read bs=1M count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 19.1825 s, 5.5 MB/s

real    0m19.197s
user    0m0.000s
sys     0m2.419s
```

可知，直接从 SD 卡读数据速度为 5.5MB/s。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

3.6. ADC

GPADC 是 12bit 采样精度的模数转换模块，模拟输入范围具体由平台所定（T113X 平台为 1.8V），只有一路 GPADC 和四路 TPADC 通道。ADC 的测试是通过文件系统 sysfs 接口来实现的。

1). 读取 GPADC 测试值

首先我们选择 gpadc0 通道，使用下面命令直接读取默认值。

```
[root@myir:~]# cat /sys/class/gpadc/data
1583
```

得到的数据为 1583，转换为电压值为：1583/1000=1.583v

将 gpadc0 通道接入 0v 再次查看数据，得到电压值为 0v。

```
[root@myir:~]# cat /sys/class/gpadc/data
0
```

将 gpadc0 通道接入 1.8v 再次查看数据，得到电压值为 1.744v。

```
[root@myir:/]# cat /sys/class/gpadc/data
1744
```

2). 读取 TPADC 测试值

tpadc 一共有四路，这里我们选择 tpadc0 为例，首先使能 tpadc0 通道。

```
[root@myir:/]# echo 0 > /sys/devices/virtual/input/input1/channel_tpadc
```

然后读取默认值，得到的电压为 1709/1000=1.709v。

```
[root@myir:/]# cat /sys/devices/virtual/input/input1/tpadc
1709
```

将 TPADC0 接入 0v，得到电压接近为 0v。

```
[root@myir:/]# cat /sys/devices/virtual/input/input1/tpadc
2
```

将 TPADC0 接入 1.8v，得到电压为 1.752v。

```
[root@myir:/]# cat /sys/devices/virtual/input/input1/tpadc
1752
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

如果需要使用其他 tpadc 就先使能通道，然后测量。其他三路 tpadc 使能命令如下，测量方法与 tpadc0 一致。

```
[root@myir:/]# echo 1 > /sys/devices/virtual/input/input1/channel_tpadc
[root@myir:/]# echo 2 > /sys/devices/virtual/input/input1/channel_tpadc
[root@myir:/]# echo 3 > /sys/devices/virtual/input/input1/channel_tpadc
```

3.7. Display

本模块由显示引擎（DE）和各类型控制器（tcon）组成。输入图层（layers）在 DE 中进行显示相关处理后，通过一种或多种接口输出到显示设备上显示，以达到将众多应用渲染的 图层合成后在显示器呈现给用户观看的作用。DE 有 2 个独立单元（可以简称 de0、de1），可以 分别接受用户输入的图层进行合成，输出到不同的显示器，以实现双显。DE 的每个独立的单元有 1-4 个通道（典型地，de0 有 4 上，de1 有 2 个），每个通道可以同时处理接受 4 个格式相同的 图层。sunxi 平台有视频通道和 UI 通道之分。视频通道功能强大，可以支持 YUV 格式和 RGB 图层。UI 通道只支持 RGB 图层。简单来说，显示模块的主要功能如下：

- 支持 lcd(hv/lvds/cpu/dsi) 输出
- 支持双显输出
- 支持多图层叠加混合处理
- 支持多种显示效果处理（alpha, colorkey，图像增强，亮度/对比度/饱和度/色度调整）
- 支持智能背光调节
- 支持多种图像数据格式输入（argb, yuv）
- 支持图像缩放处理 • 支持截屏 • 支持图像转换

1）设备树配置信息

打开配套的设备树文件 auto-t113-linux/device/config/chips/t113/configs/myir_xxx 下的 board.dts 和 uboot-board.dts，可以看到两种显示方案。(xxx 代表不同板型配置)

```
#include "sun8iw20p1.dtsi"
#include "myir-t113-lvds.dtsi"                    7 寸 LVDS 显示
//#include "myir-t113-lvds-dual.dtsi"             19 寸双路 LVDS 显示
```

2）显示方案组合

- **默认显示**

MYD-YT113X 默认 7 寸 LVDS 显示。

- **19 寸双路 LVDS 显示**

编译前在设备树 board.dts 和 uboot-board.dts 选择 19 寸双路 LVDS 显示方案，并注释 7 寸 LVDS 显示方案。（不能同时选择两个显示方案）

```
//#include "myir-t113-lvds.dtsi"
#include "myir-t113-lvds-dual.dtsi"
```

最后双路 LVDS 显示还需要修改 env.cfg 文件，在 bootcmd 添加 lvds_if_reg。

```
$HOME/T113X/auto-t113x-linux/device/config/chips/t113/configs/ myir-image-yt113s3-emmc-full/longan/env.cfg

.....
boot_dsp0=sunxi_flash read 43000000 ${dsp0_partition};bootr 43000000 0 0
boot_normal=sunxi_flash read 43000000 boot;bootm 43000000
boot_recovery=sunxi_flash read 43000000 recovery;bootm 43000000
boot_fastboot=fastboot
lvds_if_reg=mw.l 0x05461084 0xE0100000

#uboot system env config
bootdelay=3
#default bootcmd, will change at runtime according to key press
#default nand boot
bootcmd=run lvds_if_reg setargs_mmc boot_dsp0 boot_normal
```

3.8. Touch Panel

触摸有电容触摸和电阻触摸，MYD-YT113X 开发板硬件目前不支持电阻触 摸，但是支持电容触摸，米尔科技提供 LVDS 显示屏配件，见表 1-1。可根据实际需 求自行购买配件。电容屏在使用中较为灵敏，很少出现问题。另外，电容屏不需要较准。 因为根据电容屏的原理，电容屏在使用中是可以准确的识别出手指与屏幕接触的位置，具有很高的灵敏性。我们在使用中如果出现点击软件选不中的现象，一般只有一种情况：屏幕出现了问题。下面是通过 evtest 命令测试电容屏触摸功能的简单测试

1）evtest 命令测试

终端执行“evtest”进入测试界面。选择测试外设为触摸屏，这里默认为输入中断 “5”，测试界面选择 5 按下回车即可开始测试：（用户请根据自己实际 evtest 显示出的触摸设备进行选择）

```
[root@myir:/]# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      sunxi-keyboard
/dev/input/event1:      sunxi-tpadc
/dev/input/event2:      sunxi-gpadc0
/dev/input/event3:      sunxi-ir
/dev/input/event4:      audiocodec sunxi Audio Jack
/dev/input/event5:      generic ft5x06 (79)
/dev/input/event6:      gpio-keys
Select the device event number [0-6]: 5
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0
Input device name: "generic ft5x06 (79)"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 0 (ABS_X)
      Value      0
      Min        0
      Max     1023
    Event code 1 (ABS_Y)
      Value      0
      Min        0
      Max      599
    Event code 47 (ABS_MT_SLOT)
      Value      0
```

```
      Min          0
      Max          4
    Event code 53 (ABS_MT_POSITION_X)
      Value        0
      Min          0
      Max       1023
    Event code 54 (ABS_MT_POSITION_Y)
      Value        0
      Min          0
      Max        599
    Event code 57 (ABS_MT_TRACKING_ID)
      Value        0
      Min          0
      Max      65535
Properties:
  Property type 1 (INPUT_PROP_DIRECT)
Testing ... (interrupt to exit)
Event: time 205.770226, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), val
ue 0
Event: time 205.770226, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), valu
e 405
Event: time 205.770226, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), valu
e 180
Event: time 205.770226, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
Event: time 205.770226, type 3 (EV_ABS), code 0 (ABS_X), value 405
Event: time 205.770226, type 3 (EV_ABS), code 1 (ABS_Y), value 180
```

3.9. Ethernet

Linux 下网络配置的工具很多，常见的有 net-tools，iproute2，systemd-networkd，network manager 以及 connman 等，这些都可以在系统定制的时候根据实际需要进行选择。MYD-YT113X 具有一路千兆网络端口 eth0。

1）配置以太网 IP 地址

- **使用 net-tools 工具包中的 ifconfig 对网络进行手动配置**

首先通过通过 ifconfig 命令查看网络设备信息如下：

```
[root@myir:/]# ifconfig
eth0      Link encap:Ethernet  HWaddr 32:A9:D0:70:C2:14
          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:37

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  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)
```

下面介绍给 eth0 手动配置 IP 地址 192.168.0.100 的方法，命令如下：

```
[root@myir:/] # ifconfig eth0 192.168.0.100 netmask 255.255.255.0 up
```

上面的命令手动配置 eth0 的 IP 地址为 192.168.0.100，子网掩码为 255.255.255.0，以及默认配置的广播地址 192.168.0.255，并通过 up 参数进行激活，如下所示：

```
[root@myir:/]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 32:A9:D0:70:C2:14
          inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
          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:37
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **使用 iproute2 工具包中的 ip 命令对网络进行手动配置**

ifconfig 命令手动设置 IP 地址的方法也可以使用 ip addr 和 ip link 进行替代，更多 的信息请查看 https://wiki.linuxfoundation.org/networking/iproute2 中的说明。

```
[root@myir:/]# ip addr flush dev eth0
[root@myir:/]# ip addr add 192.168.0.101/24 brd + dev eth0
[root@myir:/]# ip link set eth0 up
```

如果之前已经配置过 IP 地址，再使用 ip addr add 配置的 IP 地址将会成为 Seco nd ary 地址，所以这里先使用 ip addr flush 清除之前的地址之后再进行配置然后激活。完成配置之后，通过 ip addr show 命令查看 eth0 信息如下：

```
[root@myir:/]#  ip addr show eth0
3: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN group default qlen 1000
    link/ether 32:a9:d0:70:c2:14 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.101/24 brd 192.168.0.255 scope global eth0
       valid_lft forever preferred_lft forever
```

2）修改 Mac 地址

手动修改 Mac 地址 00:0C:29:36:97:20 的方法，命令如下 ：

```
[root@myir:/]# ifconfig eth0 down
[root@myir:/]# ifconfig eth0 hw ether 00:0C:29:36:97:20
[root@myir:/]# ifconfig eth0 up
[root@myir:/]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:36:97:20
          inet addr:192.168.0.101  Bcast:192.168.0.255  Mask:255.255.255.0
          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:37
```

4. 扩展外设接口

MYD-YT113X 开发板提供了丰富的外设接口，除了基本的外设接口，还可以外接各种扩展模块。使用户的开发更加灵活方便。下面介绍米尔推出的几种选配模块的测试步骤。用户根据需求自行购买。选配模块详情参考表 1-1 选配模块列表。

4.1. MY-WiredCom 模块

MY-WiredCom 模块是米尔推出的树莓派外设接口形式，包含 RS232/RS485/CAN/SPI/I2C 等外设接口。该模块需要用户根据需要自行购买，模块详情请参考表 1-1 选配模块列表。测试前用户需要先将该模块连接到开发板 J2 接口。

1). RS485 测试

本节将使用 Linux API 配置开发板 RS485 的收发功能。Linux 的串口设备文件一般命名为/dev/ttyASn(n=0,1,2,3.....)。n 表示串口在 Linux 系统中的设备编号，“ttyAS”是内核已经定义好的串口设备名字。本节是以 MY-WiredCom 扩展板上 J2 接口为例进行测试，J2 接口的编号设备节点为 ttyAS4。其测试配置如下表：

表 4-1. RS232 接口配置

测试点	MYD-YT113X	Windows 10
硬件接口	RS485	USB-RS485 模块
设备节点	ttyAS4	com12
测试软件	rs485_read、rs485_write	sscom

这里将 MY-WiredCom 模块 J2 座子的 485A 和 485B 分别与 USB-RS485 转换器的 485A 和 485B 连接。

(rs485_read、rs485_write 测试例程在 04_Sources/Example/目录下)。

- **测试开发板 RS485 收数据**

首先设置 windows 下串口工具 sscom 定时 发送字符串，例如间隔 1000ms 发送字符串 “1234567890” 。

开发板准备接收数据，在开发板上面执行下面命令，接收数据。

命令执行完成后中断 会进入一个阻塞状态，等待接收电脑串口发送过来的数据,当收到来自电脑串口的数据后，打印出接收到的字符串 “1234567890” 在终端显示。

```
[root@myir:/]# ./rs485_read -d /dev/ttyAS4 -b 115200
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
RECV[05]: 0x12 0x34 0x56 0x78 0x90
RECV[05]: 0x12 0x34 0x56 0x78 0x90
RECV[05]: 0x12 0x34 0x56 0x78 0x90
RECV[05]: 0x12 0x34 0x56 0x78 0x90
RECV[05]: 0x12 0x34 0x56 0x78 0x90
RECV[05]: 0x12 0x34 0x56 0x78 0x90
RECV[05]: 0x12 0x34 0x56 0x78 0x90
```

- **测试开发板 RS232 发数据**

开发板上执行如下命令，会向 uart4 发送“0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13”数据，此时 windowns 下串口工具会接收到该数据。

开发板发送数据前先执行下面命令

```
[root@myir:/]# echo 238 > /sys/class/gpio/export
[root@myir:/]# echo out > /sys/class/gpio/gpio238/direction
[root@myir:/]# echo 1 > /sys/class/gpio/gpio238/value
```

然后执行发送命令

```
[root@myir:/]# ./rs485_write -d /dev/ttyAS4 -b 115200

SEND[20]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0
x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13
SEND[20]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0
x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13
SEND[20]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0
x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13
SEND[20]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0
x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13
SEND[20]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0
x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13
```

开发板上收发的数据与 windows 下 sscom 收发的数据相对应，即开发板 RS485 接收数据正常。

2). CAN 测试

本节采用 Linux 系统常用的 cansend、candump 命令进行 SocketCAN 的通讯测试。这里测试使用的两块开发板对接测试。

这里使用 MY-WiredCom 模块 J2 座子的 CANH、CANL 引脚和同类型的板子 CANH、CANL 相连。

- **初始化 CAN 网络接口**

设置 CAN 波特率，使用 can 需要先设置波特率，并开启 CAN 网络接口。参考下面命令分别将两块开发板的数据波特率设置为 20KHz，并开启 can 功能：（最高测试过 1M）

```
[root@myir:/]# ifconfig can0 down
[root@myir:/]# ip link set can0 type can bitrate 20000
[root@myir:/]# ip link set can0 up
```

至此，即开启 can 功能。

- **发送数据**

设置其中一块板子为发送，并使用 cansend 发送数据：

```
[root@myir:/]# cansend can0 123#1122334455667788
[root@myir:/]# cansend can0 123#1122334455667788
[root@myir:/]# cansend can0 123#1122334455667788
[root@myir:/]# cansend can0 123#1122334455667788
[root@myir:/]# cansend can0 123#1122334455667788
[root@myir:/]# cansend can0 123#1122334455667788
[root@myir:/]# cansend can0 123#1122334455667788
```

- **接收数据**

设置另外一块板子为接收，可以使用 candump 来查看 CAN 的接收数据：

```
[root@myir:/]# candump can0 -L
(0000000274.962878) can0 123#1122334455667788
(0000000275.568764) can0 123#1122334455667788
(0000000276.088995) can0 123#1122334455667788
(0000000276.578937) can0 123#1122334455667788
(0000000277.069075) can0 123#1122334455667788
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **统计 can0 信息**

CAN 数据收发之后显示 CAN 设备的详情和收发统计信息，其中“clock”的值代表 can 的时钟，”drop”的值代表丢包，”overrun”的值代表溢出，”error”代表总线错误。

```
[root@myir:/]# ip -details -statistics link show can0
2: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mod
e DEFAULT group default qlen 10
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 20000 sample-point 0.850
          tq 2500 prop-seg 8 phase-seg1 8 phase-seg2 3 sjw 1
          sun8i-can: tseg1 1..16 tseg2 1..8 sjw 1..4 brp 1..64 brp-inc 1
          clock 24000000
          re-started bus-errors arbit-lost error-warn error-pass bus-off
          0          3          0          1          1          0          nu
mtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535
    RX: bytes  packets  errors  dropped overrun mcast
    72         9        3       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    32         4        0       0       0       0
```

4.2. Wi-Fi 测试

1) Wi-Fi 测试

本节主要介绍 Linux 下 Wi-Fi 的配置和使用，通常 Wi-Fi 模块可以支持两种工作模式，分别是 STA 模式和 AP 模式，有些设备还支持 STA 和 AP 模式同时工作。STA 模式允许设备连接外部 Wi-Fi 热点，AP 模式将设备变成 Wi-Fi 热点，供其它设备连接。

2) STA 模式连接 WiFi 热点

下面尝试手动连接附近的 Wi-Fi 热点 SSID，这是一个采用 WPA2 加密方式 的 Wi-Fi 热点,请自行配置。

确保 wlan0 网络设备处于激活状态 。

```
[root@myir:/]# ifconfig wlan0 up
```

- **扫描附近 WiFi 热点**

扫描附近的 wifi 热点，得到附近 Wi-Fi 热点列表如下：

```
[root@myir:/]#  iw dev wlan0 scan | grep SSID
        SSID: CMCC-u37Z
        SSID: MYIR
        SSID: HUAWEI-400P0Y
        SSID: CMCC-u37Z-5G
        SSID: MYIR_5G
                * SSID List
        SSID: HUAWEI-400P0Y
```

- **wpa_passphrase 设置 wifi 名字和密码**

```
[root@myir:/]# wpa_passphrase 账号 密码 >> /etc/wpa_supplicant. conf
[root@myir:/usr/lib/ltp-testsuite/network]# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1

network={
        ssid="MYIR"
        #psk="MYIR88888888"
        psk=8c66c06459ebe47f74efcebbb3a7ed74b144e63028fb7947dcead44ed
1714c23
}
```

从一个 SSID 的 ASCII 密码生成一个 WPA PSK 进行加密操作。

- **关掉 wpa_supplicant 进程**

使用 wpa_supplicant 连接、配置 WIFI 之前，需要先关掉 wpa_supplicant 进程：

```
[root@myir:/]# killall wpa_supplicant
```

- **初始化 wpa_supplican**

wpa_supplicant 是一个连接、配置 WIFI 的工具，它的主要工作是通过 socket 与驱动交互并上报数据给用户层，而用户层也可以通过 socket 发送命令给 wpa_supplicant 调 动驱动来对 WiFi 芯片操作。它通常在后台运行，如下所示：

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
[root@myir:/]# wpa_supplicant -B -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0
```

- -B ： 在后台运行守护进程
- -D ： 驱动名称
- -c ： 配置信息的路径
- -i ： 监听的 wifi 接口

- **获取 ip 地址**

配置完成后运行 WiFi 获取脚本/etc/test/wifi-on.sh 即可连接 wifi,如下所示：

```
[root@myir:/]# udhcpc -b -i wlan0 -R
udhcpc: started, v1.29.3
udhcpc: sending discover
udhcpc: sending select for 192.168.0.166
udhcpc: lease of 192.168.0.166 obtained, lease time 86400
deleting routers
adding dns 192.168.0.1
adding dns 192.168.0.1
```

- **Ping 百度检查是否可以正常连通**

```
[root@myir:~]# ping www.baidu.com -I wlan0
PING www.baidu.com (36.152.44.95): 56 data bytes
64 bytes from 36.152.44.95: seq=0 ttl=55 time=17.005 ms
64 bytes from 36.152.44.95: seq=1 ttl=55 time=14.794 ms
64 bytes from 36.152.44.95: seq=2 ttl=55 time=21.479 ms
```

4.3. 4G /5G 模块

Linux 设备也可以外接 4G 或者 5G 模块来拨号上网，MYD-YT113X 开发板使用的是 EM05-CE 4G 模块和 RM500Q-CN 5G 模块。

拨号方式有 pppd，gobinet 以及 qmi_wwan 3 种方式，其中 pppd 比较通用，gobinet 没有用到，qmi_wwan 连接比较快，并且可以用来做 5G 模块连接，下面以 EM05-CE 模块为例进行说明，测试前要先把模块连接到开发板的 J16 接口，另外还需给 4G 模块连接好 IPEX4 代的天线。

YT113X 有两个 SIM 卡槽，分别是 J16 和 J17。4G 模块只能使用 SIM 1（J16）。5G 模块可以同时接入两张 SIM 卡，但是不能同时使用 SIM 1 和 SIM 2 进行拨号，只能通过命令切换 SIM 卡，选择使用 SIM 1 还是 SIM 2，后面会介绍如何切换 SIM 卡使用。

1). 查看 VID 和 PID

```
[root@myir:/]# lsusb
Bus 001 Device 001: ID 1d6b:0002
Bus 001 Device 003: ID 2c7c:0125
Bus 001 Device 002: ID 1a40:0101
Bus 002 Device 001: ID 1d6b:0001
Bus 001 Device 004: ID 0bda:b733
```

2c7c:0125 ： EM05-CE 的 VID 和 PID 信息。

2c7c:0800 ： RM500Q-CN 的 VID 和 PID 信息。

2). 查看 kernel 识别模块

如果 kernel 增加了此模块的 VID 和 PID 配置，那么会生成/dev/ttyUSB*的节点：

```
[root@myir:/]# ls -l /dev/ttyUSB*
crw-------    1 root     root      188,   0 Jan  1 00:00 /dev/ttyUSB0
crw-------    1 root     root      188,   1 Jan  1 00:00 /dev/ttyUSB1
crw-------    1 root     root      188,   2 Jan  1 00:00 /dev/ttyUSB2
crw-------    1 root     root      188,   3 Jan  1 00:00 /dev/ttyUSB3
```

3). 使用 AT 指令进行初步测试

使用 AT 指令可以方便的来查询信号强度，是否插入 SIM 卡，SIM 卡当前是否搜索到运营商，也可以用 AT 来打电话测试下当前卡功能。 这里进行 AT 通讯还需要知道哪个设备是通讯口，这里需要查询模块文件，EM05 和 RM500Q 采用 ttyUSB2 进行 AT 通讯。这里采用 microcom 举例，也可以用 minicom。如中断执行 microcom /dev/ttyUSB2 命令进入 AT 指令模式、键盘按 ctrl+x 退出 AT 指令模式。

- **查询信号质量**

```
[root@myir:/]# microcom /dev/ttyUSB2
at+csq
+CSQ: 27,99
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
OK
```

➢ 27,99: 27 就是信号质量，范围在 0~31 之间（99 表示无信号）数字越大代表信号越强。

- **查询卡是否识别**

```
at+cpin?
+CPIN:  READY

OK
```

➢ +CPIN:READY：READY 代表就绪。

- **查看运营商**

```
at+cops?
+COPS:  0,0,"CHN-UNICOM",7

OK
```

➢ CHN-UNICOM,7: CHN-UNICOM 代表联通， 7 代表采用 2G，3G，4G，还是 5G 根据模块手册查看。

如果上述 3 步都能正常，就可以进行拨号上网。

- **查看卡槽**

有时候 sim 卡插在卡槽一（ j16）上，但是识别在卡槽二（j17）上，这时拨号会失败。下面命令是查看当前卡槽位置和切换卡槽

查询卡槽

```
[root@myir:/]#  microcom  /dev/ttyUSB2
AT+QUIMSLOT?
+QUIMSLOT:  1

OK
```

切换卡槽为 1

```
[root@myir:/]#  microcom  /dev/ttyUSB2

AT+QUIMSLOT=1
```

```
OK
+CPIN: NOT READY

+QSIMSTAT: 1,0

+QSIMSTAT: 1,1

+CPIN: READY

+QUSIM: 1

+QIND: SMS DONE

+QIND: PB DONE
```

4). ppp 拨号测试

这里采用开发板自带的 pppd 拨号命令：

```
[root@myir:/]# pppd call quectel-ppp &
[1] 3349
[root@myir:/]# pppd options in effect:
debug              # (from /etc/ppp/peers/quectel-ppp)
nodetach                  # (from /etc/ppp/peers/quectel-ppp)
dump               # (from /etc/ppp/peers/quectel-ppp)
noauth             # (from /etc/ppp/peers/quectel-ppp)
user test                 # (from /etc/ppp/peers/quectel-ppp)
password ??????           # (from /etc/ppp/peers/quectel-ppp)
remotename 3gppp                  # (from /etc/ppp/peers/quectel-ppp)
/dev/ttyUSB3              # (from /etc/ppp/peers/quectel-ppp)
115200             # (from /etc/ppp/peers/quectel-ppp)
lock               # (from /etc/ppp/peers/quectel-ppp)
connect chat -s -v -f /etc/ppp/peers/quectel-chat-connect                 # (fro
m /etc/ppp/peers/quectel-ppp)
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
disconnect chat -s -v -f /etc/ppp/peers/quectel-chat-disconnect          # (fro
m /etc/ppp/peers/quectel-ppp)
nocrtscts                    # (from /etc/ppp/peers/quectel-ppp)
modem                # (from /etc/ppp/peers/quectel-ppp)
hide-password                # (from /etc/ppp/peers/quectel-ppp)
novj                # (from /etc/ppp/peers/quectel-ppp)
novjccomp                    # (from /etc/ppp/peers/quectel-ppp)
ipcp-accept-local                    # (from /etc/ppp/peers/quectel-ppp)
ipcp-accept-remote                    # (from /etc/ppp/peers/quectel-ppp)
ipparam 3gppp                # (from /etc/ppp/peers/quectel-ppp)
noipdefault                # (from /etc/ppp/peers/quectel-ppp)
ipcp-max-failure 30                    # (from /etc/ppp/peers/quectel-ppp)
defaultroute                # (from /etc/ppp/peers/quectel-ppp)
usepeerdns                    # (from /etc/ppp/peers/quectel-ppp)
noccp                # (from /etc/ppp/peers/quectel-ppp)
abort on (BUSY)
abort on (NO CARRIER)
abort on (NO DIALTONE)
abort on (ERROR)
abort on (NO ANSWER)
timeout set to 30 seconds
send (AT^M)
expect (OK)
AT^M^M
OK
 -- got it

send (ATE0^M)
expect (OK)
^M
ATE0^M^M
OK
 -- got it
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
send (ATI;+CSUB;+CSQ;+CPIN?;+COPS?;+CGREG?;&D2^M)
expect (OK)
^M
^M
Quectel^M
EM05^M
Revision: EM05CEFCR06A04M1G_ND^M
^M
SubEdition: V01^M
^M
+CSQ: 27,99^M
^M
+CPIN: READY^M
^M
+COPS: 0,0,"CHN-UNICOM",7^M
^M
+CGREG: 0,1^M
^M
OK
 -- got it

send (AT+CGDCONT=1,"IP","3gnet",,0,0^M)
expect (OK)
^M
^M
OK
 -- got it

send (ATD*99#^M)
expect (CONNECT)
^M
^M
```

```
CONNECT
 -- got it

Script chat -s -v -f /etc/ppp/peers/quectel-chat-connect finished (pid 3355), st
atus = 0x0
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB3
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x638d4bf3> <pcomp>
<accomp>]
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0x25c
e332e> <pcomp> <accomp>]
sent [LCP ConfAck id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0x25c
e332e> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x638d4bf3> <pcomp>
<accomp>]
rcvd [LCP DiscReq id=0x1 magic=0x25ce332e]
rcvd [CHAP Challenge id=0x1 <d805e32118c8f48f5a286b7a3d8f3395>, name =
 "UMTS_CHAP_SRVR"]
sent [CHAP Response id=0x1 <168e53c0728c7137458e5c745361ab8f>, name =
 "test"]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.
0>]
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfNak id=0x1 <addr 10.229.255.26> <ms-dns1 218.104.111.122>
 <ms-dns2 218.106.127.114>]
sent [IPCP ConfReq id=0x2 <addr 10.229.255.26> <ms-dns1 218.104.111.122>
 <ms-dns2 218.106.127.114>]
```


```
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfAck id=0x2 <addr 10.229.255.26> <ms-dns1 218.104.111.122>
 <ms-dns2 218.106.127.114>]
Could not determine remote IP address: defaulting to 10.64.64.64
local  IP address 10.229.255.26
remote IP address 10.64.64.64
primary  DNS address 218.104.111.122
secondary DNS address 218.106.127.114
Script /etc/ppp/ip-up started (pid 3369)
Script /etc/ppp/ip-up finished (pid 3369), status = 0x0
```

可以看到已经正常连接，能获取 IP。

```
[root@myir:/]# ifconfig
eth0      Link encap:Ethernet  HWaddr 76:93:32:78:12:87
          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:37

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  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)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.229.255.26  P-t-P:10.64.64.64  Mask:255.255.255.255
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:
1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:52 (52.0 B)  TX bytes:58 (58.0 B)
```

- ppp0 ： ppp0 即为拨号网卡设备，ip 地址正常获取。

5). Ping 外网测试

```
[root@myir:/]# ping www.baidu.com -I ppp0
PING www.baidu.com (112.80.248.76): 56 data bytes
64 bytes from 112.80.248.76: seq=0 ttl=55 time=56.226 ms
64 bytes from 112.80.248.76: seq=1 ttl=55 time=54.201 ms
64 bytes from 112.80.248.76: seq=2 ttl=55 time=61.772 ms
64 bytes from 112.80.248.76: seq=3 ttl=55 time=30.343 ms
64 bytes from 112.80.248.76: seq=4 ttl=55 time=41.040 ms
^C
--- www.baidu.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 30.343/48.716/61.772 ms
```

5. 网络应用

开发板出厂烧录的镜像默认包含了一些常见的网络应用程序，方便用户进行开发或调试。

5.1. PING

PING 主要用来测试网络的连通性，也可以测试网络延迟以及丢包率。配置好以太网连接之后就可以使用 PING 对网络连接进行简单的测试。

1）接线与信息输出

通过 CAT6 网线将开发板连接到交换机或路由器，控制台会显示内核输出的连接信息，如下：（通过测试 CAT6 类网线速度稳定，使用其他网线速度可能不稳定）

```
[root@myir:~]# dmesg
[  844.954902] 000: sunxi-gmac 4500000.eth eth0: Link is Up - 1Gbps/Full - flow control off
```

2）测试外网网址

```
[root@myir:/]# ping www.baidu.com
PING www.baidu.com (36.152.44.95): 56 data bytes
64 bytes from 36.152.44.95: seq=0 ttl=55 time=12.555 ms
64 bytes from 36.152.44.95: seq=1 ttl=55 time=12.235 ms
64 bytes from 36.152.44.95: seq=2 ttl=55 time=12.169 ms
^C
--- www.baidu.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 12.169/12.319/12.555 ms
```

注：ping 公网需要确保 DNS 正常工作。

上面结果显示 www.baidu.com 经过域名解析之后的 IP 地址为 36.152.44.95，icmp_seq 代表 icmp 包的编号，如果编号连续说明没有丢包；time 代表响应的延迟时间，当然这个时间越短越好。除了对以太网进行测试，ping 命令也可以用于测试 Wi-Fi。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

5.2. SSH

SSH 为 Secure Shell 的缩写，由 IETF 的网络小组（Network Working Group）所制定；SSH 为建立在应用层基础上的安全协议， 是较可靠，专为远程登录会话和其他 网络服务提供安全性的协议。通常 Linux 平台下使用 dropbear 或 OpenSSH 来实现 SSH 的服务端和客户端。下面在以太网连接上分别测试 SSH 客户端和服务端的使用。当前出厂 默认包含 openssh 7.6p1（http://www.openssh.com/）提供的客户端和服务程序。 首先配置好开发板以太网接口到 SSH 服务器的连接，配置后的以太网卡地址如下：

```
[root@myir:/]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr CE:6F:68:C9:B6:CA
          inet addr:192.168.0.134  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::bd96:938:88b8:ebb6/64 Scope:Link
          inet6 addr: 2409:8a4d:c7d:8002:9df:a35c:219b:17e3/64 Scope:Global
          inet6 addr: 2409:8a4d:c7d:8002::3/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27 errors:0 dropped:0 overruns:0 frame:0
          TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2820 (2.7 KiB)  TX bytes:3994 (3.9 KiB)
          Interrupt:37
```

当前 SSH 服务器的 IP 地址为 192.168.0.134，用 ping 命令可测试开发板和 SSH 服务器 之间的连接是否正常。

- **SSH 客户端测试**

开发板作为客户端连接 SSH 服务器，在开发板上使用 ssh 命令登陆服务器，命令和 结果如下：

```
[root@myir:/]# ssh zhaoy@192.168.1.13
The authenticity of host '192.168.0.179 (192.168.0.179)' can't be established.
ECDSA key fingerprint is SHA256:i8HTmmrawvyVAgfWzHuJKCTiZiO/KgDNbpFb
XRatY/U.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.179' (ECDSA) to the list of known hos
ts.
```

```
sur@192.168.0.179's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-84-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

37 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
*** System restart required ***
Last login: Wed May  3 17:23:28 2023 from 192.168.0.131
sur@myir:~$
```

其中 sur 为服务器上的用户名。

登录成功之后，自动进入 SSH 服务器上的 console 控制台，用户就可以在客户端对 远程服务器执行 sur 用户权限内的控制。如果需要退出，直接在当前控制台执行"exit"命 令即可。

- **SSH 服务端测试**

开发板作为 SSH 服务端，其它外部设备远程连接到此台开发板。 开发板端默认启动了 SSH 服务，因此我们也可以在其它具有 SSH 客户端的外部设备 （开发板或者 PC）上使用 ssh 命令登陆到当前的开发板上，命令和结果如下：(首次连接需要设置密码，当前开发板是无密码的)

```
sur@myir:~$ ssh root@192.168.0.102
The authenticity of host '192.168.0.102 (192.168.0.102)' can't be established.
```

```
ECDSA key fingerprint is SHA256:0RvKGGFq8awWygR0xLg9zviXlru1Ctk3FVN7+qkpGIQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.102' (ECDSA) to the list of known hosts.
root@192.168.0.102's password:
COLUMNS=145;LINES=49;export COLUMNS LINES;
[root@myir:~]#
```

上面的示例中，我们从远程以 root 账户登录到了此开发板上，并进入 console 控制台，可以对开发板执行 root 用户权限内的控制。如果需要退出，直接在控制台执行"exit " 命令即可。 OpenSSH 是使用 SSH 协议远程登录的主要连接工具。它加密所有流量以消除窃 听、连接劫持和其他攻击。此外，OpenSSH 还提供一系列大型安全隧道功能、多种身份 验证方法和复杂灵活的配置选项。用户可以根据自身需要修改位于电脑主机/etc/ssh/目录 下的配置文件 ssh_config 和 sshd_config。 例如，如果希望 SSH 服务端允许 root 账户不用密码远程登录，则可以修改 SSH 服务 端上的/etc/ssh/sshd_config，添加下面两行配置。 PermitRootLogin yes PermitEmptyPasswords yes 上面的配置有比较大的安全风险，一般用于调试阶段远程部署。实际产品中考虑到安 全性，一般都是关掉的。

5.3. SCP

SCP 是 Secure Copy 的缩写，它是 linux 系统下基于 SSH 协议的安全的远程文件拷贝命令，在系统调试阶段非常实用。 我们已经介绍过使用 SSH 协议以及 SSH 客户端和服务端进行远程登录的示例，这里 再介绍通过 SCP 命令进行文件远程拷贝的示例：

1）从远程拷贝文件到本地

```
sur@myir:~$ scp test root@192.168.0.102:/root
root@192.168.0.102's password:
test
                                      100%   13        3.0KB/s   00:00
```

进入开发板“/root”目录可以看到此文件，如下：

```
[root@myir:/root]# ls
test
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

2）从本地拷贝文件到远程

```
[root@myir:/root]# scp test sur@192.168.0.179:~/
sur@192.168.1.13's password:
```

拷贝的过程中需要按照提示输入，验证成功之后文件从开发板上拷贝到服务器上指定账户的$HOME 目录

```
sur@myir:~$ ls
test
```

通过添加”-r“参数，还可以进行目录的拷贝，具体操作请参照 scp 命令的帮助

5.4. TFTP

TFTP 使用客户端和服务器软件在不同设备之间进行连接和传输文件，TFTP 使用的是 UDP 协议，不具备登录功能，它非常简洁，特别适合在设 备和服务器端传输和备份固件，配置文件等信息。例如常见的 u-boot 中就支持 TFTP 协 议，可以通过网络加载服务器端的 Linux 系统并实现网络启动的功能。 默认的镜像文件包含 busybox 提供的 tftp 客户端程序，其命令语法如下：

```
[root@myir:/]#  tftp --help
Usage: tftp [-4][-6][-v][-l][-m mode] [host [port]] [-c command]
```

详细参数说明如下：

- -g ：获取文件
- -p ：上传文件
- -l ： 本地文件
- -r ： 远程文件
- HOST: 远程主机 IP 地址

服务端可以选择 Linux 平台下的 tftp-hpa,也可以选择 windows 平台下的 tftpd 32/64(http://tftpd32.jounin.net/tftpd32_download.html)。下面以 ubuntu 平台为例 说明 tftp 服务端的配置。

1）安装 TFTP 服务端

```
$ sudo apt-get install tftp-hpa tftpd-hpa
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **配置 TFTP 服务**

创建 TFTP 服务器工作目录,并打开 TFTP 服务配置文件,如下:

```
$ mkdir -p /home/sur/tftpboot
$ chmod -R 777 /home/sur/tftpboot
$ sudo vi /etc/default/tftpd-hpa
```

修改或添加以下字段:

```
TFTP_DIRECTORY="/home/sur/tftpboot"
TFTP_OPTIONS="-l -c -s"
```

- **重启 TFTP 服务**

```
$ sudo service tftpd-hpa restart
```

配置好 tftp 服务端之后，将一个测试文件 zImage 放置到上面配置的<WORKDIR>/tftpboot/目录，就可以在开发板上使用 tftp 客户端进行文件的下载和上传了。

```
[root@myir:/]# tftp -g -r zImage -l zImage 192.168.0.179
```

上面的命令会把 tftp 服务端/tftpboot 目录下的 zImage 下载到开发板当前目录下。

```
[root@myir:/]# tftp -p -l config -r config_01 192.168.0.179
```

上面的命令会把开发板上当前目录下的 config 文件上传到 tftp 服务端之前配置的<WORKDIR>/tftpboot 目录下，并重新命名为 config_01。

5.5. DHCP

DHCP（动态主机配置协议）是一个局域网的网络协议。指的是由服务器控制一段 IP 地址范围，客户机登录服务器时就可以自动获得服务器分配的 IP 地址和子网掩码。DHCP 也包含服务器端和客户端两种角色，在 4.2 中配置 WiFi 的 AP 模式时，测试了 DHCP 服务端模式给连接的 WiFi 设备分配 IP 地址。这里再介绍一下使用 udhcpc 命令手动获取 IP 地址的方法，方便用户在调试网络时使用。

- **使用 udhcpc 命令配置 IP 地址**

```
[root@myir:/]# udhcpc -i eth0
udhcpc: started, v1.29.3
udhcpc: sending discover
udhcpc: sending select for 192.168.0.102
udhcpc: lease of 192.168.0.102 obtained, lease time 86400
```

```
deleting routers
route: SIOCADDRT: Network is unreachable
adding dns 192.168.0.1
adding dns 192.168.0.1
```

不管通过哪种方式，最终都可以为 eth0 配置好 IP 地址，以及网关，子网掩码，DNS 等信息如下:

```
[root@myir:/]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr C2:35:61:AF:A2:D8
          inet addr:192.168.0.102  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::c035:61ff:feaf:a2d8/64 Scope:Link
          inet6 addr: 2409:8a4d:c7d:8002:a39c:43fa:f405:7e2c/64 Scope:Global
          inet6 addr: 2409:8a4d:c7d:8002::4/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:63051 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63064 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:35047097 (33.4 MiB)  TX bytes:2916065 (2.7 MiB)
          Interrupt:37
```

5.6. Iptables

iptables 是一个用于 IPv4 包过滤和 NAT 的管理工具。它用于设置、维护和检查 Linux 内核中的 IP 包过滤规则表。可以定义几个不同的表。每个表包含许多内置链，也可以包含用户定义的链。每个链是一个规则列表，它可以匹配一组数据包。每个规则指定如何处理匹配的数据包。 使用 Linux 系统的开发板通常使用 iptables 工具来配置防火墙。iptables 就根据包过滤规则所定义的方法来处理各种数据包，如放行（accept）、拒绝（reject）和丢弃（drop）等。

下面使用 iptables 来测试拦截 icmp 包，禁止网络上的其它外部设备对其进行 ping 探测。具体命令使用参见：https://linux.die.net/man/8/iptables。

1）配置开发板 iptables

在开发板上使用 iptables 配置丢弃输入的 icmp 包，不回应其他主机的 ping 探测，命令如下:

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
[root@myir:/]#  iptables -A INPUT -p icmp --icmp-type 8 -j DROP
[root@myir:/]#  iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j DROP
```

2）ping 测试

在开发主机上 ping 开发板，并指定 deadline 为 10，结果如下：

```
C:\Users\40938>ping 192.168.0.102 -w 10

正在 Ping 192.168.0.102 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.0.102 的 Ping 统计信息:
    数据包: 已发送 = 10，已接收 = 0，丢失 = 10 (100% 丢失)，
```

以上结果表明，设置防火墙后开发主机无法 ping 通开发板。

- **删掉对应的防火墙规则**

```
[root@myir:/]#  iptables -F
[root@myir:/]#  iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

- **再次测试 ping 开发板**

```
C:\Users\40938>ping 192.168.0.102 -w 10

正在 Ping 192.168.0.102 具有 32 字节的数据:
来自 192.168.0.102 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.0.102 的回复: 字节=32 时间=1ms TTL=64
```

```
来自 192.168.0.102 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.0.102 的回复: 字节=32 时间=1ms TTL=64

192.168.0.102 的 Ping 统计信息:
    数据包: 已发送 = 10，已接收 = 10，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms，最长 = 1ms，平均 = 1ms
```

清除 iptables 规则之后，再次从开发主机 ping 开发板，就可以 ping 通了。上述示例只是一个简单的演示，实际上 iptables 配合各种规则可以实现非常强大的功能，这里就 不详细介绍了。

5.7. iperf3

iperf3 是在 IP 网络上主动测量最大可实现带宽的工具。 它支持调节测试时间、缓冲区大小和协议(IPV4 和 IPV6 下的 TCP、UDP、SCTP)等各种参数。iperf3 按角色可以分 为服务端模式或客户端模式，我们可以用它来测试和查看 TCP 模式下的网络带宽，TCP 窗口值，重传的概率等，也可以测试指定 UDP 带宽下丢包率，延迟和抖动情况。

我们在开发主机上打开 Windows PowerShell，带千兆网卡的主机作为 iperf3 的服务端，被测试 的开发板作为客户端分别测试开发板网卡 TCP 和 UDP 的性能。首先在主机上安装 iperf3，如下：

将服务器和开发板通过 CAT6 网线直连，并配置好各自的 IP 地址。例如我们设置服务器 ip 为 192.168.0.179，设开发板 IP 为 192.168.0.102，并使用 ping 命令测试确保它 们之间是连通的。

注意：尽量不要连接路由器或交换机，以免测试结果受到中间设备传输转发的影响。

1）测试 TCP 性能

- **服务端（192.168.0.179）**

服务器上 iperf3 使用-s 参数表示工作在服务端模式。

```
PS D:\iperf-3.1.3-win64> .\iperf3.exe -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- **客户端（192.168.0.102）**

开发板上运行的 iperf3 程序工作在客户端，TCP 模式，其中参数说明如下：

- -c 192.168.0.102 ：工作在客户端，连接服务端 192.168.0.179
- -i 2 ：测试结果报告时间间隔为 2 秒
- -t 10 ：总测试时长为 10 秒

```
[root@myir:/]# iperf3  -c 192.168.0.179 -i 2 -t 10
Connecting to host 192.168.0.179, port 5201
[  5] local 192.168.0.102 port 49692 connected to 192.168.0.179 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-2.00   sec   218 MBytes   914 Mbits/sec  242    625 KBytes
[  5]   2.00-4.00   sec   219 MBytes   918 Mbits/sec   14    551 KBytes
[  5]   4.00-6.00   sec   221 MBytes   927 Mbits/sec    0    609 KBytes
[  5]   6.00-8.00   sec   220 MBytes   923 Mbits/sec    0    631 KBytes
[  5]   8.00-10.00  sec   220 MBytes   923 Mbits/sec  369    554 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.07 GBytes   921 Mbits/sec  625             sender
[  5]   0.00-10.00  sec  1.07 GBytes   918 Mbits/sec                  receiver

iperf Done.
```

客户端经过 10 秒之后测试结束并显示上面的测试结果，表明 TCP 带宽为 921 Mbits 左右，没有重传，测试时 TCP 窗口值为 625KBytes. 同时服务端也显示测试结果如下，然后继续监听端口等待客户端连接：

```
PS D:\iperf-3.1.3-win64> .\iperf3.exe -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.0.102, port 35716
[  5] local 192.168.0.102 port 5201 connected to 192.168.0.179 port 35718
[ ID] Interval           Transfer     Bandwidth
[  5]  0.00-2.00  sec  220 MBytes  921 Mbits/sec
[  5]  2.00-4.00  sec  219 MBytes  917 Mbits/sec
```

```
[ 5] 4.00-6.00 sec 221 MBytes 925 Mbits/sec
[ 5] 6.00-8.00 sec 219 MBytes 920 Mbits/sec
[ 5] 8.00-10.00 sec 219 MBytes 917 Mbits/sec
[ 5] 10.00-10.02 sec 1.38 MBytes 530 Mbits/sec
 - - - - - - - - - - - - - - - - - - - -
[ ID] Interval Transfer Bandwidth
[ 5] 0.00-10.02 sec 1.07 GBytes 921 Mbits/sec 78 sender
[ 5] 0.00-10.02 sec 1.07 GBytes 919 Mbits/sec receiver
-----------------------------------------------------------
Server listening on 5201
 -----------------------------------------------------------
```

2）测试 UDP 性能

- **服务端（192.168.0.179 ）**

服务器上继续运行 iperf3 使用-s 参数表示工作在服务端模式。

```
PS D:\iperf-3.1.3-win64> .\iperf3.exe -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- **客户端（192.168.0.102）**

设备上 iperf3 工作在客户端，UDP 模式，其中参数说明如下：

- -u ： 工作在 UDP 模式
- -c 192.168.0.102 ： 工作在客户端，连接服务端 192.168.0.179
- -i 2 ： 测试结果报告时间间隔为 2 秒
- -t 10 ： 总测试时长为 10 秒
- -b 100M ： 设定 UDP 传输带宽为 100Mbps.

```
[root@myir:/]# iperf3 -c 192.168.0.179 -u -i 2 -t 10 -b 100M
Connecting to host 192.168.0.102, port 5201
[ 5] local 192.168.0.102 port 36915 connected to 192.168.0.179 port 5201
[ ID] Interval Transfer Bitrate Total Datagrams
[ 5] 0.00-2.00 sec 23.8 MBytes 100 Mbits/sec 17259
```

```
[  5]   2.00-4.00   sec  23.8 MBytes   100 Mbits/sec  17265
[  5]   4.00-6.00   sec  23.8 MBytes   100 Mbits/sec  17265
[  5]   6.00-8.00   sec  23.8 MBytes   100 Mbits/sec  17265
[  5]   8.00-10.00  sec  23.8 MBytes   100 Mbits/sec  17265
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total  Datagra
ms
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.000 ms   0/86319
(0%)  sender
[  5]   0.00-10.00  sec   119 MBytes  99.4 Mbits/sec  0.186 ms   466/86313
(0.54%)  receiver

iperf Done.
```

客户端经过 10 秒之后测试结束并显示上面的测试结果，表明 UDP 在指定带宽为 100 Mbps 时没有丢包。

同时服务端也显示测试结果如下，然后继续监听 5201 端口等待客户端连接：

```
$ $ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Server listening on 5201
Accepted connection from 192.168.0.102, port 49694
[  5] local 192.168.0.102 port 5201 connected to 192.168.0.179 port 40126
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total  Datag
rams
[  5]   0.00-2.00   sec  23.8 MBytes  99.8 Mbits/sec  0.230 ms   0/17240
(0%)
[  5]   2.00-4.00   sec  23.8 MBytes  99.6 Mbits/sec  0.161 ms   67/17267  (0.
39%)
[  5]   4.00-6.01   sec  23.6 MBytes  98.7 Mbits/sec  0.926 ms   124/17208
(0.72%)
```

```
[  5]   6.01-8.00   sec  23.7 MBytes  99.8 Mbits/sec  0.171 ms  136/17330
(0.78%)
[  5]   8.00-10.00  sec  23.6 MBytes  99.2 Mbits/sec  0.186 ms  139/17258
(0.81%)
[  5]  10.00-10.00  sec  14.1 KBytes  61.1 Mbits/sec  0.186 ms  0/10 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.186 ms  466/86313
(0.54%)
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

客户端修改-b 参数，继续增大指定的 UDP 带宽，发送端能达到的最大速率即是最大带宽，丢包率取决于服务器端 CPU 性能，网卡 buffer 大小，以下方法以发送固定大小的包来测试丢包率：

```
[root@myir:/]# iperf3 -u -c 192.168.0.179 -i 2 -t 10 -b 1000M
Connecting to host 192.168.0.102, port 5201
[  5] local 192.168.0.102 port 45388 connected to 192.168.0.179 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-2.00   sec  59.0 MBytes   248 Mbits/sec  42746
[  5]   2.00-4.00   sec  59.1 MBytes   248 Mbits/sec  42766
[  5]   4.00-6.00   sec  59.5 MBytes   250 Mbits/sec  43116
[  5]   6.00-8.00   sec  60.0 MBytes   252 Mbits/sec  43455
[  5]   8.00-10.00  sec  60.0 MBytes   252 Mbits/sec  43454
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-10.00  sec   298 MBytes   250 Mbits/sec  0.000 ms  0/215537
(0%)  sender
[  5]   0.00-10.00  sec   298 MBytes   250 Mbits/sec  0.044 ms  0/215536
(0%)  receiver
```

```
iperf  Done.
```

iperf3 在测试的过程中还有很多参数可以配置，用户可以根据实际应用需要进行有针对性的调整测试。比如可以增大-t 参数的值进行长时间压力测试，或者指定-P 参数进行多个连接并发的压力测试等。关于 iperf3 测试的更多信息请参考：https://iperf.fr/iperf-doc.php#3doc。

6. 图形系统

6.1. QT

QT 是一种跨平台 C++图形用户界面应用程序开发框架。它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器。Qt 是面向对象的框架，使用特殊的代码生成扩展以及一些宏，Qt 很容易扩展，并且允许真正地组件编程。

开发板会在出厂的时候烧写带有 Qt 运行时库的系统，并且提供了一个丰富的 HMI 演示系统，具体内容可以查看《**MEasy HMI2.x** 开发手册》。

1）获取 qt 的信息

首先查看当前系统支持的 QT 版本，如下：

```
[root@myir:/]# ls usr/local/
Qt_5.12.5
```

2）QT 运行环境介绍

在运行 Qt 应用程序时，可以根据不同的软硬件要求，对 Qt 的运行环境，如平台插件，显示参数，输入设备以及光标指针等进行适当的配置。

- **qtenv.sh 脚本**

在嵌入式 Linux 系统上，可以使用多个平台插件：EGLFS，LinuxFB，DirectFB 或 Wayland。但是，这些插件的可用性取决于实际硬件平台的特性以及 Qt 的配置方式，在 MYD-YT113X 平台使用的是 linuxfb 插件。

在 MYD-YT113X 平台中，我们通过 qtenv.sh 这个脚本加载运行 QT 程序所需的环境变量，脚本内容如下：

```
[root@myir:/]# cat /etc/qtenv.sh

export  QTDIR=/usr/local/Qt_5.12.5
if [ -d $QTDIR ];then

        export  QT_ROOT=$QTDIR
        export  PATH=$QTDIR/bin:$PATH
```

```
        export   LD_LIBRARY_PATH=$QTDIR/lib:/usr/lib/cedarx/:$LD_LIBRARY_PA
TH

        export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_ROOT/plugins
        export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
        export QT_QPA_FONTDIR=$QT_ROOT/fonts

        export QML_IMPORT_PATH=$QTDIR/qml
        export QML2_IMPORT_PATH=$QTDIR/qml

        TouchDevice="generic ft5x06 (79)"
        for InputDevices in /sys/class/input/input*
        do
                DeviceName=`cat $InputDevices/name`
                if [[ $DeviceName == $TouchDevice ]];then
                    TouchDeviceNum=${InputDevices##*input}
                    export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/i
nput/event$TouchDeviceNum
                    echo "add "/dev/input/event$TouchDeviceNum "to Qt Appli
cation."
                    break
                fi
        done
        if [ ! -n "$TouchDeviceNum" ]; then
         echo "Error:Input device $TouchDevice can not be found,plz check it!
"
        fi

        export QT_QPA_PLATFORM=linuxfb
        export set TSLIB_TSDEVICE=/dev/input/dev/input/event$TouchDeviceNu
m
        export set TSLIB_CONFFILE=/etc/ts.conf
        export set TSLIB_PLUGINDIR=/usr/lib/ts
```

```
        export set TSLIB_CALIBFILE=/etc/pointercal
        export set TSLIB_CONSOLEDEVICE=none
        export set TSLIB_FBDEVICE=/dev/fb0
        export QT_QPA_GENERIC_PLUGINS=evdevtouch,evdevmouse:/dev/input/
event6（触摸设备）
        #export QT_QPA_EGLFS_INTEGRATION=eglfs_mali
        #export QT_QPA_FB_HIDECURSOR=1
        #export QT_QPA_EGLFS_HIDECURSOR=1
        #export QT_QPA_EGLFS_ROTATION=90

        #export QWS_MOUSE_PROTO=Intellimouse:/dev/input/event6

        #export DBUS_SESSION_BUS_ADDRESS=`cat /tmp/dbusaddr`
        mkdir -p /dev/shm
        #ulimit -c unlimited
        mxapp2 &
        echo "find qt5 installed done"
fi
```

- **显示参数配置**

QT 应用程序可以通过 QScreen 类或者 QDesktopWidget 获取屏幕显示相关的参数，从而编写跟屏幕匹配的应用。通过 QScreen 或 QDesktopWidget 获取屏幕分辨率，颜色深度一般是没有问题的，但由于显示驱动的原因有时候获取的物理尺寸就不一定是正确的了。此时可以通过配置和调整下面的参数，使实际界面上显示的元素适合显示屏幕的大小。

一般情况下，使用默认的配置即可，但如果确实有显示元素和实际屏幕不匹配的情况，则可以根据上面的描述对相关参数做适当的调整。

- **输入外设配置**

当嵌入式 Linux 设备上没有窗口系统（例如 XWindow 或 Weston)时，鼠标，按键或者触目设备通过直接读取 evdev 或者使用其它中间库获取输入设备信息，如 libinput 或 tslib。eglfs 和 linuxfb 平台插件包含这两种输入方式。关于 Qt5 输入设备配置如下：

linuxfb 平台插件默认使用的是 EvdevTouch 输入处理程序，这种方式常用于处理电容触摸，电容触摸驱动上报的事件坐标与实际屏幕区域坐标完全对应的话，就不需要做额外

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

的处理，如果出现反向，则可以通过环境变量进行一些调整，EvdevTouch 输入处理程序支持以下一些额外参数。

从以上代码可以看出，linuxfb 平台插件默认使用的是 EvdevTouch 输入处理程序，这种方式常用于处理电容触摸，电容触摸驱动上报的事件坐标与实际屏幕区域坐标完全对应的话，就不需要做额外的处理，如果出现反向，则可以通过环境变量进行一些调整，EvdevTouch 输入处理程序支持以下一些额外参数：

表 6-1. QT Linuxfb 插件 EVDEV 触摸处理程序相关的环境变量参数

参数	描述
/dev/input/...	指定输入设备的名称。如果未指定，Qt 将通过 libudev 或遍历可用节点来寻找合适的设备。
rotate	在某些触摸屏上，必须通过将坐标设置 rotate 为 90、180 或 270 来旋转坐标。
invertx/inverty	指定用于在输入事件中反转 X 或 Y 坐标的参数。

例如，如果 QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS 在启动应用程序之前将以下值传递给平台插件，则明确指定触摸设备为/dev/input/event6，其坐标翻转 180 度。当实际屏幕和触摸屏的方向不匹配时，这很有用。

```
export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/event6:rotate=180
```

如果要启用 tslib 支持，需要将 QT_QPA_EGLFS_TSLIB（for eglfs）或 QT_QPA_FB_TSLIB（for linuxfb）环境变量设置为 1。关于 tslib 的具体使用方法参考 https://github.com/libts/tslib/blob/master/README.md

注意：tslib 输入处理程序常用于电阻触摸，会生成鼠标事件并仅支持单点触摸，初始使用还需要进行屏幕校准。

3）启动 Qt 程序

当我们需要在 MYD-YT113X 平台上运行自己的 QT 程序时，需要修改 qtenv 脚本文件，将可执行的 QT 程序路径添加到脚本文件中，这样运行脚本之后即可配置正确环境变量.如需运行 QT 程序 qt_test,需在 qtenv.sh 脚本末尾做如下修改。

```
export QWS_MOUSE_PROTO=
        export DBUS_SESSION_BUS_ADDRESS=`cat /tmp/dbusaddr`
        mkdir -p /dev/shm
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
        ulimit -c unlimited
        #debug Launcher &
        #mxapp2 &
        /etc/qt_test &
        echo "find qt5 installed done"
fi
```

这里我们注释掉了之前默认执行的 QT 程序 mxapp2，将自己的 QT 程序 qt_test 路径加入到脚本中。

由于之前 mxapp2 默认已经启动，此时如果需要运行自己的 Qt 应用程序，需要先终止 mxapp2 再启动其他应用。

可以通过 kill 方式对进程直接退出。

```
[root@myir:/]# killall mxapp2
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

7. 多媒体应用

7.1. 视频播放

1) xplayerdemo 工具

全志系统自带播放器 xplayerdemo 可实现视频解码播放功能。

```
[root@myir:/mnt/u/video]# xplayerdemo
WARNING: awplayer <log_set_level:30>: Set log level to 3
ERROR   : awplayer <ReadPluginEntry:194>: read plugin entry adecoder-15 fail!
ERROR   : awplayer <ReadPluginEntry:194>: read plugin entry vdecoder-8 fail!
INFO    : cedarc <CedarPluginVDInit:79>: register h264 decoder success!
INFO    : cedarc <CedarPluginVDInit:84>: register mjpeg decoder success!
INFO    : cedarc <CedarPluginVDInit:86>: register mpeg2 decoder success!
WARNING: awplayer <DlOpenPlugin:112>: Invalid plugin,function CedarPluginV
DInit not found.
INFO    : cedarc <CedarPluginVDInit:92>: register mpeg4dx decoder success!
INFO    : cedarc <CedarPluginVDInit:79>: register mpeg4H263 decoder succes
s!
INFO    : cedarc <CedarPluginVDInit:90>: register mpeg4Normal decoder succ
ess!
INFO    : cedarc <CedarPluginVDInit:74>: register vc1 decoder success!
ERROR   : awplayer <ReadPluginEntry:194>: read plugin entry plugin-0 fail!
DEBUG   : awplayer <AwStreamInit:99>: aw stream init...

*************************************************************************
* This program implements a simple player,
* you can type commands to control the player.
* To show what commands supported, type 'help'.
* Inplemented by Allwinner ALD-AL3 department.
*************************************************************************
DEBUG   : awplayer <XPlayerCreate:216>: XPlayerCreate.
DEBUG   : awplayer <LogVersionInfo:34>:
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> CedarX <<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<
tag     : CedarX-2.8.0
branch: master
commit: 967535b8ff6a073cb4f38e85a4ae5fa6008014d8
date    : Mon, 15 May 2017 01:30:22 +0000 (09:30 +0800)
author:
----------------------------------------------------------------------

DEBUG   : awplayer <LayerCreate_DE:832>: LayerCreate.
INFO    : awplayer <LayerCreate_DE:857>: screen_id = 0,output_type = 1
INFO    : awplayer <LayerCreate_DE:860>: the output type: 0
DEBUG   : awplayer <LayerCreate_DE:871>: screen:w 1024, screen:h 600
DEBUG   : ionAlloc <CdxIonOpen:30>: begin ion_alloc_open

DEBUG   : ionAlloc <__GetIonMemOpsS:856>: *** get __GetIonMemOpsS ***
INFO    : cedarc <VeInitialize:1307>: *** ic_version = 0x1301000010210,
DEBUG   : cedarc <VeInitialize:1347>: *** nPhyOffset = 0x0
DEBUG   : ionAlloc <ion_alloc_open2:177>: pid: 2454, g_context = 0x5f6ba8

DEBUG   : ionAlloc <ion_alloc_open2:204>: ion alloc open ok
DEBUG   : TinaSoundControl <TinaSoundDeviceInit:146>: TinaSoundDeviceInit()
DEBUG   : awplayer <XPlayerSetVideoSurfaceTexture:597>: setVideoSurfaceText
ure, surface = 0x5f45c8
DEBUG   : awplayer <XPlayerThread:1957>: process message XPLAYER_COMMA
ND_SET_SURFACE.
DEBUG   : awplayer <XPlayerThread:2018>: ==== process message XPLAYER_C
OMMAND_SET_SUBCTRL.
DEBUG   : awplayer <XPlayerSetDeinterlace:698>: set deinterlace
DEBUG   : awplayer <XPlayerThread:2033>: ==== process message XPLAYER_C
OMMAND_SET_SUBCTRL.

demoPlayer#
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

查看播放器支持操作命令：

```
demoPlayer# help
*********************************************************************************
* This is a simple media player, when it is started, you can input commands to tell
* what you want it to do.
* Usage:
*    # ./demoPlayer
*    # set url: http://www.allwinner.com/ald/al3/testvideo1.mp4
*    # show media info
*    # play
*    # pause
*    # stop
* Command and it param is seperated by a colon, param is optional, as below:
*       Command[: Param]
* here are the commands supported:
*     help:
*                 show this help message.
*     quit:
*                 quit this program.
*     set url:
*                 set url of the media, for example, set url: ~/testfile.mkv.
*     play:
*                 start playback.
*     pause:
*                 pause the playback.
*     stop:
*                 stop the playback.
*     set speed:
*                 stop the playback.
*     seek to:
```


```
*                seek to specific position to play, position is in unit of second,
 ex, seek to: 100.
*     show media info:
*                show media information of the media file.
*     show duration:
*                show duration of the media file.
*     show position:
*                show current play position,in unit of second.
*     switch audio:
*                switch audio to a track, for example, switch audio: 2, track is
start counting from 0.
*
*********************************************************
```

2）播放视频

我们以播放挂载 u 盘中 h264_output.mp4 为例操作：

```
demoPlayer# set url:/mnt/usb/sda1/h264_output.mp4
DEBUG   : awplayer <XPlayerSetDataSourceUrl:456>: setDataSource(url), url='/m
nt/u/video/4KCanada_h264.mp4'
INFO    : awplayer <XPlayerThread:1707>: process message XPLAYER_COMMA
ND_SET_SOURCE.
DEBUG   : awplayer <XPlayerPrepare:741>: prepare
DEBUG   : awplayer <XPlayerThread:1960>: process message XPLAYER_COMMA
ND_PREPARE. mPriData->mStatus: 1
DEBUG   : demuxComponent <DemuxThread:1783>: process message DEMUX_
COMMAND_PREPARE.
DEBUG   : demuxComponent <DemuxThread:1850>: === prepare msg
DEBUG   : awplayer <CdxParserPrepare:757>: source uri 'file:///mnt/u/video/4K
Canada_h264.mp4'
......
INFO    : awplayer <XPlayerThread:1996>: xxxxxxxxxx video size: width = 1920,
 height = 960
++++ video width: 1920, height: 960
```

```
DEBUG   : awplayer <CallbackForAwPlayer:440>: info : preared
info: prepare ok.
preparing...
```

这里表示播放准备完成，执行命令 play 即可开始播放。

```
demoPlayer# play
DEBUG   : awplayer <XPlayerStart:771>: start
DEBUG   : awplayer <XPlayerThread:2140>: process message XPLAYER_COMMA
ND_START.
DEBUG   : awplayer <PlayerStart:730>: player start
DEBUG   : awplayer <BaseCompPostAndWait:61>: video decoder receive cmd:
start
debug   : cedarc <SbmFrameReset:613>: ** wait for reset sem
debug   : cedarc <ProcessThread:1653>: *** post reset sem
debug   : cedarc <SbmFrameReset:615>: ** wait for reset sem ok
debug   : cedarc <SbmFrameReset:620>: SbmFrameReset finish
DEBUG   : awplayer <BaseCompPostAndWait:61>: audio decoder receive cmd:
start
debug   : cedarc <H264ProcessExtraData2:543>: H264ProcessNaluUnit, bNeedFi
ndSPS = 0, bNeedFindPPS = 0
(Allwinner Audio Middle Layer),line(958) : Create Decoder!!=====
DEBUG   : audioDecItf <handleStart:1064>: Create libadecoder success...
(Allwinner Audio Middle Layer),line(592) : AudioDec_Installaudiolib ok
(Allwinner Audio Middle Layer),line(595) : audio decoder init start ...
(AllwinnerAlibs),line(626) : libaw_aacdec.so open, use dlopen!
(AllwinnerAlibs),line(660) : Khan----Loading 'libaw_aacdec.so' success!
......
```

7.2. Audio

本章节是测试播放音频。

1）调试工具

- **Tinyplay 播放音频工具**

将音频文件放入到 u 盘中，然后将喇叭接入 j3 接口。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
Usage: tinyplay file.wav [-D card] [-d device] [-p period_size] [-n n_periods] [-T capture time]
[root@myir:/mnt/usb/sda1]# tinyplay 01+Singalongsong.wav
playing '01+Singalongsong.wav': 2 ch, 48000 hz, 16 bit
```

- **alsamixer**

音量调节工具，执行 alsamixer 命令，可调节左右声道及音量等参数。

```
[root@myir:/]# alsamixer -a
alsamixer: option requires an argument -- 'a'
Usage: alsamixer [options]
Useful options:
  -h, --help                this help
  -c, --card=NUMBER         sound card number or id
  -D, --device=NAME         mixer device name
  -V, --view=MODE           starting view mode: playback/capture/all
Debugging options:
  -g, --no-color            toggle using of colors
  -a, --abstraction=NAME    mixer abstraction level: none/basic
```

如下图所以，使用方向“←”“→”键控制光标选择修改项，使用“↑”“↓”调节参数大小。

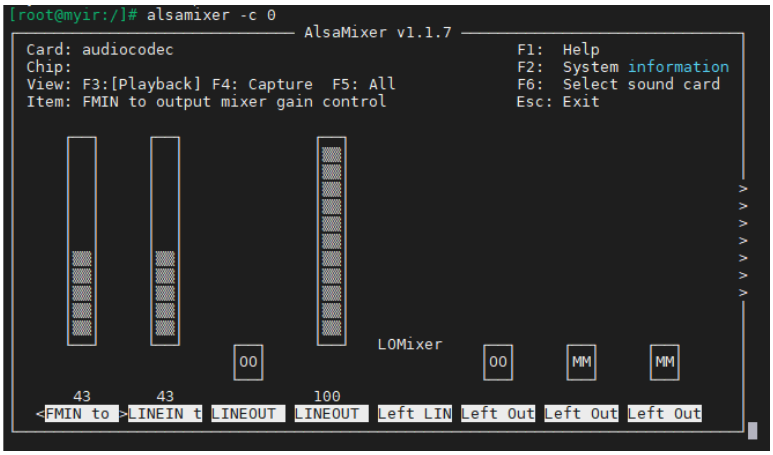

图 6-1.播放器参数调节 UI

8. 系统工具

默认映像中包含了一些常用的系统工具，便于用户在系统调试或实际部署的产品中查看和管理系统的各种资源，也可以在 SHELL 脚本或其他应用程序中调用。这些工具可能不 完全满足用户的系统定制需求，此时系统开发人员需要根据实际情况做出适当的调整。

8.1. 压缩解压工具

本节主要测试系统的解压缩工具。压缩是可以把多个文件压缩成一个压缩包可以把多个文件压缩成一个压缩包，方便进行文件的传输。而解压可以把经过压缩的压缩文件还原成原始大小方便使用。本节将在文件系统以 tar、gzip、gunzip 等工具为例进行说明。

1) tar 工具

现在我们在 Linux 中使用的 tar 工具，它不仅可以对文件打包，还可以对其进行压缩，查看，添加以及解压等一系列操作。这里是将打包操作。输入以下命令查看 tar 语法格式：

```
[root@myir:/]# tar --help
BusyBox v1.29.3 (2022-01-28 15:52:25 CST) multi-call binary.

Usage: tar c|x|t [-hvokO] [-f TARFILE] [-C DIR] [-T FILE] [-X FILE] [--exclude PA
TTERN]... [FILE]...
Create, extract, or list files from a tar file

        c         Create
        x         Extract
        t         List
        -f FILE   Name of TARFILE ('-' for stdin/out)
        -C DIR    Change to DIR before operation
        -v        Verbose
        -O        Extract to stdout
        -o        Don't restore user:group
        -k        Don't replace existing files
        -h        Follow symlinks
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
        -T FILE File with names to include
        -X FILE File with glob patterns to exclude
        --exclude PATTERN       Glob pattern to exclude
```

- **使用 tar 压缩**

新建 test.txt 文件，并输入以下命令将文件打包成.gz 格式：

```
[root@myir:/]# tar -cf test.tar.gz test.txt
[root@myir:/]# ls
test.tar.gz
```

- **使用 tar 解压**

把打包成 tar.gz 格式文件解压

```
[root@myir:/]# tar -xvf test.tar.gz
test.txt
[root@myir:/]# ls
test.txt
```

2）gzip 压缩工具

- **语法格式**

gzip 是在 Linux 系统中经常使用的一个对文件进行压缩和解压缩的命令，既方便又好用。在开发板终端输入以下命令查看 gzip 语法：

```
[root@myir:/]# gzip --hple
gzip: unrecognized option '--hple'
BusyBox v1.29.3 (2022-01-28 15:52:25 CST) multi-call binary.
Usage: gzip [-cfkdt] [FILE]...
Compress FILEs (or stdin)

        -d      Decompress
        -t      Test file integrity
        -c      Write to stdout
        -f      Force
        -k      Keep input files
```

- **用 gzip 把文件压缩**

```
[root@myir:/]# gzip test.txt
[root@myir:/]# ls
test.tar.gz
```

- **用 gunzip 把文件解压**

```
[root@myir:/]# gunzip test.txt.gz
[root@myir:/]# ls
test.txt
```

8.2. 文件系统工具

主要测试系统的文件系统工具，本节将介绍几种常见的文件系统管理工具。系统自带的文件系统工具 mount。

1）mount 挂载工具

mount 是 Linux 下的一个命令，它可以将分区挂接到 Linux 的一个文件夹下，从而将分区和该目录联系起来，因此我们只要访问这个文件夹，就相当于访问该分区了，其应用语法格式如下：

```
[root@myir:/]# mount -h
Usage:
 mount [-lhV]
 mount -a [options]
 mount [options] [--source] <source> | [--target] <directory>
 mount [options] <source> <directory>
 mount <operation> <mountpoint> [<target>]

Mount a filesystem.
```

- **挂载 U 盘**

```
[root@myir:/]# mount /dev/sda1 /mnt/
```

8.3. 磁盘管理工具

主要测试系统的磁盘管理工具，本节将介绍几种常见的磁盘管理工具。系统自带的磁盘管理工具 fdisk、dd、mkfs、du、df、cfdisk、fsck 。通过这些命令可以监控平时的磁盘使用情况。

1）fdisk 磁盘分区工具

fdisk 磁盘分区工具在 DOS、Windows 和 Linux 中都有相应的应用程序。在 Linux 系统中，fdisk 是基于菜单的命令。用 fdisk 对硬盘进行分区，可以在 fdisk 命令后面直接加上要分区的硬盘作为参数，其应用语法格式如下：

```
[root@myir:/]# fdisk -h
BusyBox v1.29.3 (2022-01-28 15:52:25 CST) multi-call binary.
Usage: fdisk [-ul] [-C CYLINDERS] [-H HEADS] [-S SECTORS] [-b SSZ] DISK
Change partition table

        -u              Start and End are in sectors (instead of cylinders)
        -l              Show partition table for each DISK, then exit
        -b 2048         (for certain MO disks) use 2048-byte sectors
        -C CYLINDERS    Set number of cylinders/heads/sectors
        -H HEADS        Typically 255
        -S SECTORS      Typically 63
```

对 eMMC 进行分区：

```
[root@myir:/]# fdisk /dev/mmcblk0
Found valid GPT with protective MBR; using GPT

Command (m for help): m
Command Action
o       create a new empty DOS partition table
p       print the partition table
q       quit without saving changes
s       create a new empty Sun disklabel

Command (m for help):
```

2）dd 拷贝命令

dd 命令用于将指定的输入文件拷贝到指定的输出文件上。并且在复制过程中可以进行格式转换。dd 命令与 cp 命令的区别在于：dd 命令可以在没有创建文件系统的软盘上进行，拷贝到软盘的数据实际上是镜像文件。类似于 DOS 中的 diskcopy 命令的作用。dd 命

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

令的格式为： dd [<if=输入文件名/设备名>] [<of=输出文件名/设备名>] [bs=块字节大小] [count = 块数]。

创建一个大小为 2M 的文件。

```
[root@myir:/]# time dd  if=/dev/zero of=ffmpeg1 bs=2M count=1 conv=fsync
1+0 records in
1+0 records out

real    0m0.094s
user    0m0.000s
sys     0m0.027s
```

3）du 磁盘用量统计工具

du 命令用于显示磁盘空间的使用情况。该命令逐级显示指定目录的每一级子目录占用文件系统数据块的情况。du 一般使用语法如下：

```
[root@myir:/]# du --help
BusyBox v1.29.3 (2022-01-28 15:52:25 CST) multi-call binary.
Usage: du [-aHLdclsxhmk] [FILE]...
Summarize disk space used for each FILE and/or directory

        -a      Show file sizes too
        -L      Follow all symlinks
        -H      Follow symlinks on command line
        -d N    Limit output to directories (and files with -a) of depth < N
        -c      Show grand total
        -l      Count sizes many times if hard linked
        -s      Display only a total for each argument
        -x      Skip directories on different filesystems
        -h      Sizes in human readable format (e.g., 1K 243M 2G)
        -m      Sizes in megabytes
        -k      Sizes in kilobytes (default)
```

部分参数说明：

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- -a:显示所有目录或文件的大小
- -h:以 K,M,G 为单位，提高信息可读性
- -k:以 KB 为单位输出
- -m:以 MB 为单位输出

统计 dd 命令生成的文件大小：

```
[root@myir:/]# du ffmpeg1
2048    ffmpeg1
[root@myir:/]# du -h ffmpeg1
2.0M    ffmpeg1
```

4) df 磁盘统计工具

用于显示目前在 Linux 系统上的文件系统的磁盘使用情况统计，一般用法如下：

```
[root@myir:/]# df -help
df: invalid option -- 'e'
BusyBox v1.29.3 (2022-01-28 15:52:25 CST) multi-call binary.
Usage: df [-PkmhT] [FILESYSTEM]...
Print filesystem usage statistics

        -P      POSIX output format
        -k      1024-byte blocks (default)
        -m       1M-byte blocks
        -h      Human readable (e.g. 1K 243M 2G)
        -T      Print filesystem type
```

部分参数说明：

- -h：可以根据所使用大小使用适当的单位显示
- -i:查看分区下 inode 的数量和 inode 的使用情况
- -T:打印出文件系统类型

查看分区下 inode 的数量和 inode 的使用情况，使用如下命令：

```
[root@myir:/]# df -h
Filesystem                Size      Used Available Use% Mo
unted on
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
/dev/mmcblk0p4      1.9G      805.6M    1.1G      41%       /
tmpfs                         490.1M    64.0K     490.0M    0%
  /tmp
tmpfs                         490.1M    32.0K     490.1M    0%
  /run
devtmpfs                480.7M    0         480.7M    0%
  /dev
/dev/mmcblk0p8      2.9G      9.0M      2.7G      0%        /
media
tmpfs                         490.1M    0         490.1M    0%
    /dev/shm
```

inode 是我们在格式化的时候系统给我们划分好的 ,inode 与磁盘分区大小有关。当我们的 inode 使用已经达到百分百时，即使我们的磁盘空间还是有剩余，我们也是写不了数据到磁盘的。

8.4. 进程管理工具

进程也是操作系统中的一个重要概念，它是一个程序的一次执行过程，程序是进程的一种静态描述，系统中运行的每一个程序都是在它的进程中运行的。Linux 系统中所有的进程是相互联系的，除了初始化进程外，所有进程都有一个父进程。新的进程不是被创建，而是被复制，或是从以前的进程复制而来。Linux 中所有的进程都是由一个进程号为 1 的 init 进程衍生而来的。Linux 系统包括 3 种不同类型的进程，每种进程都有自己的特点和属性：

- 交互进程：由一个 Shell 启动的进程，既可以在前台运行，又可以在后台运行。
- 批处理进程：这种进程和终端没有联系，是一个进程序列。这种进程被提交到等待队列顺序执行的进程。
- 监控进程(守护进程)：守护进程总是活跃的，一般是在后台运行，守护进程一般是由系统在开始时通过脚本自动激活启动或 root 启动。

对于 linux 系统来说，进程的管理是重要的一环，对于进程的管理通常是通过进程管理工具实现的，Linux 系统中比较常用的进程管理命令有以下几种：ps ，top， vmstat kill。

1）ps 显示当前进程工具

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- **语法格式**

显示当前系统进程的运行情况，一般语法如下：

```
[root@myir:/]# ps --help
BusyBox v1.29.3 (2022-01-28 15:52:25 CST) multi-call binary.

Usage: ps [-o COL1,COL2=HEADER]

Show list of processes

        -o COL1,COL2=HEADER     Select columns for display
```

部分参数组合说明：

- -u：以用户为中心组织进程状态信息显示；
- -a：与终端无关的进程；
- -x：与终端有关的进程；　（线程，就是轻量级进程；）
- 通常上面命令组合使用：aux。
- --e：显示所有进程；相当于 ax；
- -f：显示完整格式程序信息；
- 通常以上命令组合使用：ef
- -H：以进程层级显示进程数的
- -F：显示更多的程序信息

通常组合使用命令：eHF。

- **显示所有进程的信息情况**

```
[root@myir:/]# ps
PID   USER     COMMAND
    1 root     init
    2 root     [kthreadd]
    4 root     [kworker/0:0H]
    6 root     [ksoftirqd/0]
    7 root     [rcu_preempt]
    8 root     [rcu_sched]
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
  9 root      [rcu_bh]
 10 root      [migration/0]
 11 root      [lru-add-drain]
 12 root      [cpuhp/0]
 13 root      [cpuhp/1]
 14 root      [migration/1]
 15 root      [ksoftirqd/1]
 17 root      [kworker/1:0H]
 18 root      [cpuhp/2]
 19 root      [migration/2]
 20 root      [ksoftirqd/2]
 22 root      [kworker/2:0H]
 23 root      [cpuhp/3]
 24 root      [migration/3]
 ......
```

2) top 显示 linux 进程

- **语法格式**

top 命令将相当多的系统整体性能信息放在一个屏幕上。显示内容还能以交互的方式进行改变。动态的持续监控进程的运行状态，top 语法一般如下：

```
[root@myir:/]# top --help
BusyBox v1.29.3 (2022-01-28 15:52:25 CST) multi-call binary.

Usage: top [-b] [-nCOUNT] [-dSECONDS]

Provide a view of process activity in real time.
Read the status of all processes from /proc each SECONDS
and display a screenful of them.
Keys:
        N/M/P/T: sort by pid/mem/cpu/time
        R: reverse sort
        Q,^C: exit
```

```
Options:
        -b      Batch mode
        -n N    Exit after N iterations
        -d N    Delay between updates
```

- **动态查看系统进程**

```
[root@myir:/]# top
Mem: 234716K used, 769024K free, 88K shrd, 10208K buff, 62924K cached
CPU:   0% usr   2% sys   0% nic  97% idle   0% io   0% irq   0% sirq
Load average: 0.00 0.00 0.00 1/132 32743
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
32736  1903 root     R      2580   0%   2% top
 1705     1 root     S      294m  30%   0% /etc/video2lcd
 1532     1 root     S      156m  16%   0% adbd
 1903     1 root     S      3624   0%   0% -/bin/sh
 1345     1 root     S      2716   0%   0% dbus-daemon --system
    1     0 root     S      2580   0%   0% init
 1316     1 root     S      2580   0%   0% /sbin/syslogd -n
 1321     1 root     S      2580   0%   0% /sbin/klogd -n
 1417     1 root     S      2328   0%   0% /usr/sbin/dropbear -R
 1398     1 root     S      2192   0%   0% /sbin/dhcpcd -f /etc/dhcpcd.conf
 1455     1 root     S      2160   0%   0% /usr/sbin/tftpd -c -l -s /var/lib/tf
tpboot
  589     2 root     SW        0   0%   0% [vsync proc 0]
    7     2 root     SW        0   0%   0% [rcu_preempt]
 1226     2 root     SW        0   0%   0% [cec thread]
 6256     2 root     SW        0   0%   0% [kworker/u8:0]
28128     2 root     SW        0   0%   0% [kworker/u8:1]
  921     2 root     SW        0   0%   0% [kworker/0:1]
 1225     2 root     SW        0   0%   0% [hdmi proc]
 1232     2 root     SW        0   0%   0% [tve detect]
  551     2 root     SW        0   0%   0% [kworker/2:1]
 1504     2 root     SW        0   0%   0% [mali-simple-pow]
```

```
 8359     2 root     SW       0   0%   0% [kworker/2:0]
  548     2 root     SW       0   0%   0% [kworker/1:1]
  554     2 root     SW       0   0%   0% [kworker/3:1]
```

3）kill 进程终止工具

- **语法格式**

发送指定的信号到相应进程。不指定型号将发送 SIGTERM（15）终止指定进程。如果任无法终止该程序可用“-KILL”参数，其发送的信号为 SIGKILL(9) ，将强制结束进程，使用 ps 命令或者 jobs 命令可以查看进程号。root 用户将影响用户的进程，非 root 用户只能影响自己的进程。kill 命令一般语法如下：

```
[root@myir:/]# kill --help
kill: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]
    Send a signal to a job.

    Send the processes identified by PID or JOBSPEC the signal named by
    SIGSPEC or SIGNUM.  If neither SIGSPEC nor SIGNUM is present, then
    SIGTERM is assumed.

    Options:
      -s sig    SIG is a signal name
      -n sig    SIG is a signal number
      -l        list the signal names; if arguments follow `-l' they are
                assumed to be signal numbers for which names should be lis
ted
      -L        synonym for -l

    Kill is a shell builtin for two reasons: it allows job IDs to be used
    instead of process IDs, and allows processes to be killed if the limit
    on processes that you can create is reached.
```

部分参数组合说明：

- -s：指定发送的信号

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

- -p：模拟发送信号
- -l：指定信号的名称列表
- pid：要中止进程的 ID 号
- Signal：表示信号

首先使用 ps -ef 与管道命令确定要杀死进程的 PID，

```
[root@myir:/]# ps -ef | grep mxapp2
 1705 root       /etc/mxapp2
 6254 root       grep /etc/mxapp2
```

然后输入以下命令终止进程：

```
[root@myir:/]# kill 1705
```

killall 命令终止同一进程组内的所有进程，允许指定要终止的进程的名称而非 PID 进程号：

```
[root@myir:/]# killall mxapp2
```

9. 开发支持

本章主要介绍针对当前 SDK 进行二次开发的一些基本信息，当前 SDK 提供三种配置的参考镜像，一种是 t113_linux_myir_emmc_core 和 t113_linux_myir_nand，主要针对无 GUI 的应用；另外一种是 t113_linux_myir_emmc_full，在 t113_linux_myir_emmc_core 的基础上增加一些需要 GUI 的应用。关于这三种镜像的信息请参考《MYD-YT113X_SDK 发布说明》。

9.1. 开发语言

1) SHELL

Shell 是一个用 C 语言编写的程序，它是用户使用 Linux 的桥梁。Shell 既是一种命令语言，又是一种程序设计语言。常见的 Linux 的 Shell 种类众多，常见的有：

Bourne Shell（/usr/bin/sh 或/bin/sh）

- Bourne Again Shell（/bin/bash）
- C Shell（/usr/bin/csh）
- K Shell（/usr/bin/ksh）
- Shell for Root（/sbin/sh）

MYD-YT113X 支持 bourne shell 和 Bourne Again Shell 2 种：

```
[root@myir:/]# echo "echo 'myir test'" > shell_demo.sh
[root@myir:/]# sh shell_demo.sh
myir test
[root@myir:/]# bash shell_demo.sh
myir test
```

2) C/C++

C/C++是 Linux 平台下进行底层应用开发最为常用的编程语言，也是仅次于汇编的最为高效的语言。使用 C/C++进行开发通常采用的是交叉开发的方式，即在开发主机端进行开发，编译生成目标机器上运行的二进制执行文件，然后部署到目标机器上运行。

采用这种方式，首先需要安装基于 SDK，安装步骤请参《MYD-YT113X_Linux 软件开发指南》，安装完成后需要配置一下 SDK 环境。

首先添加编译工具链到环境变量下：

```
sur@myir:~/T113$ export PATH=$PATH: /home/sur/opt/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/bin
sur@myir:~/T113$ export CROSS_COMPILE=/home/sur/opt/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-gcc

sur@myir:~/T113$ arm-linux-gnueabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/home/sur/opt/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/bin/../libexec/gcc/arm-linux-gnueabi/5.3.1/lto-wrapper
Target: arm-linux-gnueabi
Configured with: /home/tcwg-buildslave/workspace/tcwg-make-release/label/docker-trusty-amd64-tcwg/target/arm-linux-gnueabi/snapshots/gcc-linaro-5.3-2016.05/configure SHELL=/bin/bash --with-mpc=/home/tcwg-buildslave/workspace/tcwg-make-release/label/docker-trusty-amd64-tcwg/target/arm-linux-gnueabi/_build/builds/destdir/x86_64-unknown-linux-gnu --with-mpfr=/home/tcwg-buildslave/workspace/tcwg-make-release/label/docker-trusty-amd64-tcwg/target/arm-linux-gnueabi/_build/builds/destdir/x86_64-unknown-linux-gnu --with-gmp=/home/tcwg-buildslave/workspace/tcwg-make-release/label/docker-trusty-amd64-tcwg/target/arm-linux-gnueabi/_build/builds/destdir/x86_64-unknown-linux-gnu --with-gnu-as --with-gnu-ld --disable-libstdcxx-pch --disable-libmudflap --with-cloog=no --with-ppl=no --with-isl=no --disable-nls --enable-c99 --with-tune=cortex-a9 --with-arch=armv7-a --with-fpu=vfpv3-d16 --with-float=softfp --with-mode=thumb --disable-multilib --enable-multiarch --with-build-sysroot=/home/tcwg-buildslave/workspace/tcwg-make-release/label/docker-trusty-amd64-tcwg/target/arm-linux-gnueabi/_build/sysroots/arm-linux-gnueabi --enable-lto --enable-linker-build-id --enable-long-long --enable-shared --with-sysroot=/home/tcwg-buildslave/workspace/tcwg-make-release/label/docker-trusty-amd64-tcwg/target/arm-linux-gnueabi/_build/builds/destdir/x86_64-unknown-linux-gnu/arm-linux-gnueabi/libc --enable-languages=c,c++,fortran,lto --enable-checking=release --disable-bootstrap --build=x86_64-unknown-linux-gnu --host=x86_64-unknown-linux-gnu --target=arm-linux-gnueabi --prefix=/home/tcwg-buildslave/workspa
```

```
ce/tcwg-make-release/label/docker-trusty-amd64-tcwg/target/arm-linux-gnueabi
/_build/builds/destdir/x86_64-unknown-linux-gnu
Thread model: posix
gcc version 5.3.1 20160412 (Linaro GCC 5.3-2016.05)
```

本节通过编写一个简单的 Hello World 实例来演示应用程序的开发，以下为在开发主机端编写的演示程序 hello.c：

```
#include<stdio.h>
int main(int argc,char *argv[])
{
    printf("hello world!\n");
     return 0;
}
```

C++编写的演示程序 hello-CXX.cpp

```
//file: hello-CXX.cpp
#include <iostream>
using namespace std;
int main(int argc,char *argv[])
{
        cout << "hello world!";
        return 0;
 }
```

接着编译应用程序，hello.c 文件使用上文导入的编译工具链 arm-linux-gnueabi-gcc，C++文件 hello-CXX.cpp 文件使用 arm-linux-gnueabi-cpp 工具链。

```
$arm-linux-gnueabi-gcc hello.c -o hello
or
$ arm-linux-gnueabi-cpp hello-CXX.cpp -o hello-CXX
.............
```

然后通过 scp 命令把生成的执行文件拷贝到目标机器上执行，结果如下：

```
[root@myir:/]# ./hello
hello world!
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
or
[root@myir:/]# ./hello-CXX
hello world!
```

更复杂的示例和开发方式请参考《MYD-YT113X_Linux 软件开发指南》应用移植部分的说明。

3) Python

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。Python 由 Guido van Rossum 于 1989 年底发明，第一个公开发行版发行于 1991 年。像 Perl 语言一样，Python 源代码同样遵循 GPL(GNU General Public License) 协议。本节主要测试 python 的使用，从 python 命令行和脚本两个方面来说明。

- **查看系统支持的 phthon 版本**

```
[root@myir:/]# python
python        python3       python3.7     python3.7m
```

- **python 命令行测试**

启动 python，并在 python 提示符中输入以下文本信息，然后按 Enter 键查看运行效果：

```
[root@myir:/]# python

Python 3.7.2 (default, Apr 25 2023, 14:12:09)
[GCC 7.3.1 20180425 [linaro-7.3-2018.05 revision d29120a424ecfbc167ef90065c
0eeb on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("myir test")
myir test
```

退出 Python 命令行，执行 exit()即退出 Python：

```
>>> exit()
[root@myir:/]#
```

- **编写脚本测试 Python**

编写一个简单的 Python 脚本程序，所有 Python 文件将以 .py 为扩展名：

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
[root@myir:/]# vi test.py
[root@myir:/]# cat test.py
#!/usr/bin/env python3
print("myir test")
```

执行脚本文件，解释器在/usr/bin/env 中以 python2 运行目录中，使用以下命令执行脚本：

```
[root@myir:/]# chmod a+x test.py
[root@myir:/]# ./test.py
myir test
```

通过脚本参数调用 Python3 解释器开始执行脚本，直到脚本执行完毕。当脚本执行完成后，解释器不再有效。

9.2. 数据库

数据库(Database)是按照数据结构来组织、存储和管理数据的仓库。数据库有很多种类型，常用的数据库有 Access、Oracle、Mysql、SQL Server、SQLite 等。

1) System SQLlite

SQLite 是一个嵌入式 SQL 数据库引擎。与大多数其他 SQL 数据库不同，SQLite 没有单独的服务器进程。SQLite 直接读写普通磁盘文件。包含多个表、索引、触发器和视图的完整 SQL 数据库包含在单个磁盘文件中。这是一款轻型的数据库，是遵守 ACID 的关联式数据库管理系统，它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百 K 的内存就够了。这款数据库的运行处理速度比 Mysql、PostgreSQL 这两款都要快。

- **SQLlite 创建数据库**

启动 sqlite3，并创建一个新的数据库 <testDB.db>,在终端界面输入以下命令就可以进入操作界面。

```
[root@myir:/]# sqlite3 testDB.db
SQLite version 3.25.3 2018-11-05 20:37:38
Enter ".help" for usage hints.
sqlite>
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

上面的命令将在当前目录下创建一个文件 testDB.db。该文件将被 SQLite 引擎用作数据库。注意到 sqlite3 命令在成功创建数据库文件之后，将提供一个 sqlite> 提示符。

数据库被创建，您就可以使用 SQLite 的 .databases 命令来检查它是否在数据库列表中，如下所示：

```
sqlite> .databases
main: /home/root/testDB.db
sqlite>
```

使用 .quit 命令退出 sqlite 提示符，如下所示：

```
sqlite> .quit
[root@myir:/]
```

如果想要详细了解 SQLite 相关信息，请参考官网：https://www.sqlite.org/docs.html。

9.3. Qt 应用程序本地化

本节讨论本地化相关的设置和测试。本地化，指的是一个程序或软件在支持国际化的基础上，给定程序特定区域的语言信息使其在信息的输入输出等处理上适应特定区域人群的使用。这里允许程序所使用的一些语言环境变量在程序执行时动态配置。本章主要针对 QT 应用的本地化，以米尔演示镜像 MEasy HMI 2.x 为例进行说明。

1）多语言

本节主要以米尔演示镜像 MEasy HMI 2.x 为例说明多语言在 QT 项目中的实际应用。

阅读下面内容前请参考《**MEasy HMI 2.x** 开发手册》第 3.1 章 环境搭建，搭建起 MEasy HMI 的编译环境。

- **打开 mxapp2 工程**

MEasy HMI 2.x 对应的工程源码为 mxapp2.tar.gz,拷贝至上述文档所构建的环境里面去，并使用 QT Creator 打开这个工程。

- **生成 ts 文件**

通过终端进入 mxapp2 工程所在源码目录，并执行下面命令生成翻译文件。

```
sur@myir:~/download/mxapp2$ lupdate mxapp2.pro
Info: creating stash file /home/qinlh/download/MXAPP/.qmake.stash
Updating 'languages/language_zh.ts'...
```


深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

```
    Found 202 source text(s) (0 new and 202 already existing)
Updating 'languages/language_en.ts'...
Found 202 source text(s) (0 new and 202 already existing)
```

工程在发布之前已完成翻译的工作，此处不会重新生成 language_zh.ts 和 language_en.ts 文件，中文显示使用的是 language_zh.ts 生成的 qm 文件，英文显示使用的是 language_en.ts 生成的 qm 文件。

- **翻译文本**

打开 languages/language_en.ts 文件，针对翻译的源字符串为<source>节点里面的内容，翻译的目标字符串为<translation>里面的内容，用户可根据需求进行修改。

```
<message>
        <location filename="../Album.qml" line="50"/>
        <source>返回</source>
        <translation type="unfinished">Return</translation>
</message>
```

- **生成 qm 文件**

Ts 文件修改完成后需要手动生成翻译模版文件供应用使用，进入 languages 目录，使用如下命令即可生成翻译模版文件。

```
sur@myir:~/download/mxapp2/languages$ lrelease language_en.ts -qm language_en.qm
Updating 'language_en.qm'...
    Generated 183 translation(s) (2 finished and 181 unfinished)
Ignored 21 untranslated source text(s)
sur@myir:~/download/mxapp2/languages$ lrelease language_zh.ts -qm language_zh.qm
Updating 'language_en.qm'...
    Generated 183 translation(s) (2 finished and 181 unfinished)
    Ignored 21 untranslated source text(s)
```

- **应用翻译文件**

翻译文件的使用需要通过应用程序代码调用，具体调用方法参考源代码“translator.cpp”文件中的 LoadLanguage 成员函数 “。

2）字体

本节主要以米尔演示镜像 MEasy HMI 2.x 为例说明字体在 QT 项目中的实际应用。阅读下面内容前请参考《**MEasy HMI 2.x** 开发手册》第 3.1 章 环境搭建，搭建起 MEasy HMI 的编译环境。

- **安装字体文件**

字体文件可以直接放置到开发板文件系统/usr/lib/fonts/目录。

```
[root@myir:/]# ls /usr/lib/fonts/msyh.ttc
/usr/lib/fonts/msyh.ttc
```

或者直接添加的 QT 工程里面。

```
sur@myir:~/download/mxapp2/fonts$ tree
├── DIGITAL
│   ├── DIGITAL.TXT
│   └── DS-DIGIB.TTF
└── fontawesome-webfont.ttf
```

- **使用字体文件**

字体文件的使用是需要通过应用的代码来调用的，具体调用方法参考 main.cpp.中 iconFontInit()函数。

```
void iconFontInit()
{
    int fontId_digital = QFontDatabase::addApplicationFont(":/fonts/DIGITAL/DS-DIGIB.TTF");
    int fontId_fws = QFontDatabase::addApplicationFont(":/fonts/fontawesome-webfont.ttf");
    QString fontName_fws = QFontDatabase::applicationFontFamilies(fontId_fws).at(0);
    QFont iconFont_fws;
    iconFont_fws.setFamily(fontName_fws);
    QApplication::setFont(iconFont_fws);
    iconFont_fws.setPixelSize(20);
}
```

3）软键盘

本章节主要以米尔演示镜像 MEasy HMI 2.x 为例说明软键盘在 QT 项目中的实际应用。

阅读下面内容前请参考《**MEasy HMI 2.x** 开发手册》第 3.1 章 环境搭建，搭建起 MEasy HMI 的编译环境。

QT 从 5.7 版本开始在官方源码里面加入了 qt vritual keyboard 组件，myir 在发布的 MEasy HMI 2.x 配套的镜像中已经使用了 QT 5.12 版本,并配置了 vritual keyboard 这个组件。

- **软键盘嵌入到 qml 代码**

Qt 提供的软键盘只能在 qml 代码里面进行调用，调用前需要定义软键盘弹出的位置及软键盘的大小，如下面代码所示。

```
InputPanel {
id: inputPanel
x: adaptive_width/8
y: adaptive_height/1.06
z:99
anchors.left: parent.left
anchors.right: parent.right

states: State {
name: "visible"
when: inputPanel.active
PropertyChanges {
target: inputPanel
        y: adaptive_height/1.06 - inputPanel.height
}
}
```

- **触发软键盘**

软键盘是通过 QML 的 TextField、TextEdit 组件触发的，用户只需要在 UI 组件中加入此组件即可在 UI 上调出软键盘并使用。如果是 qt 系统，可以使用 QT 自带的 QT virtualkeyboard 输入。

```
TextField{
id: netmask_input
InputMethodHints: Qt.ImhFormattedNumbersOnly
onAccepted: digitsField.focus = true
font.family: "Microsoft YaHei"
color: "white"
}
```

- **使用软键盘**

软键盘的使用方法请参考《**MEasy HMI 2.x** 开发手册》第 2.6 章系统设置 UI 界面。用户点击可编辑的界面组件就会弹出软键盘。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

10. 参考资料

- **Linux kernel 开源社区**

  https://www.kernel.org/

- **全志开发社区**

  https://www.aw-ol.com

- **Linux 内核看门狗介绍**

  https://www.kernel.org/doc/html/latest/watchdog/index.html

- **嵌入式 Linux 下的 Qt**

  https://doc.qt.io/qt-5/embedded-linux.html

- **Systemd 网络配置**

  https://www.freedesktop.org/software/systemd/man/systemd.network.html

- **全志官方网站**

  https://www.allwinnertech.com/

附录一 联系我们

深圳总部

地址：深圳市龙岗区坂田街道发达路云里智能园 2 栋 6 楼 04 室

负责区域：广东、广西、海南、重庆、云南、贵州、四川、西藏、香港、澳门

传真：0755-25532724

电话：0755-25622735

武汉研发中心

地址：武汉东湖新技术开发区关南园一路 20 号当代科技园 4 号楼 1601 号

电话：027-59621648

华东地区

地址：上海市浦东新区金吉路 778 号浦发江程广场 1 号楼 805 室

负责区域：上海、福建、浙江、江苏、安徽、山东

传真：021-62087085

电话：021-62087019

华北地区

地址：北京市大兴区荣华中路 8 号院力宝广场 10 号楼 901 室

负责区域：辽宁、吉林、黑龙江、北京、天津、河北、山西、内蒙古、湖北、湖南、江西、河南、陕西、甘肃、宁夏、青海、新疆

传真：010-64125474
电话：010-84675491

销售联系方式

网址：www.myir.cn
邮箱：sales.cn@myir.cn

技术支持联系方式

邮箱：support.cn@myir.cn
武汉研发中心电话：027-59621648
深圳总部技术电话：0755-22316235

如果您通过邮件获取帮助时，请使用以下格式书写邮件标题：

[公司名称/个人--开发板型号] 问题概述

这样可以使我们更快速跟进您的问题，以便相应开发组可以处理您的问题。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

附录二 售后服务与技术支持

凡是通过米尔电子直接购买或经米尔电子授权的正规代理商处购买的米尔电子全系列产品，均可享受以下权益：

1、6 个月免费保修服务周期
2、终身免费技术支持服务
3、终身维修服务
4、免费享有所购买产品配套的软件升级服务
5、免费享有所购买产品配套的软件源代码，以及米尔电子开发的部分软件源代码
6、可直接从米尔电子购买主要芯片样品，简单、方便、快速；免去从代理商处购买时，漫长的等待周期
7、自购买之日起，即成为米尔电子永久客户，享有再次购买米尔电子任何一款软硬件产品的优惠政策
8、OEM/ODM 服务

如有以下情况之一，则不享有免费保修服务：

1、超过免费保修服务周期
2、无产品序列号或无产品有效购买单据
3、进液、受潮、发霉或腐蚀
4、受撞击、挤压、摔落、刮伤等非产品本身质量问题引起的故障和损坏
5、擅自改造硬件、错误上电、错误操作造成的故障和损坏
6、由不可抗拒自然因素引起的故障和损坏

产品返修：

用户在使用过程中由于产品故障、损坏或其他异常现象，在寄回维修之前，请先致电米尔电子客服部，与工程师进行沟通以确认问题，避免故障判断错误造成不必要的运费损失及周期的耽误。

维修周期：

收到返修产品后，我们将即日安排工程师进行检测，我们将在最短的时间内维修或更换并寄回。一般的故障维修周期为 5 个工作日（自我司收到物品之日起，不计运输过程时间），由于特殊故障导致无法短期内维修的产品，我们会与用户另行沟通并确认维修周期。

维修费用：

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735

在免费保修期内的产品，由于产品质量问题引起的故障，不收任何维修费用；不属于免费保修范围内的故障或损坏，在检测确认问题后，我们将与客户沟通并确认维修费用，我们仅收取元器件材料费，不收取维修服务费；超过保修期限的产品，根据实际损坏的程度来确定收取的元器件材料费和维修服务费。

运输费用：

产品正常保修时，用户寄回的运费由用户承担，维修后寄回给用户的费用由我司承担。非正常保修产品来回运费均由用户承担。

深圳市米尔电子有限公司 MYIR Electronics Limited
Web: www.myir.cn Mail: sales.cn@myir.cn Tel: 0755-25622735