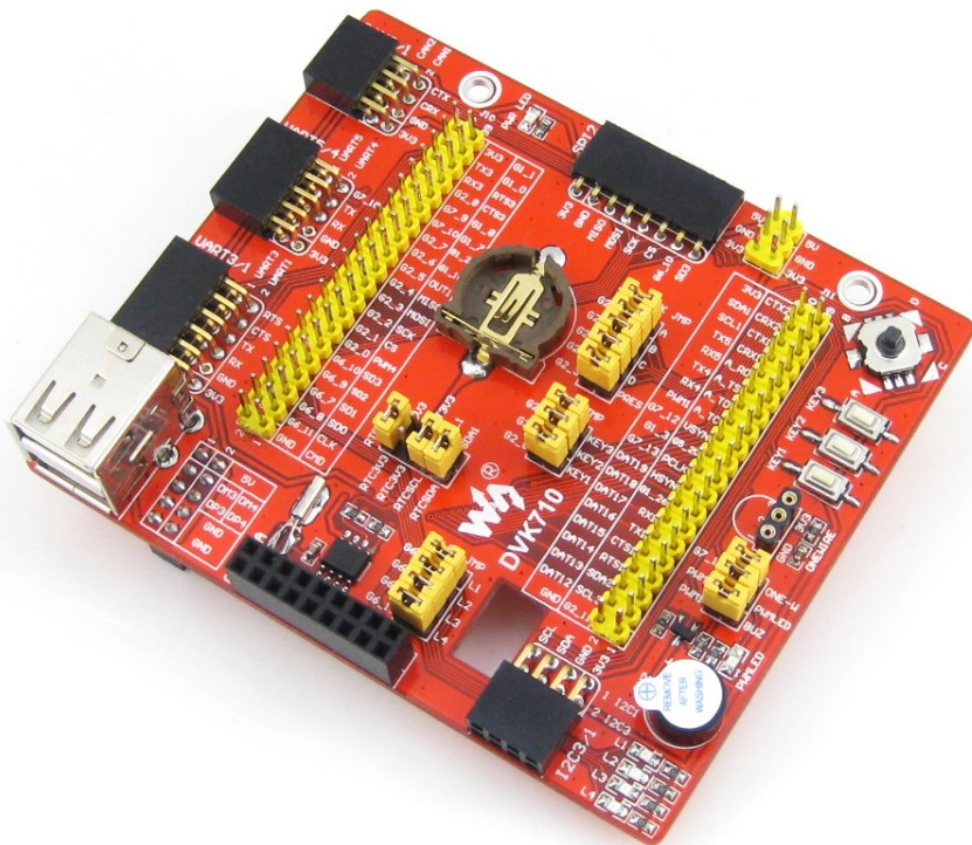


DVK710 扩展板

用户手册

2014.06.03 V1.0



版权声明

本手册所有权由深圳市微雪电子有限公司独家持有。未经本公司的书面许可，不得以任何方式或形式进行修改、分发或复制本文档的任何部分，否则一切后果由违者自负。

版本更新记录

版本	日期	说明
V1.0	2014.06.03	初始发布

目录

版权声明	1
版本更新记录	1
第一章 产品概述	1
第二章 准备工作	1
第三章 API 源码	1
第四章 功能测试	1
4.1 GPIO-BUTTONS 测试	1
4.2 JOYSTICK 摇杆测试	2
4.3 LED 测试	2
4.4 PWM 测试	3
4.5 PWM LED 测试	3
4.6 DS18B20 测试	4
4.7 RTC 测试	4
4.8 GPS 定位测试	5
4.9 MAG3110 测试	6
4.10 AT45DB E2PROM 测试	6
4.11 RS485 接收发送测试	7
4.12 USB CAMERA 测试	8
4.13 USB WIFI 测试	9
4.14 USB HOST 测试	11

第一章 产品概述

DVK710 是 MarsBoard（基于 freescale MX6Q 平台）所推出的功能扩展板。所有外接模块的驱动程序及应用程序的源码都开源提供，用户可以根据我们提供的程序学习添加更多外设模块。

第二章 准备工作

DVK710 提供的系统镜像内核版本为 3.0.35，文件系统为 ubuntu11.04，使用 windows 下工具 Mfgtools 烧写镜像到 emmc。

- 1) 在文件夹**镜像**中获取三个系统镜像 **u-boot**、**ulmage**、**oneirc.tgz**。
- 2) 具体烧写方法参考《MarS Board 用户手册_V1.2.pdf》第四章的内容进行烧写。

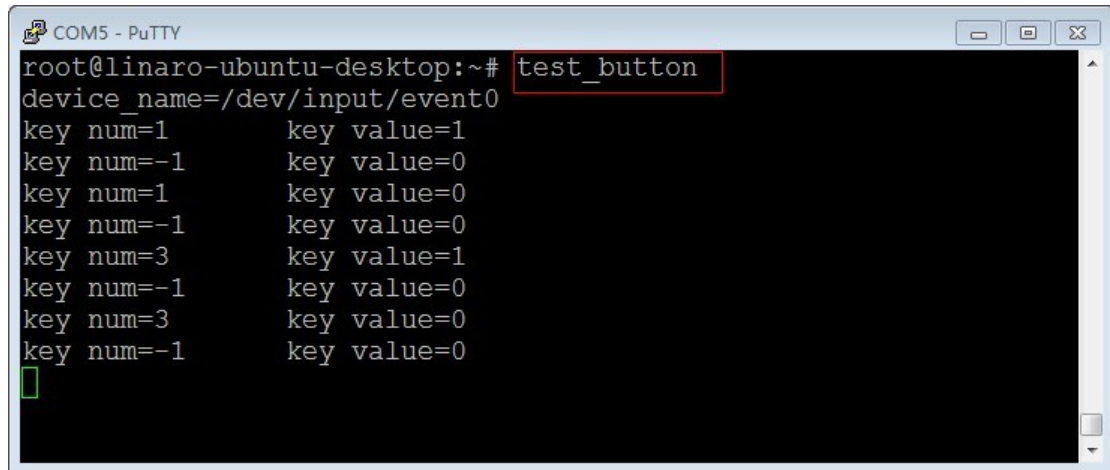
第三章 API 源码

所有测试的 API 源码在开发板/**home/linaro/API** 目录下。

第四章 功能测试

4.1 GPIO-BUTTONS 测试

- 1) 确保 KEY JMP 跳线帽短接。
- 2) 开发板终端下输入：**\$test_button**
- 3) 操作三个独立按键，终端下显示操作状态。

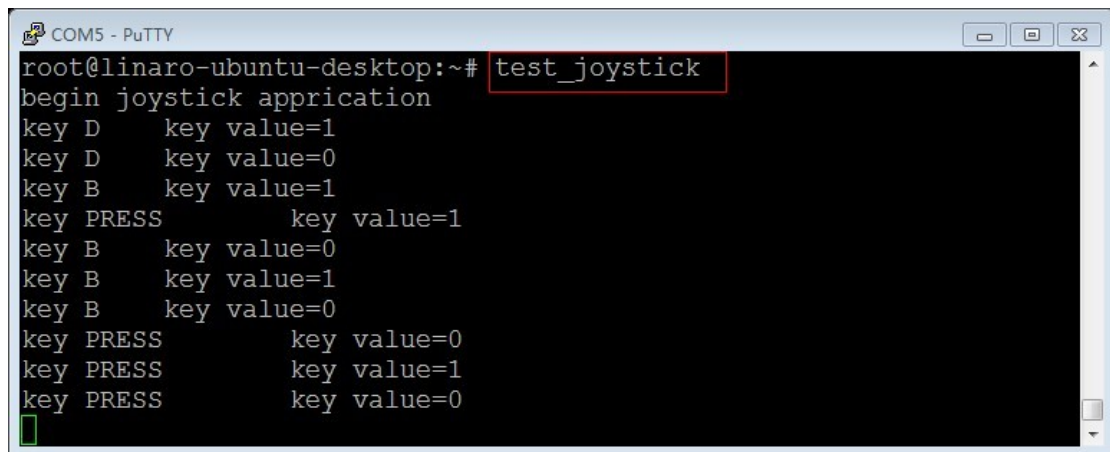


```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# test_button
device_name=/dev/input/event0
key num=1      key value=1
key num=-1     key value=0
key num=1      key value=0
key num=-1     key value=0
key num=3      key value=1
key num=-1     key value=0
key num=3      key value=0
key num=-1     key value=0
█
```

按键盘 ctrl+c 结束测试。

4.2 JOYSTICK 摇杆测试

- 1) 确保 JOYSTICK JMP 跳线帽短接。
- 2) 开发板终端下输入: `$test_joystick`
- 3) 操作 5 向摇杆, 终端显示摇杆状态。



```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# test_joystick
begin joystick apprication
key D      key value=1
key D      key value=0
key B      key value=1
key PRESS  key value=1
key B      key value=0
key B      key value=1
key B      key value=0
key PRESS  key value=0
key PRESS  key value=1
key PRESS  key value=0
█
```

按键盘 ctrl+c 结束测试。

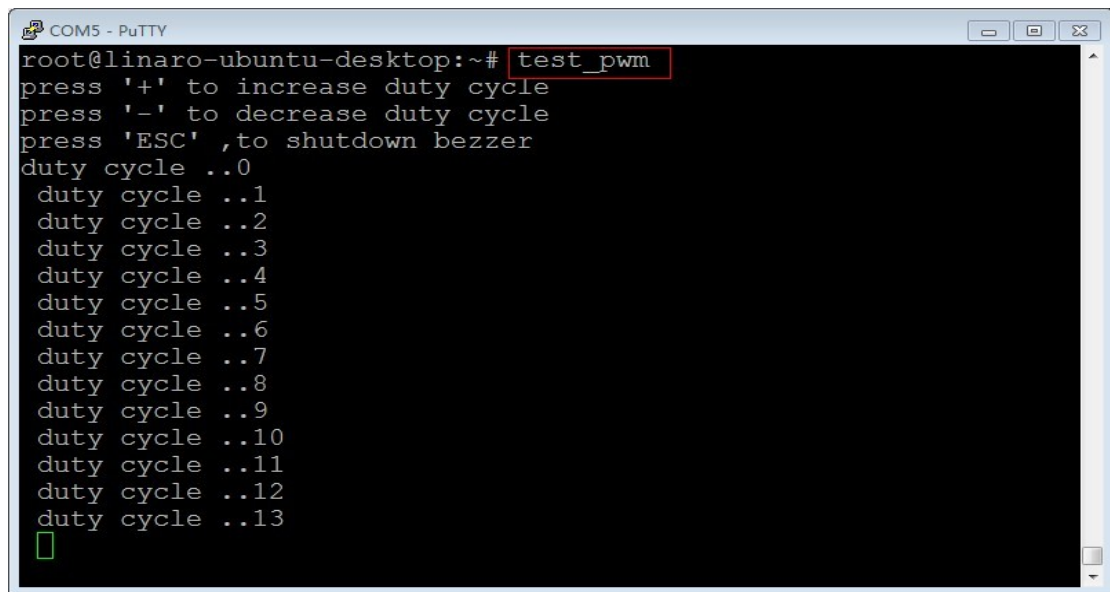
4.3 LED 测试

- 1) 确保四个 LED JMP 跳线帽短接。
- 2) 开发板终端下输入: `$test_led`
- 3) DVK710 上 led 灯闪烁变化。

按键盘 ctrl+c 结束测试。

4.4 PWM 测试

- 1) 确保 BUZ JMP 跳线帽短接。
- 2) 开发板终端下输入: `$test_pwm`
- 2) 使用键盘 “+”, “-” 对 PWM 信号脉宽进行调整。
- 3) 可以听到蜂鸣器响声不同的变化, 终端显示 pwm 信息。

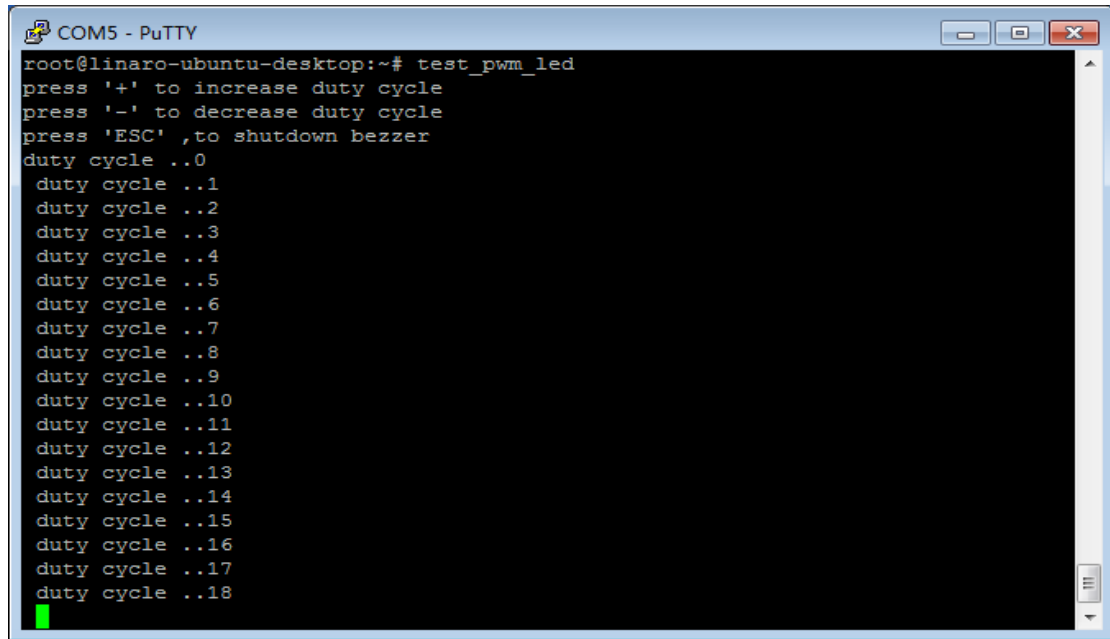
A screenshot of a terminal window titled "COM5 - PuTTY". The terminal shows the command "test_pwm" being entered at the prompt "root@linaro-ubuntu-desktop:~#". Below the command, there are instructions: "press '+' to increase duty cycle", "press '-' to decrease duty cycle", and "press 'ESC', to shutdown bezzzer". The output shows a list of "duty cycle" values from 0 to 13, each on a new line. A cursor is visible at the end of the last line.

```
root@linaro-ubuntu-desktop:~# test_pwm
press '+' to increase duty cycle
press '-' to decrease duty cycle
press 'ESC', to shutdown bezzzer
duty cycle ..0
duty cycle ..1
duty cycle ..2
duty cycle ..3
duty cycle ..4
duty cycle ..5
duty cycle ..6
duty cycle ..7
duty cycle ..8
duty cycle ..9
duty cycle ..10
duty cycle ..11
duty cycle ..12
duty cycle ..13
█
```

按键盘 ESC 结束测试。

4.5 PWM LED 测试

- 2) 确保 PWM_LED JMP 跳线帽短接。
- 2) 开发板终端下输入: `$test_pwm_led`
- 4) 使用键盘 “+”, “-” 对 PWM 信号脉宽进行调整。
- 5) 可以看到 LED 等由亮变暗, 终端显示 pwm 信息。



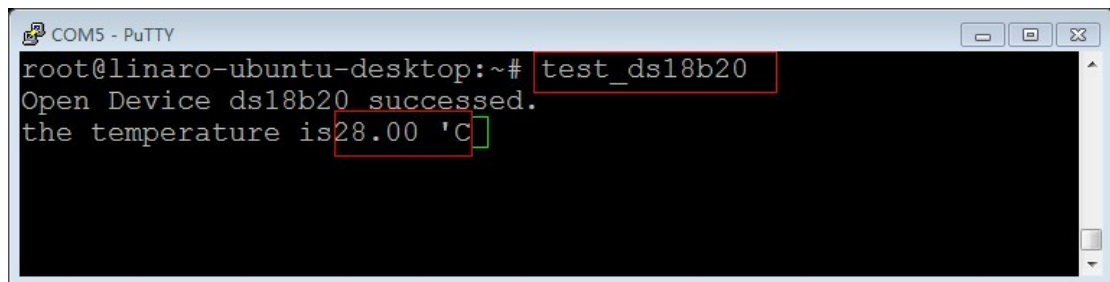
```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# test_pwm_led
press '+' to increase duty cycle
press '-' to decrease duty cycle
press 'ESC' ,to shutdown bezzar
duty cycle ..0
duty cycle ..1
duty cycle ..2
duty cycle ..3
duty cycle ..4
duty cycle ..5
duty cycle ..6
duty cycle ..7
duty cycle ..8
duty cycle ..9
duty cycle ..10
duty cycle ..11
duty cycle ..12
duty cycle ..13
duty cycle ..14
duty cycle ..15
duty cycle ..16
duty cycle ..17
duty cycle ..18
```

按键盘 ESC 结束测试。

注意：当数值为 256 时 LED 灯熄灭

4.6 DS18B20 测试

- 1) 确保 ONE-W JUMP 跳线短接
- 2) 接入 ds18b20 温度传感器到 ONEWIRE 接口。
- 3) 开发板终端输入：\$test_ds18b20
- 4) 终端下显示获取的当前环境温度。



```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# test_ds18b20
Open Device ds18b20 succeeded.
the temperature is 28.00 'C
```

按键盘 ctrl+c 结束测试。

注意：如果在没有接入 DS18B20 传感器的情况下运行测试程序，按键盘 ctrl+c 是不能退出程序的，这时只需接入传感器即可。

4.7 RTC 测试

- 1) 确保 RTC 跳线帽短接。

注意：3V3 和 BAT3V3 只能接上其中一个的跳线帽

2)开发板终端下输入命令：

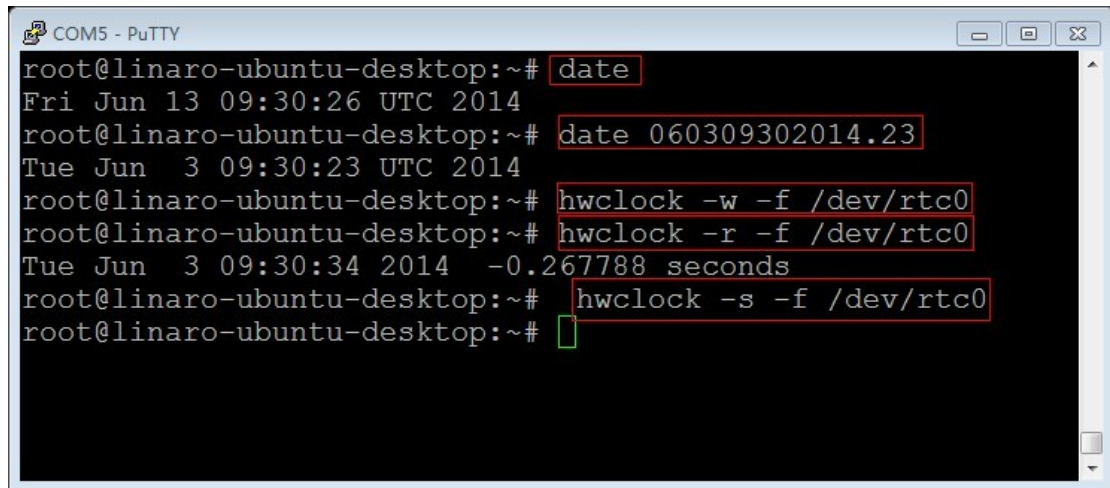
读取系统时间：\$date

设置系统时间：\$ date 060309302014.23

设置 RTC 硬件时间：\$hwclock -w -f /dev/rtc0

读取 RTC 时钟模块的硬件时间：\$hwclock -r -f /dev/rtc0

RTC 时钟模块的硬件时间同步到系统时间：\$ hwclock -s -f /dev/rtc0



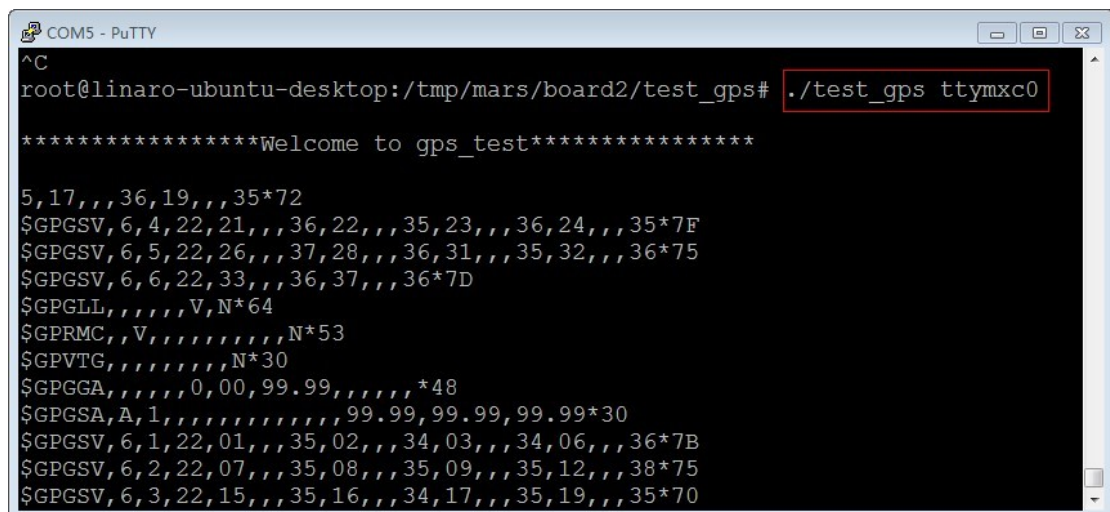
```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# date
Fri Jun 13 09:30:26 UTC 2014
root@linaro-ubuntu-desktop:~# date 060309302014.23
Tue Jun  3 09:30:23 UTC 2014
root@linaro-ubuntu-desktop:~# hwclock -w -f /dev/rtc0
root@linaro-ubuntu-desktop:~# hwclock -r -f /dev/rtc0
Tue Jun  3 09:30:34 2014  -0.267788 seconds
root@linaro-ubuntu-desktop:~# hwclock -s -f /dev/rtc0
root@linaro-ubuntu-desktop:~#
```

4.8 GPS 定位测试

1) 接入 UART GPS NEO-7M 模块到 UART1 接口。

2) 开发板终端下输入\$test_gps ttymxc0

3) 终端显示定位状态及经纬度信息。



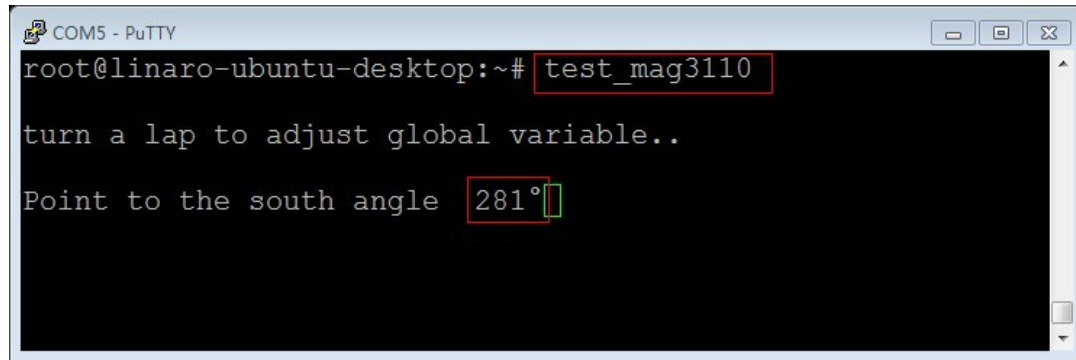
```
COM5 - PuTTY
^C
root@linaro-ubuntu-desktop:/tmp/mars/board2/test_gps# ./test_gps ttymxc0
*****Welcome to gps_test*****

5,17,,,36,19,,,35*72
$GPGSV,6,4,22,21,,,36,22,,,35,23,,,36,24,,,35*7F
$GPGSV,6,5,22,26,,,37,28,,,36,31,,,35,32,,,36*75
$GPGSV,6,6,22,33,,,36,37,,,36*7D
$GPGLL,,,,,V,N*64
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,0,00,99.99,,,,,*48
$GPGSA,A,1,,,,,,,,,99.99,99.99,99.99*30
$GPGSV,6,1,22,01,,,35,02,,,34,03,,,34,06,,,36*7B
$GPGSV,6,2,22,07,,,35,08,,,35,09,,,35,12,,,38*75
$GPGSV,6,3,22,15,,,35,16,,,34,17,,,35,19,,,35*70
```

按键盘 ctrl+c 结束测试。

4.9 MAG3110 测试

- 1) 接入 MAG3110 Board 到 I2C1 接口。
- 2) 开发板终端下输入：`$ test_mag3110_i2c1`
- 3) 旋转传感器来校准参数，校准完成后终端下显示获取的地磁信息。

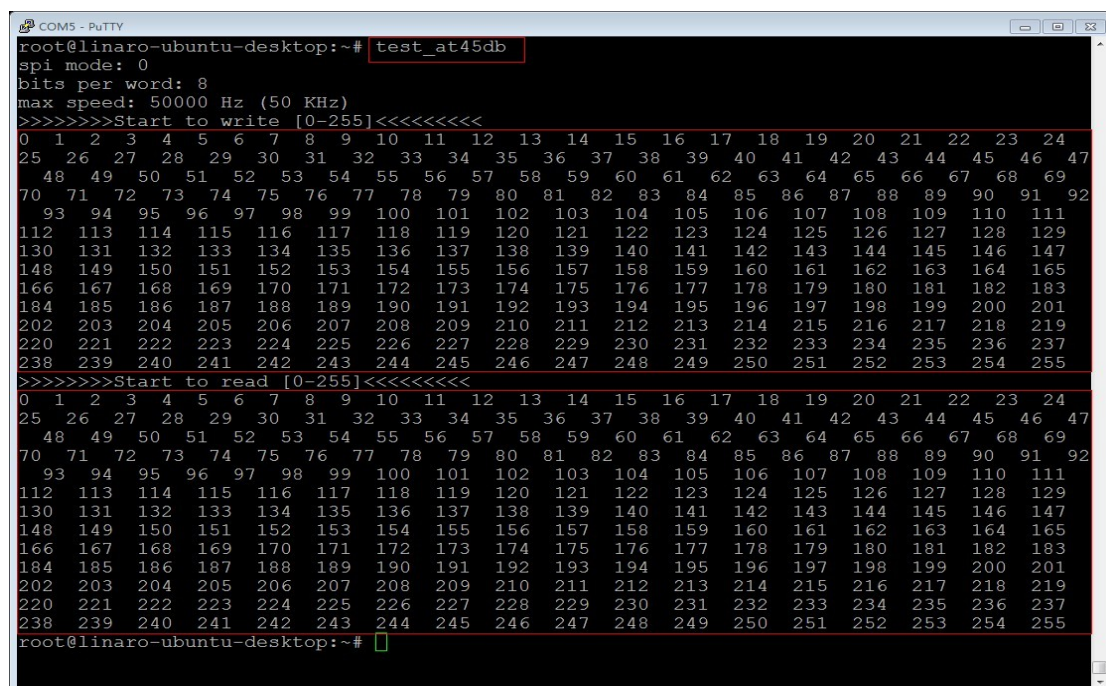


```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# test_mag3110
turn a lap to adjust global variable..
Point to the south angle 281°
```

按键盘 ctrl+c 结束测试。

4.10 AT45DB E2PROM 测试

- 1) 接入 AT45DBXX DataFlash Board 到 SPI1 接口。
- 2) 开发板终端下输入：`$test_at45db`
- 3) 终端下显示写入和读出数据。



```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# test_at45db
spi mode: 0
bits per word: 8
max speed: 50000 Hz (50 KHz)
>>>>>>>Start to write [0-255]<<<<<<<<<<<<
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201
202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219
220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237
238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
>>>>>>>Start to read [0-255]<<<<<<<<<<<<
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201
202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219
220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237
238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
root@linaro-ubuntu-desktop:~#
```

按键盘 ctrl+c 结束测试。

4.11 RS485 接收发送测试

1) 分别接入两个 RS485 Board 到 UART4 和 UART5 接口，并用杜邦线

把模块上的 A 和 B 对应连接。

2) 打开两个终端（另一个可以通过网络访问），分别输入命令：

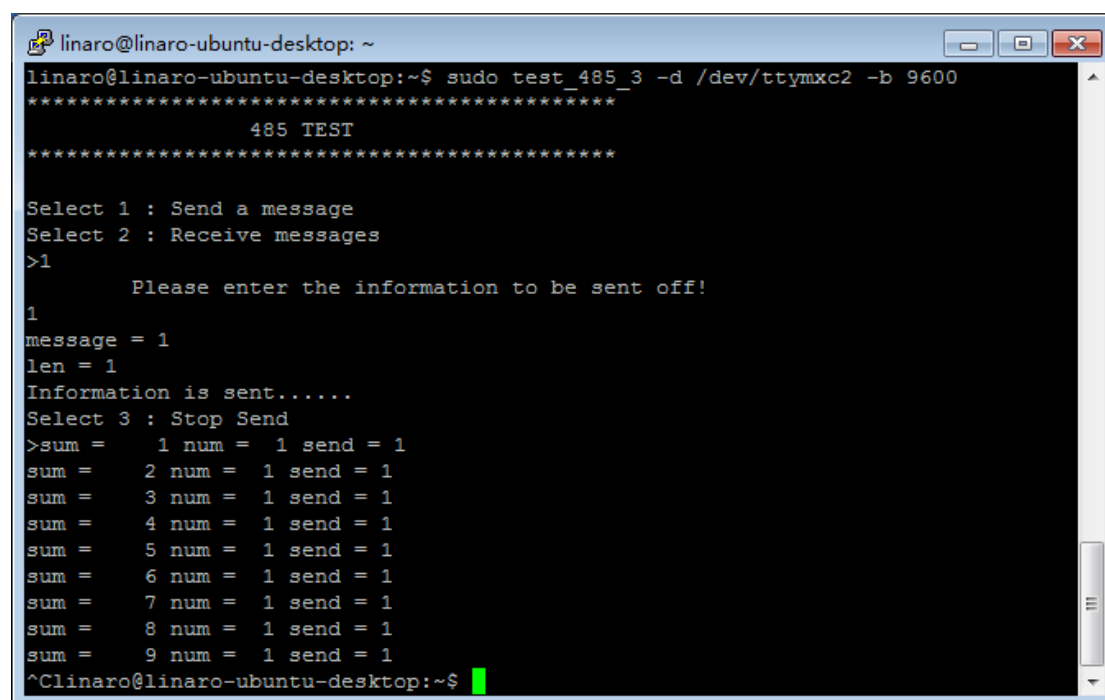
```
$test_485_3 -d /dev/ttymx2 -b 9600
```

```
$test_485_5 -d /dev/ttymx4 -b 9600
```

按照提示选择相应的功能，实现一个终端发一个终端收。

3) 终端下可以看到发送端和接收端的数据。

发送端：



```
linaro@linaro-ubuntu-desktop: ~
linaro@linaro-ubuntu-desktop:~$ sudo test_485_3 -d /dev/ttymx2 -b 9600
*****
                        485 TEST
*****

Select 1 : Send a message
Select 2 : Receive messages
>1
    Please enter the information to be sent off!
1
message = 1
len = 1
Information is sent.....
Select 3 : Stop Send
>sum = 1 num = 1 send = 1
sum = 2 num = 1 send = 1
sum = 3 num = 1 send = 1
sum = 4 num = 1 send = 1
sum = 5 num = 1 send = 1
sum = 6 num = 1 send = 1
sum = 7 num = 1 send = 1
sum = 8 num = 1 send = 1
sum = 9 num = 1 send = 1
^Clinaro@linaro-ubuntu-desktop:~$
```

接收端：

```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# test_485_5 -d /dev/ttyMXC4 -b 9600
*****
                        485 TEST
*****

Select 1 : Send a message
Select 2 : Receive messages
>2
Select 3 : Stop Receive
>
sum = 1 num = 1 recv = 1
sum = 2 num = 1 recv = 1
sum = 3 num = 1 recv = 1
sum = 4 num = 1 recv = 1
sum = 5 num = 1 recv = 1
sum = 6 num = 1 recv = 1
sum = 7 num = 1 recv = 1
sum = 8 num = 1 recv = 1
sum = 9 num = 1 recv = 1
^v
```

4.12 USB CAMERA 测试

- 1) 接入 USB Camera 到 USB Host 接口，并插入网线。
- 2) 开发板输入：\$ifconfig eth0 查看 ip。

```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 1e:ed:19:27:1a:b3
          inet addr:192.168.1.179  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:186266 errors:0 dropped:0 overruns:0 frame:0
          TX packets:294503 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15607788 (15.6 MB)  TX bytes:395882160 (395.8 MB)

root@linaro-ubuntu-desktop:~#
```

可以看到 IP 地址为 192.168.1.179

- 3) 开发板输入：

```
$cd /home/linaro/API/mjpg-streamer
```

```
$./start.sh
```

- 4) 在局域网内的 PC 机上打开浏览器，输入地址：

<http://192.168.1.179:8080/javascript.html>

注意：ip 地址要以开发板的 ip 地址一致，根据实际修改！

可以看到开发板作为 web 服务器上传的视频流。



按键盘 ctrl+c 结束测试。

4.13 USB WIFI 测试

接入 USB 无线网卡，在终端下输入相应命令：

1) 查看网络状态：

\$ifconfig -a

```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 1e:ed:19:27:1a:b3
          inet addr:192.168.1.188  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4239 errors:0 dropped:0 overruns:0 frame:0
          TX packets:362 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:657425 (657.4 KB)  TX bytes:60112 (60.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  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:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

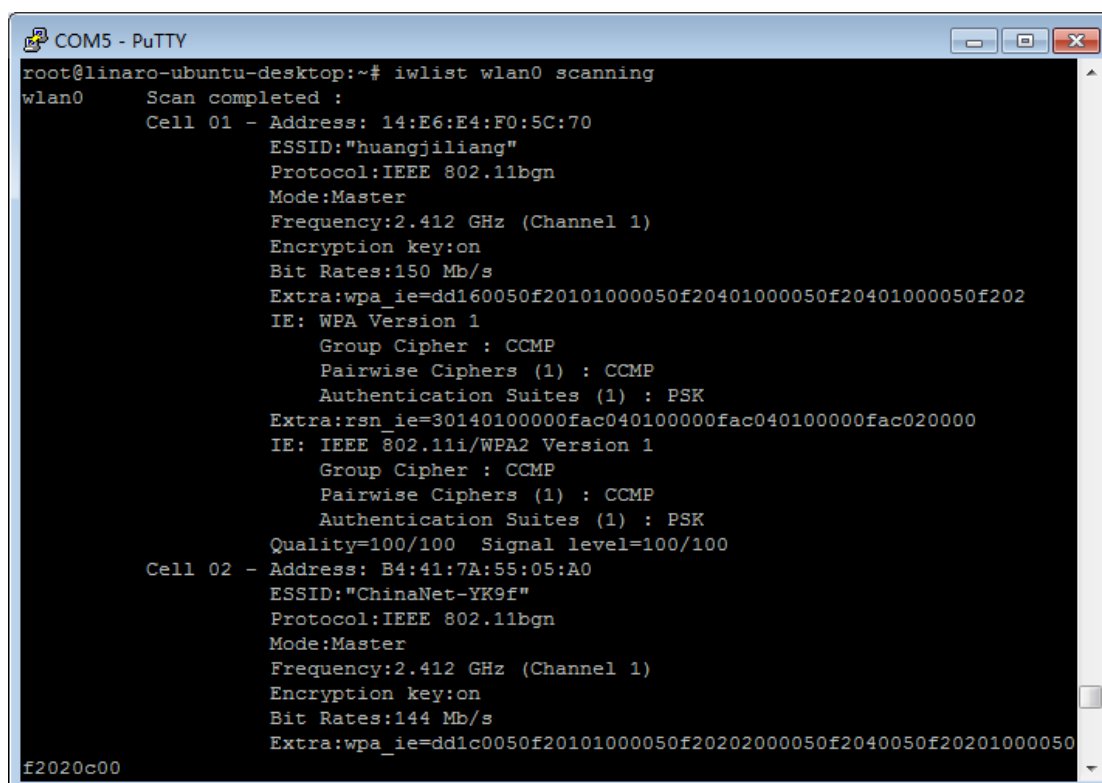
wlan0     Link encap:Ethernet  HWaddr 08:57:00:22:8f:66
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@linaro-ubuntu-desktop:~#
```

可以看到 wlanx 的网络节点则说明 usb 无线网卡驱动成功。

2) 通过命令行连接 wifi 热点:

`$iwlist wlan2 scanning`



```
COM5 - PuTTY
root@linaro-ubuntu-desktop:~# iwlist wlan0 scanning
wlan0    Scan completed :
          Cell 01 - Address: 14:E6:E4:F0:5C:70
                    ESSID:"huangjiliang"
                    Protocol:IEEE 802.11bgn
                    Mode:Master
                    Frequency:2.412 GHz (Channel 1)
                    Encryption key:on
                    Bit Rates:150 Mb/s
                    Extra:wpa_ie=dd160050f20101000050f20401000050f20401000050f202
                    IE: WPA Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                    Extra:rsn_ie=30140100000fac040100000fac040100000fac020000
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                    Quality=100/100  Signal level=100/100
          Cell 02 - Address: B4:41:7A:55:05:A0
                    ESSID:"ChinaNet-YK9f"
                    Protocol:IEEE 802.11bgn
                    Mode:Master
                    Frequency:2.412 GHz (Channel 1)
                    Encryption key:on
                    Bit Rates:144 Mb/s
                    Extra:wpa_ie=dd1c0050f20101000050f20202000050f2040050f20201000050
f2020c00
```

连接热点（WaveshareNet 即为热点名称，根据实际修改）:

禁用有线网络:

`$wpa_cli -p/var/run/wpa_supplicant remove_network 0`

扫描热点:

`$wpa_cli -p/var/run/wpa_supplicant ap_scan 1`

添加热点:

`$wpa_cli -p/var/run/wpa_supplicant add_network`

设置连接的热点的 SSID:

`$wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid ""WaveshareNet""`

设置热点的密码（根据实际修改）:

`$wpa_cli -p/var/run/wpa_supplicant set_network 0 psk ""12345678""`

设置热点为首选的连接网络:

`$wpa_cli -p/var/run/wpa_supplicant select_network 0`

设置开发板的 IP 地址和子网掩码:

`$ifconfig wlan0 192.168.1.196 netmask 255.255.255.0`

设置默认网关:

`$route add default gw 192.168.1.1`

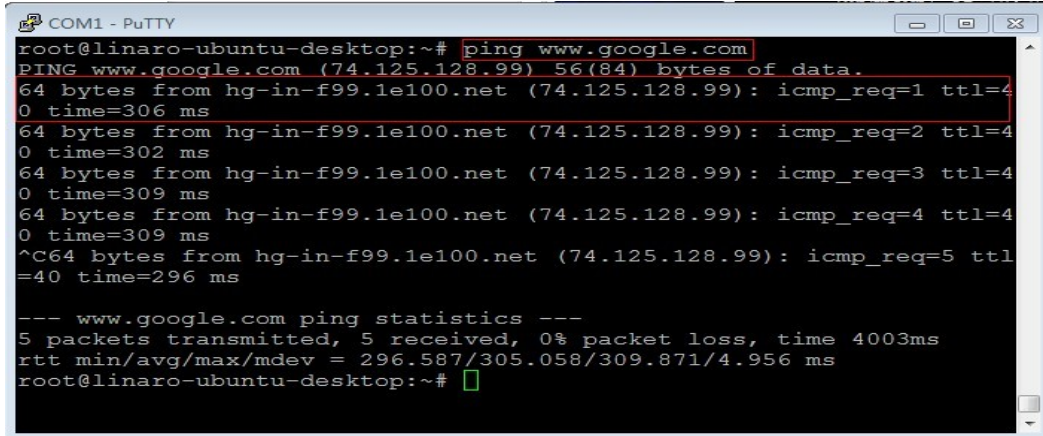
修改域名解析服务器地址:

\$vi /etc/resolv.conf

添加 nameserver 8.8.8.8

3) 测试网络情况，终端下输入：

\$ping www.baidu.com 查看连接情况。



```
COM1 - PuTTY
root@linaro-ubuntu-desktop:~# ping www.google.com
PING www.google.com (74.125.128.99) 56(84) bytes of data:
64 bytes from hg-in-f99.1e100.net (74.125.128.99): icmp_req=1 ttl=4
0 time=306 ms
64 bytes from hg-in-f99.1e100.net (74.125.128.99): icmp_req=2 ttl=4
0 time=302 ms
64 bytes from hg-in-f99.1e100.net (74.125.128.99): icmp_req=3 ttl=4
0 time=309 ms
64 bytes from hg-in-f99.1e100.net (74.125.128.99): icmp_req=4 ttl=4
0 time=309 ms
^C64 bytes from hg-in-f99.1e100.net (74.125.128.99): icmp_req=5 ttl=
=40 time=296 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 296.587/305.058/309.871/4.956 ms
root@linaro-ubuntu-desktop:~#
```

如果 ping 不通网络，可能是没有拔掉网线，也可以直接在 ubuntu 的图形界面下设置

4.14 USB HOST 测试

- 1) 接入 USB 键盘或 USB 鼠标等 USB 设备到 DVK710 的 USB 接口
- 2) 系统可以识别鼠标和键盘等 USB 设备。

注意：USB 接口在不宜接移动硬盘，容易损坏。