

IDO-SBC3568-V1B Buildroot 系统使用手册

1、硬件资源概况

1.1 主板照片

1.2 硬件资源及设备节点

2、调试

2.1 串口调试

2.2 ADB调试

2.3 ssh调试

3、UART

3.1 测试方法

4、USB

4.1 电源控制

5、TF CARD

6、以太网

6.1 静态IP设置

7、WiFi

8.1 连接蓝牙设备

9、4G

10、音频

10.1 查看声卡设备

10.2 播放音频

10.4 录音

12、RTC

12.1 获取RTC时间

12.2 设置RTC时间

13、开机自启动

14、屏幕控制

14.1 背光调节

15、按键

16、ADC

16.1 ADC转换方法

16.2 测试

17、CAN

17.1测试

18、扩展IO

19.1 测试



IDO-SBC3568-V1B Buildroot系统使用手册

深圳触觉智能科技有限公司

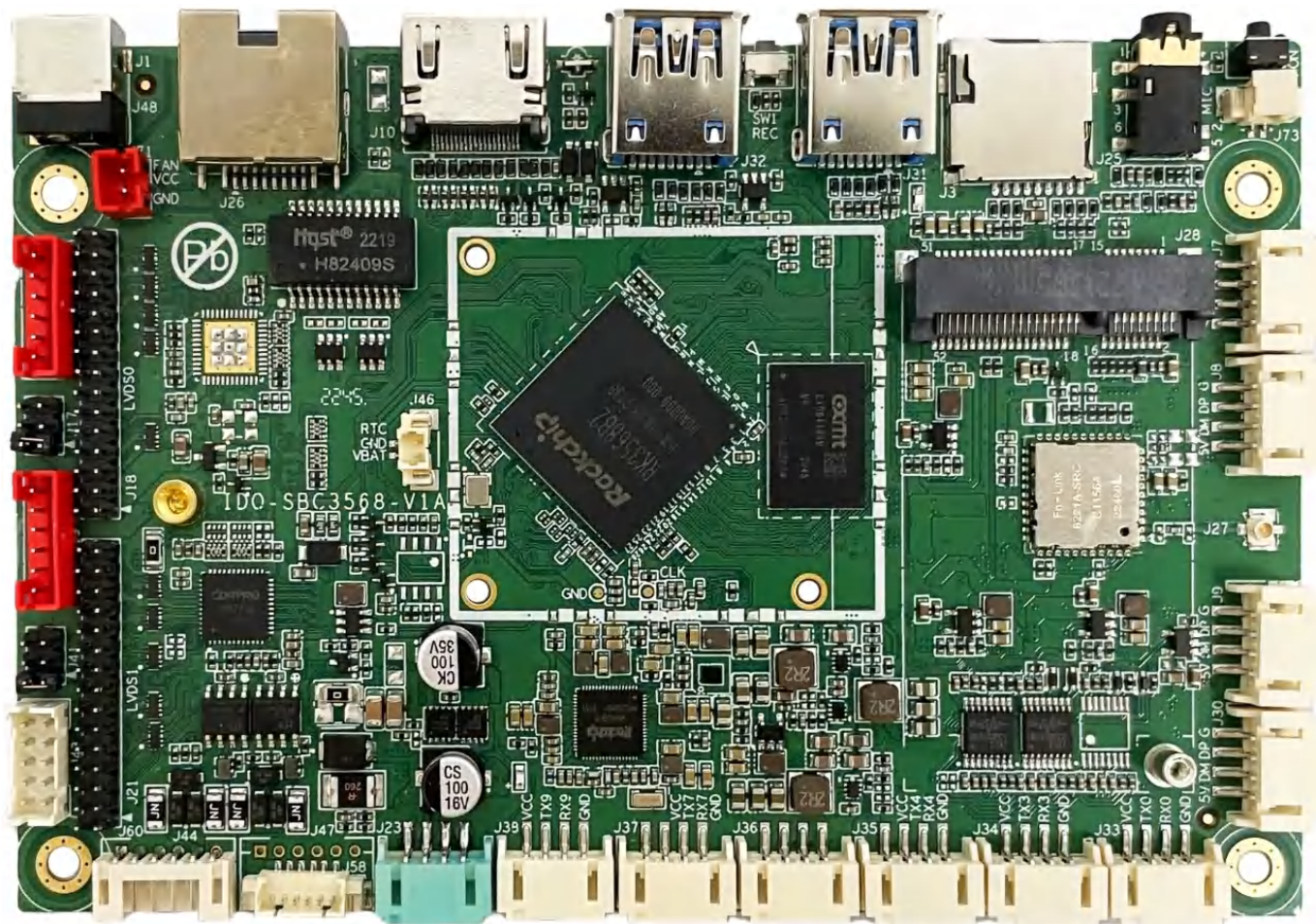
www.industio.cn

文档修订历史

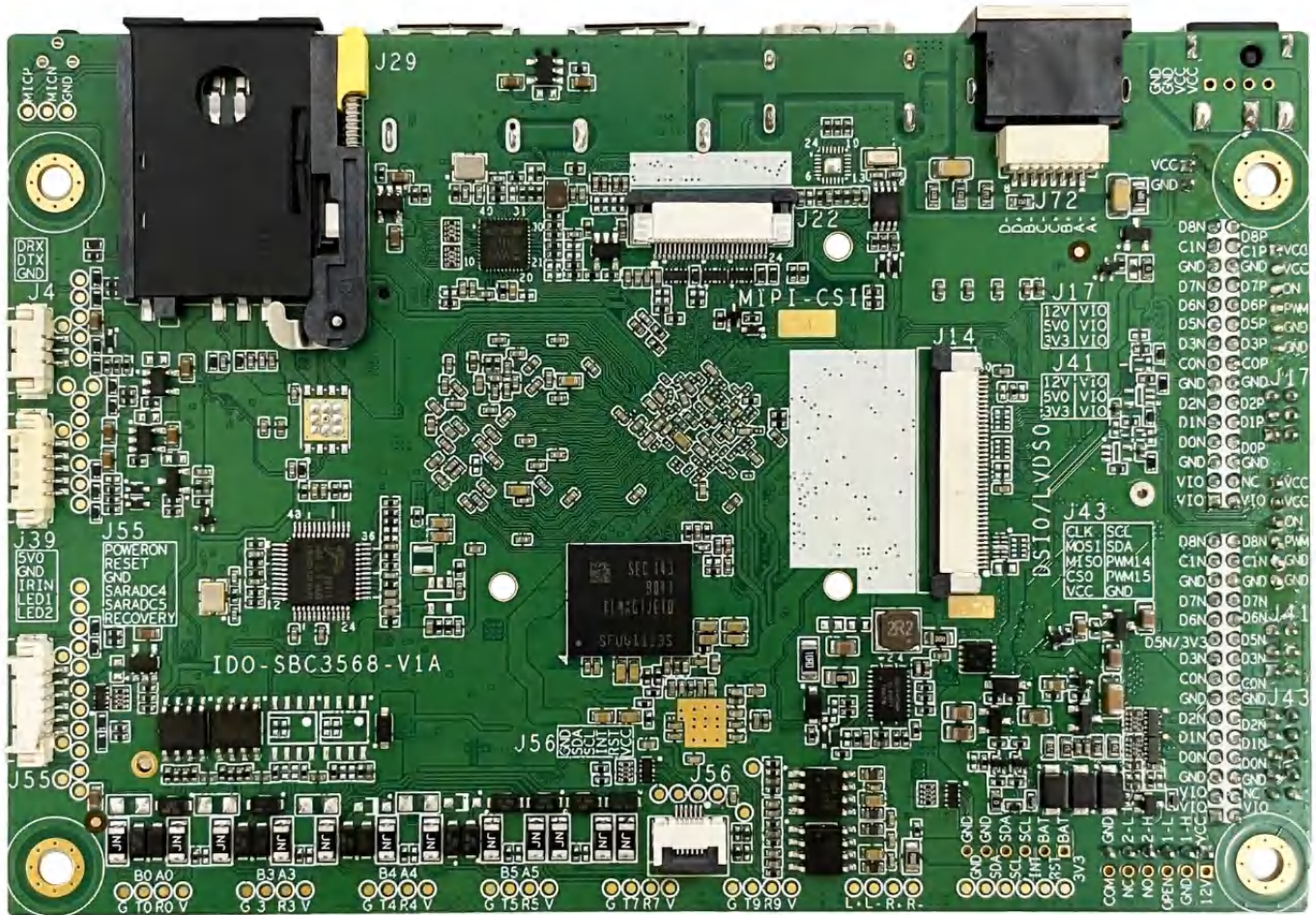
版本	修订内容	修订	审核	日期
V1.0	创建文档；	刘崇凯		2023/5/17

1、硬件资源概况

1.1 主板照片



IDO-SBC3568-V1A正面实物图



IDO-SBC3568-V1A背面实物图

1.2 硬件资源及设备节点

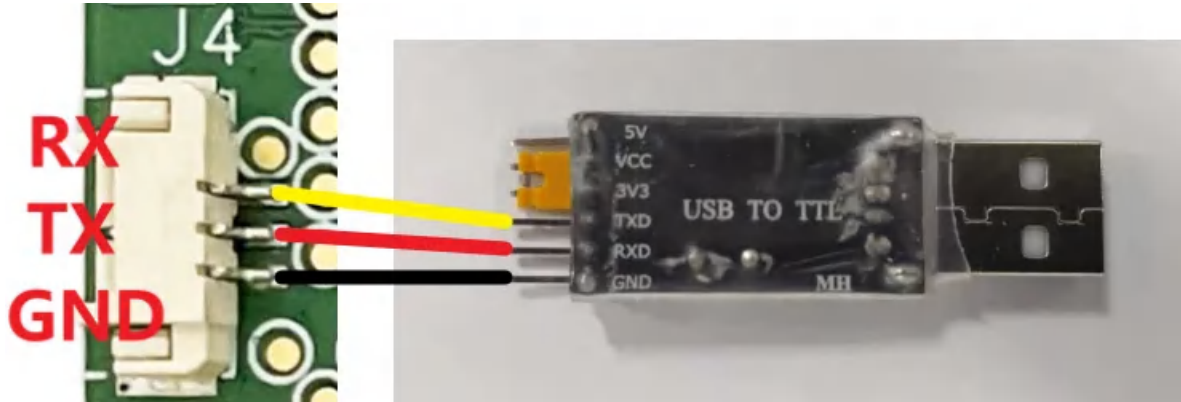
序号	名称	描述	设备节点
1	内核版本	Linux 4.19.219	
2	系统版本	buidroot	
3	内存	LPDDR4 (2G/4G/8GB选配)	
4	存储	eMMC5.1 (16GB / 32GB / 64GB / 128GB选配)	
5	供电	DC接口12V@2A	

6	显示	HDM LVDS eDP MIPI	
7	USB OTG	USB OTG Type-C	
8	USB HOST	USB3.0 HOST X 1 USB2.0 HOST X 4	
9	TF Card	TF Card x 1	
10	以太网	千兆以太网 x1	eth0
11	WIFI/BT	AM-NM371SM 2.4G	wlan0 、 hci0
12	扬声器		
13	MIC		
14	耳机	3.5mm 国标	
15	Camera	OV5648、OV8858	
16	串口	RS232 x 4 RS485 x 2	
17	调试串口	TTL x 1	
18	RTC	HYM8563 x 1	
19	系统指示灯	x1	
20	ADC按键	2路	
21	4G	1路支持USB2.0 和USB2.0 MIPI PCIE 接口4G模块	
22	POWER ON	x1	
23	CAN	x2	can1 can2

2、调试

2.1 串口调试

串口调试端口位于J4，通信参数为1500000 8 N 1，电平状态为TTL电平。

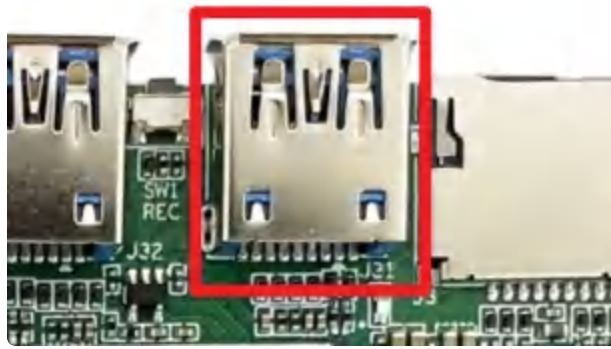


串口调试默认登录账号密码为 `linaro @ linaro`。

```
Bash |
1  Debian GNU/Linux 10 linaro-alip ttyFIQ0
2
3  linaro-alip login: linaro
4  密码:
5  上一次登录: 四 2月 14 10:13:30 UTC 2019ttyFIQ0 上
6  Linux linaro-alip 4.19.219 #57 SMP Fri Nov 4 11:14:32 CST 2022 aarch64
7
8  The programs included with the Debian GNU/Linux system are free software;
9  the exact distribution terms for each program are described in the
10 individual files in /usr/share/doc/*/copyright.
11
12 Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
13 permitted by applicable law.
14 linaro@linaro-alip:~$
15
```

2.2 ADB调试

ADB调试端口位于J31，使用TYPE-C线，连接主板的TYPE-C端口和电脑，即可在电脑上使用adb调试。



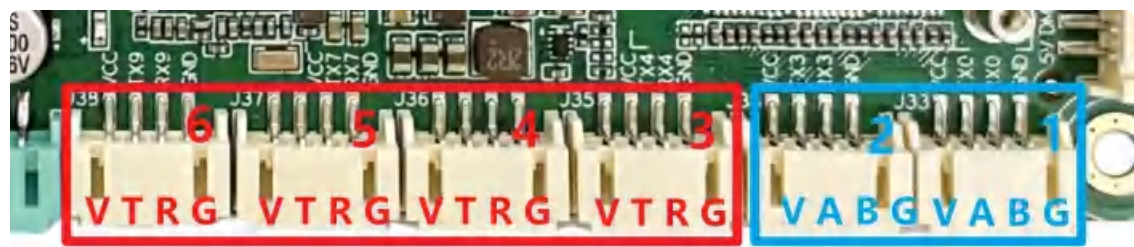
```
C:\Users\ronnie>adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
[root@RK356X:/]# ls
ls
bin          init         media  proc        sdcard      udisk
busybox.config lib          misc   rockchip_test sys          userdata
data         lib64        mnt    root        system      usr
dev          linuxrc      oem    run         timestamp   var
etc          lost+found  opt    sbin        tmp         vendor
[root@RK356X:/]#
```

2.3 ssh调试

系统默认登录账号密码为root @ rockchip。
ssh登录需要知道主板的IP，获取IP的方法，请参考第5章以太网的说明。

3、UART

主板共配置了4路串口（不包括调试串口），其中1路支持流控。



串口接口位置及引脚定义如上图所示，设备节点列表如下：

序号	丝印	功能	设备节点
1	J33	RS485	/dev/ttyS0
2	J34	RS485	/dev/ttyS3

3	J35	RS232	/dev/ttyS4
4	J36	RS232	/dev/ttyS5
5	J37	RS232	/dev/ttyS7
6	J38	RS232	/dev/ttyS9

3.1 测试方法

以上串口均可以使用microcom工具进行测试

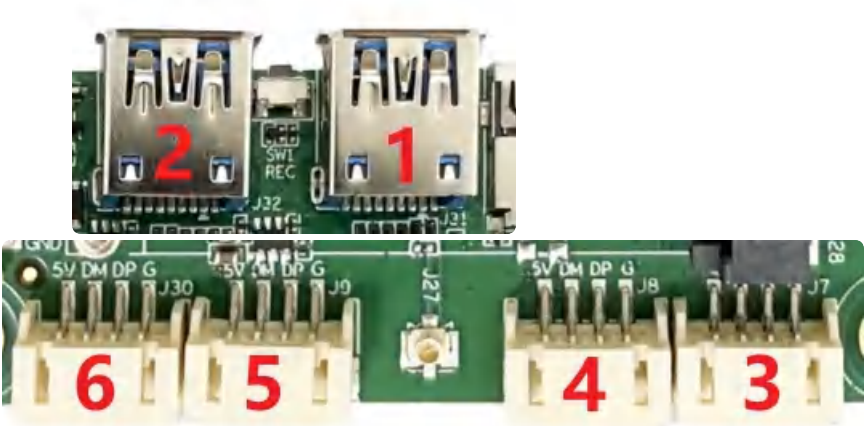
▼ Plain Text |

```
1 [root@RK356X:/]# microcom -s 115200 -p /dev/ttyS0
2 [ 2683.415483] of_dma_request_slave_channel: dma-names property of node '/s
  erial@fdd50000' missing or empty
3 [ 2683.415544] ttyS0 - failed to request DMA, use interrupt mode
4
```

当按下键盘时，串口会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【\】组合键，然后输入quit退出测试。

4、USB



USB接口如上图所示，功能说明如下

序号	丝印	功能	控电节点
----	----	----	------

2	J31	USB 2.0 HOST	/sys/class/leds/usb3_host_pwr/brightness
3	J7	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness
4	J8	USB 2.0 HOST	/sys/class/leds/usb2_fe2_pwr/brightness
5	J9	USB 2.0 HOST	/sys/class/leds/usb2_fe3_pwr/brightness
6	J30	USB 2.0 HOST	/sys/class/leds/usb2_fe4_pwr/brightness

USB OTG 切换命令

USB OTG 支持host 和device 模式的切换，软件切换方法如下

```

1  ## host
2  echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3  ## device
4  echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode

```

当USB-HOST插入U盘后，会自动挂载/media/linaro/目录下：

```

1  [root@RK356X:/]# mount
2  ...
3  /dev/sda1 on /media/usb0 type vfat (rw,nodev,noexec,noatime,nodiratime,fmask=0022,dmask=0022,codepage=936,ioccharset=utf8,shortname=mixed,errors=remount-ro)
4  ...

```

4.1 电源控制

默认所有USB-HOST的电源都是开启的，其中USB3-5我们提供了开启/关闭电源的方法。

打开USB6的电源：

```

1  [root@RK356X:/]# echo 255 > /sys/class/leds/usb2_fe4_pwr/brightness

```

关闭USB6的电源：

```
1 [root@RK356X:/]# echo 0 > /sys/class/leds/usb2_fe4_pwr/brightness
```

其他USB的电源控制方法类似。

5、TF CARD

将SD卡插入到SD卡槽中，将自动挂载到/mnt/sdcard/目录下。

```
1 [root@RK356X:/]# mount
2 ...
3 /dev/mmcblk1p1 on /mnt/sdcard type vfat (rw,noatime,uid=1000,gid=1000,fmask
  =0133,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors=remount
  -ro)
4 ...
```

6、以太网

主板配置了1个1000M以太网接口，对应的网络设备节点为eth0。



两路以太网接口默认IP获取方式为 dhcp。

6.1静态IP设置

以eth0设置静态IP地址为例，修改/etc/network/interfaces，在文件中添加如下内容

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 iface eth0 inet static
6 address 192.168.0.234
7 netmask 255.255.255.0
8 gateway 192.168.0.1
9 dns-nameservers 114.114.114.114
```

其中，dns-nameservers一项为默认dns。

7、WiFi

在使用 WIFI时连接好WiFi天线，设备节点为wlan0

```
1 [root@RK356X:/]# ifconfig wlan0
2 wlan0      Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
3             inet addr:169.254.41.145  Bcast:169.254.255.255  Mask:255.255.0.0
4             inet6 addr: fe80::b05:fca4:fb45:9468/64 Scope:Link
5             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
7             TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
8             collisions:0 txqueuelen:1000
9             RX bytes:0 (0.0 B)  TX bytes:21920 (21.4 KiB)
```

系统开机通过/etc/init.d/S80wifireconnect脚本开启WiFi服务，修改/userdata/cfg/wpa_supplicant.conf，填写正确的热点账号和密码：

```

1 [root@RK356X:/]# cat /userdata/cfg/wpa_supplicant.conf
2 ctrl_interface=/var/run/wpa_supplicant
3 ap_scan=1
4 update_config=1
5
6 network={
7     ssid="TP-LINK_B87A"
8     psk="12345678"
9     key_mgmt=WPA-PSK
10 }
11 [root@RK356X:/]#

```

重启后，将自动连接上热点：

```

1 [root@RK356X:/]# ifconfig wlan0
2 wlan0      Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
3             inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
4             inet6 addr: fe80::220a:b25:4bd:2e3a/64 Scope:Link
5             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6             RX packets:26 errors:0 dropped:0 overruns:0 frame:0
7             TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
8             collisions:0 txqueuelen:1000
9             RX bytes:5075 (4.9 KiB)  TX bytes:3913 (3.8 KiB)
10
11 [root@RK356X:/]#

```

8.1 连接蓝牙设备

设备节点为hci0，通过/usr/bin/bt_init.sh脚本开启蓝牙功能

```

1 [root@RK356X:/]# /usr/bin/bt_init.sh

```

蓝牙功能开启后，将产生hci0节点

▼

Bash

```
1 [root@RK356X:/]# hciconfig -a
2 hci0:   Type: Primary   Bus: UART
3         BD Address: F3:7A:FA:A4:5E:22   ACL MTU: 1021:8   SCO MTU: 64:1
4         DOWN
5         RX bytes:668 acl:0 sco:0 events:34 errors:0
6         TX bytes:423 acl:0 sco:0 commands:34 errors:0
7         Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
8         Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
9         Link policy: RSWITCH SNIFF
10        Link mode: SLAVE ACCEPT
```

使用hcitool测试蓝牙扫描功能

▼

Bash

```
1 [root@RK356X:/]# hciconfig hci0 up
2 [root@RK356X:/]# hcitool -i hci0 scan
3 Scanning ...
4      94:87:E0:9D:14:12      seeyou
5      4C:4F:EE:12:6C:A3      OnePlus 8 Pro
6      5C:C5:63:02:31:19      客厅的小米电视
```

9、4G

主板默认适配EC20模块（4G），上电前，正确按照模块和SIM卡，上电后，系统会自动进行拨号上网。

序号	模块名称	说明
1	EC20	4G LTE

拨号成功会产生wwan0网络节点：

```

1 [root@RK356X:/]# ifconfig wwan0
2 wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500
3         inet 10.101.61.51 netmask 255.255.255.248
4         inet6 fe80::fc:f6ff:fe8d:bab6 prefixlen 64 scopeid 0x20<link>
5         ether 02:fc:f6:8d:ba:b6 txqueuelen 1000 (Ethernet)
6         RX packets 42 bytes 7013 (6.8 KiB)
7         RX errors 0 dropped 0 overruns 0 frame 0
8         TX packets 57 bytes 4608 (4.5 KiB)
9         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 linaro@linaro-alip:~$

```

使用ping命令测试4G上网功能是否正常：

```

1 [root@RK356X:/]# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (183.232.231.174) from 10.101.61.51 wwan0: 56(84) bytes of data.
3 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=1 ttl=55 time=33.3 ms
4 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=2 ttl=55 time=48.1 ms
5 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=3 ttl=55 time=46.2 ms
6 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=4 ttl=55 time=45.2 ms
7 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=5 ttl=55 time=42.1 ms

```

10、音频

喇叭接口位于J23、MIC位于J73、耳机接口位于J23。



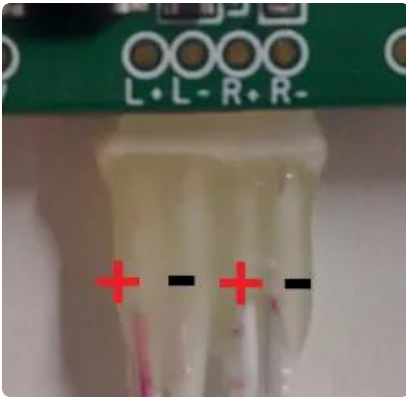
喇叭接口



耳机接口

喇叭为PH2.54 4pin接口，最大支持8Ω@5W；耳机为一路OTMP标准四节耳机座。

喇叭接线参考如下：



10.1 查看声卡设备

```

1  [root@RK356X:/]# aplay -l
2  **** List of PLAYBACK Hardware Devices ****
3  card 0: rockchipdmi [rockchip,hdm], device 0: rockchip,hdm i2s-hifi-0 [r
   rockchip,hdm i2s-hifi-0]
4      Subdevices: 1/1
5      Subdevice #0: subdevice #0
6  card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7      Subdevices: 1/1
8      Subdevice #0: subdevice #0
9  linaro@linaro-alip:~$
```

10.2 播放音频

播放到HDMI：

```

1  aplay -D plughw:0,0 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到Lineout：

不插入耳机，执行以下命令。

```
1 aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到耳机：

插入耳机，执行以下命令。

```
1 aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav
```

10.4 录音

将麦克风连接到J11。

使用arecord工具可以进行录音测试：

```
1 [root@RK356X:/]# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3
4 ^CAborted by signal 中断...
5 linaro@linaro-alip:~$
```

录音完后播放测试：

```
1 [root@RK356X:/]# aplay -D plughw:1,0 ./test.wav
2 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

12、RTC

主板/dev/rtc0为外部RTC（HYM8563），系统默认使用rtc0的时间。

12.1 获取RTC时间

```
▼ Bash |
1 [root@RK356X:/]# sudo hwclock
2 2022-11-10 02:16:23.617474+00:00
3 [root@RK356X:/]#
```

12.2 设置RTC时间

```
▼ Bash |
1 # hwclock -r
2 2022-03-18 12:01:06.991425+08:00
3 # hwclock -w
```

13、开机自启动

默认系统开机会运行/etc/rc.local脚本，将要开机执行的程序放到该脚本中即可。

14、屏幕控制

14.1 背光调节

通过修改/sys/class/backlight/backlight/brightness的值，实现背光的调节，范围取0–255，值越大，亮度越高。

设置亮度为100：

```
▼ Bash |
1 [root@RK356X:/]# chmod a+w /sys/class/backlight/backlight/brightness
2 [root@RK356X:/]# echo 100 > /sys/class/backlight/backlight/brightness
```

15、按键

主板配置了一个ADC按键SW2，对应的设备节点为/dev/input/event1。

系统运行时，短按该按键上报KEY_POWER，并且进入待机状态。

系统待机时，短按该按键，系统恢复正常运行。

系统运行时，长按该按键5秒关机。

系统关机时，短按该按键开机。

16、ADC

主板配置了2路ADC，位于J55的第4、5引脚，分别记作ADC4、ADC5。精度为10位。

16.1 ADC转换方法

$$V = (raw/1024)*1.8v$$

其中raw为对应设备节点读取的值，范围为0-1023。

序号	编号	设备节点
4	SARADC VINU4_	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
5	SARADC VINU5_	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

16.2 测试

以测试ADC2为例，其余ADC测试方法类似。

▼

Bash

```
1 [root@RK356X:/]# cat /sys/bus/iio/devices/iio:device0/in_voltage4_raw
2 1023
```

设备节点读取的raw值为1023，代入到公式计算：

$$V=(1023/1024)*1.8v=1.79v$$

即ADC4输入的电压为1.79v。

17、CAN

CAN位于J44的双排针。共有2路CAN可供使用。



序号	编号	描述
19	CAN1_TX	CAN1
20	CAN1_RX	
17	CAN2_TX	CAN2
18	CAN2_RX	

17.1测试

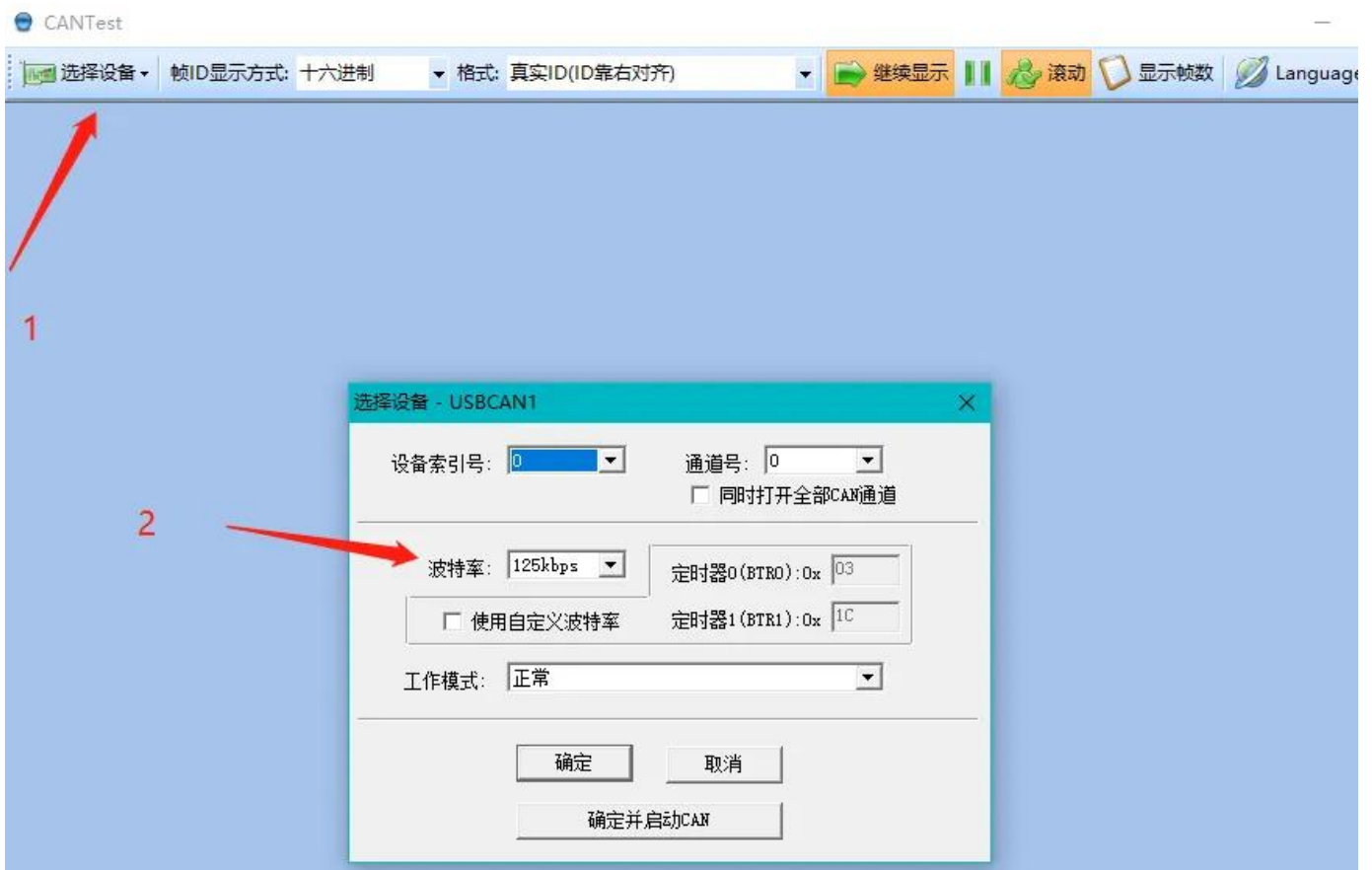
测试需要用USB转CAN工具，通过PC与板上CAN通信。

这里以CAN0为例，其余节点测试方法相同：

```
1 //关掉can
2 ifconfig can0 down
3
4 //配置can通信的波特率
5 ip link set can0 type can bitrate 125000 triple-sampling on
6
7 //开启can通信
8 ifconfig can0 up
9
10 //作为接收端接收数据
11 candump can0
12
13 //作为发送端发送数据
14 cansend can0 5A1#1122334455667788
15
```

PC软件端的接收与发送：

(1) 选择USBCAN1



(2) 启动CAN测试

The screenshot displays the CANTest application window. The top toolbar includes buttons for '继续显示' (Continue Display), '滚动' (Scroll), '显示帧数' (Show Frame Count), and 'Language'. Below this is a tab labeled 'USBCAN1 设备:0 通道:0'. The main toolbar contains '滤波设置' (Filter Settings), '启动' (Start), '停止' (Stop), '关闭' (Close), '定位' (Position), '清空' (Clear), '保存' (Save), '设备操作' (Device Operation), '接收时间标识' (Receive Time Identifier), '隐藏发送帧' (Hide Transmitted Frames), '显示发送帧' (Show Transmitted Frames), 'DBC', and '实时保存' (Real-time Save).

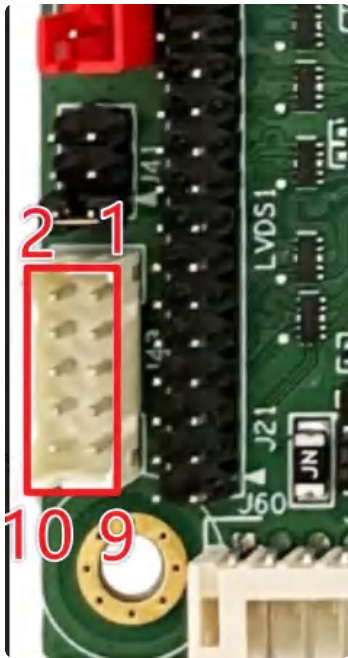
The main data table lists CAN frames with the following columns: 序号 (Serial Number), 传输方向 (Transmission Direction), 时间标识 (Time Identifier), 帧ID (Frame ID), 帧格式 (Frame Format), 帧类型 (Frame Type), 数据长度 (Data Length), and 数据(HEX) (Data(HEX)).

序号	传输方向	时间标识	帧ID	帧格式	帧类型	数据长度	数据(HEX)
00000073	接收	17:54:33.4...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000074	接收	17:54:33.7...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000075	接收	17:54:34.1...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000076	接收	17:54:34.5...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000077	发送	19:15:27.3...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000078	发送	19:15:27.8...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000079	发送	19:15:28.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000080	发送	19:15:28.2...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000081	发送	19:15:28.4...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000082	接收	19:15:34.1...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000083	接收	19:15:34.4...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000084	接收	19:15:35.2...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000085	接收	19:15:35.7...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000086	发送	19:16:07.6...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000087	发送	19:16:08.6...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000088	发送	19:16:08.9...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000089	发送	19:16:09.5...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000090	发送	19:16:16.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000091	发送	19:16:16.2...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000092	发送	19:16:16.4...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000093	发送	19:16:16.6...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000094	发送	19:16:16.8...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000095	发送	19:16:17.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000096	发送	19:16:40.3...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000097	发送	19:16:41.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000098	发送	19:16:41.2...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000099	发送	19:16:41.7...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000100	发送	19:16:47.5...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000101	发送	19:16:47.7...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000102	发送	19:16:47.9...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000103	发送	19:16:48.1...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000104	发送	19:16:48.3...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000105	接收	19:16:54.9...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000106	接收	19:16:55.4...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000107	接收	19:16:56.1...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000108	接收	19:16:56.7...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88

The bottom section, titled '基本操作' (Basic Operation), contains configuration options for sending frames. It includes a '发送方式' (Send Method) dropdown set to '正常发送' (Normal Send), radio buttons for '每次发送单帧' (Send single frame each time) and '每次发送 10 帧' (Send 10 frames each time), and a checkbox for '帧ID每发送一帧递增' (Increment frame ID by one frame sent). Below these are fields for '帧类型' (Frame Type) set to '标准帧' (Standard Frame), '帧ID(HEX):' (Frame ID(HEX):) set to '00000000', '数据(HEX):' (Data(HEX):) set to '00 01 02 03 04 05 06 07', '帧格式' (Frame Format) set to '数据帧' (Data Frame), '发送次数' (Send count) set to '1', and '每次发送间隔(ms):' (Send interval(ms):) set to '0'. '发送' (Send) and '停止' (Stop) buttons are at the bottom right.

板端发送过来的数据可以在CANtest上打印出来。

18、扩展IO



扩展接口默认配置为GPIO功能，默认作为in，可通过应用程序配置GPIO方向为in或者out。

序号	GPIO号	接口	说明
1	96	方向： /sys/class/gpio/gpio96/direction 状态： /sys/class/gpio/gpio96/value	默认方向为输入
2	13	方向： /sys/class/gpio/gpio13/direction 状态： /sys/class/gpio/gpio13/value	默认方向为输入
3	94	方向： /sys/class/gpio/gpio94/direction 状态： /sys/class/gpio/gpio94/value	默认方向为输入
4	14	方向： /sys/class/gpio/gpio14/direction 状态： /sys/class/gpio/gpio14/value	默认方向为输入
5	95	方向： /sys/class/gpio/gpio95/direction 状态： /sys/class/gpio/gpio95/value	默认方向为输入
6	146	方向： /sys/class/gpio/gpio146/direction 状态： /sys/class/gpio/gpio146/value	默认方向为输入
7	93	方向： /sys/class/gpio/gpio93/direction 状态： /sys/class/gpio/gpio93/value	默认方向为输入

8	147	方向: /sys/class/gpio/gpio147/direction 状态: /sys/class/gpio/gpio147/value	默认方向为输入
9	VCC	供电输出, 可通过修改电阻变更3.3V或5V输出	默认3.3V
10	GND	地	

19.1 测试

以GPIO96为例, 输入输出方向控制方法如下

1. 设置方向为输入, 并读取接口电平

▼

Shell

```

1  # 设置方向为输入
2  echo in > /sys/class/gpio/gpio96/direction
3  # 读取IO口电平值
4  cat /sys/class/gpio/gpio96/value

```

2. 设置方向为输出, 设置输出电平

▼

Shell

```

1  # 设置方向为输出
2  echo out > /sys/class/gpio/gpio96/direction
3  # 设置IO口输出高电平
4  echo 1 > /sys/class/gpio/gpio96/value
5  # 设置IO口输出低电平
6  echo 0 > /sys/class/gpio/gpio96/value

```