

# IDO-EVB3020 Linux Buildroot开发手册

---

## 1. 系统用户名及密码

## 2. 调试口使用

### 2.1 调试串口

### 2.2 ADB调试

## 3. 串口测试

### 3.1 功能说明

### 3.2 控电方法

### 3.3 通信测试

## 4. 以太网

### 静态IP设置

## 5.WIFI使用

## 6. Bluetooth使用

## 7.4G

## 8. USB

### 8.1 USB OTG

### 8.2 USB HOST

## 9. MIPI CSI Camera

## 10. CAN

## 11. DO

## 12.喇叭/耳机

## 13.模拟MIC

## 14. ADC 按键

## 15. ADC IN

---

# IDO-EVB3020 Linux Buildroot系统使用说明

深圳触觉智能科技有限公司

[www.industio.cn](http://www.industio.cn)

## 文档修订历史

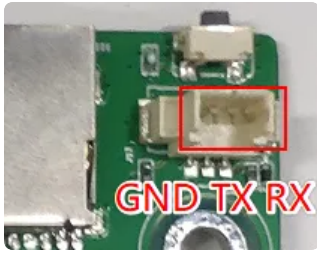
版本	修订内容	修订	审核	日期
V1.0	创建文档	FuYingz he		2023/02/ 24

## 1. 系统用户名及密码

用户	密码
root	rockchip

## 2. 调试口使用

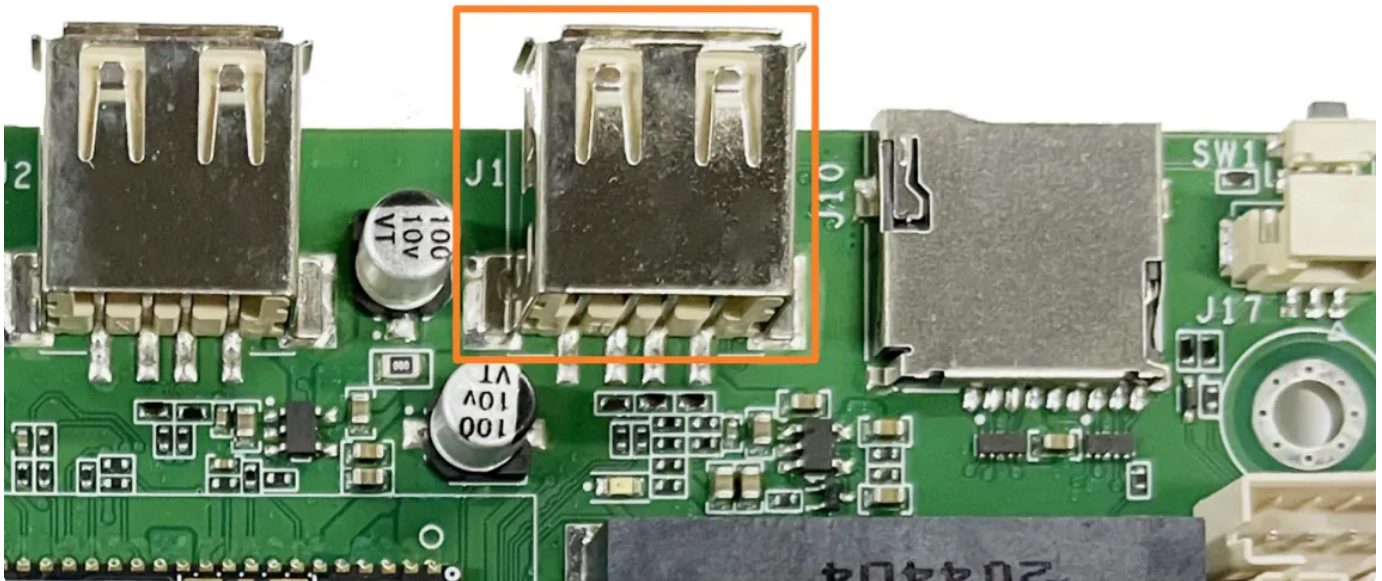
### 2.1 调试串口



调试串口为 TTL 电平，主板接口为 MX1.25 接线端子，使用 USB 转串口模块连接 PC 调试终端。  
串口参数：波特率 1500000、数据位 8bit、无校验位、停止位 1bit。

Baud rate:	1500000	Flow control
Data bits:	8	<input type="checkbox"/> DTR/DSR
Parity:	None	<input type="checkbox"/> RTS/CTS
Stop bits:	1	<input type="checkbox"/> XON/XOFF

### 2.2 ADB调试



上图橙色框内的USB接口为支持OTG模式切换，在系统上电前，使用双公头 USB 数据线连接主板和 PC 端的USB接口，在PC终端识别到 ADB 设备，即可使用 adb shell 调试。

- 1 #查看是否有adb设备
- 2 adb devices
- 3 #通过adb进入系统
- 4 adb shell

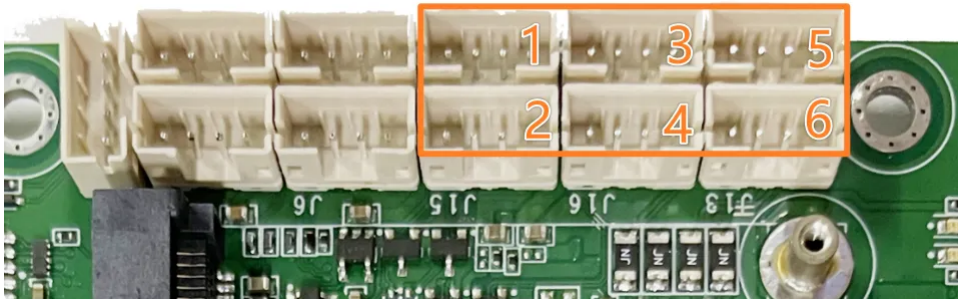
```

adb.exe devices
List of devices attached
329f1a2a423513ef    device

adb.exe shell
[root@px30_64:/]# ls
bin          lib          mnt         root        timestamp
busybox.config  lib64       oem         run         tmp
data        linuxrc     opt         sbin        udisk
dev         lost+found  ov5648_default_default.xml.bin  sdcard     userdata
etc         media       proc        sys         usr
init       misc        rockchip_test  system     var
[root@px30_64:/]#
    
```

### 3. 串口测试

#### 3.1 功能说明



开发共有四路RS232和两路RS485，均可通过修改硬件贴片变更为TTL电平，接口设备节点如下表所示：

序号	接口位置	电平类型	串口设备节点
1	J19	RS232	/dev/ttyS0
2	J15	RS232	/dev/ttyS1
3	J18	RS232	/dev/ttyS4
4	J16	RS232	/dev/ttyS5

5	J14	RS485	/dev/ttyS3
6	J13	RS485	/dev/ttyS2 (默认此接口不开启, 配置为调试串口)

## 3.2 控电方法

其中RS485接口可以通过软件控制VCC供电, 供电接口如下所示:

序号	接口位置	控电节点
1	J13	/sys/class/leds/rs485_vout1/brightness
2	J14	/sys/class/leds/rs485_vout2/brightness

以J13端口的VCC供电控制为例, 控制方法如下

```

▼ Bash |
1 #断电
2 echo 0 > /sys/class/leds/rs485_vout1/brightness
3 #开启
4 echo 1 > /sys/class/leds/rs485_vout1/brightness

```

## 3.3 通信测试

串口均可以使用microcom工具进行测试, 使用microcom 打开/dev/ttyS0, 并设置波特率为115200。

```

▼ Bash |
1 microcom -s 115200 /dev/ttyS0

```

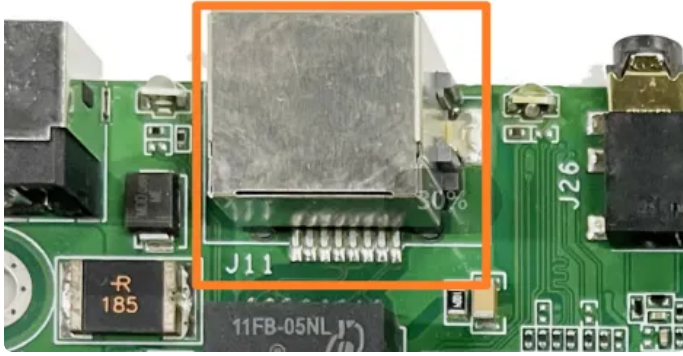
```

[root@px30_64:~]# microcom -s 115200 /dev/ttyS0
[ 84.841724] of_dma_request_slave_channel: dma-names property of node '/serial@ff030000' missing or empty
[ 84.842616] ttyS0 - failed to request DMA, use interrupt mode

```

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。Ctrl+x键停止测试。

## 4. 以太网



主板具有一路自适应 10/100Mbps 以太网接口。

网络设备节点: eth0

IP设置方式: 默认IP地址分配方式为 dhcp, 只需要将以太网接口连接路由器即可为主板动态分配 IP 地址。

```
[root@px30_64:/]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 56:E0:64:9C:A7:11
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0  errors:0  dropped:0  overruns:0  frame:0
          TX packets:4  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:603 (603.0 B)
          Interrupt:29
```

## 静态IP设置

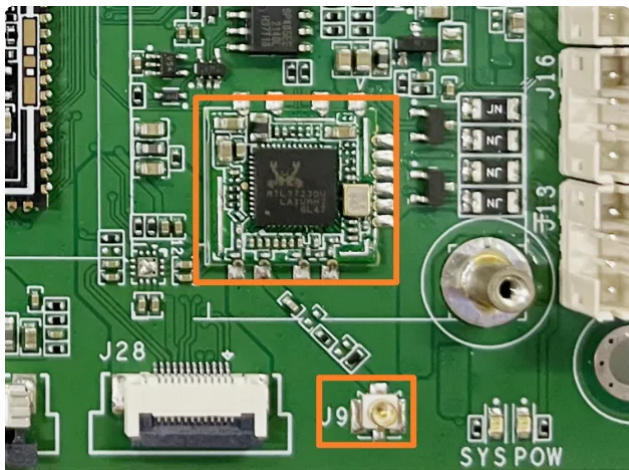
修改/etc/network/interfaces, 在文件中添加如下内容

```
▼ | Bash |
1  # interface file auto-generated by buildroot
2
3  auto lo
4  iface lo inet loopback
5
6  auto eth0
7  iface eth0 inet static
8  address 192.168.0.234
9  netmask 255.255.255.0
10 gateway 192.168.0.1
11 dns-nameservers 114.114.114.114
```

重启系统或者重启network服务, 即可让配置生效

```
1 /etc/init.d/S40network restart
```

## 5.WIFI使用



主板板载USB WiFi 模块为RTL8723DU，使用WiFi时需要连接好WiFi天线。

网络设备节点：wlan0

系统共有两个与WIFI相关的启动服务，其中/etc/init.d/S36load\_wifi\_modules用于挂载WIFI驱动，/etc/init.d/S80wifireconnect用于WIFI联网。修改/userdata/cfg/wpa\_supplicant.conf，填写正确的热点账号和密码

```
1 # 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 }
```

重启后，/etc/init.d/S80wifireconnect自启动程序将会连接热点。

```
[root@px30_64:/]# ifconfig wlan0
wlan0    Link encap:Ethernet  HWaddr F0:B0:40:ED:34:CB
         inet addr:192.168.1.187  Bcast:192.168.1.255  Mask:255.255.255.0
         inet6 addr: fe80::d8dd:332d:a365:63ec/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:30  errors:0  dropped:0  overruns:0  frame:0
         TX packets:27  errors:0  dropped:0  overruns:0  carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:6202 (6.0 KiB)  TX bytes:3255 (3.1 KiB)
```

## 6. Bluetooth使用

执行/usr/bin/bt\_init.sh脚本，开启蓝牙供电并加载蓝牙驱动，蓝牙功能开启后，将产生hci0节点。

```
[root@px30_64:/]# hciconfig -a
hci0:    Type: Primary  Bus: USB
         BD Address: F0:B0:40:EE:46:3E  ACL MTU: 1021:8  SCO MTU: 255:12
         UP RUNNING
         RX bytes:861 acl:0 sco:0 events:33 errors:0
         TX bytes:378 acl:0 sco:0 commands:33 errors:0
         Features: 0xff 0xff 0xff 0xfe 0xdb 0xfd 0x7b 0x87
         Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
         Link policy: RSWITCH HOLD SNIFF PARK
         Link mode: SLAVE ACCEPT
         Name: 'RTK BT_4.1'
         Class: 0x000000
         Service Classes: Unspecified
         Device Class: Miscellaneous,
 [ 427.033494] rtk_btcoex: BTCOEX hci_rev 0x8299
         HCI Version: 4.1 (0x7)  Revision: 0x8299
 [ 427.033951] rtk_btcoex: BTCOEX lmp_subver 0x37de
         LMP Version: 4.1 (0x7)  Subversion: 0x37de
         Manufacturer: Realtek Semiconductor Corporation (93)
```

使用hcitool测试蓝牙扫描功能

```
▼ | Bash |
1  hcitool scan
```

```
[root@px30_64:/]# hcitool scan
Scanning ...
 [ 767.194543] rtk_btcoex: hci (periodic)inq start
 [ 777.438990] rtk_btcoex: inquiry complete
         38:F9:D3:7A:40:18      JiangLaiMBP
         40:79:72:46:D0:27      SOAIY SR10
         38:00:25:C4:CD:B2      LAPTOP-7MP4TLOF
         B0:E2:35:0D:4A:13      Redmi 9A
         50:A0:09:67:AE:BE      MITV-7AEBE
```

## 7.4G

系统默认适配了 EC20 4G模块，支持自动拨号上网。使用 4G 功能前，需要先在板载 MINI PCI-E 接口插入 EC20 模块，在卡槽插入 SIM 卡，并连接好 4G 天线以保证信号的稳定。

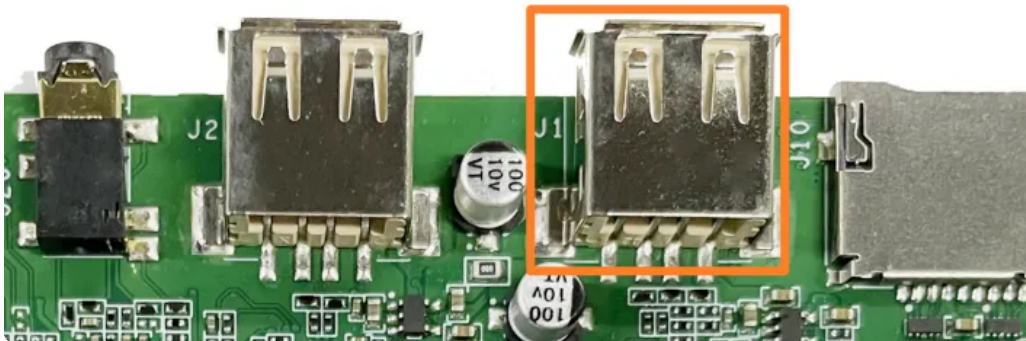


系统上电，开机自启动程序检测到有EC20插上之后执行quectel-CM进行拨号，拨号成功后wwan0将会分配到ip地址。

```
[root@px30_64:/]# ifconfig wwan0
wwan0  Link encap:Ethernet  HWaddr 3A:01:20:F9:B2:08
        inet addr:10.217.130.198  Mask:255.255.255.252
        inet6 addr: fe80::2731:87d7:13e4:3fd2/64  Scope:Link
        UP RUNNING NOARP  MTU:1500  Metric:1
        RX packets:37  errors:0  dropped:0  overruns:0  frame:0
        TX packets:48  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:4266 (4.1 KiB)  TX bytes:4830 (4.7 KiB)
```

## 8. USB

### 8.1 USB OTG

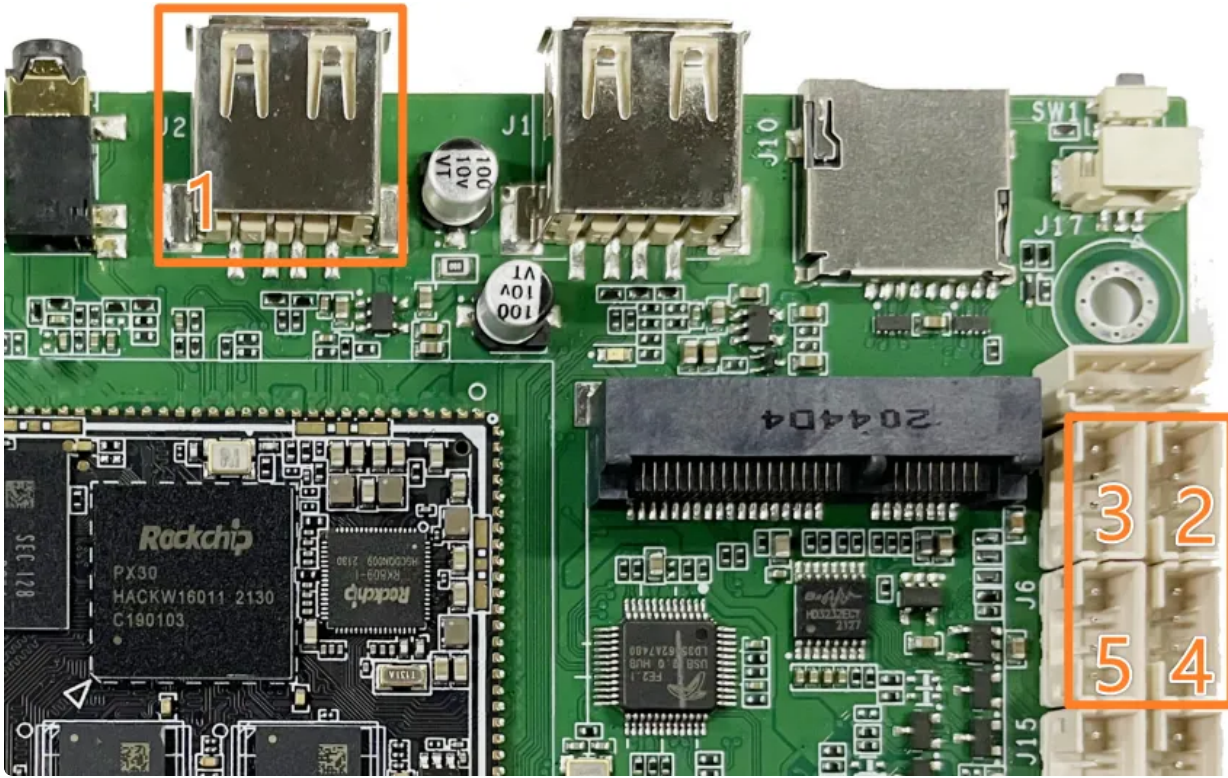


当 OTG 接口在上电时已使用 USB 数据线连接 PC 端的 USB 口，此接口会自动切换为 Device 模式，在此模式下可通过 ADB 的方式来调试主板。当USB OTG 接口没有链接USB数据线或连接U盘设备时，开始自动切换为 Host 模式。USB OTG 执行应用层调用写设备节点的方式控制接口的 Device 和 Host 模式，设置方法如下：

```
▼ Bash |
```

- 1 #设置为Host模式
- 2 echo HOST > /dev/otg\_mode
- 3 #设置为Device模式
- 4 echo DEVICE > /dev/otg\_mode

### 8.2 USB HOST



主板共有5路USB HOST 2.0接口，启动一路为Type A 接口，另外四路为 PH2.0-4 端子。支持挂载U盘，USB摄像头、USB鼠标等标准USB设备。

主板每一路USB HOST均可通过软件控制接口的供电，接口对应列表如下：

序号	位置	接口
1	TYPE-A	/sys/class/leds/usb1_pwr/brightness
2	J5	/sys/class/leds/usb2_pwr/brightness
3	J3	/sys/class/leds/usb3_pwr/brightness
4	J4	/sys/class/leds/usb4_pwr/brightness
5	J6	/sys/class/leds/usb5_pwr/brightness

以Type A接口USB HOST 供电控制为例，控制方法如下：

```

▼
                                                                    Bash |
1  #开电
2  echo 1 > /sys/class/leds/usb1_pwr/brightness
3  #断电
4  echo 0 > /sys/class/leds/usb1_pwr/brightness

```

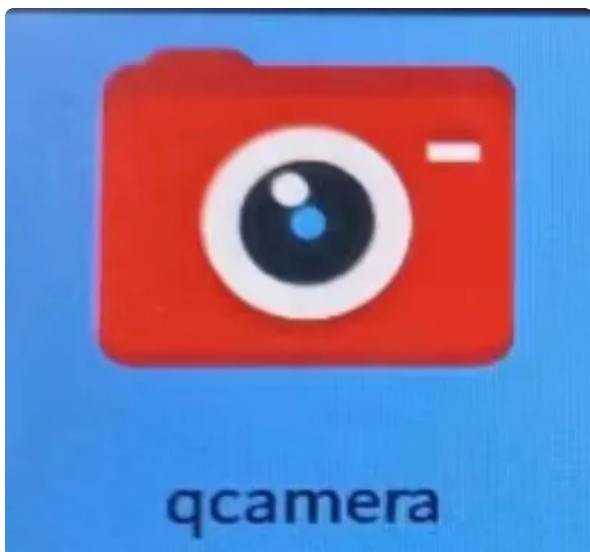
## 9. MIPI CSI Camera



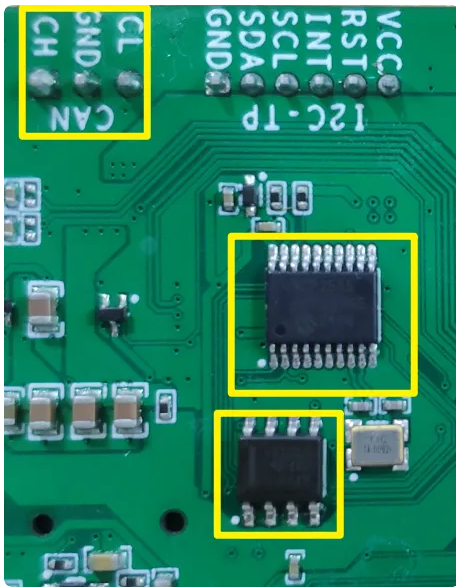
统默认支持OV5648 MIPI摄像头模组，接口位置如上图所示。

摄像头节点为： /dev/video0

可使用qcamera 软件打开摄像头。



## 10. CAN



主板使用 MCP2515-I IC实现 SPI转CAN，在使用之前先确认主板硬件电路是否贴有此MCP2515和CAN收发器芯片，芯片位置如上图所示。

默认 can0 接口状态为 down，需要设置 can 参数并执行 up 操作后才能执行收发。测试时可 将主板的 can 接口与另外一块主板 can 接口互连，或者连接 USB 转 can 工具，收发双方 设置相同的波特率等参数来测试接口功能。

关闭

```
▼ Bash |  
1 ifconfig can0 down
```

设置参数，波特率 125000（最大支持 1Mbps）

```
▼ Bash |  
1 ip link set can0 type can bitrate 125000 triple-sampling on
```

开启

```
▼ Bash |  
1 ifconfig can0 up
```

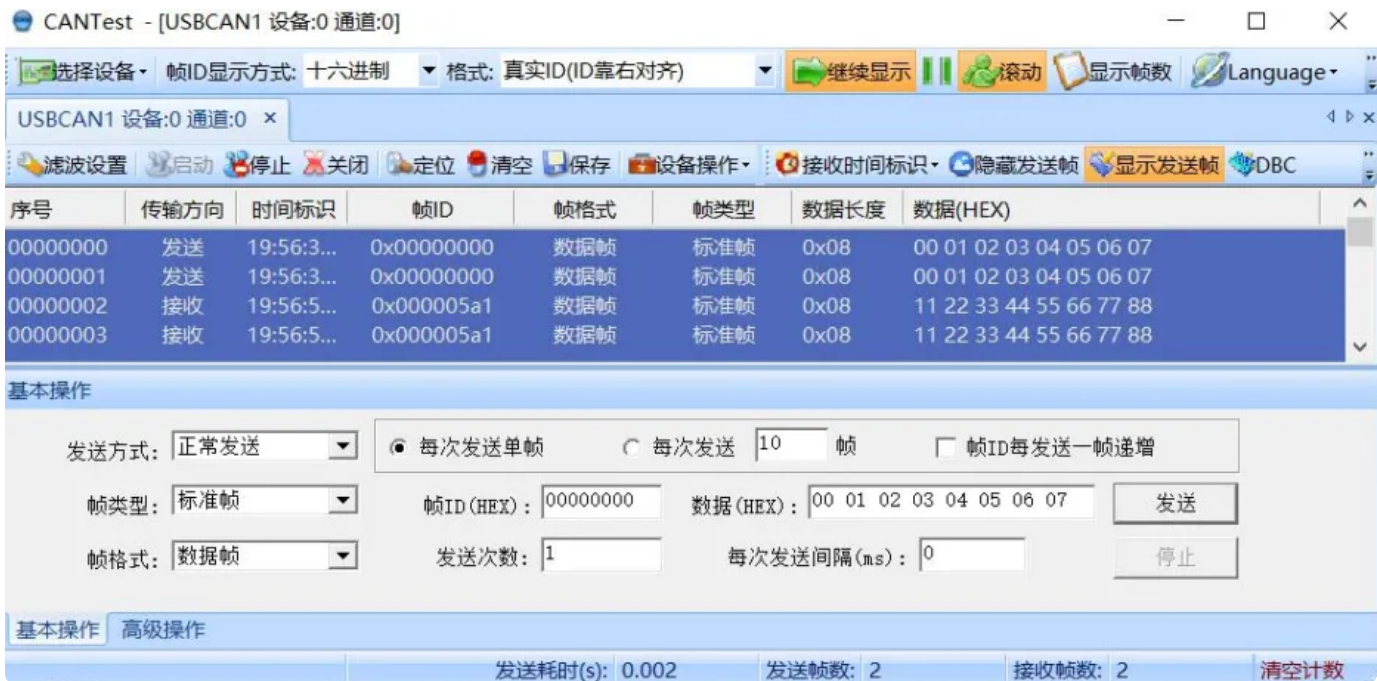
接收

```
▼ Bash |
1 candump can0
```

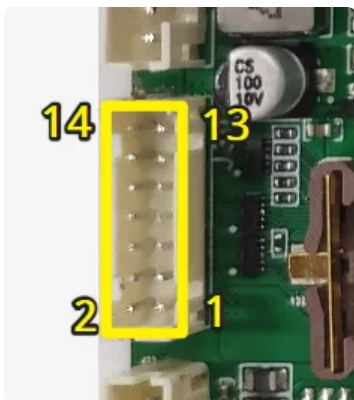
发送

```
▼ Bash |
1 cansend can0 5A1#1122334455667788
```

PC端使用CANTest软件+USB转CAN模块，实现CAN数据的收发。



## 11. DO



开发DO接口如上图所示，接口定义见下表：

序号	IO	控制节点
1	VCC 3.3V	3.3V供电
2	GND	系统地
3 ~ 14	DO1 -DO11	/sys/class/leds/do1/brightness ~ /sys/class/leds/do11/brightness

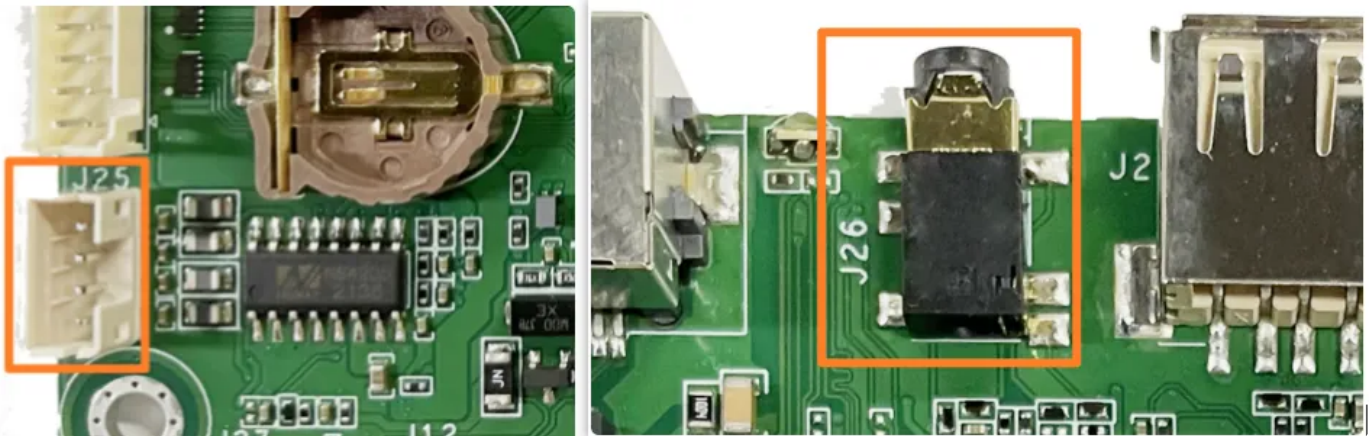
以DO1为例，控制DO1输出高低电平的方法如下：

```

▼ Bash |
1 #输出高电平
2 echo 1 > /sys/class/leds/do1/brightness
3 #输出低电平
4 echo 0 > /sys/class/leds/do1/brightness

```

## 12.喇叭/耳机



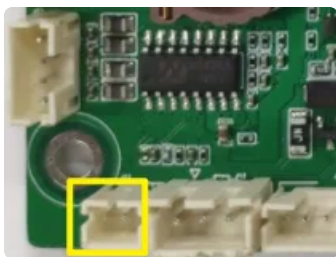
喇叭和耳机接口位置如上图所示  
音频播放测试

```

▼ Bash |
1 aplay music.wav

```

## 13.模拟MIC



录音测试

```
Bash |
1 arecord -Dhw:0,0 -c 2 -d 5 -r 44100 -f S16_LE ./record.wav
```

## 14. ADC 按键

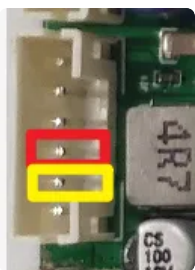


ADC按键在系统启动后可作为普通按键使用。

设备节点: /dev/input/event2

键值: KEY\_VOLUMEUP

## 15. ADC IN



主板引出两路10bit有效位的数模转化器，参考电源为1.8V，读取接口如下

序号	位置	接口
1	红色	cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
2	黄色	cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw

