

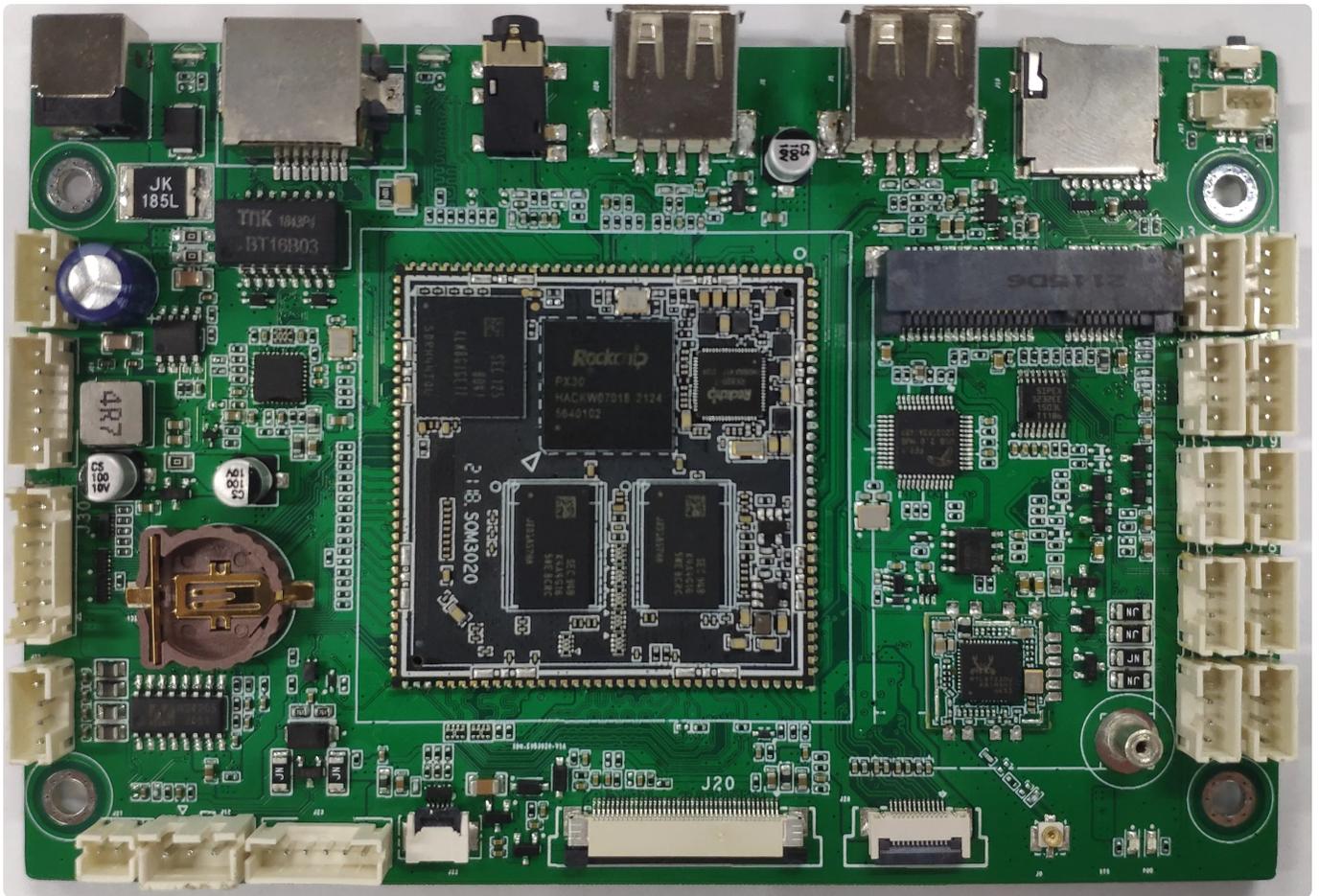
# IDO-EVB3020 Debain系统使用手册

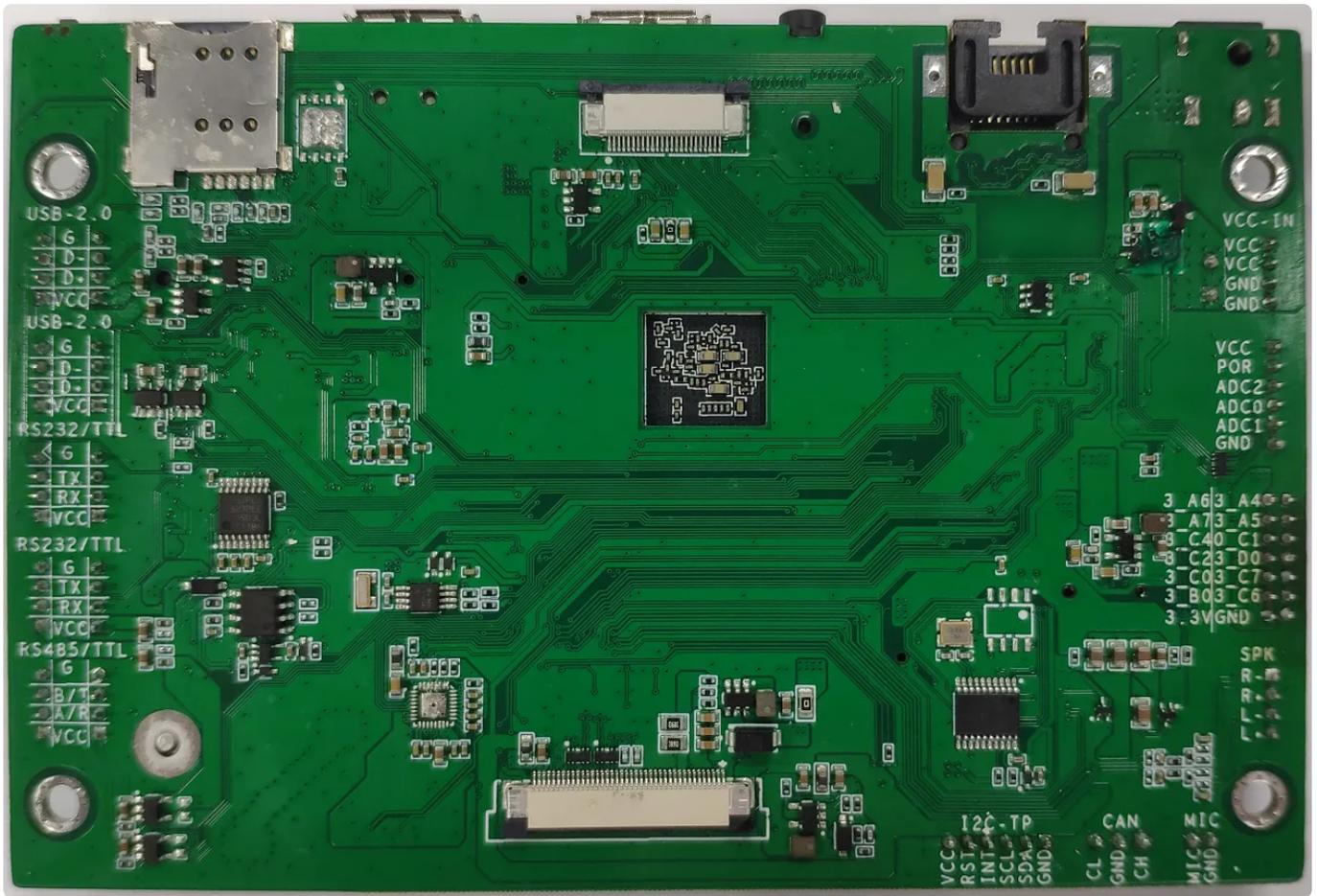
---

- 1 硬件资源概况
  - 1.1 开发板照片
  - 1.2 硬件资源及设备节点
- 2 电源及调试接口
  - 2.1 供电电源
  - 2.2 调试串口
  - 2.3 SSH 调试
  - 2.4 程序下载
- 3 功能测试及接口使用方法
  - 3.1 Ethernet
  - 3.2 WiFi
  - 3.3 蓝牙
  - 3.4 EC20
  - 3.5 USB OTG
  - 3.6 USB HOST
  - 3.7 MIPI 摄像头
  - 3.8 RS232/RS485
  - 3.9 RTC
  - 3.10 ADC 按键
  - 3.11 ADC IN
  - 3.12 DO
  - 3.13 CAN
  - 3.14 喇叭
  - 3.15 模拟MIC
  - 3.16 PDM

## 1 硬件资源概况

### 1.1 开发板照片





## 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	4.4.189	
2	文件系统	Debian / Buildroot	
3	内存	DDR4 (1GB/2GB选配)	
4	存储	eMMC 5.1 (8GB/16GB/32GB/128GB 可选配)	/dev/mmcblk1
5	供电	12V DC @2A	
6	显示	MIPI 、LVDS	
7	USB OTG	USB 2.0 OTG X 1	
8	USB HOST	USB 2.0 Type-A X 1 USB 2.0 PH-4 X 1	

9	TF Card	Micro SD 卡	/dev/mmcblk0
10	以太网	自适应10/100/100Mbps以太网	eth0
11	WiFi/BT	RT8723DU	wlan0、hci0
12	4G	支持MiniPCIE座子接口的4G模组, EC20	
13	扬声器	双声道 4Ω3W 扬声器	
14	耳机	3.5mm 耳机	
15	CAN	spi转CAN X 1	can0
16	摄像头	OV5648 mipi摄像头	
17	RTC	HYM8563	/dev/rtc0
18	RS232	RS232 X 4	/dev/ttyS0 /dev/ttyS1 /dev/ttyS4 /dev/ttyS5
19	RS485	RS485 X 2	/dev/ttyS2 (调试串口和RS485 2选1) /dev/ttyS3
20	调试串口	TTL X 1	/dev/ttyFIQ0
21	ADC按键	1路	/dev/input/event1
22	ADC IN	2路	/sys/bus/iio/devices/iio\:device0/in_voltage0_raw /sys/bus/iio/devices/iio\:device0/in_voltage1_raw

## 2 电源及调试接口

### 2.1 供电电源



开发板12V @2A DC供电，有 DC 母座和 PH2.0-4 排针两种电源输入接口，接口位置如上图所示。

## 2.2 调试串口



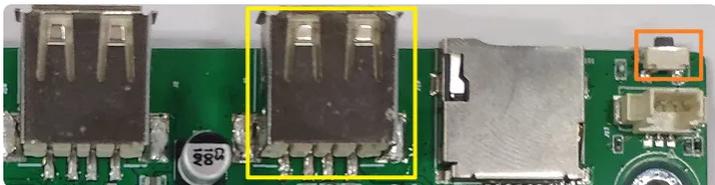
调试串口为 TTL 电平，开发板接口为 MX1.25 接线端子，使用 USB 转串口模块连接 PC 调试终端。  
串口参数：波特率 115200、数据位 8bit、无校验位、停止位 1bit。

Port:	COM8	Flow control
Baud rate:	115200	<input type="checkbox"/> DTR/DSR
Data bits:	8	<input type="checkbox"/> RTS/CTS
Parity:	None	<input type="checkbox"/> XON/XOFF
Stop bits:	1	

## 2.3 SSH 调试

用户名	密码
linaro	linaro

## 2.4 程序下载



上图橙色框内为 recovery 按键，黄色框内为 OTG 接口。开发下载固件需要将设备切换为 Loader 模式，Loader 模式的操作流程方法如下：

1. 断开开发板供电电源；
2. 使用数据线连接开发板和PC端的USB接口；
3. 按下 recovery 按键；
4. 开发板供电；

以上步骤正常操作，在烧录工具界面下端会显示“发现一个LOADER设备”，点击界面的“固件”按键，在系统文件资源管理器中选择烧录固件；加载完成后，点击“升级”按键开始下载程序，烧录完成开发板将会重新启动。



## 3 功能测试及接口使用方法

### 3.1 Ethernet



开发板具有一路自适应 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开发板动态分配 IP 地址。

### 3.2 WiFi

开发板板载USB WiFi 模块为RTL8723DU。

网络设备节点: wlan0

联网方法: Debian系统可在界面上配置SSID和密码连接附近的WiFi路由。

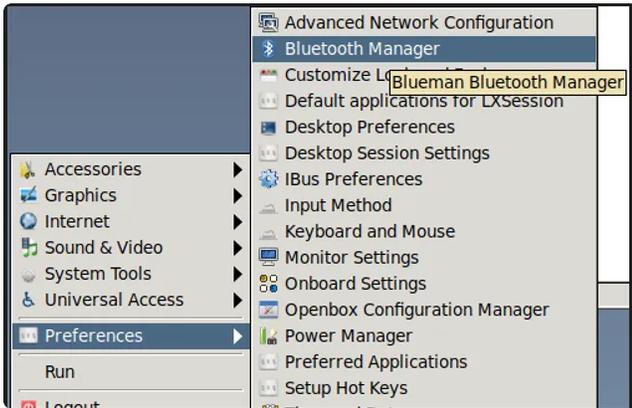
### 3.3 蓝牙

开发板板载USB WiFi+蓝牙二合一模块RTL8723DU

设备节点: hci0

蓝牙标准: 蓝牙4.2双模

连接方法: Debian系统可在界面上可以搜索附近的蓝牙并连接, 软件如下图所示:



### 3.4 EC20

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。

系统已适配有EC20 网络连接管理工具quectel-CM的拨号服务, 服务默认为enable状态, 并上电开机自动运行。

服务关闭和重启方法如下:

关闭服务:

```
Bash |
1 #停止拨号服务
2 systemctl stop ec20-dialup.service
3 #停止开机自启动
4 systemctl disable ec20-dialup.service
```

开启服务:

```
▼ | Bash |
1 #开启拨号服务
2 systemctl start ec20-dialup.service
3 #开启开机自启动
4 systemctl enable ec20-dialup.service
```

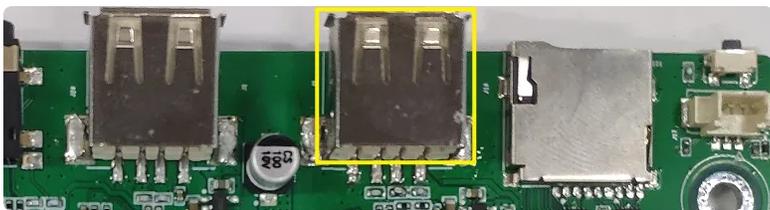
服务重启:

```
▼ | Bash |
1 systemctl reload ec20-dialup.service
```

正常拨号成功后, usb0将会分配到ip地址, 此时可以测试是否能够正常ping通外网, 如: ping www.baidu.com -I usb0 ,可以只对4G功能测试是否正常。

```
usb0: flags=4291<UP,BROADCAST,RUNNING,NOARP,MULTICAST> mtu 1500
inet 10.124.213.207 netmask 255.255.255.224 broadcast 10.124.213.223
inet6 fe80::eccf:d8ba:7f6c:7964 prefixlen 64 scopeid 0x20<link>
ether fa:24:d9:83:e9:ca txqueuelen 1000 (Ethernet)
RX packets 4 bytes 1258 (1.2 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 45 bytes 6072 (5.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

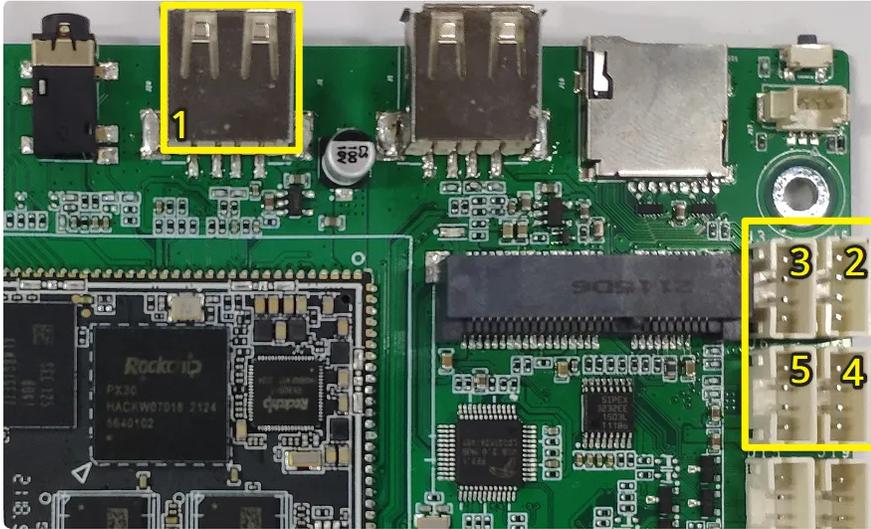
### 3.5 USB OTG



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 的方式来调试开发板。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
```

### 3.6 USB HOST



开发板共有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。

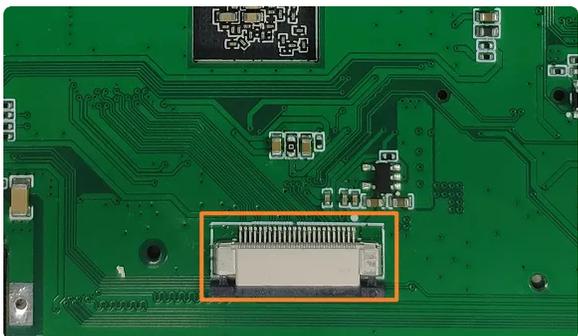
开发板每一路USB HOST均可通过软件控制接口的供电，接口对应列表如下：

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
```

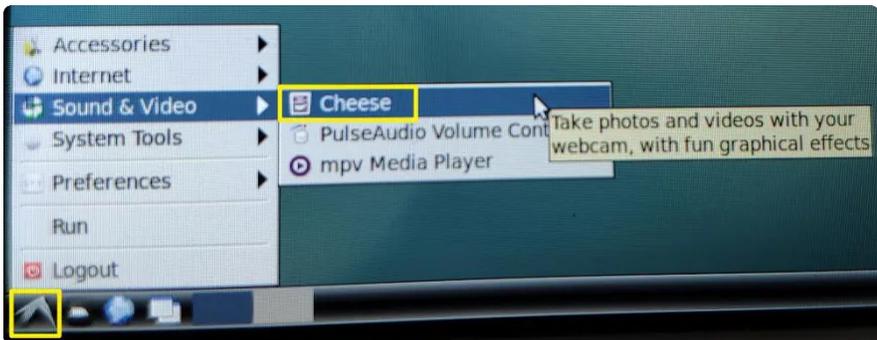
### 3.7 MIPI 摄像头



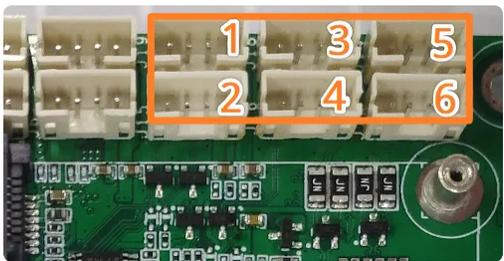
系统默认支持OV5648 MIPI摄像头模组，接口位置如上图所示。

摄像头节点为： /dev/video0

摄像头可以使用系统自带的软件 Cheese 打开摄像头测试。软件的位置如下图所示：



### 3.8 RS232/RS485



开发共有4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	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)
---	-----	-------	--------------------------------

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

```

### 3.9 RTC

RTC时间读取和设置方法如下

读取RTC硬件时间

```

▼ Bash |
1 # hwclock -r
2 2021-10-26 13:45:31.969877+0000

```

设置系统时间

```

▼ Bash |
1 # date -s "2021-10-26 21:40:00"
2 Tue Oct 26 21:40:00 UTC 2021

```

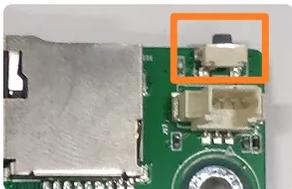
将时间写入RTC

```

▼ Bash |
1 # hwclock -w

```

### 3.10 ADC 按键

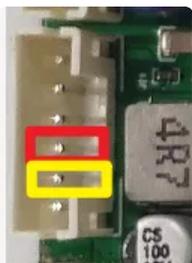


ADC按键在系统启动后可作为普通按键使用。

设备节点: /dev/input/event2

键值: KEY\_VOLUMEUP

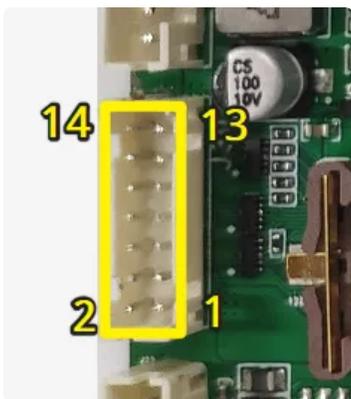
### 3.11 ADC IN



开发板引出两路10bit有效位的数模转换器，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

### 3.12 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 ~ 13	DO1 -DO12	/sys/class/leds/do1/brightness ~ /sys/class/leds/do12/brightness

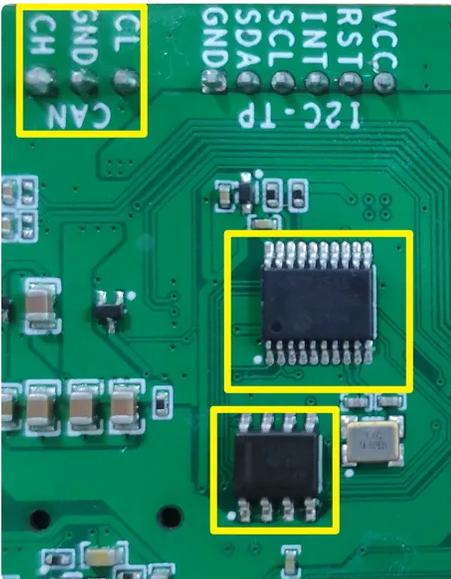
以DO1为例，控制DO1输出高低电平的方法如下：

```

▼
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

```

### 3.13 CAN



开发板使用 MCP2515-I IC实现 SPI转CAN，在使用之前先确认开发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 将开发板的 can 接口与另外一块开发板 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
```

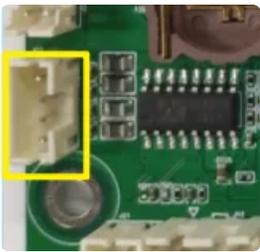
发送

```
▼ | 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。



### 3.14 喇叭



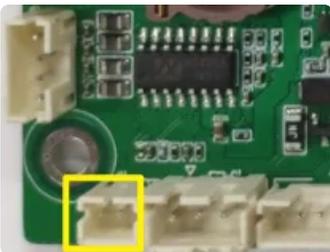
喇叭接口位置如上图所示  
音频播放测试

```

▼ | Bash |
1 aplay music.wav

```

### 3.15 模拟MIC

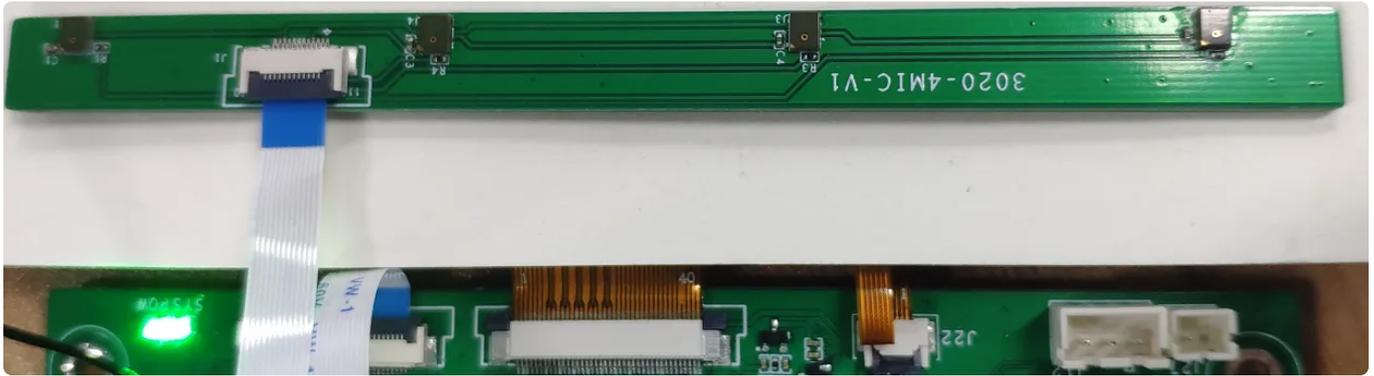


录音测试

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

## 3.16 PDM

PDM功能硬件上需要连接外部扩展4MIC阵列



录音测试

```
1 arecord -Dhw:0,1 -c 4 -d 5 -r 44100 -f S16_LE ./record.wav
```