Purple-Pi-OH RK3566 Ubuntu系统使用说明

- 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、功能及接口介绍
 - 2.1 串口调试
 - 1.2 ADB调试
 - 1.3 ssh调试
- $2 \ USB$
 - 3.1 电源控制
- 3、TF CARD
- 4、以太网
 - 4.1 查看以太网IP地址
 - 4.1.1 使用命令查看
 - 4.2 设置以太网临时IP地址
 - 4.2.1 使用命令设置
 - 4.2.2 在桌面上设置
 - 4.3 设置以太网永久静态IP
- 5 WiFi
 - 5.1 连接热点
 - 5.1.1 方式-
 - 5.1.2 方式二

 - 5.2 查看WiFi的IP地址
- 6、蓝牙
 - 6.1 连接蓝牙设备
 - 6.1.1 方式一
 - 6.1.2 方式二
- 7、音频
 - 7.1 查看声卡设备

7.2 播放音频

7.3 音量的调节

7.3.1 方式一

7.3.2 方式二

7.4 录音

8、摄像头

8.1 测试

8.1.1 测试摄像头是否存在

8.1.3抓取视频

9 RTC

9.1 获取RTC时间

9.2 设置RTC时间

9.3 RTC定时开机

10、开机自启动

11、屏幕控制

11.1 背光调节

11.2 屏幕旋转

11.2.1 临时旋转

11.2.2 永久旋转

12、按键

13、双排针功能及接口

13.1 双排针Pin功能定义

13.1.1 UART 13.1.2 SPI

13.1.3 120

13.1.4 GPIO





文档修订历史

版本	修订内容	修订	审核	日期
----	------	----	----	----

V1.0	创建文档;	谭文学		2023/03/17
V1.1	修改USB1描述为USB-HOST2.0	谭文学		2023/03/28
	验证	何伟聪		2023/04/17
				>
			17	
		XIX	U	
			\mathbf{O}	
		TX		

1、硬件资源概况

1.1 主板照片



图一 Purple Pi OH正面实物图

- KANNER /



图二 Purple Pi OH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	Debian0–Desktop	
3	内存	LPDDR4(1G/2G/4G选配)	
4	存储	eMMC5.1(8GB/16GB/32GB选配)	
5	供电	5V@2A	
6	显示	HDM	
		MIPI	
7	USB OTG	USB OTG Type-C	

8	USB HOST	USB3.0 HOST(Type-A) X 1	
		USB2.0 HOST(Type-A) X 3	
9	TF Card	TF Card x 1	
10	以太网	千兆自适应以太网 × 1	eth0
11	WIFI/BT	AM-NM372SM 2.4G	wlan0 、hci0
		AW-CM256SM 2.4G/5G	
12	耳机	2.5mm 国标	
13	Camera	OV5648、OV8858	117
14	串口	TTL x 1	
15	调试串口	TTL x 1	
16	RTC	HYM8563 x 1	· O *
17	系统指示灯	x1	
18	ADC按键		

2、功能及接口介绍

2.1 串口调试

串口调试端口位于J5,通信参数为15000008N1,电平状态为TTL电平。



串口登录默认以root用户登录,且没有设置密码。

•				120	K	X		Bas	h C 复制代码
1 2 3 4 5	root@ido app.log bin boot root@ido	:/# ls data dev etc :/#	home lib log	lost+found media mnt	opt proc root	run sbin sdcard	srv sys system	tmp udisk usr	var vendor

1.2 ADB调试

ADB调试端口位于**J7**,将SW4切换到**ON**档,使用USB–TYPEC,连接主板和电脑,即可在电脑上使用adb调试:



C:\Users	C:\Users\ronnie\Downloads>adb_shell							
adb serv	er is	out of	date. kill	ing				
* daemon	start	ed suc	cessfully *					
root@ido	:/# 1s							
ls								
app.log	data	home	lost+found	opt	run	srv	tmp	var
bin	dev	1ib	media	proc	sbin	sys	udisk	vendor
boot	etc	log	mnt	root	sdcard	system	usr	
root@ido	:/#							

1.3 ssh调试

系统默认ssh账号和密码为 ido @ 123456。



2、USB

主板共配置了4个USB接口,分别为USB1-4。其中USB1为OTG,USB2-4为HOST。



编号	类型	位置
usb1	host-2.0	J9 (下)
usb2	host-3.0	J9 (上)
usb3	host-2.0	J10 (下)
usb4	host-2.0	J10 (上)

注:如果要使用USB1,需要将SW4切换至OFF档,且adb功能无法使用。

当USB1-4插入U盘后,会自动挂载/media/ido/目录下。

```
•
1 root@ido:~# ls /media/ido/
```

2 KINGSTON

3.1 电源控制

默认所有USB的电源都是开启的,我们提供了开启/关闭电源的方法,如有需要,你可以通过该方法 控制它们的电源。

编号	电源控制节点
USB1	/sys/class/leds/usb_otg_pwr/brightness
USB2	/sys/class/leds/usb_host1_pwr/brightness
USB3	/sys/class/leds/usb_host2_pwr/brightness
USB4	/sys/class/leds/usb_host3_pwr/brightness

打开USB2的电源:

Bash 口复制代码
 root@ido:~# echo 255 > /sys/class/leds/usb_host1_pwr/brightness

关闭USB2的电源:

☑ 复制代码

Bash

1 root@ido:~# echo 0 > /sys/class/leds/usb_host1_pwr/brightness

USB1-4的电源控制方法类似。

3、TF CARD

主板配置了一个TF CARD接口,位于**J2**,当TF CARD接口插入TF卡后,会自动挂载到/media/ido/目录下。



4、以太网

主板配置了一个1000M以太网接口,位于J11,在系统中其对应的网络设备节点为eth0。



4.1 查看以太网IP地址

4.1.1 使用命令查看

系统默认以太网为动态获取IP,当以太网接口插入网线时,会自动获取IP。

```
Bash | D 复制代码
     root@ido:~# ifconfig eth0
 1
    eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 2
             inet 192.168.1.133 netmask 255.255.255.0 broadcast 192.168.1.255
 3
             inet6 fe80::3b43:b691:ded5:c497 prefixlen 64 scopeid 0x20<link>
 4
             ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)
 5
             RX packets 29 bytes 4592 (4.4 KiB)
 6
            RX errors 0 dropped 0 overruns 0 frame 0
 7
            TX packets 43 bytes 4146 (4.0 KiB)
 8
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
9
            device interrupt 40
10
11
12
     root@ido:~#
```

4.2 设置以太网临时IP地址

4.2.1 使用命令设置

•	Bash 日复制代码
1 root@ido:~# ifconfig eth0 192.168.1.123	
4.2.2 在桌面上设置 4.3 设置以太网永久静态IP	
新建/etc/netplan/ 00–installer–config.yaml ,然后写入如下内容(注意纲	宿进以Tab为单位):
-	Bash C 复制代码
<pre>1 network: 2 version: 2 3 renderer: NetworkManager 4 ethernets: 5 eth0: 6 dhcp4: no 7 dhcp6: no 8 addresses: [192.168.1.10/24] 9 gateway4: 192.168.1.1 10 nameservers: 11 addresses: [8.8.8.8, 11</pre>	4.114.114.114]
然后重启网络	
	Bash C 复制代码
1 root@ido:~# netplan apply	
重启网络后,eth0的ip地址已经变成刚才设置的静态ip。	

	1	<pre>root@ido:~# ifconfig eth0</pre>
	2	eth0: flags=4163 <up,broadcast,running,multicast> mtu 1500</up,broadcast,running,multicast>
	3	inet 192.168.1.123 netmask 255.255.255.0 broadcast 192.168.1.255
	4	inet6 fe80::b7c2:552b:4127:ee1c
	5	ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)
	6	RX packets 19 bytes 3281 (3.2 KB)
	7	RX errors 0 dropped 0 overruns 0 frame 0
	8	TX packets 36 bytes 5592 (5.5 KB)
	9	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
1	LØ	device interrupt 40

设置静态IP后,断电重启设备依旧生效。

5、WiFi

系统上电默认会打开WiFi,对应的网络设备节点为wlan0。

```
☑ 复制代码
                                                            Bash
   root@ido:~# ifconfig wlan0
1
   wlan0: flags=4099<UP,BROADCAST,MULTICAST>
2
                                             mtu 1500
           ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
3
           RX packets 0 bytes 0 (0.0 B)
4
           RX errors 0 dropped 0 overruns 0 frame 0
5
           TX packets 0 bytes 0 (0.0 B)
6
           TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
7
8
9
   root@ido:~
```

5.1 连接热点

连接热点可以在桌面上操作,也可以使用命令行操作。

5.1.1 方式-

鼠标左键点击桌面右下角的网络图标,即可看到WiFi热点列表:





点击要连接的热点,弹出密码输入窗口:

nm-tray - wireless password – 💌 × Password is needed for connection to 'TP-LINK_B87A':	
 ✓ OK × Cancel	

如果有连接键盘,直接输入密码即可;如果没有连接键盘,可以使用系统自带的软键盘。



点击桌面的Onboard图标,即可打开系统自带的软键盘。

使用软键盘输入密码后,点击【OK】连接热点



连接成功后,桌面右下角的网络图标将改变:



5.1.2 方式二

新建/etc/netplan/01-wifi-sta.yaml,并按照以下格式填写内容:

•	Bash D 复制代码
1 2 3 4 5 6 7 8	<pre>network: wifis: wlan0: dhcp4: true access-points: "TP-LINK_B87A": password: "12345678" version: 2</pre>
其	其中TP–LINK_B87A为要连接的热点名称,12345678为连接密码。
偱	8改成功后执行以下命令进行连接:
•	Bash 日 复制代码
1 2	root@ido:~# killall wpa_supplicant root@ido:~# netplan apply
等 5.2: 個	等待几秒钟后,将成功连接WiFi热点。 查看WiFi的IP地址 使用ifconfig命令可查看连接热点后获取的IP:
•	Bash D 复制代码
1 2 3 4 5 6 7 8 9 10 11 12	<pre>root@ido:~# ifconfig wlan0 wlan0: flags=4163<up,broadcast,running,multicast> mtu 1500 inet 192.168.1.165 netmask 255.255.255.0 broadcast 192.168.1.255 inet6 fe80::984a:9a2f:77b4:e899 prefixlen 64 scopeid 0x20<link/> ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet) RX packets 83 bytes 10479 (10.4 KB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 35 bytes 5285 (5.2 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 root@ido:~#</up,broadcast,running,multicast></pre>

测试wifi的网络:

```
root@ido:~# ping www.baidu.com -I wlan0
1
   PING www.a.shifen.com (14.215.177.39) from 192.168.1.117 p2p0: 56(84) byte
2
    s of data.
3
   64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=17.8 ms
   64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=9.30 ms
4
5
   64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=25.7 ms
   64 bytes from 14.215.177.39 (14.215.177.39): icmp seg=4 ttl=54 time=42.1 ms
6
   64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=13.1 ms
7
   64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=6 ttl=54 time=39.8 ms
8
   64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=7 ttl=54 time=28.1 ms
9
```

6、蓝牙

系统开机默认打开蓝牙,对应的网络节点为hci0。

•	Bash 日复制代码
1	root@ido:~# hciconfig
2	hci0: Type: Primary Bus: UART
3	BD Address: 70:D5:2B:5B:22:22 ACL MTU: 1021:8 SC0 MTU: 255:12
4	UP RUNNING
5	RX bytes:1665 acl:0 sco:0 events:57 errors:0
6	TX bytes:6311 acl:0 sco:0 commands:57 errors:0
7	
8	root@ido:~#

6.1 连接蓝牙设备

6.1.1 方式

将鼠标放到桌面右下角蓝牙图标,右键->Devices





在弹出的窗口中,点击【Search】:



将看到附近的蓝牙设备列表:



选中要连接设备,右键->Connect,即可连接该设备:

	4			
0	8	Bluetooth De	vices	- 0 ×
Network	Adapter Device Vie	ew Help		
	Q Search O	🛚 🔶 😡 Setup	- 🥔 Send File	
	seeyou			
Onboard	94:87:E0:9D:14	4:12	Connect	
	EE-10-FE-EF-CC-73 Miscellaneous EE:10:FE:EF:CC:73	EF-CC-73	Connect To:	
		:73	Network Access Point	
				Info
Htop			Send note	_
			Send a File	_
_			Pair	
			Trust	
Computer			Setup	
		^	Rename device	3/s 😡 😂
			Remove	

6.1.2 方式二

使用命令行方式,待补充。

7、音频

主板配置了一路耳机接口,位于J6,具备进行音频输出及录音功能。



7.1 查看声卡设备

Bash | D 复制代码 root@ido:~# aplay -l 1 **** List of PLAYBACK Hardware Devices **** 2 3 card 0: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0 [r ockchip,hdmi i2s-hifi-0] Subdevices: 1/1 4 Subdevice #0: subdevice #0 5 6 • card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81 7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0] Subdevices: 1/1 7 Subdevice #0: subdevice #0 8 9 root@ido:~#

注: MIPI固件仅有rk809-codec一个声卡。

7.2 播放音频

播放到HDMI (仅HDMI固件支持)

-

Bash | D 复制代码

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

播放到耳机

插入耳机,执行以下命令:



7.3 音量的调节

7.3.1 方式一

鼠标点击桌面右下角的音量图标



然后滑动鼠标进行音量调节:



当需要静音是,点击静音按钮:



7.3.2 方式二

7.4 录音

需要接入带麦克风的耳机。

使用arecord工具可以进行录音测试:

```
Bash D 复制代码
```

☑ 复制代码

Bash

//HDMI固件 1 2 root@ido:~# arecord -D hw:1,0 -r 48000 -c 2 -f S16 LE test.wav Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, St 3 ereo ^CAborted by signal Interrupt... 4 5 root@ido:~# 6 7 //MIPI固件 root@ido:~# arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE test.wav 8 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, St 9 ereo ^CAborted by signal Interrupt... 10 root@ido:~# 11

录音完后播放测试:

1	
T	// NUMI LIANY

- 2 root@ido:~# aplay -D plughw:1,0 ./test.wav
- 3 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Ste
 reo
- 4 root@ido:~#
- 5
- 6 //MIPI固件
- 7 root@ido:~# aplay -D plughw:0,0 ./test.wav
- 8 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Ste
 reo

9 root@ido:~#

8、摄

主板默认适配OV5648和OV8858摄像头,摄像头接口位于J22。



8.1 测试

8.1.1 测试摄像头是否存在

```
Bash | D 复制代码
     root@ido:~# media-ctl -p -d /dev/media0
 1
 2
     . . .
     - entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)
 3
                  type V4L2 subdev subtype Unknown flags 0
 4
                  device node name /dev/v4l-subdev2
 5
             pad0: Sink
 6
                     [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
                     <-> "m00_b_ov5648 2-0036":0 [ENABLED]
             pad1: Source
 9
10
                     [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
11
                     -> "rkisp-csi-subdev":0 [ENABLED]
12
     - entity 70: m00_b_ov5648 2-0036 (1 pad, 1 link)
13
14
                  type V4L2 subdev subtype Sensor flags 0
15
                  device node name /dev/v4l-subdev3
             pad0: Source
16
17 -
                     [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
                     -> "rockchip-csi2-dphy0":0 [ENABLED]
18
19
     root@ido:~#
```

结果显示m00_b_ov5648,说明摄像头存在,最高分辨率支持2592x1944。

8.1.3抓取视频

使用v4l2-ctl工具可以抓取摄像头的视频数据流。

Bash 🛛 复制代码

```
root@ido:~# v4l2-ctl --verbose -d /dev/video0 --set-fmt-video=width=1920,h
 1
     eight=1080,pixelformat='NV12' --stream-mmap=4 --set-selection=target=crop,
     flags=0,top=0,left=0,width=1920,height=1080 --stream-to=./out.yuv
2
    VIDIOC QUERYCAP: ok
3
    VIDIOC_G_FMT: ok
4
    VIDIOC S FMT: ok
     Format Video Capture Multiplanar:
5
             Width/Height
 6
                               : 1920/1080
 7
             Pixel Format
                               : 'NV12' (Y/CbCr 4:2:0)
8
             Field
                               : None
9
             Number of planes : 1
10
             Flags
                               •
             Colorspace
                               : Default
11
             Transfer Function : Default
12
13
             YCbCr/HSV Encoding: Default
             Ouantization
                               : Full Range
14
15
             Plane 0
                               5
16
                Bytes per Line : 1920
                Size Image
17
                               : 3110400
18
     VIDIOC G SELECTION: ok
19
     VIDIOC S SELECTION: ok
20
                     VIDIOC_REQBUFS returned @ (Success)
                     VIDIOC QUERYBUF returned 0 (Success)
21
                     VIDIOC QUERYBUF returned 0 (Success)
22
                     VIDIOC QUERYBUF returned 0 (Success)
23
                     VIDIOC QUERYBUF returned 0 (Success)
24
                     VIDIOC_QBUF returned 0 (Success)
25
                     VIDIOC_QBUF returned 0 (Success)
26
27
                     VIDIOC QBUF returned 0 (Success)
28
                     VIDIOC QBUF returned 0 (Success)
                     VIDIOC STREAMON returned 0 (Success)
29
     cap dqbuf: 0 seq:
                            1 bytesused: 3110400 ts: 1384.549991 (ts-monotoni
30
     c, ts-src-eof)
     cap dgbuf: 1 seq:
31
                            2 bytesused: 3110400 ts: 1384.616490 delta: 66.499
    ms (ts-monotonic, ts-src-eof)
     cap dqbuf: 2 seq:
                            3 bytesused: 3110400 ts: 1384.682975 delta: 66.485
32
     ms (ts-monotonic, ts-src-eof)
     cap dqbuf: 3 seq:
                            4 bytesused: 3110400 ts: 1384.749486 delta: 66.511
33
     ms (ts-monotonic, ts-src-eof)
     cap dqbuf: 0 seq:
                            5 bytesused: 3110400 ts: 1384.816022 delta: 66.536
34
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
35
     cap dqbuf: 1 seq:
                            6 bytesused: 3110400 ts: 1384.882509 delta: 66.487
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
     cap dqbuf: 2 seq:
                           7 bytesused: 3110400 ts: 1384.949025 delta: 66.516
36
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
```

```
37
     cap dqbuf: 3 seq: 8 bytesused: 3110400 ts: 1385.015545 delta: 66.520
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
38
     cap dqbuf: 0 seq:
                           9 bytesused: 3110400 ts: 1385.082051 delta: 66.506
    ms fps: 15.04 (ts-monotonic, ts-src-eof)
39
                           10 bytesused: 3110400 ts: 1385.148567 delta: 66.516
     cap dqbuf: 1 seq:
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
40
     cap dqbuf: 2 seq:
                           11 bytesused: 3110400 ts: 1385.215079 delta: 66.512
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
41
                           12 bytesused: 3110400 ts: 1385.281594 delta: 66.515
     cap dqbuf: 3 seq:
    ms fps: 15.04 (ts-monotonic, ts-src-eof)
42
                           13 bytesused: 3110400 ts: 1385.348115 delta: 66.521
     cap dqbuf: 0 seq:
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
43
                           14 bytesused: 3110400 ts: 1385.414669 delta: 66.554
     cap dqbuf: 1 seq:
     ms fps: 15.03 (ts-monotonic, ts-src-eof)
44
                           15 bytesused: 3110400 ts: 1385.481133 delta: 66.464
     cap dqbuf: 2 seq:
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
45
                           16 bytesused: 3110400 ts: 1385.547656 delta: 66.523
     cap dqbuf: 3 seq:
    ms fps: 15.04 (ts-monotonic, ts-src-eof)
46
     cap dqbuf: 0 seq:
                           17 bytesused: 3110400 ts: 1385.614172 delta: 66.516
    ms fps: 15.04 (ts-monotonic, ts-src-eof)
47
     cap dqbuf: 1 seq:
                           18 bytesused: 3110400 ts: 1385.680680 delta: 66.508
     ms fps: 15.04 (ts-monotonic, ts-src-eof)
48
                           19 bytesused: 3110400 ts: 1385.747241 delta: 66.561
     cap dqbuf: 2 seq:
     ms fps: 15.03 (ts-monotonic, ts-src-eof)
49
     cap dqbuf: 3 seq:
                           20 bytesused: 3110400 ts: 1385.813714 delta: 66.473
     ms fps: 15.03 (ts-monotonic, ts-src-eof)
50
     ^C
```

按Ctrl-C停止抓取,视频流保存到文件out.yuv。

使用ffplay工具播放抓取的视频流:

```
Bash D 复制代码
     root@ido:~# ffplay -f rawvideo -video_size 1920x1080 -pix_fmt nv12 ./out.y
 1
     uv
2
     ffplay version 4.2.4-1ubuntu1.0firefly1 Copyright (c) 2003-2020 the FFmpe
     q developers
3
       built with gcc 9 (Ubuntu 9.3.0-17ubuntu1~20.04)
       configuration: --prefix=/usr --extra-version=1ubuntu1.0firefly1 --toolch
4
     ain=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aar
     ch64-linux-gnu --arch=arm64 --enable-gpl --disable-stripping --enable-avre
     sample --disable-filter=resample --enable-avisynth --enable-gnutls --enabl
     e-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs
     2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libflite
     --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-l
     ibgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmys
     ofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-lib
     pulse --enable-librsvg --enable-librubberband --enable-libshine --enable-l
     ibsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libth
     eora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-l
     ibvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxm
     l2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable
     -omx --enable-openal --enable-opencl --enable-opengl --enable-sdl2 --enabl
     e-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --en
     able-frei0r --enable-libx264 --enable-libdrm --enable-librga --enable-rkmp
     p --enable-version3 --disable-libopenh264 --disable-vaapi --disable-vdpau
     --disable-decoder=h264 v4l2m2m --disable-decoder=vp8 v4l2m2m --disable-dec
     oder=mpeg2 v4l2m2m --disable-decoder=mpeg4 v4l2m2m --disable-muxer='ac3,ea
     c3,mlp,truehd' ---disable-encoder='ac3_fixed,ac3,mlp,spdif,truehd' ---disabl
     e-demuxer='ac3,eac3,mlp,truehd,dts,dtshd' --disable-parser='aac,ac3,mlp' -
     -disable-decoder='ac3,eac3,mlp,dolby_e' --enable-shared --disable-doc
                     56. 31.100 / 56. 31.100
5
       libavutil
6
       libavcodec
                      58. 54.100 / 58. 54.100
                      58. 29.100 / 58. 29.100
7
       libavformat
                      58. 8.100 / 58. 8.100
8
       libavdevice
      libavfilter
                      7. 57.100 / 7. 57.100
 9
                              0 / 4.
       libavresample
                       4.
10
                           0.
                                       0. 0
11
     libswscale
                       5.
                          5.100 / 5.
                                       5.100
       libswresample
12
                       3.
                          5.100 / 3.
                                        5.100
       libpostproc
                           5.100 / 55.
13
                      55.
                                        5.100
     Option -pix fmt is deprecated, use -pixel format.
14
15
     libGL error: failed to create dri screen
16
     libGL error: failed to load driver: rockchip
17
     libGL error: failed to create dri screen
     libGL error: failed to load driver: rockchip
18
19 • [rawvideo @ 0x7f3c000ba0] Estimating duration from bitrate, this may be in
     accurate
     Input #0, rawvideo, from './out.yuv':
20
```

21 Duration: 00:00:04.00, start: 0.000000, bitrate: 622075 kb/s 22 Stream #0:0: Video: rawvideo (NV12 / 0x3231564E), nv12, 1920x1080, 622

080 kb/s, 25 tbr, 25 tbn, 25 tbc



9、RTC

23

主板包含2个RTC,其中/dev/rtc0为外部RTC(HYM8563),/dev/rtc1为CPU内部的RTC (RK808)。系统默认使用rtc0的时间。

9.1 获取RTC时间

Bash D 复制代码

- 1 root@ido:~# hwclock
- 2 2022-11-10 02:16:23.617474+00:00

9.2 设置RTC时间

Bash D 复制代码

D 复制代码

```
1 root@ido:~# hwclock
```

- 2 **2022**-11-18 **08**:30:40.778874+00:00
- 3 root@ido:~# date -s '2022-11-10 10:17:00'
- 4 Thu Nov 10 10:17:00 UTC 2022
- 5 root@ido:~# hwclock -w
- 6 root@ido:~# hwclock
- 7 2022-11-18 08:31:06.829691+00:00
- 8 root@ido:~#

9.3 RTC定时开机

CPU内部的rtc1支持支持定时开机功能。

以下测试关机60秒定时开机:

```
1
    root@ido:~# echo +60 > /sys/class/rtc/rtc1/wakealarm
 2
    root@ido:~# poweroff
             Stopping Session c1 of user ido.
 3
             Stopping Session c2 of user root.
 4
    [ OK ] Removed slice system-modprobe.slice.
 5 📼
 6 [ OK ] Stopped target Bluetooth.
 7
     . . .
8 [ OK ] Reached target Shutdown.
9 • [ OK ] Reached target Final Step.
10 [ OK ] Finished Power-Off.
11 [ OK ] Reached target Power-Off.
12 [ 1195.829009] Local MAC: 82:4c:21:62:f5:35
13 [ 1195.834186] set rk809 rtc alarm!
14 [ 1195.837515] reboot: Power down
15
```

等待60秒后, 主板会开机。

10、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。

11、屏幕控制

11.1 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 @ 复制代码
    1 root@ido:~# echo 100 > /sys/class/backlight/backlight/brightness
    2 root@ido:~#
```

11.2 屏幕旋转

使用xrandr工具可以实现屏幕的旋转。

11.2.1 临时旋转

系统启动后,执行xrandr –o normal,inverted,left,right,可以实现临时旋转屏幕方向,其中normal 表示顺时针旋转0度,inverted表示顺时针旋转180度,left表示顺时针旋转270度,right表示顺时针旋转90度。

Bash C 复制代码

```
1 root@ido:~# xrandr -o inverted
```

11.2.2 永久旋转

修改/etc/default/xrandr启动文件,可以实现永久旋转。

以旋转180度为例:

•		Bash D 复制代码
1	<pre>root@ido:~# cat /etc/default/xrandr</pre>	
2	#!/bin/sh	
3	/usr/bin/xrandr -o inverted	
4	root@ido:~#	

这样修改后,每次重启设备,桌面将旋转180度。

12、按键

主板共配置3个按键:

标记	名称	功能
SW1	reset	硬件复位
SW2	recovery	用于烧录,或上报KEY_VOLUMEUP
SW5	power	电源键,用于开关机及待机唤醒,或上报KEY_POWER



13、双排针功能及接口

主板预留了40Pin的双排针接口,包含一路UART、一路I2C、一路SPI及19个GPIO。



13.1 双排针Pin功能定义

各个Pin的功能定义见下表:

序号	定义	电平	说明
----	----	----	----

1	VDD_3V3	3.3V	供电输出3.3V
2	VDD_5V	5V	可做5V输入供电,也可输出5V
3	I2C_SDA	3.3V	I2C数据信号
4	VDD_5V	5V	可做5V输入供电,也可输出5V
5	I2C_SCL	3.3V	I2C时钟信号
6	GND	GND	电源地
7	GPIO_7	3.3V	gpio编号为23
8	UART_TXD	3.3V	/dev/ttyS0 发送
9	GND	GND	电源地
10	UART_RXD	3.3V	/dev/ttyS0 接收
11	GPIO_0	3.3V	gpio编号为15
12	GPIO_1	3.3V	gpio编号为22
13	GPIO_2	3.3V	gpio编号为20
14	GND	GND	电源地
15	GPIO_3	3.3V	gpio编号为21
16	GPIO_4	3.3V	gpio编号为124
17	VDD_3V3	3.3V	供电输出3.3V
18	GPIO_5	3.3V	gpio编号为125
19	SPI_MOSI	3.3V	SPI数据信号
20	GND	GND	电源地
21	SPI_MISO	3.3V	SPI数据信号
22	GPIO_6	3.3V	gpio编号为130
23	SPI_SCLK	3.3V	SPI时钟信号
24	SPI_CS0	3.3V	SPI片选信号0
25	GND	GND	电源地

26	SPI_CS1	3.3V	SPI片选信号1
27	GPIO_30	3.3V	gpio编号为138
28	GPIO_31	3.3V	gpio编号为139
29	GPIO_21	3.3V	gpio编号为132
30	GND	GND	电源地
31	GPIO_22	3.3V	gpio编号为121
32	GPIO_26	3.3V	gpio编号为120
33	GPIO_23	3.3V	gpio编号为122
34	GND	GND	电源地
35	GPIO_24	3.3V	gpio编号为127
36	GPIO_27	3.3V	gpio编号为123
37	GPIO_25	3.3V	gpio编号为133
38	GPIO_28	3.3V	gpio编号为126
39	GND	GND	电源地
40	GPIO_29	3.3V	gpio编号为131

13.1.1 UART

双排针包含一路UART接口,电平类型为TTL电平,系统中对应的设备节点为/dev/ttyS0。



13.1.2 SPI

双排针包含一路SPI接口,系统中对应的设备节点为/dev/spidev3.0。



13.1.3 I2C

双排针包含一路I2C接口,系统种对应的设备节点为/dev/i2c-2。



13.1.4 GPIO

双排针共包含19个GPIO接口。



以pin7的GPIO_7为例,对应系统的gpio number 为23,可通过sysctl的方式控制此路GPIO的方向和 电平。

输出模式

```
    Shell @ 复制代码
    # 设置GPIO方向为输出
    echo out > /sys/class/gpio/gpio23/direction
    # 设置输出高电平
    echo 1 > /sys/class/gpio/gpio23/value
    # 设置输出低电平
    echo 0 > /sys/class/gpio/gpio23/value
    注 输出高指3.3v、输出低指0v。
```

输入模式

```
    Shell 回复制代码
    # 设置GPIO方向为输入
    echo in > /sys/class/gpio/gpio23/direction
    # 读取GPIO接口电平
    cat /sys/class/gpio/gpio23/value
```

返回0指输入低电平(0v),返回1指输入高电平(3.3v)。

- Att P. WWW. HOUSE