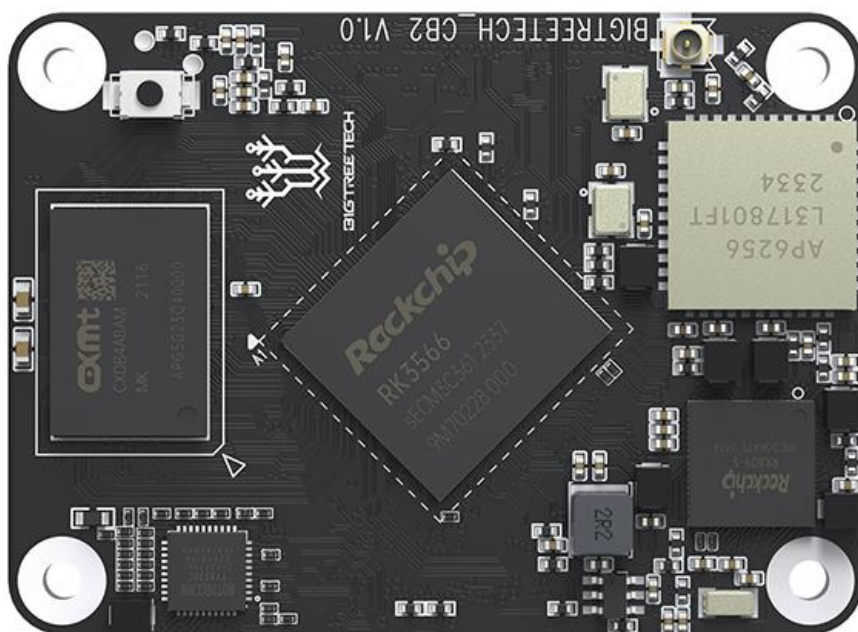


BIGTREE TECH

CB2

用户手册



目录

目录	2
修订历史	4
一、产品简介	5
1.1 产品特点	5
1.2 产品参数	5
1.3 尺寸图	6
二、外设接口	7
2.1 Pin 脚说明	7
三、接口介绍	12
3.1 SW1 按键说明	12
3.2 40 pin GPIO	13
四、烧录系统	14
4.1 下载系统镜像	14
4.2 烧录系统到 MicroSD 卡	14
4.3 烧录系统到 eMMC	15
4.3.1 使用 RKDevTool 烧录系统到 eMMC (Windows)	15
4.3.2 使用 MicroSD 卡烧录系统到 eMMC	17
4.4 擦除 eMMC	18
4.4.1 使用 UMS 擦除 eMMC (Windows)	18
4.4.2 使用 RKDevTool 擦除 eMMC (Windows)	18
4.4.3 从 MicroSD 卡启动系统后擦除 eMMC	19
五、系统配置	20
5.1 使用网线	20
5.2 设置 WiFi	20
5.3 配置 overlays	20
5.4 配置显示屏	21
5.5 SPI 转 CAN 的使用	22

5.6 CSI 相机使用及 crowsnest 配置	22
5.7 蓝牙的使用	23
5.8 3.5mm 圆口耳机设置	25
六、SSH 连接设备	27
七、注意事项	29

修订历史

版本	日期	修改说明
v1.00	2024/04/24	初稿
v1.01	2024/05/28	增加 minimal 版本系统的账户信息
v1.02	2024/06/05	增加 40-Pin GPIO 表格及新的计算方式
v1.03	2025/03/26	增加 40-Pin GPIO 复用功能描述
v1.04	2025/04/14	V3.0.1 版本系统，u-boot loader 模式变更为 ums 模式

一、产品简介

BIGTREETECH CB2 是针对树莓派 CM4 缺货问题推出的替代方案,所有输出信号采用 2 个 100PIN 微型 BTB 高速连接座与外面扩展底板方便快速连接(包括千兆以太网、HDMI 等),板载 2.4G、5GWIFI BT5.2。

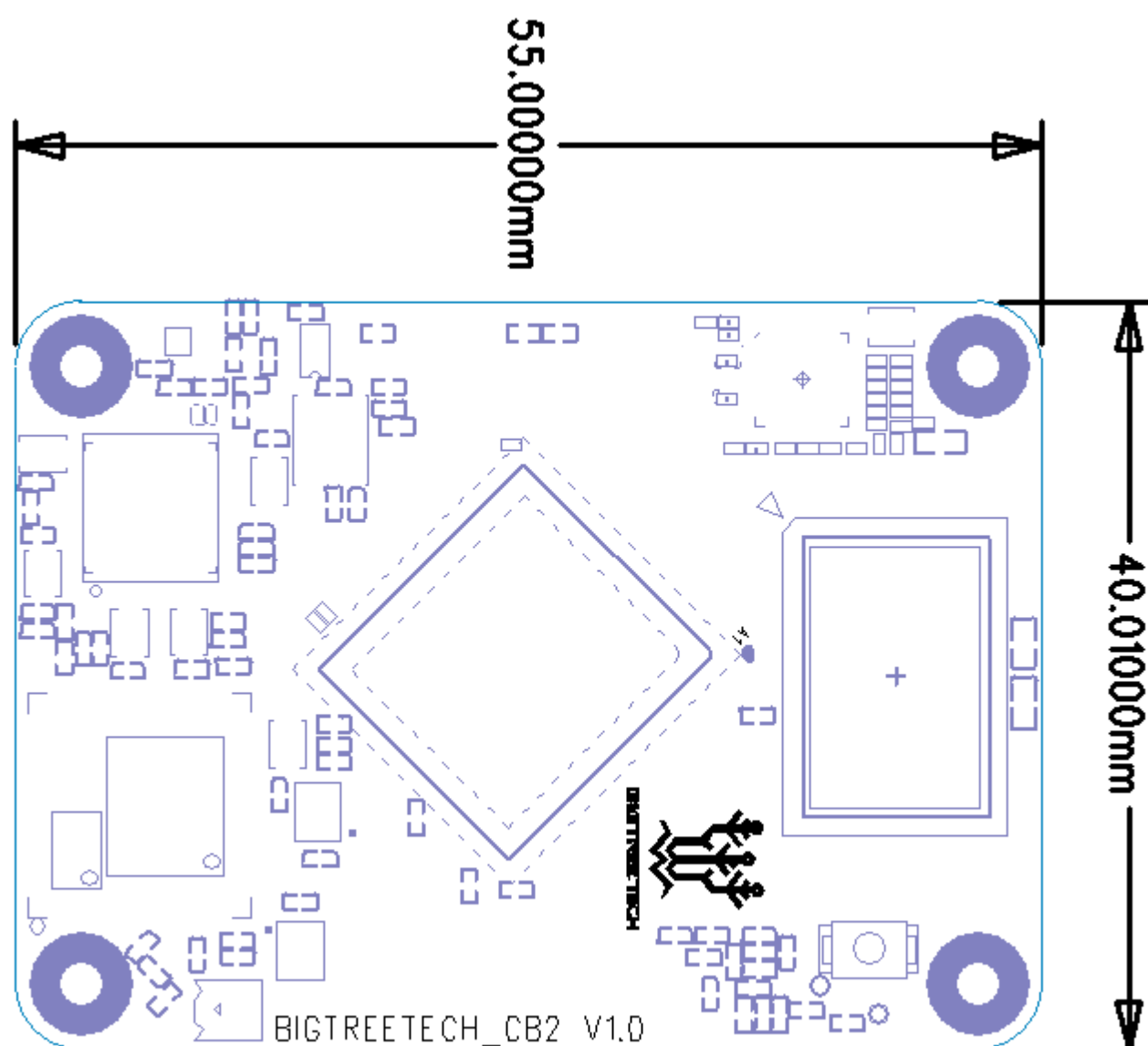
1.1 产品特点

- CPU: 瑞芯微 RK3566, 四核 Cortex-A55 @1.8GHz
- GPU: Mali-G52 1-Core-2EE
- NPU: 0.8 TOPS NPU
- RAM: 2GB/4GB LPDDR4
- 板载 EMMC
- MIPI DSI 显示支持
- 摄像头 2-Lane MIPI CSI2
- 3 路 USB2.0 端口 1 路 USB3.0
- PCIe 2.1 1x1 Lane
- 千兆以太网+433Mbps WiFi+BT5.0
- 40-pin GPIO
- 与树莓派 CM4 完全相同的 BTB 座子

1.2 产品参数

1. 核心板外观尺寸: 40mm*55mm
2. 核心板安装尺寸: 33mm*48mm
3. 核心板输入电压: $5V \pm 5\%$ /2A
4. 核心板输出电压: $3.3V \pm 2\%$ /100mA
5. 核心板输出电压: $1.8V \pm 2\%$ /100mA
6. 核心板 WIFI:2.4G/5G, 802.11 ac/a/b/g/n/无线标准

1.3 尺寸图



二、外设接口

2.1 Pin 脚说明

PIN	Connector	Signal	Description
1	A connector_01	GND	
2	A connector_02	GND	
3	A connector_03	GBIT_MDI3_P	
4	A connector_04	GBIT_MDI1_P	
5	A connector_05	GBIT_MDI3_N	
6	A connector_06	GBIT_MDI1_N	
7	A connector_07	GND	
8	A connector_08	GND	
9	A connector_09	GBIT_MDI2_N	
10	A connector_10	GBIT_MDIO_N	
11	A connector_11	GBIT_MDI2_P	
12	A connector_12	GBIT_MDIO_P	
13	A connector_13	GND	
14	A connector_14	GND	
15	A connector_15	1000M_LED	
16	A connector_16	CAMERAB_PDN_L	
17	A connector_17	100M_LED	
18	A connector_18	SPDIF_TX_M2	
19	A connector_19	PWM3_IR	
20	A connector_20	NC	
21	A connector_21	WORKING_LEDEN_H	
22	A connector_22	GND	
23	A connector_23	GND	
24	A connector_24	GPIO0_C3	
25	A connector_25	GPIO4_C2	
26	A connector_26	GPIO4_C5	
27	A connector_27	GPIO4_C3	
28	A connector_28	GPIO0_C0	
29	A connector_29	GPIO0_A0	
30	A connector_30	GPIO3_D7	
31	A connector_31	GPIO0_C1	
32	A connector_32	GND	
33	A connector_33	GND	
34	A connector_34	NC	
35	A connector_35	GPIO0_B3	
36	A connector_36	GPIO0_B4	
37	A connector_37	GPIO0_A6	
38	A connector_38	GPIO3_C3	
39	A connector_39	GPIO4_A2	
40	A connector_40	GPIO3_C2	
41	A connector_41	GPIO0_C4	

42	A connector_42	GND	
43	A connector_43	GND	
44	A connector_44	GPIO3_C1	
45	A connector_45	GPIO4_A3	
46	A connector_46	GPIO1_A1	
47	A connector_47	GPIO4_C6	
48	A connector_48	GPIO1_A0	
49	A connector_49	GPIO0_C0	
50	A connector_50	GPIO0_C7	
51	A connector_51	GPIO0_D0	DEBUG UART
52	A connector_52	GND	
53	A connector_53	GND	
54	A connector_54	GPIO3_A1	
55	A connector_55	GPIO0_D1	DEBUG UART
56	A connector_56	GPIO4_B3	
57	A connector_57	SDC0-CLK	SDCARD Clock signal
58	A connector_58	GPIO4_B2	
59	A connector_59	GND	
60	A connector_60	GND	
61	A connector_61	SDC0-D3	SDCARD Data3 signal
62	A connector_62	SDC0-CMD	SDCARD CMD signal
63	A connector_63	SDC0-D0	SDCARD Data0 signal
64	A connector_64	NC	
65	A connector_65	GND	
66	A connector_66	GND	
67	A connector_67	SDC0-D1	SDCARD Data1 signal
68	A connector_68	NC	
69	A connector_69	SDC0-D2	SDCARD Data2 signal
70	A connector_70	NC	
71	A connector_71	GND	
72	A connector_72	NC	
73	A connector_73	GPIO0_B5	
74	A connector_74	GND	
75	A connector_75	GPIO3_D2	
76	A connector_76	GPIO3_D3	SDCARD detect
77	A connector_77	VCC_5V	5V IN /2A
78	A connector_78	NC	
79	A connector_79	VCC_5V	5V IN /2A
80	A connector_80	GPIO4_B5	
81	A connector_81	VCC_5V	5V IN /2A
82	A connector_82	GPIO4_B4	
83	A connector_83	VCC_5V	5V IN /2A
84	A connector_84	3V3	3.3v out /200mA
85	A connector_85	VCC_5V	5V IN /2A
86	A connector_86	3V3	3.3v out /200mA
87	A connector_87	VCC_5V	5V IN /2A

88	A connector_88	1V8	1.8v out /100mA
89	A connector_89	GPI03_B4	
90	A connector_90	1V8	1.8v out /100mA
91	A connector_91	NC	
92	A connector_92	PWRON	
93	A connector_93	RECOVERY	
94	A connector_94	NC	
95	A connector_95	GPI04_A1	
96	A connector_96	NC	
97	A connector_97	GPI04_A5	
98	A connector_98	GND	
99	A connector_99	PMIC_PWRON	
100	A connector_100	AP-RESET	
101	B connector_1	USB_OTG0_ID	
102	B connector_2	PCIE20_CLKREQn_M2	
103	B connector_3	USB_OTG0_DM	
104	B connector_4	LINEOUTL	
105	B connector_5	USB_OTG0_DP	
106	B connector_6	LINEOUTR	
107	B connector_7	GND	
108	B connector_8	GND	
109	B connector_9	PCIE20_PERSTn_M2	
110	B connector_10	PCIE20_REFCLKP	
111	B connector_11	GPI04_B0	
112	B connector_12	PCIE20_REFCLKN	
113	B connector_13	GND	
114	B connector_14	GND	
115	B connector_15	MIPI_CSI_RX_D0N	
116	B connector_16	PCIE20_RXP	
117	B connector_17	MIPI_CSI_RX_D0P	
118	B connector_18	PCIE20_RXN	
119	B connector_19	GND	
120	B connector_20	GND	
121	B connector_21	MIPI_CSI_RX_D1N	
122	B connector_22	PCIE20_TXP	
123	B connector_23	MIPI_CSI_RX_D1P	
124	B connector_24	PCIE20_TXN	
125	B connector_25	GND	
126	B connector_26	GND	
127	B connector_27	MIPI_CSI_RX_CLKON	
128	B connector_28	USB3-DM	
129	B connector_29	MIPI_CSI_RX_CLKOP	
130	B connector_30	USB3-DP	
131	B connector_31	GND	
132	B connector_32	GND	
133	B connector_33	MIPI_CSI_RX_D2N	

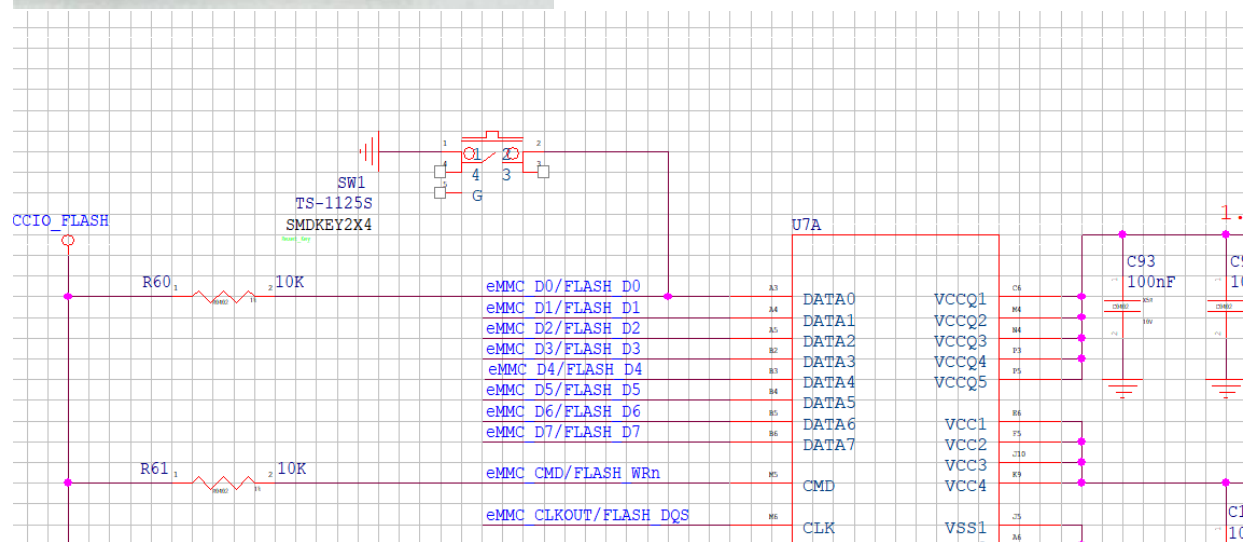
134	B connector_34	MIC1_IN	
135	B connector_35	MIPI_CSI_RX_D2P	
136	B connector_36	MIC2_IN	
137	B connector_37	GND	
138	B connector_38	GND	
139	B connector_39	MIPI_CSI_RX_D3N	
140	B connector_40	MIPI_CSI_RX_CLK1N	
141	B connector_41	MIPI_CSI_RX_D3P	
142	B connector_42	MIPI_CSI_RX_CLK1P	
143	B connector_43	GPIO4_A7	
144	B connector_44	GND	
145	B connector_45	HP_SNS	
146	B connector_46	USB2_HOST2_DP	
147	B connector_47	HP_DET_L	
148	B connector_48	USB2_HOST2_DM	
149	B connector_49	SARADC_VIN2	
150	B connector_50	GND	
151	B connector_51	HCEC	HDMI CEC
152	B connector_52	USB3_HOST1_DP	
153	B connector_53	HHPD	HDMI Hotplug
154	B connector_54	USB3_HOST1_DM	
155	B connector_55	GND	
156	B connector_56	GND	
157	B connector_57	MIPI_DSI_TX0_D0N	
158	B connector_58	USB3_HOST1_SSTXP	
159	B connector_59	MIPI_DSI_TX0_D0P	
160	B connector_60	USB3_HOST1_SSTXN	
161	B connector_61	GND	
162	B connector_62	GND	
163	B connector_63	MIPI_DSI_TX0_D1N	
164	B connector_64	USB3_HOST1_SSRXP	
165	B connector_65	MIPI_DSI_TX0_D1P	
166	B connector_66	USB3_HOST1_SSRXN	
167	B connector_67	GND	
168	B connector_68	GND	
169	B connector_69	MIPI_DSI_TX0_CLKN	
170	B connector_70	HTX2P	HDMI TX2 Positive.
171	B connector_71	MIPI_DSI_TX0_CLKP	
172	B connector_72	HTX2N	HDMI TX2 Negative.
173	B connector_73	GND	
174	B connector_74	GND	
175	B connector_75	MIPI_DSI_TX1_D0N	
176	B connector_76	HTX1P	HDMI TX1 Positive.
177	B connector_77	MIPI_DSI_TX1_D0P	
178	B connector_78	HTX1N	HDMI TX1 Negative.
179	B connector_79	GND	

180	B connector_80	GND	
181	B connector_81	MIPI_DSI_TX1_D1N	
182	B connector_82	HTXOP	HDMI TX0 Positive.
183	B connector_83	MIPI_DSI_TX1_D1P	
184	B connector_84	HTXON	HDMI TX0 Negative.
185	B connector_85	GND	
186	B connector_86	GND	
187	B connector_87	MIPI_DSI_TX1_CLKN	
188	B connector_88	HTXCP	HDMI CLK Positive.
189	B connector_89	MIPI_DSI_TX1_CLKP	
190	B connector_90	HTXCN	HDMI CLK Negative.
191	B connector_91	GND	
192	B connector_92	GND	
193	B connector_93	MIPI_DSI_TX1_D2N	
194	B connector_94	MIPI_DSI_TX1_D3N	
195	B connector_95	MIPI_DSI_TX1_D2P	
196	B connector_96	MIPI_DSI_TX1_D3P	
197	B connector_97	GND	
198	B connector_98	GND	
199	B connector_99	HSDA	HDMI I2C
200	B connector_100	HSCL	HDMI I2C

三、接口介绍

3.1 SW1 按键说明

SW1 按住会将 eMMC 信号线与 GND 短路，禁止主控与 eMMC 通信。



3.2 40 pin GPIO

PI2/CB2 40-Pin GPIO											
function5	function4	function3	function2	function1	Pin	function1	function2	function3	function4	function5	
				3.3V	1 2	5V					
			I2C4_SDA_M0	GPIO4_B2(gpiochip4/gpio10)	3 4	5V					
			I2C4_SCL_M0	GPIO4_B3(gpiochip4/gpio11)	5 6	GND					
				GPIO3_A1(gpiochip3/gpio1)	7 8	GPIO0_D1(gpiochip0/gpio25)	UART2_TX_M0				
				GPIO3_A1(gpiochip3/gpio1)	7 8	GPIO0_D1(gpiochip0/gpio25)	UART2_TX_M0				
				GND	9 10	GPIO0_D0(gpiochip0/gpio24)	UART2_RX_M0				
	UART0_CTSn	PWM0_M1		GPIO0_C7(gpiochip0/gpio23)	11 12	GPIO0_B0(gpiochip0/gpio8)					
		UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0(gpiochip1/gpio0)	13 14	GND					
		UART3_TX_M0	I2C3_SCL_M0	GPIO1_A1(gpiochip1/gpio1)	15 16	GPIO4_C6(gpiochip4/gpio22)	PWM13_M1			UART9_RX_M1	
				3.3V	17 18	GPIO4_A3(gpiochip4/gpio3)				UART7_RX_M2	
	SPI1_MOSI_M1			GPIO3_C1(gpiochip3/gpio17)	19 20	GND					
UART5_TX_M1	SPI1_MISO_M1			GPIO3_C2(gpiochip3/gpio18)	21 22	GPIO0_C4(gpiochip0/gpio20)	PWM5		UART0_RTSn		
UART5_RX_M1	SPI1_CLK_M1			GPIO3_C3(gpiochip3/gpio19)	23 24	GPIO4_A2(gpiochip4/gpio2)				UART7_TX_M2	
				GND	25 26	GPIO0_A6(gpiochip0/gpio6)					
			I2C1_SDA	GPIO0_B4(gpiochip0/gpio12)	27 28	GPIO0_B3(gpiochip0/gpio11)	I2C1_SCL				
				GPIO3_D6(gpiochip3/gpio30)	29 30	GND					
				GPIO3_D7(gpiochip3/gpio31)	31 32	GPIO0_C0(gpiochip0/gpio16)	PWM1_M0		UART0_RX		
	UART0_TX		PWM2_M0	GPIO0_C1(gpiochip0/gpio17)	33 34	GND					
UART9_TX_M1		SPI3_MISO_M1	PWM12_M1	GPIO4_C5(gpiochip4/gpio21)	35 36	GPIO0_A0(gpiochip0/gpio0)					
			PWM4	GPIO0_C3(gpiochip0/gpio19)	37 38	GPIO4_C3(gpiochip4/gpio19)	PWM15_IR_M1	SPI3_MOSI_M1			
				GND	39 40	GPIO4_C2(gpiochip4/gpio18)	PWM14_M1	SPI3_CLK_M1			

GPIO 引脚的计算方式如下：

$$\text{GPIO4_B2} = (\text{'B'} - \text{'A'}) * 8 + 2 = 1 * 8 + 2 = \text{gpiochip4/gpio10}$$

$$\text{GPIO3_D7} = (\text{'D'} - \text{'A'}) * 8 + 7 = 3 * 8 + 7 = \text{gpiochip3/gpio31}$$

四、烧录系统

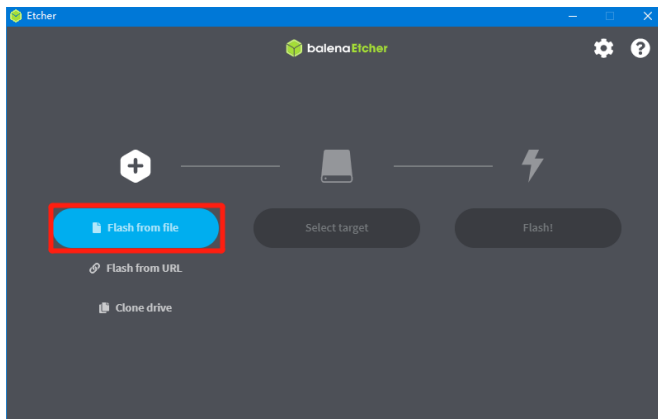
4.1 下载系统镜像

只能下载安装我们提供的系统镜像：

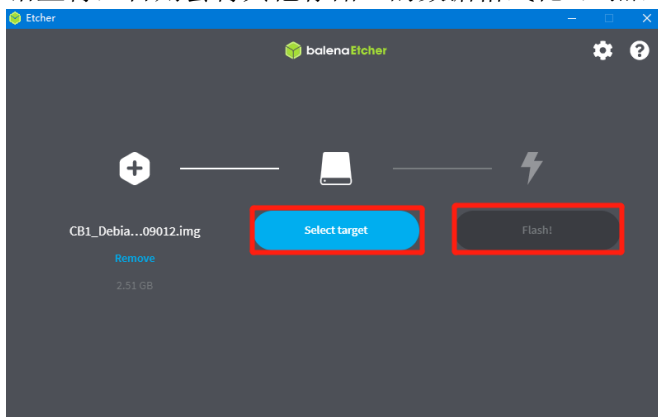
<https://github.com/bigtreotech/CB2/releases>

4.2 烧录系统到 MicroSD 卡

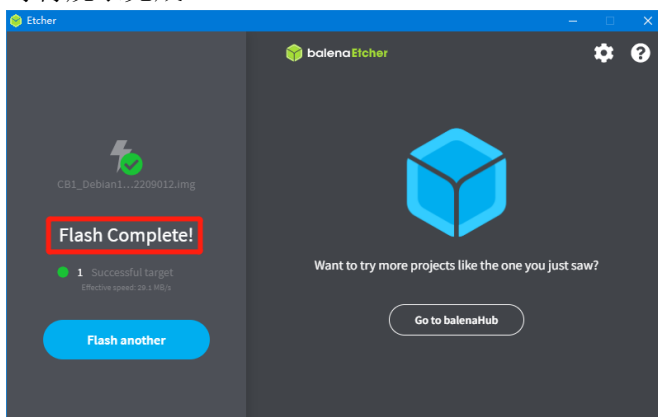
1. 下载 balenaEtcher (<https://www.balena.io/etcher/>) 软件，安装并运行
2. 将 Micro SD 卡通过读卡器插入到电脑
3. 选择下载到电脑中的镜像



4. 选择待烧录的 Micro SD 卡（烧录镜像会将 Micro SD 卡格式化，千万注意不要选错盘符，否则会将其他存储上的数据格式化），点击“烧录”



5. 等待烧录完成

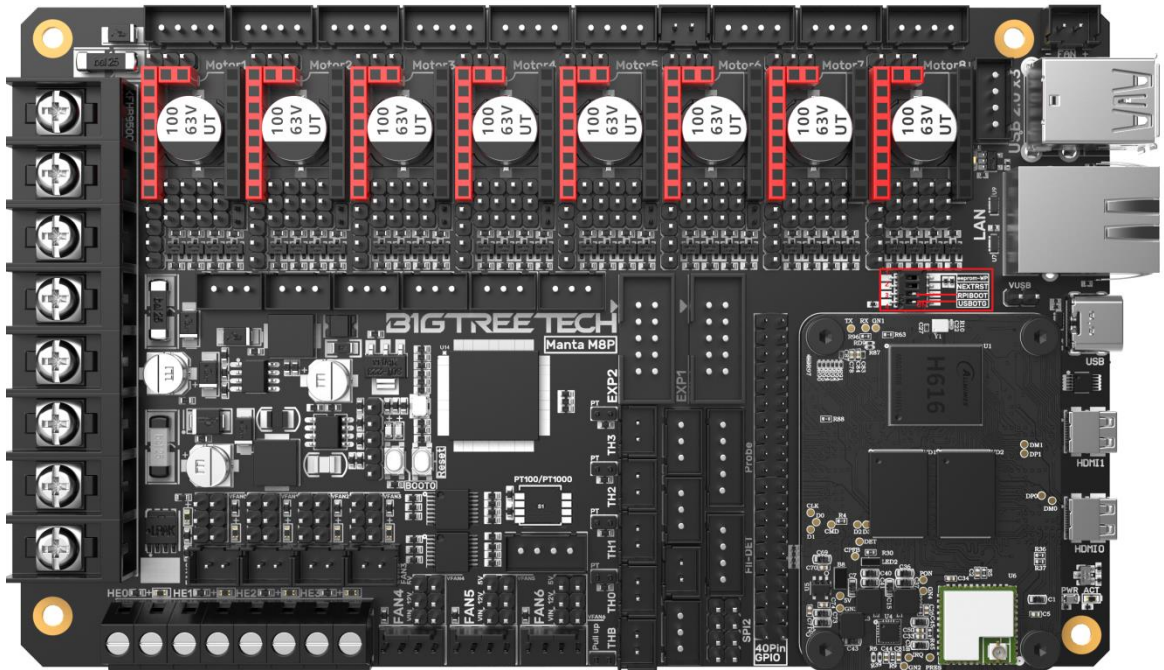


4.3 烧录系统到 eMMC

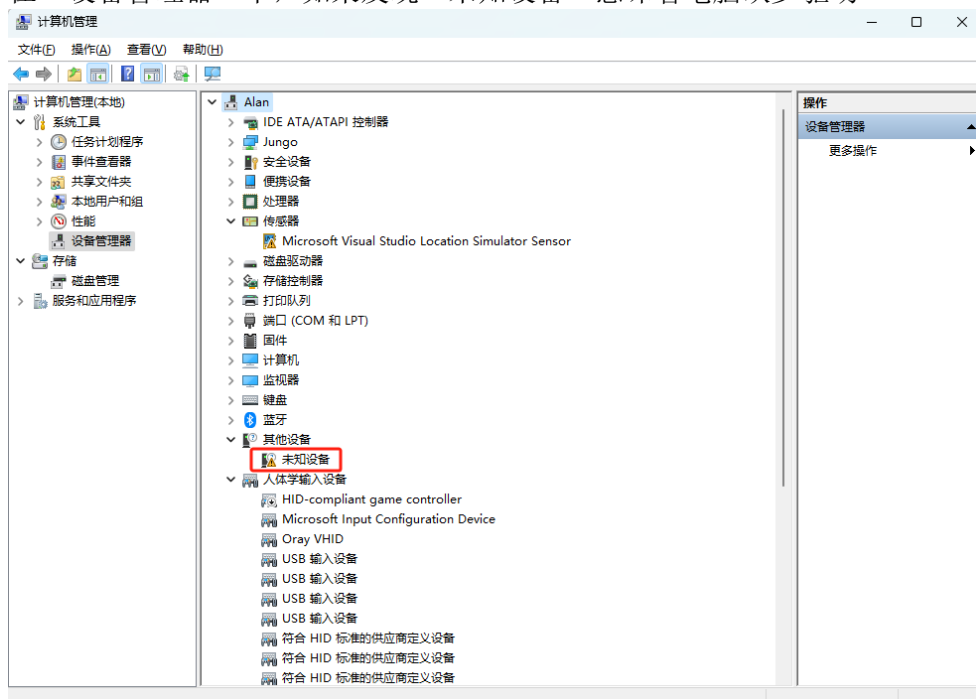
4.3.1 使用 RKDevTool 烧录系统到 eMMC (Windows)

下载 RKDevTool (<https://github.com/bigtreetech/CB2>) 到电脑上并解压。并且注意不要插 MicroSD 卡。

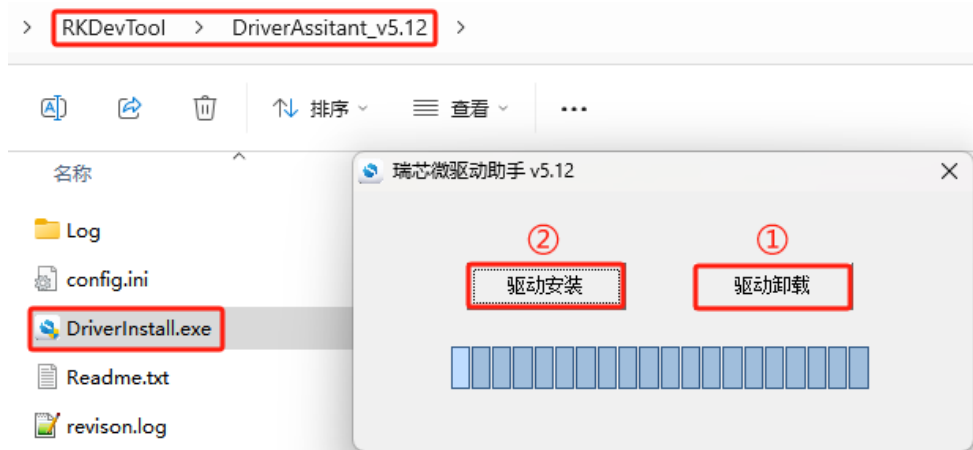
1. 将拨码开关的 4 (USBOTG)、3 (RPBOOT) 拨到 ON 进入 BOOT 模式



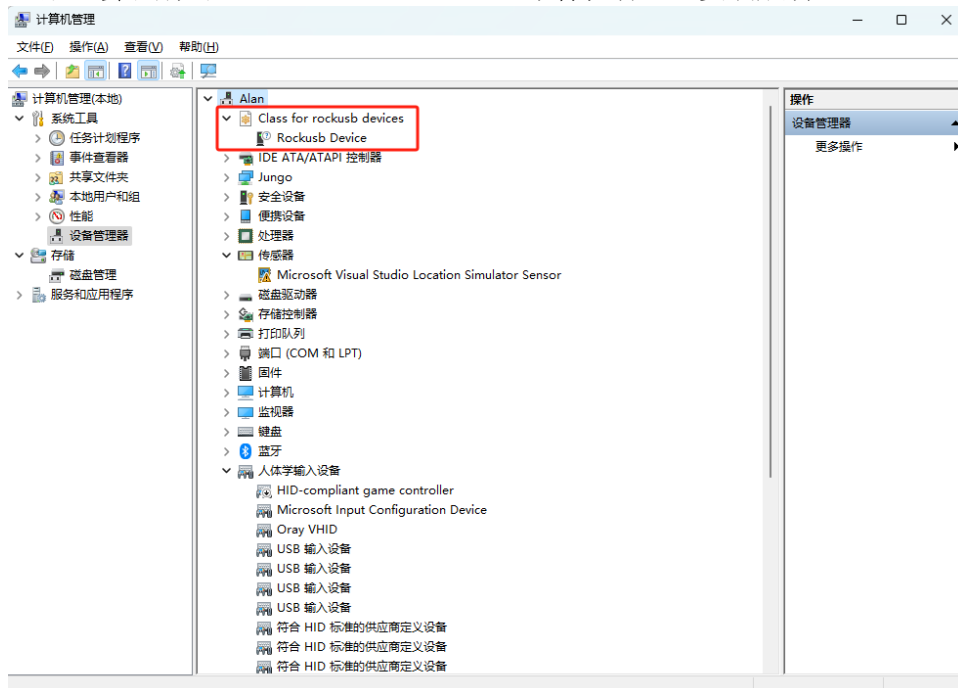
2. 然后用 Type-C 线插到电脑上。
3. 安装驱动
 - (1) 在“设备管理器”中，如果发现“未知设备”意味着电脑缺少驱动



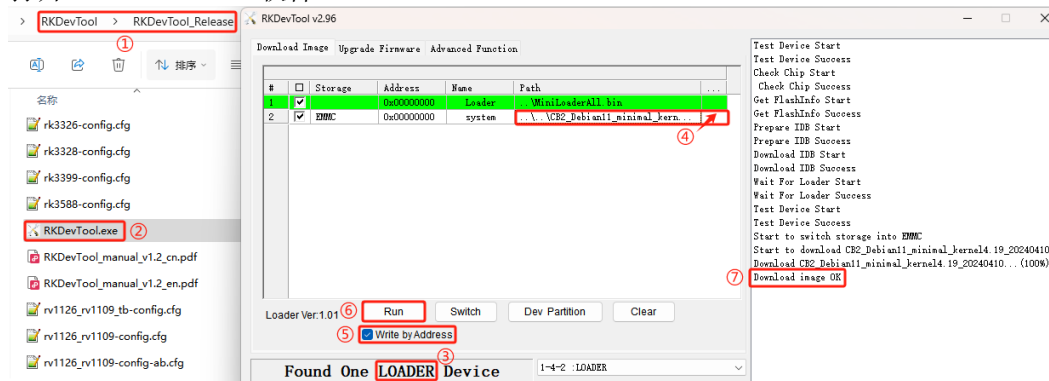
- (2) 打开下载的 RKDevTool 中的 DriverAssitant 工具，先点击“①驱动卸载”，再点击“②驱动安装”，这样可以保证安装的驱动为最新版本的。



- (3) 等待安装完成后，按住“Recovery”键，重新拔插一下 Type-C 线，“设备管理器”会识别出“Rockusb Device”，意味着驱动已经安装成功



4. 打开“RKDevTool”软件



注意：软件中的参数默认如图所示，正常情况下仅需要设置④“.img 系统实际的路径”即可。如果您软件中的参数与图中不一致，请手动修改为一致。

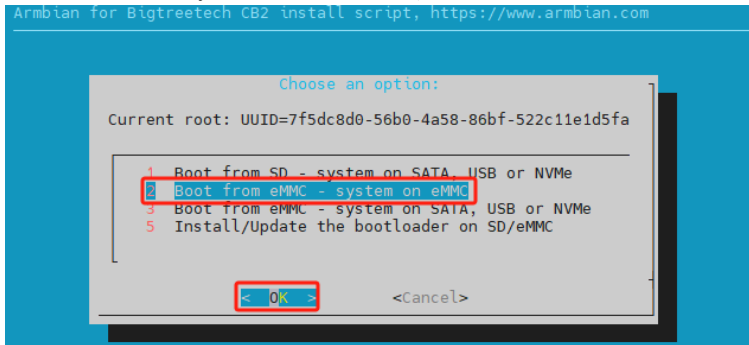
#	<input type="checkbox"/>	Storage	Address	Name	Path
1	<input checked="" type="checkbox"/>		0x00000000	Loader	..\MiniLoaderAll.bin
2	<input checked="" type="checkbox"/>	EMMC	0x00000000	System	.img 系统实际的路径

- ① 找到下载的工具所在的路径
 - ② 打开 RKDevTool 工具
 - ③ 软件会识别出一个“LOADER”或者“MASKROOM”的设备
 - ④ 选择要烧录的系统（系统镜像需要提前解压为.img 文件，此工具不支持直接烧录压缩后的.xz 文件）
 - ⑤ 勾选“Write by Address”
 - ⑥ 点击“Run”，开始烧录系统
 - ⑦ “Download image OK”意味着系统已经烧录成功
5. 烧录完成后，请将 USB OTG 的拨码开关拨到 OFF 挡位，此时即可正常开机使用了。注意：eMMC 内的文件无法像 MicroSD 卡那样直接被电脑访问，所以无法通过修改 system.cfg 配置文件的方式配置 WiFi 网络，只能用网线或者 USB 转 UART 连接终端，然后通过终端配置。

4.3.2 使用 MicroSD 卡烧录系统到 eMMC

1. 先将系统烧录到 MicroSD 卡中，然后将 MicroSD 卡插到主板的卡槽，然后等待系统启动。
2. 通过网线，WiFi 或者 USB 转 UART 连接到系统的终端，登录系统
login: biqu
password: biqu

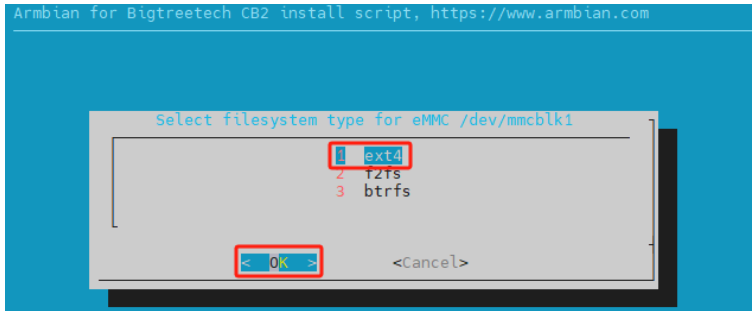
- (1) 运行 `sudo nand-sata-install` 命令，在弹出的界面中，选择 “2 Boot From eMMC - system on eMMC”，然后选择 “OK”



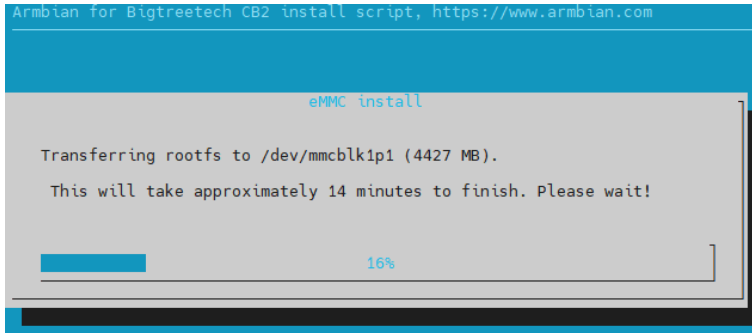
- (2) 选择 “Yes”，开始擦除并烧录系统到 eMMC



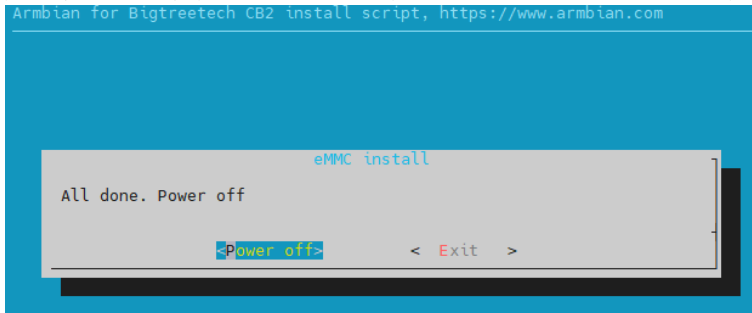
- (3) 选择文件系统为 “1 ext4”，然后选择 “OK”



- (4) 等待烧录完成



- (5) 烧录完成后会弹窗提示是否关机，选择 “Power off” 关机



- (6) 关机后断电，然后拔出 MicroSD 卡，重新再通电即可从 eMMC 启动

4.4 擦除 eMMC

当不使用 eMMC，而使用 MicroSD 卡作为系统卡时，最好将 eMMC 的数据擦除，以免主板错误的从 eMMC 启动。

4.4.1 使用 UMS 擦除 eMMC (Windows)

若 eMMC 中已经烧录过 **V3.0.1 及其之后** 版本的系统，电脑会将 eMMC 识别为 UMS 设备（类似 U 盘一样的设备）。UMS 模式相对于 Loader 模式有以下好处：

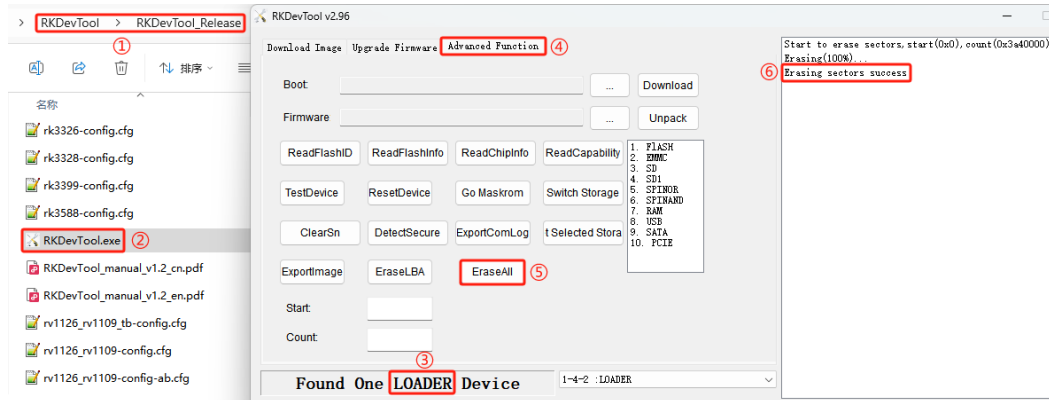
- 可以直接修改 /boot/ 分区中的配置信息
- 可以直接像 Micro SD 卡那样，直接烧录系统到 eMMC
- 可以通过软件擦除 eMMC 中的所有内容

- 参照“4.3.1 使用 RKDevTool 烧录系统到 eMMC”中的步骤，将主板连接到电脑。
- 电脑会将 eMMC 识别为 UMS 设备。
- 安装 [SD Card Formatter](#) 软件，格式化 eMMC 的 UMS 设备。（请不要直接使用 windows 系统提供的格式化功能，因为它无法完全擦除 eMMC 中的数据）

4.4.2 使用 RKDevTool 擦除 eMMC (Windows)

若 eMMC 中已经烧录过 V2.0.0 及其之前版本的系统，电脑会将 eMMC 识别为 Loader 模式的设备。

1. 参照“4.3.1 使用 RKDevTool 烧录系统到 eMMC”中的步骤，将主板连接到电脑
2. 打开“RKDevTool”软件



- ① 找到下载的工具所在的路径
- ② 打开 RKDevTool 工具
- ③ 软件会识别出一个“LOADER”的设备，如果是“MASKROOM”则说明 eMMC 中没有数据，不需要擦除
- ④ 点击“Advanced Function”
- ⑤ 点击“EraseAll”开始擦除 eMMC 中的数据
- ⑥ “Erasing sectors success”擦除完成

4.4.3 从 MicroSD 卡启动系统后擦除 eMMC

1. 参照“4.3.2 使用 MicroSD 卡烧录系统到 eMMC”中的步骤，登录到系统终端
2. 运行 `sudo mkfs /dev/mmcblk1` 命令，然后输入“y”确认。

```
biqu@BTT-CB2:~$ sudo mkfs /dev/mmcblk1
mke2fs 1.46.2 (28-Feb-2021)
/dev/mmcblk1 contains a ext2 file system
created on Wed Apr 24 06:30:21 2024
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 7634944 4k blocks and 1908736 inodes
Filesystem UUID: 51dbd34e-8aef-4f29-9f98-e535341ed141
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

biqu@BTT-CB2:~$
```

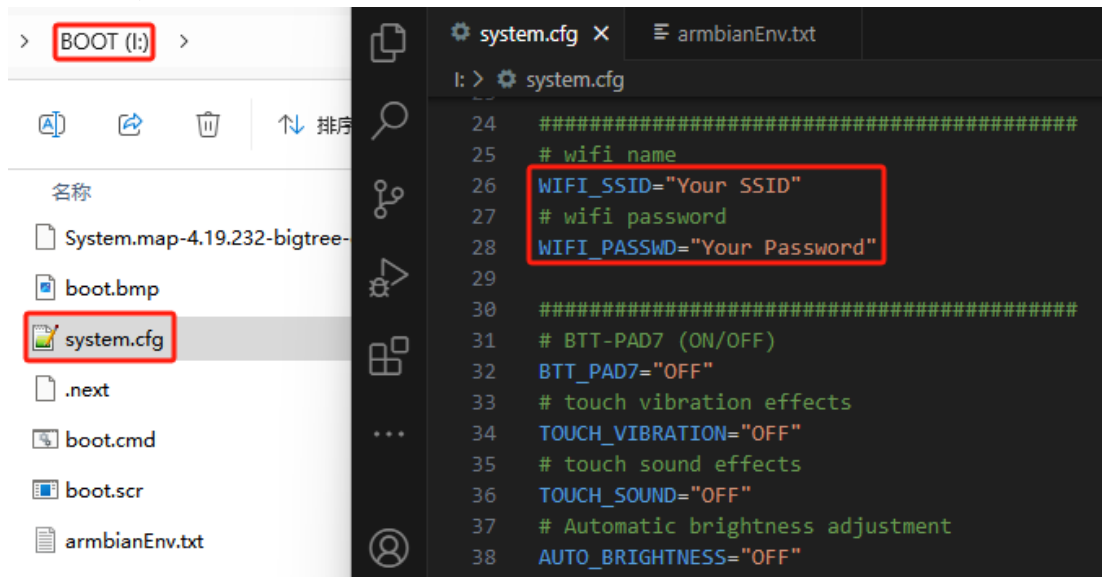
五、系统配置

5.1 使用网线

网线即插即用，不需要额外的设置

5.2 设置 WiFi

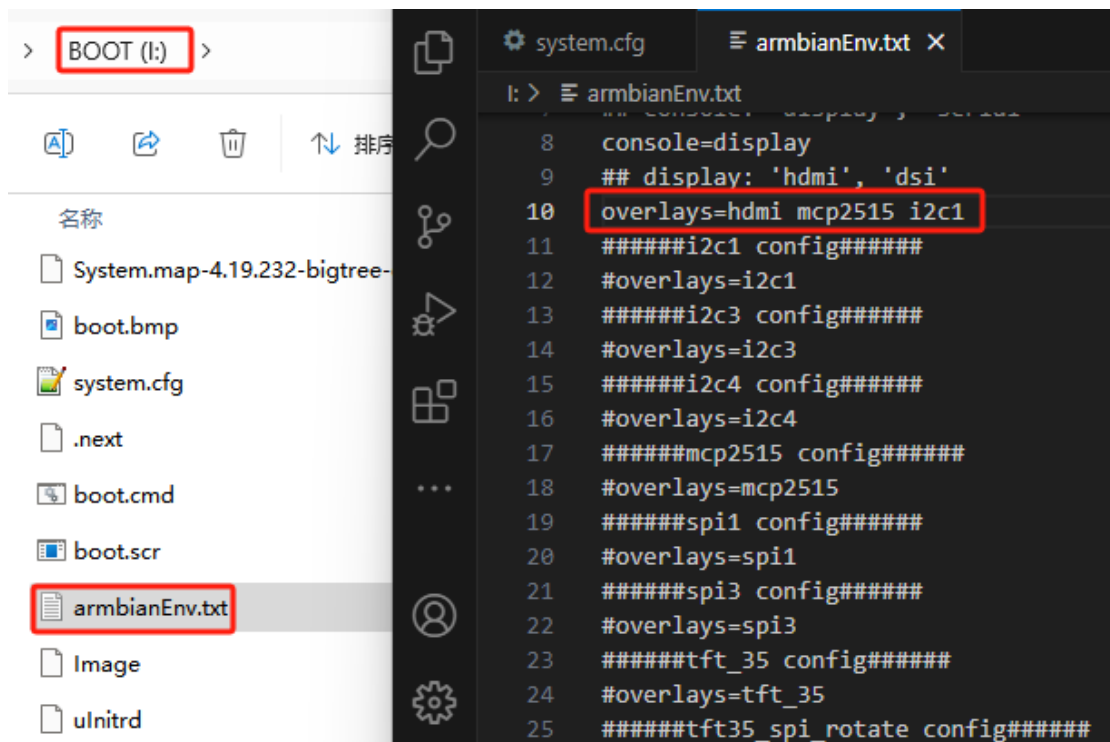
系统镜像烧录完成后，MicroSD 卡会有一个被电脑识别的 FAT32 分区，此分区下有个名为“system.cfg”的配置文件，打开后将 Your SSID 替换为实际的 WIFI 名称，Your Password 替换为实际的密码



5.3 配置 overlays

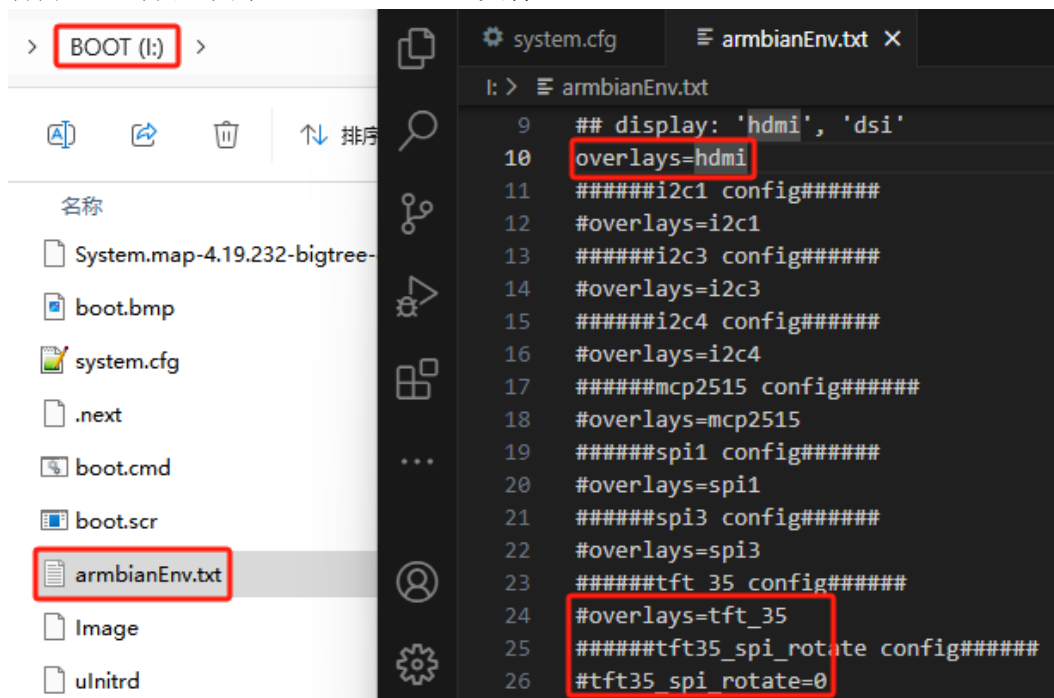
打开 BOOT 分区下的 armbianEnv.txt 文件, 设置 overlays 的值。配置文件中同一时间仅支持打开一行 overlays, 如果打开了多行 overlays 的配置, 只会生效最后一行的配置。如果有打开多个 overlays 配置的需求, 可以将多个配置的内容放在同一行 overlays 后面, 并且多个配置中间用一个空格隔开。例如我们需要同时使用 DSI 屏幕、mcp2515 SPI 转 CAN 模块, 和 I2C1:

```
overlays=dsi mcp2515 i2c1
```



5.4 配置显示屏

1. 打开 BOOT 分区下的 armbianEnv.txt 文件

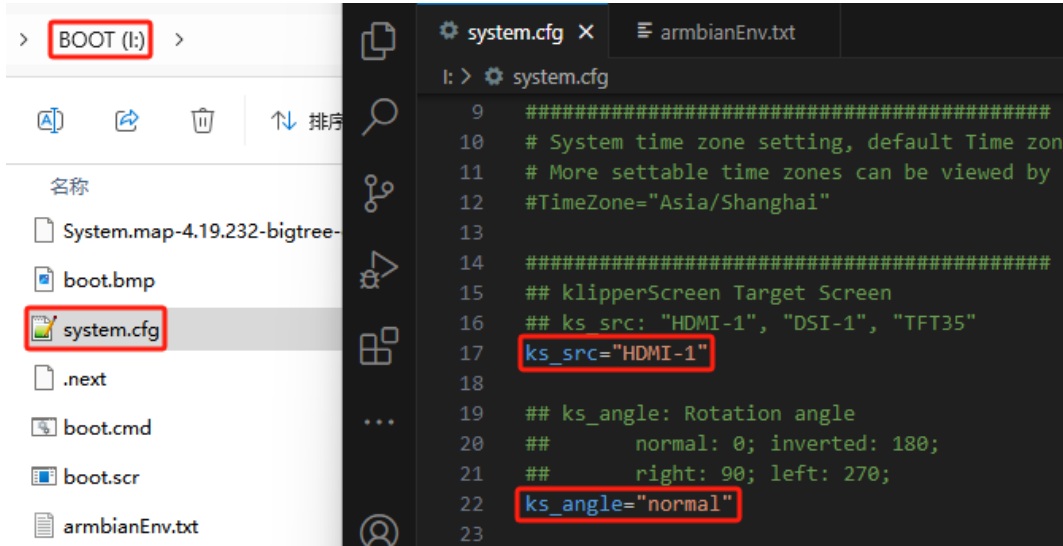


2. overlays 默认设置为 hdmi，代表系统默认使用 hdmi 屏幕。可以将其修改为实际使用的屏幕，可设置的选项如下：
 - “hdmi” : [HDMI 接口的屏幕](#)
 - “dsi” : [DSI 接口的屏幕](#)
 - “tft_35” : [SPI 接口 3.5 寸屏幕](#)

其中“tft_35”还有一个参数“tft35_spi_rotate”在系统级旋转显示界面，默认的“0”代表不旋转，可使用的参数还有“90”，“180”，“270”。

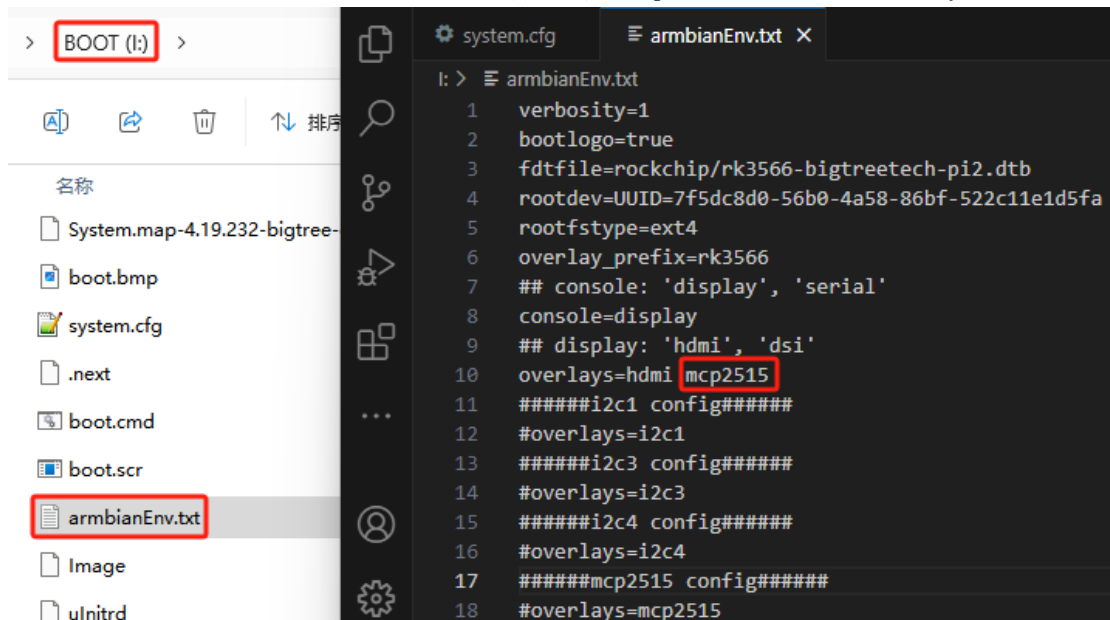
注意：屏幕只能选择使用其中的一个，无法同时使用多个屏幕

3. 设置 KlipperScreen，打开 BOOT 分区下的 system.cfg 文件，设置屏幕的类型“ks_src”，和旋转角度“ks_angle”



5.5 SPI 转 CAN 的使用

打开 BOOT 分区下的 armbianEnv.txt 文件，将“mcp2515”添加到 overlays 的配置中



5.6 CSI 相机使用及 crowsnest 配置

无论是 rpi v1.3 的 ov5647 还是 rpi v2 的 imx219 均不需要在 armbianEnv.txt 文件中配置 overlays，即插即用。

crownsnest.conf 文件中的配置如下图所示：

device: /dev/video0 # CSI 相机的节点固定为 video0

custom_flags: --format=UYVY # 当前系统 CSI 相机不支持默认的 YUYV，需要设置为支持的 UYVY 格式

```
[crownsnest]
log_path: /home/biqu/printer_data/logs/crownsnest.log
log_level: verbose # Valid Options are quiet/verbose/debug
delete_log: false # Deletes log on every restart, if set to true
no_proxy: false

[cam 1]
mode: ustreamer # ustreamer - Provides mjpg and snapshots. (All devices)
# camera-streamer - Provides webrtc, mjpg and snapshots. (rpi + Raspi OS based only)
enable_rtsp: false # If camera-streamer is used, this enables also usage of an rtsp server
rtsp_port: 8554 # Set different ports for each device!
port: 8080 # HTTP/MJPEG Stream/Snapshot Port
# See Log for available ...
device: /dev/video0 # widthxheight format
resolution: 640x480 # If Hardware Supports this it will be forced, otherwise ignored/coerced.
max_fps: 15 # You can run the Stream Services with custom flags.
custom_flags: --format=UYVY # Add v4l2-ctl parameters to setup your camera, see Log what your cam is capable of.
#v4l2ctl:
```

5.7 蓝牙的使用

1. 扫描蓝牙设备，输入如下命令，出现如下列表的蓝牙设备，如下图

bluetoothctl --timeout 15 scan on

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisi
root@Hurakan:~# bluetoothctl --timeout 15 scan on
Discovery started
[CHG] Controller 50:41:1C:F1:1B:DD Discovering: yes
[NEW] Device 61:81:3F:1B:B0:79 61-81-3F-1B-B0-79
[NEW] Device 67:06:15:E1:7A:62 67-06-15-E1-7A-62
[NEW] Device 78:77:40:B5:D8:02 78-77-40-B5-D8-02
[NEW] Device 61:C5:14:23:27:CC 61-C5-14-23-27-CC
[NEW] Device 7B:42:68:66:19:07 廖金花
[NEW] Device 6F:D8:78:63:4F:CD 6F-D8-78-63-4F-CD
[NEW] Device 4C:E8:2E:37:02:CE 4C-E8-2E-37-02-CE
[NEW] Device 51:22:49:FC:CF:C1 51-22-49-FC-CF-C1
[NEW] Device 73:B9:DB:2D:F1:08 73-B9-DB-2D-F1-08
[NEW] Device 54:8B:65:80:80:85 54-8B-65-80-80-85
```

2. 找到自己的蓝牙设备，比如我的蓝牙设备名字是 HONOR xSport PRO，在设备列表中找到对应的蓝牙 MAC ID 如下图

```
[CHG] Device 42:70:F4:03:91:BA ManufacturerData Value:
10 07 7a 1f 3b 4d ef 5c 68 ..z.;M.\h
[CHG] Device 4E:B0:A9:B4:33:11 RSSI: -75
[CHG] Device 45:69:88:0D:E0:7B RSSI: -92
[CHG] Device 7F:E1:35:CF:F8:A3 RSSI: -77
[CHG] Device 4E:B0:A9:B4:33:11 RSSI: -88
[CHG] Device 04:7A:0B:19:E7:AF ManufacturerData Key: 0x038f
[CHG] Device 04:7A:0B:19:E7:AF ManufacturerData Value:
0a 10 ff ff 64 93 15 36 c3 5c de 20 11 08 08 .....d..6.\. ...
10 17 25 34 ..%4
[NEW] Device E0:9D:FA:50:CD:4F HONOR xSport PRO
[CHG] Device 04:7A:0B:19:E7:AF Class: 0x000a0110
[CHG] Device 04:7A:0B:19:E7:AF Icon: computer
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000fdaa-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[NEW] Device 90:0F:0C:2F:50:C2 XIAOYONG
root@Hurakan:~#
```


3. 连接蓝牙设备，输入如下命令，连接成功如下图

```
bluetoothctl connect E0:9D:FA:50:CD:4F
```

```
[CHG] Device 90:0F:0C:2F:50:C2 UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
root@bigtreotech-cb2:~# bluetoothctl connect E0:9D:FA:50:CD:4F
Attempting to connect to E0:9D:FA:50:CD:4F
[CHG] Device E0:9D:FA:50:CD:4F Connected: yes
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F ServicesResolved: yes
[CHG] Device E0:9D:FA:50:CD:4F Paired: yes
Connection successful
root@bigtreotech-cb2:~#
```

- (1) 若出现如下图输出，请重新打开蓝牙设备，然后重新按 1 和 2 的步骤连接蓝牙设备

```
[CHG] Device 04:7A:0B:19:E7:AF Class: 0x000a0110
[CHG] Device 04:7A:0B:19:E7:AF Icon: computer
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000fdad-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 04:7A:0B:19:E7:AF UUIDs: 00001200-0000-1000-8000-00805f9b34fb
root@bigtreotech-cb2:~# bluetoothctl connect E0:9D:FA:50:CD:4F
Device E0:9D:FA:50:CD:4F not available
root@bigtreotech-cb2:~#
root@bigtreotech-cb2:~#
root@bigtreotech-cb2:~#
root@bigtreotech-cb2:~#
root@bigtreotech-cb2:~# bluetoothctl --timeout 15 scan on
Discovery started
[CHG] Controller 50:41:1C:F1:B:D0 Discovering: yes
[NEW] Device 4F:8C:BC:B9:0B:27 4F-8C-BC-B9-0B-27
```

- (2) 若如下图输出，请输入如下命令，然后重新进行 1 和 2 步骤

```
bluetoothctl remove E0:9D:FA:50:CD:4F (您的蓝牙设备对应的 MAC ID)
```

```
rfskill block bluetooth
```

```
sleep 3s
```

```
rfskill unblock bluetooth
```

```
pulseaudio -k
```

```
pulseaudio --start
```

```
[DEL] Device 40:60:97:F3:85:D6 40-60-97-F3-85-D6
root@bigtreotech-cb2:~# bluetoothctl connect E0:9D:FA:50:CD:4F
Attempting to connect to E0:9D:FA:50:CD:4F
[CHG] Device E0:9D:FA:50:CD:4F Connected: yes
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device E0:9D:FA:50:CD:4F ServicesResolved: yes
Failed to connect: org.bluez.Error.Failed
root@bigtreotech-cb2:~# bluetoothctl remove E0:9D:FA:50:CD:4F
[DEL] Device E0:9D:FA:50:CD:4F HONOR xSport PRO
Device has been removed
root@bigtreotech-cb2:~# rfskill block bluetooth
```

4. 蓝牙使用中途退出语音播放功能，如果不能再次使用蓝牙，需要手动删除对应的播放进程，用 ps 命令查看播放的进程号，然后用 kill -9 进程号 删除对应的播放进程。如下图所示

```
biqu@bigtreotech-cb2:~$ ps
  PID TTY          TIME CMD
 2094 pts/0        00:00:00 bash
 2270 pts/0        00:00:00 aplay
 2347 pts/0        00:00:00 ps
biqu@bigtreotech-cb2:~$ kill -9 2270
```


5.8 3.5mm 圆口耳机设置

1. 输入命令：

aplay -l

查看对应的声卡，如下图所示：（由图所示耳机口的声卡对应的是 card 0）

```

[ General system configuration (beta): armbian-config ]

Last login: Wed Apr 10 02:18:28 UTC 2024 on tty1
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk817-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi-0]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
card 1: rockchipbt [rockchip,bt], device 0: fe420000.i2s-bt-sco-pcm bt-sco-pcm-0 [fe420000.i2s-bt-sco-pcm-0]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#

```

2. 输入命令：

amixer -c 0 contents（0 表示的上述的 aplay -l 所找到的 card 0）

查看播放通道和录音通道设置，如下图所示：

```

root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~# amixer -c 0 contents
numid=3,iface=MIXER,name='PCM'
; type=INTEGER,access=rw---R--,values=2,min=0,max=252,step=0
; values=255,255
| dBscale-min=-95.00dB,step=0.37dB,mute=0
numid=2,iface=MIXER,name='Capture MIC Path'
; type=ENUMERATED,access=rw-----,values=1,items=2
; Item #0 'MIC OFF'
; Item #1 'Main Mic'
; values=0
numid=4,iface=MIXER,name='Capture Volume'
; type=INTEGER,access=rw---R--,values=2,min=0,max=255,step=0
; values=255,255
| dBscale-min=-95.00dB,step=0.37dB,mute=0
numid=1,iface=MIXER,name='Playback Path'
; type=ENUMERATED,access=rw-----,values=11,items=11
; Item #0 'OFF'
; Item #1 'RCV'
; Item #2 'SPK'
; Item #3 'HP'
; Item #4 'HP_NO_MIC'
; Item #5 'BT'
; Item #6 'SPK_HP'
; Item #7 'RING_SPK'
; Item #8 'RING_HP'
; Item #9 'RING_HP_NO_MIC'
; Item #10 'RING_SPK_HP'
; values=0
root@bigtreetech-cb2:~#

```

3. 输入命令：

amixer -c 0 cset numid=1 3

设置播放通道，如下图所示：

```

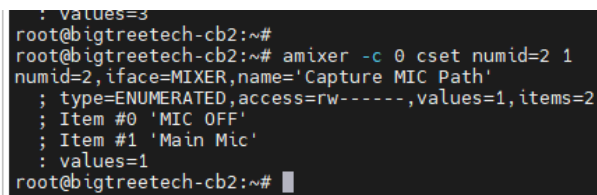
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~#
root@bigtreetech-cb2:~# amixer -c 0 cset numid=1 3
numid=1,iface=MIXER,name='Playback Path'
; type=ENUMERATED,access=rw-----,values=11,items=11
; Item #0 'OFF'
; Item #1 'RCV'
; Item #2 'SPK'
; Item #3 'HP'
; Item #4 'HP_NO_MIC'
; Item #5 'BT'
; Item #6 'SPK_HP'
; Item #7 'RING_SPK'
; Item #8 'RING_HP'
; Item #9 'RING_HP_NO_MIC'
; Item #10 'RING_SPK_HP'
; values=3
root@bigtreetech-cb2:~#

```

4. 输入命令:

```
amixer -c 0 cset numid=2 1
```

设置录音通道，如下图所示：



```
: values=3
root@bigtreotech-cb2:~#
root@bigtreotech-cb2:~# amixer -c 0 cset numid=2 1
numid=2,iface=MIXER,name='Capture MIC Path'
; type=ENUMERATED,access=rw-----,values=1,items=2
; Item #0 'MIC OFF'
; Item #1 'Main Mic'
: values=1
root@bigtreotech-cb2:~#
```

5. 输入如下命令播放音频，音频文件目录 xxx 加音频文件名 xxxxx.wav

```
aplay -D plughw:0,0 /xxx/xxxxx.wav
```

6. 输入如下命令录音（其中 10 表示录音 10 秒），录音存放的目录是 xxx，文件名 xxxxx.wav

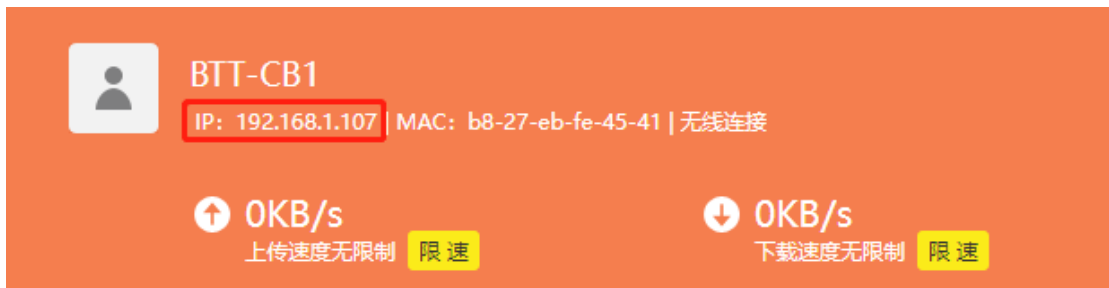
```
sudo arecord -Dhw:0,0 -d 10 -f cd -r 44100 -c 2 -t wav /xxx/xxxxx.wav
```

7. 输入如下命令播放录音

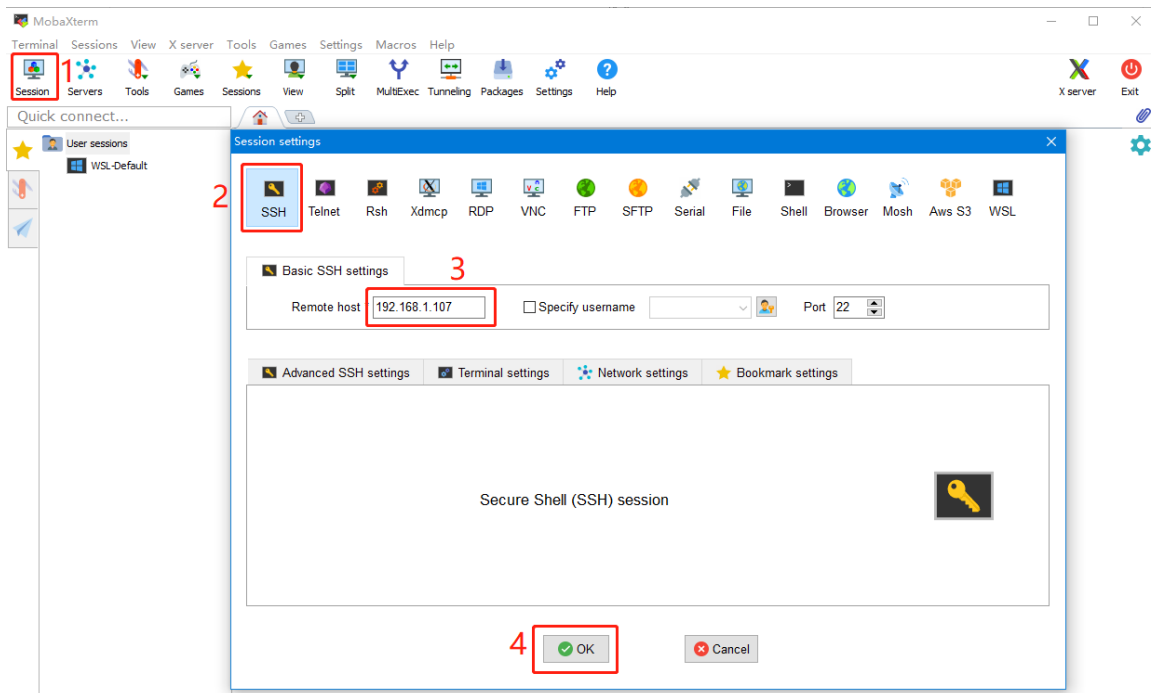
```
aplay -D plughw:0,0 /xxx/xxxxx.wav
```

六、SSH 连接设备

1. 安装 ssh 软件 MobaXterm: <https://mobaxterm.mobatek.net/download-home-edition.html>
2. 通电后等待系统启动, 大概 1~2 分钟
3. 设备连上 WIFI 或者插上网线后, 会被自动分配一个 IP
4. 进入路由器管理界面找到设备的 IP (这里应为 BTT-CB2)



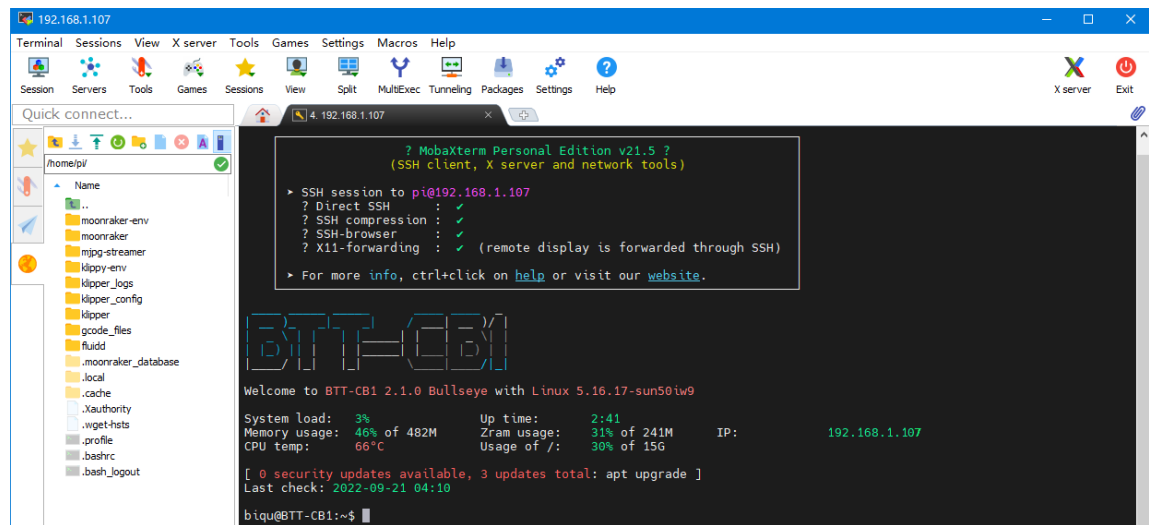
5. 打开已经安装的 MobaXterm 软件, 点击 “Session”, 在弹出的窗口中点击 “SSH”, 在 Remote host 一栏中输入设备的 IP 地址, 点击 “OK” (注意: 电脑和设备必须要在同一个局域网下)



6. 输入登录名和登录密码进入 SSH 终端界面

登录名 login as: biqu

密码: biqu



七、注意事项

1. 上电后大概 10s 左右，系统进入 kernel 阶段。此时 power 灯常亮，act 灯会不断的闪烁，代表系统在正常运行。

2. Klipper 系统

root 管理员：

login: root

password: root

biqu 普通用户：

login: biqu

password: biqu

Minimal 系统

root 管理员：

login: root

password: root

Minimal 系统是标准的 Armbian 启动流程，只有管理员帐户“root”。首次启动后，系统会引导用户在终端中创建自己的普通账户。

3. PCIe M.2 接口不支持热插拔，需要预先插上固态硬盘才能识别到设备。
4. 使用 eMMC 启动时，不要插 MicroSD 卡。使用 MicroSD 卡启动时，需要将 eMMC 中的数据擦除。

如果您还需要此产品的其他资源，可以到 <https://github.com/bigtreetech/> 上自行查找，如果无法找到您所需的资源，可以联系我们的售后支持（service005@biqu3d.com）。

若您使用中还遇到别的问题，欢迎您联系我们，我们定会细心为您解答；若您对我们的产品有什么好的意见或建议，也欢迎您回馈给我们，我们也会仔细斟酌您的意见或建议，感谢您选择 BIGTREETECH 制品，谢谢！