

双椒派™ E2000D 开发板使用说明

(V1.2)

2023 年 3 月 17 日

版权声明：

本文档用于指导用户的相关应用和开发工作，版权归属飞腾信息技术有限公司和武汉双椒派信息技术有限责任公司共同所有，受法律保护。任何未经书面许可的公开、复制、转载、篡改行为将被依法追究法律责任。

免责声明：

本文档仅提供阶段性数据，并不保证该等数据的准确性及完整性。飞腾信息技术有限公司和武汉双椒派信息技术有限责任公司共同对此文档内容享有最终解释权，且保留随时更新、补充和修订的权利。

如有技术问题，可联系武汉双椒派信息技术有限责任公司获取支持，因不当使用本文档造成的损失，本公司概不承担任何责任。

修改历史

版本	位置	修改内容	修改人
V0.1		创建文档	
V1.1		更新了 V1.1 板卡的信息	
V1.2		更新了 V1.2 板卡的信息	

目 录

1. 概况	4
2. 装箱清单	4
3. 硬件资源	4
3.1 原理构成	5
3.1.1 CPU 和存储	5
3.1.2 电源和时钟	5
3.1.3 高速接口	5
3.1.4 视频音频输出接口	6
3.1.5 低速接口	6
3.1.6 MicroSD 卡接口	6
3.1.7 音频输出接口	6
3.2 接口定义	6
3.2.1 40Pin 低速接口	7
3.2.2 FLASH 烧写插座	8
3.2.3 以太网插座	9
3.2.4 USB 3.0 插座	9
3.2.5 USB 2.0 插座	10
3.2.6 MiniDP 插座	10
3.2.7 电源输入插座	10
3.2.8 天线插座	11
3.2.9 音频输入输出插针	11
3.2.10 CANBus	11
3.2.11 调试串口	11
4. 使用方法	12
4.1 准备工作	12
4.2 上电启动	12
4.3 固件和操作系统烧写	12
5. 注意事项	13
6. 故障排除	14
7. 附录	15
7.1 启动提示信息	15

1. 概况

双椒派开发板是基于飞腾 E2000D 处理器的开发板，主要面向教育领域，在有限的成本下提供尽量丰富的功能。CPU 内含一个当前主流的 ARM V8 内核，主频达到 1.5GHz，内存 4GB，并且具有 USB 2.0、以太网等高速接口，GPIO、串口、I2C 等常见低速接口，低速接口所在的 40pin 连接器与树莓派基本兼容，以利用现有的扩展模块。本板还有采用飞腾 E2000S 处理器的型号，外形与本板相同，CPU 采用单核型号，功耗更低。本开发板的技术指标如下：

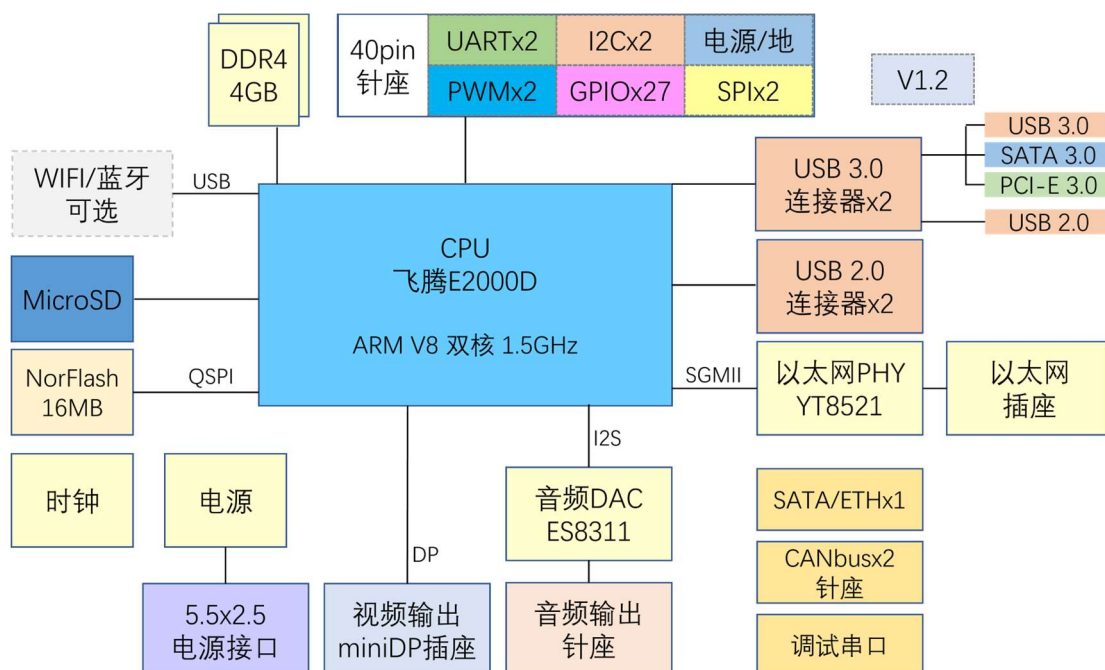
参数名称	参数值
	E2000D
CPU	ARM V8 双核 1.5GHz
内存	DDR4 4GB
FLASH	SPI FLASH 16MB
存储卡	TF (MicroSD) 卡插座
通用 I/O	40pin 通用连接器，包括 3x SPI，4xUART 串口（其中两个与 I2C 复用），2xI2C，2xPWM 输出，24xGPIO（与前述接口复用），5V 电源，3.3V 电源
以太网口	10/100/1000M 以太网，RJ45 接口，第二网口从 SATA 插座引出，与 SATA 复用
WIFI/蓝牙	USB 可选
USB 3.0	USB 3.0 Type-A 接口 2 个 或 PCIE Gen3 x1 接口 2 个（与 USB 插座复用）
USB 2.0	USB 2.0 OTG Type-A 接口 2 个
SATA	SATA 3.0 接口
CANBUS	CAN-FD 接口 2 个，针座
JTAG	JTAG 测试焊盘
音频输出	I2S 双声道 音频输入、输出，针座
电源输入	5.5x2.5 圆插座
功耗	<15W
电源	5V/3A（不包含外接 USB 设备功耗）
板卡尺寸	113mm x 72mm x 20mm
重量	
工作温度	0 到 70℃
存储温度	-10 到 80℃

2. 装箱清单

编号	设备或部件	数量	备注
1	开发板	1 块	
2	5V/3A 电源适配器，5.5x2.5 圆口电缆	1 条	
3	mmc 卡 32GB，带操作系统（选配）	1 个	
4	双椒派 E2000D 开发板使用说明书	1 份	

3. 硬件资源

3.1 原理构成



3.1.1 CPU 和存储

本板以 E2000D CPU 为核心，配备了两颗 DDR4 内存颗粒作为随机存储器，用于运行程序，总内存容量 4GB。一片 16MB 容量的 SPI NOR FLASH 用于存储启动程序，支持 QSPI 模式，启动速度更快。如果操作系统比较大，超出了 FLASH 的容量，可以存放到优盘或者 TF 卡中。CPU 首先从 FLASH 启动，或者从 FLASH 启动 uboot 后再从优盘加载操作系统，甚至直接从优盘或者 TF 卡启动。

3.1.2 电源和时钟

芯片周围有时钟和电源电路，其中时钟电路负责为 CPU 提供 50MHz 系统时钟和 100MHz 高速接口时钟等。电源电路负责将从电源接口或者 40pin 插座输入的 5V 电源转换为 0.8V、1.2V、1.8V、2.5V、3.3V 等各种电压，供给 CPU 和板上芯片。

3.1.3 高速接口

本板带有两个 USB 3.0 接口，以 Type-A 插座形式引出，插座上还有两路 USB 2.0 接口，以便兼容 USB 2.0 设备。USB 3.0 接口还可通过修改软件配置为 PCI-e 3.0 x1 lane 的信号，如果将接口通过转接板转为 PCI-E、M.2、Mini-PCIE 等形式的插槽后，可连接显卡、人工智能加速卡、固态硬盘等设备。还可通过修改软件配置为 SATA 3.0 接口，可以连接硬盘。

本板还有 3 个 USB 2.0 接口，其中两个从 USB2.0 Type-A 插座引出，支持 OTG，一个用来连接板上内置的 WIFI/蓝牙模块。

本板有一路以太网接口，以 RJ45 网口插座形式引出，支持 10/100/1000M 自适应，以太网 MAC 为 CPU 内置，以太网 PHY 为外置单独芯片。

本板有一个 SATA 连接器，可以连接 SATA 接口的硬盘。本板没有提供硬盘的电源，需要用户自行给硬盘供电。这个 SATA 信号还可以通过修改软件配置为 SGMII 接口，如果外部连接一个以太网物理层接口芯片（PHY 芯片），则可以扩展出第二路以太网。

3.1.4 视频音频输出接口

本板支持一路 Display Port 视频输出，附带音频信号输出，插座形式为 Mini-DP。支持 DisplayPort 1.4 / Embedded DisplayPort 1.4 协议，支持音频输出。在 DP 接口运行在最高链路速度 HBR2（5.4Gbps）的情况下，支持最大分辨率 1920x1080@60MHz，实际应用中由于受到板材和电缆质量等影响，链路速度降低到 HBR1，则无法达到上述分辨率。建议通过电缆直连支持 DP 接口的显示器，如果通过转换器转成 HDMI，则可能出现不兼容无显示的情况。

3.1.5 低速接口

本板的低速接口包括 1 个单独的三线调试串口，40p 针座中包含，1 个普通三线串口，还有 2 个串口和 2 个 I2C 接口、3 个 SPI 接口、2 个脉宽调制（PWM）信号输出、24 个 GPIO 信号。这么多信号并不能同时使用，因为存在多个信号共用一个引脚的现象，这也是现代 CPU 的通用做法。E2000 处理器提供了一种 MIO 控制器，可以配置成串口或者 I2C 口，本板的 2 个串口和 I2C 复用接口就是由 MIO 控制器提供的。

低速信号所在 40p 连接器还有 5V 电源输入/输出，3.3V 电源输出、地引脚。其中 5V 电源引脚直接与电源插座并联，在从圆口电源插座供电的情况下，可以向外输出 5V 电源，如果开发板的 5V 电源功耗较大，比如 USB 口挂载了移动硬盘等大功率设备，还可从 40p 插座的 5V 引脚向开发板辅助供电。低速接口的详细定义见接口定义章节。

3.1.6 MicroSD 卡接口

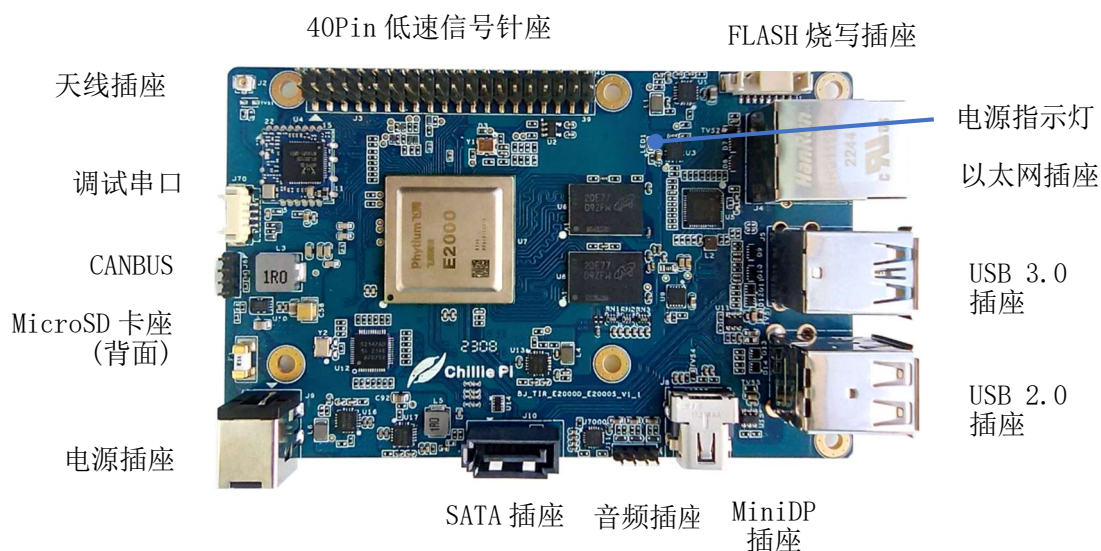
本板有一个 MicroSD 卡（或者称为 TF 卡）插座，支持 SD 3.0 协议，用于存储操作系统。

3.1.7 音频输出接口

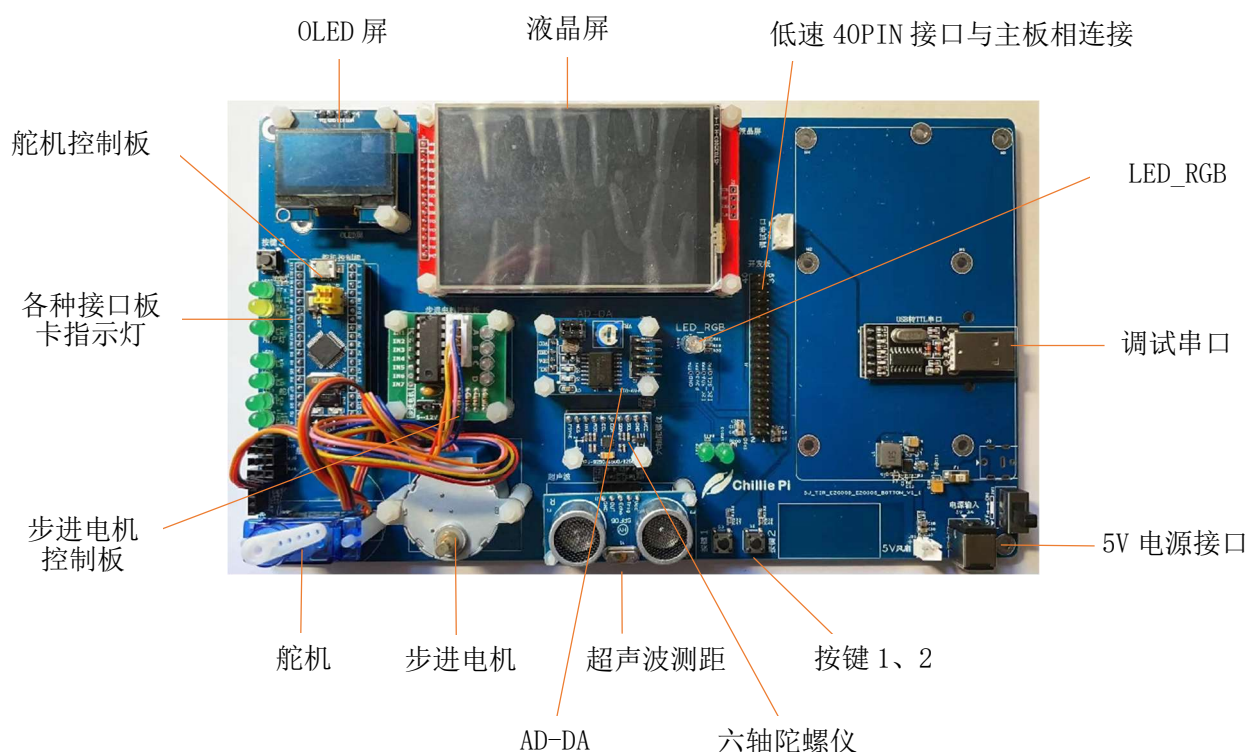
本板含有一个音频数模转换器，支持双声道音频输出和话筒输入，为了节约 PCB 面积引出了针座，没有 3.5mm 插座。音频 DAC 芯片的型号是 ES8311

3.2 接口定义

本板的对外接口分布如下：

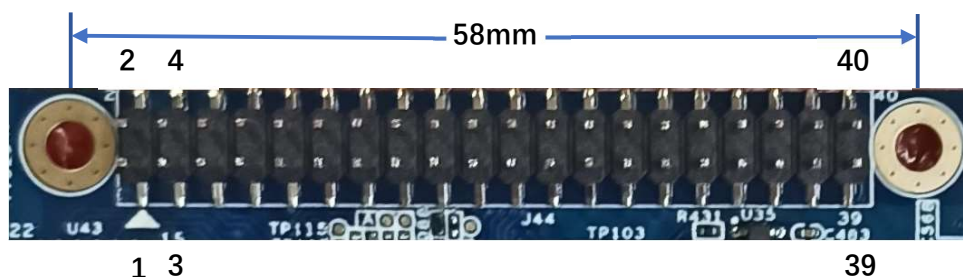


底板接口定义：



3.2.1 40Pin 低速接口

所有低速接口合并成一个 40pin 双排针插座，间距 2.54mm，其引脚分配和两侧固定孔的位置都尽量和树莓派保持兼容。连接器第一脚的位置如下图，电路板丝印上有白色三角标识。



40pin 插座 J44 的引脚分配如下表，引脚名称实际就是所连接的 CPU 的引脚的名称，其中表示出所有复用的功能，一些开发板用到的特殊功能用彩色字体标出，相同功能的接口颜色相同，以便于识别。

双椒派 D 引脚名	树莓派引脚名	引脚号	RPI 引脚名	双椒派 D 引脚名
3.3V 输出	3.3V VDC	1 2	5.0 VDC	5V 输出/输入

NF_RB_N1/JTAGM_TDO/ MI09_B / LBC_RB_N3/ GPI05_1	GPI02/I2C0_ SDA1	3	4	5.0 VDC	5V 输出/输入
NF_CE_N1/JTAGM_TDI/ MI09_A / LBC_RB_N2/ GPI05_0	GPI03/I2C0_ SCL1	5	6	GND	GND
NF_DATA4/ SPIM0_CSN2 /LBC_CLK/ GPI04_11	GPI04/GPCLK 0	7	8	GPI014/UAR T_TxD	SE_GPI09/ PWM1 / GPI00_15
GND	GND	9	0	GPI015/UAR T_RxD	UART2_RXD
LBC_ALE/RGMII0_TXD0/ SPIM3_SCLK / GPI05_12	GPI017	1	1	GPI018/PWM /SPI1_CEO	UART2_TXD /PCIE0_PRSENT2/ SPIM2_CSN3 / GPI03_0
LBC_CS_N0/RGMII0_TXD1/ SPIM3_TXD / GPI05_13	GPI027	1	1	GND	
LBC_CS_N1/RGMII0_RXD0/ SPIM3_RXD / GPI05_14	GPI022	1	1	GPI023	NF_REN_WRN/SD0_DATA6/LBC_CS_N6/ GPI04_4
3.3V 输出	3.3V VDC	1	1	GPI024	NF_DATA2/ SPIM0_CSN0 /LBC_PAR0/ GPI04_9
SPIM2_TXD /MDIO1/LBC_RB_N5/ GPI03_4	GPI010/SPI0_ _MOSI	1	2	GND	GND
SPIM2_RXD /MDC0/LBC_CS_N4/ GPI03_5	GPI09/SPI0_ MISO	2	2	GPI025	NF_CLE/SD0_DATA7/LBC_CS_N7/ GPI04_5
SPIM2_SCLK /SPIS_SCLK/ TACH1_1 /MDC1/ GPI03_3	GPI011/SPI0_ _SCLK	2	2	GPI08/SPI0 _CEO	SPIM2_CSN0 /MDIO0/LBC_CS_N5/ GPI03_6
GND	GND	2	2	GPI07/SPI0 _CE1	UART2_RTS_N/PCIE0_PRSENT3/ SPIM2_CSN1 /MIO10_B/ GPI03_2
SE_GPI026/ MI06_B / GPI02_4	GPI00/I2C_I D_SDA0	2	2	GPI01/I2C_ ID_SCL0	SE_GPI025/ MI06_A / GPI02_3
NF_WEN_CLK/SD0_DATA5/LBC_RB_N7/ GPI04_3	GPI06/GPCLK 1	2	3	GND	GND
SD0_CLK/ GPI04_1	GPI06/GPCLK 2	3	3	GPI012/PWM 0	SE_GPI011/ PWM2 / GPI01_1
SE_GPI07/ PWM0 / GPI00_13	GPI013/PWM1	3	3	GND	GND
NF_DATA1/SD0_WP_N/ SPIM0_RXD /LBC_WE_N1/ GPI04_8	GPI019/PWM1 /SPI1_MISO	3	3	GPI016/SPI 1_CEO	NF_DATA3/PCIE1_PRSENT/ SPIM0_CSN1 /LBC_PAR1/ GPI04_10
NF_DATA5/PCIE2_PRSENT/ SPIM0_CSN3 /LBC_CS_N2/ GPI04_12	GPI026	3	3	GPI020/GPI 020_MOSI	NF_DATA0/SD0_VOLT1/ SPIM0_TXD /LBC_WE_N0/ GPI04_7
GND	GND	3	4	GPI021/SPI 1_SCLK	NF_ALE/ SPIM0_SCLK /LBC_BCTL/ GPI04_6

注意：

1. GPIO 插座与 CPU 之间没有静电防护器件，所以应避免带电热插拔外设，以免损坏 CPU

3.2.2 FLASH 烧写插座

板上预留了 8 脚的 PH 2.0 连接器用于烧写 FLASH 芯片，就是简单地把 FLASH 芯片的全部 8 个引脚引出，可以连接到支持“在系统烧写（ISP）”功能的编程器。



8

1

连接器 J10 的引脚定义为：

引脚编号	引脚名称	功能描述
1	CS#	片选，低电平有效
2	SO/I01	串行输出或者 I01
3	WP/I02	写保护或者 I02
4	VSS	GND
5	SI/I00	串行输入或者 I00
6	SCK	串行时钟
7	I03/RST#	复位或者 I03
8	VDD	3.3V 电源

3.2.3 以太网插座

以太网插座是标准的 RJ45 连接器，连接双绞线网线，引脚分配这里从略，位号 J4。



3.2.4 USB 3.0 插座

USB 3.0 插座提供标准的 USB 3.0 信号，也提供 USB 2.0 信号，USB 3.0 信号可配置为 PCI-E 信号。这个连接器是上下重叠的双 USB 口，每个口对应 9 个信号线。



本插座位号 J5，目前采用 TXGA 公司的 FUS327-FDBU1K 连接器，引脚定义为：

引脚编号	引脚名称	功能描述
1	VBUS	电源
2	D-	USB 2.0 数据
3	D+	
4	GND	GND
5	StdA_SSRX-	U3P0_RXN 或 PCIE2_RXN

6	StdA_SSRX+	U3P0_RXP 或 PCIE2_RXP
7	GND_DRAIN	GND
8	StdA_SSTX-	U3P0_TXN 或 PCIE2_TXN
9	StdA_SSTX+	U3P0_TXP 或 PCIE2_TXP
10	VBUS	电源
11	D-	USB 2.0 数据
12	D+	
13	GND	GND
14	StdA_SSRX-	U3P1_RXN 或 PCIE1_RXN
15	StdA_SSRX+	U3P1_RXP 或 PCIE1_RXP
16	GND_DRAIN	GND
17	StdA_SSTX-	U3P1_TXN 或 PCIE1_TXN
18	StdA_SSTX+	U3P1_TXP 或 PCIE1_TXP

注：此处的连接器型号可能会变更为兼容型号，并且不会另行通知。

3.2.5 USB 2.0 插座

USB 2.0 连接器采用标准的上下重叠标准 USB 口。



本插座位号 J7，目前采用 TXGA 公司的 FUS208-FDBW3K 连接器，引脚定义为：

引脚编号	引脚名称	功能描述
1	5V	5V 电源输出
2	D-	USB2_P2_DM
3	D+	USB2_P2_DP
4	GND	GND
5	5V	5V 电源输出
6	D-	USB2_P3_DM
7	D+	USB2_P3_DP
8	GND	GND

注：此处的连接器型号可能会变更为兼容型号，并且不会另行通知。

3.2.6 MiniDP 插座

MiniDP 插座为标准的显示输出插座，本板包含 1 个数据 lane 和一个 AUX lane，连接器定义为标准定义，这里从略，位号 J8



3.2.7 电源输入插座

电源插座用于提供 5V 电源，正常情况采用 2 安培的电源既可，可以用普通的手机充电器供电。在外接大功率的设备，如 USB 口的 AI 加速棒时，需要采用 3~4A 输出电流

的电源，位号 J9。

3.2.8 天线插座

WIFI 和蓝牙（BT）模块共用一个天线插座，插座形式为 IPEX 1 代，建议使用 2.4GHz/5.8GHz 双频，WIFI 和蓝牙二合一天线。位号 J2。



3.2.9 音频输入输出插针

音频信号从本板 SATA 和 mini-DP 口之间的插针，包括右声道输出、左声道输出、话筒输入、地三个信号，交流耦合。



位号 J11，三角形标记为第 1 脚，针脚的定义为：

引脚编号	引脚名称	功能描述
1	OUT_L	左声道输出
2	OUT_R	右声道输出
3	MIC_IN	音频输入
4	GND	低

3.2.10 CANBus

本板有两路 CANBus 接口，支持 CAN 2.0 协议和 CAN FD 协议，板上没有收发器，需要用户自己扩展。引出到板边的针座。



位号 J6，三角形标记为第 1 脚，这个插座没有地信号，需要用户从其它插座引地线，这里的信号不是差分信号，地是必须的。针脚的定义为：

引脚编号	引脚名称	功能描述
1	CAN1_TX	CANBUS 控制器 1，发送
2	CAN1_RX	CANBUS 控制器 1，接收
3	CAN0_TX	CANBUS 控制器 0，接收
4	CAN0_RX	CANBUS 控制器 0，发送

3.2.11 调试串口

本板有一个调试串口，一般用于在显示器和网口未准备好时调试板卡。电平 3.3V LVCMOS 电平，习惯称为 TTL 串口。



位号 J70，靠近 J70 字样的是 1 脚，针脚定义为：

引脚编号	引脚名称	功能描述
1	P3V3	3.3V 电源输出
2	DEBUG_UART1_RXD	串口发送
3	DEBUG_UART1_TXD	串口接收
4	GND	地

4. 使用方法

4.1 准备工作

把开发板从防静电袋中取出，简单检查没有松动或者掉落的器件，把开发板放置在桌面上，将 USB 转 TTL 串口线的 USB 口连接计算机，另一侧的串口插座。

在 windows 电脑上执行“开始->运行-> devmgmt.msc”，打开设备管理器，寻找 USB 转串口设备，记下串口号。



运行终端程序 putty，在开始连接窗口中，先选择串口模式“Serial”，在 Serial line 一栏填入上一步发现的串口号（形式为 COMx，x 代表数字）、在 Speed 栏填入波特率“115200”，点击 Open，打开串口。

把含有操作系统的优盘插到两个 USB 3.0 插座中的一个口。

如果内部的操作系统带有视频输出，可以把 miniDP 口通过 miniDP 转 HDMI 电缆与带有 DP 口的显示器相连，如果使用 DP 转 HDMI 的转接头，有些型号可能无法正常工作。

4.2 上电启动

把电源适配器插入交流插座，把圆形插头插入电源插座，开发板上红色电源指示灯会立即点亮，串口会输出信息，顺次启动 uboot 和操作系统，提示信息详见文后附录。

4.3 固件和操作系统烧写

开发板的启动程序存放在 FLASH 芯片中，本板使用的是 uboot，uboot 启动后可以选择启动优盘中的操作系统等程序，如果 uboot 因为升级或误操作被损坏无法启动，需要用编程器通过 FLASH 烧写插座烧写，支持通过电缆在系统编程（ISP）的编程器都可以，这里以硕飞公司的 SP16-B 或者 SP20-B 编程器举例说明烧写过程

1. 先制作一根转接电缆，一端是 PH 2.0 8pin 插头，连接开发板，一端是 FC 10p 插头（俗称灰排线），连接编程器，两者的连接关系如下表。

编程器插座	开发板插座	网络
1		空
2	1	VCC
3	7	CS_N
4		空
5	2	SI (I00)
6		空
7	3	S0 (I01)
8		空
9	6	SCK
10	8	GND

实物如下图，PH2.0 插头的红线是 1 脚，灰排线插头的红色是 1 脚。实际使用的信号线有 6 条。



2. 打开编程软件 FlyPRO，通过菜单“芯片->选择芯片”，选择 Macronix 公司的 MX25L12835F [ISP]，一定要选择带 ISP 字样的型号，这样才能通过电缆烧写。打开“操作->操作选项”勾选上“向目标板提供电源”，电压选择“3.3V”。因为开发板上电状态下，CPU 会访问 FLASH 芯片，这时编程会由于总线冲突失败。所以必须断开本板的电源，改由编程器供电。
3. 确认开发板的 USB 供电插头已经断开，开发板没有上电。在编程软件中加载 uboot 二进制文件，选择“操作->单次烧录”开始烧写，这部分操作和用插座烧录芯片相同，但是速度更慢，整个过程大约需要十分钟左右。
4. 烧写成功后拔下烧写电缆，插上 USB 供电插头，给开发板上电。

5. 注意事项

开发板使用过程中，需注意以下几方面问题：

1. 防静电：为了防止人体静电打坏板卡，在拿取板卡前要先抚摸较大的金属物体（如电脑机箱、桌腿等）泄放静电。有条件的最好佩戴防静电手环。
2. 防短路：放置开发板时，注意其下不能垫导电物体，如镀铝的袋子、螺丝帽、金属壳的笔等。板上的线缆和连接器应注意固定，不要使金属部分触碰到电路板。
3. 防机械损伤：本板使用时要防止从桌面坠落，连线不能拖到地面，以防被牵扯坠地。

6. 故障排除

使用过程中碰到简单的故障可以尝试自行排除，故障分为硬件故障和软件故障两种，具体如下表：

问题现象	解决方法
开发板不上电，电源指示灯不亮	触摸检查板卡是否发热，如果发热不发热则可能没有供电。检查电源适配器（手机充电器）是否正常工作，电源线是否正常，可以尝试给手机充电或者更换其它已知正常的电源 如果板卡发热，则需要返厂维修
板卡正常上电串口无输出	分辨是完全无输出还是有一些乱码，有乱码可能是波特率设置错误，需要恢复 115200 的波特率。如果完全无输出，检查 TTL 转串口线是否正常，换用别的正常的线。核对 40p 连接器上连接的引脚位置是否正确，发送和接收是否接反。 尝试重新烧写 uboot
板卡正常上电，显示器无显示	检查显示电缆是否正常，换用其它好的电缆，检查显示器的输入源选择是否正确，检查是否用了 DP 转 HDMI 或 VGA 的转接线，这些转接线可能不被支持，或者因为过热或供电不足暂停工作，需要冷却或者连接外接电源。通过串口检查操作系。 检查串口是否有输出，如果有输出，是否运行到了初始化显示阶段以后
板卡正常上电和显示和串口输出正常，USB 口不正常	USB 口如果过流，可能会短暂停止向 USB 口供电，尝试拔掉所有 USB 口的连线，开发板下电冷却十五分钟，再尝试。或者插入已知正常的优盘，尝试读取。 如果有条件，将相同的软件拷贝到另一块开发板上测试，看是否正常。
板卡正常上电和显示和串口输出正常，40p 接口不正常	如果有条件，将相同的软件拷贝到另一块开发板上测试，看是否正常。 查看 40p 连接器的正面或背面是否粘上了导电碎屑导致相邻针短路

7. 附录

7.1 启动提示信息

系统从上电到启动到 uboot，串口输出的信息如下：

```
P: power on...
P: init stack done
P: check reset
N: wait for scp ready
P: chip_id : KPA2911522290000
N: 5: c0000000; 6: c0000000; 7: c0000003; 4: 80000002
P: Booting Trusted Firmware
P: BL1: v2.3(release):E2000-v1.03
P: BL1: Built : 11:14:15, Dec 1 2022
P: BL1: RAM 0x30c01000 - 0x30c14000
P: Generic delay timer configured with mult=1 and div=50
P: Entry point address = 0x38180000
P: SPSR = 0x3c9
P: Argument #0 = 0x0
P: Argument #1 = 0x0
P: Argument #2 = 0x0
P: Argument #3 = 0x0
P: Argument #4 = 0x0
P: Argument #5 = 0x0
P: Argument #6 = 0x0
P: Argument #7 = 0x0
flash tpye = default flash
get flash speed = 1
flash set done
N: pre svc call fun = 0xc2000f02 -- pm-1 = 0, pm-2 = 30c19920,
pm-3 = 0
N: CPU info :
N: magic = 0x54460020
N: version = 0x2
N: size = 0x100
N: clust0_pll = 0x708
N: clust1_pll = 0x708
N: clust2_pll = 0x3e8
N: dmu_pll = 0x258
N: pll config
Parametr :pll 0 frq = 1800
Parametr :pll 1 frq = 1800
Parametr :pll 2 frq = 1000
Parametr :pll 3 frq = 1200
Parametr :pll 4 frq = 600
N: reg_pll_c0_real = 1800 MHZ
N: reg_pll_c1_real = 1800 MHZ
N: reg_pll_c2_real = 1000 MHZ
N: reg_pll_dmu_0_real = 600 MHZ
N: scmi pll config
get the ch->info->scmi_mbx_mem is 0x32a10400
N: get the return= 0x0
gpiol_13 ret = 8256
gpiol_13 ret = 8256
gpiol_13 ret = 8256
gpiol_13 ret = 8256
gpiol_13 ret = 8256
gpiol_13 ret = 8256
```

```
gpiol_13 ret = 8256
gpiol_13 ret = 8256
gpiol_13 ret = 8256
gpiol_13 ret = 8256
cpu_type ret = 0x0
cpu_id:D/S
Is demo board!
sata device
board_type ret = 0x2040
board_id:D1
lsd pad parameter no auto!
parameter pad_base = 0x380fa000
ERROR:board pad info no found!!!
pcie parameter no auto!
N: pre svc call fun = 0xc2000f03 -- pm-1 = 0, pm-2 = 30c19920,
pm-3 = 0
N: pcie fun = 0x0 -- pm addr = 30c19920
N: PCIE info :
N: magic = 0x54460021
N: version = 0x4
N: size = 0x100
N: independent_tree = 0x0
N: base_cfg = 0x50007
N: size = 0x100
N: PEU 0 - C0 base_config = 0x10203
N: PEU 0 - C1 base_config = 0x10203
N: PEU 0 - C0 eq = 0x7
N: PEU 0 - C1 eq = 0x7
N: PEU 1 - C0 base_config = 0x10003
N: PEU 1 - C1 base_config = 0x10003
N: PEU 1 - C2 base_config = 0x10003
N: PEU 1 - C3 base_config = 0x10003
N: PEU 1 - C0 eq = 0x7
N: PEU 1 - C1 eq = 0x7
N: PEU 1 - C2 eq = 0x7
N: PEU 1 - C3 eq = 0x7
I: PEU 0 pre_init
I: Parametr : peu 0 mode 0x3
I: set peu0 bif is 0x0
I: Parametr : peu 0 speed 0xf
I: Parametr : peu 0 ctr 0 mode 0x1
I: Parametr : peu 0 ctr 1 mode 0x1
I: set psu phy reg
I: Parametr : peu 0 init stat 0x1
N: PEU 0 phy init
I: Parametr : peu 0 split mode 0x3
I: read peu0 clk_status.
I: peu clk_status1 0x5, target clk_status 0xf
I: peu clk_status2 0x5
I: Parametr : peu 1 init stat 0x1
I: PEU 1 pre_init
I: Parametr : peu 1 mode 0xf
I: Parametr : peu 1 split mode 0x2
I: set peul bif is 0x2
I: Parametr : peu 1 speed 0xff
I: Parametr : peu 1 ctr 0 mode 0x1
I: Parametr : peu 1 ctr 1 mode 0x1
I: Parametr : peu 1 ctr 2 mode 0x1
I: Parametr : peu 1 ctr 3 mode 0x1
N: PEU 1 phy init
I: split mode is not X4!.
I: read peul clk_status.
I: peu clk_status1 0xff, target clk_status 0xff
```



```

I: peu clk_status2 0xff
I: Parametr : independent_tree --- 0x0
Parametr : peu 0 port stat 0x3
I: Parametr : peu 0 split mode 0x3
I: set peu msien 0x0
I: ocn msien 0x10000
I: Parametr : peu 0 ctr 0 mode 0x1
I: Parametr : peu 0 ctr 1 mode 0x1
I: Parametr : peu 1 init stat 0x1
I: Parametr : independent_tree --- 0x0
Parametr : peu 1 port stat 0x0
I: set peu msien 0x0
I: ocn msien 0x10000
I: Parametr : peu 1 ctr 0 mode 0x1
I: Parametr : peu 1 ctr 1 mode 0x1
I: Parametr : peu 1 ctr 2 mode 0x1
I: Parametr : peu 1 ctr 3 mode 0x1
vhub device set!
phy parameter no auto!
N: pre svc call fun = 0xc2000015 -- pm-1 = 3f, pm-2 = 484, pm-
3 = 3030300
usb 0x31b00000 init.
usb 0x31c00000 init.
sgmii0 init.
set mode: 0x3, speed: 0x3 phy_base: 0x32100000, ctrl_base:
0x3200c000.
1
0x32100000 .
7
gsd sata phy init.
SATA 1 phy init seq start.
sgmii2 init.
set mode: 0x3, speed: 0x3 phy_base: 0x32300000, ctrl_base:
0x32010000.
1
0x32300000 .
7
gsd_phy3 dpl nothing.

U-Boot 2022.01-v1.40 (Jan 10 2023 - 00:33:59 -0800)

DRAM: Phytium ddr init
ddr parameter no auto!
CH 0 :
force spd, dimm info is from parm
tAAmin = 13750ps
tRCDmin = 13750ps
tRPmin = 13750ps
tRASmin = 32000ps
tRCmin = 45750ps
tFAWmin = 30000ps
tRRD_Smin = 5300ps
tRRD_Lmin = 6400ps
tCCD_Lmin = 5000ps
tWRmin = 15000ps
tWTR_Smin = 2500ps
tWTR_Lmin = 7500ps
tRFC1min = 160000ps
tRFC2min = 110000ps
tRFC4min = 90000ps
Dimm_Capacity = 4GB
DDR4 UDIMM/2 Bank Groups/4 Banks/Column 10/Row

```

```

17/X16/1 Rank/NO ECC/Standard
Modual:Unknown=0x0
N: pre svc call fun = 0xc2000f04 -- pm-1 = 0, pm-2 = 30c19920,
pm-3 = 0
N: ddr fun = 0x0 -- pm = 0x30c19920, pm2 = 0x0
N: parameter mcu: v0.2

Mcu Start Work ...
N: scmi get clock value.
get clock value 0x23c34600 600000000
CTL Freq = 600Mhz
Lmu Freq = 1200Mhz
get s3 = 0
ch = 0
the dimm info is from uboot...
FORC03 = 0x0
FORC04 = 0x0
FORC05 = 0x0
FORC03 = 0x0
FORC04 = 0x0
FORC05 = 0x0
FORC03 = 0x0
FORC04 = 0x0
FORC05 = 0x0
parameter set ecc disable!
Dimm_Capacity = 4GB
MEM encrypt configuration begin...
MEM encrypt bypass end...
TZC configuration begin...
TZC bypass end...
use_0x10 == 0x30004
use_0x14 == 0x30100
ctl_cfg_begin.....
pi_cfg_begin.....
phy_cfg_begin.....
fast mode
caslat = 17
wrlat = 16
tinit = 960000
r2r_diffcs_dly = 4
r2w_diffcs_dly = 5
w2r_diffcs_dly = 3
w2w_diffcs_dly = 7
r2w_samecs_dly = 4
w2r_samecs_dly = 0
r2r_samecs_dly = 0
w2w_samecs_dly = 0
adapter_alg -- 0-0-0-0-0-0-0
rtt_wr = dis
rtt_park = 80ohm
ron = 34ohm
val_cpudrv = 34
rtt_nom = 48ohm
val_cpuodt = 48
vref_dev = 10
vrefsel = 0x45
dq_oe_timing = 0x42
rank_num_decode = 1
mem_dp_reduction = 0x1
set_phy_indep_init_mode
set_pi_dram_init_en
set_pi_start & ctl_start.....
wait init complete.....

```

```

init complete done.....
wait complete done.....
rddqs_lat = 0x2
tdfi_phy_rdlat = 0x1f
begin software ntp training...
rank_num: 0
phy_write_path_lat_add =-1-1-1-1-1-1-1-1

slice_num 0
pi_bist data bist fail!

slice_num 1
pi_bist data bist fail!

slice_num 2
pi_bist data bist fail!

slice_num 3
pi_bist data bist fail!
phy_write_path_lat_add = 0 0 0 0-1-1-1-1-1

slice_num 0
pi_bist data bist pass!
start_addr = 0x80000000
end_addr = 0x80000100

slice_num 1
pi_bist data bist pass!
start_addr = 0x80000000
end_addr = 0x80000100

slice_num 2
pi_bist data bist pass!
start_addr = 0x80000000
end_addr = 0x80000100

slice_num 3
pi_bist data bist pass!
start_addr = 0x80000000
end_addr = 0x80000100
rank 0 wdqlv1!
r2r_diffcs_dly = 4
r2w_diffcs_dly = 6
w2r_diffcs_dly = 4
w2w_diffcs_dly = 6
r2w_samecs_dly = 4

rank 0
training success

ctl-bist: addr_space = 0x14
bist test start...
data bist pass!
bist test end...

ctl-bist: addr_space = 0x14
bist test start...
addr bist pass!
bist test end...
BIST OK, access memory!

skip_scrub_flag = 0, scrub init
start_addr = 0x0, addr_space = 32
zero memory start...
zero memory end
N: pre svc call fun = 0xc2000f05 -- pm-1 = 0, pm-2 = 30c19920,
pm-3 = 0
N: CPU info :
N: magic = 0x54460013
N: version = 0x1
N: size = 0x100
N: core_bit_map = 0x100
I: BL1: Loading BL2
V: Using Memmap
V: FIP header looks OK.
V: Using FIP
I: Loading image id=1 at address 0xffe00000
I: Image id=1 loaded: 0xffe00000 - 0xffe0c61d
V: BL1: BL2 memory layout address = 0x30c00000
N: BL1: Booting BL2
P: Entry point address = 0xffe00000
P: SPSR = 0x3c5
P: Argument #0 = 0x0
P: Argument #1 = 0x30c00000
P: Argument #2 = 0x0
P: Argument #3 = 0x0
P: Argument #4 = 0x0
P: Argument #5 = 0x0
P: Argument #6 = 0x0
P: Argument #7 = 0x0
N: BL2: v2.3(release):E2000-v1.03
N: BL2: Built : 11:14:15, Dec 1 2022
I: BL2: Doing platform setup
N: img_id: 4, image_base: 0x38100000, img_size: 0x10000
N: Copy BL32 image from: 0x38100000 to: 0xfc000000 size:
0x10000
I: BL2: Loading image id 3
V: Using Memmap
V: FIP header looks OK.
V: Using FIP
I: Loading image id=3 at address 0xffa00000
I: Image id=3 loaded: 0xffa00000 - 0xffa1234a
I: BL2: Skip loading image id 23
I: BL2: Skip loading image id 25
I: BL2: Skip loading image id 4
I: BL2: Skip loading image id 21
I: BL2: Skip loading image id 5
I: BL2: Skip loading image id 27
N: BL1: Booting BL31
P: Entry point address = 0xffa00000
P: SPSR = 0x3cd
P: Argument #0 = 0xffe0d760
P: Argument #1 = 0x0
P: Argument #2 = 0x0
P: Argument #3 = 0x0
P: Argument #4 = 0x0
P: Argument #5 = 0x0
P: Argument #6 = 0x0
P: Argument #7 = 0x0
P: entry bl31 form bl2.
N: bl_params - x0 = ffe0d760
N: TLB: start setup TLB
mmmap:

```

```

VA:0x38040000    PA:0x38040000    size:0x100000    attr:0x1    suspend_finish_entry addr =0x3818a690
granularity:0x40000000
VA:0x0           PA:0x0           size:0x80000000    attr:0x8    N: Phytium System Service Call: 0xc2000012
granularity:0x40000000    N: PFDI: get vectors : addr 30c199f8
VA:0xfc000000    PA:0xfc000000    size:0x2000000    attr:0xa    N: PFDI: system_off_entry : 3818a660
granularity:0x40000000    N: PFDI: system_reset_entry : 3818a678
VA:0xffa00000    PA:0xffa00000    size:0x12000    attr:0x2    N: PFDI: suspend_start_entry : 3818a6a8
granularity:0x40000000    N: PFDI: suspend_end_entry : 3818a6c4
VA:0xffalf000    PA:0xffalf000    size:0x1000    attr:0x8    N: PFDI: suspend_finish_entry : 3818a690
granularity:0x40000000    1.9 GiB
VA:0xffa00000    PA:0xffa00000    size:0x600000    attr:0xa    MMC: clk = 1200000000Hz
granularity:0x40000000    clk = 1200000000Hz
VA:0x80000000    PA:0x80000000    size:0x80000000    attr:0x1a    PHYTIUM MCI: 0, PHYTIUM MCI: 1
granularity:0x40000000    Loading Environment from SPIFlash... *** Warning - bad CRC,
using default environment

V: Translation tables state:
V: Xlat regime: EL3
V: Max allowed PA: 0xffffffff
V: Max allowed VA: 0xffffffff
V: Max mapped PA: 0xffffffff
V: Max mapped VA: 0xffffffff
V: Initial lookup level: 1
V: Entries @initial lookup level: 4
V: Used 4 sub-tables out of 4 (spare: 0)
N: start enable mmu
N: mmu setup done
N: TLB: start setup TLB
N: BL31: v2.3(release):E2000-v1.03
N: BL31: Built : 11:14:18, Dec 1 2022
I: GICv3 without legacy support detected.
I: ARM GICv3 driver initialized in EL3
P: Generic delay timer configured with mult=1 and div=50
N: PFDI: start setup
N: GMAC addr setup-mac force N: mac = 553721160c
I: BL31: Initializing runtime services
I: BL31: Initializing BL32
W: BL31: BL32 initialization failed
I: BL31: Preparing for EL3 exit to normal world
I: Entry point address = 0x381c0000
I: SPSR = 0x3c9
V: Argument #0 = 0x200
V: Argument #1 = 0x0
V: Argument #2 = 0x0
V: Argument #3 = 0x0
V: Argument #4 = 0x0
V: Argument #5 = 0x0
V: Argument #6 = 0x0
V: Argument #7 = 0x0
N: cm temp context restore
N: runtime setup
N: 5: c0000000; 6: c0000000; 7: c0000003; 4: 80000002
N: wait for scp ready
N: 5: c0000000; 6: c0000000; 7: c0000003; 4: 80000002
N: enable scp->ap os chanel interrupt.
N: [os chanel] = 1
N: enable ap->scp os chanle interrupt.
N: [os chanel] = 1
N: scmi_core_reset_addr_set.
PBF relocate done
system_off_entry addr =0x3818a660
system_reset_entry addr =0x3818a678
suspend_entry addr =0x3818a6a8
suspend_end_entry addr =0x3818a6c4

In: uart@2800d000
Out: uart@2800d000
Err: uart@2800d000
Net: eth0: ethernet@3200c000, eth1: ethernet@3200e000, eth2:
ethernet@32010000, eth3: ethernet@32012000
pci_0:2.0 00:02.00 - 1db7:dc01 pcie@40000000
0x06 (Bridge device)
pci_0:3.0 00:03.00 - 1db7:dc01 pcie@40000000
0x06 (Bridge device)
pci_0:4.0 00:04.00 - 1db7:dc01 pcie@40000000
0x06 (Bridge device)
pci_0:5.0 00:05.00 - 1db7:dc01 pcie@40000000
0x06 (Bridge device)
scanning bus for devices...
SATA link 0 timeout.
AHCI 0001.0301 32 slots 1 ports 6 Gbps 0x1 impl SATA mode
flags: 64bit ncq stag pm led clo only pmp pio slum part ccc
apst
SATA link 0 timeout.
AHCI 0001.0301 32 slots 1 ports 6 Gbps 0x1 impl SATA mode
flags: 64bit ncq stag pm led clo only pmp pio slum part ccc
apst
Hit any key to stop autoboot: 0
Couldn't find partition scsi 0:2
Can't set block device
Couldn't find partition scsi 0:2
Can't set block device
Couldn't find partition scsi 0:2
Can't set block device
Wrong Image Format for bootm command
ERROR: can't get kernel image!
E2000#

```