

第1章 产品简介

1.1 产品概述

EBF6ULL S1 Mini 开发板是野火电子基于 NXP i.MX 6ULL 系列处理器设计的一款低功耗单板电脑，工业级主频最高可达 792MHz，具体见图 1-1。



图 1-1 EBF6ULL S1 Mini 开发板

EBF6ULL S1 Mini 开发板提供完整的 SDK 驱动开发包、核心板封装库，底板应用参考设计原理图，可帮助客户大大缩减产品的开发时间，加快产品上市。目前提供 Linux 4.1.15 版本的内核，配套的系统自带 python、Qt 等组件。

开发板整板由 EBF6ULL S1 核心板模组和底板组成，具体见图 1-2，元件采用工业级选料，其中核心板板载主控芯片 512MB DDR3L 内存，256/512MB Nand-FLASH（或 8GB eMMC）。



图 1-2 核心板外观

核心板 PCB 采用 8 层黑色沉金设计，单面放元件，整体尺寸 39x39mm，共 140 个引脚，引脚间距 1.0mm，除了 SEMC 总线引脚外，芯片其余 IO 均引出。该核心板适用于工业控制、手持扫码、喷墨打印机、轨道交通、无人机控制和音频输出等领域。

1.1 EBF6ULL 核心板硬件资源

野火 EBF6ULL 核心板包含多种配置版本，它按照核心板上 FLASH 存储器类型进行区分，具体见图 1-3。

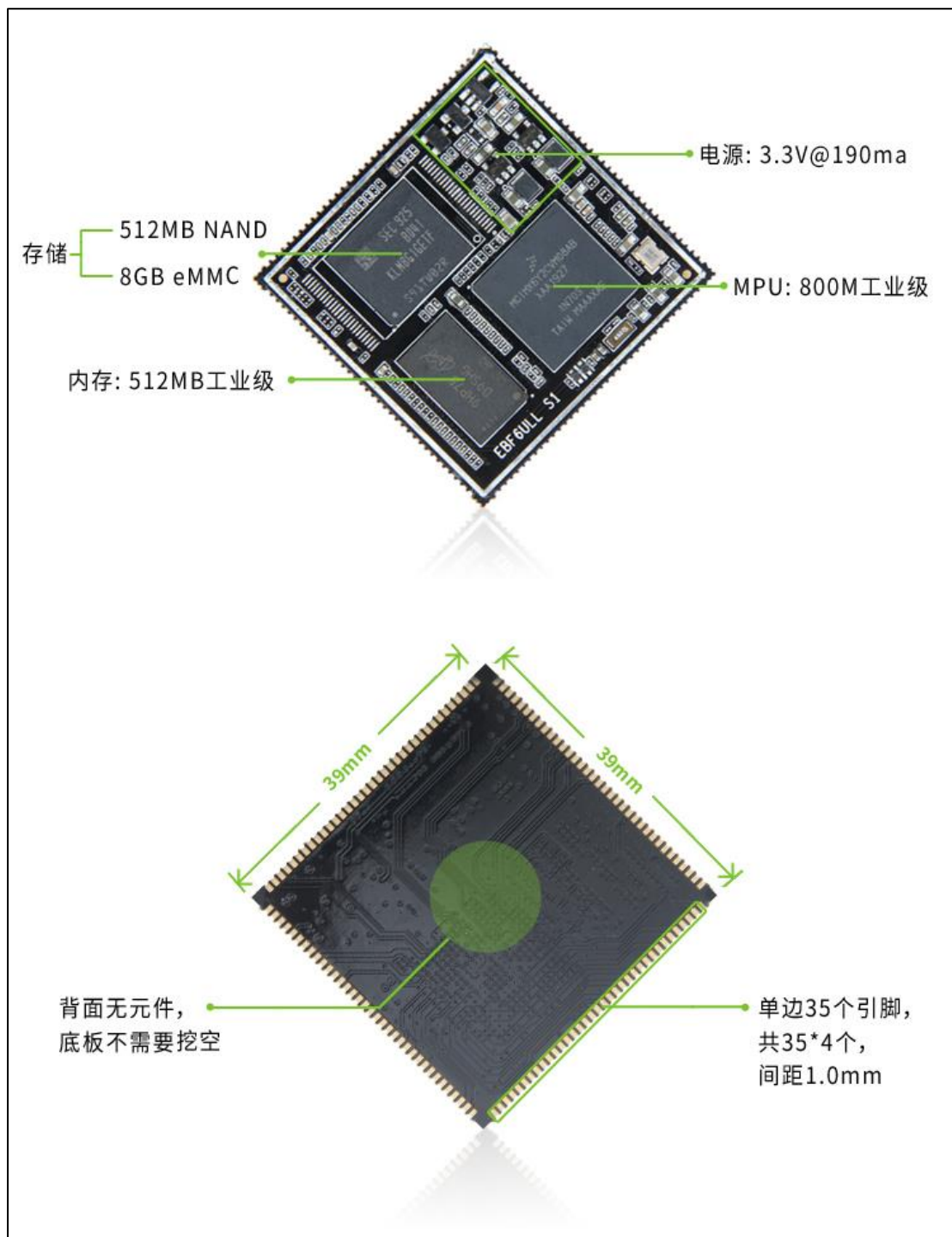


图 1-3 EBF6ULL S1 邮票孔核心板内部

EBF6ULL S1 核心板模组参数如下:

- CPU: NXP MCIMX6Y2CVM08AB 处理器
- 频率: 工业级最高可达 792MHz
- DDR3L 内存: 板载 512MB DDR3L
- FLASH 存储器示不同型号的板子, 使用以下其中一种配置:
 - 板载 256MB 或 512MB 的 Nand-FALSH

- 板载 8GB 的 eMMC
 - PCB: 8 层黑色沉金, 尺寸为 39 x 39mm
 - 封装: 邮票孔封装, 单面元件, 背面没有元件, 底板不需要挖槽
- 使用不同存储器版本的开发板时, 主要区别在于设置不同的启动方式, 以及使用系统时要注意其容量大小。

1.2 EBF6ULL Mini 底板硬件资源

EBF6ULL Mini 底板硬件资源见图 1-4。

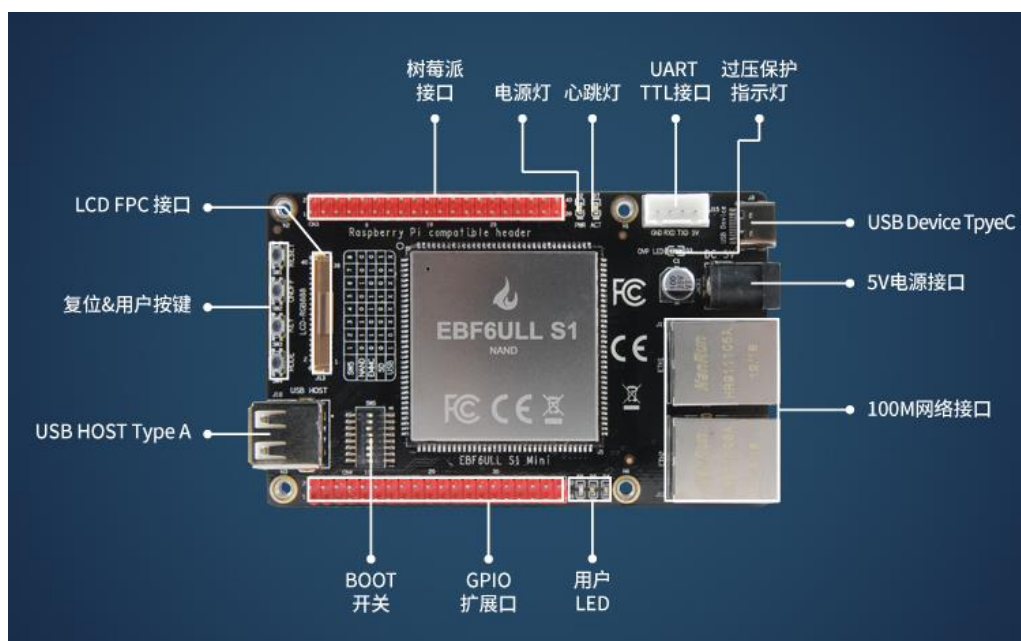


图 1-4 EBF6ULL S1 Mini 开发板硬件资源图（正面）

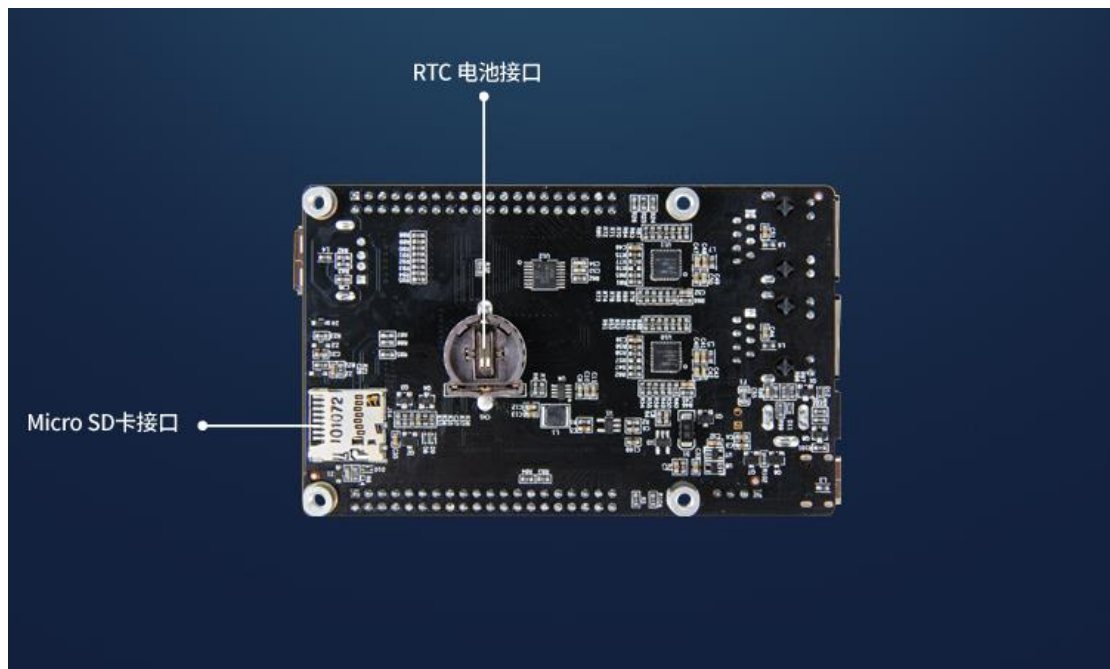


图 1-5 EBF6ULL S1 Mini 开发板硬件资源图（背面）

EBF6ULL S1 Mini 底板参数如下：

- PCB：6 层黑色沉金，尺寸为 100x61.8mm
- 100M 以太网接口：2 路百兆以太网接口
- 5V 电源接口：使用 $5V \pm 2\%$ 单电源供电
- 过压保护指示灯：该灯亮时表示电压超出范围
- USB Device 接口：使用 Type C 接口引出
- UART TTL 接口：主控器串口引脚直接引出，TTL 电平
- 树莓派接口：包含 UART、I2C、SPI、PWM 等兼容树莓派的 IO 接口
- 心跳灯与电源灯：系统运行后心跳灯会持续闪烁
- LCD FPC 接口：含 24 位 RGB 接口及 I2C 触摸屏控制接口
- 复位&按键：共 4 个按键，分别为复位、ON/OFF、普通按键及 Mode 模式切换按键
- USB Host Type A：1 路使用 Type A 引出 USB Host 接口
- Boot 开关：一个 8 位拨码开关，支持切换 NAND、eMMC、SD 及 USB 启动方式
- GPIO 扩展口：包含主控器的其它 GPIO
- RTC 电池接口：可接入型号为 CR1220 的电池为 RTC 模块供电
- SD 卡槽：micro SD 卡插槽，支持 SD 卡 3.0
- IO 扩展：包含一个 74LV595PW 芯片，把某 4 路 IO 扩展成 8 路

第2章 让开发板跑起来

本章主要讲解如何使用配套的开发板，让开发板的系统运行起来！

2.1 了解开发板的启动方式

i.MX 系列芯片支持多种启动方式，在我们配套的开发板上主要使用其中的 Nand-FLASH、eMMC、SD 卡及 USB 启动方式。

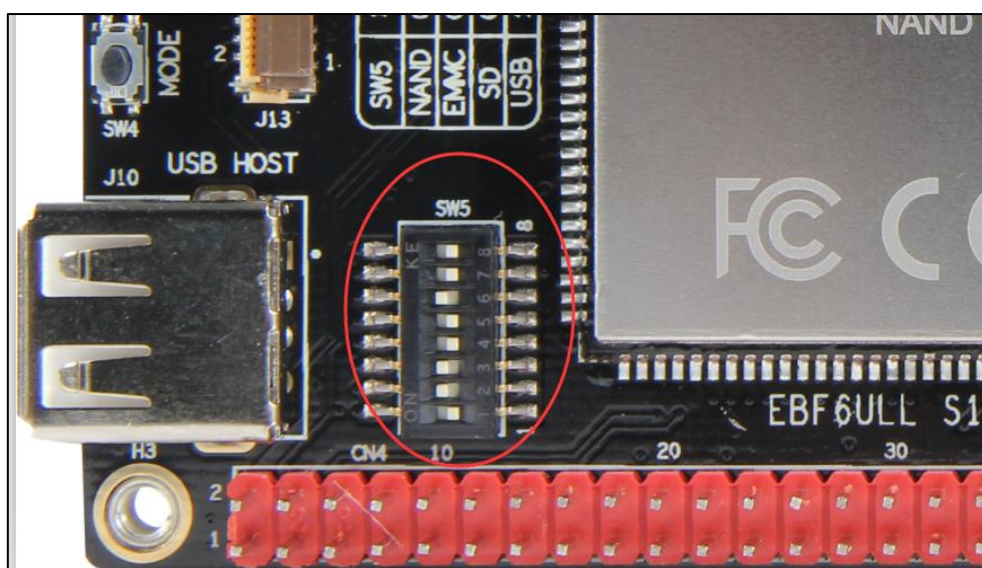


图 2-1 boot 拨码开关（请根据具体启动方式进行配置）

拨码开关的一侧会写着 ON 字样，把拨码调至该侧表示对应的选项为高电平 1，调至另一侧为 0，不同启动方式的见表 2-1，也可以直接查看开发板上的丝印说明，表中的 X 表示任意电平均可。

表 2-1 拨码开关配置的启动方式

编号	名称	Nand FLASH	eMMC	SD	USB
1	MODE0	0	0	0	1
2	MODE1	1	1	1	0
3	CFG1-4	1	0	0	X
4	CFG1-5	0	1	0	X
5	CFG1-6	0	1	1	X
6	CFG1-7	1	0	0	X
7	CFG2-3	0	1	0	X
8	CFG2-5	0	0	1	X

开发板上电后会根据拨码开关的状态从不同的存储器加载代码运行，故上电前需要根据自己开发板使用的存储器进行配置，如 Nand FLASH 核心板就配置为 0-1-1-0-0-1-0-0，其它依次类推。

其中的 USB 启动模式主要用来配合 NXP 官方的 mfgtool 工具烧录镜像。

2.2 硬件准备

要进行本章的实验操作，需要准备如下硬件：

- ☐ 开发板
- ☐ DC **5V** 电源
- ☐ Mini 接口的 USB 线
- ☐ 一台 Windows 系统的电脑
- ☐ 配套的屏幕（可选）

2.3 启动步骤

开发板出厂时默认都烧录了 Linux 系统，确认设置了正确的 boot 启动方式，只要上电就会自动运行。

以下为开发板的启动操作步骤：

- (1) 根据自己开发板的版本和表 2-1 设置拨码开关（出厂配置通常是配套的，检查确认即可）。
- (2) 使用 DC **5V** 电源给开发板供电，注意由于开发板底板设备较多，功耗大。只使用 USB 线给开发板供电是无法正常运行的！
- (3) 按下电源开关，给开发板上电。
- (4) 对于带屏幕的开发板，可以直接通过触摸屏控制开发板。
- (5) 开发板支持鼠标和键盘，有需要可以通过 USB 接口连接至开发板。
- (6) 对于不带屏幕的开发板，可以通过串口终端控制开发板。关于串口终端的使用请参考下一小节的内容。

2.4 设置 Putty 连接终端

开发板支持使用串口终端进行控制，并且为方便使用，开发板上的串口通过 CH340 转换成 USB 通讯，使得开发板与电脑连接时直接用 USB 线即可。使用时需要在电脑先安装 CH340 芯片的 USB 转串口驱动。

连接串口终端时可以使用普通的串口调试工具收发字符串，不过这样控制起来非常不方便，此处我们推荐使用 Putty 终端软件。如果使用的不是 Putty 软件，以下连接开发板串口终端的操作步骤也大体相似，可以参考它来操作。

以下操作步骤在 Windows 系统的开发主机进行：

- (1) 安装 CH340 芯片的 USB 转串口驱动，该驱动安装程序可在开发板配套的资料里找到。

- (2) 使用 Mini USB 线连接电脑与开发板，注意在开发板端要接入到板子上的“USB 转串口”接口，并通过 DC 电源供电，不能只使用 USB 线供电。
- (3) 连接并供电后，在“我的电脑”设备管理器的“端口”设备下会新增一个“USB-SERIAL CH340”设备，请查看自己电脑上该 COM 口的编号，这在不同的电脑上编号是不同的，见图 2-2。



图 2-2 查看 COM 口

- (4) 安装并打开 Putty 软件，新建会话，在连接的协议处选择使用“Serial”，表示通过串口连接终端，把 Putty 的串口的通讯速率配置成开发板默认值，即“115200”，本例子使用的端口号为“COM4”，注意该端口号要根据自己的实验环境进行选择，即在步骤（3）中查看的端口号。具体见图 2-3。

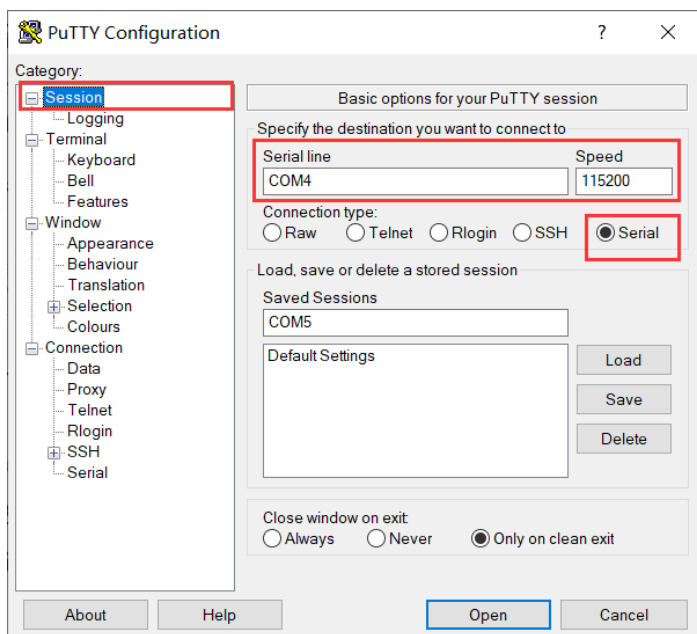


图 2-3 使用 SERIAL 协议连接终端

2.5 登录到终端

设置好连接软件后，打开终端软件的连接即可连接至开发板的串口终端。

- (1) 如果是在开发板开机前就建立了串口终端连接，那么在开机时会看到开发板在启动时的信息输出，见图 2-4。

```
vxM){'^NET: Registered protocol family 26
NET: Registered protocol family 10
sit: IPv6 over IPv4 tunneling driver
NET: Registered protocol family 17
can: controller area network core (rev 20120528 abi 9)
NET: Registered protocol family 29
can: raw protocol (rev 20120528)
can: broadcast manager protocol (rev 20120528 t)
can: netlink gateway (rev 20130117) max_hops=1
Bluetooth: RFCOMM TTY layer initialized
Bluetooth: RFCOMM socket layer initialized
Bluetooth: RFCOMM ver 1.11
Bluetooth: BNEP (Ethernet Emulation) ver 1.3
Bluetooth: BNEP filters: protocol multicast
Bluetooth: BNEP socket layer initialized
Bluetooth: HIDP (Human Interface Emulation) ver 1.2
Bluetooth: HIDP socket layer initialized
8021q: 802.1Q VLAN Support v1.8
Key type dns_resolver registered
```

图 2-4 开发板启动时的终端输出（部分）

- (2) 如果是在开发板开机后才建立连接，开发板没有输出，这时直接按几下回车即可，见图 2-5。

```
Connecting to COM4...
Connected.

Freescall i.MX Release Distro 4.1.15-2.0.0 imx6ul7d /dev/ttyMxc0
imx6ul7d login: █
```

图 2-5 按回车后终端的输出

- (3) 无论是以上哪种情况，开发板的启动流程执行完毕时，只要按回车后终端都会提示 login，此时终端在等待用户的输入，它需要知道我们希望以哪个用户名登录终端。我们的开发板默认用户为：root，不带密码。所以在该提示界面中输入 root 并回车即可，见图 2-6。

```
Freescall i.MX Release Distro 4.1.15-2.0.0 imx6ul7d /dev/ttyMxc0
imx6ul7d login: root
root@imx6ul7d:~# █
```

图 2-6 在 login 提示中输入 root 并回车

- (4) 至此，我们就成功通过串口登录到开发板的终端了，接下来我们就可以使用各种命令来控制开发板。

第3章 查看系统信息

登录到终端后，可在终端下输入如下命令查看系统信息。

3.1 查看 CPU 信息

/proc/cpuinfo 文件存储了 CPU 的信息，可通过如下命令查看：

```
cat /proc/cpuinfo
```

```
root@imx6ull14x14evk:~# cat /proc/cpuinfo
processor       : 0
model name     : ARMv7 Processor rev 5 (v7l)
BogoMIPS      : 3.00
Features      : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xc07
CPU revision   : 5

Hardware       : Freescale i.MX6 Ultralite (Device Tree)
Revision      : 0000
Serial        : 0000000000000000
```

图 3-1 CPU 信息

从图 3-1 可以看到，我们使用的硬件平台是飞思卡尔 i.MX6 Ultralite，是一款 ARMv7 架构的处理器。

3.2 查看内核版本

/proc/version 文件保存了内核的版本信息，我们可以通过如下命令来获取。

```
cat /proc/version
```

```
root@imx6ull14x14evk:/proc/1/fd# cat /proc/version
Linux version 4.1.15-2.1.0+g30278ab (rich@rich) (gcc version 5.3.0 (GCC) ) #19 SMP PREEMPT Wed Aug 7 20:07:49 CST 2019
```

图 3-2 内核版本信息

从图 3-2 中，我们可以看到当前使用的内核版本是 4.1.15。

3.3 查看内存信息

通过 free 命令可查看系统的内存大小。

```
free
```

如图 3-3 所示，输出信息共有三行，六列。其中第一行记录了我们的内存使用情况，可以看到我们内存容量为 494M（506268/1024），当前已使用了 93M，还剩下 400M 可用。shared 表示表示的是多个进程共享的内存总量，这里只占有了 580 个字节；Buffers/cached 表示当前磁盘缓存的大小为 31476 个字节。

```
root@imx6ull14x14evk:~# free
```

	total	used	free	shared	buffers	cached
Mem:	506268	95924	410344	580	0	31476
-/+ buffers/cache:		64448	441820			
Swap:	0	0	0			

图 3-3 内存使用情况

3.1 查看 FLASH 存储器容量

/proc/partitions 文件包含了存储器的分区信息，查看分区信息可以了解板子的 Nand-FLASH 存储器容量。

可使用如下命令查看：

```
cat /proc/partitions
```

其单位为 blocks 的数量，对于本示例的开发板，Nand-FLASH 的每个 Block 大小为 1024Byte，eMMC 则为 512Byte。

3.1.1 Nand-Flash 存储器

图 3-4 是使用 256MB Nand-FLASH 开发板的命令输出信息。

```
root@imx6ull14x14evk:~# cat /proc/partitions
major minor #blocks name

1        0        65536 ram0
1        1        65536 ram1
1        2        65536 ram2
1        3        65536 ram3
1        4        65536 ram4
1        5        65536 ram5
1        6        65536 ram6
1        7        65536 ram7
1        8        65536 ram8
1        9        65536 ram9
1       10        65536 ram10
1       11        65536 ram11
1       12        65536 ram12
1       13        65536 ram13
1       14        65536 ram14
1       15        65536 ram15
31        0        65536 mtdblock0
31        1       16384 mtdblock1
31        2       16384 mtdblock2
31        3      163840 mtdblock3
```

图 3-4 256MB Nand-FLASH 开发板的分区信息

其中的 mtdblock 开头的都是属于 Nand-FLASH 存储器的数据块，把它们所有的 blocks 加起来就可以算出容量(每个 block 大小为 1024Byte)：

```
65536+16384+16384+163840 (Block)
= 262144*1024 (Byte)
```

$$= 262144 * 1024 / 1024 / 1024 \text{ (MByte)}$$
$$= 256 \text{ MByte}$$

3.1.2 eMMC 存储器

图 3-5 是使用 8GB eMMC 开发板的命令输出信息。

```
root@imx6ull14x14evk:~# cat /proc/partitions
major minor #blocks name
1        0      65536 ram0
1        1      65536 ram1
1        2      65536 ram2
1        3      65536 ram3
1        4      65536 ram4
1        5      65536 ram5
1        6      65536 ram6
1        7      65536 ram7
1        8      65536 ram8
1        9      65536 ram9
1       10      65536 ram10
1       11      65536 ram11
1       12      65536 ram12
1       13      65536 ram13
1       14      65536 ram14
1       15      65536 ram15
179      0    7634944 mmcblk1
179      1     512000 mmcblk1p1
179      2    7020544 mmcblk1p2
179     24         512 mmcblk1rpm
179     16       4096 mmcblk1boot1
179      8       4096 mmcblk1boot0
```

图 3-5 8GB eMMC 开发板的分区信息

其中的 mmcblk 开头的都是属于 eMMC 存储器的数据块，把它们所有的 blocks 加起来就可以算出容量（每个 block 大小为 512Byte）：

$$7634944 + 512000 + 7020544 + 512 + 4096 + 4096 \text{ (Block)}$$
$$= 15176192 * 512 \text{ (Byte)}$$
$$= 15176192 * 512 / 1024 / 1024 / 1024 \text{ (GByte)}$$
$$= 7.2 \text{ GByte}$$

算出的最终结果比 8G 小一点，这跟 SD 卡标称值比实际值小的原因一样，不要纠结。

销售与服务网站

东莞野火电子科技有限公司

地址：东莞市大岭山镇石大路 2 号艺华综合办公大楼 301

官网：www.embedfire.com

电话：0769-33894118

论坛：www.firebbs.cn

邮箱：firege@embedfire.com

QQ：313303034

淘宝：<https://fire-stm32.taobao.com>



关注野火公众号，可免费获取野火全部产品的资料。

附录

- Putty 软件下载地址: <https://putty.org>, 推荐下载其中的 zip 包, 它包含其它丰富的组件。