

# SG2042 计算平台

## 用户手册

User Manual

文档版本: V0.2

发布日期: 2023/03/04

## 版本信息

日期	版本	说明
2023.02.28	V0.1	1.初始版本
2023.03.04	V0.2	1.内容补充

## 法律信息

**版权所有** © 2023 澎峰（北京）科技有限公司。本公司保留一切权利，本文档中的内容可随时更改，恕不另行通知。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

### 商标声明

PerfMPL®等为澎峰（北京）科技有限公司公司注册商标。

本文档所提及的其他所有商标或注册商标，由各自的所有人拥有。

### 澎峰科技

地址：北京市海淀区紫雀路 55 号院 9 号楼翠湖科创平台 304 室

邮箱：xianyi@perfxlab.com

中文网址：www.perfxlab.cn, www.perfxlab.com

# 目 录

版本信息 .....	2
法律信息 .....	3
基础软件环境 .....	5
1. 软件源码 .....	5
2. 系统软件 .....	5
SD 卡镜像制作说明 .....	7
1. 环境要求 .....	7
2. 使用源码构建镜像 .....	7
3. 制作 SD 卡的准备工作 .....	9
4. 获取 Ubuntu 镜像 .....	9
5. 制作可引导的 microSD 卡 .....	9
方法 1: 使用 balenaEtcher .....	9
方法 2: 直接使用 dd 命令 .....	10
6. 从 microSD 引导启动 .....	13
7. 使用 NVMe SSD 和 microSD 卡 .....	13
常见问题说明 .....	14

## 基础软件环境

### 1. 软件源码

SG2042 计算平台的相关软件资源均采用开源方式, 欢迎有兴趣的开发者和团队一起加入开发和维护, 有关的软件源码包获取地址:

软件类别	下载路径
Bootloader	<a href="https://github.com/sophgo/bootloader-riscv">https://github.com/sophgo/bootloader-riscv</a>
Linux Kernel	<a href="https://github.com/sophgo/linux-sophgo">https://github.com/sophgo/linux-sophgo</a>
BIOS	TBD
Compiler	TBD
官方镜像	<a href="http://219.142.246.77:65000/sharing/dKlPHukbe">http://219.142.246.77:65000/sharing/dKlPHukbe</a>

### 2. 系统软件

为方便用户使用, 降低用户开发难度并节约用户开发时间, 澎峰科技在 SG2042 计算平台的系统中集成了丰富的高性能计算工具和相关的加速库。相关的系统软件信息如下表所列:

分类	软件名称	安装路径	版本号
操作系统	Ubuntu	/	22.04.1
		/	22.10
编译器	ZCC	TBD	TBD
计算库	PerfMPL	TBD	TBD
计算框架	PerfXAPI	TBD	TBD
MPI	OpenMPI	TBD	TBD
Benchmarks	hpl	TBD	TBD

	stream	TBD	TBD
	iozone	TBD	TBD

# SD 卡镜像制作说明

## 1. 环境要求

构建镜像需要安装有如下 Linux 操作系统的 PC 环境：

- Fedora 系统
- Ubuntu 系统

另外因为制作镜像需要构建 uroot，uroot 编译依赖 go 编译器，因此必须确认系统已经安装 go 1.17，Ubuntu 环境下的 go 1.17 安装方法可参考此网页指引安装：<https://tecadmin.net/how-to-install-go-on-ubuntu-20-04/>。

安装后可参考下面的命令检查 go 版本是否正确。

```
$ go version
go version go1.17 linux/amd64
```

## 2. 使用源码构建镜像

1. 使用 git 命令下载源码包

```
$ git clone https://github.com/sophgo/bootloader-riscv.git
$ git clone https://github.com/sophgo/linux-sophgo.git
```

2. 构建用于 BSP 包的交叉编译工具 GCC（RISC-V 版）

进入 bootloader-riscv 与 linux-sophgo 所在的同级别文件目录，使

用下列命令构建用于 RISC-V 的交叉编译工具链：

```
$ CHIP=mango
$ source bootloader-riscv/scripts/envsetup.sh
$ build_rv_gcc
```

3. 命令正确执行完毕后可以得到如下的文件夹结构：

```

.
├── bootloader-riscv
├── linux-sophgo
└── gcc-riscv
    ├── gcc-riscv64-unknown-elf
    └── gcc-riscv64-unknown-linux-gnu

```

4. 镜像构建命令如下：

```

$ CHIP=mango
$ source bootloader-riscv/scripts/envsetup.sh
$ build_rv_all

```

5. 命令执行完毕后，生成的输出文件位于 install/soc\_mango/riscv64 文件目录下：

```

.
├── bsp-debs
│   ├── linux-headers-5.19.17+.deb
│   ├── linux-image-5.19.17+-dbg.deb
│   └── linux-image-5.19.17+.deb
├── fw_jump.bin
├── fw_jump.elf
├── initrd.img
├── mango.dtb
├── mango_evb_v0.1.dtb
├── mango_multi_chips.dtb
├── mango_multi_chips_pld.dtb
├── mango_pld.dtb
├── riscv64_Image
└── rootfs.cpio

```



```
|— sd.img  
|— vmlinux  
└— zsbld.bin
```

### 3. 制作 SD 卡的准备工作

- 能运行 Linux OS 的 PC 环境；
- SG2042 EVB；
- microSD 卡一张(16GB 或以上容量)，microSD 卡读卡器；
- NVMe SSD（可选）；
- 串口线（MicroUSB - USB Type-A）；

### 4. 获取 Ubuntu 镜像

- 直接下载镜像：[Ubuntu image](#)。该镜像基于 Ubuntu 官方预装的服务器镜像构建。
- 或者使用你自己编译好的 sd.img 文件，位于目录 install/soc\_mango/riscv64。

以下均使用 sd.img 来指代 Ubuntu 镜像。

### 5. 制作可引导的 microSD 卡

#### 方法 1：使用 balenaEtcher

1. 下载并安装：[balenaEtcher](#)。
2. 打开 balenaEtcher，点击“Flash from file”按钮并选择你需要使用的 sd.img。

3. 点击 “Select target” 按钮并选择你需要写入 sd.img 的 microSD 卡
4. 点击 “Flash!” 按钮，等待写入完成。

## 方法 2：直接使用 dd 命令

1. 使用 dd 命令写入 sd.img 至 microSD 卡：

```
# To find the block device name of your microSD Card.  
# For example, the microSD Card drive is /dev/sdc. Checking the name  
of your device is a key step,  
# as writing to the wrong device might corrupt or destroy your data.
```

```
$ sudo dd if=sd.img of=/dev/sdc bs=32M
```

```
160+0 records in  
160+0 records out  
5368709120 bytes (5.4 GB, 5.0 GiB) copied, 108.587 s, 49.4 MB/s
```

2. 重新调整 microSD 卡的 root 分区大小(可选)：

```
# Change partition table of your microSD Card.
```

```
$ sudo fdisk /dev/sdc
```

```
Welcome to fdisk (util-linux 2.37.2).
```

```
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.
```

```
Command (m for help): p
```

```
Disk /dev/sdc: 29.72 GiB, 31914983424 bytes, 62333952 sectors
```

```
Disk model: MassStorageClass
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0x5c9f9baa

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	262143	260096	127M	c W95	FAT32 (LBA)
/dev/sdc2		262144	524287	262144	128M	c W95	FAT32 (LBA)
/dev/sdc3		524288	10485759	9961472	4.8G	83	Linux

Command (m for help): d

Partition number (1-3, default 3): 3

Partition 3 has been deleted.

Command (m for help): n

Partition type

p primary (2 primary, 0 extended, 2 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (3,4, default 3):

First sector (524288-62333951, default 524288):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (524288-62333951, default 62333951):

Created a new partition 3 of type 'Linux' and of size 29.5 GiB.

Partition #3 contains a ext4 signature.

Do you want to remove the signature? [Y]es/[N]o: N

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read partition table.

*Syncing disks.*

*# Check partitions of your microSD Card.*

*\$ sudo fdisk -l /dev/sdc*

*Disk /dev/sdc: 29.72 GiB, 31914983424 bytes, 62333952 sectors*

*Disk model: MassStorageClass*

*Units: sectors of 1 \* 512 = 512 bytes*

*Sector size (logical/physical): 512 bytes / 512 bytes*

*I/O size (minimum/optimal): 512 bytes / 512 bytes*

*Disklabel type: dos*

*Disk identifier: 0x5c9f9baa*

<i>Device</i>	<i>Boot</i>	<i>Start</i>	<i>End</i>	<i>Sectors</i>	<i>Size</i>	<i>Id</i>	<i>Type</i>
<i>/dev/sdc1</i>		<i>2048</i>	<i>262143</i>	<i>260096</i>	<i>127M</i>	<i>c</i>	<i>W95 FAT32 (LBA)</i>
<i>/dev/sdc2</i>		<i>262144</i>	<i>524287</i>	<i>262144</i>	<i>128M</i>	<i>c</i>	<i>W95 FAT32 (LBA)</i>
<i>/dev/sdc3</i>		<i>524288</i>	<i>62333951</i>	<i>61809664</i>	<i>29.5G</i>	<i>83</i>	<i>Linux</i>

*# Force checking your file system.*

*\$ sudo e2fsck -f /dev/sdc3*

*e2fsck 1.46.5 (30-Dec-2021)*

*Pass 1: Checking inodes, blocks, and sizes*

*Pass 2: Checking directory structure*

*Pass 3: Checking directory connectivity*

*Pass 4: Checking reference counts*

*Pass 5: Checking group summary information*

*cloudimg-rootfs: 79598/575424 files (0.0% non-contiguous),*

*1022378/1150203 blocks*

*xingxg@vmware:~/sophgo/install/soc\_mango/riscv64\$ sudo resize2fs*

*/dev/sdc3*

```
resize2fs 1.46.5 (30-Dec-2021)
```

```
Resizing the filesystem on /dev/sdc3 to 7726208 (4k) blocks.
```

```
The filesystem on /dev/sdc3 is now 7726208 (4k) blocks long.
```

3. 拷贝镜像至 microSD 卡 root 分区的/home/ubuntu:

```
$ cp sd.img /mnt/home/Ubuntu
```

## 6. 从 microSD 引导启动

1. 将 microSD 卡插入到 SG2042 EVB，并将串口线连接至电脑 USB 端口，等待驱动安装完成并识别，然后给 EVB 上电。
2. 输入登录名 ubuntu 以及登录密码 sophgo。
3. 任何操作均需要 sudo 权限。

## 7. 使用 NVMe SSD 和 microSD 卡

如果需要使用 NVMe SSD 和 microSD 卡组合的方式启动系统，需要执行以下步骤。

1. 使用 dd 命令拷贝 sd.img 到 NVMe 硬盘。
2. 重新调整 NVMe 硬盘的 root 分区大小。
3. 使用 fdisk 命令删除 microSD 卡的 root 分区，这一步非常重要，因为 microSD 卡和 NVMe 硬盘的 root 分区标签一样，会导致 CPU 无法识别。
4. 重启并且使用 NVMe 硬盘上的 Ubuntu 系统。

## 常见问题说明

1. 源码从哪里下载？

Github: <https://github.com/sophgo/>。

2. 登录名和密码不对？

登录名: ubuntu

登录密码: sophgo

3. 大于 8GB 内存不识别？

下载并使用最新的 bootloader-riscv 制作新的系统镜像（或下载最新的系统镜像），并重新制作 microSD 启动卡。

----- 此页留白 -----