

第 1 章



内核与根文件系统

嵌入式系统是面向产品、面向应用的专用计算机系统。嵌入式系统的软件包括系统软件和应用软件, 系统软件的核心是操作系统。在不同的应用中, 嵌入式操作系统的选择也是多种多样的。在高端的嵌入式处理器平台上, 大量的应用是基于 Linux 操作系统开发的。

本章介绍如何将 Linux 操作系统内核与根文件系统移植到树莓派上, 为后续扩展树莓派的功能打下基础。

1.1 树莓派简介

树莓派 (Raspberry Pi) 是一款卡片式计算机, 由英国一个非营利机构“树莓派基金会”开发, 其最初的目的是用于对少年儿童进行计算机普及教育。2012 年 2 月, 这款信用卡大小的单板计算机横空出世, 型号是树莓派 B 型, 随后不久又发布了一款低配版的 A 型。25 美元的价位、超高的性价比使树莓派迅速得到市场的回应, 至此一发不可收拾, 差不多每一两年就有一个新款发布, 性能也在不断提高。树莓派基金会还专门成立了一个企业部门, 由博通工程师 Eben Upton 任 CEO。

迄今为止的历代树莓派都采用博通公司的 SoC 处理器。第一代处理器型号是 BCM2835, 基于 Arm1176JZF-S 架构, 指令集为 Armv6Z, 属于 Arm11 系列, CPU 主频为 700MHz, 采用博通 VideoCore IV 图形处理器 (Graphics Processing Unit, GPU)。它的性能相当于 20 世纪末的 300MHz Pentium II 处理器, GPU 的性能与 2001 年的家用游戏机 Xbox 相当。

第二代树莓派曾短暂使用 BCM2836 处理器, 为 4 核 Cortex-A7 架构, 主频为 900MHz, 指令集为 Armv7-A。这仍然是一个 32 位的处理器, 但从开发者 Eben Upton 在 2015 年 2 月 2 日发布于树莓派官网 Blog 的信息来看, 其性能已比第一代产品提高了 4~6 倍。从第二代的 Pi 2Bv1.2 版开始普遍采用 BCM2837 系列的处理器, 它是一个四核的 Cortex-A53

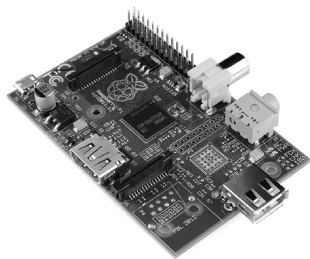
处理器,支持 64 位指令集Armv8-A,主频也从 900MHz 提高到 1.4GHz。

从第三代开始有了板载无线通信设备:蓝牙和 WiFi。在 Pi 3 B+ 上更是 2.4G/5G 双频 802.11ac WiFi 和千兆以太网。但由于这一代有线网是通过 USB 控制器 LAN7515/LAN9514 连接的,受 USB 2.0 限制,实际传输能力在 300Mb/s 左右。

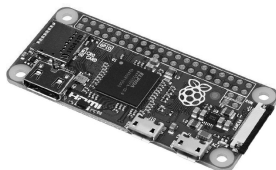
第四代树莓派采用 BCM2711 处理器,和第二代、第三代相比,主要的变化有:①将原有的一个标准 HDMI (High Definition Multimedia Interface, 高清晰度媒体接口) 换成了两个 microHDMI, 扩展了显示能力;②增加了 USB 3.0 接口,以太网不再受制于 USB 2.0 的传输能力;③将电源接口从 microUSB 换成了 USB Type-C, 提高了供电能力。尤其重要的是,处理器架构从 Cortex-A53 升级到 Cortex-A72, 虽然主频仅比 Pi 3 B+ 提高了 0.1GHz, 但性能有了大幅度的提升, GPU 升级到 VideoCore VI, 主频也从 400MHz 提高到 500MHz。原有的 GPU 支持 OpenGL ES 2.0, 新的 GPU 支持 OpenGL ES 3.2。表 1.1 是截至 2019 年发布的各种型号的树莓派,图 1.1 是树莓派代表性型号的外观。

表 1.1 树莓派系列

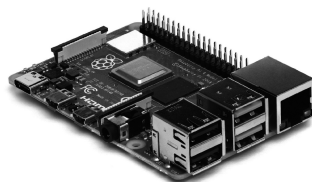
型号	处理器架构	指令集	发布时间
Pi B	Arm1176JZF-S	Armv6Z	2012
Pi A	Arm1176JZF-S	Armv6Z	2013
Pi B+	Arm1176JZF-S	Armv6Z	2014
Pi A+	Arm1176JZF-S	Armv6Z	2014
Pi Zero	Arm1176JZF-S	Armv6Z	2015 ^①
Pi 2 B	4xCortex-A7	Armv7-A	2015
Pi 3 B	4xCortex-A53	Armv8-A	2016
Pi 3 B+	4xCortex-A53	Armv8-A	2018
Pi 3 A+	4xCortex-A53	Armv8-A	2018
Pi 4 B	4xCortex-A72	Armv8-A	2019 ^②



(a) Pi 1 A



(b) Pi Zero



(c) Pi 4 B

图 1.1 树莓派外观

① 2017 年推出带 WiFi 的版本。

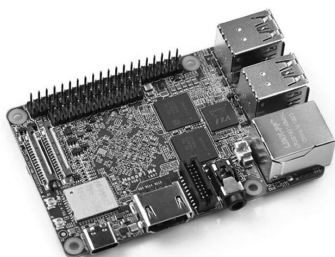
② Pi 4 有多种板载内存配置。2019 年刚推出时有 1GB、2GB、4GB 三个版本,2020 年 5 月推出 8GB 版本。

2020年12月,树莓派发布了装在键盘里的计算机 Pi 400 (见图 1.2),它是在 Pi 4 的基础上包装了一个键盘,只要外接一个显示器就是一台完整的计算机。

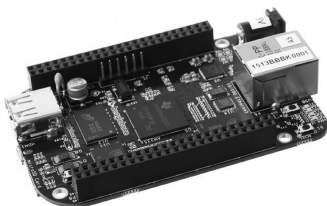


图 1.2 Pi 400 (键盘里的计算机)

到 2019 年底,各种型号的树莓派已累计售出 3000 万件。由于树莓派的成功,其他一些计算机开发商也仿照树莓派开发了类似的卡片式计算机产品 (见图 1.3)。



(a) Nano Pi(2015.8)



(b) BB-Black(2013.4)



(c) BananaPi-M3(2015.12)

图 1.3 各种卡片式计算机产品

树莓派这类计算机结构简单、体积小、耗电低,却拥有与普通计算机几乎相同的功能和性能,可以很方便地植入各种应用系统中。这种具有计算机的基本结构但又不具备普通计算机形态的计算机,业界称为“嵌入式系统”。所谓的“嵌入式”是指它是嵌在产品中的,是面向产品的专用计算机。人们看到的只是产品本身,看不到计算机。目前大部分的计算机产品都属于嵌入式计算机。嵌入式系统广泛应用于工业控制、机电、航天、通信、环境监测、汽车电子、家用电器等各种使用微处理器系统的环境。

1.1.1 核心处理器

计算机系统的硬件核心是 CPU (Central Processing Unit, 中央处理器)。对于通用计算机来说, CPU 是一个独立的芯片,大家所熟知的个人计算机中的 Intel 或 AMD 的处理器皆如此。CPU 还需要配合其他外设 (如中断控制器、I/O 接口、总线、存储器等) 才能

组成一个计算机系统。但对于嵌入式处理器来说就是另一回事了。嵌入式处理器中,除了 CPU 以外,本身就已经包含了系统所需要的各种片内接口和存储器,其中最具技术含量的就是 CPU。嵌入式处理器的 CPU 又被称作内核 (core)。目前比较著名的嵌入式内核包括 Arm、MIPS、PowerPC 等。

所有树莓派的处理器都使用博通的 SoC (System on Chip) BCM283X 系列,该处理器属于 Arm 体系结构。前期的处理器是 32 位的,树莓派第三代和第四代处理器是 64 位的核心架构 (Cortex-A53 和 Cortex-A72)。

1.1.2 树莓派操作系统

与通用计算机一样,嵌入式计算机也是由硬件和软件构成的。软件中最重要的就是操作系统核心。根据树莓派官方网站 <https://www.raspberrypi.org> 提供的信息,树莓派支持 Android、FreeBSD、OpenBSD (BSD UNIX 的变种)、Plan 9 (源于 Bell 实验室的 UNIX 分布式操作系统)、Windows 10 Iot Core、RISC OS, 以及多种 Linux 发行版。大多数树莓派应用都基于 Linux 操作系统内核,官网提供的 Linux 发行版名为 Raspberry Pi OS,它是基于 Debian 的针对树莓派的定制版本。

1.1.3 树莓派接口

图 1.4 是树莓派 4B 正面布局。它有 4 个 USB 主控制接口 (其中 2 个是 USB 3.0)、1 个 RJ45 有线网接口、耳机音频输出插口、2 个微型高清视频 HDMI 输出接口和一组 20×2 的 GPIO 引脚。板上的 CSI (Camera Serial Interface, 摄像头串行接口) 用于连接摄像头、DSI (Display Serial Interface, 显示器串行接口) 可以连接显示设备。树莓派本身不带音频输入接口,但可以通过 I2S (Inter-IC Sound, 又写作 IIS) 接 A/D 转换器或 USB 声卡等设备实现音频输入。系统使用 USB Type C 接口供电。4 代之前的 B 型板上只有一个标准的 HDMI 输出,4 个 USB 2.0 接口,供电通过 microUSB。对于不同型号,其某些接口的位置也会有所不同。

1.2 开发工具

由于目标系统是 Linux 操作系统,因此本书所有开发过程均在 Linux 操作系统上完成,PC 开发环境是 Ubuntu/x86-64。大多数针对 Linux 的开发工具在 Linux 平台上都可以很方便地获得,而且在开发过程中,使用相同的操作系统,PC 和目标平台可以比较方便地对照操作。

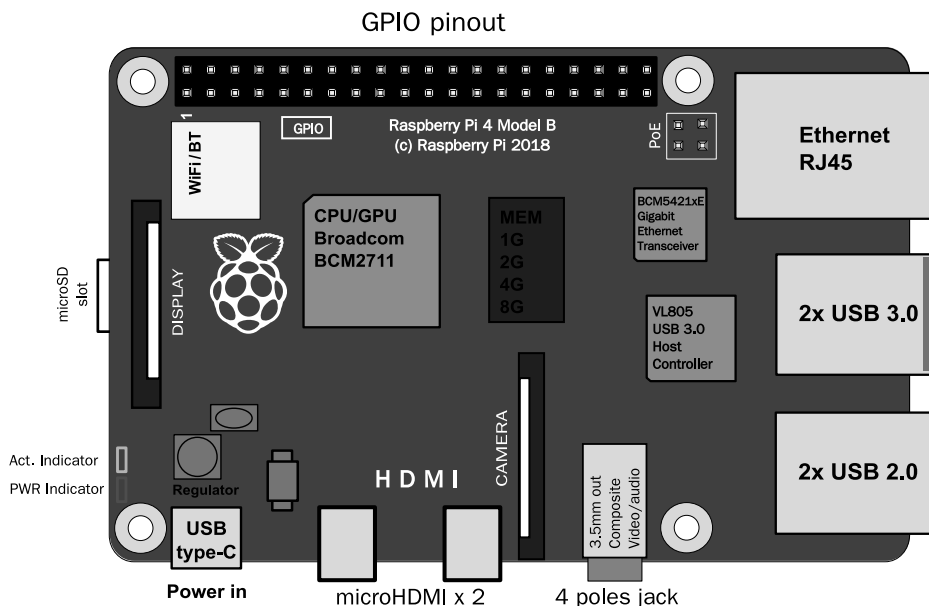


图 1.4 树莓派 4B 接口

原则上说, 使用 Linux 的任何发行版, 甚至是在其他操作系统中的 Linux 虚拟机, 本书所涉及的方法依然有效, 只是不同 Linux 发行版的开发工具安装会有一些差别。

1.2.1 编译工具

由于树莓派 4B 的处理器采用的是 Cortex-A 架构, 使用 Arm 指令集, 与 PC 的 x86 处理器使用的指令集完全不同。因此 PC 的编译工具不能直接用于开发树莓派的软件, 需要另外安装一套针对 Arm 的编译工具。在 PC 上安装的这组工具被称为“交叉编译工具链”, 或简称“交叉编译器”。Arm 交叉编译器可以在 PC 上用源码编译生成, 也可以通过软件仓库直接安装二进制程序。源码编译安装过程比较烦琐, 如无特殊要求, 可以通过包管理器安装二进制程序。Debian 或 Ubuntu 系统可以通过下面的命令安装交叉编译工具:

```
# apt-get install g++-arm-linux-gnueabi
```

在命令行操作中, 提示符“#”表示超级用户权限, “\$”表示普通用户权限。在本书中, PC 端的命令用这两个不同的提示符来严格区分这两类用户。获取超级用户的权限可以通过 sudo 命令, 也可以用 su 命令直接切换到超级用户。在 PC 上, 各人有自己的使用习惯, 本书对此也不再区分, 仅以提示符“#”表示。而在树莓派个人应用系统上, 为了提高效率, 在没有特别要求的情况下, 一般都以超级用户的身份操作。

依照依赖关系,包管理器会安装所有 Arm 架构的 g++ 的基础包,包括 Arm 版本的 binutils 和 glibc。为了简化以后的命令操作,安装后,建议到安装路径下(一般是 /usr/bin)将所有带有 arm-linux-gnueabi- 前缀的命令进行 arm-linux- 前缀的链接,命令操作如下:

```
# for f in $(ls arm-linux-gnueabi-*); \
do ln -s $f $(echo $f|sed "s/gnueabi-//"); \
done
```

之后在交叉编译时不需要用 arm-linux-gnueabi-gcc 这么长的命令,只需输入 arm-linux-gcc 即可。

如果按 64 位 Arm 指令集 Armv8-A 编译,则需要安装 aarch64-linux-gnu 编译工具链:

```
# apt-get install g++-aarch64-linux-gnu
```

按类似的方法,进行一组 aarch64-linux- 前缀的链接。

本书以树莓派 4B 为研究对象。由于树莓派 4B 的处理器同时支持 32 位和 64 位指令集,为了更大限度地发挥系统性能,采用 64 位指令集,交叉编译器也使用 aarch64-linux- 前缀。

理论上说,也可以将编译器直接安装在树莓派系统上。构建针对目标系统的编译环境将在第 5 章介绍。但由于树莓派资源有限,性能也不及 PC,不适宜大批量的软件编译,而且最初阶段的移植也必须在 PC 上编译。对于有明确应用背景的嵌入式系统来说,在目标系统上安装开发环境的意义也不大。

1.2.2 调试接口

树莓派提供串口调试功能。在基本系统安装阶段,开发人员可以用串口调试器连接到树莓派上,用于系统监控。目前多数 PC 已没有专用的 RS-232 串口,图 1.5(b) 是一种 USB-232 的适配器(转接器),可作为调试器。在 PC 上,可以使用 minicom 或 screen 命令进行串口调试,这里推荐使用 minicom。监控设备文件是 /dev/ttyUSBx。设备文件名中的数字序号可能会有变化,可用命令 dmesg 检查插入的 USB-232 适配器生成的设备文件名,例如:

```
$ dmesg | tail
...
[ 1930.183656 ] usbcore: registered new interface driver pl2303
[ 1930.183664 ] usbserial: USB Serial support registered for pl2303
[ 1930.183673 ] pl2303 1-5:1.0: pl2303 converter detected
[ 1930.184225 ] usb 1-5: pl2303 converter now attached to ttyUSB0
```

不同型号的串口适配器可能会有不同的显示。当操作系统正常启动,完成网络安装配置之后,还可以使用网络方式连接树莓派。

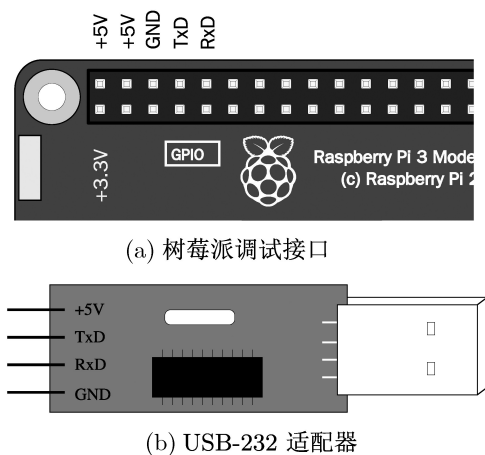


图 1.5 实验系统连接示意图

串口数据发送端标记为 TxD, 接收端标记为 RxD。图 1.5中, 将调试器的 TxD、RxD 与树莓派的 RxD (pin 10)、TxD (pin 8) 对接, 即一方的发送端连到对方的接收端, 地线连到树莓派的 GND (pin 6)。树莓派通过专门的线路供电, 串口适配器的 +5V 不用连接。

在 PC 上, 设备文件 `/dev/ttyUSBx` 属于 `dialout` 组。为避免 `minicom` 频繁越权操作, 建议将开发者个人用户加入 `dialout` 组。可以使用如下的 `adduser` 命令 (这里假设开发者用户名为 `user`):

```
# adduser user dialout
```

也可以直接用文本编辑器编辑 `/etc/group` 文件, 在 `dialout` 行上添加 `user` 用户。添加的组员权限在下一次登录后生效。

`minicom` 的选项中包含了指定设备和参数的内容, 为简化操作, 首次使用 `minicom` 时, 用带有 `-s` 的选项启动, 以便于直接打开串口配置界面:

```
$ minicom -s
```

在串口设置功能中 (serial port setup), 按图 1.6 设置串口设备 (`/dev/ttyUSB0`)、格式 (115200 8N1, 波特率 115200b/s, 数据位 8 位, 无校验, 1 个停止位)。波特率和格式是由 Bootloader 和内核决定的, 使用串口与其他设备连接时, 这些参数应根据调试对象的要求设置。

设置好波特率、数据位、停止位、校验方式后, 选择 `Save setup as dfl` 保存设置参数, 再回到 `minicom` 主界面。以后再启动 `minicom` 时, 如果不改变通信参数, 不再需要 `-s` 选项。

```

+-----+
| A -   Serial Device       : /dev/ttyUSB0 |
| B - Lockfile Location    : /var/lock     |
| C -   Callin Program     :              |
| D -   Callout Program    :              |
| E -   Bps/Par/Bits       : 115200 8N1   |
| F - Hardware Flow Control : No          |
| G - Software Flow Control : No          |
|                                     |
|   Change which setting?              |
+-----+
| Screen and keyboard |
| Save setup as dfl   |
| Save setup as..    |
| Exit                |
| Exit from Minicom   |
+-----+

```

图 1.6 设置串口参数

1.3 树莓派的外存储器

树莓派没有板载的只读存储器 (Read Only Memory, ROM), 软件需要安装在 microSD 卡上。为了方便软件管理, microSD 卡一般可以分成两个分区: 第一个分区安装引导器、内核映像; 第二个分区安装操作系统。当选择 Linux 操作系统时, 第一个分区格式化成 VFATFS (以下称 BOOT 分区), 第二个分区作为 Linux 的根文件系统, 必须支持 inode。Linux 支持的文件系统很多, 如 Ext2/Ext3/Ext4FS 文件系统、ReiserFS 文件系统、YAFFS 文件系统等。本书中统一使用 Ext4FS (以下称 Ext4 分区)。

1.3.1 SD 卡分区

将 microSD 卡插入 PC, 使用 fdisk 命令的 -l 选项, 可以看到当前所有磁盘的分区情况:

```

# fdisk -l
...
Disk /dev/mmcblk0: 3648 MB, 3825205248 bytes, 7471104 sectors
116736 cylinders, 4 heads, 16 sectors/track
Units: cylinders of 64 * 512 = 32768 bytes

Device            StartLBA      EndLBA        Sectors  Size Id Type
/dev/mmcblk0p1    2048         198655        196608  96.0M  e Win95 FAT16
/dev/mmcblk0p2    198656       7471103       7272448  3551M  83 Linux

```

一些笔记本带有 SD 卡插槽。通过 SD 适配器 [见图 1.7(a)] 插入笔记本电脑的 microSD 卡会被识别为 MMC 设备, 设备文件是 `/dev/mmcblk0` (也可能是 `/dev/mmcblk1`); 如果通过 USB 读卡器转接 (见图 1.7(b)), 设备文件名可能是 `/dev/sdb` 或 `/dev/sdc`, 取决于内核对设备的识别情况。操作时务必认清操作对象, 因为此时使用了超级用户权限, 错误的操作将可能导致 PC 系统软件故障。上面操作的是一个 4GB 的 SD 卡 (因计算方法问题, 实际显示的是 3684MB), 目前有两个分区, 第一分区 96MB、FAT16 格式 (分区格式 Id^①为 e), 剩余空间分给第二分区, Linux 格式 (分区格式 Id 为 83)。在本书讨论的系统设计中, 所有软件安装后可控制在 4GB 以内。如果使用 16GB 的 microSD 卡进行开发, 则基本可以不用操心存储空间问题。



图 1.7 microSD 卡适配器

用不带选项的 `fdisk` 命令进入分区操作环境:

```
# fdisk /dev/mmcblk0
```

首先使用 `fdisk` 的 `d` 命令删除原有的分区, 再使用 `n` 命令创建两个新的分区。未经压缩的 Linux 内核映像大小约 20MB, 再考虑 Bootloader 的占用, 第一分区可留出 256MB, 剩余部分留给第二分区。然后使用 `w` 命令将分区表写入 SD 卡, 保存分区划分方式, 退出 `fdisk` 环境。

再使用 `mkfs` 命令分别对这两个分区格式化:

```
# mkfs.vfat /dev/mmcblk0p1
# mkfs.ext4 /dev/mmcblk0p2
```

拔出 SD 卡再重新插入, 如果 PC 识别出两个磁盘分区, 表示以上分区无误。

1.3.2 Bootloader

Bootloader (引导加载器) 的目的是加载操作系统内核, 向内核传递参数, 加载根文件系统, 引导操作系统运行。Bootloader 还负责核心软件的升级。一旦操作系统启动, Bootloader

^① 分区格式 Id 是用于标记不同分区上文件系统格式的一个字节数字。

的任务便暂告终结，直到系统重启。

树莓派的 Bootloader没有开放源代码，但二进制代码允许免费获得。二进制代码在 <https://www.github.com/raspberrypi/firmware/>下。使用时，只需要将这个项目下的 boot 目录中的内容复制到 microSD 卡的 BOOT 分区。其中的.dtb 文件和 overlays 目录中的.dtbo 文件在编译内核时生成，可以将相应的文件进行替换。kernel*.img 是内核映像文件，其编译过程将在后面介绍。重要的配置文件有两个：config.txt 和 cmdline.txt (文本文件)。

config.txt 用于设置系统时钟、GPU内存分配、显示器设置、内核引导方式等。其中与内核启动相关的设置如下：

```
initramfs initramfs.gz followkernel
kernel kernel8.img
cmdline cmdline.txt
device_tree=bcm2711-rpi-4-b.dtb
```

它表明内核映像文件是 kernel8.img，内核启动的命令行信息写在 cmdline.txt 文件中，以 initramfs.gz 作为初始化 RAMDisk 根文件系统，设备树文件是 bcm2711-rpi-4-b.dtb。device_tree 项的设置不是必需的，Bootloader 会根据硬件平台选择相应的设备树文件。

Bootloader 向内核传递的启动参数写在 cmdline.txt 文件中，一般会传递下面的参数：^①

```
console=tty1 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 \
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline \
fsck.repair=yes rootwait
```

以上设置了本机显示器终端 tty1 和串口终端 ttyAMA0、根文件系统和类型。顾名思义，cmdline 是传递给内核的命令行参数，它必须写在一行。下面在向该文件添加内容时也不能无续行符换行。

从 Bootloader向 Linux 内核传递的参数通过 PROCFS 文件系统传到 /proc/cmdline 文件中，内核启动的初始化脚本会读取这些参数，根据这些参数确定系统的行为。

1.4 Linux 内核

树莓派支持多种操作系统，其中以 Linux 操作系统最为典型。Android 操作系统同样也基于 Linux 内核。本系统的构建从移植 Linux 内核开始。

^① 反斜线“\”是 Shell 命令行的续行符。命令行可以接受很长的字符，实际操作中多数情况无须换行。因为书籍版面的限制，这里把参数分多行打印。后面 Shell 脚本或编译配置命令时同样情况，不再一一说明。