



MCU嵌入式操作系统 发展趋势

及RT-Thread介绍



目录

- 微控制器市场情况
- RT-Thread介绍
- RT-Thread发展情况

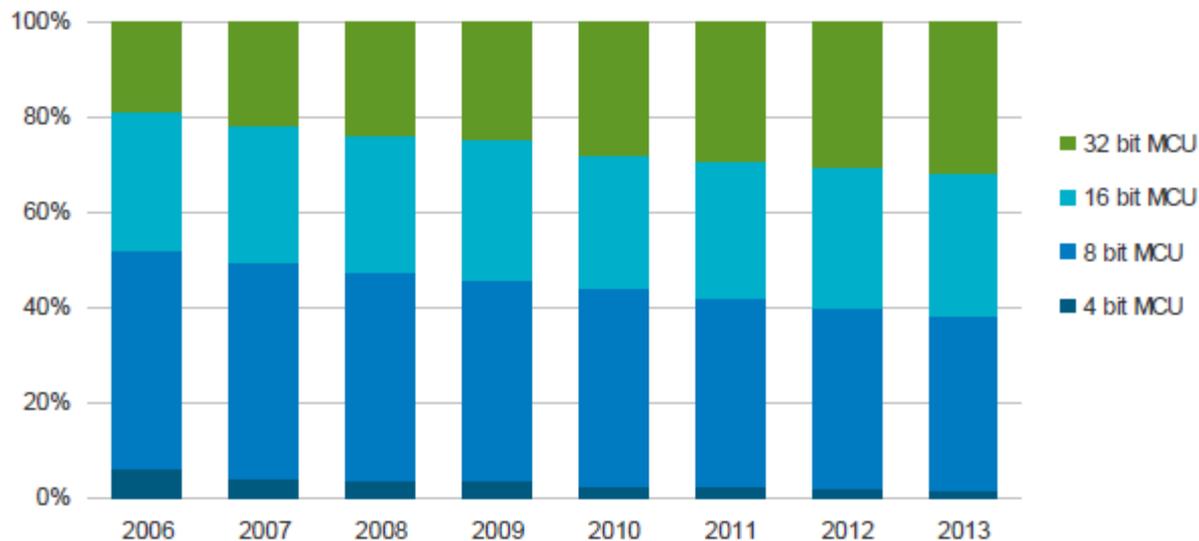
MCU现况

- 8位、16位单片机：
 - 大量产品依然在使用着8位、16位单片机；
- 32位处理器：
 - 体积越来越小，功耗也相应的降低
 - 成本、价格越来越低
- 以同等价位的32位处理器替换原有的8位单片机，越来越显示出它的性价比优势：
 - 功能更多；
 - 性能更强；

8位、32位处理器趋势图

China MCU Market share by Product 2006-2013

iSuppli



32 bit MCU's market share continue to increase in next few years; and 8 bit and 4 bit revenue will be declined. In Worldwide, the trend is more obvious than China.

微控制器

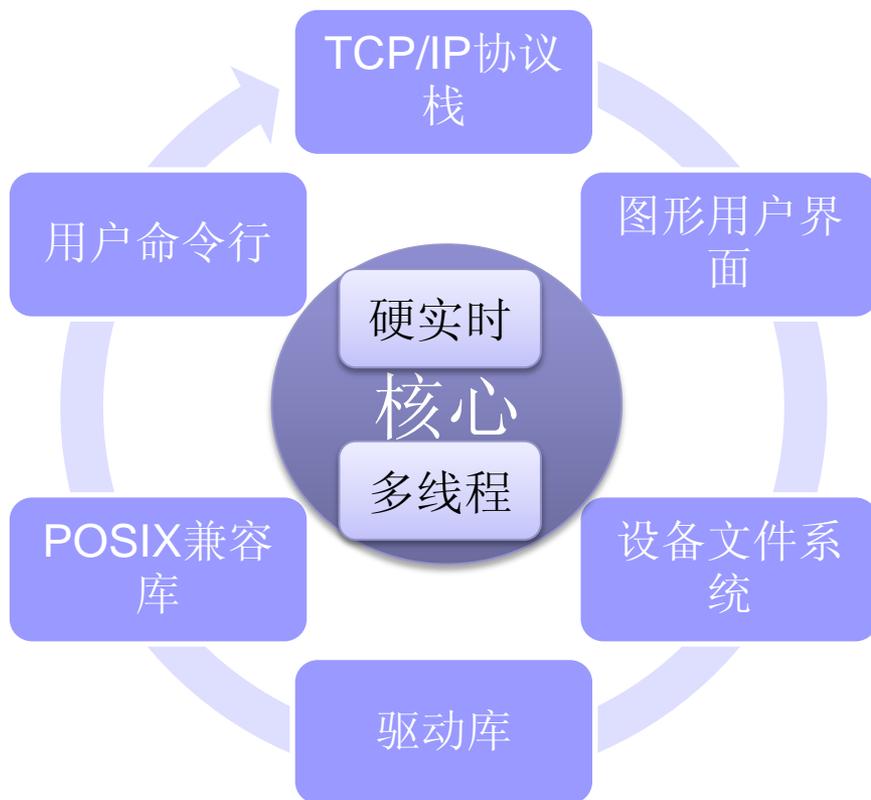
- 微控制器(Microcontroller Unit, 即MCU):
 - 通常指单芯片微型计算机, 被大量的应用在电机控制、条码阅读器、消费类电子、楼宇安全与门禁控制、工业控制与自动化和白色家电等领域。
- 32位微控制器的需求点:
 - 算法复杂性增加。
 - 将多个子系统合并到单个电子板中, 以便降低系统成本、减小占用空间、降低功耗
 - 包括有线和/或无线连接。
- ARM Cortex™ -M系列处理器
 - 低功耗、低成本
 - 周边生态链完整

RT-Thread介绍

- 2005年底，源于在ARM7TDMI芯片平台上开发一个小型的手持设备；
 - 当时的状况：
 - 商业OS: VxWorks、ucos
 - 免费OS: ecos, RTEMS, uclinux
 - 小型化，资源紧凑的情况下：
 - 自行开发一个小型的OS；
- 另外的初衷，希望有一个平台，以后写的应用都可以基于这个平台（可移植化好的平台）。

RT-Thread实时操作系统

- RT-Thread是一个开源的实时操作系统，包括：



开源社区方式

- 秉承开源理念，开放RT-Thread实时操作系统的源代码：
 - 2006年发布第一个版本；
 - 2010年开始发布RT-Thread 0.4.0系列版本；
- 2011年底将发布RT-Thread 1.0.0正式版本。
- 社区化方式发展：
 - 更开放，可以让更多的人投入进来；
 - 多样性，随着参与人的不同，大家的思维都是发散的。
 - 缺点：是否有凝聚力。

RT-Thread 近况

- 随着ARM Cortex-M系列芯片的应用、普及，原有很多8位、16位单片机都纷纷转到32位微处理器芯片中；
 - 自RT-Thread 0.3.0开始由于其实时性，小型的footprint，稳定的TCP/IP协议栈而被广泛应用。
 - 网友曾戏称：**ARM Cortex-M3 + RT-Thread是绝配**；
- 成为国内最大的原创性操作系统开发社区：
 - 涉及国内各地近30名开发人员；
 - 也包括一些国外开发成员。
- ARM Connected Community在中国的合作伙伴。

一些使用RT-Thread的案例

■ 打印机解决方案：

- 基于ARM Cortex-M3系列微控制器；
- 便利的高可移植性、跨平台性；
- LPC1700系列平台；

■ 电视台发射塔监控

- 基于以太网的无人职守发射塔；
- 高可靠性设备：对实时性、可靠性都有非常高的要求。
- TI LM3S系列平台；

RT-Thread目前的发展情况

- 更多的便利性：
 - 驱动、驱动框架一直是嵌入式操作系统、实时操作系统的大难题。
 - 从字符设备、块设备；
 - 到一些BUS设备：SPI BUS、USB Host/Device/OTG、SDIO等；
 - 甚至是一些Audio设备，多机通信等。
 - 针对稳定的数据存储，也将包括更多的Flash上的文件系统：
 - FAT on NFTL (NAND Flash Translation Layer)
 - YAFFS2
 - JFFS2等

RT-Thread目前的发展情况

■ 更多的互联性：

- 设备间的互联性也随着物联网概念的提出而越来越被提及。
- TCP/IP v4/v6协议栈；
- 无线网络，WIFI，3G上网卡等。

RT-Thread目前的发展情况

■ 更大的包容性：

- 构架上，不仅仅支持ARM芯片，也会包括一些FPGA软核；
- 这个也是社区多样性、思维发散性的体现。

未来.....

- 32位系统将被应用得越来越多，
- 嵌入式操作系统也将朝着越来越开放的方式发展，
- 用户希望拿到手的是一个马上能用的系统，而不是需要自己去一个个编写驱动。