RT-Thread 现状与下一版本展望

http://www.rt-thread.org

Bernard Xiong
bernard.xiong@gmail.com
微博：@午夜熊

RT-Thread的开发方式、流程

- 社区方式开发
  - 维护人 （maintainer）
    - + 开发人员 （developer）
  - 内核/组件/分支
  - 维护人也担当测试、评审员的角色
- 稳定性依赖于社区的测试及发布前的测试
  - 也试图建立一些自动化测试工具

RT-Thread的开发方式、流程

- Google SVN:
  - ***https***://rt-thread.googlecode.com/svn/trunk/
- wiki:
  - http://www.rt-thread.org/dokuwiki/doku.php

- 社区需求/兴趣、解决问题 →
  - 新组件 → 提交到svn中 → 成为新组件维护人
  - 新分支 → 提交到svn中 → 成为新分支维护人
  - 补丁 → 维护人 review → 合并到svn中

RT-Thread 0.4.0 分支

- 2010.10 发布0.4.0 beta1版本
- 2011.07 发布0.4.0 beta2版本

- 2011.10 预计发布 0.4.0 rc版本
  - 这个版本发布后，相应的功能都全部冻结，不会再行做大的修改；
  - 如果没什么问题，将演变为正式版本。

- 2011.12 预计发布这个分支的正式版本

0.4.0 分支的目标

- 更好的可移植性；
- 更好的易用性；
  - 反映出RT-Thread的设计哲学：simple
  - 能够相当于ecos, RTEMS等国外实时操作系统

POSIX标准

- POSIX Threads <不依赖于编译器>
- POSIX libdl

- 完整的newlib移植
  - 提供C运行库接口
  - 为stdio相匹配的，提供设备文件系统。

应用模块

- 传统RTOS，通常应用与内核牢牢的绑定在一起，整个系统是一个线性地址空间：
  - 系统规模
  - 团队规模
- 当前包括ARM9支持，ARM Cortex-M3上实现了SRAM运行的支持，在积极探索放在Flash上的实现；
- 应用模块的编程模式与通常用户程序完全相同，main函数入口，执行完退出；
  - 不同点在于libc中stdio的处理，整个系统只有一个全局的stdin/stdout/stderr。

新增的平台分支

- 0.4.0 相比较 0.3.x而言，增加了很多的平台分支，甚至是一些新的芯片架构：
  - Cortex-M0架构
  - MIPS架构
  - AVR32架构
  - NIOS-II架构等等
- 但相比较而言，这些新的平台分支也存在着一些问题！！

新增的平台分支：EFM32分支

- 分支维护人：王罡 (onelife)
  - Master of Science in Embedded System at Nanyang Technological University
  - Embedded Software Engineer at Eminent Group Pte. Ltd.

- 俺很荣幸能够参加到 RT-Thread 这样一个优秀的国产开源项目中来。俺知道在这个项目上高手如云、卧虎藏龙，所以其实是本着学习的态度在向各位仁兄讨教。 倘若真的能尽一点微薄之力，实在是让俺喜不自禁。

- 在此恭请各位仁兄向俺（轻）拍砖！
- 俺很遗憾不能到现场和仁兄们交流，谨此遥祝大家交流愉快！

新增的平台分支：AT91SAM9260

- 分支维护人：罗辉
  - 5年linux驱动开发经验，对vxworks和ucosii等实时操作系统有一定的了解。
  - 目前供职于杭州一家安防企业，主要 从事linux BSP研发工作。

- 于2010年年初开始关注rt-thread，年底开始正式着手RT-Thread AT91SAM9260分支移植；
- 目前基本搭建好了9260分支的基本开发环境，另外，SD卡驱动协议层基本开发完毕。

组件的变化：网络

- LwIP，轻型协议栈 1.3.2版本及1.4.0版本
  - 解决邮箱溢出的问题；
  - 解决errno的问题；
  - 解决网络接口断开、连接的问题，以及DHCP的问题；
- uIP，无缝与RT-Thread兼容
  - 加入类似LwIP的兼容层；
  - 维护人：itspy@深圳

辅助工具

- scons构建工具
  - 灵活的命令行编译；
  - 动态生成Keil MDK、IAR等集成开发环境工程文件；
- Eclipse工程构建向导
  - 基于scons构建工具；
  - 方便的Eclipse/CDT开发环境；

RT-Thread 0.4.0正式定型

- 经典的RT-Thread 0.3.x分支
- 稳定的RT-Thread 0.4.0分支
- 可以称之为 RT-Thread 1.0.0版本

下一代 RT-Thread 版本

- 名称？
- 英文名称，实际上更应该具备中文名称。

下一代 RT-Thread 版本

- 设备框架
  - rt_device_init
  - rt_device_open/close
  - rt_device_read/write
  - rt_device_control
- 目前包含字符型设备、块设备、网络设备、图形设备、RTC、声音设备等。
- 实际设备类型还有很多：
  - 如何得到一个剪裁方便，驱动容易编写的框架；
  - 更多的面向对象特性，抽象操作方法形成ops列表；

下一代 RT-Thread 版本

- USB
  - 相比较UART而言，USB更方便，速度更快。
  - 有主从之分，也更复杂
  - USB Device Stack
  - USB Host Stack

下一代 RT-Thread 版本

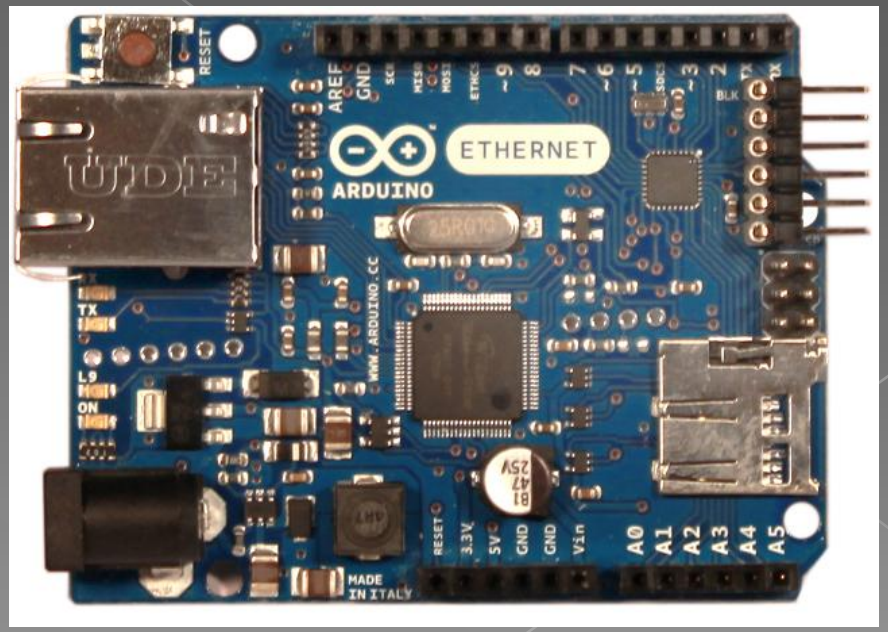
- BSD协议栈
  - 有了LwIP、uIP，是否还需要其他协议栈？
  - 面临的问题：
    - LwIP的路由转发功能比较弱；
    - 也有些用户反映LwIP协议栈稳定行不足。
- 其他一些考虑：
  - IPv6
  - 6LoWPAN: IPv6 over Low-Power wireless Area Networks

下一代 RT-Thread 版本

- Flash上的文件系统
  - UFFS
    - 在0.4.x分支上已经有部分支持，但驱动及测试还需要投入精力；
  - YAFFS2
    - prife - 电子科技大学在读硕士生
  - NFTL for FAT

下一代 RT-Thread 版本

- Arduino
  - 是一个基于开放源代码的软硬件平台，构建于开放源代码 simple I/O 接口板，并且具有使用类似 Java，C 语言的Processing/Wiring开发环境。



下一代 RT-Thread 版本

- Arduino
  - 国外非常活跃的开源软硬件平台，有很多shield扩展；
    - 特点1：简单
      - 硬件简单，接口标准
    - 特点2：简单
      - 软件简单，使用的是从C++改装的简化语言；
    - 特点3：简单
      - IDE简单，下载简单，一个编辑 器 + USB电缆就搞定。
  - ARM平台的Arduino
    - 原始硬件是ATMEL AVR，8位单片机
    - 新一代ARM芯片的Arduino

下一代 RT-Thread 版本

- RT-Thread for Arduino版本：
  - 社区方式发展
    - 针对于入门级别
  - 硬件：基于ARM Cortex-M3
  - 软件：
    - RT-Thread + Arduino 软件库
    - 多Arduino模块方式运行
      - 尽量不更改Arduino原有的使用方式、开发方式，而是把Arduino的一个loop做为一个应用模块线程方式运行。

下一代 RT-Thread 版本

- 针对于Arduino再提USB：
  - 基于USB方式的应用模块烧写器；
  - ADB over USB的实现；