第 37 卷 第 2 期
Vol.37 No.2

计 算 机 工 程
Computer Engineering

2011 年 1 月
January 2011

• 开发研究与设计技术 • 文章编号：1000—3428(2011)02—0256—02 文献标识码：A 中图分类号：TP316

TinyOS 2.0 在 CC2430 上的移植

徐敬东，赖锡盛

(南开大学信息技术科学学院，天津 300071)

摘 要：无线传感器网络操作系统的移植是指修改硬件无关层使其在不同的硬件平台上运行。为此，在分析嵌入式操作系统 TinyOS 2.0 的层次机构和硬件抽象层的基础上，阐述如何实现基于 SoC 方案的低功耗 CC2430 平台的移植，并成功移植了部分底层模块。测试结果表明，已移植模块能够正常工作，为进一步移植工作打下了良好的基础。

关键词：无线传感器网络；操作系统；移植；TinyOS 2.0 系统；CC2430 平台

Transplantion of TinyOS 2.0 in CC2430

XU Jing-dong, LAI Xi-sheng

(College of Information and Technical Science, Nankai University, Tianjin 300071, China)

【Abstract】 The transplantion of operating system for the wireless sensor networks is to modify the hardware-irrelevant layer which makes it work on the different hardware platforms. On this issue, based on the analysis of the structure and hardware-abstract hierarchical on TinyOS 2.0, this paper expatiates how to implement the transplantion on the SoC-based and low-power CC2430 platform and tests the function of the modules. Test results show that the transplanted module works correctly, which lays a solid foundation for the future work.

【Key words】 Wireless Sensor Network(WSN); operating system; transplantion; TinyOS 2.0 system; CC2430 platform

DOI: 10.3969/j.issn.1000-3428.2011.02.090

1 概述

无线传感器网络(Wireless Sensor Network, WSN)是一种应用相关的网络。为满足不同环境需求，需要选择或设计特定的硬件节点和操作系统，而许多操作系统在设计之初支持的硬件节点类型又十分有限，因此需要对某些操作系统进行移植。本文选用的无线传感器节点的核心部件为 Chipcon 公司的低功耗低成本 CC2430 芯片，采用加州大学伯克利分校专门为无线传感器网络开发的嵌入式操作系统 TinyOS 作为软件平台，成功移植了 Radio、Timer、Flash、UART、GeneralIO 和 PIN 等 6 个底层模块。测试结果表明，移植的 6 个底层模块能够正常工作。

2 CC2430 芯片

CC2430 芯片沿用了以往 CC2420 芯片的架构，在单个芯片上整合了低功耗 8051 微控制器内核、高性能 2.4 GHz DSSS(直接序列扩频)射频收发器、128 KB 可编程闪存、8 KB 的 SRAM、1 个通用 16 位定时器、2 个 8 位定时器、1 个 MAC 定时器、具有 8 路输入 8 位~14 位的 ADC、2 个支持多种通信协议的 usART 和 21 个通用 I/O 引脚。工作电流损耗为 27 mA，在接收和发射模式下，电流损耗低于 27 mA 和 25 mA，能在超短时间内完成休眠模式到主动模式的转换，特别适合要求电池寿命极长的应用。该芯片不需要专利授权使用费，在很多应用环境下可以很容易地进行部署并完成任务。

3 TinyOS 操作系统及其可移植性

TinyOS 是 UC Berkeley 的 David Culler 领导的小组为 WSN 量身定制的一种嵌入式操作系统。TinyOS 的硬件抽象层通常是 3 级抽象结构，称作 HAA(Hardware Abstraction Architecture)[1]。

整个硬件抽象层分为硬件表示层(HPL)、硬件适应层(HAL)和硬件接口层(HIL)3 层。其中，直接访问软/硬件边界接口的是硬件表示层(Hardware Presentation Layer, HPL)，该层的主要任务是直接描述硬件提供的功能，即通过存储单元和 I/O 映射端口访问硬件和通过硬件中断来处理任务[2]。

在 HPL 上的是硬件适应层(Hardware Adaptation Layer, HAL)，该层是整个硬件抽象层的核心，它通过使用 HPL 层提供的原始接口构造更高级的抽象从而隐藏硬件资源使用的复杂性。

硬件抽象层的最高层是硬件接口层(Hardware Interface Layer, HIL)，该层把 HAL 层提供的接口转换成硬件独立的接口，隐藏了各个平台之间的差异，简化了应用程序的开发。

4 TinyOS 操作系统的移植

TinyOS 移植首先要考虑的问题是开发环境与工具链的选择和硬件抽象层的重写。由于 NesC 预处理器生成的代码是配合 gcc 生成的，但是 gcc 编译器并不支持 8051，因此需要寻找专门的 8051 编译器进行 TinyOS 的移植和开发。

4.1 开发环境

在 Windows 和 Linux 两大主要平台上有许多 8051 编译器，其中使用最广泛并且经常进行更新的有 2 种：KEIL 和 Small Device C Complier(SDCC)。由于 SDCC 是一个新兴的开源项目，因此在移植过程中经常会出现许多问题，使一些模块无法正常工作。而且在调试中，SDCDB 只是简单地驻留在 0 地址，当单步执行代码时也没有任何调试信息。相比于 SDCC，KEIL 提供了一套良好的开发调试环境，因此，最终选择使用 KEIL 开发工具进行移植工作。

基金项目：天津市信息化基金资助项目(082044012)

作者简介：徐敬东(1965－)，女，教授、博士生导师，主研方向：网络与信息安全，对等计算，移动计算，网络管理；赖锡盛，硕士研究生

收稿日期：2010-07-20 **E-mail：**xujd@nankai.edu.cn

4.2　工具链

图 1 是当前使用的整套工具链的概况，它基于 TinyOS 2.x、NesC 1.1.3、avr-gcc 3.4.3、PERL v5.8.6 和 KEIL C51 version 7.20H。移植在 Windows XP 下的 cygwin 中进行。

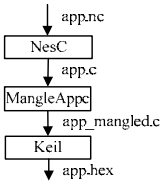


图 1　工具链

NesC 程序经过编译后生成 app.c 文件，它包含了运行 TinyOS 应用程序所需要的全部代码，但是该文件不能直接编译到 8051 平台运行，因此需要对输出文件 app.c 进行修改[3]。通过 mangling script 对 app.c 进行变换生成 app_mangled.c 作为 KEIL 编译器的输入，生成 app.hex 后写入芯片运行。

4.3　TinyOS 在 CC2430 上的移植

设置完开发环境和工具链之后，下一步就是进行 TinyOS 在 CC2430 上的移植。TinyOS 是基于不同抽象层组件的系统，它提供系统组件(例如初始化、调度等)、库组件(例如随机数生成器、队列等)和平台相关的组件。当移植到 CC2430 或者其他新的平台时，可以通过添加底层硬件的抽象使用已有的库组件和系统组件。图 2 是目前整个移植工作已完成的模块。

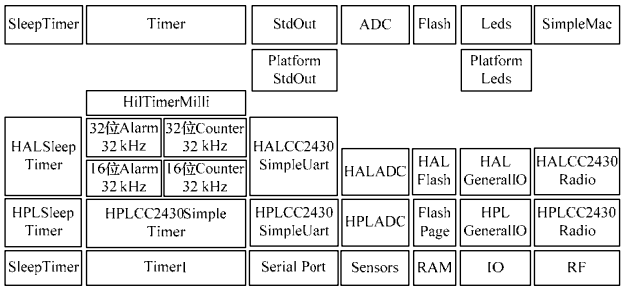


图 2　已移植模块

对各组件的描述如下：

(1)Radio。通过 SimpleMac 接口实现该组件。SimpleMac 接口实现了基于 802.15.4 标准的数据包的收发，而且能对收发器的各种参数进行设置。HPL 层的 HPLCC2430Radio 组件主要对无线收发器的寄存器进行操作，其实质是对 CC2430 的各个寄存器进行读写。该模块为整个硬件平台提供了 SPI 同步串行通信机制。在 HAL 层的 HALCC2430Radio 组件中对 HPL 层进行进一步抽象，实现对数据包的发送和接收、对无线模块状态的控制以及载波监听的功能。其中发送数据包前首先进行一定时间的清洁信道检测(Clear Channel Assessment, CCA)，如果信道忙则放弃此次发送，否则开始发送。目前 SimpleMac 接口不是按照标准 TinyOS 中的主动消息模式构建，只支持简单的数据包收发，不支持路由和重传等功能。

(2)Flash。通过 SimpleFlash 接口实现该组件。SimpleFlash 接口实现了读、写和删除 3 个操作。移植过程中使用的节点为 CC2430-F128(即 Flash 大小为 128 KB)，整个 Flash 被分成每页为 2 KB 的 64 个页面。写操作分为 DMA 和 CPU 2 种方式[4]。出于能耗考虑，移植中使用 DMA 方式进行写操作。擦除操作通过对 FCTL.ERASE 置 1，将所有字节设置为 1。HPL 层的 FlashPage 组件直接对底层寄存器操作以完成读写和删除，在 HALFlash 中提供 read、write 和 erase，对连续的存储地址进行读写删除操作。根据移植工作中的需求，目前没有按照标准 TinyOS 2.0 中的抽象进行实现(卷、日志、大型对象和小型对象)。

(3)Timer。通过通用的 TinyOS Timer 接口 Alarm 和 Counter 实现该组件。目前已完成的模块是 Timer1 和 SleepTimer。Timer1 是一个能够实现输入捕获和输出比较的独立的 16 位计时器，它包括 3 个捕获比较通道和 1 个 16 位的计数器。HPL 层 HPLCC2430Timer1AlarmCounterC 组件直接对计数器和比较通道相关的寄存器进行操作，为上层 Alarm 和 Counter 2 个接口提供开始、停止和设置定时周期等接口，在 HAL 层对其进行进一步封装。该组件将 Timer1 设置为输出比较模式，根据外部时钟和间隔时间产生一定时间间隔的方波，达到设置周期定时的目的。SleepTimer 包含一个工作在 32.768 kHz 时钟上的 24 位上增计数器和一个 24 位的输出比较通道。该定时器设置系统从进入到退出休眠模式(仅 PM0-2)时间，通过 3 个 SFR(ST0:ST1:ST2)进行输出比较，产生一定时间的间隔。其中，HPLSleepTimer 组件直接读写 SFR 和与 SleepTimer 相关的寄存器，而在 HALSleepTimer 中则向更高层提供设置时间间隔的接口 start。

(4)ADC。通过 Read 接口实现该组件。CC2430 的 AD 转换精度为 14 位，其转换方式为代尔塔-西格玛方式，有 8 路独立通道和 4 路差分通道，而且每个通道都具备 DMA 功能。Read 接口能够读取单个数值，当需要读取多个数值时，应用程序必须发起多值读取调用或者直接进行 DMA 传送。

(5)Pin。通过 GeneralIO 接口实现该组件。CC2430 总共有 21 个可编程的 I/O 引脚。通过设置一组 SFR 寄存器的位和字节，可使这些引脚作为通常的 I/O 口或者连接 ADC、计时器或 USART 部件的外围设备 I/O 口使用。其中，HPLCC2430GeneralIO 组件向上层提供各个引脚的接口，而 HALCC2430GeneralIO 提供调用 I/O 中断接口。在 LED 组件中则是调用底层操作 I/O 的接口来实现 LED 灯的开关操作。

(6)UART。通过 SerialByteComm 接口实现该组件。HPLCC2430SimpleUART 中主要是对串口的通信方式、波特率和停等位等串口物理特性进行设置，而在 HALSimpleUART 中实现 SerialByteComm 接口。SerialByteComm 提供 put 和 get 2 个操作，其中，put 操作将一个字节的数据写入串口，而 get 操作将一个字节的数据从串口读出[5]。通过该接口的抽象使 UART 接口中实现发送接收单个字节的功能。

4.4　移植工作的不足

上文详细说明了当前移植工作所完成的基本组件，但是目前的移植工作中还存在以下 3 个亟待解决的问题：

(1)主动消息层。目前在 Radio 组件中使用的是 SimpleMac 接口，实现了 IEEE 802.15.4 数据包的收发。但是该接口还不支持路由和数据的重传，在收发数据包时也没有实现 CSMA/CA 协议，并且不符合标准 TinyOS 的结构。

(2)精确计时器。Timer 组件在使用过程中没有达到一定的精确度，其原因是在市面上的 CC2430 芯片的初期产品中存在故障，使计时器会偶尔错过一些事件，该问题只有在厂商推出更加成熟的解决方案之后才能得以解决。

(3)芯片的自身特性。在已移植的组件中仍然有许多芯片自身的特性没有进行支持，比如随机数生成器等。

5　CC2430 平台 TinyOS 的测试

已移植模块的测试工作在无线龙 CC2430 节点上进行。表 1 给出了测试环境，包括节点数量、调用组件和测试软件。

(下转第 260 页)