

Linux下ARM和CC2530的串口通信

季峰

(武汉职业技术学院 电子信息工程学院, 湖北 武汉 430074)

摘要: 基于Linux下ARM Cortex A8和CC2530的串口通信技术, 可实现Cortex A8和CC2530间的串口通信。Cortex A8的串口通信是采用C语言编写, 并采用交叉编译的方法实现; CC2530的串口通信则是基于ZStack协议栈并使用IAR编译器开发的。该方法可为以Cortex A8为处理器、CC2530为网络节点的无线网络组建提供技术支持。

关键词: Linux; ARM; CC2530; 串口通信

中图分类号: TP39

文献标识码: A

文章编号: 2095-1302 (2016) 01-0034-03

0 引言

由于ARM嵌入式系统具有体积小、功耗低、噪音低、成本低、数据处理能力强等优点, 被广泛用作自动控制、智能终端等设备的硬件平台。通常ARM嵌入式平台是在操作系统的支持下工作, 常见的操作系统有Wince、uCOS、uCLinux、嵌入式Linux等, 其中, Linux操作系统具有稳定性高、源代码公开等优点, 而被广泛用于工业控制、机器人、物联网等领域。Linux-ARM嵌入式系统是Linux操作系统与ARM硬件平台的结合, 兼具了Linux操作系统和ARM嵌入式的优点, 是开发智能控制设备的常用组合。物联网是继计算机技术、互联网技术之后世界信息产业发展的新浪潮, 它所包含的无线传感网具有使用、安装便利的特点。基于IEEE 802.15.4协议的ZigBee技术是随着物联网的兴起而发展起来的一种无线传感网组网方案^[1], 基于该技术, Ti公司推出了系列网络处理器芯片, 其中CC2530是Ti公司推出的第二代网络处理芯片, 集成了RF收发器、增强型8051控制器、可编程存储器等模块, 具有更卓越的RF性能, 更小的封装尺寸和IR产生电路, 并支持多种协议如ZigBee PRO、ZigBee RF4CE等^[2], 由其组成的无线传感网可用于居家、消费类电子产品、工业、医疗等领域。虽然CC2530所采用的CPU技术成熟, 有一定的计算、控制等能力, 但在控制通信较复杂的场合下不能满足需求, 因而需要性能更强的处理器作为上位机, 如Linux-ARM嵌入式系统。由此可见, 采用Linux-ARM为上位机、CC2530为下位机是一种优秀的智能无线网络组网方案。在由Linux-ARM和CC2530组成的网络系统中, Linux-ARM和CC2530间的通信是该组网方案必须解决的一项技术问题。Linux-ARM嵌入式系统^[3]和CC2530均拥有串口通信方式, 这可成为两者传递信息的通道。基于以上分析, 本文设计了Linux系统

下ARM Cortex A8的串口通信; 使用ZStack协议栈开发了CC2530的串口通信, 实现了Cortex A8和CC2530间的通信。

1 Linux-ARM 串口通信

S5PV210AH是三星基于ARM Cortex A8内核生产的一款处理器, 主频达1 GHz, 被广泛应用于数据采集、图像处理、智能终端、自控设备、物联网网关等方面。在Linux-ARM系统中, Cortex A8的串口通信是在Linux操作系统的控制下完成的, 其程序流程图如图1所示。

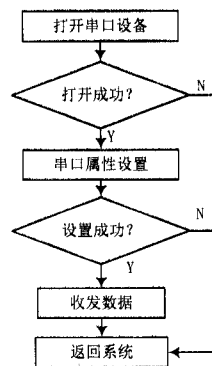


图1 Linux下ARM Cortex A8串口通信流程图

为在Linux-ARM系统中进行串口通信, 首先需打开串口设备, 而Linux系统对设备的管理是通过文件实现的, 对串口的操作等同于对文件的操作。Linux系统中的设备管理文件存放在/dev目录中, 串口的文件名一般为ttyS*, 其中*表示不同的串口, 是从0开始的整数。串口可通过open()函数打开: fd=open("/dev/ttyS0", O_RDWR), 语句中的参数ttyS0表示串口1, O_RDWR表示可对串口进行读写操作, fd是open()函数的返回值, 称为文件描述符, 在随后的操作中, Linux系统的内核可使用该文件描述符来访问文件。如果fd=-1则表示打开失败, 程序通过return返回系统, 若打开成功则进行串口参数设置。串口参数设置过程如图2所示, 首先是获取串口参数, 而后根据需要进行参数配置, 最后对串口进行参数

设置, 该过程主要涉及到的有 `tcgetattr()` 函数、`termios` 结构体和 `tcsetattr()` 函数。

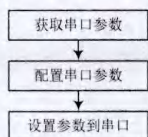


图2 Linux-ARM的串口设置过程

在串口参数配置的过程中, `termios` 结构体是完成配置的关键, 该结构体包含在 `termios.h` 头文件中, 其内容如下:

```

struct termios
{
    unsigned short c_iflag; 输入模式标志
    unsigned short c_oflag; 输出模式标志
    unsigned short c_cflag; 控制模式标志
    unsigned short c_lflag; 本地模式标志
    unsigned short c_line; 线路规则
    unsigned short c_cc[NCC]; 控制字
}
  
```

串口的详细配置步骤如下:

(1) 定义一个 `termios` 结构体变量 `opt`, 使用 `tcgetattr()` 函数把串口参数提取到 `opt` 中, 相应的语句是 `tcgetattr(fd, &opt)`。为避免打开串口后写入缓存数据的干扰, 可使用 `tcflush(fd, TCIOFLUSH)` 语句清除缓存中的数据。

(2) 串口参数的基本配置包括波特率、校验位、停止位、流控等。在配置程序中可使用 `cfsetispeed(&opt, B115200)` 和 `cfsetospeed(&opt, B115200)` 分别把串口通信的输入输出波特率设为 115 200 b/s; 而数据位、奇偶位、停止位则是通过 `c_cflag` 参数的设置实现: 可使用 `opt.c_cflag &= ~CSIZE` 和 `opt.c_cflag |= CS8` 把数据位设置为 8 位; 使用 `opt.c_cflag &= ~PARENB` 和 `opt.c_iflag &= ~INPCK` 让串口通信时不采用奇偶校验; 使用 `opt.c_cflag &= CSTOPB` 使串口有 1 位停止位。

(3) 串口参数设置。该过程是把 `opt` 结构体中的参数设置到串口中, 可调用 `tcsetattr(fd, TCSANOW, &opt)` 函数, 其中的参数 `TCSANOW` 表示不等数据传输完毕就立即改变串口属性。如果参数设置成功, `tcsetattr()` 函数的返回值为 0, 据此值可做以下判断: 只有返回值为 0 时才进入串口通信的下一步, 若返回值不为 0 则返回到操作系统。

串口参数设置完成后, 串口即可进入数据收发状态。数据的接收可调用 `read(fd, RD_buff, RD_buff_size)` 函数进行, 该函数表示从 `fd` 所指示的文件中读取 `RD_buff_size` 个字节到 `RD_buff` 缓冲区中。数据的发送可使用 `write(fd, WR_buff, WR_buff_size)` 函数, 是从 `WR_buff` 所指的缓冲区输出 `WR_buff` 个字节到 `fd` 所指示的文件中。数据收发任务完成后, 使用 `close(fd)` 函数关闭串口。

通信中的 Cortex A8 是由无线龙科技有限公司提供的, 程序的开发是在 Linux 环境中进行的, 并使用 `arm-Linux-gcc` 进行交叉编译^[4], 在 XP 操作系统中通过 PC 机的超级终端下载到 Cortex A8 中。

2 CC2530 串口通信

CC2530 的开发是在 IAR 集成环境中使用 ZStack 2.5.1 协议栈进行的, ZStack 的目录结构如图 3 所示。

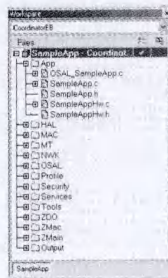


图3 ZStack 协议栈的目录结构

ZStack 是从 ZMain 层中的 `main()` 函数开始执行的, `main()` 函数含有系统初始化函数 `osal_init_system()`, 该函数可完成新任务的登记。在 `osal_init_system()` 中找到 `osalInitTasks()`, 并在该函数的最后添加新任务的初始化函数 `SampleApp_Init(taskID)`, `SampleApp_Init()` 的函数体定义在 `SampleApp.c` 文件中, 可在 `SampleApp_Init()` 函数中添加串口初始化函数调用语句和串口任务登记函数调用语句。协议栈的工作机制是一种轮询方式, 当系统进入运行函数 `osal_start_system()` 后, 在 `do while` 循环体的作用下开始轮询。当协议栈的操作系统抽象层 (OSAL) 在查看 `taskEvents` 的过程中发现有事件发生, 会记下该事件的 `idx` 值, 而后通过 `events = (tasksArr[idx])(idx, events)` 语句调用登记在 `tasksArr[]` 数组中的函数指针, 随后进入到任务处理函数。由此可见在添加新任务时, 需在 `tasksArr[]` 数组的最后添加新任务的函数指针, 并在 `SampleApp.c` 文件中修改相应的函数体。

本文采用的 CC2530 是由网蜂科技提供的 ZigBee 开发套件, 在串口通信开发的过程中, CC2530 被定义为协调器, 其开发步骤分为如下三步:

(1) 串口初始化。找到应用层 (App) 中的操作系统接口文件 `OSAL_SampleApp.c`, 并在该文件中找到 `osalInitTasks()` 的函数体, `osalInitTasks()` 中的最后一个函数 `SampleApp_Init(taskID)` 被作为串口通信任务。在 `SampleApp.c` 中找到 `SampleApp_Init()` 的函数体, 并添加函数调用语句 `MT_Uartinit()`, 如下所示:

```

void SampleApp_Init (uint8 task_id)
{
    SampleApp_TaskID = task_id;
    SampleApp_NwkState = DEV_INIT;
}
  
```



```
SampleApp_TransID = 0;
MT_Uartinit ();
MT_UartRegisterTaskID (task_id);
.....
}
```

MT_UartInit() 函数位于 MT 目录下 MT_UART.c 文件中, 是串口初始化函数, 可完成 CC2530 的串口参数设置。根据 uartConfig.baudRate = MT_UART_DEFAULT_BAUDRATE 语句找到 MT_UART_DEFAULT_BAUDRATE 的位置, 并修改为 #define MT_UART_DEFAULT_BAUDRATE HAL_UART_BR_115200, 使串口的波特率为 115 200 b/s; 根据语句 uartConfig.flowControl = MT_UART_DEFAULT_OVERFLOW 查找到 MT_UART_DEFAULT_OVERFLOW, 并将其修改为 #define MT_UART_DEFAULT_OVERFLOW FALSE, 修改好之后关闭流控。

(2) 登记任务号。任务号的登记是通过 MT_UartRegisterTaskID() 函数完成的, 该函数的调用语句可直接跟在 MT_Uartinit() 函数调用语句之后, 其功能是把串口事件通过 task_id 登记在 SampleApp_Init() 中。

(3) 串口通信。在 SampleApp.c 文件中找到 SampleApp_ProcessEvent() 函数, 添加接收或发送语句, 发送使用 HalUARTWrite() 函数, 接收使用 HalUARTRead() 函数, 这两个函数位于 MT_UART.h 库中。

为验证 CC2530 串口通信的效果, 使用串口调试助手对 CC2530 的串口通信进行了验证, 其结果如图 4 所示。在测试

作者简介: 季峰 (1975—), 男, 河南荥阳人, 博士, 副教授。主要研究方向为微电子器件、电子电路设计、高职教育教学。

(上接第 33 页)

统的示意图。其目的是实现能源介质数据采集、能源设备状态监控、在线运行管理、优化节能调度和基础能源管理等功能。

工业网关整合通讯接口服务器、工控机、工控软件的功能于一体。系统通过工业网关采集必要的底层系统的数据, 并将这些数据自动、完整、实时地推送到实时/历史数据库, 达到能源管理系统的综合监控和管理要求。

4 结语

针对工业现场设备复杂繁多、多种现场总线长期共存而导致企业两化建设与维护成本居高不下的现状, 提出基于工业互联网网关的 ICT 中间平台通信标准化方案, 既可避免已有现场设备的更新换代, 又能实现低成本的现场设备互联互通。并基于工业互联网的应用提出工业通信网关需要具备的六项重要特性。为工业互联网多样化的设备之间互联互通提

作者简介: 李立群, 男, 硕士研究生在读, 同济大学机械与能源工程学院。

刘 林, 男, 就职于上海旋思科技有限公司。

宋蕴璞, 女, 工学博士, 副教授, 硕士生导师。研究方向为机械系统信息化集成处理领域的数学建模, 嵌入式系统开发等。

过程中使用的语句是 HalUARTWrite (0, "Hello, this is serial data from CC2530.\n", 40), 可以看出串口调试助手完整地接收到了来自 CC2530 的信息。



图 4 CC2530 串口通信测试

3 结语

对 Linux ARM Cortex A8 和 CC2530 的串口通信进行了研究, 分别采用交叉编译和 ZStack 协议栈开发了 Cortex A8 和 CC2530 间的串口通信程序, 该方法为 Cortex A8 上位机与 CC2530 协调器间的通信提供了解决方案。

参考文献

- [1] 季峰, 黄艳华, 虞沧. 高职物联网通信方向技能体系及培训方案研究 [J]. 黄河水利职业技术学院学报, 2015, 27 (1): 59-61, 74.
- [2] 章伟聪, 俞新武, 李忠成. 基于 CC2530 及 ZigBee 协议栈设计无线网络传感器节点 [J]. 计算机系统应用, 2011, 20 (7): 184-187, 120.
- [3] 周峰华, 潘文亮, 刘军亮, 等. Linux 下 ARM 和单片机的串口通信设计 [J]. 仪表技术, 2011 (8): 27-28, 32.
- [4] 季峰. 基于 Socket 的物联网网关网络通信编程与实现 [J]. 信息通信, 2014 (10): 81-82.

供一种标准、易行的方案。

参考文献

- [1] 姜黎萍. 宝钢股份罗泾区能源管理系统 [J]. 冶金自动化, 2012, 36 (2): 7-11.
- [2] 罗安明, 韩新民. 现场总线技术的发展和趋势 [J]. 机电产品开发与创新, 2011, 24 (1): 9-11.
- [3] 闵杰. 标准是工业互联网竞争制高点 [N]. 中国电子报, 2015-04-14 (001).
- [4] 桂健. 兼容多种通信方式的智能网关设计与实现 [D]. 福建: 福建师范大学, 2014.
- [5] 柴天佑, 金以慧, 任德祥, 等. 基于三层结构的流程工业现代集成制造系统控制工程 [J]. 2002, 9 (3): 1-6.
- [6] 杨秋昊. 异构网实时通信的研究与应用 [D]. 南京: 南京理工大学, 1998.
- [7] 宋慧欣. 破解“工业控制系统信息安全”迷局 [J]. 自动化博览, 2012 (7): 30-35.