

基于HCI协议层的蓝牙技术应用开发

马兴, 梁剑, 陈前斌

(重庆邮电学院, 重庆 400065)

摘要: 蓝牙是一种短距离无线通信技术, 目的是取代终端的连接电缆, 因此从蓝牙协议栈中的HCI层入手, 对该层的主要功能模块和工作机制进行了分析, 并阐述了基于HCI层的蓝牙应用开发方案, 最后给出了一个蓝牙应用开发的设计实例——蓝牙二维条码扫描仪系统, 该系统证实了基于HCI层进行蓝牙技术应用开发的可行性。

关键词: HCI; 蓝牙技术; 开发方案

中图分类号: TN914.4 **文献标识码:** A

The Development of Bluetooth Application Based on HCI Layer

MA Xing, LIANG Jian, CHEN Qian-bin

(College of Telecommunications Information Engineering, CUPT, Chongqing 400065, P. R. China)

Abstract: Bluetooth is a wireless technology for short distance communication to replace cable. In this paper, the main function modules of Bluetooth HCI layer are illustrated and the development schemes of bluetooth application based on HCI are discussed. A design example is also presented — bluetooth two-dimension scanner system. In this system, the bluetooth application based on HCI layer proves to be feasible.

Key words: HCI; bluetooth technology; development scheme

0 引言

蓝牙是一种短距离的无线通信方式, 其目的是取代设备之间的电缆连接。蓝牙设备运行在 2.4 GHz ISM (Industrial Scientific Medical) 频段, 基带 (BaseBand) 协议支持电路交换与包交换方式, 使用跳频扩谱 (FHSS) 技术进行数据传输。至多 8 个活动的蓝牙设备可以共享信道带宽, 形成所谓的匹克网 (piconet)。每一个 piconet 中, 只有 1 个蓝牙主设备, 其余 7 个为从设备。另外, 可容许有更多的从设备以暂停 (park) 的方式锁定到主设备。多个 piconet 可以相互重叠, 形成散射网 (scatternet)。从 HCI 层出发,

可以将蓝牙协议分为硬件与软件 2 个协议块。硬件部分由射频 (RF)、基带 (BB)、链路管理器 (LM)、HCI 低层嵌入式驱动程序构成, 硬件部分也可称为蓝牙子系统。软件部分由 HCI 高层软件驱动程序、逻辑链路控制适配协议 (L2CAP)、串口仿真协议 (RFCOMM)、业务发现协议 (SDP)、电话控制协议 (TCS) 构成。蓝牙协议模型见图 1。

1 HCI 的剖析

如何基于 HCI 进行蓝牙技术应用开发, 首先必须了解 HCI 涉及到的相关内容, 我们可以将其划分为 HCI 软件与 HCI 硬件 2 部分讨论。

· 收稿日期: 2002-03-29

· 作者简介: 马兴 (1975-), 男, 山西大同人, 重庆邮电学院 2000 级硕士研究生, 主要研究方向是通信与信息系统。

· 78 ·

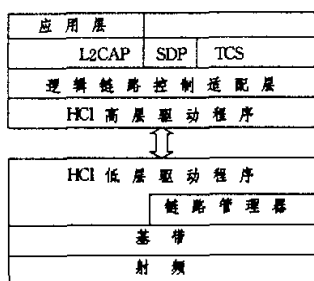


图1 蓝牙协议模型
Fig.1 Bluetooth protocol model

1.1 HCI 软件

HCI 软件结构可以分为两大平面结构:数据平面与控制平面。数据平面负责数据通过链路的转移,控制平面负责链路的管理与控制。

从控制平面来看,主机端的HCI驱动程序(HCI高层驱动程序)与蓝牙子系统的嵌入式HCI驱动程序,是不完全对称的。主机端的HCI驱动程序,一方面通过接口被蓝牙应用程序调用(本文所提到的蓝牙应用程序是相对于HCI层而言的,这种应用包括HCI层以上的蓝牙协议软件,或是直接基于HCI层的客户应用程序),实现对上层应用的承载;另一方面实现了协议中的HCI功能集,使主机可以向蓝牙子系统发送HCI指令,或接收子系统返回的HCI事件。蓝牙子系统的嵌入式HCI驱动程序完成的也是两个功能:一是实现了协议中的HCI指令功能集,从而可以接收主机端HCI驱动程序发送的指令,或是向主机返回相应的HCI事件;二是对基带和链路管理器进行控制,实现物理链路的建立,通过HCI事件将相关结果返回给主机端HCI驱动程序。

从数据平面来看,HCI驱动程序的实现功能和方式与控制平面极其相似,只不过主机端发送的是数据,并非HCI指令,而且数据的传送必须在链路建立之后才能进行。图2为蓝牙HCI软件结构。

1.2 HCI 的硬件

图2中的传输驱动程序实际上就是对HCI传输硬件设备的驱动。在蓝牙规范的描述中,蓝牙子系统与主机之间共有3类数据转移的方式,每一种数据在硬件接口均具有一定的封装格式,分别是由3种传输层支持,目前的3个HCI传输层分别为UART、RS232和USB。

RS232与UART传输层都是利用UART进行

串行通信,但是二者应用环境有所不同。当蓝牙子系统与主机位于同一印刷电路板时,可以使用UART传输层,这时传输信道的误码率可以达到通信质量的要求。虽然UART并未规范电气特征,但是在信号电平兼容的情况下,可以实现主机与蓝牙子系统的无缝接口,简化了硬件电路。如果信号电平不能兼容,可以在蓝牙子系统与主机之间加入适当的电平转换电路,从而实现设备间的正常通信。在UART传输方式下,主机与子系统之间不使用波特率协商机制。

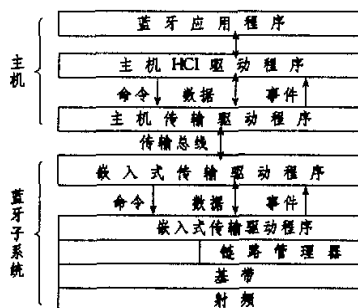


图2 蓝牙HCI软件结构
Fig.2 Bluetooth HCI software architecture

如果蓝牙子系统与主机位于不同的印刷电路板时,使用RS232传输方式。RS232规定了详尽的电气特征和链路协议,这有利于RS232信号在远距离传输情况下,可以处理高的线路误码率。如果使用RS232传输方式,主机与子系统之间需要完成2个功能:一个是协商波特率、奇偶校验、停止位比特数和协议模式;另一个是向发送端通知线路错误。

与RS232、UART不一样的是,USB提供了一种简单的传输方式,不需要额外的开销来标识任务类型,这种传输方式主要是描述蓝牙数据如何映射到USB端点。

2 基于HCI的蓝牙技术应用开发

本文所提到的主机是一个广义的概念,可以是MPU为核心的计算机,也可以是MCU为核心的控制电路,二者用途各异。MCU为核心的控制电路适用于嵌入式应用开发(即蓝牙终端产品,如蓝牙耳机),MPU为核心的计算机适合于人机接口软件开发(如用户通过蓝牙设备接入局域网、浏览网页)。

蓝牙技术应用开发可以从不同的协议层入手。

如果直接从蓝牙子系统做起,由于涉及到射频、基带等硬件协议层,开发周期必然加长。而对于一个企业,如何缩短产品的上市时间是至关重要的,基于HCI协议层的蓝牙技术应用开发能够有效地缩短开发周期。目前,一些公司相继推出了各种蓝牙模块,这些模块实现了蓝牙子系统的全部或部分功能,屏蔽了射频、基带两个硬件协议层,可以直接在HCI基础上进行开发。表1是一些公司推出的蓝牙模块。

为了后面讨论的方便,我们将这些蓝牙模块分成2类。第1类为只实现了链路控制功能的模块,第2类为链路控制与嵌入式HCI驱动功能均实现的模块。这2类模块的主要不同之处是,第2类包括一个RISC(精简指令集计算机)处理器,实现链路控制和HCI功能,而第一类的RISC则不执行HCI功能。选用哪一类蓝牙模块,要根据实际情况而定。

表1 部分公司推出的蓝牙模块

Tab.1 Bluetooth modules developed by some companies				
公司名称	产品号	集成的功能		支持的HCI传输层
		链路控制	HCI驱动程序	
Ericsson	ROK 101 007	是	是	USB UART
TI	BSN6030	是	是	UART
Lucent	W7400	是	是	USB UART
Alcatel	TBD	是	是	UART
National Semiconductor	LMX5001	是	否	没有
Silicon Wave	SiW1601	是	否	没有

2.1 开发方案1

如果我们选用集成了链路控制与HCI驱动功能的蓝牙模块,那么就可以在一个比较完整的平台上进行开发。这些蓝牙模块不仅提供1个RISC处理器或者多个HCI传输层接口,而且提供了HCI与链路管理器嵌入式软件,使用这种类型的模块,硬件上的工作就是进行蓝牙模块与主机之间的传输接口连接。如果主机具有USB或UART接口,在蓝牙模块与主机信号电平兼容的情况下,接口连接可以不需要其它辅助硬件电路。许多情况下,USB或UART接口集成在主机的核心芯片上,所以可以实现蓝牙模块与主机的无缝接口。

这种模型下的蓝牙技术应用开发,主要工作集中在蓝牙主机端软件实现上,包括HCI驱动程序与传输驱动程序。正是由于这种蓝牙模块实现了嵌入式HCI协议,所以应用开发可以直接从HCI层开始,避免了基带与射频的硬件实现,是一种全软件式的

开发模式,开发成果具有较好的可移植性。通过封装HCI协议层,可以生成标准的HCI接口函数,为上层应用程序的开发提供一个完整的平台。图3为开发方案。

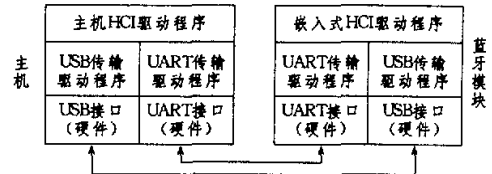


图3 集成链路控制功能与HCI协议的蓝牙模块开发方案
Fig.3 Development scheme for bluetooth modules integrated link control function and HCI protocol

2.2 开发方案2

只集成链路控制功能的蓝牙模块,同样实现了射频与基带的功能,但是,不提供HCI传输层接口和嵌入式HCI驱动程序。由于此类模块的接口一般不具有通用性,所以,利用这种模块进行开发,必须设计接口功能转换电路。

这种接口功能转换电路是相当灵活的,可以转换为HCI协议层定义的USB、UART或RS232接口,也可以转换为PCI等其它I/O总线格式。由于在接口转换方面具有高度的灵活性,因此,可以简化蓝牙模块与主机端的硬件设计。这时,我们可以利用FPGA的现场可编程特性,完成主机与蓝牙模块的接口功能转换,不需要两端具有对称的USB或UART接口和驱动程序。虽然,这种开发模型省掉了HCI协议层中的传输部分,但是并不违背蓝牙的协议模型。因为这种HCI传输功能,本身就是蓝牙协议实现的一种过渡,是以蓝牙子系统与主机具有对等的传输驱动程序为基础的,并不对蓝牙数据帧格式产生影响。图4为硬件开发方案。

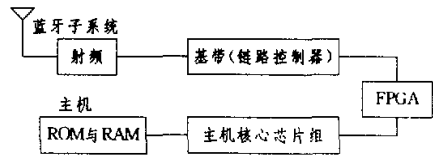


图4 只集成链路控制功能的蓝牙模块硬件开发方案
Fig.4 Hardware development scheme for bluetooth modules with link control function

由于这种蓝牙模块不包含嵌入式HCI驱动程序,所以嵌入式HCI驱动程序的功能将转嫁给主机实现。这样,与前一种开发模式相比,主机的HCI层驱动程序要复杂一些,因为需要增加嵌入式HCI驱

动程序所执行的一些功能。此类蓝牙模块的RISC不执行HCI功能,因此,模块中的内存要求有所降低。图5为软件开发方案。

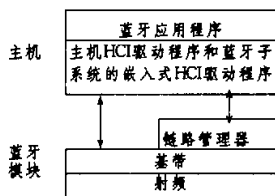


图5 只集成链路控制功能的蓝牙模块软件开发方案
Fig.5 Software development scheme for bluetooth modules with link control function

2.3 2种开发方案的简单比较

方案1需要在主机端进行软件编程,实现HCI高层驱动程序与传输驱动程序的功能;硬件部分视情况而定,如果主机与蓝牙模块信号电平匹配,可以实现接口间的无缝连接,如果信号电平不匹配,只需增加接口电平转换电路(并不是实现接口功能转换的电路)。这种方案对硬件要求不高,由于主机与蓝牙模块之间的通信有传输驱动程序参与,所以速度较慢。

方案2需要在主机端实现HCI高层驱动程序与HCI低层驱动程序的功能,硬件部分要进行接口功能的转换实现。所以,方案2对硬件要求较高,由于完全使用硬件进行主机与蓝牙模块通信,所以速度较快。

3 一个蓝牙技术应用开发实例

笔者基于HCI成功开发了一个蓝牙应用产品——蓝牙二维条码扫描仪系统,并参加了中国重庆第三届高交会。整个系统由2部分组成:蓝牙二维条码扫描仪与条码接收端。本项目的蓝牙模块采用Ericsson公司的,型号为ROK 101 007,工作电压为3.3V。此模块实现了蓝牙的链路控制与HCI功能,根据本文第3部分的开发方案介绍,宜采用方案1进行开发。另外,该模块还提供了UART与USB接口,所以将蓝牙二维条码扫描仪的控制电路与蓝牙模块集成在同一块印刷电路板上也是非常方便的。下面从硬件与软件两个方面加以介绍。

3.1 硬件设计

蓝牙二维条码扫描仪硬件由3部分组成:扫描

器、控制电路(即主机)、蓝牙模块,开发的重点是控制电路。在硬件实现上,控制电路主要由MCU、外部存储单元RAM、电源系统、振荡电路、复位电路、接口电平转换电路构成,其中MCU与RAM是整个控制电路的核心。MCU采用MCS51系列通用的单片机,信号电平属于TTL 5V。由于扫描仪接口信号电平为RS232标准,蓝牙模块UART接口信号电平为3.3V,所以MCU与扫描仪、蓝牙模块不能实现无缝接口,必须通过接口电平转换电路进行连接。图6为蓝牙二维条码扫描仪的硬件结构。

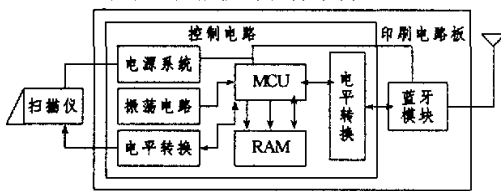


图6 蓝牙二维条码扫描仪硬件结构

Fig.6 Hardware architecture of two-dimension bluetooth scanner

条码接收端硬件比较简单,主机为微型计算机,所以不需对主机电路进行设计。蓝牙模块与微机采用串行方式进行通信,微机选用RS232接口,蓝牙模块利用UART接口。虽然通信双方的接口方式不一样,但是我们只使用了RS232的数据收发线(RXD、TXD),对其它控制与状态线并未使用(这些控制线的功能由软件实现),所以,通过接口电平转换电路对微机与蓝牙模块进行连接是可行的,这就避免了复杂的接口功能转换设计。硬件结构见图7。



图7 条码接收端硬件结构

Fig.7 Hardware architecture of code receiver

3.2 软件设计

3.2.1 蓝牙二维条码扫描仪嵌入式软件设计

蓝牙二维条码扫描仪实现piconet中主设备的功能,即主动与条码接收端进行连接,这个功能是通过汇编语言进行嵌入式开发实现的。包括特殊功能寄存器设置、蓝牙链路建立子程序、条码数据接收子程序(接收条码扫描仪由接口送入的数据)、条码数据发送子程序4部分软件功能块。

在程序开发中,首先实现对特殊功能寄存器(SFR)的设置,完成通信的基本要求;与扫描仪接口的波特率(9.6 kbit/s)、与蓝牙模块接口的波特率

(57.6 kbit/s)、定时器初值、中断允许与禁止等,这样才能使MCU进行有序地工作。

实现蓝牙链路的建立,首先必须对蓝牙模块进行初始化(在条码接收端软件设计中有介绍)。初始化完成后,调用HCI-inquiry子程序查询条码接收端蓝牙模块的地址(48 bit)之后,利用查询到的地址,并调用HCI-connect子程序实现蓝牙链路的建立。

链路建立成功后,在定时器与串行中断的控制下,可以进行条码数据的接收、存储、转发,这3步是通过外部RAM的双指针循环结构方式实现的。

3.2.2 条码接收端软件设计

条码接收端程序可分为计算机串口的初始化、对蓝牙模块的操作、接收二维条码数据等的功能块程序。计算机串口的初始化用以启动串口;蓝牙模块的操作的目的是要在客户和服务器之间建立起无线链路;二维条码数据的接收就是接收通过蓝牙链路来传送的数据。

第一步要做的工作就是初始化计算机的串口,因为应用程序和蓝牙模块之间的通信(HCI指令的发送、HCI事件的接收、数据的传输)都是通过计算机的串口完成的。串口初始化是由HCI-Initial-com(char * lpszcom-name, UInt32 baudrate)函数完成,其中参数lpszcom-name表示串口名,如"com1"或"com2";参数baudrate为设置的串口波特率,如57600。串口启动之后就可以对蓝牙设备操作了。

条码接收端的蓝牙模块被设置为从设备,由蓝牙二维条码扫描仪对其发起连接,要完成链路的建立,必须首先对条码接收端的蓝牙模块进行初始化。

初始化操作只需要调用相关的HCI指令函数即可完成。这些函数包括:复位HCI-Reset()、设置扫描HCI-Write-Scan-Enable(3)、设置加密认证HCI-Write-Authentication-Enable(0)、设置事件过滤器HCI-Set-Event-Filter()、设置接收连接超时HCI-

Write-Connection-Accept-Timeout(0x2000)、设置呼叫超时HCI-Write-Page-Timeout(0x3000),上述函数中的数值是供本项目开发特定使用的。这些函数实现的功能在HCI规范中有详细的描述。

蓝牙模块初始化完成后,程序启动一个从线程开始等待扫描仪的连接,因前面在设置事件过滤器的时候已经将其设置为能自动接收连接请求,所以整个连接过程不需要用户的干预而自动完成。从线程的创建是通过调用CreateThread(NULL, 0, (LPTHREAD-START-ROUTINE) Receive, this, 0, &.dwThread)完成的,其中Receive是线程函数, this是对话框指针。在这一步完成以后,扫描仪和条码接收端之间的通信链路就建立起来了,扫描仪的数据就可通过蓝牙链路传递到条码接收端。

4 结束语

蓝牙产品的开发重在有特色的应用,将蓝牙技术与现实生活中的各种终端产品相结合,如条码扫描器、耳机、电话等,从而使人们摆脱各种线缆的束缚。目前蓝牙的开发,特别是与终端产品相结合的嵌入式开发,正在短距离无线通信中显示出一股强劲的气势,蓝牙技术的大展期必将很快地到来。

参 考 文 献

- [1] Core, Version 1.0B, Bluetooth SIG, 1999. Specification of the Bluetooth System[S].
- [2] Bluetooth EBSK[Z]. Ericsson, 2001.
- [3] NATHAN Muller J. Bluetooth Demystified [M]. 北京:人民邮电出版社, 2001.
- [4] 朱明程. Xilinx 数字系统现场集成技术[M]. 江苏南京:东南大学出版社, 2001.
- [5] 金纯. 蓝牙技术[M]. 北京:电子工业出版社, 2001. (编辑:龙能芬)

欢迎广大读者订阅《重庆邮电学院学报》(自然科学版)

邮发代号:78-77