

转载

believe209

2015-10-17 16:14:14

15690

收藏 15

分类专栏：

blueZ


 文章标签：

bluetooth

blueZ

hci

sdp

 **blueZ** 专栏收录该内容

1 订阅

3 篇文章

订阅专栏

hci_inquiry() //查询周围蓝牙设备并获取其地址
sdp_connect() //链接到蓝牙设备的sdp服务器
sdp_list_append() //添加sdp查询列表项
sdp_service_search_attr_req() //带服务属性的服务查询请求，查询蓝牙设备，有那些服务及每个服务的属性
sdp_uuid16_create() //为sdp查询创建通用唯一标识(UUID)
rfcomm_read_config() //创建串口练级，链接好后，在/dev/bluetooth/rfcomm/目录下会出现以程序给定的本地信道号命名的串口设备名/dev/bluetooth/rfcomm/0

ioctl() //
open() //
termios() //
write() //
read() //
ioctl() //

```
//main.c line; 136
static int create_dev(int ctl, int dev, uint32_t flags, bdaddr_t *bdaddr, int argc, char **argv)
{undefined
    struct rfcomm_dev_req req;
    int err;

    memset(&req, 0, sizeof(req));
    req.dev_id = dev;
    req.flags = flags;
    bacpy(&req.src, bdaddr);

    if (argc < 2) {undefined
        if ((err = rfcomm_read_config(rfcomm_config_file)) < 0) {undefined
            perror("Can't open RFCOMM config file");
            return err;
        }

        bacpy(&req.dst, &rfcomm_opts[dev].bdaddr);
        req.channel = rfcomm_opts[dev].channel;

        if (bacmp(&req.dst, BDADDR_ANY) == 0) {undefined
            fprintf(stderr, "Can't find a config entry for rfcomm%d\n", dev);
            return -EFAULT;
        }
    } else {undefined
        str2ba(argv[1], &req.dst);
```

```
        if (argc > 2)
            req.channel = atoi(argv[2]);
        else
            req.channel = 1;
    }

    if ((err = ioctl(ctl, RFCOMMCREATEDEV, &req)) < 0 )
        perror("Can't create device");

    return err;
}
```

```
//parser.c line: 1705
int rfcomm_read_config(char *filename)
{undefined
    extern FILE *yyin;
    char file[MAXPATHLEN + 1];
    int i;

    for (i = 0; i < RFCOMM_MAX_DEV; i++) {undefined
        rfcomm_opts[i].bind = 0;
        bacpy(&rfcomm_opts[i].bdaddr, BDADDR_ANY);
        rfcomm_opts[i].channel = 1;
    }

    if (filename) {undefined
        snprintf(file, MAXPATHLEN, "%s", filename);
    } else {undefined
        snprintf(file, MAXPATHLEN, "%s/.bluetooth/rfcomm.conf", getenv("HOME"));



















        if ((getuid() == 0) || (access(file, R_OK) < 0))
            snprintf(file, MAXPATHLEN, "%s/rfcomm.conf", CONFIGDIR);
    }

    if (!(yyin = fopen(file, "r")))
        return -1;

    lineno = 1;
    yyparse();

    fclose(yyin);

    return 0;
}
```

	线程	3篇
	进程	1篇
	uboot	3篇
	C#	69篇
	fedora	2篇
	windows	17篇
	VNC	13篇
	GPS	
	GPRS	
	centos	1篇
	linux命令	6篇
	IOS	12篇
	bluetooth	21篇
	blueZ	3篇
	Vxworks	
	工具	1篇
	BlueDroid	4篇
	SQL Server	21篇

#####

http://blog.chinaunix.net/u2/83623/showart_1348867.html

原文出处:

<http://people.csail.mit.edu/albert/bluez-intro/c401.html>

第四章 基于BlueZ的C语言蓝牙编程

有很多理由促使我们选用C替代其他高级语言来例如Python来开发蓝牙应用程序。Python环境可能并不适合于嵌入式系统。因为嵌入式系统对程序的大 小，运行速度，和占用的存储空间有严格的限制，这些都使得像Python之类的解释性语言无法在嵌入式系统上应用。程序员需要对本地的蓝牙适配器进行更好 的控制，或者需要建立一套动态链接库以便于其他应用程序的链接以取代单一的应用程序。就像上述描述的这些，BlueZ是一款强大的蓝牙通信协议栈，它扩展 的API使得用户方便操纵大量的蓝牙资源。但是BlueZ没有官方的描述文档，甚至非官方的文档也寥寥无几。初学者在BlueZ的官方邮件列表上请求相关 的文档，通常的得到的回复是被告知请通过仔细阅读源代码来了解API的功能。阅读BlueZ的源代码对于初学者来说是一项相当费时的工作，在短期内取得的进展是相当有限的，很可能成为很多蓝牙编程初学者的拦路虎。

本章简要叙述了基于BlueZ的C语言蓝牙编程的方法。本章为C程序员进一步阐述了第二章中涉及的知识。

4.1 选择一个通信的对象

Example 4-1是一个查找周边蓝牙设备的简单应用程序。程序首先获取系统的蓝牙设备号，扫描周边的蓝牙设备，然后查找每一个被搜索到的蓝牙设备的名称。后边有对数据结构和函数的详细描述。

Example 4-1. simplescan.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <bluetooth/bluetooth.h>
#include <bluetooth/hci.h>
#include <bluetooth/hci_lib.h>

int main(int argc, char **argv)
{undefined
    inquiry_info *ii = NULL;
    int max_rsp, num_rsp;
    int dev_id, sock, len, flags;
    int i;
    char addr[19] = { 0 };
    char name[248] = { 0 };

    dev_id = hci_get_route(NULL);
    sock = hci_open_dev( dev_id );
    if (dev_id < 0 || sock < 0) {undefined
        perror("opening socket");
        exit(1);
    }

    len = 8;
    max_rsp = 255;
    flags = IREQ_CACHE_FLUSH;
    ii = (inquiry_info*)malloc(max_rsp * sizeof(inquiry_info));
```

```
num_rsp = hci_inquiry(dev_id, len, max_rsp, NULL, &ii, flags);
if( num_rsp < 0 ) perror("hci_inquiry");

for (i = 0; i < num_rsp; i++) {undefined
    ba2str(&(ii+i)->bdaddr, addr);
    memset(name, 0, sizeof(name));
    if (hci_read_remote_name(sock, &(ii+i)->bdaddr, sizeof(name),
        name, 0) < 0)
        strcpy(name, "[unknown]");
    printf("%s %s\n", addr, name);
}

free( ii );
close( sock );
return 0;
}
```

4.1.1 编译

编译需要使用gcc链接libbluetooth这个库。

```
gcc -o simplescan simplescan.c -lbluetooth
```

4.1.2. 解释

```
typedef struct {undefined
    uint8_t b[6];
} __attribute__((packed)) bdaddr_t;
```

蓝牙设备的地址采用结构体bdaddr_t来描述，BlueZ中队蓝牙地址的存储和操纵都使用bdaddr_t结构体，BlueZ提供两个函数来进行字符串到蓝牙地址的转换。

```
int str2ba( const char *str, bdaddr_t *ba );
```

```
int ba2str( const bdaddr_t *ba, char *str );
```

str2ba把形如XX:XX:XX:XX:XX:XX(XX标识48位蓝牙地址的16进制的一个字节)的字符串转化6字节的bdaddr_t结构，ba2str完成相反的功能。

本地蓝牙适配器被分配一个从0开始的识别号码。程序在分配系统资源时必须指定使用那一个蓝牙适配器，通常的话系统只有一个蓝牙适配器，把参数NULL传给hci_get_route可以获得第一个有效的蓝牙适配器识别号。

```
int hci_get_route( bdaddr_t *bdaddr );
int hci_open_dev( int dev_id );
```

[note]将适配器的设备号指定为0是不恰当的，因为它并不总代表第一个可用的蓝牙适配器。例如系统有两个蓝牙适配器，第一个被disable掉了，那么第一个有效的设备号就是2。

如果存在多个蓝牙适配器，选择“01:23:45:67:89:AB”作为蓝牙适配器的地址，将指示这个地址的指针char *representation传给hci_devid函数，用这个函数替代hci_get_route。

很多蓝牙操作都需要打开一个套接口，hci_open_dev函数可以打开特定资源号的一个套接口，确切的说hci_open_dev打开的套接口建立了一条和本地蓝牙适配器控制器的连接，而不是和远端蓝牙设备的连接。使用这个套接口发送命令到蓝牙控制器可以实现底层的蓝牙操作，这部分在4.5中有详细的讨论。

选择好本地蓝牙适配器并进行系统资源分配后，程序就可以开始扫描周边的蓝牙设备了，在这个例程中，hci_inquiry函数完成对蓝牙设备的搜寻，并将返回的设备信息数据记录在变量ii中。遇到错误时，它将返回-1并设置errno变量。

```
int hci_inquiry(int dev_id, int len, int max_rsp, const uint8_t *lap, inquiry_info **ii, long flags);
```

hci_inquiry 的参数需要使用设备资源号而非套接口，所以我们使用hci_get_route函数的返回值dev_id传递给它。查询时间最长持续1.28 * len秒。max_rsp个设别返回的信息都被存储在变量ii中，这个变量必须有足够的空间来存储max_rsp返回的结果。我们推荐max_rsp取值 255来完成标准10.24秒的查询工作。

如果标志位flag设置为IREQ_CACHE_FLUSH，那么在进行查询操作时会把先前一次查询记录的cache刷新，否则flag设置为0的话，即便先前查询的设备已经不处于有效范围内，先前查询的记录也将被返回。

inquiry_info结构体定义如下

```
typedef struct {undefined
    bdaddr_t bdaddr;
    uint8_t pscan_rep_mode;
    uint8_t pscan_period_mode;
    uint8_t pscan_mode;
    uint8_t dev_class[3];
    uint16_t clock_offset;
} __attribute__((packed)) inquiry_info;
```

在大多数场合，我们仅用到成员bdaddr，它标识了设备的蓝牙地址。有些场合我们也会用到成员dev_class，它标识了被检测到的蓝牙设备的一些信息(例如，识别这个设备是打印设备，电话，个人电脑等)，详细地对应关系可以参见蓝牙设备分配号[3]。其余的成员在用于底层通信，一般情况并不常用。感兴趣的读者可以阅读蓝牙内核规范[4]获取更多的信息。一旦周围的蓝牙设备和其蓝牙地址被检测到，程序可以将此设备的名称提供给用户，hci_read_remote_name函数可以完成这个功能。

```
int hci_read_remote_name(int sock, const bdaddr_t *ba, int len,
                        char *name, int timeout)
```

hci_read_remote_name函数在规定的超时时间内使用套接口通过蓝牙地址ba去获取蓝牙设备的名称，成功返回0，并将获取的蓝牙设备名称存入name中；失败时返回-1并设置相应的errno。

Notes

[1]<http://www.bluez.org/lists.html>

[2] for the curious, it makes a call to socket(AF_BLUETOOTH, SOCK_RAW, BTPROTO_HCI), followed by a call to bind with the specified resource number.

[3]<https://www.bluetooth.org/foundry/assignnumb/document/baseband>

[4]<http://www.bluetooth.org/spec>

#####

<http://blog.csdn.net/ifq/archive/2006/06/03/772074.aspx>

今天用bluez的lib写了个小程序调试,运行通过.对linux下蓝牙编程有了点初步的认识.
功能就是检索周围是否有其它蓝牙设备,并得到他们的友好设备名.

```
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>

#include <bluetooth/bluetooth.h> //蓝牙的3个头文件.
#include <bluetooth/hci.h>
#include <bluetooth/hci_lib.h>

int main ( int argc , char **argv )
{
    inquiry_info *ii = NULL;
    int max_rsp, num_rsp;
    int dev_id, sock, len, flags;
    int i;
    char addr [19] = { 0 };
    char name [248] = { 0 };
    dev_id = hci_get_route (NULL); //得到本地第一个可用的蓝牙设备

    sock = hci_open_dev(dev_id); //用打开蓝牙设备.
    if( dev_id<0 || sock < 0) {
        perror("opening socket error");
        exit(1) ;
    }
    len = 8 ;
    max_rsp = 255 ;
    flags = IREQ_CACHE_FLUSH;
    ii = (inquiry_info*)malloc (max_rsp* sizeof ( inquiry_info)) ;

    printf("start search...\n");
    num_rsp = hci_inquiry(dev_id , len , max_rsp , NULL , &ii , flags) ; //检索周围是否有设备
    if ( num_rsp < 0 ) perror ("hci_inquiry error") ;
    for ( i = 0 ; i < num_rsp ; i++) {
        ba2str (&(ii+i)->bdaddr , addr ) ;
        memset (name , 0 , sizeof (name)) ;
        if( hci_read_remote_name ( sock , &( ii+i )->bdaddr , sizeof (name) ,
            name , 0) < 0) //查询设备的友好设备名
            strcpy (name , "[unknown]") ;
        printf ("%s %s \n", addr , name ) ;
    }
    printf("end search.\n");
    free(ii);
    close(sock);
    return 0;
}
```

最后把这个程序交叉编译了一下,也通过了.但是还没有测试.明天再来传到板子上.
还没有正式在板子上运行过蓝牙,可能还有很多问题.



believe209
码龄15年 暂无认证

81	7万+	137万+	184万+	
原创	周排名	总排名	访问	等级
1万+	304	323	102	1218

积分 粉丝 获赞 评论 收藏

恒

私信 关注

搜博文文章

热门文章

windows下添加路由 119939 IP数据报格式详解 77022 tftp命令使用说明 51636 C#播放声音【六种方法】 50291 Linux 下使用USB 网络 48468

最新评论

C# 窗口传递信息 脑子娃: 这是搬运过来的嘛? 有实现过吗? 第四种方法能详细说一下嘛? otg接口形式 Wi达布溜: 你这笔记怎么做的和其他人的不一样 😊 检测ID为低还会协商么? 别人说... Aspose.Cells 使用汇总 你好请坐再见: 请问加载200m文件时报outo fmemory的错应该如何处理 linux下bluetooth编程 (三) HCI层编程 chaoshuaihaohao: 这些图片原文档有链接吗? linux下基于QT的串口程序 zhangshuyang321: 串口参数设置部分有一个错了, 是myCom->setDataBits(QSerial...

您愿意向朋友推荐“博客详情页”吗?

😞

😐

😐

😄

😄

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

Linux环境下的GDB调试方法 【C#基础知识】获取网卡的ip地址和MAC ARP协议工作原理[同网段及跨网段] 2020年 1篇 2017年 33篇 2016年 99篇 2015年 130篇 2014年 26篇 2013年 5篇 2012年 7篇

蓝牙开发代码 (C语言) 蓝牙c开发 两个完整的c程序,基于BlueZ的C语言蓝牙编程详细过程和代码

【蓝牙】基于Bluez的Bluetoothctl BLE开发 love131452098的博客 709 对于BLE的开发,大部分的BLE开发者都是在BLE芯片原厂的SDK基础上进行BLE的应用开发,跟更多的是基于SDK的应用开发而对于BLE的协议、框架知...

抢沙发

评论

请发表有价值的评论, 博客评论不欢迎灌水, 良好的社区氛围需大家一起维护。

用C写蓝牙通讯程序: 扫描、读取、发送 believe的专栏 1万+ 在Linux下,通过bluez 蓝牙库可以用C语言轻松实现蓝牙通信。在ubuntu下可以用 apt-get install libbluetooth-dev 安装该库。下面是几个简单示例。一个...

bluez的一些网站 Linux Job 63 BlueZ 的wiki http://wiki.bluez.org/ An Introduction to Bluetooth Programming http://people.csail.mit.edu/albert/bluez-intro/ 其中一章的翻译, 基于BlueZ...

蓝牙bluez5的开发方法及入门教程 霍宏鹏的专栏 1377 1 摘要 这篇文章的主要目的是告诉大家应该如何使用bluez进行开发,由于bluez的文档实在太少了,入门门槛实在太高了,很多人无从下手,准备写一个...

基于BlueZ的C语言蓝牙编程 (hci_inquiry) caobai的专栏 3431 【转】基于BlueZ的C语言蓝牙编程 (hci_inquiry) 2015-03-14 17:06:52 分类: 原文地址: 【转】基于BlueZ的C语言蓝牙编程 (hci_inquiry) 作者: glietht...

Linux下Bluez的编程实现 07-16 Linux下Bluez的编程实现

蓝牙C语言编程 2944 基于BlueZ的C语言蓝牙编程 有很多理由促使我们选用C替代其他高级语言来例如Python来开发蓝牙应用程序。Python环境可能并不适合于嵌入式系统。因...

蓝牙DEMO程序 04-10 开发蓝牙时以它为参考通讯方式,操作方式,功能比较全面

linux c 蓝牙开发环境,基于BlueZ的C语言蓝牙编程 最新发布 weixin_35975197的博客 249 原文出处: 第四章 基于BlueZ的C语言蓝牙编程有很多理由促使我们选用C替代其他高级语言来例如Python来开发蓝牙应用程序。Python环境可能并不适合...

基于BlueZ 的BLE蓝牙开发 Audio and Video Development 1万+ 蓝牙,是一种支持设备短距离通信(一般10m内)的无线电技术。工作在全球通用的2.4GHz ISM(即工业、科学、医学)频段。数据速率为1Mb/s。时分双...

Linux下Bluez编程实现 2538 本文非原创,如原作者不同意转载,请告知。 源链接: http://elsila.blog.163.com/blog/static/173197158201092205656895/#__RefHeading__47_192...

蓝牙 bluez 的编程 C C++ weixin_33748818的博客 481 蓝牙 bluez 的编程 C C++ 简介 bluez目录有一个libbluetooth.a文件 有一个目录 lib目录里面存储这网络连接的部分代码 基于库的代码编程. 在linux下如果自...

基于蓝牙的数据传输系统的设计 SingleCode的专栏 1万+ 蓝牙 (Bluetooth) 是由东芝、爱立信、IBM、Intel和诺基亚于1998年5月共同提出的近距离无线数据通讯技术标准。它能够在10米 的半径范围内实现单点...

ubuntu bluetooth编程环境搭建 whb_fei的专栏 1606 Ubuntu Bluetooth环境搭建 1. 准备 准备: bluez-libs-3.36.tar.gz (bluetooth所需要的库及头文件) 2. 安装 tar -xvf bluez-libs-3.36.tar.gz ...

gatttool的使用 热门推荐 luotong86的专栏 1万+ 先搜索BLE设备

编程设计思想 YuhangZeng_的博客 460 场景化与抽象化: 涉及的主体与角色,交互过程,子过程, message; 虚拟世界 (编程) ->现实世界 (人的互动), 软件中 (分层) -硬件中 (分层) -现...

基于labview的温湿度数据采集_基于Raspberry Pi (树莓派)的MCC数据采集卡应用 weixin_39921689的博客 36 简介树莓派是一种单板式计算机系统,可轻松连接鼠标、键盘、显示器等外设,并运行基于Linux®的操作系统,低于300人民币的成本,使树莓派尤其适...

©2021 CSDN 皮肤主题: 创作都市 设计师:CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

believe209 关注

2 0 15

专栏目录

?

举报