

HC-05 嵌入式蓝牙串口通讯模块

AT 指令集

HC-05 嵌入式蓝牙串口通讯模块（以下简称模块）具有两种工作模式：命令响应工作模式和自动连接工作模式，在自动连接工作模式下模块又可分为主（Master）从（Slave）和回环（Loopback）三种工作角色。当模块处于自动连接工作模式时，将自动根据事先设定的方式连接的数据传输；当模块处于命令响应工作模式时能执行下述所有 AT 命令，用户可向模块发送各种 AT 指令，为模块设定控制参数或发布控制命令。通过控制模块外部引脚（PIO11）输入电平，可以实现模块工作状态的动态转换。

串口模块用到的引脚定义：

- 1、PIO8 连接 LED，指示模块工作状态，模块上电后闪烁，不同的状态闪烁间隔不同。
- 2、PIO9 连接 LED，指示模块连接成功，蓝牙串口匹配连接成功后，LED 长亮。
- 3、PIO11 模块状态切换脚，高电平-->AT 命令响应工作状态，低电平或悬空-->蓝牙常规工作状态。
- 4、模块上已带有复位电路，重新上电即完成复位。

设置为主模块的步骤：

- 1、PIO11 置高。
- 2、上电，模块进入 AT 命令响应状态。
- 3、超级终端或其他串口工具，设置波特率 38400，数据位 8 位，停止位 1 位，无校验位，无流控制。
- 4、串口发送字符“AT+ROLE=1\r\n”，成功返回“OK\r\n”，其中\r\n为回车换行。
- 5、PIO 置低，重新上电，模块为主模块，自动搜索从模块，建立连接。

指令详细说明

(AT 指令不区分大小写，均以回车、换行字符结尾：\r\n)

1、测试指令：

| 指令 | 响应 | 参数 |
|----|----|----|
| AT | OK | 无 |

2、模块复位（重启）：

| 指令 | 响应 | 参数 |
|----------|----|----|
| AT+RESET | OK | 无 |

3、获取软件版本号：

| 指令 | 响应 | 参数 |
|-------------|------------------------|-------------|
| AT+VERSION? | +VERSION:<Param> OK | Param：软件版本号 |

举例说明：

```
at+version?\r\n
+VERSION:2.0-20100601
OK
```

4、恢复默认状态：

| 指令 | 响应 | 参数 |
|---------|----|----|
| AT+ORGL | OK | 无 |

出厂默认状态：

- . 设备类：0
- . 查询码：0x009e8b33
- . 模块工作角色：Slave Mode
- . 连接模式：指定专用蓝牙设备连接模式
- . 串口参数：波特率— 38400bits/s;停止位：1位；校验位：无
- . 配对码：“1234”
- . 设备名称：“H-C-2010-06-01”
-

5、获取模块蓝牙地址：

| 指令 | 响应 | 参数 |
|----------|---------------------|--------------|
| AT+ADDR? | +ADDR:<Param> OK | Param：模块蓝牙地址 |

蓝牙地址表示方法：NAP：UAP：LAP（十六进制）

举例说明：

模块蓝牙设备地址为：12：34：56：ab:cd:ef

```
at+addr?\r\n
+ADDR:1234:56:abcdef
OK
```

6、设置/查询设备名称：

| 指令 | 响应 | 参数 |
|-----------------|---|------------------------------|
| AT+NAME=<Param> | OK | Param：蓝牙设备名称 默认名称：“HC-05” |
| AT+NAME？ | 1、+NAME:<Param> OK——成功 2、FAIL——失败 | |

例如：

AT+NAME=HC-05\r\n ——设置模块设备名为：“HC-05”

OK

AT+NAME=“HC-05”\r\n ——设置模块设备名为：“HC-05”

OK

at+name=Beijin\r\n ——设置模块设备名为：“Beijin”

OK

at+name=“Beijin”\r\n ——设置模块设备名为：“Beijin”

OK

at+name?\r\n

+NAME: Bei jin

OK

7、获取远程蓝牙设备名称：

| 指令 | 响应 | 参数 |
|-------------------|--|------------------------------------|
| AT+RNAME？<Param1> | 1、+NAME:<Param2> OK——成功 2、FAIL——失败 | Param1：远程蓝牙设备地址 Param2：远程蓝牙设备地址 |

蓝牙地址表示方法：NAP：UAP：LAP（十六进制）

例如：

模块蓝牙设备地址为：00:02:72:od:22:24，设备名称为：Bluetooth

at+rname? 0002, 72, od2224\r\n

+RNAME:Bluetooth

OK

8、设置/查询—模块角色：

| 指令 | 响应 | 参数 |
|-----------------|---------------------|--|
| AT+ROLE=<Param> | OK | Param：参数取值如下： 0——从角色（Slave） 1——主角色（Master） 2——回环角色（Slave-Loop） 默认值：0 |
| AT+ROLE？ | +ROLE:<Param> OK | |

模块角色说明：

Slave(从角色)——被动连接；

Slave-Loop(回环角色)——被动连接，接收远程蓝牙主设备数据并将数据原样返回给远程蓝牙主设备；

Master（主角色）——查询周围 SPP 蓝牙从设备，并主动发起连接，从而建立主、从蓝牙设备间的透明数据传输通道。

9、设置/查询—设备类：

| 指令 | 响应 | 参数 |
|------------------|---|---|
| AT+CLASS=<Param> | OK | Param：设备类 |
| AT+ CLASS ? | 1、 + CLASS:<Param> OK——成功 2、 FAIL——失败 | 蓝牙设备类实际上是一个 32 位的参数,该参数用于指出设备类型,以及所支持的服务类型。 默认值：0 具体设置见附件 1：设备类说明 |

为了能有效地对周围诸多蓝牙设备实施过滤，快速查询或被查询自定义蓝牙设备，用户可以将模块设置为非标准蓝牙设备类，如：0x1f1f（十六进制）。

10、设备/查询—查询访问码：

| 指令 | 响应 | 参数 |
|----------------|--------------------------|---------------------------|
| AT+IAC=<Param> | 1、 OK——成功 2、 FAIL——失败 | Param：查询访问码 默认值：9e8b33 |
| AT+ IAC ? | +IAC：<Param> OK | 具体设置见附件 2：查询访问码说明 |

访问码设置为 GIAC（General Inquire Access Code:0x9e8b33）通用查询访问码，可用来发现或被发现周围所有的蓝牙设备；为了能有效地在周围诸多蓝牙设备中快速查询或被查询自定义蓝牙设备，用户可以将模块查询访问码设置成 GIAC 和 LIAC 以外的数字，如 9e8b3f。

举例：
AT+IAC=9e8b3f\r\n
OK
AT+IAC?\r\n
+IAC: 9e8b3f
OK

11、设置/查询—查询访问模式：

| 指令 | 响应 | 参数 |
|-----------------------------------|---------------------------------------|--|
| AT+INQM=<Param>,<Param2>,<Param3> | 1、 OK——成功 2、 FAIL——失败 | Param：查询模式 0——inquiry_mode_standard |
| AT+ INQM ? | +INQM：<Param>,<Param2>,<Param3> OK | 1——inquiry_mode_rssi Param2:最多蓝牙设备响应数 Param3:最大查询超时 超时范围：1~ 48 （折合成时间：1.28秒~ 61.44秒） 默认值：1, 1, 48 |

举例：
AT+INQM=1,9,48\r\n——查询模式设置：带 RSSI 信号强度指示，超过 9 个蓝牙设备响应则终止查询，设定超时为 48x1.28=61.44 秒。
OK
AT+INQM\r\n
+INQM:1, 9, 48
OK

12、设置/查询—配对码：

| 指令 | 响应 | 参数 |
|-----------------|----------------------|--------------------------|
| AT+PSWD=<Param> | OK | Param：配对码 默认名称：“1234” |
| AT+PSWD? | +PSWD :<Param> OK | |

13、设置/查询—串口参数：

| 指令 | 响应 | 参数 |
|-----------------------------------|---------------------------------------|---|
| AT+UART=<Param>,<Param2>,<Param3> | OK | Param1：波特率 (bits/s) |
| AT+UART? | +UART=<Param>,<Param2>,<Param3> OK | 取值如下 (十进制)： 4800 9600 19200 38400 57600 115200 23400 460800 921600 1382400 Param2：停止位 0——1位 1——2位 Param3：校验位 0——None 1——Odd 2——Even 默认设置：9600, 0, 0 |

举例：设置串口波特率：115200，2位停止位，Even 校验

AT+UART=115200, 1,2,\r\n

OK

AT+UART?

+UART:115200,1,2

OK

14、设置/查询—连接模式：

| 指令 | 响应 | 参数 |
|------------------|----------------------|--|
| AT+CMODE=<Param> | OK | Param： 0——指定蓝牙地址连接模式 (指定蓝牙地址由绑定指令设置) 1——任意蓝牙地址连接模式 (不受绑定指令设置地址的约束) 2——回环角色 (Slave-Loop) 默认连接模式：0 |
| AT+CMODE? | +CMODE:<Param> OK | |

15、设置/查询—绑定蓝牙地址：

蓝牙地址表示方法：NAP：UAP：LAP（十六进制）

| 指令 | 响应 | 参数 |
|-----------------|----------------------|--------------------------------|
| AT+BIND=<Param> | OK | Param——绑定蓝牙地址 |
| AT+ BIND ? | + BIND:<Param> OK | 默认绑定蓝牙地址： 00:00:00:00:00:00 |

蓝牙地址表示方法：NAP：UAP：LAP（十六进制）

绑定指令只有在指定蓝牙地址连接模式时有效！

举例说明：

在指定蓝牙地址连接模式下，绑定蓝牙设备地址:12:34:56:ab:cd:ef

命令及响应如下：

AT+BIND=1234,56,abcdef\r\n

OK

AT+BIND?\r\n

+BIND:1234:56:abcdef

OK

16、设置/查询—LED 指示驱动及连接状态输出极性：

| 指令 | 响应 | 参数 |
|------------------------------|-----------------------------------|---|
| AT+POLAR=<Param1> , <Param1> | OK | Param1:取值如下 |
| AT+ BIND ? | + POLAR=<Param1> , <Param1> OK | 0——PI08 输出低电平点亮 LED 1——PI08 输出高电平点亮 LED Param2:取值如下 0——PI09 输出低电平指示连接成功 1——PI09 输出高电平指示连接成功 默认设置：1,1 |

HC-05 蓝牙模块定义：PI08 输出驱动 LED 指示工作状态；PI09 输出指示连接状态。

举例说明：

PI08 输出低电平点亮 LED，PI09 输出高电平指示连接成功。

命令及响应如下：

AT+POLAR=0,1\r\n

OK

AT+POLAR?\r\n

+POLAR=0,1

OK

17、设置 PIO 单端口输出：

| 指令 | 响应 | 参数 |
|----------------------------|----|---|
| AT+PIO=<Param1> , <Param2> | OK | Param1 :PIO 端口序号(十进制数) Param2 : PIO 端口输出状态 0——低电平 1——高电平 |

HC-05 型蓝牙模块为用户提供 PIO 端口资源：PI0 0~ PI07和 PI010, 用户可用来扩展输入、输出端口。

举例说明：

1、PI010 端口输出高电平

AT+PIO=10,1\r\n

OK

2、PIO10 端口输出高电平

AT+PIO=10,0\r\n

OK

18、设置 PIO 多端口输出：

| 指令 | 响应 | 参数 |
|-----------------|----|---------------------------|
| AT+MPIO=<Param> | OK | Param :PIO 端口序号掩码组合(十进制数) |

HC-05型蓝牙模块为用户提供 PIO端口资源：PI00~ PI07和 PI010, 用户可用来扩展输入、输出端口。

PIO端口序号掩码=(1<<端口序号)

PIO端口序号掩码组合=(PIO端口序号掩码 1|PIO端口序号掩码 2|.....)

如：

PI02端口掩码=(1<<2) =0x004

PI010端口掩码=(1<<10) =0x400

PI02和 PI010端口掩码组合=(0x004|0x400) =0x404

举例说明：

1 PI010和 PI02端口输出高电平

AT+MPIO=404\r\n

OK

2 PI04端口输出高电平

AT+PIO=004\r\n

OK

3 PI010端口输出高电平

AT+PIO=400\r\n

OK

5 所有端口输出低电平

AT+MPIO=0\r\n

OK

19 查询 PIO端口输入

| 指令 | 响应 | 参数 |
|-----------|-----------------------|---|
| AT+MPIO ? | +MPIO : <Param> OK | Param——PIO 端口值 (16bits) Param[0]=PI00 Param[1]=PI01 Param[2]=PI02 Param[10]=PI010 Param[11]=PI011 |

HC-05型蓝牙模块为用户提供 PIO端口资源：PI00~ PI07和 PI010~ PI011, 用户可用来扩展输入、输出端口。

20 设置 /查询—寻呼扫描、查询扫描参数：

| 指令 | 响应 | 参数 |
|---|---|--|
| AT+IPSCAN=<Param1> , <Param2> , <Param3> , <Param4> AT+IPSCAN ? | OK +IPSCAN: <Param1> , <Param2> , <Param3> , <Param4> OK | Param1 : 查询时间间隔 Param2 : 查询持续时间 Param3 : 寻呼时间间隔 Param4 : 寻呼持续时间 上述参数均为十进制数。 默认值 : 1024,512,1024,512 |

举例说明：

```
at+ipscan=1234,500,1200,250\r\n
OK
at+ipscan?
+IPSCAN:1234,500,1200,250
```

21 设置 / 查询— SNIFF节能参数：

| 指令 | 响应 | 参数 |
|--|---|--|
| AT+SNIFF=<Param1> , <Param2> , <Param3> , <Param4> | OK | Param1：最大时间 Param2：最小时间 |
| AT+IPSCAN ? | +SNIFF: <Param1> , <Param2> , <Param3> , <Param4> | Param3：尝试时间 Param4：超时时间 上述参数均为十进制数。 默认值：0,0,0,0 |

22 设置 / 查询安全、加密模式：

| 指令 | 响应 | 参数 |
|----------------------------|----------------------------------|--|
| AT+SENM=<Param>,<Param2> , | 1、OK——成功 2、FAIL——失败 | Param：安全模式，取值如下： 0——sec_mode0+off |
| AT+ SENM ? | + SENM :<Param>,<Param2> , OK | 1——sec_mode1+non_secure 2——sec_mode2_service 3——sec_mode3_link 4——sec_mode_unknown Param2 加密模式，取值如下： 0——hci_enc_mode_off 1——hci_enc_mode_pt_to_pt 2——hci_enc_mode_pt_to_pt_and_bcast 默认值：0,0 |

23 从蓝牙配对列表中删除指定认证设备 (Authenticated Device)：

| 指令 | 响应 | 参数 |
|------------------|----|--------------|
| AT+PMSAD=<Param> | OK | Param：蓝牙设备地址 |

举例说明：

从配对列表中删除蓝牙地址为：12:34:56:ab:cd:ef的设备

```
at+mrsad=1234,56,abcdef\r\n
```

OK——删除成功

或

```
at+mrsad=1234,56,abcdef\r\n
```

FAIL——配对列表中不存在 12:34:56:ab:cd:ef 蓝牙设备

24 从蓝牙配对列表中删除所有认证设备 (Authenticated Device)：

| 指令 | 响应 | 参数 |
|----------|----|----|
| AT+RMAAD | OK | 无 |

举例说明：

从配对列表中删除所有蓝牙设备

```
at+maad\r\n
```

OK

25 从蓝牙配对列表中查找指定的认证设备 (Authenticated Device)：

| 指令 | 响应 | 参数 |
|----|----|----|
|----|----|----|

| | | |
|-----------------|------------------------|--------------|
| AT+FSAD=<Param> | 1、OK——成功 2、FAIL——失败 | Param：蓝牙设备地址 |
|-----------------|------------------------|--------------|

举例说明：

从配对列表中查找蓝牙设备：12:34:56:ab:cd:ef

at+fsad=1234,56,abcdef\r\n

OK——配对列表中存在 12:34:56:ab:cd:ef 蓝牙设备。

at+fsad=1234,56,abcde0\r\n

FAIL——配对列表中不存在 12:34:56:ab:cd:e0 蓝牙设备。

26 获取蓝牙配对列表中认证设备数 (Authenticated Device Count):

| 指令 | 响应 | 参数 |
|----------|----------------------|------------------|
| AT+ADCN? | +ADCN: <Param> OK | Param：配对列表中蓝牙设备数 |

举例说明：

at+adcn?

+ADCN:0——配对信任列表中没有蓝牙设备

OK

27 获取最近使用过的蓝牙认证设备地址 (Most Recently Used Authenticated Device):

| 指令 | 响应 | 参数 |
|----------|----------------------|--------------------|
| AT+MRAD? | +MRAD: <Param> OK | Param：最近使用过的蓝牙设备地址 |

举例说明：

at+mrad?

+MRAD:0:0:0——最近没有使用信任蓝牙设备

OK

28 获取蓝牙模块工作状态：

| 指令 | 响应 | 参数 |
|-----------|-----------------------|--|
| AT+STATE? | +STATE: <Param> OK | Param：模块工作状态 返回值如下： “INITIALIZED”——初始化状态 “READY”——准备状态 “PAIRABLE”——可配对状态 “PAIRED”——配对状态 “INQUIRING”——查询状态 “CONNECTING”——正在连接状态 “CONNECTED”——连接状态 “DISCONNECTED”——断开状态 “NUKNOW”——未知状态 |

举例说明：

at+state?

+STATE:INITIALIZED——初始化状态

OK

29 初始化 SPP规范库 (Initialise the spp profile lib):

| 指令 | 响应 | 参数 |
|---------|------------------------|----|
| AT+INIT | 1、OK——成功 2、FAIL——失败 | 无 |

30 查询蓝牙设备

| 指令 | 响应 | 参数 |
|----|----|----|
|----|----|----|

| | | |
|--------|---|---|
| AT+INQ | +INQ: <Param1> <Param2> <Param3> , OK | Param1 : 蓝牙地址 Param2 : 设备类 Param3 : RSSI 信号强度 |
|--------|---|---|

举例说明 1:

```
at+init\r\n ——初始化 SPP库 (不能重复初始化)
OK
at+iac=9e8b33\r\n ——查询任意访问码的蓝牙设备
OK
at+class=0\r\n ——查询各种蓝牙设备类
at+inqm=1,9,48\r\n ——查询模式:带 RSSI 信号强度指示,超过 9 个蓝牙设备响应则终止查询,设定超
时为 48x1.28=61.44秒。
At+inq\r\n ——查询周边蓝牙设备
+INQ:2:72:D2224,3E0104,FFBC
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC0
+INQ:1234:56:0,1F1F,FFC1
+INQ:2:72:D2224,3F0104,FFAD
+INQ:1234:56:0,1F1F,FFBE
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFBE
+INQ:2:72:D2224,3F0104,FFBC
OK
```

举例说明 2:

```
at+iac=9e8b33\r\n ——查询任意访问码的蓝牙设备
OK
at+class=1f1f\r\n ——查询设备类为 0x1f1f 的蓝牙设备
OK
at+inqm=1,9,48\r\n ——查询模式:带 RSSI 信号强度指示,超过 9 个蓝牙设备响应则终止查询,设定超
时为 48x1.28=61.44秒。
At+inq\r\n ——过滤、查询周边蓝牙设备
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC0
+INQ:1234:56:0,1F1F,FFC2
OK
```

举例说明 3:

```
at+iac=9e8b3f\r\n ——查询访问码为 0x9e8b3f 的蓝牙设备
OK
at+class=1f1f\r\n ——查询设备类为 0x1f1f 的蓝牙设备
OK
at+inqm=1,1,20\r\n ——查询模式:带 RSSI 信号强度指示,超过 1 个蓝牙设备响应则终止查询,设定超
时为 20x1.28=25.6秒。
At+inq\r\n ——过滤、查询周边蓝牙设备
+INQ:1234:56:ABCDEF,1F1F,FFC2
OK
```

31 取消查询蓝牙设备:

| 指令 | 响应 | 参数 |
|---------|----|----|
| AT+INQC | OK | 无 |

32 设备配对:

| 指令 | 响应 | 参数 |
|---------------------------|------------------------|---------------------------------------|
| AT+PAIR=<Param1>,<Param2> | 1、OK——成功 2、FAIL——失败 | Param1 : 远程设备蓝牙地址 Param2 : 连接超时(秒) |

举例说明:

与远程蓝牙设备：12:34:56:ab:cd:ef配对，最大配对超时 20秒。

At+pai=1234,56,abcdef,20\r\n

OK

33 设备连接：

| 指令 | 响应 | 参数 |
|-----------------|------------------------|----------------|
| AT+LINK=<Param> | 1、OK——成功 2、FAIL——失败 | Param：远程设备蓝牙地址 |

举例说明：

与远程蓝牙设备：12:34:56:ab:cd:ef建立连接

at+fsad=1234,56,abcdef\r\n——查询蓝牙设备 12:34:56:ab:cd:ef是否在配对列表中

OK

at+link=1234,56,abcdef\r\n——查询蓝牙设备 12:34:56:ab:cd:ef在配对列表中,不需查询可直接连接。

OK

34 断开连接

| 指令 | 响应 | 参数 |
|---------|--|----|
| AT+DISC | 1 +DISC:SUCCESS——断开连接成功 OK 2 +DISC:LINK_LOSS——连接丢失 OK 3 +DISC:NO_SLG——没有 SLC连接 OK 4 +DISC:TIMEOUT——断开超时 OK 5 +DISC:ERROR——断开错误 OK | 无 |

35 进入节能模式：

| 指令 | 响应 | 参数 |
|--------------------|----|--------------|
| AT+ENSNIFF=<Param> | OK | Param：设备蓝牙地址 |

36 退出节能模式

| 指令 | 响应 | 参数 |
|--------------------|----|--------------|
| AT+EXSNIFF=<Param> | OK | Param：设备蓝牙地址 |

附录 1: AT指令错误代码说明

错误代码返回形式——ERROR:(error_code)

| error_code(十六进制数) | 注释 |
|-------------------|--------|
| 0 | AT命令错误 |

| | |
|----|-------------------|
| 1 | 指令结果为默认值 |
| 2 | PSKEY写错误 |
| 3 | 设备名称太长 (超过 32个字节) |
| 4 | 设备名称长度为零 |
| 5 | 蓝牙地址 : NAP太长 |
| 6 | 蓝牙地址 : UAP太长 |
| 7 | 蓝牙地址 : LAP太长 |
| 8 | PIO序号掩码长度为零 |
| 9 | 无数 PIO序号 |
| A | 设备类长度为零 |
| B | 设备类数字太长 |
| C | 查询访问码长度为零 |
| D | 查询访问码数字太长 |
| E | 无效查询访问码 |
| F | 配对码长度为零 |
| 10 | 配对码太长 (超过 16个字节) |
| 11 | 模块角色无效 |
| 12 | 波特率无效 |
| 13 | 停止位无效 |
| 14 | 校验位无效 |
| 15 | 配对列表中不存在认证设备 |
| 16 | SPP库没有初始化 |
| 17 | SPP库重复初始化 |
| 18 | 无效查询模式 |
| 19 | 查询超时太大 |
| 1A | 蓝牙地址为零 |
| 1B | 无效安全模式 |
| 1C | 无效加密模式 |

附录 2: 设备类说明

The Class of Device/Service(CoD) is a 32 bits number that of 3 field specifies the service supported by the device. Another field specifies the minor device class, which describes the device type in more detail

The Class of Device /Service (CoD) field has a variable format. The format is

indicated using the ' within the CoD .The length of the Format Type field is variable and ends with two bits different from' 11' .The version field starts at the least significant bit of the CoD and may extend upwards. In the ' format#1' of the CoD (format Type field =00), 11 bits are assigned as a bit – mask(multiple bits can be set) each bit corresponding to a high level generic category of service class. Currently 7 categories are defined. These are primarily of a ' public service' nature. The remaining 11 bits are used to indicate device type category and other device-specific characteristics. Any reserved but otherwise unassigned bits, such as in the Major Service Class field, should be to 0.

Figure 1.2: The Class of Device/Service field (format type). Please note the order in which the octets are sent on the air and stored in memory. Bit number 0 is sent first on the air .

1. MAJOR SERVICE CLASSES

Bit no Major Service Class

- 13 Limited Discoverable Mode [Ref #1]
- 14 (reserved)
- 15 (reserved)
- 16 Positioning(Location identification)
- 17 Networking (LAN,Ad hoc, ...)
- 18 Rendering (Printing ,Speaker ,...)
- 19 Capturing (Scanner ,Microphone ,...)
- 20 Object Transfer (v-Inbox, v-Folder ,...)
- 21 Audio (Speaker ,Microphone ,Headset service ,...)
- 22 Telephony (Cordless telephony, Modem, Headset service ,...)
- 23 Information (WEB-server, WAP- server ,...)

TABLE 1.2: MAJOR SERVICE CLASSES

[Ref #1 As defined in See Generic Access Profile,Bluetooth SIG]

2. MAJOR DEVICE CLASSES

The Major Class segment is the highest level of granularity for defining a Bluetooth Device. The main function of a device is used to determine the major Class grouping. There are 32 different possible major classes. The assignment of this Major Class field is defined in Table 1.3.

- 1 2 1 1 1 0 9 8 Major Device Class
- 0 0 0 0 0 Miscellaneous [Ref #2]
- 0 0 0 0 1 Computer (desktop, notebook, PDA, organizers, ...)
- 0 0 0 1 0 Phone (cellular ,cordless ,payphone,modem ,...)
- 0 0 0 1 1 LAN/Network Access point
- 0 0 1 0 0 Audio/Video (headset, speaker, stereo, video display, vcr ...)
- 0 0 1 0 1 Peripheral (mouse, joystick, keyboards ...)
- 0 0 1 1 0 Imaging (printing, scanner, camera, display, ...)
- 1 1 1 1 1 Uncategorized, specific device code not specified
- X X X X All other values reserved

TABLE 1.3: MAJOR DEVICE CLASSES

[Ref #2:Used where a more specific Major Device Class is not suited (but only as

specified as in this document) .Devices that do not have a major class assigned can use the all-1 code until ' classified']

3. THE MINOR DEVICE CLASS FIELD

The ' Minor Device Class field' (bits 7 to 2 in the CoD), are to be interpreted only in the context of the Major Device Class (but interpreted of the Service Class field). Thus the meaning of the bits may change, depending on the value of the ' Major Device Class field' . When the Minor Device Class field indicates a device class ,then the primary device class should be reported, e. g . a cellular phone that can work as a cordless handset should

4. MINOR DEVICE CLASS FIELD- COMPUTER MAJOR CLASS

Minor Device Class

7 6 5 4 3 2 bit no of CoD

0 0 0 0 0 0 Uncategorized,code for device not assigned

0 0 0 0 0 1 Desktop workstation

0 0 0 0 1 0 Server-class computer

0 0 0 0 1 1 Laptop

0 0 0 1 0 0 Handheld PC/PDA(c clam shell)

0 0 0 1 0 1 Palm sized PC/PDA

0 0 0 1 1 0 Wearable computer (Watch sized)

X X X X X X All other values reserved

TABLE 1.4: SUB DEVICE CLASS FIELD FOR THE ' COMPUTER ' MAJOR CLASS

5. MINOR DEVICE CLASS FIELD – PHONE MAJOR CLASS

Minor Device Class

7 6 5 4 3 2 bit no of CoD

0 0 0 0 0 0 Uncategorized,code for device not assigned

000001 Cellular

0 0 0 0 1 0 Cordless

0 0 0 0 1 1 Smart phone

0 0 0 1 0 0 Wired modem or voice gateway

0 0 0 1 0 1 Common ISDN Access

0 0 0 1 1 0 Sim Card Reader

X X X X X X All other values reserved

TABLE 1.5: SUB DEVICE CLASSES FOR THE ' PHONE ' MAJOR CLASS

6. MINOR DEVICE CLASS FIELD – LAN/NETWORK ACCESS PHONE MAJOR CLASS

Minor Device Class

7 6 5 bit no of CoD

0 0 0 Fully available

0 0 1 1 – 17% utilized

0 1 0 1 7 - 33% utilized

0 1 1 3 3 – 50% utilized

1 0 0 5 0 – 67% utilized

1 0 1 6 7 – 83% utilized

1 1 0 8 3 – 99% utilized

1 1 1 No service available [REF #3]

XXX All other values reserved

TABLE 1.6: THE LAN/NETWORK ACCESS POINTE LOAD FACTOR FIELD

[Ref #3: " Device is fully utilized and cannot accept additional connections at this time, please retry later"]

The exact loading formula is not standardized. It is up to each LAN/Network Access Point implementation to determine what internal conditions to report as a utilization of communication requirement is that the box .As a recommendation, a client that locates multiple LAN/Network Access Points should attempt to connect to the one reporting the lowest load.

Minor Device Class

4 3 2 bit no of CoD

0 0 0 Uncategorized (use this value if no other apply)

XXX All other values reserved

TABLE 1.7: RESERVED SUB-FIELD FOR THE LAN/NETWORK ACCESS POINTE

7. MINOR DEVICE CLASS FIELD – AUDIO/VIDEO MAJOR CLASS

Minor Device Class

7 6 5 4 3 2 bit no of CoD

0 0 0 0 0 0 Uncategorized, code not assigned

0 0 0 0 0 1 Device conforms to the Headset profile

000010 Hands-free

0 0 0 0 1 1 (Reserved)

0 0 0 1 0 0 Microphone

0 0 0 1 0 1 Loudspeaker

0 0 0 1 1 0 Headphones

0 0 0 1 1 1 Portable Audio

0 0 1 0 0 0 Car audio

0 0 1 0 0 1 Set-top box

0 0 1 0 1 0 HiFi Audio Device

001011 VCR

0 0 1 1 0 1 Camcorder

0 0 1 1 1 0 Video Monitor

0 0 1 1 1 1 Video Display and Loudspeaker

0 1 0 0 0 0 Video Conferencing

0 1 0 0 0 1 (Reserved)

0 1 0 0 1 0 Gaming/Toy [Ref #4]

XXXXXX All other values reserved

[Ret #4: Only to be used with a Gaming/Toy device that makes audio/video capabilities available via Bluetooth]

TABLE 1.8: SUB DEVICES FOR THE ' AUDIO/VIDEO' MAJOR CLASS

8. MINOR DEVICE CLASS FIELD – PERIPHERAL MAJOR CLASS

Minor Device Class

7 6 bit no of CoD

0 1 Keyboard

1 0 Pointing device

1 1 Carbo keyboard /pointing device

X X X All other values reserved

TABLE1.9: THE PERIPHERAL MAJOR CLASS KEYBOARD/POINTING DEVICE FIELD

Bits 6 and 7 independantly specify mouse, keyboard or carbo mouse/keyboard devices. These may be combined with the lower bits in a multifunctional device.

Minor Device Class

5 4 3 2 bit no of CoD

0 0 0 0 Uncategorized device

0 0 0 1 Gamepd

0 0 1 1 Remote control

0 1 0 0 Sensing device

0 1 0 1 Digitizer tablet

X X X X All other values reserved

TABLE1.10: RESERVED SUB-FIELD FOR THE DEVICE TYPE

9. MINOR DEVICE CLASS FIELD – IMAGING MAJOR CLASS

Minor Device Class

7 6 5 4 bit no of CoD

X X X 1 Display

X X 1 X Camera

X 1 X X Scanner

1 X X X Printer

X X X X All other values reserved

TABLE 1.11: THE IMAGING MAJOR CLASS BITS 7 TO 7

Bits 4 to 7 independent ly specify bi splay, camera, scanner or printer. These may be combined in a multifunctional device.

Minor Device Class

3 2 bit no of CoD

0 0 Uncategorized, default

X X All other values reserved

TABLE 1. 12: THE IMAGING MAJOR CLASS BITS 2 AND 3

Bits 2 and 3 are reserved

附录 3: 查询访问码说明 (The Inquiry Access Codes)

The General-and Device-Specific Inquiry Access Codes (DIACs)

The Inquiry Access Code is the first level of filtering when finding Bluetooth devices. The main purpose of defining multiple IACs is to limit the number of responses that are received when scanning devices within range.

0. 0x9E8B33 — General/Unlimited Inquiry Access Code(GIAC)

1. 0x9E8B00 — Limited Dedicated Inquiry Access Code(LIAC)

2. 0x9E8B01 ~ 0x9E8B32 RESERVED FOR FUTURE USE

3. 0x9E8B34 ~ 0x9E8B3F RESERVED FOR FUTURE USE

The Limited Inquiry Access Code(LIAC) is only intended to be used for Limited time periods in scenarios where both sides have been explicitly caused to enter this state, usually by user action. For further explanation of the use of the LIAC, please refer to the Generic Access Profile.

In contrast it is allowed to be continuously scanning for the General Inquiry Access Code (GIAC) and respond whenever inquired.