HC-05 Embedded Bluetooth serial communication module AT order set

The HC-05 Embedded Bluetooth Serial Communication Module (hereinafter referred to as the module) has two operating modes: Command Response Mode and Automatic Connection Mode. In the Automatic Connection Mode, the module can function in three roles: Master (Master), Slave (Slave), and Loopback (Loopback). When the module is in Automatic Connection Mode, it will automatically connect to data transmission according to pre-set methods; when in Command Response Mode, it can execute all the following AT commands. Users can send various AT commands to the module to set control parameters or issue control commands. By controlling the input level of an external pin (PIO11) of the module, dynamic switching of the module's operating state can be achieved.

Pin definition used by the serial port module:

1、 The LED is connected to the PIO8 to indicate the working status of the module. The LED flashes after the module is powered on, and the flashing interval varies with different states.

2、 The LED of P9 is connected to indicate that the module connection is successful. After the Bluetooth serial port matching connection is successful, the LED is long on.

3、 PIO11 Module status switch foot, high level--> AT command response working state, low level or suspended--> Bluetooth normal working state.

4 The reset circuit is already on the module, and the reset is completed when the power is reconnected.

Steps to set the main module:

1、 PIO11 Place high.

2. Power on, and the module enters the AT command response state.

3. Super terminal or other serial port tools, set the baud rate 38400, data bits 8, stop bits 1, no parity, no flow control.

4. Send the character "AT+ROLE=1\r\n" through the serial port and return "OK\r\n" successfully, where \r\n is the carriage return.

5、 Lower the PIO, re-power on, the module is the master module, automatically search for the slave module, and establish the connection.

Detailed instructions

(AT commands are case-insensitive and end with carriage return and line feed characters: \n)

1, test instruction:

Instruct	Respond	Parameter
AT	OK	Not have

2, Module reset (restart):

Instruct	Respond	Parameter
AT+RESET	OK	Not have

3, Get software version number:

Instruct	Respond	Parameter
AT+VERSION？	+VERSION：<Param> OK	Param: Software version number

illustrate:

at+version?\r\n +

VERSION:2.0-20100601

OK

4. Restore the defa-

ult state:

Instruct	Respond	Parameter
AT+ORGL	OK	Not have

Factory default status:

① . Equipment: 0

② Query code: 0x009e8b33

③ . Module working roles: Slave Mode

④ . Connection mode: Specifies the connection mode of a dedicated Bluetooth device

⑤ Serial port parameters: baud rate-38400bits/s; stop bit: 1 bit; parity: none

⑥ The pairing code: "1234"

⑦ . Equipment name: "H-C-2010-06-01"

.....

5, Obtain the Bluetooth address of the module:

Instruct	Respond	Parameter
AT+ADDR？	+ADDR：<Param> OK	Param: Module Bluetooth address

Bluetooth address representation method:

NAP: UAP: LAP (hexadecimal) Example:

The address of the module Bluetooth device is 12:34:56:ab:cd:ef

at+addr?\r\n +ADDR:

1234:56:abcdef OK

6. Set/query device name:

Instruct	Respond	Parameter
AT+NAME=<Param>	OK	Param: Name of the Blueto-oth device Default name: "HC-05"
AT+NAME？	1、 +NAME: <Param> OK——Success 2、 FAIL—— be defea-ted	

for instance:

AT+NAME=HC-05\r\n —— Set the module device name to "HC-05" OK

AT+NAME= "HC-05" \r\n —— Set the module device name to: "HC-05" OK

at+name=Beijin\r\n —— The name of the module device is "Beijin"

OK

at+name=“Beijin”\r\n —— The name of the module device is "Beijin"

OK

at+name?\r\n

+NAME: Bei jin

OK

7, Get the name of a remote Bluetooth device:

Instruct	Respond	Parameter
AT+RNAME？<Param1>	1、 +NAME: <Param2> OK—— success 2、 FAIL—— be defeated	Param1: Address of a re-mote Bluetooth device Param2: Address of a re-mote Bluetooth device

Bluetooth address representation: NAP: UAP: LAP (hexad-ecimal)

for instance:

The address of the module Bluetooth device is: 00:02:72: od:22:24, and the device name is: Bluetooth at+rname? 0002,72,od2224\r\n

+RNAME:Bluetooth

OK

8. Settings/Queries-Module Roles:

Instruct	Respond	Parameter
AT+ROLE=<Param>	OK	Param: The parameter values are as follows: 0—— From the role (Slave) 1—— The main role (Mast-er) 2—— The loop role (Slave-Loop) Default value: 0
AT+ ROLE？	+ ROLE:<Param> OK	

Module role description:

Slave (from the role) —— passive connection;

Slave-Loop (loopback role) —— Passive connection, receiving data from the remote Bluetooth master device and returning the data to the remote Bluetooth master device as it is;

Master (main role) —— Query the surrounding SPP Bluetooth slave device and initiate the connection actively, so as to establish a transparent data transmission channel between the master and slave Bluetooth devices.

9. Settings/Queries-Device category:

Instruct	Respond	Parameter
AT+CLASS=<Param>	OK	Param: Device class The Bluetooth device class is actually a 32-bit parameter that indicates the type of device and the type of service it supports. Default value: 0 See Attachment 1: Equipment Description for specific Settings
AT+ CLASS？	1、 + CLASS: <Param> OK——Success 2、 FAIL—— be defeated	

In order to effectively filter many bluetooth devices around, quickly query or be queried custom Bluetooth devices, users can set the module to non-standard Bluetooth device class, such as: 0x1f1f (hexadecimal).

10. Equipment/Query-Query access code:

Instruct	Respond	Parameter
AT+IAC=<Param>	1、 OK—— success 2、 FAIL—— be defeated	Param: Query access code Default value: 9e8b33 See Attachment 2 for specific Settings: Query Access Code Description
AT+ IAC？	+IAC：<Param> OK	

Set the access code to GIAC (General Inquire Access Code: 0x9e8b33) for general query access code, which can be used to discover or be discovered by all nearby Bluetooth devices; to effectively query or be queried by custom Bluetooth devices among many nearby Bluetooth devices, users can set the module query access code to a number other than GIAC and LIAC, such as 9e8b3f. For example:

AT+IAC=9e8b3f\r\n

OK

AT+IAC?\r\n

+IAC: 9e8b3f

OK

11. Settings/Queries-Query access mode:

Instruct	Respond	Parameter
AT+INQM=<Param>,<Param2>,<Param3>	1、 OK—— success 2、 FAIL—— be defeated	Param: query mode 0——inquiry_mode_standard 1——inquiry_mode_rssi Param2: The maximum number of Bluetooth devices that respond Param3: Maximum query timeout Timeout range: 1~48 (Equivalent to time: 1.28 seconds ~ 61.44 seconds) Default value: 1,1,48
AT+ INQM？	+INQM：<Param>,<Param2>,<Param3> OK	

give an example:

AT+INQM=1,9,48\r\n—— Query mode Settings: with RSSI signal strength indicator, the query is terminated when more than 9 Bluetooth devices respond, and the timeout is set to 48xl. 28=61.44 seconds.

OK
AT+INQM\r\n
+INQM:1, 9, 48
OK

12. Settings/Query-Pairing code:

Instruct	Respond	Parameter
AT+PSWD=<Param>	OK	Param: Pairing code Default name: "1234"
AT+ PSWD？	+ PSWD ：<Param> OK	

13. Settings/Query-Serial port parameters:

Instruct	Respond	Parameter
AT+UART=<Param>,<Param2>,<Param3>	OK	Param1: The baud rate (bits/s) is as follows (decimal): 4800 9600 19200 38400 57600 115200 23400 460800 921600 1382400 Param2: stop bit 0——1 bit 1——2 bit Param3: parity bit 0——None 1——Odd 2——Even Default Settings: 9600,0, 0
AT+ UART？	+ UART=<Param>,<Param2>, <Param3> OK	

For example: Set the serial port baud rate: 115200,2 stop bits, Even parity AT+UART=
115200,1,2,\r\n
OK
AT+UART?
+UART:115200,1,2
OK

14. Settings/Query-Connection mode:

Instruct	Respond	Parameter
AT+CMODE=<Param>	OK	Param： 0—— Designated Bluetooth address connection mode (the designated Bluetooth address is set by the binding instruction) 1—— Any Bluetooth address connection mode (not restricted by the address set by the binding instruction) 2—— Loopback role (Slave-Loop) The default connection mode is 0
AT+ CMODE？	+ CMODE:<Param> OK	

15. Set/Query-Bind Bluetooth address:

Bluetooth address representation: NAP: UAP: LAP (hexadecimal)

Instruct	Respond	Parameter
AT+BIND=<Param>	OK	Param—— Bind Bluetooth address The default Bluetooth address is bound: 00:00:00:00:00:00
AT+ BIND？	+ BIND:<Param> OK	

Bluetooth address representation: NAP: UAP: LAP (hexadecimal)

The binding instruction is only valid when the specified Bluetooth address connection mode is used!

illustrate:

In the specified Bluetooth address connection mode, bind the Bluetooth device address: 12:34:56:ab:cd:ef The command and response are as follows:

AT+BIND=1234，56，abcdef\r\n

OK

AT+BIND?\r\n

+BIND:1234:56:abcdef

OK

16. Settings/Query-LED indicator drive and connection status output polarity:

Instruct	Respond	Parameter
AT+POLAR=<Param1>，<Param1>	OK	Param1: The values are as follows 0——PIO8, output low level to light up LED 1——PIO8, output high level to light up LED Param2: the values are as follows 0——PIO9 output low level indicates successful connection 1——PIO9 output high level indicates successful connection default setting: 1, 1
AT+ BIND？	+ POLAR=<Param1>，<Param1> OK	

HC-05 Bluetooth module definition: PIO8, output drives the LED to indicate the working status; PIO9 output indicates the connection status.

illustrate:

PIO8 outputs a low level to light up the LED, and PIO9 outputs a high level to indicate a successful connection.

The command and response are as follows:

AT+POLAR=0，1\r\n

OK

AT+POLAR?\r\n

+POLAR=0，1

OK

17. Set the single port output of PIO:

Instruct	Respond	Parameter
AT+PIO=<Param1>，<Param2>	OK	Param1: The port number of the PIO (decimal number) Param2: The output status of the PIO port is 0—— low level and 1—— high level

The HC-05 Bluetooth module provides users with PIO port resources: PIO0~PIO7 and PIO10, which can be used to expand input and output ports.

illustrate:

1、 PIO10 The port outputs a high level

AT+PI0=10，1\r\n

OK
2、 PIO10 The port output high level AT+PIO=10,0\r\n
OK

18. Set up the multi-port output of PIO:

Instruct	Respond	Parameter
AT+MPIO=<Param>	OK	Param: Pio port number mask combination (decimal number)

The HC-05 Bluetooth module provides the user with PIO port resources: PIO0~PIO7 and PIO10, which can be used to expand input and output ports.

PIO Port number mask = (1 <<port number)
PIO port number mask combination = (PIO port number mask 1|PIO port number mask 2|.....)

in compliance with:
PIO2 port mask = (1 <<2) =0x004
PIO10 Port mask = (1 <<10) =0x400
PIO2 and PIO10 port mask combination = (0x004|0x400) =0x404
illustrate:
1. The PIO10 and PIO2 ports output high level
AT+MPIO=404\r\n
OK
2. The PIO4 port outputs a high level
AT+PIO=004\r\n
OK
3. PIO10, port output high level
AT+PIO=400\r\n
OK
5. All ports output low level
AT+MPIO=0\r\n
OK

19. Query the INPUT of the PIO port

Instruct	Respond	Parameter
AT+MPIO？	+MPIO：<Param> OK	Param——PIO port value (16 bits) Param[0]=PI00 Param[1]=PI01 Param[2]=PI02 ..... Param[10]=PI010 Param[11]=PI011

The HC-05 Bluetooth module provides users with PIO port resources: PIO0~PIO7 and PIO10~PIO11, which can be used to expand input and output ports.

20. Settings/Query-Paging scan, query scan parameters:

Instruct	Respond	Parameter
AT+IPSCAN=<Param1>, <Param2>, <Param3>, <Param4> AT+IPSCAN？	OK +IPSCAN: <Param1>, <Param2>, <Param3>, <Param4> OK	Param1: Query time interval Param2: Query duration Param3: Paging time interval Param4: The duration of the call is a decimal number. Default value: 1024,512,1024,512

illustrate:

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

21. Settings/queries-SHIFF

Energy saving parameters:

Instruct	Respond	Parameter
AT+SNIFF=<Param1>, <Param2>, <Param3>, <Param4>	OK	Param1: Maximum time Param2: Minimum time Param3: Try time Param4: The above parameters are all decimal numbers. Default value: 0,0,0,0
AT+IPSCAN?	+SNIFF: <Param1>, <Param2>, <Param3>, <Param4>	

22. Set/Query security and encryption mode:

Instruct	Respond	Parameter
AT+SENM=<Param>,<Param2>,	1、 OK—— success 2、 FAIL—— be defeated	Param: Security mode, which is as follows: 0——sec_mode0+off 1——sec_mode1+ non_secure 2——sec_mode2_service 3——sec_mode3_link 4——sec_mode_ unknown Param2 encryption mode, the value is as follows: 0——hci_enc_mode_off 1——hci_enc_ mode_pt_to_pt 2——hci_enc_mode_pt_ to_pt_and_bcast default value: 0,0
AT+ SENM?	+ SENM :<Param>,<Param2>, OK	

23. Remove the specified authentication device (Authenticated Device) from the Bluetooth pairing list:

Instruct	Respond	Parameter
AT+PMSAD=<Param>	OK	Param: Bluetooth device address

illustrate:

Remove the device with bluetooth address 12:34:56:
ab:cd:ef from the pairing list at+rmsad=1234,56,
abcdef\r\n
OK—— The deletion was successful
perhaps
at+rmsad=1234,56,abcdef\r\n
FAIL—— There is no Bluetooth device with the address
12:34:56:ab:cd:ef in the pairing list

24. Remove all authenticated devices (Authenticated Device) from the Bluetooth pairing list:

Instruct	Respond	Parameter
AT+RMAAD	OK	Not have

illustrate:

Remove all Bluetooth devices from the pairing list
at+rmaad\r\n
OK

25. Find the specified authentication device (Authenticated Device) from the Bluetooth pairing list:

Instruct	Respond	Parameter

AT+FSAD=<Param>	1、 OK—— success 2、 FAIL—— be defeated	Param: Bluetooth device address

illustrate:
Find the Bluetooth device from the pairing list: 12:34:56:ab:cd:ef at+fsad=1234,56, abcdef\r\n
OK—— There is a Bluetooth device with the address 12:34:56:ab:cd:ef in the pairing list.
at+fsad=1234,56,abcde0\r\n
FAIL—— There is no Bluetooth device 12:34:56:ab:cd:e0 in the pairing list.

26. Get the number of authenticated devices in the Bluetooth pairing list (Authenticated Device Count):

Instruct	Respond	Parameter
AT+ADCN?	+ADCN：<Param> OK	Param: The number of Bluetooth devices in the pairing list

illustrate:
at+adcn?
+ADCN: 0—— There is no Bluetooth device in the paired trust list
OK

27. Obtain the address (Most Recently Used Authenticated Device) of the Bluetooth authentication device recently used:

Instruct	Respond	Parameter
AT+MRAD?	+ MRAD ：<Param> OK	Param: The address of the Bluetooth device recently used

illustrate:
at+mrad?
+MRAD: 0: 0: 0 —— No recent use of trusted Bluetooth devices OK

28. Obtain the working status of the Bluetooth module:

Instruct	Respond	Parameter
AT+STATE?	+ STATE：<Param> OK	Param: Module working status The return values are as follows: "INITIALIZED" --Initialization state "READY" --Ready state "PAIRABLE" --Pairable state "PAIRED" --Paired state "INQUIRING" --Inquiry state "CONNECTING" --Connecting state "CONNECTED" --Connected state "DISCONNECTED" --Disconnected state "NUKNOW" --Unknown state

illustrate:
at+state?
= STATE: INITIALIZED—— Initialization state
OK

29. Initialize the SPP specification library (Initialise the spp profile lib):

Instruct	Respond	Parameter
AT+INIT	1、 OK—— success 2、 FAIL—— be defeated	Not have

30. Query Bluetooth devices

Instruct	Respond	Parameter

AT+INQ	+INQ: <Param1>,<Param2>,<Param3>, …… OK	Param1: Bluetooth address Param2: Equipment Param3: RSSI signal strength

Example 1:

at+init\r\n —— Initialize the SPP library (do not initialize repeatedly)
OK
at+iac=9e8b33\r\n —— Query bluetooth devices with any access code
OK
at+class=0\r\n —— Query various Bluetooth device classes
at+inqm=1,9,48 —— Query mode: with RSSI signal strength indicator, the query is terminated when more than 9 Bluetooth devices respond, and the timeout is set to 48x1.28= 61.44 seconds.
At+inq —— Query nearby Bluetooth devices
+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

Example 2:

at+iac=9e8b33\r\n —— Query any Bluetooth device with access code
OK
at+class=1f1f\r\n —— Query a Bluetooth device with class 0x1f1f
OK
at+inqm=1,9,48 —— Query mode: with RSSI signal strength indicator, the query is terminated when more than 9 Bluetooth devices respond, and the timeout is set to 48x1.28= 61.44 seconds.
At+inq —— Filter and query nearby Bluetooth devices
+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

Example 3:

at+iac=9e8b3f\r\n —— Query the Bluetooth device with access code 0x9e8b3f
OK
at+class=1f1f\r\n —— Query a Bluetooth device with class 0x1f1f
OK
at+inqm=1,1,20\r\n —— Query mode: with RSSI signal strength indicator, the query is terminated when more than one Bluetooth device responds, and the timeout is set to 20x 1.28=25.6 seconds.
At+inq —— Filter and query nearby Bluetooth devices
+INQ:1234:56:ABCDEF,1F1F,FFC2
OK

31. Cancel query of Bluetooth device:

Instruct	Respond	Parameter
AT+INQC	OK	Not have

32. Equipment pairing:

Instruct	Respond	Parameter
AT+PAIR=<Param1>,<Param2>	1、 OK—— success 2、 FAIL—— be defeated	Param1: Bluetooth address of remote device Param2: Connection timeout (seconds)

illustrate:

Pair with a remote Bluetooth device: 12:34:56:ab:cd:ef,
with a maximum pairing timeout of 20 seconds. At+pai=
1234,56,abcdef,20\r\n
OK

33. Equipment connection:

Instruct	Respond	Parameter
AT+LINK=<Param>	1、 OK—— success 2、 FAIL—— be defeated	Param: Bluetooth address of remote device

illustrate:
Connect to a remote Bluetooth device: 12:34:56:ab:cd:ef
at+fsad=1234,56, abcdef\r\n—— query bluetooth device 12:34:56:ab:cd:ef whether it is in the pairing list
OK
at+link=1234,56,abcdef\r\n—— Query Bluetooth device 12:34:56:ab:cd:ef In the pairing list, you can connect directly without querying. OK

34. Disconnect the connection

Instruct	Respond	Parameter
AT+DISC	1. +DISC: SUCCESS—— The connection is successfully disconnected OK 2, +DISC: LINK_LOSS—— connection lost OK 3. +DISC: NO_SLC—— No SLC connection OK 4, +DISC: TIMEOUT—— Disconnection timeout OK 5, +DISC: ERROR —— Disconnected error OK	Not have

35. Enter energy saving mode:

Instruct	Respond	Parameter
AT+ENSNIFF=<Param>	OK	Param: Bluetooth address of the device

Exit energy saving mode

Instruct	Respond	Parameter
AT+EXSNIFF=<Param>	OK	Param: Bluetooth address of the device

Appendix 1: AT instruction error code description

Error code return form ——ERROR: (error_code)

error_code (hexadecimal number)	Explanatory note
0	The AT command is incorrect

1	The command result is the default value
2	PSKEY Write errors
3	The device name is too long (more than 32 bytes)
4	The length of the device name is zero
5	Bluetooth address: NAP is too long
6	Bluetooth address: UAP is too long
7	Bluetooth address: LAP is too long
8	The length of the OPIO sequence mask is zero
9	There are a lot of IOU numbers
A	The length of the device is zero
B	The device number is too long
C	The query access code length is zero
D	The query access code number is too long
E	Invalid query access code
F	The length of the paired code is zero
10	The pairing code is too long (more than 16 bytes)
11	The module role is invalid
12	The波特率无效.
13	Stop position invalid
14	The check digit is invalid
15	There is no authentication device in the pairing list
16	The SPP library was not initialized
17	The SPP library was reinitialized
18	Invalid query mode
19	The query timeout is too large
1A	The Bluetooth address is zero
1B	Invalid safety mode
1C	Invalid encryption mode

Appendix 2: Equipment description

The Class of Device/Service(COD) is a 32 bifs number that of 3 field specifies the service supported by the device. Another field spcifies theminor device class, which descrbes 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(mulfiple 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 krder 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,Misrophone,…)
20 0bject Transfer (v-Inbox, v-Folder,…)
21 Audio (Speaker,Microphone,Headset service,…)
22 Telephony (Cordless telephony, Modem, Headset service,…)
23 Imformation (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 Table1.3.
1 2 1 1 1 0 9 8 Major Device Class
0 0 0 0 0 Miscel laneous [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 Periphereal (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: MAJOE DEVICE CLASSES
[Ref #2:Used where a more specific Major Device Class is not suited (but only as

specifiedas 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 decvice 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 nof 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(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

TABLE1.5: SUB DEVICE CLASSES FOR THE' PHONE' MAJOR CLASS

6. MINOR DEVICE CLASS FIELD – LAN/NETWORK ACCESS POINE 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
TABLE1.6: THE LAN/NETWORK ACCESS POINE 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
TABLE1.7:RESERVED SUB-FIELD FOR THE LAN/NETWORK ACCESS POINE
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]
X X X X X X 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/VIOEO’ 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 Combo 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 combo mouse/keyboard devices. These may be combined with the lower bits in a multifunctional decice.
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 TMAGING 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

Appendix 3: Description of query access code (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.