



# **UART interface BT controller initial guide**

## **Version 1.3**

Realtek Semiconductor Corporation

All Rights reserved

No. 2, Innovation Road II, Hsinchu Science Park,

Hsinchu 300, Taiwan

[www.realtek.com](http://www.realtek.com)

## Revision History

Revision	Date	Change Description
V0.6	2013/11/15	Draft by gordon
V1.0	2013/12/17	Initial Release by ally
V1.1	2014/4/1	Remove patch structure by champion
V1.2	2015/4/17	Config file example changes according 8761 by mike
V1.3	2015/6/9	Add low power mode and wakeup function by champion

# 目录

Revision History.....	2
目录 .....	3
1 概述 .....	4
1.1 供电和 pin 脚定义 .....	4
1.2 UART 设定.....	4
1.2.1 UART 参数.....	4
1.2.2 efuse 定义.....	5
1.2.3 波特率设定.....	6
1.2.4 UART RTS/CTS 接法.....	6
1.3 PCM 设定 .....	7
1.4 32K clock .....	7
1.4.1 efuse definition .....	7
1.4.2 external/internal 设定 .....	8
1.5 生成 patch code.....	8
1.6 configure file .....	8
1.6.1 Configure file 结构.....	8
1.6.2 Configure file 例子.....	9
2 初始化过程.....	11
2.1 Link establishment .....	12
2.1.1 发送 sync packet。.....	12
2.1.2 发送 config packet .....	12
2.1.3 发送 pure ack.....	13
2.2 确定卡片型号.....	13
2.3 load patch code 和 configure file.....	14
2.4 Change baud rate .....	14
2.5 Download patch code and configure file .....	15
2.5.1 HCI_VENDOR_DOWNLOAD command.....	15
2.5.2 Download procedure.....	16
2.5.3 Check download success .....	16
2.6 Change Hardware Flow Control .....	17
3. 低功耗模式及唤醒功能.....	17
3.1 蓝牙低功耗模式.....	17
3.2 主机唤醒蓝牙.....	18
3.3 蓝牙唤醒主机.....	18

# 1 概述

本文档主要介绍 Realtek UART interface (H5) 蓝牙芯片的初始化流程，供第三方开发参考使用。

## 1.1 供电和 pin 脚定义

这部分内容请参考 HW datasheet。

需要注意的是 BT\_RST pin (EN\_CHIP/BT\_DIS#):

Host 在供电状态下可以通过操作 BT\_RST pin 进行 disable/enable BT 的动作:

1. 拉低 BT\_RST pin, **1s** 后, BT controller 才能真正进入 power down 模式;
2. 拉高 BT\_RST pin, **300ms** 后, BT controller 初始化过程结束, 可以接受 sync packet。

## 1.2 UART 设定

### 1.2.1 UART 参数

BT controller 上电后使用默认的 UART 参数, host 需要根据芯片的设定来初始化平台的 UART 参数, 完成 sync 过程后, 再根据需求通过 vendor command 和 configure file 设定新的 UART 参数。

其中可以修改的参数有:

1. 波特率:
  - 1) 在 sync 完成后即可以根据需求通过 vendor command 修改;
  - 2) Controller 在收到最后一个 download patch command 之后, 回复 command complete event 之前会修改 baudrate 为 configure file 中设定的值;
  - 3) Configure file 中需要设定与 vendor command 相同的 baud rate 值。
2. Hardware flow control: 通过 configure file 修改。
  - 1) Controller 在收到最后一个 download patch command 之后, 回复 command complete event 之前修改为 configure file 中的设定。
  - 2) 建议在 sync 和 download patch code 的过程中关闭 host 端 HW flow control, 在发送最后一笔 download patch code command 后再根据 configure file 的配置决定是否打开 host 端的 flow control。
  - 3) 为了保证传输的效率, 建议 download patch 结束后同时打开 host 和 controller 的 HW flow control。

其他参数不建议修改, 直接使用 BT controller 的默认值即可, 如果因为特殊需求必须修改, 建议联系我们修改 efuse 设定。

为了保证传输的效率, 建议采用 H5 protocol, 打开 flow control。

UART 主要参数见下表。

表 1：UART 参数

Parameters	Options
Protocol	H4 or H5
Flow control	On or Off
Parity	None\even\odd
Number of data bits	8
Stop bit	1 stop bit
Baud rate	115200, ..., 300000 bps

BT controller UART interface 默认参数见下表，具体设定可能会有差异，请向芯片提供者确认或者通过读 efuse 确认(参看 1.2.2)。

表 2: Realtek BT controller UART interface 默认参数

芯片型号	Baud rate	Protocol	Parity	Stop bit	Hardware Flow Control
RTL8723A	115200	H5	Even	1	Disable
RTL8723B	115200	H5	Even	1	Disable
RTL8761AT	115200	H4	None	1	Disable
RTL8761ATL	115200	H5	Even	1	Disable

## 1.2.2 efuse 定义

1. 波特率：0x0c ~ 0x0f (4 bytes, 具体设定参看 1.2.3)
2. parity 和 hardware flow control: 0x18 ~ 0x 1b
  - Bit[0] : parity\_en
    - If set to 1, enable parity.
    - If set to 0, disable parity.
  - Bit[1] : parity\_even
    - Set 1 to enable even parity
  - Bit[2] : hw\_fctrl\_on
    - If set to 1, enable hardware flow control.
    - If set to 0, disable hardware flow control.

### 1.2.3 波特率设定

Host 和 controller 实际工作的波特率可能与设定值都有一定的偏差，为了保证能正常工作，建议 host 和 controller 之间实际波特率误差小于 3%。

以下误差百分比为负数表示实际波特率比设定值偏小，误差百分比为正数表示实际波特率比设定值偏大。

#### 1. 8723A

8723AS UART Baudrate Settings	0x0C	0x0D	0x0E	0x0F	Baudrate Error (%)
115200	1D	70	00	00	-0.22%
230400	10	60	00	00	-1.36%
460800	08	60	00	00	-1.36%
921600	04	60	00	00	-1.36%
2000000	02	50	00	00	+0%
3000000	01	80	00	00	+2.56%
3250000	01	70	00	00	+2.56%
4000000	01	50	00	00	+0%

#### 2. 8723B\8703A\8821A\8761A 等

具体设定参看 **RTL8761A\_FineTune\_BaudrateTable.txt** 文档。

### 1.2.4 UART RTS/CTS 接法

- 为了保证数据传输的效率，建议 HOST\_UART\_RTS\_OUT 接到 BT\_UART\_CTS\_IN，同时打开 HW flow control。
- HOST\_UART\_CTS\_IN，可以经由电阻接 LOW（避免漏电，并让 UART 可以正常运作），若 HOST\_UART\_CTS\_IN 永远是 Input Pin，可以接 GND。
- BT\_UART\_RTS\_OUT 不要接到 HOST\_UART\_CTS\_IN。
  - 尽量使用 BT\_UART\_RTS\_OUT 当做是 BT 32K Clock Input pin（参看 1.4）。
    - Input Voltage Range: VIO\_UART~3.3V
    - 3.3V Tolerance，不会有漏电
  - 当不使用时，空接，不要接 GND。

下图是使用 external 32K 的接法，如果使用 internal 32K，则 BT\_UART\_RTS\_OUT 空接，其它不变。

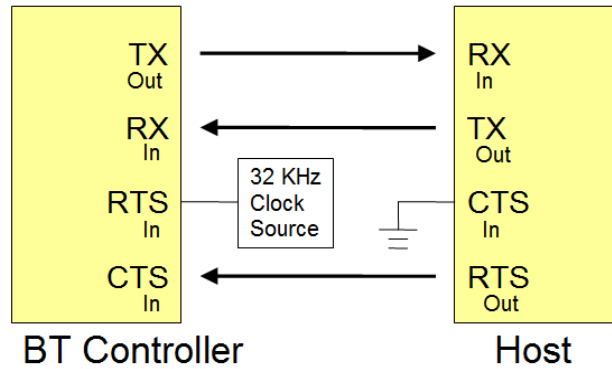


图 1.2.1 UART interface 连接示意图

## 1.3 PCM 设定

BT controller 的 PCM 参数需要根据 host 参数在 configure file（参看 1.6 节）中设定，参数定义参看 **PCM setting\_Description\_v03.pdf** 文档，如果需要，也可以将 host 参数告诉我们由我们计算 controller 的设定。

## 1.4 32K clock

如果 host 平台有可用的 32K clock，建议 BT controller 使用 external 32K clock（设定参看 1.4.1），并且使用平台上的 32K clock，否则使用 internal 32K。

### 1.4.1 efuse definition

1. 8723A
  - 32K clock 选择: WIFI efuse 0x0B[7]
    - WIFI efuse 0x0B[7] = 1, enable UART\_RTS pin as 32k clock input pin.
    - WIFI efuse 0x0B[7] = 0, UART\_RTS pin is output pin.
2. 8723B/8761A/8821A 等
  - 32K clock 选择: 0x27[7]
    - 0: internal
    - 1: external
  - BT\_UART\_RTS pin direction: 0x1E3[0]
    - 0: output
    - 1: input
  - Select low power clock: OXFE
    - 1: 32KHz
    - 0: 32.768KHz

## 1.4.2 external/internal 设定

1. 0x27[7] = 0, 0x13e[0] = 0: internal 32K, UART\_RTS pin is output.
  2. 0x27[7] = 1, 0x13e[0] = 1: external 32K, 32k clock input pin is UART\_PIN.
- Efuse 设定要与平台硬件连接保持一致，参看 1.2.4。

## 1.5 生成 patch code

目前 Realtek 的蓝牙芯片，都需要在初始化的时候加载 patch code。  
使用我们提供的 patch code 再加上后面的 config 文件一起加载给芯片。

## 1.6 configure file

Realtek BT controller 支持通过 config 文件动态调整 HW 设定（configure 文件直接拷贝到 patch code 之后再一起 download 即可）。

通常会使用 configure file 修改如下参数：

1. 蓝牙地址；
2. UART 参数；
3. PCM 设定；
4. 32K clock 选择，要与平台硬件连线对应。

如果有其他特殊要求，可以直接联系我们。

### 1.6.1 Configure file 结构

Configure 文件包含两个 section: Signature Section 和 data section。

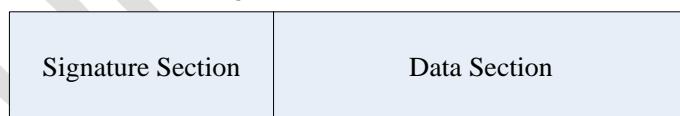


图 1.6.1 configure file format

1. signature section



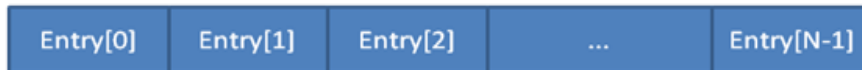
- Signature Field (4 octets)
  - Must be 0x8723ab55
  - If the signature field is expected, data\_lenth field and data section are valid.
- Data\_Length Field (2 octets)



- The total length of Data Section, exclude length of Signature and Data\_Length.

## 2. data section

- Composed by many entries



- The structure of one entry

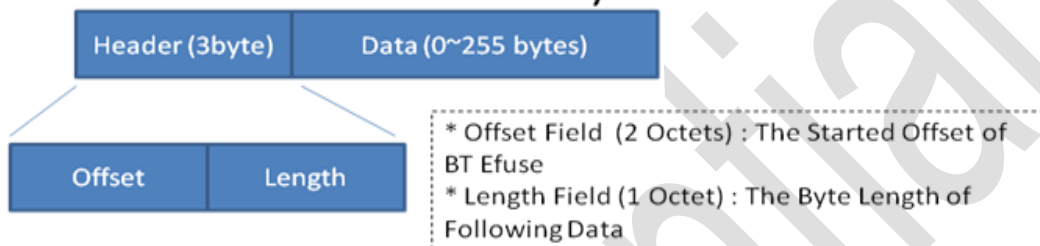


图 1.6.2 data format

## 1.6.2 Configure file 例子

以 8761a 卡片的 config 为例。

00000000h:	55 AB 23 87	31 00	F4 00 08 01 00 00 00 05 50 00
00000010h:	00 DC 00 10 02 80 92 04 50 C5 EA 19 E1 1B F1 AF		
00000020h:	5F 01 A4 0B	27 00 01 67	FE 00 01 01 5B 01 04 0B
00000030h:	0B 0B 0A	E3 01 01 00	

1. Data length: 0x31
2. PCM setting (具体内容参看 **PCM setting\_Description\_v03.pdf**)  
f4 00 08 01 00 00 00 05 50 00 00
3. UART baudrate and setting: (具体内容参看 **RTL8761A\_FineTune\_BaudrateTable.txt**)  
0c 00 10 ...
4. 32k clock (efuse 具体定义参看 1.4)
  - External 32K
    - external 32k: 27 00 01 E7
    - BT low power clock : FE 00 01 00
    - UART\_RTS pin direction: E3 01 01 01
  - Internal 32K
    - Internal 32k: 27 00 01 67
    - BT low power clock : FE 00 01 01

- UART\_RTS pin direction: E3 01 01 00

Confidential

## 2 初始化过程

初始化过程如下，host 在初始化结束时改变自身 hardware flow control 设置之前，一直处于 flow control disable 状态，并且 HOST\_UART\_RTS\_OUT 要设为低电平。

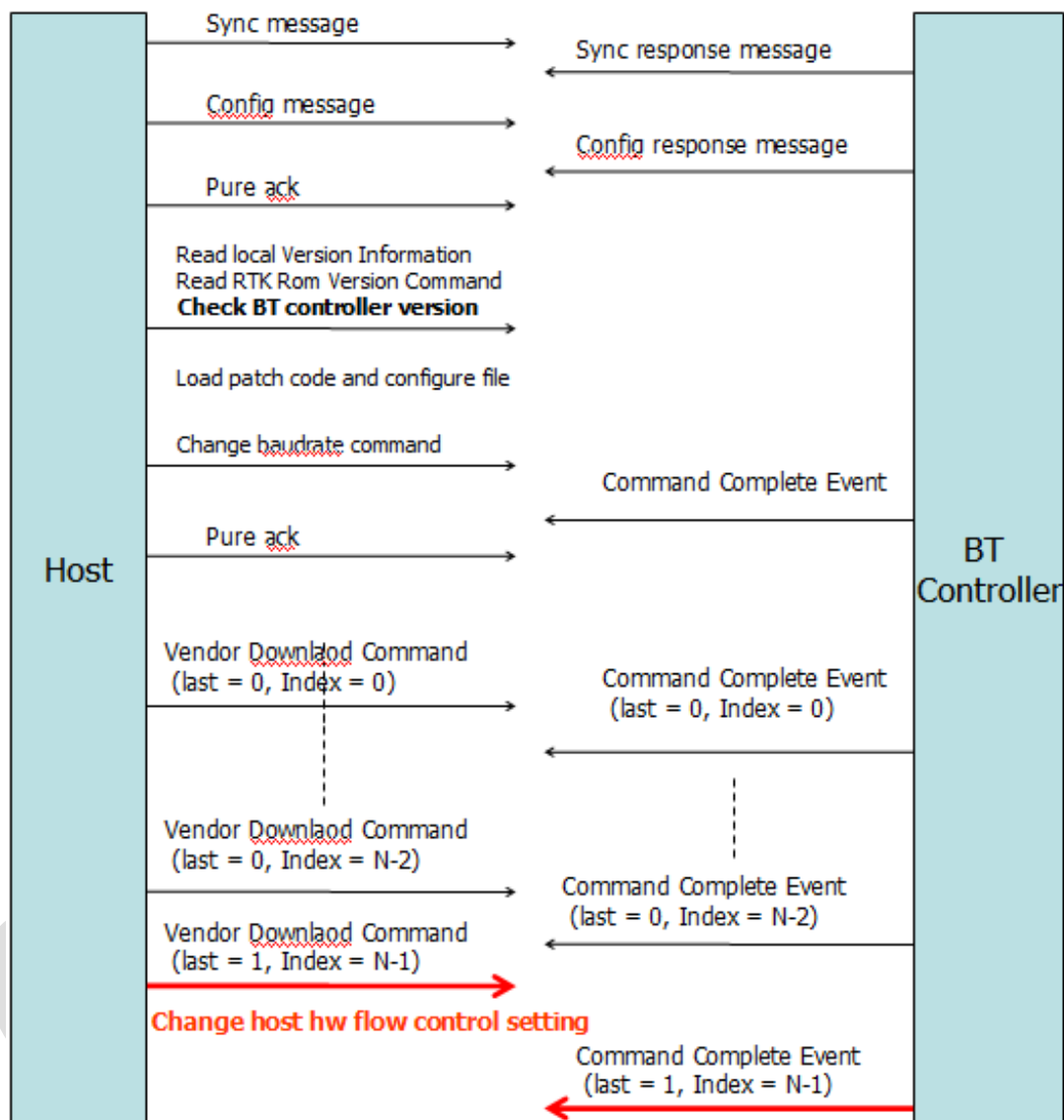


图 2.1 初始化过程

## 2.1 Link establishment

### 2.1.1 发送 sync packet。

sync 包的发送是 UART interface 初始化的第一步，sync packet 格式如下：

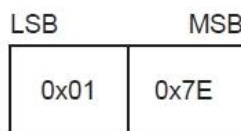


图 2.1.1 sync message format

需要注意的是在芯片上电后，需要等待一定的时间（约 300ms）才能保证正常接收 sync packet，所以可能会出现开始的几个 sync 包无响应的现象，因此建议设定重传和 timeout 机制：

1. 在发送 sync packet 150ms 内等不到 response 时，重新发送 sync packet；
2. 设定发送 sync packet 的 timeout 时间（5s），超时仍然收不到正确的 response packet，即认为 sync packet 发送失败，可能是平台其他问题引起的，请重新检查硬件设定和初始化设定。

Sync response packet 如下图所示，收到该 response 后，即可认为 sync 包发送成功。

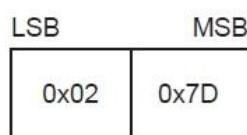


图 2.1.2 sync response message format

如果采用 UART H5，编码后的 packet 如下：

1. Sync message: 0xc0, 0x00, 0x2f, 0x00, 0xd0, 0x01, 0x7e, 0xc0。
2. Sync response message: 0xc0, 0x00, 0x2f, 0x00, 0xd0, 0x02, 0x7d, 0xc0。

### 2.1.2 发送 config packet

在 sync 成功之后，需要继续发送 config packet，处理流程与 sync 包的处理流程类似。格式见图 2.1.3。建议仍然实现 retry 和 timeout 机制，当然，通常来说不会发生这种情况。Config response 格式见图 2.1.4。

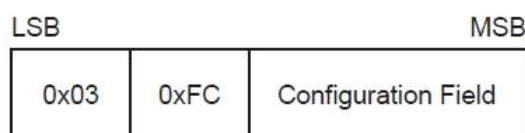


图 2.1.3 config message format

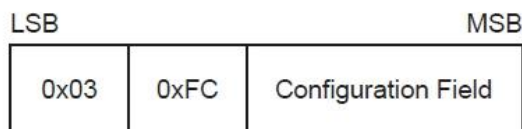


图 2.1.4 config response message format

### 2.1.3 发送 pure ack

在上述两个操作结束后，建议 host 发送一个 pure ack packet，告诉 BT controller，host 收到了 configure response packet。

## 2.2 确定卡片型号

在 download patch code 之前，Realtek 蓝牙芯片的 default Local Version Information 如下表，download patch code 之后，LMP subversion 和 HCI revision 会发生变化，并且每一版 patch code 都会不同。

表 4：默认 Local version Information

芯片型号	LMP Subversion	HCI Revision
RTL8723A	0X1200	0X000B
RTL8723B	0X8723	0X000B
RTL8761A	0X8761	0X000A
RTL8821A	0X8821	0X000A

Host 在 download patch code 之前需要检查如下信息 Patch code 是否存在：如果 patch code 已经存在，则不需要重复 download。

HCI\_Read\_Local\_Version\_Information:

- 1) 根据 LMP Subversion 和 HCI Revision 确定芯片型号（参看表 4）。
- 2) 如果读到的 LMP Subversion 和 HCI Revision 不是表 4 中的任何一个，则可以认为 patch code 已经存在，不需要重复 download。

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Version_Information	0x0001		Status, HCI Version, HCI Revision, LMP Version, Manufacturer_Name, LMP Subversion

## 2.3 load patch code 和 configure file

Host 分别从文件中读出 patch code 和 configure file，组成一个新的文件（结构如下图，最大不超过 24K），通过 HCI vendor command 下载到 BT controller（参看 2.5）。

一般来说，在 download 之前需要从其他地方读取 BD address，并替换 configure file 中的 BD address，否则，download 相同 configure file 的设备会出现 BD address 相同的情况。

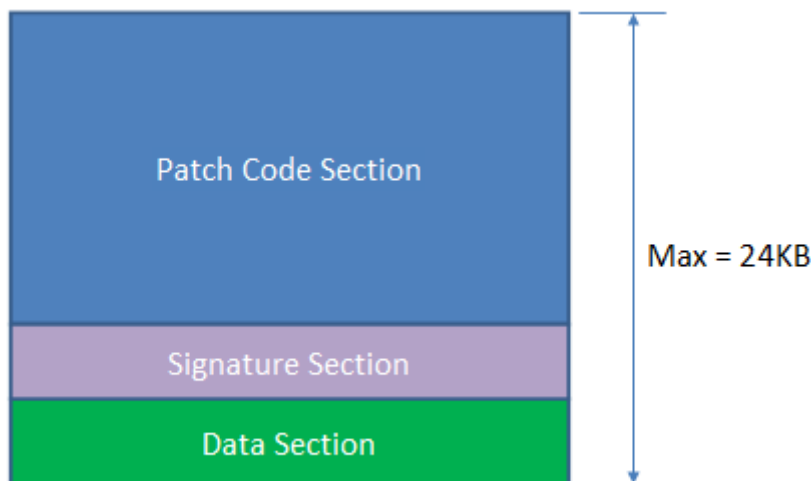


图 2.3.1 patch code 和 configure file 结构

## 2.4 Change baud rate

建议在 download patch code 之前先将 host 和 controller 的波特率统一修改成 configure file 中的设定值。

Command	Opcode	OCF	OGF	Parameter length	Parameter	Return Parameter
Set Baudrate	0XFC17	0X0017	0X3F	4 byte	Baudrate	Status

Baudrate 设定参看 1.2.3。

BT controller 收到这个 command 之后，会以旧的 baudrate rate 回应 command complete event，host 收到 event 后，仍然以旧的 baudrate 发送 pure ack，BT controller 收到 pure ack 后才会修改到新的波特率。在收到 pure ack 之前，BT controller 会每隔一段时间发送一个 command complete event 给 host（时间间隔小于 250ms）。因此，host 可以根据是否再收到 command complete event 来判断 BT controller 是否收到 pure ack，也就是是否已经改变波特率了。

Host 在确认 BT controller 波特率已经改变后，需要同时改变自己的波特率，从而才能够继续通信。

## 2.5 Download patch code and configure file

使用 HCI vendor command 完成 download patch 的功能。

### 2.5.1 HCI\_VENDOR\_DOWNLOAD command

Command	Opcode	Command Parameters	Return Parameters
HCI_VENDOR_DOWNLOAD	0XFC20	Index, Data[]	Status, Index

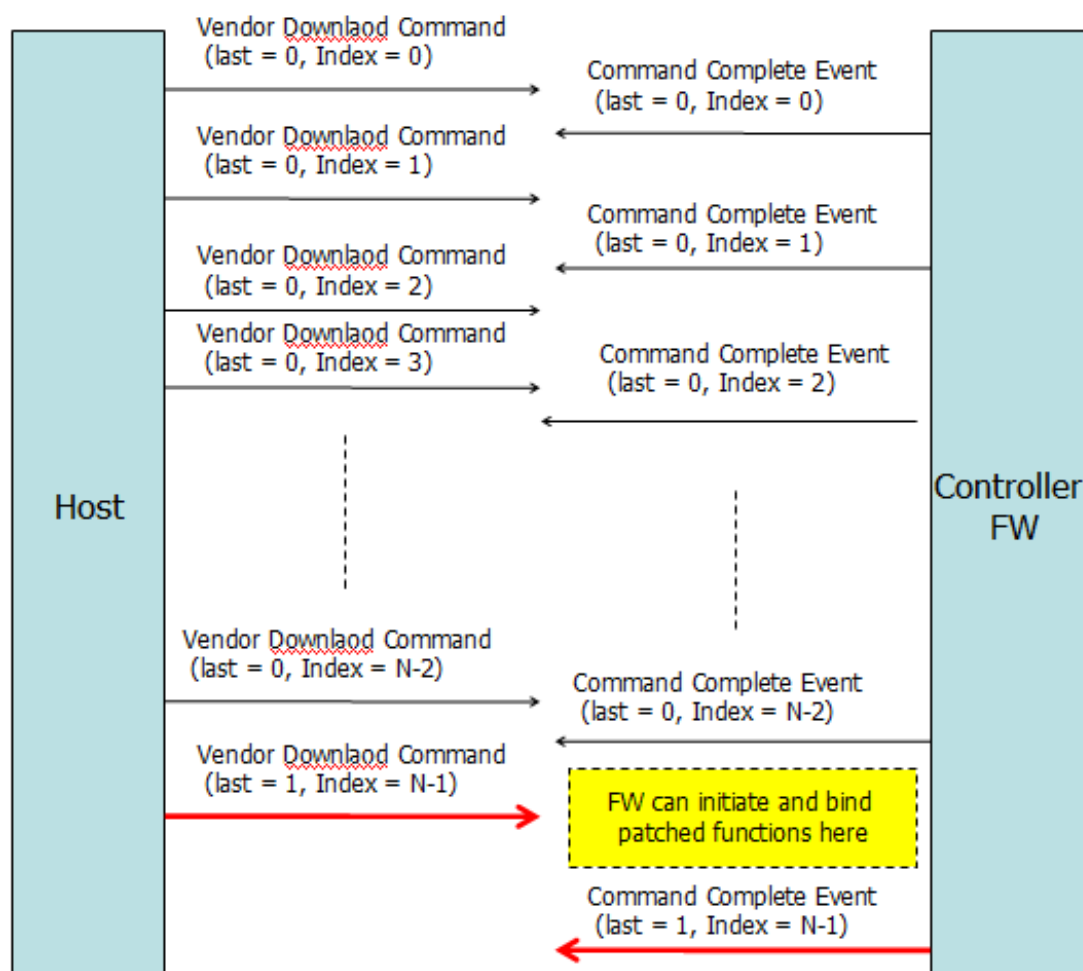
#### 1. Command Parameters

- Index (1 Octet)
  - bit[7]: 1 is the last and 0 is the start/continuous block
  - Bit[6:0]: the sequence number. The number shall be incremented by 1 after host send each commands to the controller. The reset value is 0.
- Data (4N Octets)
  - The length is the multiple of 4 bytes. If it is not the last block, host may sends 252 byte in data field to reduce the time in download procedure.

#### 2. Return Parameters:

- Status (1 Octet): 0x00 means Vendor Download command succeed. Otherwise, mean failed
- Index (1 Octet): Return the index from received Vendor Download command

## 2.5.2 Download procedure



在 download 的过程中有以下几点需要注意：

1. 在 BT controller 上电后，host 端发送第一个 HCI command 前必须完成 Patch code download 过程。
2. Download 过程中不能发送其他的 HCI command。
3. Host 发送最后一个 download command (last = 1, Index = N-1) 后，BT controller 需要根据 patch code 和 configure file 的设定重新配置 controller，因此需要等待一段时间（300ms）才能回复 command complete event。BT controller 是在新的配置状态回复 command complete event。

## 2.5.3 Check download success

通过以下两点可以确认 patch code 已经 download 成功。



1. 收到 controller 回复的最后一个 command complete event (last = 1 and index = N-1)。
2. LMP Subversion 和 HCI Revision 已经非初始值。

## 2.6 Change Hardware Flow Control

在 download patch code 完成之前, host 一直是 hardware flow control disable 状态, BT controller 收到最后一个 download command (last = 1, Index = N -1) 后, 会根据 configure file 的设定重新配置 controller (比如是否打开 Hardware Flow Control, PCM setting, BD address 等), 发送最后一个 command complete event 时已经工作在新的配置状态。所以 host 在发送完最后一个 download patch command 之后, 收到 command complete event 之前, 需要根据 configure file 的设定修改 host uart 的 Hardware Flow Control 设定。

至此, 蓝牙的初始化过程结束。

## 3. 低功耗模式及唤醒功能

### 3.1 蓝牙低功耗模式

蓝牙 controller 进入低功耗模式 (底电流<1mA) 必须有特定版本的固件支持, 且符合以下条件 controller 会自动进入低功耗模式, 不需要 host 进行设置

#### 1. Non-link mode

- 1) 放 6s 之后就会自动进入 LPS。
- 2) Page Scan、Inquiry Scan、LE Adv、LE Scan 这四个状态下的 interval 不能小于 0x80。

#### 2. Link mode

- 1) Sniff、Page Scan、Inquiry Scan、LE Adv、LE Scan 这四个状态下的 interval 不能小于 0x80。
- 2) 两条 Link 不能够进入 LPS。
- 3) 有 eSCO、SCO 不会进入 LPS。
- 4) 先进入 Sniff Mode 才能进入 LPS, 如果 Sniff Mode 进不去就不会进 LPS, Sniff 参数建议:  $N_{\text{sniff attempt}}$  设置为大于 1 的值(Sniff Mode Command)。

#### 3. LE specific

- 1) LE master/slave: Connection Interval (conn\_interval)不能小于 0x80, 否则也不能进入 LPS。
- 2) LE master: Connection Interval (conn\_interval)、 Initiator Scan Interval(LE\_Scan\_Interval in LE Create Connection Command)也不能小于 0x80, 否则不能进入 LPS。

#### 4. HCI commands/Events

- 1) Read Inquiry Scan Activity Command
- 2) Write Inquiry Scan Activity Command
- 3) Read Page Scan Activity Command

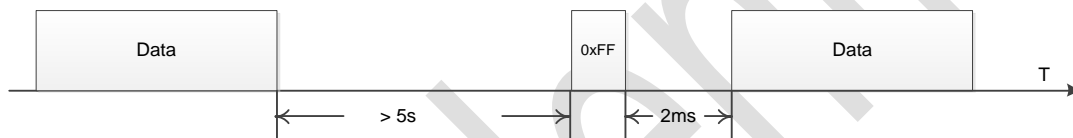
- 4) Write Page Scan Activity Command
- 5) LE Set Advertising Parameters Command
- 6) LE Set Scan Parameters Command
- 7) Sniff Mode Command ( $N_{\text{sniff attempt}}$ )
- 8) Mode Change Event: Return the actual sniff interval in the Interval parameter
- 9) LE Create Connection Command(Initiator Scan Interval)

(基本上现在提供的正式版 patch 都是给 40M 使用，26M 要额外提供。)

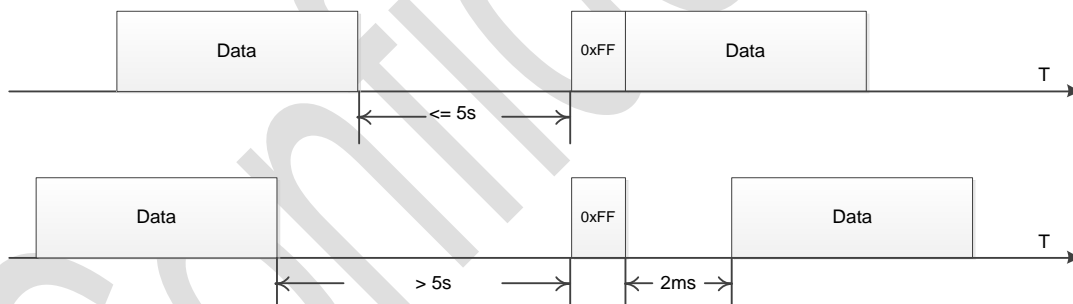
## 3.2 主机唤醒蓝牙

蓝牙 controller 支持通过 UART inbandside signal 唤醒，只要 host 发包给 controller，controller 会自动醒来，但是由于 8761AT 醒来需要时间，H4 没有重传机制，可能存在小概率丢包，请 host 实现以下机制确认不会丢包

**Non-link mode**：每次发包前都会丢 0xFF，如果和上一次发包间隔大于 5s，发完 0xff 之后 Delay 2ms



**Link mode**：进入 sniff mode 后，只要 host 有数据要送都会先丢 0xff，如果和上一次发包间隔大于 5s，发完 0xff 之后 Delay 2ms



## 3.3 蓝牙唤醒主机

主机进入睡眠状态，通过蓝牙唤醒。

配置 Pin25 (BT\_WAKE\_HOST) 作为 Wakeup output pin

0xDF, 0x01, 0x01, 0x41, 0xE2, 0x01, 0x01, 0x50, // wake up GPIO config

0xXX 0xXX: Offset

0xXX: Length

0xXX, ...: Values

controller 可以通过判断 2 个 event 间隔大于 2 秒就在发包之前去拉一下 BT\_WAKE\_HOST 来实现唤醒主机，并且 delay 1ms 发送 event 给 host 防止丢包

