Flairmesh Bluetooth Application Interface (BAI) Messages Reference

Ver.2.2
Jun. 5, 2025

Release Record

Version	Release Date	Comments
1.0	Feb. 25, 2022	First release.
1.1	Jul. 18, 2022	Add simple AG profile.
1.2	Nov. 11, 2022	Add HFP profile. Fix the error of start index of SPP bypass channel.
1.3	Nov. 30, 2022	Add new message “RO” for switching communication roles for SPP connection. Update the description of “IQ” and add new message “FN” for device name. Update the description of the new “MT” parameters.
1.4	Mar. 1, 2023	Add new messages of “PD” “SV” “MI” and “GV”.
1.5	Mar. 19, 2023	Add new messages of broadcast.
1.6	Mar. 21, 2023	Update messages of broadcast, add broadcast group.
1.7	Mar. 24, 2023	Add transceiver mode of broadcast.
1.8	Apr. 11, 2024	Add “PN” for legacy pairing.
1.9	Jul.4, 2024	Add messages of the GATT client functions under Auracast Broadcast Mode for FMA120.
2.0	Jul. 16, 2024	Add messages of the Auracast™ receiver function for FMB121.
2.1	Oct. 18, 2024	Add messages of the advertising SID and broadcast ID.
2.2	Jun. 5, 2025	Update messages of the Auracast™ receiver function for FMB121.

Table of Contents

Table of Tables

Table of Figures

No table of figures entries found.

1. Introduction

Flairmesh's Bluetooth Application Interface (BAI) is an end-to-end, easy to use ASCII format protocol supporting host application access to the functionality of Flairmesh's Modules. It supports configuring and accessing the IOs, digital peripherals on the module, controlling Bluetooth profiles running on the module and sending/receiving user's data to/from it. The underlying protocol data unit is a packet composed of octets with framing to permit transmission over stream-based connections such as UART, USB virtual COM and SPP-like GATT based Flairmesh property profile named "iGate". Except for the raw data field which has a leading length field, all numeric values are in hex string whose length depends on the data width. Textual strings in the command fields are always ASCII characters.

In this document:

The Host is the controller party, for example, an application running on a smartphone or a MCU connected to the UART of the module.

The Module is the controlled party, which is a Flairmesh module.

Messages are the packets exchanged between the Host and the Module. There're three types of messages.

- Commands are sent from the Host to the Module. In examples commands are shown with a "➔".
- Responses are sent from the Module to the Host to acknowledge received commands. In examples responses are shown with a "➔".
- Indicators are sent from the Module to the Host to notify events generated by the Module. In examples responses are shown with a "➔".

1.1. General message packet format

[BC:]{XX}[=NumericParam][,lengthOfRawData,RawData][,…][<CR><LF>]

Fields in [] is optional, while in {} is mandatory. [] and {} are only used to identify different fields and not included in the actual message packets.

Description of fields:

[BC:]: This optional three ASCII characters "BC:" command header is only used when the Host sends commands to the Module on UART/USB virtual COM. It is not used when a Mobile App sends commands via the "iGate" control channel. It is not used in responses and indicators. In this document, the command header is always shown in examples to make it clearer to readers.

{XX}: This mandatory field which always contains two ASCII characters identifies the message name, which might be individual command, response or Indicator.

[=NumericParam]: If a message has numeric parameters, the first parameter shall have a leading "=" and the following parameters shall have a leading "," to make it more readable to human.

[,lengthOfRawData,RawData]: This field contains raw byte data exchanged between the Host and the Module, lengthOfRawData is ASCII coded numeric hex values such as "A0" or "01BE" depending on the allowed data length.

[<CR><LF>]: This optional field contains two ASCII characters, which are the "carriage return" and "new line". This field is only used with messages on UART/USB virtual COM, it shall not be used on "iGate" control channel.

2. BAI command list

All the available BAI commands are listed and briefly described in the tables below. The detailed description of each command can be given in chapter 4.

Table 2.1 BAI Command List

Command	Short Description	Comments
General Commands		
PF	Query or configure the profiles of the module.	
AD	Query the Bluetooth address of the module.	
TP	Query or change the Tx Power of the module.	
CD	Query or configure the Class of Device of the module.	
FT	Query or configure the features of the module.	
MM	Query or configure Man-In-The-Middle protection feature.	
IO	Query or configure IO capability of local device.	
MT	Query or configure force to be master feature.	
SN	Query or configure the sniff mode.	
SP	Query or change the deep sleep mode.	
PN	Query or change the fixed pin code of the module.	
NM	Query or change the local friendly name of the module.	
IF	Query or change the host interface	
BR	Query or change the UART baud rate.	
UM	Query or change the UART mode.	
DB	Query or change the default bypass mode.	
IQ	Inquiry the Bluetooth device.	
MD	Query or change the state of discoverable mode.	
PA	Query or change the state of pairing mode.	
CA	Query or change the state of connectable mode.	
NC	Confirm or deny the numeric comparison.	
PK	Input the Passkey when pairing.	
PN	Query or change the fixed PIN used in legacy pairing.	
CP	Clear the paired Bluetooth device list.	
CT	Connect to remote Bluetooth device.	
DC	Disconnect with remote Bluetooth device.	
BP	Configure the runtime bypass mode.	
SPP Commands		
SM	Query or change the service name of SPP profile.	
CS	Connect to the remote SPP device.	
DS	Disconnect with the remote SPP device.	
SS	Query the SPP state of each SPP instance.	
DT	Send data packet to remote SPP device.	
RO	Query or change communication role of SPP connection.	
HID Commands		
CI	Connect to the remote HID host.	

DI	Disconnect with the remote HID host.	
IS	Query the HID state.	
KR	Send keyboard report to remote HID host.	
AS	Send ASCII string to remote HID host.	
OPP Commands		
CO	Connect to a remote OPPS device	
DO	Disconnect with the remote OPP device	
OS	Query the OPP state	
OA	OPPC push object file name	
OY	OPPC push object file type	
OT	OPPC push object packet data	
BLE/iGate Commands		
LU	Query or change the iGate service UUID128.	
LP	Query or change the BLE connection parameters.	
LB	Query or change the bondable state.	
DL	Disconnect with remote BLE central device.	
LS	Query or change the state of BLE connection.	
LD	Send data packet to remote BLE central device.	
HFP Commands		
CH	Connect to the remote HFP device.	
DH	Disconnect with the remote HFP device.	
HS	Query the state of HFP channel.	
AR	Accept or Reject call.	
HU	Hang Up call.	
TC	Transfer call.	
MU	Mute or unmute the microphone.	
LR	Last numbers redial.	
HV	Check or set the volume of HFP voice.	

3. BAI response and indicator list

All the available BAI indicators are listed and briefly described in the tables below. The detailed description of each command can be given in chapter 4.

Table 3.1 BAI Indicator List

Message	Short Description	Comments
General Responses and Indicators		
OK	Indicates a command was accepted by the module.	
ER	Indicates there is an error detected in the command sent by the host.	
AP	State of Module as an application.	
AD	Bluetooth address of the module.	
TP	Tx Power of the module	
CD	Class of Device of the module.	
PF	Configuration of profiles of the module.	

FT	Features of the module.	
MM	States of Man-In-The-Middle protection.	
IO	Configuration of IO capability of local device.	
MT	Configuration of force to be master feature.	
SN	Configuration of sniff feature.	
SP	The deep sleep state.	
PN	Fixed PIN used in legacy pairing.	
NM	Local friendly name of the module.	
IF	Host interface of the module	
BR	UART baud rate.	
UM	Configuration of UART mode.	
DB	Default configuration of bypass mode.	
MD	Discoverable state.	
PA	State of pairing mode.	
CA	State of connectable mode	
NC	Six digits decimal number of numeric comparison.	
PK	Passkey request.	
PI	Pairing result.	
IR	Inquiry result.	
FD	Address and name of found device.	
LC	List the connected devices	
SPP Responses and Indicators		
SM	Service name of the SPP profile.	
SS	State of SPP channel.	
CS	Result of connect attempt to a remote SPP device.	
DT	Data packet received from remote SPP device.	
RO	Communication role of SPP connection.	
HID Responses and Indicators		
IS	State of HID.	
CI	Result of connect attempt to a remote HID host.	
KR	Keyboard report received from remote HID host.	
OPP Responses and Indicators		
OS	State of OPP channel.	
CO	Result of connect attempt to a remote OPP device.	
OA	Object name pushed by an OPPC device	
OY	Object type pushed by an OPPC device	
OT	Object data packet pushed by an OPPC device	
BLE/iGate Responses and Indicators		
LU	Configuration of iGate service UUID128.	
LP	BLE connection parameters	
LB	State of bondable	
LS	State of BLE connection	
LD	Data packet received from remote BLE central device	

HFP Responses and Indicators		
HS	State of HFP channel.	
CH	Result of connect attempt to a remote HFP device.	
CC	Call State	
HI	Service status	
HV	Volume of HFP voice	

4. BAI message reference

4.1. General messages

4.1.1. OK—General Response of Success

4.1.1.1. Description:

This message is a general response when a command is accepted by the Module successfully.

4.1.1.2. Syntax:

Synopsis:
{OK}[<CR><LF>]

4.1.1.3. Parameter Description:

There is no parameter for this message.

4.1.1.4. Examples:

Ex. 4.1. To make Module discoverable:

➔ BC:MD=01<CR><LF> ← make Module discoverable.

← OK<CR><LF> ← Command accepted.

4.1.2. ER—General Response of Error

4.1.2.1. Description:

This message is a general response when there is an error with the command sent by the host.

4.1.2.2. Syntax:

Synopsis:
{ER=ErrCode}[<CR><LF>]

4.1.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
ErrCode	The error code to give the reason of an error. Value: 01h—03h 01: The command is not allowed in current state. 02: The command is not given in proper format. 03: The command cannot be recognized.	M	

4.1.2.4. Examples:

Ex. 4.2.

➔ BC:MD=01<CR><LF> ← make Module discoverable.

← ER=01<CR><LF> ← response from the module to indicate the command is not allowed in current state since the Module is already discoverable.

Ex. 4.3.

➔ BC:CS=00189600ABCD<CR><LF> ← connect to the SPP device

← ER=01<CR><LF> ← response from the module to indicate the command is not allowed in current state, the possible reason can be the SPP device has already connected or there is another connect attempting is being performed.

Ex. 4.4.

➔ BC:XX<CR><LF>

← ER=03<CR><LF> ← response from the module to indicate the command cannot be recognized.

4.1.3. PD—Power down

4.1.3.1. Description:

This message powers down or resets the module.

4.1.3.2. Syntax:

Synopsis:
[BC:]{PD=value}[<CR><LF>]

4.1.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
value	01: Reset the module. 02: Power down the module, after that, a high pulse on VREG_EN pin will power on the module.	O	If the parameter is not presented in command, the Module reports current value in response.

4.1.3.4. Examples:

Ex. 4.5. To reset the Module:

➔ BC:PD=01<CR><LF> ← Reset the Module.

← OK<CR><LF> ← The command is accepted.

← Copyright(c) 2021-2023 Flairmesh Technologies. FMB100 1.3.3.0<CR><LF> ← The power on copyright info.

Ex. 4.6. To power down the Module:

→ BC:PD=02<CR><LF> ← Power down the Module.

← OK<CR><LF> ← After that the Module powers down itself.

Ex. 4.7. Power down the Module when its VCHG pin is connected:

→ BC:PD=02<CR><LF> ← Power down the Module.

← ER=01<CR><LF> ← The Module won't power down when its VCHG pin is powered.

4.1.4. AD—Bluetooth address

4.1.4.1. Description:

This message queries and reports the Bluetooth address of the Module.

4.1.4.2. Syntax:

Synopsis:
[BC:]{AD}[<CR><LF>]
{AD=BtAddr}[<CR><LF>]

4.1.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BtAddr	The Bluetooth address of the module. Value: 12 digits number	O	Parameter only exists in response.

4.1.4.4. Examples:

Ex. 4.8. Query the Bluetooth address of the Module:

→ BC:AD<CR><LF> ← Query the Bluetooth address of the Module.

← AD=5CB6CC00ABCD<CR><LF> ← Response the module's Bluetooth address is 5C:B6:CC:00:AB:CD.

4.1.5. TP—Tx power

4.1.5.1. Description:

This message queries, sets or reports the transmit power of the Module. It is recommended to only decrease the power below the default values which have been fully verified for FCC/CE/BQB approvement.

If the parameter is not presented, the Module reports current transmit power configuration by the Indicator TP.

4.1.5.2. Syntax:

Synopsis:
[BC:]{TP}[=DefaultTx,MaximumTx,InquiryTx][<CR><LF>]
{TP=DefaultTx,MaximumTx,InquiryTx}[<CR><LF>]

4.1.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
DefaultTx	Default TX power in dBm. The default TX power used for paging, inquiry, and their responses, and as the initial power for new ACL links. Value: a 2 characters length hex numeric value Default: per firmware version and Bluetooth power class.	O	If the parameter is not presented in command, the Module reports current value in response.
MaximumTx	Maximum TX power in dBm. Bluetooth power control may raise the TX power up to this value. Value: a 2 characters length hex numeric value Default: per firmware version and Bluetooth power class.	O	If the parameter is not presented in command, the Module reports current value in response.
InquiryTx	TX power in dBm used in inquiry. Value: a 2 characters length hex numeric value	O	If the parameter is not presented in command, the Module reports current value in response.

4.1.5.4. Examples:

Ex. 4.9. Query the Tx Power setting of the Module:

➔ BC:TP<CR><LF> ← Query current Tx Power configuration.

← TP=14,14,14<CR><LF> ← Report current default Tx power, maximum Tx power and inquiry Tx power are all 20dBm (for FMB110).

Ex. 4.10. Set the Tx Power of Module:

➔ BC:TP=10,10,10<CR><LF> ← Set default Tx Power, maximum TX power and inquiry TX power to 16dBm (only valid for FMB110).

← TP=10,10,10<CR><LF> ← The module reports the new Tx Powers.

4.1.6. CD—Class of device

4.1.6.1. Description:

This message queries, sets or reports the Class of Device (COD) of the Module. Although COD can be changed by this command, it is recommended only use it to query the default value automatically set by the firmware. The firmware sets it depending on enabled profiles so that it has the best compatibility with other Bluetooth devices.

4.1.6.2. Syntax:

Synopsis:
[BC:]{CD}[=Cod][<CR><LF>]
{CD=Cod}[<CR><LF>]

4.1.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Cod	The COD of the Module Value: a 6 digits number Default: per firmware version	O	If the parameter is not presented in command, the Module reports current value in response.

4.1.6.4. Examples:

Ex. 4.11. Query the COD setting of the Module:

➔ BC:CD<CR><LF> ← Query the current COD configuration.

← CD=003F00<CR><LF> ← Response that the current COD as 003F00.

Ex. 4.12. Set the COD of Module:

➔ BC:CD=000540<CR><LF> ← Set a new COD of 000540 (not recommended).

← OK<CR><LF> ← Command accepted.

4.1.7. FD—Factory default or inquiry result

This message resets the module's configuration to its factory default values. It is also used as an indicator for remote devices in an inquiry or pairing.

4.1.7.1. Syntax:

Synopsis:
[BC:]{FD}[<CR><LF>]
{FD=Index,BtAddr,RSSI,COD}[,DevName][<CR><LF>]

4.1.7.2. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Index	Device index in an inquiry. 0x00~0x20.		
BtAddr	Found device address.		
RSSI	RSSI in two-hex values complement code.		
COD	Class of device.		
DevName	Device name found in an inquiry.		

4.1.7.3. Examples:

Ex. 4.13. Reset the module's all settings to its factory default:

← BC:FD<CR><LF> ← Reset the module's configurations.

→ ← The module will erase the configurations and power reset itself with the default values.

Refer to 4.1.22.4 for examples of its application in inquiry.

4.1.8. FT—User features

4.1.8.1. Description:

This message queries, sets or reports the user features of the Module.

4.1.8.2. Syntax:

Synopsis:
[BC:]{FT}[=ATPowerOn,ACPaired,ATLinkLost,Interval,DiscMode,DiscTimeout][<CR><LF>]
{FT=ATPowerOn,ACPaired,ATLinkLost,Interval,DiscMode,DiscTimeout}[<CR><LF>]

4.1.8.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
ATPowerOn	The attempt times of auto connect the last connected device after power on. Value: 00h—FFh 00: No auto connect attempt will be performed after power on. 01-FE: The attempt times of auto connect after power on. FF: The auto connect attempt will be performed permanently. Default: 00	O	If the parameter is not presented in command, the Module reports current value in response.
ACPaired	Auto connects after paired with a device. Value: 00 or 01 00: Disabled 01: Enabled Default: 00 (Enabled)	O	Same as above.
ATLinkLost	The attempt times of reconnect after link lost. Value: 00h—FFh 00: No reconnect attempt will be performed after link lost. 01-FE: The attempt times of reconnect after link lost. FF: The reconnect attempt will be performed permanently. Default: 00	O	Same as above.
Interval	The interval between each reconnect attempt after link lost. The unit is second. Value: 00h—FFh Default: 0A (10 seconds)	O	Same as above.

DiscMode	The discoverable mode. Value: bitmask Bit 0,1: 0h—3h 0: The module enters or quit discoverable mode by the command BC:MD=XX. 1: The module will enter discoverable mode automatically when paired device list is empty. 2: The module will enter discoverable mode automatically when power on. 3: The module will enter discoverable mode automatically when there is no connection. Bit 2: BLE automatic bondable when no connection 0: disabled 1: enabled	O	Same as above. Even if the discoverable is set one of the auto mode (01 h—03h), it can also be controlled by the command BC:MD=xx.
DiscTimeout	The timeout of discoverable status. The unit is second. Value: 0000h—FFFFh 0000: No timeout for discoverable status. 0001-FFFF: The timeout in second of discoverable status. Default: 0078	O	If the parameter is not presented in command, the Module reports current value in response.

Notes:

1. The default feature configuration may be different per software version.

4.1.8.4. Examples:

Ex. 4.14. Query the user features of the Module:

➔ BC:FT<CR><LF> ← Query current feature configuration.

← FT=00,01,00,0A,00,0078<CR><LF> ← The module responses current feature configuration.
Auto connection after power on disabled.
Auto connect after paired enabled.
Auto reconnect after link lost disabled.
The interval of auto reconnect has been set to 10s.
Discoverable mode as manually controlled by MD command. BLE automatically bondable is off.
The timeout of discoverable is 120s.

Ex. 4.15. Set the user features of the Module:

➔ BC:FT=01,01,00,0A,00,0078<CR><LF> ← Set the module features:
Auto connection after power on enabled.
Auto connect after paired enabled.
Auto reconnect after link lost disabled.
The interval of auto reconnect has been set to 10s.
Discoverable mode as manually controlled by MD command. BLE automatically bondable is off.
The timeout of discoverable is 120s.

← OK<CR><LF> ← Command accepted.

4.1.9. MM—Man-In-The-Middle

4.1.9.1. Description:

This message queries, sets or reports the Man-In-The-Middle protection setting of the Module. To enable Man-In-The-Middle protection in paring, the module's IO capability can't be set to "No IO".

4.1.9.2. Syntax:

Synopsis:
[BC:]{MM}[=State][<CR><LF>]
{MM=State}[<CR><LF>]

4.1.9.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
State	The new state of Man-In-The-Middle protection. Value: 00h or 02h 00: Deactivated 01: Activated 02: Activated and auto confirm the numeric comparison. Default: 02 (Activated and auto confirm)	O	If the parameter is not presented in command, the Module reports current value in response.

Notes:

1. Some Android devices require Man-In-The-Middle protection in pairing to set up a SPP link.

4.1.9.4. Examples:

Ex. 4.16. Query the Man-In-The-Middle protection setting :

➔ BC:MM<CR><LF> ← Query the current Man-In-The-Middle protection setting.

➔ MM=00<CR><LF> ← The Module reports it is deactivated.

Ex. 4.17. Active the Man-In-The-Middle protection feature :

➔ BC:MM=01<CR><LF> ← The Host actives Man-In-The-Middle protection setting.

➔ OK<CR><LF> ← Command accepted.

Ex. 4.18. Active the Man-In-The-Middle protection feature, and let the module confirm the numeric comparison automatically:

➔ BC:MM=02<CR><LF> ← The Host actives Man-In-The-Middle protection feature and requests that the Module shall automatically confirm the numeric comparison. Thus, no NC indicator will be output by the Module.

➔ OK<CR><LF> ← Command accepted.

4.1.10. IO—IO capability

4.1.10.1. Description:

This message queries, sets or reports the IO (input and output) capability of the Module used in pairing. The IO capability and the Man-In-The-Middle protection setting determine how the Module pairs with a peer Bluetooth device.

4.1.10.2. Syntax:

Synopsis:
[BC:]{IO}[=IoCapability][<CR><LF>]
{IO=IoCapability}[<CR><LF>]

4.1.10.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
IoCapability	The new IO capability of local device. Value: 00h – 03h 00: Display Only. The local device can only display 01: Display Yes/No. The local device can display and select Yes or No. 02: Keyboard Only. The local device can only input. 03: No IO. The local device has no IO capability Default: Depending on firmwares.	O	If the parameter is not presented in command, the Module reports current value in response.

Notes:

1. When the Man-In-The-Middle protection feature is enabled, different IO capability will cause different pairing procedure. In case of "**Display Yes/No**", both remote and local device will prompt a six digits numbers, the user should compare and confirm if the two numbers are the same or not, and then select **Yes** or **No** on the remote device accordingly, for the module side, the host MCU should send the command BC:NC=01 or BC:NC=00 to confirm or deny the numeric comparison. In case of "**Keyboard Only**", the remote device will prompt a six digits number as passkey, the user should input the same number at the Module side by command BC:PK (refer to section 4.1.30).
2. The "**Display Only**" and "**No IO**" are not allowed when the Man-In-The-Middle protection feature is enabled.

4.1.10.4. Examples:

Ex. 4.19. Query the IO capability setting :

➔ BC:IO<CR><LF> ← Query current IO capability.

← IO=03<CR><LF> ← The Module responses current IO capability setting as "No IO".

Ex. 4.20. Set the IO capability as "Keyboard Only" :

➔ BC:IO=02<CR><LF> ← Set the IO capability to "Keyboard Only".

← OK<CR><LF> ← Command accepted.

4.1.11. MT—Force-To-Be-Master

4.1.11.1. Description:

This message queries, sets or reports the Force-To-Be-Master policy of the Module.

Normally when a Bluetooth connection is made, the initiator will become the master and the acceptor will become the slave. Since the module can not have 3 or more connections in slave roles simultaneously, the force-to-be-master policy can be used to make role switch on the incoming connections.

4.1.11.2. Syntax:

Synopsis:
[BC:]{MT}[=Policy][<CR><LF>]
{MT=Policy}[<CR><LF>]

4.1.11.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Policy	The policy to be used to control the connection role. Value: 00h ~02h 00: the module won't request master-slave switch. 01: the module requests role switch on incoming connections. 02: the module requests role switch on incoming connections. And when it initiates a connection, if the remote device requests a role switch, it will make an additional switch so that it remains the master finally.	O	If the parameter is not presented in command, the Module reports current value in response.

4.1.11.4. Examples:

Ex. 4.21. Query the setting of Force-to-be-master:

➔ BC:MT<CR><LF> ← Query current setting of Force-to-be-master.

← MT=00<CR><LF> ← The Module reports it is deactivated.

Ex. 4.22. Active the Force-to-be-master:

➔ BC:MT=01<CR><LF> ← The Host activates the Force-to-be-master feature.

← OK<CR><LF> ← Command accepted.

4.1.12. SN—Sniff mode parameters

4.1.12.1. Description:

This message queries, sets or reports the parameters used in sniff mode.

4.1.12.2. Syntax:

Synopsis:

[BC:]{SN}[=State,MinInterval,MaxInterval,Attempt,Timeout,PassiveDuration][<CR><LF>]
{SN=EffectiveInterval}[<CR><LF>]

4.1.12.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
State	The new state of sniff mode. Value: 00h or 01h 00: Deactivated 01: Activated Default: 00 (Deactivated)	O	If the parameter is not presented in command, the Module reports current value in response.
MinInterval	Minimum acceptable interval in milliseconds Value: 0002h—FFFEh; only even values, up to max, are valid Time = MinInterval x 0.625 ms Time Range: 1.25 ms to 40.9 seconds	O	Same as above.
MaxInterval	Maximum acceptable interval in milliseconds Value: 0002h—FFFEh; only even values, up to max, are valid Time = MaxInterval x 0.625 ms Time Range: 2.5 ms to 40.9 seconds	O	Same as above.
Attempt	Number of Baseband receive slots the slave shall listen when the slave is not treating this as a scatternet link. Value: 0001h—7FFFh Time = Timeout x 1.25 ms Time Range: 0.625ms – 40.9 Seconds Time = Attempt	O	Same as above.
Timeout	Number of Baseband receive slots the slave shall listen when the slave is not treating this as a scatternet link. Value: 0001h—7FFFh Time = Timeout x 1.25 ms Time Range: 1.25ms to 40.9 seconds	O	Same as above.
PassiveDuration	The time in seconds that the module will keep in Passive mode Value: 0001—FFFFh Time range: 1 seconds to 65535 seconds	O	Same as above.
EffectiveInterval	Effective interval in for the connection just set up. Value: 0002h—FFFEh; only even values, up to max, are valid Time = MinInterval x 0.625 ms Time Range: 1.25 ms to 40959 ms	M	This parameter is only used by indicator to show the current connection's sniff interval.

4.1.12.4. Examples:

Ex. 4.23. Query the parameters of sniff mode of the Module:

➔ BC:SN<CR><LF> ← Query the current parameters of sniff mode.

← SN=00,0320,0320,0004,0640,0004<CR><LF> ← The Module reports the sniff mode is deactivated currently, the MinInterval is 500ms(800x0.625ms),

the MaxInterval is 500ms(800x0.625ms),
the Attempt is 5 ms,
the Timeout is 1 second,
the Passive duration is 4 seconds,

Ex. 4.24. Active the sniff mode with specified parameters:

➔ BC:SN=01,0012,0012,0001,0001,0001<CR><LF> ← active the sniff mode with the specified parameters:
the MinInterval is 11.25ms(18x0.625ms),
the MaxInterval is 11.25ms(800x0.625ms),
the Attempt is 1.25 ms,
the Timeout is 1.25 ms,
the Passive duration is 1 second

← OK<CR><LF> ← Command accepted.

Ex. 4.25. Indicate effective sniff interval when a connection is set up:

← IS=01,CCFA00386B90<CR><LF> ← A HID connection is undergoing, peer address is “CCFA00386B90”.
← CI=0,CCFA00386B90<CR><LF> ← Connection status is OK.
← IS=02,CCFA00386B90<CR><LF> ← Connection state is “Connected”.
← KR=00,00,00,00<CR><LF> ← A HID Output report.
← SN=0012<CR><LF> ← Indicates the current connection’s sniff interval is 11.25ms.
← SB=0048,0318,0012,0012 <CR><LF> ← Indicates the sniff subrating parameters of current connection is:
the transmit latency is 45ms (72x0.625ms)
the receive latency is 495ms (792x0.625ms)
the remote timeout is 11.25 ms
the local timeout is 11.25 ms

4.1.13. SB—Sniff sub-rating

4.1.13.1. Description:

This message reports the effective sniff sub-rating parameters of current connection.

4.1.13.2. Syntax:

Synopsis:
{SB=TransmitLatency,ReceiveLatency,RemoteTimeout,LocalTimeout}[<CR><LF>]

4.1.13.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
TransmitLatency	The maximum delay caused by sub-rating when sending data to the Host. Time = TransmitLatency x 0.625 ms	M	
ReceiveLatency	The maximum delay caused by sub-rating when receiving data from the Host. Time = ReceiveLatency x 0.625 ms	M	

RemoteTimeout	The time the Host keeps in sniff before entering sub-rating. Time = ReceiveTimeout x 0.625 ms	M	
LocalTimeout	The time the Module keeps in sniff before entering sub-rating. Time = LocalTimeout x 0.625 ms	M	

4.1.13.4. Examples:

Refer to the examples in section 4.1.12.4.

4.1.14. SP—Deep sleep

4.1.14.1. Description:

This message queries, sets or reports the Module's deep sleep mode.

The Module enables deep sleep by default so it will go into deep sleep automatically when possible after power on. It can be woken up when a preconfigured PIO change or UART RX data happens. If it's woken up by UART, some leading bytes on UART RX might be lost so please considering only use some messages don't contain important user data to wake up it.

4.1.14.2. Syntax:

Synopsis:
[BC:]{SP}[=State][<CR><LF>]
{SP=State}[<CR><LF>]

4.1.14.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
State	The new state of deep sleep. Value: 00h or 01h 00: Deep sleep disabled 01: Deep sleep enabled Default: 00 (disabled)	O	If the command is sent via UART when the Module is in deep sleep, state parameter is ignored, and the Module will wake up and reply SP=00.

4.1.14.4. Examples:

Ex. 4.26. Query the current deep sleep state :

➔ BC:SP<CR><LF> ← Query current deep sleep state.

← SP=00<CR><LF> ← The Module reports deep sleep is disabled.

Ex. 4.27. Enable the deep sleep mode :

➔ BC:SP=01<CR><LF> ← The Host enables the deep sleep mode.

← OK<CR><LF> ← Command accepted.

Ex. 4.28. To wake up the Module from deep sleep:

➔ BC:SP=00<CR><LF> ← The Host wakes up the Module from deep sleep.

← SP=00<CR><LF> ← The Module responses that it is woken up and the deep sleep mode has been disabled.

4.1.15. SV—Speaker volume

4.1.15.1. Description:

This message queries, sets or reports the speaker volume.

4.1.15.2. Syntax:

Synopsis:
[BC:]{SV}[=Volume][<CR><LF>]
{SV=Volume}[<CR><LF>]

4.1.15.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Volume	The volume of the speaker output. Value: 00h ~ 0Fh, 0F is the highest level of 0dB, the power step is 3dB. Default: 0A	O	If the parameter is not given in command, the Module reports current value in response.

4.1.15.4. Examples:

Ex. 4.29. Query the speaker volume:

→ BC:SV<CR><LF> ← Query the current speaker volume.

← SV=0A<CR><LF> ← The Module reports the volume is 0A, -15dB.

Ex. 4.30. Set the speaker volume when an audio connection existed:

→ BC:SV=08<CR><LF> ← Set the speaker volume to 08, -21dB.

← OK<CR><LF> ← Command accepted when an audio connection existed.

Ex. 4.31. Set the speaker volume while no audio connection existed:

→ BC:SV=08<CR><LF> ← Set the speaker volume to 08, -21dB.

← ER=01<CR><LF> ← The new volume is stored but cannot be applied.

4.1.16. MI—Microphone mute control

4.1.16.1. Description:

This message queries, sets or reports the microphone mute state.

4.1.16.2. Syntax:

Synopsis:
[BC:]{MI}[=State][<CR><LF>]
{MI=State}[<CR><LF>]

4.1.16.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
State	The new state of microphone mute. Value: 00h or 01h 00: Deactivated 01: Activated Default: 00	O	If the parameter is not given in command, the Module reports current value in response.

4.1.16.4. Examples:

Ex. 4.32. Query the microphone mute state:

➔ BC:MI<CR><LF> ← Query the current microphone mute state.

← MI=00<CR><LF> ← The Module reports its microphone is unmuted.

Ex. 4.33. Mute the microphone when and audio connection existed:

➔ BC:MI=01<CR><LF> ← Mute the microphone.

← OK<CR><LF> ← Command accepted and microphone is muted.

Ex. 4.34. Mute the microphone while no audio connection existed:

➔ BC:MI=01<CR><LF> ← Mute the microphone.

← ER=01<CR><LF> ← There's no audio connection so the command is rejected.

4.1.17. NM—Bluetooth device name

4.1.17.1. Description:

This message queries, sets or reports the Module's Bluetooth device name.

4.1.17.2. Syntax:

Synopsis:
[BC:]{NM}[=Name][<CR><LF>]
{NM=Name}[<CR><LF>]

4.1.17.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments

Name	The new Bluetooth device name. Length: 1—30 characters Default: Per software version.	O	If the parameter is not presented, the Module responses its current Bluetooth device name.

4.1.17.4. Examples:

Ex. 4.35. Query the Bluetooth device name :

➔ BC:NM<CR><LF> ← Query current Bluetooth device name.

← NM=FMB100<CR><LF> ← The Module responses the current Bluetooth device name "FMB100".

Ex. 4.36. Set a new Bluetooth device name :

➔ BC:NM=MY_BT_DEVICE<CR><LF> ← Set the Bluetooth device name to "MY_BT_DEVICE"

← OK<CR><LF> ← Command accepted.

Notes:

1. Many OSs use cache for the device name of paired Bluetooth devices. A re-pairing might be needed to make the new name effected.

4.1.18. BR—Query or change the UART baud rate

4.1.18.1. Description:

This message queries, sets or reports the Module's UART baud rate.

4.1.18.2. Syntax:

Synopsis:
[BC:]{BR}[=BaudRate][<CR><LF>]
{BR=BaudRate}[<CR><LF>]

4.1.18.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BaudRate	The new baud rate of the Module. Value: 01h—15h 01: 1200 02: 1800 03: 2400 04: 4800 05: 7200 06: 9600 07: 14400 08: 19200 09: 38400 0A: 56000	O	If the parameter is not presented, the Module responses its current UART baud rate.

	0B: 57600 0C: 115200 0D: 128000 0E: 230400 0F: 256000 10: 460800 11: 921600 12: 1382400 13: 1843200 14: 2764800 15: 3686400 Default: 06 (9600)		

4.1.18.4. Examples:

Ex. 4.37. Query the baud rate of the Module :

➔ BC:BR<CR><LF> ← Query the baud rate.

← BR=06<CR><LF> ← The Module reports the baud rate as 9600.

Ex. 4.38. Set the baud rate of Module:

➔ BC:BR=0C<CR><LF> ← The Host set the baud rate to 115200.

← OK<CR><LF> ← Command accepted.

Notes:

1. The response will be sent using current baud rate before switching to new baud rate.

4.1.19. UM—Query or configure the UART mode

4.1.19.1. Description:

This message queries, sets or reports the Module's UART mode.

4.1.19.2. Syntax:

Synopsis:
[BC:]{UM}[=StopBits,Parity,Priority][<CR><LF>]
{UM}[=StopBits,Parity,Priority][<CR><LF>]

4.1.19.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
StopBits	The stop bits of UART mode Value: 00h or 01h 00: 1 stop bit 01: 2 stop bits Default: 00 (1 stop bit)	O	If the parameter is not presented, the Module responses its current UART baud rate.
Parity	The parity of UART mode	O	Same as above.

	Value: 00h – 02h 00: No parity 01: Odd parity 02: Even parity Default: 00 (No parity)		
Priority	The priority mode Value: 00h or 01h 00: Throughput priority 01: Low latency priority Default: 01 (Low latency priority)	O	Same as above.

4.1.19.4. Examples:

Ex. 4.39. Query the UART mode of the Module:

➔ BC:UM<CR><LF> ← Query the UART mode.

← UM=00,00,01<CR><LF> ← The Module reports the UART mode as 1 stop bit, no parity and low latency priority.

Ex. 4.40. Set the UART mode of Module:

➔ BC:UM=01,01,00<CR><LF> ← Set the UART mode as 2 stop bits, odd parity and throughput priority.

← OK<CR><LF> ← Command accepted.

Notes:

1. The response will be sent in current UART mode.

4.1.20. BP—bypass mode

4.1.20.1. Description:

This message queries, sets or reports the runtime bypass mode of the Module. When the Module powers on it has no Bluetooth connection with other devices. The data on its UART interface are considered as messages and parsed according to the format defined in 1.1. After it is connected, depending on its bypass mode settings, it may enter transparent mode in which the data on UART are forwarded to Bluetooth channel directly.

There are 5 different bypass mode explained below:

0. No bypass, message mode

This is the default mode when power on and no connection exists. All data on UART interface are considered as messages.

1. Bypass to First Connected Channel

When the Module becomes connected and no other connection exists before, the Module automatically enters transparent mode and all data on UART after connection are considered as user data and forwarded to the connected Bluetooth channel.

In multiple SPP connections exist under this mode, only the first SPP channel is used to forward user data. Data received from other SPP channel are abandoned silently.

If there are several connections have been established, the host cannot set the Module to this Bypass mode since it's hard to determine which is the first connected channel.

2. Bypass to HID ASCII Channel

When working in this mode, the data sent to the Module via UART port can only be ASCII characters 20h—7Eh and 0Dh, the Module will map it into USB keyboard (US layout) scan codes and send them to remote Bluetooth HID host. This mode can only enter when there exists Bluetooth HID connection.

3. **Bypass to All SPP Channels (Mixture)**

   When working in this mode, the data sent to the Module via UART port will be forwarded transparently to all the connected SPP channels. At the same time, the content received from each of the remote SPP device will be transparently output by the Module via UART port without a channel identifier to indicate the data source.

4. **Bypass to Specified Channel**

   Like the **Bypass to First Connected Channel** mode, but the Bypass channel will be specified by the parameter [BypassChannel]

Table 4.1 Bypass Mode

Mode	Channel	Data Sent to UART	Data Output from UART	Comments
Message mode	N/A	BAI commands	BAI responses or indicators	
Bypass to First Connected Channel	SPP channel	Data forward transparently to SPP channel	Raw data received from remote SPP device.	The bypass channel will be selected to the first connected Bluetooth channel.
	HID channel	Raw HID input report to the HID host. About the HID input report, please refer to section 4.3.4.4.	Raw HID output report	
Bypass to HID ASCII Channel	HID channel	ASCII characters(20h—7Eh) forwarded to the Bluetooth HID host	BAI indicator of HID output report	
Bypass to All SPP Channel	Connected SPP channel	Raw data to be sent to all connected SPP device	Raw data received from all connected SPP device	
Bypass to Specified Channel	SPP channel	Raw data to be sent to remote SPP device	Raw data received from remote SPP device.	the Bypass channel will be specified by the parameter [Channel]
	HID channel	Raw data (HID reports) to be sent the HID host. About the HID report, please refer to section 4.3.4.4.	ASCII indicators	

Additionally, there are 2 different speed modes when the Module works in bypass mode:

0. **Normal speed mode**

   The Module still looks for “BP” command when transparently forward data to Bluetooth channels, so the speed is affected accordingly.

1. **High speed mode**

   The Module doesn’t check the data and they are forwarded directly, the throughput will be higher than normal speed but this mode cannot exit until the Bluetooth connection is disconnected from the remote side.

Notes:

1. Bypass mode can only be entered when Bluetooth connection exists. It can be set in message mode and become effective automatically when a connection is made.
2. Bypass mode exits automatically when the corresponding Bluetooth connection disconnects.
3. “BP” command can be used to put the Module back to message mode if it’s in a bypass mode with normal speed.

It's the only command allowed in bypass mode.

4. When sending this command, there should be a 1 second interval both before and after itself:
   <1 second idle on UART> BC:BP=00,00<CR><LF><1 second idle on UART>

4.1.20.2. Syntax:

Synopsis:
[BC:]{BP=Mode,Channel,SpeedMode}[<CR><LF>]

4.1.20.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Mode	The bypass mode: Value: 00h—04h 00: No bypass, message mode 01: Bypass to First Connected Channel 02: Bypass to HID ASCII Channel 03: Bypass to All SPP 04: Bypass to Specified Channel Default: 01(Bypass to First Connected Channel)	O	If the parameter is not presented, the Module responses its current bypass mode settings.
Channel	The bypass channel: Value: 00h—07h and 10h 01—07: the SPP channel ID. Up to 7 SPP channels available. 08: The HID channel. 09: Reserved. 0A: The BLE channel.	O	The parameter will only take effect when the Bypass channel mode is **Bypass to Specified Channel**
SpeedMode	The bypass speed mode: Value: 00h or 01h 00: Normal speed 01: High speed	O	

4.1.20.4. Examples:

Ex. 4.41. Set the bypass mode to "No bypass" when it is in bypass mode:

	← The Host keeps the UART TX idle for 1 second.
➔ BC:BP=00,00,00<CR><LF>	← The Host set the bypass mode to **Message Mode**.
← OK<CR><LF>	← Command accepted.

Ex. 4.42. Query the bypass mode setting :

➔ BC:BP<CR><LF>	← Query the current bypass.
← BP=01,00,00<CR><LF>	← The Module responses that current bypass mode is **Bypass to First Connected Channel** and speed mode is normal.

Ex. 4.43. Set the bypass mode to **Bypass to Specified Channel**:

➔ BC:BP=04,03,00<CR><LF>	← Set the bypass mode to **Bypass to Specified Channel**, the channel is specified to SPP channel 03, **Normal Speed**.

← OK<CR><LF> ← Command accepted.

Ex. 4.44. Set the bypass mode to **Bypass to Specified Channel, High Speed**:

→ BC:BP=04,03,01<CR><LF> ← Set the bypass mode to **Bypass to Specified Channel**, the channel is specified to SPP channel 03, **High Speed**.

← OK<CR><LF> ← Command accepted.

4.1.21. DB—Default bypass mode

4.1.21.1. Description:

This message queries, sets or reports the Module's default bypass settings. The settings changed by "BP" only affects the run-time values. To set a default bypass setting for each time it powers on, use this message instead.

4.1.21.2. Syntax:

Synopsis:
[BC:]{DB}[=Mode,Channel,SpeedMode][<CR><LF>]
{DB=Mode,Channel,SpeedMode}[<CR><LF>]

4.1.21.3. Parameter Description:

Please refer to 4.1.20.3.

4.1.21.4. Examples:

Ex. 4.45. Query current default bypass mode settings of the Module:

→ BC:DB<CR><LF> ← Query the default Bypass mode.

← DB=01,00,00<CR><LF> ← The Module responses the default Bypass mode as **Bypass to First Connected Channel** and speed mode is normal.

Ex. 4.46. Set the default bypass mode settings:

→ BC:DB=01,00,01<CR><LF> ← The Host set the default Bypass mode as **Bypass to First Connected Channel** and speed mode is high.

← OK<CR><LF> ← Command accepted.

4.1.22. IQ—Inquiry nearby Bluetooth devices

4.1.22.1. Description:

This command let the Module to inquiry the Bluetooth devices nearby. Found devices will be indicated by "IR" indicator.

4.1.22.2. Syntax:

Synopsis:
[BC:]{IQ}[=Mode,MaxNum,Timeout,COD][<CR><LF>]

4.1.22.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Mode	Inquiry mode, value: 00h—02h 00: do not inquiry device name 01: inquiry both RSSI and device name 02: do not inquiry RSSI and device name in inquiry	O	If the parameter is not presented, the default mode is 01.
MaxNum	The maximum number of found device in an inquiry Value: 00h—20h	O	If the parameter is not presented, the default value is 0x20.
Timeout	The inquiry timeout. The time the inquiry is performed for is in fact timeout * 1.28 seconds. Value: 0000h~0030.	O	If the parameter is not presented, the default timeout is 08.
COD	Class of device, used to only show devices with bits set in the cod. Six hex values defined as standard Bluetooth COD.	O	If the parameter is not presented, the default mode is 0, which means all devices within range are reported regardless of their class of device.

4.1.22.4. Examples:

Ex. 4.47. Inquiry nearby Bluetooth devices, the device name is got after the inquiry stage:

➔ BC:IQ<CR><LF> ← inquire the Bluetooth device.

← AP=01<CR><LF> ← indicate the Module is now inquiring.

← FD=00,00189600000C,D3,007A020C<CR><LF> ← indicate the 1st found device's address, the RSSI is -45dBm, the device's COD is 007A020C.

← IR=01<CR><LF> ← indicate there is total1 Bluetooth device found.

← AP=00<CR><LF> ← indicate Module is now in idle.

← FN=00, 00189600000C,D3,7A020C,iPhone<CR><LF> ← indicate the 1st device name is "iPhone".

Ex. 4.48. Inquiry nearby Bluetooth devices, the device name is got in inquiry stage:

➔ BC:IQ<CR><LF> ← inquire the Bluetooth device.

← AP=01<CR><LF> ← indicate the Module is now inquiring.

← FD=00,00189600000C,D3,7A020C,iPhone<CR><LF> ← indicate the 1st found device's address, the RSSI is -45dBm, the device's COD is 007A020C.

← IR=01<CR><LF> ← indicate there is total1 Bluetooth device found.

← AP=00<CR><LF> ← indicate Module is now in idle.

4.1.23. FN—Found device name

This message reports a new remote device found in an inquiry.

4.1.23.1. Syntax:

Synopsis:
{FN=Index,BtAddr,DevName][<CR><LF>]

4.1.23.2. Parameter Description:

Please refer to 4.1.7.2.

4.1.23.3. Examples:

Please refer to 4.1.22.4.

4.1.24. AP—State of module

4.1.24.1. Description:

This message reports the inquiry state of the Module.

4.1.24.2. Syntax:

Synopsis:
{AP=StateCode}[<CR><LF>]

4.1.24.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
StateCode	The state of the Module. Value: 00h—01h, FFh 00: The Module is idle and ready. 01: The Module is now inquiring for Bluetooth device. FF: The Module is now initializing or power off.	M	

4.1.24.4. Examples:

Refer to the examples in section 4.1.22.4

4.1.25. IR—Inquiry result

4.1.25.1. Description:

This indicator indicates the total number of found device in last inquiring.

4.1.25.2. Syntax:

Synopsis:
{IR=DevCount}[<CR><LF>]

4.1.25.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
DevCount	The device count found in the last inquiring. Value: 00h—0Ch	M	

4.1.25.4. Examples:

Refer to the examples in section 4.1.22.4.

4.1.26. MD—Module discoverable

4.1.26.1. Description:

This message queries, sets or reports the Module's discoverable status. Only when the Module is discoverable, it can be searched by nearby Bluetooth devices.

4.1.26.2. Syntax:

Synopsis:
[BC:]{MD}[=Status][<CR><LF>]
{MD=Status}[<CR><LF>]

4.1.26.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Status	The new status of discoverable. Value: 00h or 01h 00: Not discoverable 01: Discoverable Default: 00 (Not discoverable)	O	If the parameter is not presented in command, the Module reports its current discover status.

4.1.26.4. Examples:

Ex. 4.49. Query the current discoverable status of the Module:

➔ BC:MD<CR><LF> ← query the current discoverable status.

← MD=00<CR><LF> ← report the Module is not discoverable currently.

Ex. 4.50. Make Module discoverable:

➔ BC:MD=01<CR><LF> ← make Module discoverable.

← OK<CR><LF> ← Command accepted.

4.1.27. PA—Pairable

4.1.27.1. Description:

This message queries, sets or reports the Module's pairable state. Only when it is enabled, it can be paired/bonded with another Bluetooth device.

4.1.27.2. Syntax:

Synopsis:
[BC:]{PA}[=Status][<CR><LF>]

4.1.27.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Status	The new status of pairing mode. Value: 00h or 01h 00: Pairing/Bonding disabled 01: Pairing/Bonding enabled Default: 01 (Enabled)	O	If the parameter is not presented, the Module reports current status of pairing mode.

4.1.27.4. Examples:

Ex. 4.51. To query the current pairing mode status of the Module:

➔ BC:PA<CR><LF> ← query the current pairable mode state.

← PA=01<CR><LF> ← report the pairable is enabled currently.

Ex. 4.52. To disable the pairing mode of the Module:

➔ BC:PA=00<CR><LF> ← disable pairable.

← OK<CR><LF> ← Command accepted.

4.1.28. CA—Connectable

4.1.28.1. Description:

This message queries, sets or reports the Module's connectable state. Only when the connectable is enabled, it can be connected by another Bluetooth device.

4.1.28.2. Syntax:

Synopsis:
[BC:]{CA}[=State][<CR><LF>]
{CA=State}[<CR><LF>]

4.1.28.3. Parameter Description:

Parameter	Description	Mandatory	Comments

		or Optional	
State	The new state of connectable mode. Value: 00h or 01h 00: connect disabled 01: connect enabled Default: 01 (Enabled)	O	If the parameter is not presented, the Module reports current state of connectable mode.

4.1.28.4. Examples:

Ex. 4.53. To query the current connectable mode state of the Module:

➔ BC:CA<CR><LF> ← query the current connectable state.

← CA=01<CR><LF> ← report the connectable is enabled currently.

Ex. 4.54. To disable the connectable mode of the Module:

➔ BC:CA=00<CR><LF> ← disable connectable.

← OK<CR><LF> ← Command accepted.

4.1.29. NC—Numeric comparison

4.1.29.1. Description:

In pairing, if the Man-In-The-Middle protection is activated and the IO capability is configured as "**Display Yes/No**", the module may output the number for numeric comparison by the indicator NC. And then the Host can use NC command to confirm or deny it.

4.1.29.2. Syntax:

Synopsis:
[BC:]{NC=Confirmation}[<CR><LF>]
{NC=RandomNumber}[<CR><LF>]

4.1.29.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Confirmation	The confirmation of numeric comparison. Value: 00h or 01h 00: deny the numeric comparison. 01: confirm the numeric comparison.	M	This parameter is only used in command.
RandomNumber	A six digital decimal value to be displayed to the user, the same number must be shown in the peer device's display when a successful pairing is made.	M	This parameter is only used in indicator.

4.1.29.4. Examples:

Ex. 4.55. To deny the numeric comparison:

← NC=276385<CR><LF> ← indicate the number for numeric comparison is 276385.

➔ BC:NC=00<CR><LF> ← deny the numeric comparison.

← OK<CR><LF> ← Command accepted.

Ex. 4.56. To confirm the numeric comparison:

← NC=276385<CR><LF> ← indicate the number for numeric comparison is 276385.

→ BC:NC=01<CR><LF> ← confirm the numeric comparison.

← OK<CR><LF> ← Command accepted.

4.1.30. PK—Passkey

4.1.30.1. Description:

In pairing, if the module's IO capability is set as "**Keyboard Only**", the peer Bluetooth device may display a number on its display and the Host needs to use this command to send the same number as the passkey to finish pairing successfully.

4.1.30.2. Syntax:

Synopsis:
[BC:]{PK=Number}[<CR><LF>]
{PK}[<CR><LF>]

4.1.30.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Number	The passkey numbers Value: 000000 – 999999	M	This parameter is only used in command.

4.1.30.4. Examples:

Ex. 4.57. Input the passkey number when pairing:

← PK<CR><LF> ← indicates there is a Bluetooth device is passkey request

→ BC:PK=104536<CR><LF> ← input the passkey number: 104536

← OK<CR><LF> ← Command accepted.

← FD=00,6805711900AE,7FFF,00000000,Samsung Galaxy S5<CR><LF> ← Indicate peer device initiating the pairing, whose address is "6805711900AE" and name is "Samsung Galaxy S5". The RSSI and COD are unknown.

← PI=00,6805711900AE<CR><LF> ← Pairing done successfully.

4.1.31. PN—Fixed PIN

4.1.31.1. Description:

When pairing with a remote device with version 2.0 or lower, legacy pairing will be used, requiring a fixed PIN code to match the one set with the peer device. This message can be used to query and set the PIN code of the module.

4.1.31.2. Syntax:

Synopsis:
[BC:]{PN=NumberString}[<CR><LF>]
{PN=NumberString}[<CR><LF>]

4.1.31.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
NumberString	The PIN code, the maximum allowed length is 16. Value: usually a code as “0000” or “1234” is used with old Bluetooth devices.	O	If the parameter is not presented, the Module reports current state of connectable mode.

4.1.31.4. Examples:

Ex. 4.58. Input the passkey number when pairing:

➔ BC:PN=104536<CR><LF> ← Set the fixed PIN code as “104536”.
← OK<CR><LF> ← Command accepted.
← BC:PN<CR><LF> ← Query the current fixed PIN code.
← PN=104536<CR><LF> ← The current PIN code is “104536”.

4.1.32. PI—Pairing Result

4.1.32.1. Description:

This message indicates the pairing results.

4.1.32.2. Syntax:

Synopsis:
{PI=Results,BdAddr}[<CR><LF>]

4.1.32.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Results	The pairing results. Value: 00h or 01h 00: Pairing successfully done. 01: Pairing failed.	M	
BdAddr	The Bluetooth address of the peer Bluetooth device.	M	

4.1.32.4. Examples:

Refer to the examples in section 4.1.30.4.

4.1.33. CP—Clear the paired Bluetooth device list

4.1.33.1. Description:

This command clears the paired device list stored in the Module. If the Module is in connection, it will disconnect and then clear the paired device list.

4.1.33.2. Syntax:

Synopsis:
[BC:]{CP}[=BdAddr][<CR><LF>]

4.1.33.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth device to be removed.	O	If the parameter is given, only the corresponding Bluetooth device is removed.

4.1.33.4. Examples:

Ex. 4.59. Clear the paired device list:

➔ BC:CP<CR><LF> ← Clear the paired device list.

← OK<CR><LF> ← Command accepted.

Ex. 4.60. Remove the specified device from paired device list:

➔ BC:CP=00189600000A<CR><LF> ← Remove device 00:18:96:00:00:0A from the paired list.

← OK<CR><LF> ← Command accepted.

4.2. SPP messages

4.2.1. SM—Service name of SPP

4.2.1.1. Description:

This message queries, sets or reports the service name of SPP profile. The service name can be used by peer device's Bluetooth stack to distinguish it from other SPP devices.

4.2.1.2. Syntax:

Synopsis:
[BC:]{SM}[=Name][<CR><LF>]
{SM=Name}[<CR><LF>]

4.2.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Name	The service name of SPP profile. Length: 1—16 characters Default: SPP Dev.	O	If the parameter is not presented, the Module reports current service name.

4.2.1.4. Examples:

Ex. 4.61. To query current service name of SPP profile:

➔ BC:SM<CR><LF> ← query current service name.

← SM=SPP Dev<CR><LF> ← report current service name, it's "SPP Dev".

Ex. 4.62. To change the service name of SPP profile:

➔ BC:SM=GPS Dev<CR><LF> ← change the service name to "GPS Dev"

← OK<CR><LF> ← Command accepted.

4.2.2. CS—Connect to a remote SPP device

4.2.2.1. Description:

This command starts a SPP connection to the last connected SPP device if no parameter is given, or a remote Bluetooth SPP device whose address is set as the parameter. It is also used as an indicator to report the error code happens in the connection attempt.

4.2.2.2. Syntax:

Synopsis:
[BC:]{CS}[=BdAddr][<CR><LF>]
{CS=Error,BdAddr}[<CR><LF>]

4.2.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth SPP device to connect.	O	If the parameter is not presented, the Module initiates connection to the last connected SPP device.
Error	The error code of the connection attempt. The high nibble indicates the channel ID (0h—xh, x is the maximum SPP instance count, refer to 4.1.1) and the low nibble indicates the result code (0h—Bh). Result code value: 0h—Bh		The parameter is only used in indicator.

	0: No error, connect succeeded. 1: Service search failed. 2: Service level connection establishment failed. 3: Profile instance already connected. 4: RFCOMM connection failed to be established. 5: Requested server channel not registered by this profile instance. 6: Connection attempt timed out. 7: The remote device rejected the connection. 8: The remote device terminated the connection. 9: Unsuccessful due to an abnormal disconnect while establishing the RFCOMM connection. A: The connection attempt failed because there is already a connection to that remote device on the requested RFCOMM channel. B: Connect failed due to invalid frame size request from app.		

4.2.2.4. Examples:

Ex. 4.63. Connect to the last connected SPP device:

➔ BC:CS<CR><LF>	← connect to the last connected device with the SPP profile.
← SS=01,00189600ABCD<CR><LF>	← the Module is now connecting to the last connected device which address is 00:18:96:00:AB:CD.
← CS=00,00189600ABCD<CR><LF>	← connecting result: success.
← SS=02,00189600ABCD<CR><LF>	← the Module is now connected to the last connected device.

Ex. 4.64. Connect to the specified device with the SPP profile:

➔ BC:CS=00189600000A<CR><LF>	← connect to the specified device 00:18:96:00:00:0A with the SPP profile.
← SS=01,00189600000A<CR><LF>	← the Module is now connecting to the specified SPP device which address is 00:18:96:00:00:0A.
← CS=00,00189600000A<CR><LF>	← connecting result: success.
← SS=02,00189600000A<CR><LF>	← the Module is now connected to the specified SPP device.

4.2.3. DS—Disconnect with the remote SPP device

4.2.3.1. Description:

This command requests the Module to disconnect with a connected Bluetooth SPP device.

4.2.3.2. Syntax:

Synopsis:
[BC:]{DS}[=BdAddr][<CR><LF>]

4.2.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth SPP device to disconnect.	O	If the Bluetooth address parameter is not presented, the Module will disconnect with all the connected SPP devices.

4.2.3.4. Examples:

Ex. 4.65. Disconnect with all the connected SPP devices:

➔ BC:DS<CR><LF> ← disconnect with all the connected SPP devices.

← SS=00<CR><LF> ← the SPP channel 0 of Module is now disconnected and is connectable.

← SS=10<CR><LF> ← the SPP channel 1 of Module is now disconnected and is connectable.

← SS=30<CR><LF> ← the SPP channel 3 of Module is now disconnected and is connectable.

Ex. 4.66. Disconnect with the specified device:

➔ BC:DS=00189600000A<CR><LF> ← disconnect with the specified device 00:18:96:00:00:0A.

← SS=00<CR><LF> ← the SPP channel 0 of Module is now disconnected and is connectable.

4.2.4. SS—State of SPP connections

4.2.4.1. Description:

This message queries and reports the state of each SPP channel.

4.2.4.2. Syntax:

Synopsis:
[BC:]{SS}[<CR><LF>]
{SS=SppState,BdAddr}[<CR><LF>]

4.2.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
SppState	The state of each SPP channel. The high nibble indicates the channel No. (0h-xh, x is the maximum SPP instance count, refer to 4.1.1) and the low nibble indicates the state(0h-2h). State Value: 0h-2h	M	This parameter is only used in indicator.

	0: idle and connectable. 1: connecting. 2: connected.		
BdAddr	The Bluetooth address of remote SPP device.	M	This parameter is only used in indicator.

4.2.4.4. Examples:

Ex. 4.67. To query the state of each SPP channel:

➔ BC:SS<CR><LF> ← query the state of each SPP channel.

← SS=01,00189600ABCD<CR><LF> ← the SPP channel 0 of Module is now connecting to the remote device which address is 00:18:96:00:AB:CD.

← SS=12,00189601ABCD<CR><LF> ← the SPP channel 1 of Module is now connected with the remote device which address is 00:18:96:01:AB:CD.

← SS=22,00189603ABCD,43<CR><LF> ← the SPP channel 2 of Module is now connected with the remote device which address is 00:18:96:02:AB:CD, the name ID is 43.

← SS=30<CR><LF> ← the SPP channel 3 of Module is now connectable

← SS=40<CR><LF> ← the SPP channel 4 of Module is now connectable

← SS=50<CR><LF> ← the SPP channel 5 of Module is now connectable

4.2.5. DT—Send data to remote SPP device

4.2.5.1. Description:

This message sends data to or reports data from the connected remote SPP device.

4.2.5.2. Syntax:

Synopsis:
[BC:]{DT=Channel,DataLen,Data}[<CR><LF>]
{DT=Channel,DataLen,Data}[<CR><LF>]

4.2.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Channel	The SPP channel on which the data happens. Value: 00h—0xh (x is the maximum SPP instance count, refer to 4.1.1) 00—0x: the channel ID of SPP	M	
DataLen	The length in bytes of the data to be sent. Value: 00h-FFh	M	
Data	Raw bytes of data in their original form.	M	

4.2.5.4. Examples:

Ex. 4.68. To send data use SPP channel 0:

➔ BC:DT=00,0A,1234567890<CR><LF> ← send a ASCII string “1234567890” use SPP channel 0, the data

length is 10(Dec).

← OK<CR><LF> ← Command accepted.

4.2.6. RO—Communication role of SPP connection

4.2.6.1. Description:

This message queries or changes communication role of a SPP connection.

Per the limitation of the firmware, the module can only have up to two simultaneous SPP connections in slave roles.

4.2.6.2. Syntax:

Synopsis:
[BC:]{RO=}[Role,]{BdAddr}[<CR><LF>]
{RO=Role,BdAddr}[<CR><LF>]

4.2.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Role	The communication role. 00: master 01: slave 02: only valid in response, means the role requested cannot be switched into.	O	If the parameter is not presented, the Module report the current role.
BdAddr	The Bluetooth address of the remote Bluetooth SPP device.	M	

4.2.6.4. Examples:

Ex. 4.69. Switch the SPP connection to master role:

→ BC:RO=00,5CB6CCB18F0B<CR><LF> ← Set the role to master for the SPP connection to 5C:B6:CC:B1:8F:0B.

← OK<CR><LF> ← Command accepted.

← RO=00,5CB6CCB18F0B<CR><LF> ← Connection role is master now.

Ex. 4.70. Switch the SPP connection to master role:

→ BC:RO=00,5CB6CCB18F0B<CR><LF> ← Set the role to master for the SPP connection to 5C:B6:CC:B1:8F:0B.

← OK<CR><LF> ← Command accepted.

← RO=02,5CB6CCB18F0B<CR><LF> ← Role switch failed, current role not change. This might be caused by that the remote device cannot be put into slave role.

4.3. HID messages

4.3.1. CI—Connect to a remote HID host

4.3.1.1. Description:

This command starts a HID connection to the last connected HID device if no parameter is given, or a remote Bluetooth HID device whose address is set as the parameter. It is also used as an indicator to report the error code happens in the connection attempt.

4.3.1.2. Syntax:

Synopsis:
[BC:]{CI}[=BdAddr][<CR><LF>]
{CI=Error,BdAddr}[<CR><LF>]

4.3.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth HID host to connect.	O	If the parameter is not presented, the Module initiates connection to the last connected HID device.
Error	The error code of the connection attempt. Value: 00h—04h 00: No error, connect attempt succeeded. 01: Connection failed. 02: Out of resource. 03: Timeout waiting for connection. 04: Disconnected remotely during setup.	M	The parameter is only used in indicator.

4.3.1.4. Examples:

Ex. 4.71. Connect to the last connected HID device:

➔ BC:CI<CR><LF>	← connect to the last connected HID device.
← IS=01,00189600ABCD<CR><LF>	← the Module is now connecting to the last connected HID host which address is 00:18:96:00:AB:CD.
← CI=00,00189600ABCD<CR><LF>	← connecting result: success.
← IS=02,00189600ABCD<CR><LF>	← the Module is now connected to the last connected HID host.

Ex. 4.72. Connect to the specified HID host:

➔ BC:CI=00189600000A<CR><LF>	← connect to the specified HID host: 00:18:96:00:00:0A.
← IS=01,00189600000A<CR><LF>	← the Module is now connecting to the specified HID host which address is 00:18:96:00:00:0A.
← CI=00,00189600000A<CR><LF>	← connecting result: success.
← IS=02,00189600000A<CR><LF>	← the Module is now connected to the specified HID host.

4.3.2. DI—Disconnect with the remote HID host

4.3.2.1. Description:

This command requests the Module to disconnect with a connected Bluetooth HID device.

4.3.2.2. Syntax:

Synopsis:
[BC:]{DI}[=BdAddr][<CR><LF>]

4.3.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth HID device to disconnect.	O	If the Bluetooth address parameter is not presented, the Module will disconnect current HID connection.

4.3.2.4. Examples:

Ex. 4.73. Disconnect with the connected HID device:

➔ BC:DI<CR><LF> ← disconnect with the connected HID host.

← IS=00<CR><LF> ← current HID connection is down and the Module is connectable as a HID device now.

Ex. 4.74. Disconnect with the specified device:

➔ BC:DI=00189600000A<CR><LF> ← disconnect with the specified HID device: 00:18:96:00:00:0A.

← IS=00<CR><LF> ← current HID connection is down and the Module is connectable as a HID device now.

4.3.3. IS—State of HID connection

4.3.3.1. Description:

This message queries or reports the state of HID connection.

4.3.3.2. Syntax:

Synopsis:
[BC:[{IS}[<CR><LF>]
{IS=HidState}[,BdAddr][<CR><LF>]

4.3.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
HidState	The state of HID channel. State Value: 00h-02h	M	This parameter is only used in indicator.

	0: idle and connectable. 1: connecting. 2: connected.		
BdAddr	The Bluetooth address of remote HID device.	O	This parameter is only used in indicator.

4.3.3.4. Examples:

Ex. 4.75. To query the state of HID channel:

➔ BC:IS<CR><LF> ← query the state of HID channel.

← IS=01,00189600ABCD<CR><LF> ← the HID channel of Module is now connecting to the remote HID host which address is 00:18:96:00:AB:CD.

4.3.4. KR—HID report

4.3.4.1. Description:

This message sends a HID report to or reports a HID report from the remote HID device.

4.3.4.2. Syntax:

Synopsis:
[BC:]{KR=HidReport}[<CR><LF>]
{KR=HidReport}[<CR><LF>]

4.3.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
HidReport	The HID report.	M	

4.3.4.4. Hid Report Format and Examples:

Please note that not all HID input reports in this section are supported for a given firmware configuration. Please

Start (1Byte)	Report Id (1Byte)	Data (8 Bytes for Keyboard Report and Joystick/Gamepad, 2 Bytes for Consumer Report, 5 Bytes for Mouse)

Notes:

1. In **Message mode**, each byte in HID report is encoded as 2 ASCII characters represents the hex values of the raw bytes. After encoded they are separated by commas. In **Bypass mode**, all HID reports are in raw format (binary). About the Message and Bypass mode, please refer to section 4.1.20.

1. Keyboard Report:

A1	01	Modifier	00	ScanCode1	ScanCode2	ScanCode3	ScanCode4	ScanCode5	ScanCode6

The **Modifier** byte is a bit mask interpreted as shown in Table 4.2. For example, you can use 02h or 20h to turn a lower case 'a' into an upper case 'A'.

Table 4.2 Bit Mask of Modifier Byte in Keyboard Report

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Right GUI	Right Alt	Right Shift	Right Ctrl	Left GUI	Left Alt	Left Shift	Left Ctrl

The **ScanCode** is defined by the USB HID Spec.

Ex. 4.76. If the key A and the Right Shift are pressed, the keyboard report should be:

A1	01	20	00	04	00	00	00	00	00

Ex. 4.77. If all of the pressed keys have been released, the keyboard report should be:

A1	01	00	00	00	00	00	00	00	00

Ex. 4.78. To send a keyboard report to HID host:

➔ BC:KR=A1,01,00,00,04,00,00,00,00,00<CR><LF> ← send a keyboard report to the HID host. The key A is pressed.

← OK<CR><LF> ← Command accepted.

➔ BC:KR=A1,01,00,00,00,00,00,00,00,00<CR><LF> ← send a keyboard report to the HID host. The pressed key is released.

← OK<CR><LF> ← Command accepted.

2. Consumer Key Report:

A1	02	LowByte	HighByte

The **LowByte** and **HighByte** are bit mask interpreted as shown in Table 4.3:

Table 4.3 Consumer Key Function

Consumer Key Function	LowByte	HighByte
AC Home	01	00
AL Email Reader	02	00
AC Search	04	00
AL Keyboard Layout (Virtual Apple Keyboard Toggle)	08	00
Volume Up	10	00
Volume Down	20	00
Mute	40	00
Play/Pause	80	00
Scan Next Track	00	01
Scan Previous Track	00	02
Stop	00	04
Eject	00	08
Fast Forward	00	10
Rewind	00	20
Stop/Eject	00	40
AL Internet Browser	00	80

Ex. 4.79. To increase the volume, the consumer key report should be:

A1	02	10	00

Ex. 4.80. To release the consumer key, the consumer key report should be:

A1	02	00	00

Ex. 4.81. To send a consumer key report to HID host:

➔ BC:KR=A1,02,10,00<CR><LF>	← send a consumer key report to the HID host. The Volume Up key is pressed.
← OK<CR><LF>	← Command accepted.
➔ BC:KR=A1,02,00,00<CR><LF>	← send a consumer key report to the HID host. The pressed key is released.
← OK<CR><LF>	← Command accepted.
➔ BC:KR=A1,02,08,00<CR><LF>	← send a consumer key report to the HID host to popup the Virtual Apple Keyboard.
← OK<CR><LF>	← Command accepted.
➔ BC:KR=A1,02,00,00<CR><LF>	← send a consumer key report to the HID host. The pressed key is released.
← OK<CR><LF>	← Command accepted.

3. Mouse Report:

A1	03	Buttons	XmXl	YlXh	YhYm	Wheel

The **Buttons** is a bit mask interpreted as shown in Table 4.4:

Table 4.4 Bit Mask of Buttons Byte in Mouse Report

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button 8	Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1

The **XhXmXl** compose the movement on X axis. The range is from -2048(800h) to +2047(7FFh).

The **YhYmYl** compose the movement on Y axis. The range is from -2048(800h) to +2047(7FFh).

The **Wheel** is the movement of wheel. The range is from -127(81h) to +127(7Fh).

Ex. 4.82. To press the left button of the mouse, the mouse report should be:

A1	03	01	00	00	00	00

Ex. 4.83. To move the mouse towards top-right(X:3 pixel, Y:-4 pixel), the mouse report should be:

A1	03	00	03	C0	FF	00

XhXmXl = 003h = 3 (Decimal)

YhYmYl = FFCh = -4 (Decimal)

Ex. 4.84. To scroll up for 1 line, the mouse report should be:

A1	03	00	00	00	00	FF

Wheel = FFh = -1 (Decimal)

Ex. 4.85. To send a mouse report to HID host:

➔ BC:KR=A1,03,01,00,00,00,00<CR><LF>	← send a mouse report to the HID host. The Button1(Left Button) is pressed.
← OK<CR><LF>	← Command accepted.
➔ BC:KR=A1,03,00,00,00,00,00<CR><LF>	← send a mouse report to the HID host. The pressed key is released.
← OK<CR><LF>	← Command accepted.
➔ BC:KR=A1,03,00,03,C0,FF,00<CR><LF>	← send a mouse report to the HID host. The mouse is move to upper-right.

← OK<CR><LF>	← Command accepted.
→ BC:KR=A1,03,00,00,00,00,00<CR><LF>	← send a mouse report to the HID host. The mouse is stopped.
← OK<CR><LF>	← Command accepted.
→ BC:KR=A1,03,00,00,00,00,FF<CR><LF>	← send a mouse report to the HID host. The mouse wheel scroll up for one line.
← OK<CR><LF>	← Command accepted.
→ BC:KR=A1,03,00,00,00,00,00<CR><LF>	← send a mouse report to the HID host. The mouse is stopped.
← OK<CR><LF>	← Command accepted.

4. Joystick/Gamepad:

A1	04	Throttle	X	Y	Z	Rz	HatSwitch	Buttons1	Buttons2

The **Throttle** is the throttle value. The range is from -127(81h) to +127(7Fh).

The **X** is the position of X axis of left stick. The range is from -127(81h) to +127(7Fh).

The **Y** is the position of Y axis of left stick. The range is from -127(81h) to +127(7Fh).

The **Z** is the position of Z axis (generally, it is used as X axis of right stick). The range is from -127(81h) to +127(7Fh).

The **Rz** is the rotation of Z axis (generally, it is used as Y axis of right stick). The range is from -127(81h) to +127(7Fh).

The **HatSwitch** is the direction of hat switch. The range is from 00h to 07h, represents Top(00h), Top-right(01h), Right(02h), Bottom-right(03h), Bottom(04h), Bottom-left(05h), Left(06h), Top-Left(07h). The value out of range is invalid, and the hat switch will not move.

The **Buttons1** is a bit mask of first 8 buttons(Button1—Button8). Each bit represents one button.

The **Buttons2** is a bit mask of second 8 buttons(Button9—Button16). Each bit represents one button.

Ex. 4.86. To set the Throttle to 5, Left X to -2, Left Y to 3, Right X to 2, Right Y to -5, the joystick report should be:

A1	04	05	FE	03	02	FB	08	00	00

Throttle = 05h = 5 (Decimal)

X = Left X = FEh = -2 (Decimal)

Y = Left Y = 03h = 3 (Decimal)

Z = Right X = 02h = 2 (Decimal)

Rz = Right Y = FBh = -5 (Decimal)

HatSwitch = 08 = No movement

Buttons1 = 00h = No button be pressed.

Buttons2 = 00h = No button be pressed

Ex. 4.87. To set the Throttle to -10, Button2, Button3 and Button 15 pressed, Hat Switch to Bottom-left, the joystick report should be:

A1	04	F6	00	00	00	00	05	06	40

Throttle = F6h = -10 (Decimal)

X = Left X = 00h = 00 (Decimal)

Y = Left Y = 00h = 00 (Decimal)

Z = Right X = 00h = 00 (Decimal)

Rz = Right Y = 00h = 00 (Decimal)

HatSwitch = 05 = Bottom-left

Buttons1 = 06h = Button2 and Button3 be pressed.

Buttons2 = 40h = Button15 be pressed

Ex. 4.88. To send a joystick/gamepad report to HID host:

➔ BC:KR=A1,04,05,FE,03,02,FB,08,00,00<CR><LF>	← send a joystick/gamepad report to the HID host.
← OK<CR><LF>	← Command accepted.
➔ BC:KR=A1,04,F6,00,00,00,00,05,06,40<CR><LF>	← send a joystick/gamepad report to the HID host.
← OK<CR><LF>	← Command accepted.
➔ BC:KR=A1,04,00,00,00,00,00,08,00,00<CR><LF>	← send a joystick/gamepad report to the HID host.
← OK<CR><LF>	← Command accepted.

5. User Defined Report:

A1	FF	Length	D1	D2	D3	D4	……	……	Dn

The **Length** is the length of report data in byte.

The **D1**, **D2**, ……,**Dn** are the report data, here **n** equal to the Length.

Ex. 4.89. If a user defined report has 10 report data, the report should be:

A1	FF	0A	01	02	03	04	05	06	07	08	09	0A

Length = 0Ah = 10 (Decimal)

Report data = 01 02 03 04 05 06 07 08 09 0A

Note:

1. For Bypass mode, the **Length** can up to FFh(255 in decimal), while for Message mode, the maximum **Length** is 55h(85 in decimal).

6. Hid Output Report from Hid Host:

The Hid output report from a common Hid Host contains four bytes: IdleRate,NumLock,CapsLock,ScrollLock.

4.3.5. AS—Send ASCII string to remote HID host

4.3.5.1. Description:

This command sends an ASCII string to the remote HID host.

4.3.5.2. Syntax:

Synopsis:
[BC:]{AS=AsciiStr}[<CR><LF>]

4.3.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
AsciiStr	The ASCII string needs to be sent to HID host. Only the ASCII character in the range of 20h—7Eh and 08h, 09h, 0Dh can be included in this parameter. The data out of range will be thrown away.	M	

Notes:

1. An escape character ('\') is available like which has been widely used in C/C++ language. In this case, "\r" or "\R" represents Enter(0Dh), "\b" or "\B" represents Backspace(08h), "\t" or "\T" represents Tab(09h), and "\\" represents '\' character.

4.3.5.4. Examples:

Ex. 4.90. Send an ASCII string to HID host:

➔ BC:AS=Hello Bluetooth HID<CR><LF> ← send an ASCII string to the HID host.

← OK<CR><LF> ← Command accepted.

Ex. 4.91. Send a ASCII string to HID host:

➔ BC:AS=!@#$%^abcdef9876<CR><LF> ← send an ASCII string to the HID host.

← OK<CR><LF> ← Command accepted.

Ex. 4.92. To send an ASCII string to HID host:

➔ BC:AS=Hello\tWorld!\b\rThis is \\Flairmesh\\<CR><LF> ← send an ASCII string to the HID host.

← OK<CR><LF> ← Command accepted.

The HID host will receive and display as below:

Hello World

This is \Flairmesh\

4.4. OPP messages

4.4.1. CO—Connect to a remote OPPS device

4.4.1.1. Description:

This command starts an OPP connection to the last connected OPP server device if no parameter is given, or a remote OPP device whose address is set as the parameter. It is also used as an indicator to report the error code happens in the connection attempt.

4.4.1.2. Syntax:

Synopsis:
[BC:]{CO}[=BdAddr][<CR><LF>]
{CO=Error,BdAddr}[<CR><LF>]

4.4.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth OPPS device to connect.	O	
Error	The error code of the connection attempt. Result code value: 00h—08h 00: Connect attempt succeeded. 01—08: Connect attempt failed	M	

4.4.1.4. Examples:

Ex. 4.93. Connect to the last connected OPPS device:

➔ BC:CO<CR><LF> ← connect to the last connected OPPS device.

← OS=01,00189600ABCD<CR><LF> ← the Module is now connecting to the last connected OPPS device which address is 00:18:96:00:AB:CD.

← CO=00,00189600ABCD<CR><LF> ← connecting result: success.

← OS=02,00189600ABCD,0374<CR><LF> ← the Module is now connected to the last connected OPPS device, the maximum allowed packet size for the connection is 0x374.

Ex. 4.94. Connect to the specified OPPS device:

➔ BC:CO=00189600000A<CR><LF> ← connect to the specified OPPS device: 00:18:96:00:00:0A.

← OS=01,00189600000A<CR><LF> ← the Module is now connecting to the specified OPPS device which address is 00:18:96:00:00:0A.

← CO=00,00189600000A<CR><LF> ← connecting result: success.

← OS=02,00189600000A,0374<CR><LF> ← the Module is now connected to the specified OPPS device, the maximum allowed packet size for the connection is 0x374.

4.4.2. DO—Disconnect with the remote OPPS device

4.4.2.1. Description

This command requests the Module to disconnect with a connected Bluetooth OPPS device.

4.4.2.2. Syntax:

Synopsis:
[BC:]{DO}[=BdAddr][<CR><LF>]

4.4.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth OPP device to disconnect.	O	If the Bluetooth address parameter is not presented, the Module will disconnect current OPP connection.

4.4.2.4. Examples:

Ex. 4.95. To disconnect with all of the connected OPP devices:

➔ BC:DO<CR><LF> ← disconnect with all of the connected OPP devices.

← OS=00<CR><LF> ← the OPP channel of Module is now disconnected and is connectable.

Ex. 4.96. To disconnect to the specified OPP device:

➔ BC:DO=00189600000A<CR><LF> ← disconnect with the specified device 00:18:96:00:00:0A.

← OS=00<CR><LF> ← the OPP channel of Module is now disconnected and is connectable.

4.4.3. OS—Query state of OPP connection

4.4.3.1. Description:

This message queries or reports the state of OPP connection.

4.4.3.2. Syntax:

Synopsis:
[BC:]{OS}[<CR><LF>]
{OS=OppState}[,BdAddr,MaxPktSize][<CR><LF>]

4.4.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
OppState	The state of OPP connection. State Value: 00h-02h 00: idle and connectable. 01: connecting. 02: connected	M	
BdAddr	The Bluetooth address of remote OPP device.	O	
MaxPktSize	When the OPP connection is set up, this field indicates a maximum packet size negotiated with the remote device. Please note that there's another packet size limitation for "OT" command. Range: 0000-FFFF	O	

4.4.3.4. Examples:

Ex. 4.97. To query the state of OPP channel:

➔ BC:OS<CR><LF> ← query the state of OPP channel.

← OS=02,00189602ABCD,0374<CR><LF> ← the OPP channel is now connected with the remote device which address is 00:18:96:02:AB:CD, the maximum allowed packet size is 0x374.

4.4.4. OA—OPPC push object file name/OPPS accept or reject

4.4.4.1. Description:

When the module works as OPP client, this command is used to send the file size and file name of the object which will be pushed to the remote OPPS device.

When the module works as OPP server, this command is used to accept or reject a pushed in object. It is also used as an indicator when a remote OPP client push a file in.

4.4.4.2. Syntax:

Synopsis:
[BC:]{OA=ObjSize,NameLen,FileName}[<CR><LF>]
[BC:]{OA=AllowPushIn}[<CR><LF>]
{OA=ObjSize,NameLen,FileName}[<CR><LF>]

4.4.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
ObjSize	The size in bytes of the object which will be pushed. Value: 00000000h—FFFFFFFFh	M	
NameLen	The length in bytes of the object name. The terminator 0000 is NOT included. The name length must be even, because the name is encoded in UTF-16. Value: 02h—FEh	M	
FileName	The file name in UTF-16 of the object. The terminator 0000 should NOT be included.	M	
AllowPushIn	The bool value to accept or reject a pushed in object . Value: 00h or 01h 00: deny the object. 01: accept the object.	M	

4.4.4.4. Examples:

Ex. 4.98. As an OPP client, to send the file size and file name of the object:

➔BC:OA=00000002,0A,a.txt ← the object size is 2 Bytes, the name length is 10 Bytes. Here the read "a.txt" is actually encoded in UNICODE so the 5 characters are encoded into 10 bytes, the raw bytes after ',' are "00 61 00 2E 00 74 00 78 00 74".

← OK<CR><LF> ← Command accepted.

Ex. 4.99. As an OPP server, when the remote OPP client pushes in an object:

➔ OA=00016CBE,000A,a.txt ← the object total size is 0x16CBE Bytes. The file name length is 10. The file name is "**a.txt**" which is encoded in UNICODE so the raw bytes here after the ',' are "00 61 00 2E 00 74 00 78 00 74".

➔ OT=00,033A,… ← the first packet of the pushed in object, with packet size 0x33A, and contents as …

← BC:OA=01<CR><LF> ← Accept the pushed in object.

4.4.5. OY—OPPC push object file type

4.4.5.1. Description:

This command is used to send the file type of the object which will be pushed to the remote OPPS device. It is also used as

an indicator when a remote OPPC pushes in type information.

4.4.5.2. Syntax:

Synopsis:
[BC:]{OY=TypeLen,Type}[<CR><LF>]
{OY=TypeLen,Type}[<CR><LF>]

4.4.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
TypeLen	The length in bytes of the object type. The terminator 00 is NOT included. Value: 01h—FFh	M	
Type	The type of the object. The terminator 00 should NOT be included.	M	

4.4.5.4. Examples:

Ex. 4.100. To send the file type of the object:

➔ BC:OY=0C,text/x-vcard<CR><LF> ← the type length is 12(0Ch).
the object type is "**text/x-vcard**".

← OK<CR><LF> ← Command accepted.

4.4.6. OT—OPPC push object data packet

4.4.6.1. Description:

This command is used to send a data packet of the object which will be pushed to the remote OPPS device. It is also used as an indicator when an OPP client device pushes in data.

4.4.6.2. Syntax:

Synopsis:
[BC:]{OT=Final,PacketLen,Pakcet}[<CR><LF>]
{OT=Final,PacketLen,Packet}[<CR><LF>]

4.4.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Final	To indicate if this is the final or only packet of this object. Value: 00h or 01h 00: This packet is not the final packet of this object 01: This packet is the final packet of this object.	M	
PacketLen	The length in bytes of the packet. Value: 01h—F8/FEh	M	For the first packet of an object, the length

			shall be less than 248. For the following packets, the length shall be less than 254.
Packet	The raw data of the packet.	M	

4.4.6.4. Examples:

Ex. 4.101. To send a data packet of the object:

➔ BC:OT=00,01,a ← this is a packet of this object which is not the final one
the length of this packet is 1 Byte, payload is 'a'.

← OK<CR><LF> ← Command accepted.

Ex. 4.102. To send the last data packet of the object:

➔ BC:OT=01,01,b ← this is the last packet of this object
the length of this packet is 1 Byte, payload is 'b'.

← OK<CR><LF> ← Command accepted.

Ex. 4.103. A data packet pushed by remote OPP client, this indicator is shown in HEX mode:

← 4F 54 3D 30 31 2C 33 30 2C 42 45 47 49 4E 3A 56 43 41 52 44 0D 0A 56 45 52 53 49 4F 4E 3A 32 2E 31 0D 0A 4E 3A 4D 69 63 68 61 65 6C 0D 0A 45 4E 44 3A 56 43 41 52 44 0D 0A 0D 0A

← this is the final/only packet of this object (30 31 = 01h)
the length of this packet is 48 Bytes (33 30 = 30h).
the raw data of the packet is displayed in red(42 45 47…… 52 44 0D 0A).

4.5. BLE/iGate messages

4.5.1. LU—iGate service UUID128

4.5.1.1. Description:

The module works as a BLE peripheral and it implements a proprietary GATT based service named iGate which can be taken as a virtual SPP channel between the peripheral and connected central device such as mobile phones, tablets and PCs. The service UUID is also used in the advertisement packet of the module so it can be distinguished from nearby devices when common central devices make a scanning.

This message queries, sets or reports the UUID128 for the iGate service.

4.5.1.2. Syntax:

Synopsis:
[BC:]{LU}[=Word0,…,Word7][<CR><LF>]
{LU=Word0,…,Word7}[<CR><LF>]

4.5.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Word0	The least significant word of the service UUID128, for example, "31F6" of a UUID128 string "C14D2C0A-401F-B7A9-841F-E2E93B80F631". The word value itself is also LSB Hex Value: xxxx	O	If the parameter is not presented in command, the Module reports current value in response.
Wordn	The words of the service UUID128. Hex Value: xxxx	O	Same as above.
Word7	The most significant word of the service UUID128, for example, "4DC1" of a UUID128 string "C14D2C0A-401F-B7A9-841F-E2E93B80F631". The word value itself is also LSB Hex Value: xxxx	O	Same as above.

4.5.1.4. Examples:

Ex. 4.104. Query the iGate service UUID128:

➔ BC:LU<CR><LF> ← query the current service UUID128.

← LU=31F6,803B,E9E2,1F84,A9B7,1F40,0A2C,4DC1<CR><LF> ← report the UUID128, which in string format is "C14D2C0A-401F-B7A9-841F-E2E93B80F631".

Ex. 4.105. Set the iGate service UUID128:

➔ BC:LU=31F6,803B,E9E2,1F84,A9B7,1F40,0A2C,4DC1<CR><LF> ← set service UUID128 to "C14D2C0A-401F-B7A9-841F-E2E93B80F631".

← OK<CR><LF> ← Command accepted.

4.5.2. LP—Query or change the BLE connection parameters

4.5.2.1. Description:

This command queries, sets or reports the preferable BLE connection parameter characteristic of Module. The connection parameters affect the data transmission delay and throughput. Smaller interval allows lower delay and higher throughput with higher power consumption. Bigger latency allows the module omits more connection events from the central device, which results in lower power consumption of the module. The supervision timeout defines the period during which if no connection event happens would be taken as a disconnection state. Bigger supervision timeout will make the central device (for example, a mobile phone) not show disconnected state in time when the module is out of range. Please contact Flairmesh for recommendation if not-default values are required.

Once configured, the configuration will take effect at the next time when a Bluetooth connection is being established.

This setting is stored in NVM in the Module.

4.5.2.2. Syntax:

Synopsis:
[BC:]{LP}[=MinInterval,MaxInterval,Latency,SupervisionTimeout][<CR><LF>]
{LP=MinInterval,MaxInterval,Latency,SupervisionTimeout}[<CR><LF>]

{LP=EffectiveInterval,EffectiveLatency,EffectiveSupervisionTimeout][<CR><LF>]

4.5.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
MinInterval	The minimum value for the connection interval. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default 0048		If the parameter is not presented, the Module reports current value.
MaxInterval	The maximum value for the connection interval. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define, must be no less than MinInterval) Default 0058		Same as above.
Latency	The slave latency for the connection in number of connection events. Hex Value: xxxx (0000 to 03E8) Default 0004		Same as above.
SupervisionTimeout	The connection supervisor timeout in unit of 10ms. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default 0190		Same as above.
EffectiveInterval	The connection interval of current BLE connection.	M	This indicator and parameter is sent to connected remote BLE device on iGate control channel.
EffectiveLatency	The connection latency of current BLE connection.	M	Same as above.
EffectiveSupervisionTimeout	The connection supervision timeout of current BLE connection.	M	Same as above.

4.5.2.4. Examples:

Ex. 4.106. To query current preferred connection parameters of the Module:

➔ BC:LP<CR><LF> ← query the current BLE connection parameters.

← LP=0048,0058,0004,0190<CR><LF> ← report the current BLE connection parameters.

Ex. 4.107. To set the preferred connection parameters of the Module:

➔ BC:LP=0020,0030,0004,00C8<CR><LF> ← set new BLE connection parameters.

← OK<CR><LF> ← Command accepted.

4.5.3. LB—Query or change the BLE Bondable State

4.5.3.1. Description:

This command queries, sets or reports the bondable state of BLE channel. Only when the Module is bondable, it can be

scanned and connected by a new BLE Central device. Once become bondable, the module will keep in the state for 60 seconds. If no successful pairing is made during this period, the module returns to idle state. If a connection and pairing is made the module also quits its bondable state and becomes bonded with the paired central device. Then the module's BLE is only connectable to the bonded device until a new bondable state is triggered. Please note the "BC:CP" can clear the paired device list including the bonded BLE central device.

4.5.3.2. Syntax:

Synopsis:
{BC:LB}[=State]{<CR><LF>}
{LB=State<CR><LF>}

4.5.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
State	The new state of advertising. Value: 00h or 01h 00: Not bondable. 01: Bondable to new BLE centrals.	O	If the parameter is not presented, the Module reports current bondable state.

4.5.3.4. Examples:

Ex. 4.108. Query the current bondable state of the Module:

➔ BC:LB<CR><LF> ← query the current bondable state.

← LB=00<CR><LF> ← report the Module is not bondable now.

Ex. 4.109. Make Module bondable:

➔ BC:LB=01<CR><LF> ← make Module bondable.

← OK<CR><LF> ← Command accepted.

4.5.4. DL—Disconnect with remote BLE central device

4.5.4.1. Description:

This command requests the Module to disconnect with the connected remote BLE central device.

4.5.4.2. Syntax:

Synopsis:
[BC:]{DL}[=BdAddr][<CR><LF>]

4.5.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments

BdAddr	The Bluetooth address of the connected central device.	O	If no address is given, it disconnects the current connected BLE central.

4.5.4.4. Examples:

Ex. 4.110. Disconnect with the connected devices:

➔ BC:DL<CR><LF> ← disconnect with current connected device.

← LS=00<CR><LF> ← the Module is now disconnected and not connectable to BLE central devices, to make it connectable to the bonded central device, use "BC:LS=01", to make it bondable to a new central device, use "BC:LB=01".

Ex. 4.111. Disconnect with a device by its address:

➔ BC:DL=2CF0A29782E6<CR><LF> ← disconnect the connection with device "2C:F0:A2:97:82:E6".

← LS=00<CR><LF> ← the Module is now disconnected and not connectable.

4.5.5. LS—Query or change the state of BLE connection

4.5.5.1. Description:

This command queries, sets or reports the status of BLE connection.

4.5.5.2. Syntax:

Synopsis:
[BC:]{LS}[=State][<CR><LF>]
{LS=State}[,BdAddr][<CR><LF>]

4.5.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
State	The new state of connection. Value: 00h or 01h 00: Idle, not connectable. 01: Connecting. 02: Connected, only valid in indicator.	O	If the parameter is not presented, the Module reports current state.
BdAddr	The Bluetooth address of the connected BLE central device.	O	Only available when State=02

4.5.5.4. Examples:

Ex. 4.112. Query the state of BLE connection:

➔ BC:LS<CR><LF> ← query the state of BLE connection.

← LS=02,2CF0A29782E6<CR><LF> ← currently the Module is connected to the BLE central device whose address is 2C:F0:A2:97:82:E6.

Ex. 4.113. Query the state of BLE connection:

➔ BC:LS<CR><LF> ← query the state of BLE connection.

← LS=01<CR><LF> ← currently the Module is only connectable to its bonded central device.

Ex. 4.114. Query the state of BLE connection:

→ BC:LS<CR><LF> ← query the state of BLE connection.

← LS=00<CR><LF> ← currently the Module is idle.

Ex. 4.115. Request the module become connectable to its bonded central device:

→ BC:LS=01<CR><LF> ← request the Module to be connectable to its bonded central device.

← LS=01<CR><LF> ← the Module becomes connectable to its bonded central device.

Ex. 4.116. Request the module become idle/not connectable to its bonded central device:

→ BC:LS=00<CR><LF> ← request the Module to be idle.

← LS=00<CR><LF> ← the Module becomes idle.

4.5.6. LD—Send data packet to remote BLE central

This message sends data to or reports data from the connected BLE central device.

4.5.6.1. Syntax:

Synopsis:
{BC:LD=DataLen,Data<CR><LF>}
{LD=DataLen,Data<CR><LF>}

4.5.6.2. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
DataLen	The length in bytes of the data to be sent. Value: 00h-14h	M	
Data	The raw data.	M	

4.5.6.3. Examples:

Ex. 4.117. To send data to remote BLE central device:

→ BC:LD=0A,1234567890<CR><LF> ← send a data packet, the data length is 10(Dec).

← OK<CR><LF> ← Command accepted.

Ex. 4.118. A data packet is received from the BLE central device:

← LD=0A,1234567890<CR><LF> ← a data packet received from BLE central device, the data length is 10(Dec), its contents is ASCII string “1234567890”.

4.6. AG messages

4.6.1. CG—Connect to a remote HFP device

4.6.1.1. Description:

This command starts an AG HFP connection to the last connected HF device if no parameter is given, or a remote Bluetooth HFP device whose address is set as the parameter. It is also used as an indicator to report the error code happens in the connection attempt.

his message queries, sets or reports the service name of SPP profile. The service name can be used by peer device's Bluetooth stack to distinguish it from other SPP devices.

4.6.1.2. Syntax:

Synopsis:
[BC:]{CG}[=Addr][<CR><LF>]
[BC:]{CG}[=Addr,AutoConnectAudio][<CR><LF>]

4.6.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the Bluetooth HFP device to connect.	O	If the parameter is not presented, the Module initiates connection to the last connected HF device.
AutoConnectAudio	If the audio connection is set up automatically after the signaling channel is setup. 00: False 01: True	O	If the parameter is not presented, the default value is 01/True.

4.6.1.4. Examples:

Ex. 4.119. Connect to a HFP device and open audio channel automatically:

➔ BC:CG=84AD8D0228F0<CR><LF>	← connect to the specified device 84:AD:8D:02:28:F0 which is a HF device, automatically open audio channel when signaling channel is built.
← GS=01,84AD8D0228F0<CR><LF>	← Connecting signaling channel.
← GS=02,84AD8D0228F0<CR><LF>	← Signaling channel connected.
← GS=03,84AD8D0228F0<CR><LF>	← Connecting audio channel.
← GS=04,84AD8D0228F0<CR><LF>	← Audio channel connected.
← GC=02<CR><LF>	← Audio channel connected, "Wide band speech" (mSBC) codec is used.

Ex. 4.120. Only connect to a HFP device's signaling channel:

➔ BC:CG=84AD8D0228F0,00<CR><LF>	← connect to the signaling channel of device 84:AD:8D:02:28:F0.
← GS=01,84AD8D0228F0<CR><LF>	← Connecting signaling channel.
← GS=02,84AD8D0228F0<CR><LF>	← Signaling channel connected.

4.6.2. DG—Disconnect with a remote HFP device

4.6.2.1. Description:

This command disconnects with current connected HF device.

4.6.2.2. Syntax:

Synopsis:
[BC:]{DG}[<CR><LF>]

4.6.2.3. Parameter Description:

There is no parameter for this message.

4.6.2.4. Examples:

Ex. 4.121. Disconnect a HFP device only has an opened signal channel:

← GS=02,84AD8D0228F0<CR><LF> ← When a signaling channel has been connected.
➔ BC:DG<CR><LF> ← Disconnect.
← GS=06,84AD8D0228F0<CR><LF> ← Disconnecting signal channel.
← GS=00<CR><LF> ← AG profile becomes idle and connectable for new connection.

Ex. 4.122. Disconnect a HFP device has an opened audio channel:

← GS=04,84AD8D0228F0<CR><LF> ← When an audio connection has been connected.
➔ BC:DG<CR><LF> ← Disconnect.
← GS=05,84AD8D0228F0<CR><LF> ← Disconnecting audio channel.
← GS=06,84AD8D0228F0<CR><LF> ← Disconnecting signal channel.
← GS=00<CR><LF> ← Audio channel is disconnected, while signaling channel is kept on.

4.6.3. GA—Open/close audio connection

4.6.3.1. Description:

This command starts or stops an AG HFP audio connection. A signaling channel must existed when the command is used.

4.6.3.2. Syntax:

Synopsis:
[BC:]{GA=enable}[<CR><LF>]

4.6.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Enable	A flag to start/stop audio connection. 00: Stop 01: Start	O	If the parameter is not presented, the Module initiates connection to the last connected

			HFP device.

4.6.3.4. Examples:

Ex. 4.123. Start audio connection with a connected HFP device:

← GS=02,84AD8D0228F0 <CR><LF>	← When a signaling channel has been connected.
→ BC:GA=01<CR><LF>	← Start audio connection with currently connected remote HFP device.
← GS=03,84AD8D0228F0 <CR><LF>	← Connecting audio channel.
← GS=04,84AD8D0228F0 <CR><LF>	← Audio channel connected.
← GC=02<CR><LF>	← Audio channel connected, “Wide band speech” (mSBC) codec is used.

Ex. 4.124. Stop audio connection with a connected HFP device:

← GS=04,84AD8D0228F0 <CR><LF>	← When a signaling channel has been connected.
→ BC:GA=00<CR><LF>	← Start audio connection with currently connected remote HFP device.
← GS=05,84AD8D0228F0 <CR><LF>	← Connecting audio channel.
← GS=02,84AD8D0228F0 <CR><LF>	← Audio channel is disconnected, while signaling channel is kept on.

4.6.4. GC—Audio codec in use

4.6.4.1. Description:

When an audio connection is made, this indicator shows the current audio codec in use.

4.6.4.2. Syntax:

Synopsis:
{GA=codec}[<CR><LF>]

4.6.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
codec	01: cvsd, narrow band, as defined in Bluetooth Sig’s HFP spec. 02: msbc, wide band, as defined in Bluetooth Sig’s HFP spec.	M	

4.6.4.4. Examples:

Refer to the examples in 4.6.1.4.

4.6.5. GS—State of AG connection

4.6.5.1. Description:

This message queries and reports the state of AG profile.

4.6.5.2. Syntax:

Synopsis:

[BC:]{GS}[<CR><LF>]
{GS=State}[,BdAddr][<CR><LF>]

4.6.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the remote HFP device.	O	Not available when in idle.
State	The state code of the AG profile. 00: Idle, ready for new connection. 01: Connecting signaling channel. 02: Signaling channel connected. 03: Connecting audio channel. 04: Audio channel connected. 05: Disconnecting audio channel. 06: Disconnecting signal channel.		The parameter is only used in indicator.

4.6.5.4. Examples:

Ex. 4.125. Connect to the last connected SPP device:

➔ BC:GS<CR><LF>	← connect to the last connected device with the SPP profile.
← GS=04,00189600ABCD<CR><LF>	← the Module is now connecting to the last connected device which address is 00:18:96:00:AB:CD.
← CS=00,00189600ABCD<CR><LF>	← connecting result: success.
← SS=02,00189600ABCD<CR><LF>	← the Module is now connected to the last connected device.

Ex. 4.126. Connect to the specified device with the SPP profile:

➔ BC:CS=00189600000A<CR><LF>	← connect to the specified device 00:18:96:00:00:0A with the SPP profile.
← SS=01,00189600000A<CR><LF>	← the Module is now connecting to the specified SPP device which address is 00:18:96:00:00:0A.
← CS=00,00189600000A<CR><LF>	← connecting result: success.
← SS=02,00189600000A<CR><LF>	← the Module is now connected to the specified SPP device.

4.6.6. GV—Remote speaker volume

4.6.6.1. Description:

This message queries, sets and reports the remote speaker volume of a connected HFP device.

4.6.6.2. Syntax:

Synopsis:
[BC:]{GV}[=Value] [<CR><LF>]
{GV=Value}[<CR><LF>]

4.6.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Value	The remote speaker volume of the connected HFP device. 00~0F as defined in HFP profile.		If the parameter is not given in command, the Module reports current value in response.

4.6.6.4. Examples:

Ex. 4.127. Upon a new AG connection, the remote HFP speaker volume is reported:

← …	← Other indications.
← GS=03,84AD8D0228F0<CR><LF>	← the Module is now connecting to a HFP device with address 84:AD:8D:02:28:F0.
← GV=0D<CR><LF>	← The remote speaker gain is 0x0D.
← …	← Other indications.

Ex. 4.128. When a remote connected HFP changes its speaker volume:

← GV=0C<CR><LF>	← The remote speaker gain is decreased to 0x0C.
← GV=0B<CR><LF>	← The remote speaker gain is decreased to 0x0B.

Ex. 4.129. Set the remote speaker gain when the AG profile is connected:

→ BC:GV=0B<CR><LF>	← Set the remote speaker gain to 0x0B.
← OK<CR><LF>	← The remote speaker gain is set to 0x0B.

Ex. 4.130. Set the remote speaker gain while the AG profile is not connected:

→ BC:GV=0B<CR><LF>	← Set the remote speaker gain to 0x0B.
← ER=01<CR><LF>	← The command is rejected.

4.7. HFP Messages

4.7.1. CH— Connect to a remote audio gateway

4.7.1.1. Description:

This command starts a connection to the remote Bluetooth HFP audio gateway device (generally, it's a mobile phone, tablet or laptop, etc.). If the Bluetooth address parameter is not presented, the Bluetooth module will attempt to connect to the last connected audio gateway device. It is also used as an indicator to report the error code of the connection attempt.

4.7.1.2. Syntax:

Synopsis:
[BC:]{CH}[=BdAddr][<CR><LF>]

{CH=Error,BdAddr}[<CR><LF>]

4.7.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the remote AG device to connect.	O	If no Bluetooth address is given, it connects the last connected AG devices.
Error	The error code of the connection attempt. Result code value: 01h—08h 01: sdp failed 02: slc failed 03: no more connection allowed 04: rfcomm connection failed 05: rfcomm channel not allocated 06: timeout 07: rejected 08: security failed 09: aborted for incoming connection	M	

4.7.1.4. Examples:

Ex. 4.131. To connect to the last connected AG device:

➔ BC:CH<CR><LF>	← *connect to the last connected AG device.*
← HS=01, 047295DF3C1C<CR><LF>	← *the Bluetooth module is now connecting to the last connected HFP device which address is 04:72:95:DF:3C:1C.*
← CC=00<CR><LF>	← *call state: idle.*
← HI=01<CR><LF>	← *network or service present.*
← HS=02, 047295DF3C1C<CR><LF>	← *the Bluetooth module is now connected to the AG device.*

Ex. 4.132. To connect to the specified HFP device:

➔ BC:CH=047295DF3C1C<CR><LF>	← *connect to the specified AG device: 04:72:95:DF:3C:1C.*
← CH=04, 047295DF3C1C<CR><LF>	← *the connection fails with error code 04.*
← HS=00<CR><LF>	← *the HFP state becomes idle.*

4.7.2. DH—Disconnect with the remote HFP device

4.7.2.1. Description:

This message requests the Module to disconnect with a connected Bluetooth AG device. If no Bluetooth address is given, it disconnects currently connected HFP devices.

4.7.2.2. Syntax:

Synopsis:
[BC:]{DH}[=BdAddr][<CR><LF>]

4.7.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
BdAddr	The Bluetooth address of the remote AG device.	O	If no Bluetooth address is given, it disconnects currently connected HFP devices.

4.7.2.4. Examples:

Ex. 4.133. To disconnect with currently connected HFP device:

➔ BC:DH<CR><LF> ← *disconnect currently connected HFP devices.*

← HS=00<CR><LF> ← *the HFP connection state becomes idle and connectable.*

Ex. 4.134. To disconnect to the specified A2DP source device:

➔ BC:DH=047295DF3C1C<CR><LF> ← *disconnect with the specified HFP device: 04:72:95:DF:3C:1C.*

← HS=00<CR><LF> ← *the HFP connection state becomes idle and connectable.*

4.7.3. HS—Query the state of HFP channel

4.7.3.1. Description:

This message queries or reports the state of HFP connection.

4.7.3.2. Syntax:

Synopsis:
[BC:]{HS}[<CR><LF>]
{HS=HfpState}[,BdAddr][<CR><LF>]

4.7.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
HfpState	The state of HFP connection. State Value: 00h-02h 00: idle and connectable. 01: connecting. 02: connected	M	
BdAddr	The Bluetooth address of remote OPP device.	O	

4.7.3.4. Examples:

Refer to the examples in 4.11.1.2.

4.7.4. AR—Answer or reject an incoming call

4.7.4.1. Description:

This message accepts or rejects an incoming call.

4.7.4.2. Syntax:

Synopsis:
[BC:]{AR=AcceptOrReject}[<CR><LF>]

4.7.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
AcceptOrReject	Accept or reject. Value: 00h or 01h 00h: Reject an incoming call 01h: Accept an incoming call	M	

4.7.4.4. Examples:

Ex. 4.135. To accept an incoming call:

← CC=01,01<CR><LF>	← *indicate there is an incoming call, and remote device support in-band ring.*
→ BC:AR=01<CR><LF>	← *accept the incoming call.*
← OK<CR><LF>	← *response from the module to indicate the command is adopted.*
← CC=04<CR><LF>	← *indicate the call state is activated.*

Ex. 4.136. To reject an incoming call:

← CC=01,01<CR><LF>	← *indicate there is an incoming call, and remote device support in-band ring.*
→ BC:AR=00<CR><LF>	← *reject the incoming call.*
← OK<CR><LF>	← *response from the module to indicate the command is adopted.*
← CC=00<CR><LF>	← *indicate the call is rejected and the call state is idle.*

4.7.5. CC—Call State

4.7.5.1. Description:

This message reports the call state of HFP connection.

4.7.5.2. Syntax:

Synopsis:
{CC=CallState}[,InBandRing][<CR><LF>]

4.7.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
CallState	The call state of HFP. Value: 00h—09h 00h: Idle 01h: Incoming Call. In this case, the second parameter will be available to indicate whether an in-band is supported. 02h: Incoming Held (Three Way Call) 03h: Outgoing Call 04h: Active Call 05h: Call Waiting (Three Way Call) 06h: Outgoing Call (Three Way Call) 07h: Held Active (Three Way Call) 08h: Held Remaining (Three Way Call) 09h: Conference Call (Three Way Call)	M	
InbandRing	The in-band ring state of the incoming call Value: 00h or 01h 00h: The remote HFP device does not support in-band ring. 01h: The remote HFP device supports in-band ring.	O	This parameter is only presented when the call state is incoming call.

4.7.5.4. Examples:

Refer to the examples in **Error! Reference source not found.**.

4.7.6. HU—Hang up an active call

4.7.6.1. Description:

This message hangs up an active call.

4.7.6.2. Syntax:

Synopsis:
BC:HU<CR><LF>

4.7.6.3. Parameter Description:

There is no parameter for this message.

4.7.6.4. Examples:

Ex. 4.137. To hang up an active call:

← CC=04<CR><LF> ← *indicate the call state is activated.*

→ BC:HU<CR><LF> ← *hang up the active call.*

← OK<CR><LF> ← *response from the module to indicate the command is adopted.*

← CC=00<CR><LF> ← *indicate the call state is activated.*

4.7.7. TC—Transfer a call

4.7.7.1. Description:

This message transfers a call's audio between the Bluetooth module and connected AG device (the mobile phone). When a call is answered by the message "AR", the audio is at the module by default.

4.7.7.2. Syntax:

Synopsis:
[BC:]{TC}[<CR><LF>]

4.7.7.3. Parameter Description:

There is no parameter for this message.

4.7.7.4. Examples:

Ex. 4.138. To transfer the active call's audio between the Bluetooth module and mobile phone:

→ BC:TC<CR><LF> ← *transfer the active call's audio to the mobile phone.*

← OK<CR><LF> ← *response from the module to indicate the command is adopted.*

→ BC:TC<CR><LF> ← *transfer the active call's audio to the Bluetooth mobile.*

← OK<CR><LF> ← *response from the module to indicate the command is adopted.*

4.7.8. MU—Mute or unmute the microphone

4.7.8.1. Description:

This message mutes or unmutes the microphone.

4.7.8.2. Syntax:

Synopsis:
[BC:]{MU}[<CR><LF>]

4.7.8.3. Parameter Description:

There is no parameter for this message.

4.7.8.4. Examples:

Ex. 4.139. To mute or unmute the microphone:

→ BC:MU<CR><LF> ← *mute the microphone.*

← OK<CR><LF> ← *response from the module to indicate the command is adopted.*

→ BC:MU<CR><LF> ← *unmute the microphone.*

← OK<CR><LF> ← *response from the module to indicate the command is adopted.*

4.7.9. LR—Last number redial

4.7.9.1. Description:

This message redials the last number.

4.7.9.2. Syntax:

Synopsis:
BC:LR<CR><LF>

4.7.9.3. Parameter Description:

There is no parameter for this message.

4.7.9.4. Examples:

Ex. 4.140. To redial the last number:

➔ BC:LR<CR><LF> ← *redial the last number.*

← OK<CR><LF> ← *response from the module to indicate the command is adopted.*

← CC=03<CR><LF> ← *indicate the call state is outgoing (dialing).*

4.7.10. HI—Network or service status

4.7.10.1. Description:

This message queries, sets or reports the network or service status on the connected AG device.

4.7.10.2. Syntax:

Synopsis:
{HI=Status}[<CR><LF>]

4.7.10.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Status	The network or service status. 00: not present 01: present	M	

4.7.10.4. Examples:

Ex. 4.141. The module reports network present on the connected AG device:

← HI=01<CR><LF> ← *report the network present.*

4.7.11. HV—Check or set the volume of HFP voice

4.7.11.1. Description:

This message queries, sets or reports the remote speaker volume of the connected AG device.

4.7.11.2. Syntax:

Synopsis:
[BC:]{HV}[=Volume][<CR><LF>]
{HV=Volume}[<CR><LF>]

4.7.11.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Volume	The remote speaker volume to set. 00~0F: Sixteen volume level as defined in HFP spec.	O	

4.7.11.4. Examples:

Ex. 4.142. To check current volume level of each HFP channel:

➔ BC:HV<CR><LF> ← *check the volume of HFP level.*

← HV=0F<CR><LF> ← *report the volume level of HFP channel 0 is 15(Dec).*

Ex. 4.143. To set the volume of HFP channel 0 to level 10:

➔ BC:HV=0A<CR><LF> ← *set the volume of HFP channel 0 to level 10.*

← OK<CR><LF> ← *response from the module to indicate the command is adopted.*

4.8. GATT Client Messages

Some of Flairmesh's Bluetooth modules and dongles support GATT client functions. Among these is the FMA120, which also serves as an Auracast™ transmitter. Users can leverage this functionality to access BLE peripheral devices while simultaneously broadcasting audio to Auracast™ sink devices. The current firmware of the FMA120 supports two GATT links to different peripherals simultaneously.

4.8.1. IQ—Inquiry Nearby Devices (Extended Scan)

4.8.1.1. Description:

This message scans nearby LE devices using the extended scan feature introduced in Bluetooth 5.0. For legacy LE devices that only include the device name in the scan response, please use the "IL" command as described in section 4.8.2.

4.8.1.2. Syntax:

Synopsis:

[BC:]{IQ}=[RssiThreshold,][UUIDfilter][<CR><LF>]
[BC:]{IQ=00}[<CR><LF>]

4.8.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
RssiThreshold	The RSSI threshold that an advertising report must exceed to be reported. Hex Value: xx, two's complement. Example: C0 means -64dBm	O	
UUIDfilter	The UUID16 or UUID128 to be used as a filter. Only advertisements containing the specified service will be reported. Hex Value: xxxx (UUID16) or xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx (UUID128)	O	
00	Stop the current scanning.	M	The inquiry doesn't have a timeout. Use the message with "00" to stop an ongoing inquiry.

4.8.1.4. Examples:

Ex. 4.144. Scan without any filter:

➔ BC:IQ<CR><LF> ←Start inquiry with a default RSSI limit of -70dBm.

← OK<CR><LF> ← Command accepted.

← LS=01<CR><LF> ← The LE central state becomes “scanning”.

← FD=00,5CB6CCB00F9F,D8,13FF0000<CR><LF> ← A device with public address (type 00) 5CB6CCB00F9F is found, the RSSI is -40dBm (indicated by the field “D8”), the advertising includes legacy flags indicating scannable/connectable.

← FD=01,48B7B67602F2,BF,13FF0000,S23 Ultra <CR><LF>← A device with a random address and the name “S23 Ultra”.

Ex. 4.145. Stop the current scanning:

➔ BC:IQ=00<CR><LF> ←Stop the current scanning.

← OK<CR><LF> ← Command accepted.

← LS=00<CR><LF> ← The LE central state becomes “idle”.

Ex. 4.146. Scan nearby devices advertising “Battery service” with a default RSSI threshold:

➔ BC:IQ=180F<CR><LF> ←Start the inquiry with a UUID16 filter of “180F”.

← OK<CR><LF> ← Command accepted.

← LS=01<CR><LF> ← The LE central state becomes “scanning”.

← FD=01,6EF4207B7974,BF,13FF0000,S23 Ultra <CR><LF>← A device with a random address and the name “S23 Ultra”.

Ex. 4.147. Scan nearby devices advertising “Battery service” with an RSSI threshold of -60dBm:

➔ BC:IQ=C4,180F<CR><LF> ←Start inquiry with an RSSI threshold of -60dBm.

Ex. 4.148. Scan nearby devices advertising UUID128 “F455A208597D11ECBF630242AC130002” with a default RSSI threshold -70dBm:

➔ BC:IQ=F455A208597D11ECBF630242AC130002<CR><LF> ←Start inquiry with a UUID128 filter.

Ex. 4.149. Scan nearby devices advertising UUID128 “F455A208597D11ECBF630242AC130002” with a RSSI threshold -60dBm:

➔ BC:IQ=C4,F455A208597D11ECBF630242AC130002<CR><LF>

←Start inquiry with a UUID128 filter with an RSSI threshold of -60dBm.

4.8.2. IL—Inquiry Nearby Devices (Legacy Active Scan)

4.8.2.1. Description:

This message initiates a scan for nearby LE devices using the legacy scan method defined prior to Bluetooth 5.0. Some older LE devices only include the device name in their scan response, making this message necessary to retrieve the name during the inquiry stage.

4.8.2.2. Syntax:

Synopsis:
[BC:]{IL}=[RssiThreshold,][UUIDfilter][<CR><LF>]
To stop a legacy scan, please use [BC:]{IQ=00}[<CR><LF>]

4.8.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
RssiThreshold	The RSSI threshold that an advertising report must exceed to be reported. Hex Value: xx, two's complement. Example: C0 means -64dBm	O	
UUIDfilter	The UUID16 or UUID128 to be used as a filter. Only advertisements containing the specified service will be reported. Hex Value: xxxx (UUID16) or xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx (UUID128)	O	

4.8.2.4. Examples:

Ex. 4.150. Scan a legacy/active scan without any filter:

➔ BC:IL<CR><LF> ←Start inquiry with a default RSSI limit of -70dBm.

← OK<CR><LF> ← Command accepted.

← LS=01<CR><LF> ← The LE central state becomes “scanning”.

← FD=00,5CB6CCB00F9F,D8,04000000,FMA100<CR><LF>

← A device with a public address and the name “FMA100”, and this is a “scan response”.

← FD=01,48B7B67602F2,BF,00000000,S23 Ultra <CR><LF>

← A device with a random address and the name “S23 Ultra”, and this is a “connectable undirected” advertisement.

Ex. 4.151. Stop the current legacy scanning:

➔ BC:IQ=00<CR><LF> ←Stop the current scanning.

← OK<CR><LF> ← Command accepted.

← LS=00<CR><LF> ← The LE central state becomes “idle”.

Ex. 4.152. Scan nearby devices advertising “Battery service” with default RSSI threshold:

➔ BC:IL=180F<CR><LF> ←Start the inquiry with a UUID16 filter of “180F”.

← OK<CR><LF> ← Command accepted.

← LS=01<CR><LF> ← The LE central state becomes “scanning”.

← FD=01,6EF4207B7974,BF,00000000,S23 Ultra <CR><LF>← A device with a random address and the name “S23 Ultra”.

Ex. 4.153. Scan nearby devices advertising “Battery service” with RSSI threshold of -60dBm:

➔ BC:IL=C4,180F<CR><LF> ←Start active scan with an RSSI threshold of -60dBm.

Ex. 4.154. Scan nearby devices advertising UUID128 “F455A208597D11ECBF630242AC130002” with a default RSSI threshold -70dBm:

➔ BC:IL=F455A208597D11ECBF630242AC130002<CR><LF> ←Start active scan with a UUID128 filter.

Ex. 4.155. Scan nearby devices advertising UUID128 “F455A208597D11ECBF630242AC130002” with a RSSI threshold -60dBm:

➔ BC:IQ=C4,F455A208597D11ECBF630242AC130002<CR><LF>

←Start active scan with a UUID128 filter with an RSSI threshold of -60dBm.

4.8.3. FD—Found Devices

4.8.3.1. Description:

This message reports the found device in an LE scan.

4.8.3.2. Syntax:

Synopsis:
{FD=addrType,addr,RSSI,advInfoExtended,deviceName}[<CR><LF>]
{FD=addrType,addr,RSSI,advInfoLegacy,deviceName}[<CR><LF>]

4.8.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
addrType	The type of the address of the found device. 00: Public address. 01: Random address.	M	
addr	The address of the found device.	M	
RSSI	The RSSI value of the found device. Hex Value: xx, two's complement.	M	

	Example: C0 means -64dBm		
advInfoExtended	The advertisement information for an extended scan report consists of 8 hex values, including the advertising type (2 hex), SID of the advertising set (2 hex), and the interval of the periodic advertising (4 hex). The valid advertising type is a bit map containing the following bit fields. bit 0 - Connectable Advertising bit 1 - Scannable Advertising bit 2 - Directed Advertising bit 3 - Scan Response bit 4 - Legacy advertising PDUs used	M	
advInfoLegacy	The advertisement information for a legacy scan report, totally 8 hex value includes the advertising type (2 hex) and the remaining 6 hex values are all 0. The valid advertising type: 00: connectable undirected. 01: connectable directed. 02: discoverable. 03: non-connectable. 04: scan response.	M	
deviceName	The device name of the found device.	O	May not be included if no device name is found.

4.8.3.4. Examples:

Ex. 4.156. Scan without any filter:

➔ BC:IQ<CR><LF> ←Start inquiry with a default RSSI limit of -70dBm.

← OK<CR><LF> ← Command accepted.

← LS=01<CR><LF> ← The LE central state becomes “scanning”.

← FD=00,5CB6CCB00F9F,D8,13FF0000<CR><LF>

← A connectable/scannable legacy advertising PDU with RSSI -40dBm, there’s no adv. set, not a periodic one.

Ex. 4.157. Scan a legacy/active scan without any filter:

➔ BC:IL<CR><LF> ←Start inquiry with a default RSSI limit of -70dBm.

← OK<CR><LF> ← Command accepted.

← LS=01<CR><LF> ← The LE central state becomes “scanning”.

← FD=00,5CB6CCB00F9F,D8,04000000,FMA100<CR><LF> ←A scan response from a device with RSSI -40dBm.

← FD=01,48B7B67602F2,BF,00000000,S23 Ultra <CR><LF>← A connectable undirect advertisement with RSSI -65dBm.

4.8.4. LS—LE Central State

4.8.4.1. Description:

This message reports the current LE central state.

4.8.4.2. Syntax:

Synopsis:
{LS=state}[<CR><LF>]

4.8.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
state	A string contains two characters. 00: Idle. 01: Scanning. 02: Connecting. 03: Pairing.	M	

4.8.4.4. Examples:

Ex. 4.158. Inquire nearby devices and connect to a found device:

➔ BC:IL<CR><LF> ←Start inquiry with a default RSSI limit of -70dBm.
← OK<CR><LF> ← Command accepted.
← LS=01<CR><LF> ← The LE central state becomes “scanning”.
← FD=01,7E51875C1872,C6,13FF0000,S23 Ultra<CR><LF> ← Found a device with the name “S23 Ultra”.
➔ BC:IQ=00<CR><LF> ← Stop the current scan.
← OK<CR><LF> ← Command accepted.
← LS=00<CR><LF> ← LE central state becomes “idle”.
➔ BC:GT=01,7E51875C1872<CR><LF> ← Connect to the found device with address “01,7E51875C1872”.
← LS=02,01,7E51875C1872<CR><LF> ← LE central state becomes “connecting”.
← LS=03,01,7E51875C1872<CR><LF> ← LE central state becomes “pairing”.
← GT=01,01,7E51875C1872,00100000<CR><LF> ← Pairing with remote device, connection ID is “00100000”.
← LM=00100000,007C<CR><LF> ← The MTU size of the connection “00100000” is 124 bytes.
← SD=00100000,1801,0001,0009<CR><LF> ← A service with UUID16 0x1801, handle range of 0001~0009.
← SD=00100000,1800,0014,001A<CR><LF> ← A service with UUID16 0x1800, handle range of 0001~0009.
← SD=00100000,1849,0028,004D<CR><LF> ← A service with UUID16 0x1849, handle range of 0028~004D.
← SD=00100000,184C,005A,007E<CR><LF> ← A service with UUID16 0x184C, handle range of 005A~007E.
← SD=00100000,1855,0082,FFFF<CR><LF> ← A service with UUID16 0x1855, handle range of 0082~FFFF.
← SD=00100000,0000<CR><LF> ← The end of the list of found services.
← PI=00,8C6A3BA53608<CR><LF> ← Pairing successfully with the remote device, the public address is 8C6A3BA53608.
← GT=02,01,7E51875C1872,00100000<CR><LF> ← Connected with remote device, connection ID is “00100000”.
← LS=00<CR><LF> ← LE central state becomes “idle”.
← LP=01,7E51875C1872,0018,0000<CR><LF>
← The connection parameters have been updated to an interval of 30ms with zero slave latency.

4.8.5. GT—GATT Connection State

4.8.5.1. Description:

This message can be used to set up a new GATT connection with a remote address. It also reports the connection state when it changes.

4.8.5.2. Syntax:

Synopsis:
[BC:]{GT}[<CR><LF>]
[BC:]{GT=addrType,addr}[<CR><LF>]
{GT=state}[,addrType,addr,connectionID][<CR><LF>]

4.8.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
No param	If no parameters are provided with this message, the module will respond with the current states for all LE central and GATT connection instances.		
state	Connection state. 00: idle 01: pairing 02: Connected	M	
addrType	The type of the address of the remote device. 00: Public address. 01: Random address.	O	Applicable when there's a connection.
addr	The address of the remote device.	O	Applicable when there's a connection.
connectionID	An 8-hex value representing a dynamically allocated connection ID.	O	Applicable when there's a connection.

4.8.5.4. Examples:

Please refer to examples in 4.8.4.4.

4.8.6. SD—Service Discovered

4.8.6.1. Description:

After a GATT connection is established, the GATT client on the central device will automatically begin discovering all primary services on the remote GATT server. The discovered services are reported by this message.

4.8.6.2. Syntax:

Synopsis:

{SD=connectionID,UUID,handleStart,handleEnd}[<CR><LF>]
{SD=connectionID,0000}[<CR><LF>]

4.8.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated connection ID.	M	
UUID	UUID16 or UUID128 of the found service. 4-hex value for UUID16. 32-hex value for UUID128.	M	
handleStart	The handle at which a service begins. 4-hex value.	M	
handleEnd	The handle at which a service ends. 4-hex value.	M	
0000	A constant 0000 following a connection ID signifies the end of a GATT service list.	M	

4.8.6.4. Examples:

Please refer to examples in 4.8.4.4.

4.8.7. CD—Characteristic Discovery

4.8.7.1. Description:

This message is used to register and set up access to a range of characteristic handles for subsequent operations. It is also used to report all the registered handles and their properties. This message must be sent before accessing any characteristics.

4.8.7.2. Syntax:

Synopsis:
[BC:]{CD=connectionID,handleStart,handleEnd}[<CR><LF>]
{CD=connectionID,UUID,handleDeclare,handleValue,property }[<CR><LF>]
{CD=connectionID,0000}[<CR><LF>]

4.8.7.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated connection ID.	M	
handleStart	The handle at which a requested access range begins. 4-hex value.	M	
handleEnd	The handle at which a requested access range ends.	M	

	4-hex value.		
UUID	UUID16 or UUID128 of the registered characteristic. 4-hex value for UUID16. 32-hex value for UUID128.	M	
handleDeclare	The handle of the declaration of a characteristic. 4-hex value.	M	
handleValue	The handle of the value of a characteristic. 4-hex value.	M	
property	A bit map of characteristic property. Bit 0: configure broadcast. Bit 1: read. Bit 2: Write without response. Bit 3: Write with response. Bit 4: Notify. Bit 5: Indicate. Bit 6: Signed write. Bit 7: has extended property descriptor.	M	
0000	A constant 0000 following a connection ID signifies the end of a registered characteristic list.	M	

4.8.7.4. Examples:

Ex. 4.159. Register the access handle range of characteristics:

← GT=02,01,6F2F20072199,00200000<CR><LF> ← Connected with remote device, connection ID is “00100000”.

➔ BC:CD=00200000,0001,FFFF<CR><LF> ← Register access to all characteristics from 0001 to FFFF.

← CD=00200000,2A05,0002,0003,20<CR><LF> ← A characteristic with UUID16 “2A05” and the property ““indicate”.

…

← GT=00200000,2A00,0015,0016,02 <CR><LF> ← Device name characteristic, the value handle is 0016.

….

← CD=00200000,0000<CR><LF> ← The end of the characteristic list.

4.8.8. DD—Descriptor Discover

4.8.8.1. Description:

This message can be used to discover and report characteristic descriptors within a given handle range.

4.8.8.2. Syntax:

Synopsis:
[BC:]{DD=connectionID,handleStart,handleEnd}[<CR><LF>]
{DD=connectionID,UUID16,handle}[<CR><LF>]
{DD=connectionID,0000}[<CR><LF>]

4.8.8.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated connection ID.	M	
handleStart	The handle at which a requested discovery range begins. 4-hex value.	M	
handleEnd	The handle at which a requested discovery range ends. 4-hex value.	M	
UUID16	The UUID16 defined in Bluetooth Spec. for characteristic descriptors. 290A: Value Trigger Settings, usually used with a characteristic with “Notify” property. An invalid UUID16 will be returned for non-descriptor handles.	M	
handle	The handle of the found descriptor.	M	
0000	A constant 0000 following a connection ID signifies the end of a descriptor list.	M	

4.8.8.4. Examples:

Ex. 4.160. Set the encryption key:

➔ BC:DD=00100000,0010,0011<CR><LF> ← Discover all descriptors in handle range of 0010~0011.
← DD=00100000,DDC7,0010<CR><LF> ← “DDC7” is not a valid descriptor UUID16.
← DD=00100000,2902,0011<CR><LF> ← Found a descriptor “Client Characteristic Configuration” on handle 0011.
← DD=00100000,0000<CR><LF> ← The end of the descriptor list.

4.8.9. CV—Characteristic Value

4.8.9.1. Description:

This message can be used to read/write the value of a characteristic. It is also used when a notification is received from the remote GATT server. When used for a write operation, the write type is "write without response".

4.8.9.2. Syntax:

Synopsis:
[BC:]{CV=connectionID,handleValue}[<CR><LF>]
{CV=connectionID,handleValue,bytesInPrintableFormat }[<CR><LF>]
[BC:]{CV=connectionID,handleValue,bytesInPrintableFormat}[<CR><LF>]

4.8.9.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated	M	

	connection ID.		
handleValue	The handle at which a read/write access is requested. 4-hex value.	M	Please do not try to read/write to the characteristic's declaration handle.
bytesInPrintableFormat	A hex-value string. Each pair of hex values represents a byte of the read/write data from/to the characteristic.	O	Applicable for a write or notification.

4.8.9.4. Examples:

Ex. 4.161. Read the device name characteristic:

← CD=00100000,2A00,0008,0009,02<CR><LF> ← Name characteristic value handle 0009 for connection "00100000".

…

← CD=00200000,2A00,0015,0016,02<CR><LF> ← Name characteristic value handle 0016 for connection "00200000".

→ BC:CV=00100000,0009<CR><LF> ← Read the device name for connection "00100000".

← CV=00100000,0009,464D41313030<CR><LF> ← Device name is "FMA100" for connection "00100000".

→ BC:CV=00200000,0016<CR><LF> ← Read the device name for connection "00200000".

← CV=00200000,0009,53323320556C747261<CR><LF> ← Device name is "S23 Ultra" for connection "00200000".

Ex. 4.162. Write to a characteristic without response, ensure that the property has the bit "04" set:

→ BC:CV=00100000,000E,3031<CR><LF> ← Write two bytes "0x30 0x31" to handle 000E for connection "00200000".

← WR=00100000,000E,0000<CR><LF> ← The operation was successful, indicated by an error code of "0000".

4.8.10. CW—Characteristic Value Write With Response

4.8.10.1. Description:

This message can be used to write the value of a characteristic with the write type "write with response".

4.8.10.2. Syntax:

Synopsis:
[BC:]{CW=connectionID,handleValue,bytesInPrintableFormat}[<CR><LF>]

4.8.10.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated connection ID.	M	
handleValue	The handle at which a read/write access is requested. 4-hex value.	M	Please do not try to read/write to the characteristic's declaration handle.
bytesInPrintableFormat	A hex-value string. Each pair of hex values represents a	M	

	byte of the write data to the characteristic.		

4.8.10.4. Examples:

Ex. 4.163. Set the encryption key:

➔ BC:CW=00200000,0011,0100<CR><LF> ← Write two bytes of “0x01 0x00” to a descriptor to enable notification.

← WR=00200000,0011,0000<CR><LF> ← The operation was successful, indicated by an error code of “0000”.

4.8.11. WR—Write Operation Result

4.8.11.1. Description:

This message serves as an indication of the result of a write operation.

4.8.11.2. Syntax:

Synopsis:
{WR=connectionID,handle,errorCode}[<CR><LF>]

4.8.11.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated connection ID.	M	
handle	The handle at which a write access has been requested. 4-hex value.	M	
errorCode	0000 indicates a successful operation, other values indicate errors occurred.	M	A common error is writing to a handle with an incorrect write type.

4.8.11.4. Examples:

Please refer to samples in 4.8.9.4 and 4.8.10.4.

4.8.12. PI—Pairing Indication

4.8.12.1. Description:

This message serves as an indication of the result of a pairing process.

4.8.12.2. Syntax:

Synopsis:
{PI=result,addr}[<CR><LF>]

4.8.12.3. Parameter Description:

Parameter	Description	Mandatory	Comments

		or Optional	
result	A 2-hex value. 00: pairing successful. Other values means error occurred.	M	
addr	The Bluetooth address of paired device. The resolved public address will be used in the indication for a device using a resolvable random address. For subsequent connections with the same device, the public address will be employed, provided the pairing information remains intact.	M	

4.8.12.4. Examples:

Please refer to the sample in 4.8.4.4.

4.8.13. LM—Link GATT MTU Size

4.8.13.1. Description:

This message serves as an indication of the GATT MTU size of a connection.

4.8.13.2. Syntax:

Synopsis:
{LM=connectionID,size}[<CR><LF>]

4.8.13.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated connection ID.	M	
size	A 4-hex value representing the mtu size.	M	

4.8.13.4. Examples:

Please refer to the sample in 4.8.4.4.

4.8.14. LP—Link Connection Parameter

4.8.14.1. Description:

This message serves as an indication of the updated parameters of a LE connection.

4.8.14.2. Syntax:

Synopsis:
{LP=connectionID,interval,slaveLatency}[<CR><LF>]

4.8.14.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
connectionID	An 8-hex value representing a dynamically allocated connection ID.	M	
interval	A 4-hex value representing the interval. Unit is 1.25ms.	M	
slaveLatency	A 4-hex value representing the slave latency.	M	

4.8.14.4. Examples:

Please refer to the sample in 4.8.4.4.

4.9. Auracast™ Receiver Messages

Some of Flairmesh's Bluetooth modules support the Auracast™ receiver function. These modules can typically scan for Auracast™ transmitters and receive the broadcast without needing a Public Broadcast Assistant (PBA). This process is facilitated by the messages described in this chapter.

4.9.1. BI—Inquiry Nearby Broadcasts

4.9.1.1. Description:

This message scans for nearby broadcast transmitters and reports the detected broadcasts.

4.9.1.2. Syntax:

Synopsis:
[BC:]{BI}=[timeout][<CR><LF>]
[BC:]{BI=00}[<CR><LF>]
{BI=scanState}[<CR><LF>]
{BI=addrType,addr,RSSI,broadcastIDs,broadcastName}[<CR><LF>]

4.9.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
timeout	The scan will time out and stop after the specified duration. Hex Value: xx in seconds. Example: 10 means 16 seconds.	O	If no timeout is given, the inquiry can be stopped by a "BC:BI=00".
00	Terminate the ongoing scan.	M	
scanState	The scan state. 00: the scan request is accepted. 01: the scan request is failed.		

	02: reserved. 03: the scan is timeout. 04: the scan is in progress. 05: the scan is stopped.		
addrType	The type of the address of the found broadcast device. 00: Public address. 01: Random address.	M	
addr	The address of the found broadcast device.	M	
RSSI	The RSSI value of the detected device. Hex Value: xx, two's complement. Example: C0 means -64dBm	M	
broadcastIDs	This field contains two IDs for a broadcast. The first two hexadecimal numbers represent the advertising set ID, while the last six hexadecimal numbers represent the broadcast ID. These IDs are necessary to initiate synchronization from the module. For example, in the ID 0F001746, the advertising set ID is 0F, and the broadcast ID is 001746.	M	
broadcastName	The name of the detected broadcast.	M	

4.9.1.4. Examples:

Ex. 4.164. Perform a scan without a timeout:

➔ BC:BI<CR><LF> ← Initiate a scan for nearby broadcasts.

← OK<CR><LF> ← Command accepted.

← BI=00<CR><LF> ← The start scan request has been accepted.

← BI=04<CR><LF> ← The scan is currently in progress.

← BI=00,5CB6CCB18E70,D5,0F001746,FlooGoo_B18E70<CR><LF>

← A device with a public address (type 00) 5CB6CCB18E70 is broadcasting, the RSSI is -43dBm, the advertising set ID is 0F and the broadcast ID is 001746, the broadcast name is “FlooGoo_B18E70”.

➔ BC:BI=00<CR><LF> ← Stop the ongoing scan.

← OK<CR><LF> ← Command accepted.

← BI=00<CR><LF> ← The stop scan request has been accepted.

← BI=05<CR><LF> ← The scan has stopped.

Ex. 4.165. Scan with a timeout of 5 seconds:

➔ BC:BI=05<CR><LF> ← Initiate a scan for nearby broadcasts with a timeout of 5 seconds.

← OK<CR><LF> ← Command accepted.

← BI=00<CR><LF> ← The start scan request has been accepted.

← BI=04<CR><LF> ← The scan is currently in progress.

← BI=00,5CB6CCB18E70,D5,0F001746,FlooGoo_B18E70<CR><LF>

← BI=03<CR><LF> ← The scan has timed out after 5 seconds.

4.9.2. BA—Broadcast State

4.9.2.1. Description:

This message serves to synchronize with or disconnect from a broadcast. It also reports the current state of broadcast reception.

4.9.2.2. Syntax:

Synopsis:
[BC:]{BA}=[broadcastIDs][<CR><LF>]
[BC:]{BA=00}[<CR><LF>]
{BA=sourceID,broadcastIDs,syncState,encryptionState,bisState}[<CR><LF>]

4.9.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
broadcastIDs	This field contains two IDs for a broadcast. The first two hexadecimal numbers represent the advertising set ID, while the last six hexadecimal numbers represent the broadcast ID. These IDs are necessary to initiate synchronization from the module. For example, in the ID 0F001746, the advertising set ID is 0F, and the broadcast ID is 001746.	O	If no parameter is provided, the module will report the current receiving state.
00	Stop receiving the current broadcast.	M	
sourceID	An ID assigned by the module identify a broadcast. This ID must be used when providing the corresponding broadcast key if the module requests it. 01~FF	M	
syncState	The state of periodic advertising synchronization. 00: not synchronized. 01: synchronize information requested. 02: synchronized. 03: failed to synchronize. 04: no PAST.	M	
decryptionState	The decryption state. 00: not encrypted. 01: code requested. 02: decrypting. 03: bad code.	M	
bisState	A bit map of the synchronized BIS (broadcast isochronous stream). 00: not streaming. 03: both left and right channel are streaming.	M	

4.9.2.4. Examples:

Ex. 4.166. Start receiving the broadcast with advSID “0F” and broadcastID “008A2A”:

...

← BI=00,5CB6CCB18E70,D5,0F008A2A,FlooGoo_B18E70<CR><LF>

← A device with a public address (type 00) 5CB6CCB18E70 is broadcasting. The RSSI is -43dBm, the advertising set ID is 0F, and the broadcast ID is 008A2A, the broadcast name is “FlooGoo_B18E70”.

...

→ BC:BA=0F008A2A<CR><LF>	← Start receiving the broadcast with advSID “0F” and broadcastID “008A2A”.
← OK<CR><LF>	← Command accepted.
← BA=01,0F0082A2A,02,00,00<CR><LF>	← Synced with a broadcast with source ID of “01”, the broadcast is not encrypted, and no BIS is streaming.
← BA=01,0F0082A2A,02,00,03<CR><LF>	← Synced with the broadcast, it is not encrypted, and both two BISs (left and right) are streaming.

Ex. 4.167. Stop receiving the current broadcast.

→ BC:BA=00<CR><LF>	← Stop receiving the current broadcast.
← OK<CR><LF>	← Command accepted.
← BA=01,0F0082A2A,02,00,00<CR><LF>	← Synced with the broadcast, it is not encrypted, and no BIS is streaming.
← BA=01,0F0082A2A,00,00,00<CR><LF>	← Not synced with the broadcast with source ID of “01”.

4.9.3. BK—Broadcast Encryption Code

4.9.3.1. Description:

This message provides the encryption code for decrypting a broadcast. When a receiving broadcast is encrypted, the module uses the “BA” message to indicate that a decryption code is required. At that point, the “BK” message can be used to set the code. If the code is correct, the receiving state will change to streaming; if not, another “BA” will be prompted for another code.

4.9.3.2. Syntax:

Synopsis:
[BC:]{BK=sourceID,code}[<CR><LF>]

4.9.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
sourceID	The ID of the broadcast source to be decrypted. Hex Value: xx. Example: 01, an ID of 1.	M	
code	The code to be used for decryption. Usually, it’s a text string that functions like a PIN code, with a maximum allowed length of 16 bytes.	O	If no parameter is provided, the module reports the current

	Example: 1234. The ASCII string "1234" represents a code consisting of 4 bytes: "0x31 0x32 0x33 0x34".		receiving state.

4.9.3.4. Examples:

Ex. 4.168. Start receiving an encrypted broadcast with advSID “0F” and broadcastID “0095A4”:

...

← BI=00,5CB6CCB18E70,D5,0F0095A4,FlooGoo_B18E70<CR><LF>

← A device with a public address (type 00) 5CB6CCB18E70 is broadcasting. The RSSI is -43dBm, the advertising set ID is 0F, and the broadcast ID is 0095A4, the broadcast name is “FlooGoo_B18E70”.

...

→ BC:BA=0F,0095A4<CR><LF>	← Start receiving the broadcast with advSID “0F” and broadcastID “0095A4”.
← OK<CR><LF>	← Command accepted.
← BA=01,00,5CB6CCB18E70,02,00,00<CR><LF>	← Synced with a broadcast with source ID of “01”, the broadcast is not encrypted, and no BIS is streaming.
← BA=01,00,5CB6CCB18E70,02,01,00<CR><LF>	← The broadcast is synced and encrypted, and a code is requested to decrypt.
→ BC:BK=01,1211<CR><LF>	← Set the decryption code “1211” for the broadcast with ID “01”.
← OK<CR><LF>	← Command accepted.
← BA=01,00,5CB6CCB18E70,02,01,03<CR><LF>	← Synced with the broadcast, it is encrypted and code is requested, the BISs are streaming.
← BA=01,00,5CB6CCB18E70,02,02,03<CR><LF>	← Synced with the broadcast, it is decrypted successfully, the BISs are streaming.

4.9.4. BN—Preset Broadcast Name

4.9.4.1. Description:

This message sets the target broadcast name that the module will automatically search for and connect to after powering on.

4.9.4.2. Syntax:

Synopsis:
[BC:]{BN=broadcastName}[<CR><LF>]

4.9.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
broadcastName	The name of the target broadcast. Minimum 4 bytes, maximum 32 bytes.	M	

4.9.4.4. Examples:

Ex. 4.169. Set the target broadcast name to “FlooGoo_Auracast”:

➔ BC:BN=FlooGoo_Auracast<CR><LF> ← Set the preset broadcast name to “FlooGoo_Auracast”.

← OK<CR><LF> ← Command accepted.

4.9.5. BE—Preset Broadcast Decryption Code

4.9.5.1. Description:

Sets the decryption code for the target broadcast that the module will use when automatically searching for and connecting to it after power on.

4.9.5.2. Syntax:

Synopsis:
[BC:]{BE=code}[<CR><LF>]

4.9.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
code	The decryption code. Maximum 16 bytes.	M	

4.9.5.4. Examples:

Ex. 4.170. Set the target broadcast’s decryption key to “1234”:

➔ BC:BE=1234<CR><LF> ← Start the preset broadcast decryption code to “1234”.

← OK<CR><LF> ← Command accepted.

4.9.6. BV—Broadcast Stream Volume

4.9.6.1. Description:

Use this message to set or get the broadcast volume. If a preset broadcast is defined using “BN”, this message can also specify the default volume to apply when automatically tuning into it.

4.9.6.2. Syntax:

Synopsis:
[BC:]{BV}[=volume][<CR><LF>]
{BV=volume}[<CR><LF>]

4.9.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments

volume	The volume of the current receiving broadcast. Hex Value: 00~FF.	O	If no parameter is provided, the module reports the current receiving volume.

4.9.6.4. Examples:

Ex. 4.171. Get and set the volume of a receiving broadcast:

...

← BS=01,00,5CB6CCB18E70,02,02,03<CR><LF> ← Synced with the broadcast, it is decrypted successfully, the BISs are streaming.

...

→ BC:BV<CR><LF> ← Get the current volume.

← BV=C8<CR><LF> ← The current volume is 0xC8.

← BC:BV=D0<CR><LF> ← Set the volume to 0xD0.

← OK<CR><LF> ← Command accepted.

4.10. Auracast™ Transmitter Messages

4.10.1. BN—Preset Broadcast Name

Please refer to 4.9.4.

4.10.2. BE—Preset Broadcast Encryption Code

Please refer to 4.9.5.

4.10.3. BI—Broadcast IDs

4.10.3.1. Description:

Some of Flairmesh's Bluetooth modules allow users to set and read the broadcast ID information of the current broadcast. This information can be used to add the broadcast source to a connected broadcast receiver's BASS server. For details on accessing a remote GATT server, please refer to Chapter 4.8.

This message set and read the advertising SID and broadcast IDs of the current broadcast.

4.10.3.2. Syntax:

Synopsis:
[BC:]{BI}[<CR><LF>]
[BC:]{BI=broadcastID[<CR><LF>]

{BI=advAndBroadcastID}[<CR><LF>]

4.10.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
broadcastID	The broadcast ID for a BASS “Add Source” Operation. Hex Value: xxxxxx. Default value: the lap of the module’s Bluetooth address.	O	If no broadcastID is given, the current advertising SID and broadcast ID are reported.
advAndBroadcastID	This field contains two IDs for a broadcast. The first two hexadecimal numbers represent the advertising set ID, while the last six hexadecimal numbers represent the broadcast ID. These IDs are necessary to initiate synchronization from the module. For example, in the ID 0F001746, the advertising set ID is 0F, and the broadcast ID is 001746.	M	

4.10.3.4. Examples:

Ex. 4.172. Read the current IDs:

➔ BC:BI<CR><LF> ← Read the current IDs.

← BI=0FB18E70<CR><LF> ← The advertising SID is “0F” and the broadcast ID is “B18E70”.

Ex. 4.173. Set the broadcast ID (the advertising SID cannot be set, which is automatically allocated by the firmware):

➔ BC:BI=123456<CR><LF> ← Set the broadcast ID to “123456”.

← OK<CR><LF> ← Command accepted.

To send an “Add Source” command to a connected BASS server, use the advertising ID and broadcast ID along with the following parameters to add the broadcast to the connected device,
Bluetooth address type: 0x00
Bluetooth address: can be accessed by “BC:AD” message
PA_Sync: 0x02
PA_Interval: 0x168
Num_subgroups: 1
BIS_sync: 0x03 (in 4 bytes: 0x03, 0x00, 0x00, 0x00)

4.11. BLE/Broadcast Messages

Broadcast is quite valuable for local networks that need to transfer data of small size between multiple endpoints. Using broadcast messages with Flairmesh’s long-range Bluetooth modules can build a mesh-like network quickly and securely.

4.11.1. BK— Set the Encryption Key

4.11.1.1. Description:

To build a secure network, broadcast data is encrypted by using standard AES128 CBC algorithms. Broadcast senders and receivers shall be set to use the same encryption key to be able to communicate with their peers.
The key can only be written and never readable.

4.11.1.2. Syntax:

Synopsis:
[BC:]{BK=Word0,…,Word7}[<CR><LF>]

4.11.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Word0 .. Wordn	Each of the words is a 4 ASCII character representation of the hex value of the key. Hex Value: xxxx	M	

4.11.1.4. Examples:

Ex. 4.174. Set the encryption key:

➔ BC:BK=31F6,803B,E9E2,1F84,A9B7,1F40,0A2C,4DC1<CR><LF> ← set the 128bits encryption key to "31F6803BE9E21F84A9B71F400A2C4DC1".

← OK<CR><LF> ← Command accepted.

4.11.2. BC—Broadcast Role

4.11.2.1. Description:

This message queries, sets or reports the role of the module in a broadcast network.
A module can switch its role in runtime. When it's in the sender role, the power consumption when idle is very low since there are no data to broadcast. When it's in the receiver or transceiver role, the average current consumption is below 3mA for FMB100 and below 4mA for FMB110 when deep sleep is enabled.

4.11.2.2. Syntax:

Synopsis:
[BC:]{BC}[=Role][<CR><LF>]
{BC=Role}[<CR><LF>]

4.11.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments

Role	The role of the module in a broadcast network. 00: receiver. 01: sender. 02: transceiver. 03: off. 04: as receiver temporarily, reset upon power on 05: as sender temporarily, reset upon power on 06: as transceiver temporarily, reset upon power on 07: off temporarily, reset upon power on	O	If no parameter is given, it reports the current value.

4.11.2.4. Examples:

Ex. 4.175. Set the role as sender:

➔ BC:BC=01<CR><LF> ← *Set the role as sender.*

← OK<CR><LF> ← *done.*

Ex. 4.176. Query the current role:

➔ BC:BC<CR><LF> ← *Query the current role.*

← BC=00<CR><LF> ← *It's a receiver.*

4.11.3. BG—Broadcast Group

4.11.3.1. Description:

Within a broadcast network, endpoints are divided into groups. Modules can communicate within the same group

4.11.3.2. Syntax:

Synopsis:
[BC:]{BG=Group}[<CR><LF>]
{BG=Group<CR><LF>]

4.11.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Group	Only a 3-byte length non-zero value/string is allowed. The group name is not encrypted in the air, it's used as a filter or channel number by endpoints. Default Value: ASCII string "FMB"	M	

4.11.3.4. Examples:

Ex. 4.177. Query the current broadcast group name:

➔ BC:BG<CR><LF> ← *Read the current broadcast group name.*

← BG=FMB<CR><LF> ← *The current group name is "FMB".*

Ex. 4.178. Set the broadcast group name:

➔ BC:BG=001<CR><LF> ← *Set the broadcast group name to "001".*

← OK<CR><LF> ← *Command accepted and the group name is "001" now.*

4.11.4. BD—Broadcast Data

4.11.4.1. Description:

A sender can use this message to send data to all the receivers. Every receiver in the range also reports the broadcasted data with this message.

The maximum length of the sending data is 15 bytes. The received data is always 16 bytes which is a zero-padded copy of the sending data.

The host shall apply a counter or time stamp in the broadcast data to prevent replay attacks.

4.11.4.2. Syntax:

Synopsis:
[BC:]{BD=DataLen,Data}[<CR><LF>]
{BD=10,Data}<CR><LF>]

4.11.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
DataLen	The length in bytes of the data to be sent or received. Value: 00h-14h	M	
Data	The raw data.	M	

4.11.4.4. Examples:

Ex. 4.179. Send data to the network:

➔ BC:BD=01,a<CR><LF> ← *Send one byte data "a" to the network.*

← OK<CR><LF> ← *response from the module when the data has been sent out, so another sending can be taken.*

In the meanwhile, at the receiver side within the same network:

← BD=10,a<15 bytes of 0 padding><CR><LF> ← *"a" is received by a receiver.*

5. Appendix A: Examples

5.1. SPP Master

This example shows how to use the Module as a SPP Master device. A SPP master device means the device who will initiate the SPP connection to a remote SPP Slave device.

Ex. 5.1:

➔ BC:PF=01,00,00<CR><LF>	← configure the module profiles: 1 SPP channel, no HID profile supported. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
← AP=00<CR><LF>	← Indicate that the Module has performed a reboot and is ready now.
➔ BC:FT=FF,00,FF,0A,01,0078<CR><LF>	← configure the module features: enable the auto connection after power on. disable the auto connect after paired. enable auto reconnect after link lost. set the interval of auto reconnect to 10s. configure the discover mode as 01: auto discoverable when empty. configure the timeout of discoverable as 120 seconds. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
➔ BC:IQ<CR><LF>	← inquire the Bluetooth device. This command is not necessary if the host knows the Bluetooth address of remote device.
← AP=01<CR><LF>	← indicate the Module is now inquiring.
← IR=03<CR><LF>	← indicate there are 3 Bluetooth devices found.
← AP=00<CR><LF>	← indicate Module is now in idle.
← FD=02,00189600000A,FFC6,SPP_DEV<CR><LF>	← indicate the 3rd found device's address, RSSI and name.
← FD=01,00189600000B,FFC7<CR><LF>	← indicate the 2nd found device's address and the name is not gotten.
← FD=00,00189600000D,FFC8,BT_DEV_1<CR><LF>	← indicate the 1st found device's address and name.
➔ BC:CS=00189600000A<CR><LF>	← connect to the specified device 00:18:96:00:00:0A with the SPP profile.
← SS=01,00189600000A<CR><LF>	← the Module is now connecting to the specified SPP device which address is 00:18:96:00:00:0A.
← CS=00,00189600000A<CR><LF>	← connecting result: success.
← SS=02,00189600000A<CR><LF>	← the Module is now connected to the specified SPP device and working in Bypass mode, the Bypass channel is the SPP channel 0. The host can now exchange the raw data with the remote SPP device by sending and receiving data via the UART port.
……	
……	
	← Keep the UART port idle for 1 second.
➔ BC:BP=00,00,00<CR><LF>	← change the bypass mode to **Message Mode**.
	← Keep the UART port idle for 1 second.
← OK<CR><LF>	← Command accepted.
➔ BC:DS=00189600000A<CR><LF>	← disconnect with the specified device 00:18:96:00:00:0A.
← SS=00<CR><LF>	← the SPP channel 0 of Module is now disconnected and is connectable.

5.2. SPP Slave

This example shows how to use the Module as a SPP Slave device. A SPP Slave device means the device who is waiting for and will accept the SPP connection request from a remote SPP Master device.

Ex. 5.2:

➔ BC:PF=04,00,00<CR><LF>	← configure the module profiles: 4 SPP channels, no HID profile supported. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
← AP=00<CR><LF>	← Indicate that the Module has performed a reboot and is ready now.
➔ BC:FT=00,00,00,0A,01,0078<CR><LF>	← configure the module features: disable the auto connection after power on. disable the auto connect after paired. disable auto reconnect after link lost. set the interval of auto reconnect to 10s. configure the discover mode as 01: auto discoverable when empty. configure the timeout of discoverable as 120 seconds. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
➔ BC:MD=01<CR><LF>	← make Module discoverable. This command is not necessary if the remote device know the Bluetooth address of the Module.
← OK<CR><LF>	← Command accepted.
← SS=01,00189600000B<CR><LF>	← a remote SPP device is connecting to the Module. Its Bluetooth address is 00:18:96:00:00:0B.
← CS=00,00189600000B<CR><LF>	← connecting result: success.
← SS=02,00189600000B<CR><LF>	← the Module is now connected to the remote SPP device and working in Bypass mode, the Bypass channel is the SPP channel 0. The host can now exchange the raw data with the remote SPP device by sending and receiving data via the UART port.
……	
……	
← SS=00<CR><LF>	← the remote SPP device has disconnected with the Module. The Module is now disconnected and is connectable.

5.3. HID Device, ASCII Characters

This example shows how to use the Module as a HID Device(a keyboard) to send the ASCII characters to the HID Host(a computer, tablet, mobile phone, etc.).

Ex. 5.3:

➔ BC:PF=00,01,00<CR><LF>	← configure the module profiles: 1 HID channel, no SPP profile supported. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.

← AP=00<CR><LF>	← Indicate that the Module has performed a reboot and is ready now.
→ BC:FT=FF,00,FF,0A,01,0078<CR><LF>	← configure the module features: enable the auto connection after power on. disable the auto connect after paired. enable auto reconnect after link lost. set the interval of auto reconnect to 10s. configure the discover mode as 01: auto discoverable when empty. configure the timeout of discoverable as 120 seconds. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
→ BC:BP=02,00,00<CR><LF>	← change the bypass mode to **Bypass to HID ASCII Channel**.
← OK<CR><LF>	← Command accepted.
→ BC:CI<CR><LF>	← connect to the last connected HID host.
← IS=01,00189600ABCD<CR><LF>	← the Module is now connecting to the last connected HID host which address is 00:18:96:00:AB:CD.
← CI=00,00189600ABCD<CR><LF>	← connecting result: success.
← IS=02,00189600ABCD<CR><LF>	← the Module is now connected to the last connected HID host. The host can now send ASCII characters to the HID host by sending data to the Module via UART port.
→ Hello, this is Flairmesh!	← the ASCII characters need to send to remote HID device
……	
……	
	← Keep the UART port idle for 1 second.
→ BC:BP=00,00<CR><LF>	← change the bypass mode to **Message mode**.
	← Keep the UART port idle for 1 second.
← OK<CR><LF>	← Command accepted.
→ BC:DI<CR><LF>	← disconnect the HID channel.
← IS=00<CR><LF>	← the HID channel of the Module is now disconnected and is connectable.

5.4. HID Device, raw HID reports

This example shows how to use the Module as a HID Device(a keyboard, mouse, gamepad/joystick) to send the HID report to the HID Host(a computer, tablet, mobile phone, etc.).

Ex. 5.4:

→ BC:PF=01,01,00<CR><LF>	← configure the module profiles: 1 HID channel, 1 SPP profile supported. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
← AP=00<CR><LF>	← Indicate that the Module has performed a reboot and is ready now.
→ BC:FT=FF,00,FF,0A,01,0078<CR><LF>	← configure the module features: enable the auto connection after power on. disable the auto connect. enable auto reconnect after link lost.

	set the interval of auto reconnect to 10s. configure the discover mode as 01: auto discoverable when empty. configure the timeout of discoverable as 120 seconds. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
← OK<CR><LF>	← Command accepted.
→ BC:CI<CR><LF>	← connect to the last connected HID host.
← IS=01,00189600ABCD<CR><LF>	← the Module is now connecting to the last connected HID host which address is 00:18:96:00:AB:CD.
← CI=00,00189600ABCD<CR><LF>	← connecting result: success.
← IS=02,00189600ABCD<CR><LF>	← the Module is now connected to the last connected HID host. The host can now send ASCII characters to the HID host by sending data to the Module via UART port.
→ A1 01 00 00 04 00 00 00 00 00	← (shown in HEX mode) the keyboard report in raw data (binary), the key A is pressed
→ A1 01 00 00 00 00 00 00 00 00	← (shown in HEX mode) the keyboard report in raw data (binary), the pressed key is released
→ A1 02 10 00	← (shown in HEX mode) the consumer key report in raw data (binary), the Volume Up key is pressed
→ A1 02 00 00	← (shown in HEX mode) the consumer key report in raw data (binary), the pressed key is released
→ A1 03 01 00 00 00 00	← (shown in HEX mode) the mouse report in raw data (binary), the left button is pressed
→ A1 03 00 03 C0 FF 00	← (shown in HEX mode) the mouse report in raw data (binary), the mouse is move to upper-right, X = 3, Y = -4.
→ A1 03 00 00 00 00 FF	← (shown in HEX mode) the mouse report in raw data (binary), the wheel is scroll up.
→ A1 03 00 00 00 00 00	← (shown in HEX mode) the mouse report in raw data (binary), the mouse is stopped, pressed button is released
→ A1 04 05 FE 03 02 FB 08 00 00	← (shown in HEX mode) the joystick/gamepad report in raw data (binary), the Throttle = 5, Left X = -2, Left Y = 3, Right X = 2, Right Y = -5, no movement on Hat Switch, no button is pressed.
→ A1 04 F6 00 00 00 00 05 06 40	← (shown in HEX mode) the joystick/gamepad report in raw data (binary), the Throttle = -10, Left X = 0, Left Y = 0, Right X = 0, Right Y = 0, Hat Switch is move towards left bottom, Button2 and Button3 is pressed, Button15 is pressed.
→ A1 04 00 00 00 00 00 08 00 00	← (shown in HEX mode) the joystick/gamepad report in raw data (binary), the Throttle = 0, no movement on Stick, no movement on Hat Switch, no Button is pressed.
……	
……	
	← Keep the UART port idle for 1 second.
→ BC:BP=00,00<CR><LF>	← change the bypass mode to **Message mode**.
	← Keep the UART port idle for 1 second.
← OK<CR><LF>	← Command accepted.

➔ BC:DI<CR><LF>	← disconnect the HID channel.
← IS=00<CR><LF>	← the HID channel of the Module is now disconnected and is connectable.

5.5. OPPC Device

This example shows how to use the Module as an OPPC device to push object to a remote OPPS device.

Ex. 5.5:

➔ BC:PF=00,00,00,00,01<CR><LF>	← configure the module profiles: OPPC is enabled, no SPP, HID profile is supported. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
← AP=00<CR><LF>	← Indicate that the Module has performed a reboot and is ready now.
➔ BC:FT=00,00,00,0A,01,0078<CR><LF>	← configure the module features: disable the auto connection after power on; disable the auto connect after paired; disable auto reconnect after link lost; set the interval of auto reconnect to 10s. configure the discover mode as 01: auto discoverable when empty. configure the timeout of discoverable as 120 seconds. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
➔ BC:IQ<CR><LF>	← inquire the Bluetooth device. This command is not necessary if the host knows the Bluetooth address of remote device.
← AP=01<CR><LF>	← indicate the Module is now inquiring.
← IR=03<CR><LF>	← indicate there are 3 Bluetooth devices found.
← AP=00<CR><LF>	← indicate Module is now in idle.
← FD=02,00189600000A,FFC6,OPPS_DEV<CR><LF>	← indicate the 3rd found device's address, RSSI and name.
← FD=01,00189600000B,FFC7<CR><LF>	← indicate the 2nd found device's address and the name is not gotten.
← FD=00,00189600000D,FFC8,BT_DEV_1<CR><LF>	← indicate the 1st found device's address and name.
➔ BC:CO=00189600000A<CR><LF>	← connect to the specified device 00:18:96:00:00:0A with the OPP profile.
← OS=01,00189600000A<CR><LF>	← the Module is now connecting to the specified OPPS device which address is 00:18:96:00:00:0A.
← CS=00,00189600000A<CR><LF>	← connecting result: success.
← OS=02,00189600000A<CR><LF>	← the Module is now connected to the specified OPPS device. The host can now start to push object to the remote OPPS device
➔42 43 3A 4F 41 3D 30 30 30 30 30 30 33 30 2C 31 30 2C 00 43 00 61 00 72 00 64 00 2E 00 76 00 63 00 66 0D 0A	← (shown in HEX mode) the object size is 48 Bytes(30 30 30 30 30 30 33 30=00000030h). the name length is 16 Bytes(31 30 = 10h). the file name is "**Card.vcf**" (00 43 = 'C', 00 61 = 'a', 00 72 = 'r', 00 64 = 'd', 00 2E = '.', 00 76 = 'v', 00 63 = 'c', 00 66 = 'f')
← OK<CR><LF>	← Command accepted.

➔ BC:OY=0C,text/x-vcard<CR><LF>	← the type length is 12(0Ch). the object type is "**text/x-vcard**".
← OK<CR><LF>	← Command accepted.
➔ 42 43 3A 4F 54 3D 30 31 2C 33 30 2C 42 45 47 49 4E 3A 56 43 41 52 44 0D 0A 56 45 52 53 49 4F 4E 3A 32 2E 31 0D 0A 4E 3A 4D 69 63 68 61 65 6C 0D 0A 45 4E 44 3A 56 43 41 52 44 0D 0A 0D 0A	← (Shown in HEX values) this is the final/only packet of this object (30 31 = 01h). the length of this packet is 48 Bytes (33 30 = 30h). the raw data of the packet is displayed in red (42 45 47…… 52 44 0D 0A).
← OK<CR><LF>	← Command accepted.
➔ BC:DO=00189600000A<CR><LF>	← disconnect with the specified device 00:18:96:00:00:0A with the OPP profile as we have already finished the object push.
← OS=00<CR><LF>	← the module is disconnected with the remote OPPS device.

5.6. OPPS Device

This example shows how to use the Module as an OPPS device to receive object pushed by an remote OPPC device.

Ex. 5.6:

➔ BC:PF=00,00,00,00,03<CR><LF>	← configure the module profiles: both OPPC and OPPS is enabled. no SPP, HID profile is supported. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
← AP=00<CR><LF>	← Indicate that the Module has performed a reboot and is ready now.
➔ BC:FT=00,00,00,0A,03,0000<CR><LF>	← configure the module features: disable the auto connection after power on; disable the auto connect after paired; disable auto reconnect after link lost; set the interval of auto reconnect to 10s. configure the discover mode as 03: auto discoverable when there is no connection, and the discoverable state will never timeout. This command is only needed when the first time use this Module.
← OK<CR><LF>	← Command accepted.
← OS=01,00189600000A<CR><LF>	← a remote OPPC device is initiating an OPP connection request, its address is 00:18:96:00:00:0A.
← CS=00,00189600000A<CR><LF>	← connecting result: success.
← OS=02,00189600000A<CR><LF>	← the Module is now connected with the remote OPPC device.
←4F 41 3D 30 30 30 30 30 30 33 30 2C 31 30 2C 00 43 00 61 00 72 00 64 00 2E 00 76 00 63 00 66 0D 0A	← (Shown in HEX values) the remote OPPC device is started to push an object. the object size is 48 Bytes (30 30 30 30 30 30 33 30=00000030h). the name length is 16 Bytes (31 30 = 10h). the file name is "**Card.vcf**" (00 43 = 'C', 00 61 = 'a', 00 72 = 'r', 00 64 = 'd', 00 2E = '.', 00 76 = 'v', 00 63 = 'c', 00 66 = 'f').

← OY=0C,text/x-vcard<CR><LF>	← the object type length is 12(0Ch). the object type is "**text/x-vcard**".
← 4F 54 3D 30 31 2C 33 30 2C 42 45 47 49 4E 3A 56 43 41 52 44 0D 0A 56 45 52 53 49 4F 4E 3A 32 2E 31 0D 0A 4E 3A 4D 69 63 68 61 65 6C 0D 0A 45 4E 44 3A 56 43 41 52 44 0D 0A 0D 0A	← (Shown in HEX mode) this is the final/only packet of this object (30 31 = 01h). the length of this packet is 48 Bytes (33 30 = 30h). the raw data of the packet is displayed in red (42 45 47…… 52 44 0D 0A).
← OS=00<CR><LF>	← the module is disconnected with the remote OPPC device.

5.7. BLE/iGate Peripheral of iOS/Android Device

This example shows how to use the Module as a Bluetooth low energy peripheral with iOS/Android or other BLE central devices to exchange data with apps running on those host devices.

Ex. 5.7:

→ BC:LB=01<CR><LF>	← Set the Module to bondable state, then open the iOS/Android app and start a new scanning and connection.
← OK<CR><LF>	← Command accepted.
← LS=01<CR><LF>	← Indicate that the Module get connection request from central device and is in connecting state.
← LS=02,2CF0A29782E6<CR><LF>	← Indicate that the Module has set up a successful pairing with the connected central device, the iGate profile has also been configured by the mobile app, the bi-directional data transmission channel has been set up.
← LD=05,hello<CR><LF>	← Indicate that the Module has received a ASCII string "hello" from the connected mobile app.
→ BC:LD=02,OK<CR><LF>	← Request the Module to send "OK" to the connected mobile app.
← LS=01<CR><LF>	← Indicate that the connection is lost and the module becomes connectable to the bonded device with address "2C:F0:A2:97:82:E6". This state won't timeout. The module also automatically enters this state after power up if it has been bonded with a BLE central device.
← LS=02,2CF0A29782E6<CR><LF>	← Indicate that the Module connects back to bonded device with address "2C:F0:A2:97:82:E6".

5.8. HFP AG

This example shows how to use the Module as a HFP AG to work with Bluetooth headsets.

Ex. 5.8:

→ BC:IQ<CR><LF>	← inquire the Bluetooth device. This command is not necessary if the host knows the Bluetooth address of remote device.
← AP=01<CR><LF>	← indicate the Module is now inquiring.

← FD=00,84AD8D0228F0,FFFFFFE1,00240418<CR><LF> ← indicate the found device's address, RSSI and COD.
← IR=01<CR><LF> ← indicate there are 3 Bluetooth devices found.
← AP=00<CR><LF> ← indicate Module is now in idle.
← FD=00,84AD8D0228F0,FFFFFFE1,00240418, Ranson's Powerbeats <CR><LF> ← indicate the found device's address, RSSI, COD and name.
← BC:CG=84AD8D0228F0<CR><LF> ← Connect to the HFP device and automatically open audio connection.
← GS=01,84AD8D0228F0 <CR><LF> ← Connecting signaling channel.
← GS=02,84AD8D0228F0 <CR><LF> ← Signaling channel connected.
← GS=03,84AD8D0228F0 <CR><LF> ← Connecting audio channel.
← GS=04,84AD8D0228F0 <CR><LF> ← Audio channel connected.
← GC=02<CR><LF> ← Audio channel connected, "Wide band speech" (mSBC) codec is used.

5.9. GATT Client with Auracast™ Transmitter

This example demonstrates how to use Flairmesh's FMA120 dongle to connect to and access the GATT server on two LE peripheral devices while simultaneously broadcasting with Auracast.

Ex. 5.9:

→ BC:ST<CR><LF> ←Query the state of the FMA120's audio.
← ST=06<CR><LF> ← Auracast broadcast is transmitting.
Please refer to the project FlooCast on GitHub for message "ST".
→ BC:IL<CR><LF> ←Start legacy scan with a default RSSI limit of -70dBm.
← OK<CR><LF> ← Command accepted.
← LS=01<CR><LF> ← The LE central state becomes "scanning".
← FD=01,7E51875C1872,C6,13FF0000,S23 Ultra<CR><LF> ← Found a device with the name "S23 Ultra".
→ BC:IQ=00<CR><LF> ← Stop the current scan.
← OK<CR><LF> ← Command accepted.
← LS=00<CR><LF> ← LE central state becomes "idle".
→ BC:GT=01,7E51875C1872<CR><LF> ← Connect to the found device with address "01,7E51875C1872".
← LS=02,01,7E51875C1872<CR><LF> ← LE central state becomes "connecting".
← LS=03,01,7E51875C1872<CR><LF> ← LE central state becomes "pairing".
← GT=01,01,7E51875C1872,00100000<CR><LF> ← Pairing with remote device, connection ID is "00100000".
← LM=00100000,007C<CR><LF> ← The MTU size of the connection "00100000" is 124 bytes.
← SD=00100000,1801,0001,0009<CR><LF> ← A service with UUID16 0x1801, handle range of 0001~0009.
← SD=00100000,1800,0014,001A<CR><LF> ← A service with UUID16 0x1800(GAS), handle range of 0001~0009.
← SD=00100000,1849,0028,004D<CR><LF> ← A service with UUID16 0x1849, handle range of 0028~004D.
← SD=00100000,184C,005A,007E<CR><LF> ← A service with UUID16 0x184C, handle range of 005A~007E.
← SD=00100000,1855,0082,FFFF<CR><LF> ← A service with UUID16 0x1855, handle range of 0082~FFFF.
← SD=00100000,0000<CR><LF> ← The end of the list of found services.
← PI=00,8C6A3BA53608<CR><LF> ← Pairing successfully with the remote device, the public address is 8C6A3BA53608.
← GT=02,01,7E51875C1872,00100000<CR><LF> ← Connected with remote device, connection ID is "00100000".
← LS=00<CR><LF> ← LE central state becomes "idle".
→ BC:CD=01000000,0014,001A<CR><LF> ← Register access to characteristics from handles 0014 to 001A,

which is the range of the generic access service indicated by SD. We can register all characteristics by specifying a range from 0001 to FFFF. We'll use the "0001,FFFF" for the second peripheral device in this example.

← CD=00100000,2A00,0015,0016,02<CR><LF>

← Found the device name characteristic (UUID16 2A00) value handle "0016", the property is "read".

← CD=00100000,2A01,0017,0018,02<CR><LF> ← Found the appearance characteristic value handle "0018".

← CD=00100000,2AA6,0019,001A,02<CR><LF>

← Found the central address resolution characteristic value handle "001A".

← CD=00100000,0000<CR><LF> ← The end of the list of found characteristics.

→ BC:CV=01000000,0016<CR><LF> ← Read the device name characteristic via handle "0016".

← CV=00100000,0016,53323320556C747261<CR><LF> ← The device name is "S23 Ultra".

→ BC:IL<CR><LF> ← Start legacy scan for the second device.

← OK<CR><LF> ← Command accepted.

← LS=01<CR><LF> ← The LE central state becomes "scanning".

← FD=01,5CB6CCB00F9F,E1,04000000,FMA100<CR><LF> ← Found a device with the name "FMA100".

→ BC:IQ=00<CR><LF> ← Stop the current scan.

← OK<CR><LF> ← Command accepted.

← LS=00<CR><LF> ← LE central state becomes "idle".

→ BC:GT=00,5CB6CCB00F9F<CR><LF> ← Connect to the found device with address "00,5CB6CCB00F9F".

← LM=00200000,007C<CR><LF> ← The MTU size of the connection "00200000" is 124 bytes.

← SD=00200000,1801,0001,0006<CR><LF> ← Generic Attribute service, handle range of 0001~0006.

← SD=00200000,1800,0007,000B<CR><LF> ← Generic Access service, handle range of 0007~000B.

← SD=00200000,F455A208597D11ECBF630242AC130002,000C,0016<CR><LF>

← A service with UUID128 0xF455A208597D11ECBF630242AC130002, handle range of 000C~0016.

← SD=00200000,00001100D10211E19B2300025B00A5A5,005A,007E<CR><LF>

← A service with UUID128 0x00001100D10211E19B2300025B00A5A5, handle range of 0017~001F.

← SD=00200000,180F,005A,007E<CR><LF> ← Battery service, handle range of 005A~007E.

← SD=00200000,180A,0082,FFFF<CR><LF> ← Device Information service, handle range of 0025~FFFF.

← SD=00200000,0000<CR><LF> ← The end of the list of found services.

← PI=00,5CB6CCB00F9F<CR><LF>

← Pairing successfully with the remote device, the public address is 5CB6CCB00F9F.

← GT=02,00,5CB6CCB00F9F,00200000<CR><LF> ← Connected with remote device, connection ID is "00200000".

← LS=00<CR><LF> ← LE central state becomes "idle".

← LP=00,5CB6CCB00F9F,0018,0000<CR><LF> ← The connection interval is 30ms, with a slave latency of 0.

→ BC:CD=02000000,0001,FFFF<CR><LF> ← Register access to characteristics from 0001 to FFFF.

← CD=00200000,2A05,0002,0003,20<CR><LF> ← Found the service change characteristic value handle "0003".

← CD=00200000,2B29,0005,0006,0A<CR><LF> ← Found the client supported features characteristic value handle "0006".

← CD=00200000,2A00,0008,0009,02<CR><LF> ← Found the device name characteristic value handle "0009".

← CD=00200000,2A01,000A,000B,02<CR><LF> ← Found the appearance characteristic value handle "000B".

← CD=00200000,81EB77BD89B844948A097F83D986DDC7,000D,000E,04<CR><LF>

← Found a characteristic declared by a UUID128, its value handle is "000E" and the property is "write without response".

← CD=00200000,81EB77BD89B844948A097F83D986DDC7,000F,0010,10<CR><LF>

← Found a characteristic declared by a UUID128, its value handle is "0010" and the property is "notify".

← CD=00200000,6C1CEF073377410EB23147F76C5A39E1,0012,0013,08<CR><LF>

← Found a characteristic declared by a UUID128, its value handle is "0013" and the property is "write with response".

← CD=00200000,6C1CEF073377410EB23147F76C5A39E1,0014,0015,10<CR><LF>

← Found a characteristic declared by a UUID128, its value handle is “0015” and the property is “notify”.

← CD=00200000,00001101D10211E19B2300025B00A5A5,0018,0019,08<CR><LF>

← Found a characteristic declared by a UUID128, its value handle is “0019” and the property is “write with response”.

← CD=00200000,00001102D10211E19B2300025B00A5A5,001A,001B,10<CR><LF>

← Found a characteristic declared by a UUID128, its value handle is “001B” and the property is “notify”.

← CD=00200000,00001103D10211E19B2300025B00A5A5,001D,001E,16<CR><LF> ← Found a characteristic declared by a UUID128, its value handle is “001E” and the property is “read, notify and write without response”.

← CD=00200000,2A19,0021,0022,12<CR><LF> ← Found the battery level characteristic value handle “0022”.

← CD=00200000,2A29,0026,0027,02<CR><LF>

← Found the Manufacturer Name String characteristic value handle “0027”.

← CD=00200000,2A24,0028,0029,02<CR><LF> ← Found the Model Number String characteristic value handle “0029”.

← CD=00200000,2A25,002A,002B,02<CR><LF> ← Found the Serial Number Stringcharacteristic value handle “002B”.

← CD=00200000,2A27,002C,002D,02<CR><LF>

← Found the Hardware Revision String characteristic value handle “002D”.

← CD=00200000,2A26,002E,002F,02<CR><LF>← Found the Firmware Revision String characteristic value handle “002F”.

← CD=00200000,2A28,0030,0031,02<CR><LF> ← Found the Software Revision String characteristic value handle “0031”.

← CD=00200000,0000<CR><LF> ← The end of the list of found characteristics.

→ BC:DD=02000000,0011,0011<CR><LF> ← Discover the characteristic descriptor in range of 0011~0011.

← DD=00200000,2902,0011<CR><LF> ← Found a descriptor “Client Characteristic Configuration” on 0011.

← DD=00200000,0000<CR><LF> ← The end of the list of found descriptors.

→ BC:DD=02000000,0016,0016<CR><LF> ← Discover the characteristic descriptor in range of 0016~0016.

← DD=00200000,2902,0016<CR><LF> ← Found a descriptor “Client Characteristic Configuration” on 0016.

← DD=00200000,0000<CR><LF> ← The end of the list of found descriptors.

→ BC:CW=02000000,0011,0100<CR><LF> ← Enable notification on handle 0010 by writing 0x0001 to handle 0011.

← WR=00200000,0011,0000<CR><LF> ← Write to handle 0011 successful.

→ BC:CW=02000000,0016,0100<CR><LF> ← Enable notification on handle 0015 by writing 0x0001 to handle 0016.

← WR=00200000,0016,0000<CR><LF> ← Write to handle 0016 successful.

← CV=00200000,0010,303030<CR><LF> ← The remote GATT server on connection "00200000" sent a notification with the value of the ASCII string "000" on handle 0010.

→ BC:CV=02000000,000E,3031<CR><LF> ← Write a value of the ASCII string “01” on handle 000E by write without response.

5.10. Build a Broadcast Network

This example shows how to use Flairmesh's modules to build a broadcast network.

Ex. 5.10:

For a sender:

➔ BC:BC=01<CR><LF> ← Set the role as sender.

← OK<CR><LF> ← Command accepted.

➔ BC:BK=31F6,803B,E9E2,1F84,A9B7,1F40,0A2C,4DC1<CR><LF> ← set the 128bits encryption key.

← OK<CR><LF> ← Command accepted.

For a receiver:

➔ BC:BC=00<CR><LF> ← Set the role as sender.

← OK<CR><LF> ← Command accepted.

➔ BC:BK=31F6,803B,E9E2,1F84,A9B7,1F40,0A2C,4DC1<CR><LF> ← set the 128bits encryption key.

← OK<CR><LF> ← Command accepted.

A broadcast network may have multiple senders and receivers. And each device can switch its role in runtime.

On a sender:

➔ BC:BD=0F,<00 01>I am a sender<CR><LF> ← Send a text message "I am a sender" which is 13bytes, and prefix it with an index of raw hex value 0x0001.

← OK<CR><LF> ← Command accepted

On all receivers in the range:

← BD=10,<00 01>I am a sender<00><CR><LF> ← A text message "I am a sender" is received, it has an index of 0x0001 and one 0 padding. The total length is always 16 bytes.