NovaComm Control Interface (NCCI) For Bluetooth Low Energy User Guide

Ver.1.0

July 11, 2013

Release Record

Version	Release Date	Comments
0.1	Mar 03, 2012	Draft
0.2	May 19, 2011	1. Add **NM** command.
0.3	Jun 8, 2012	1. Add **PD** command.
0.4	July 8, 2012	1. Correct some spell error. 2. Update details of **RT** status.
0.5	July 18, 2012	1. Update details of **PD** command. 2. Add **RI** command.
0.6	Aug 10, 2012	1. Update details of RT status. A new state code 05 is added to indicate the communication is secure.
0.7	Sept 10, 2012	1. The **AD**, **AP**, **CC**, **NM**, **BR** are changed to “available” with Ver.0.7. 2. **FD** is added. 3. Description of **NM** is changed. The length of device name is limited to 19. 4. **RT** can be used to query/poll the status.
0.8	July 4, 2013	1. Remove the unsupported commands and indicators. 2. Fix errors in the parameter description for several commands and indicators.
1.0	July 11, 2013	1. Add the **UD** command for set the UUID128 of the GATT service. 2. Change BTLE to BLE. 3. Fix several errors in the parameter descriptions. 4. Add descriptions for the paired device list.

Table of Contents

Table of Tables

Table of Figures

1. Introduction

The NovaComm Control Interface (NCCI) is a set of ASCII commands and indicators with which the user can control the NovaComm's Bluetooth Module via UART interface by a host (PC, MCU, etc.).

The commands are used to control the Bluetooth Module sent by host. The indicators (or the response of the commands) are output from the Bluetooth Module to the host to indicate the status of the module.

NCCI for Bluetooth Low Energy (BLE) can be considered as a subset and/or extension of the common NCCI.

This document only described the interface of the BLE peripheral role. There's another document for the central role. Different firmware is used for different role.

1.1. Default UART Configuration

The default configuration of UART is given below:

Baud rate: 2400

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

The configuration of UART can be changed by NCCI and the new configuration is stored and used for further communication.

2. Command and Indicator Syntax

2.1. General Syntax

The general syntax of NCCI command is shown as below:

BC:<CMD>[=Para1][,Para2][,RawData][,…]<CR><LF>

The general syntax of NCCI indicator is shown as below:

<IDC>[=Para1][,Para2][,RawData][,…]<CR><LF>

Description of each field:

BC: is the command line prefix.

CMD is the basic command. All of the commands are listed in section 3.

IDC is the basic indicator. All of the indicators are listed in section 4.

= is the separator between command/indicator and parameter. It's only needed if a parameter is presented.

Para1 is the first parameter. Not all of the commands have a parameter.

, is the separator between parameters. It's only needed if subsequent parameter is presented.

Para2 is the second parameter if available.

RawData is the raw data which will be sent by the command. Only parts of the commands have this field.

<CR><LF> is the terminator of the command line.

Notes:

1. *If a parameter is mandatory, it will be surrounded by { }. If a parameter is optional, it will be surrounded by [ ].*
2. *<CR> means Carriage Return, and <LF> means Line-Feed.*
3. *All of the parameters are composed of ASCII characters while the **RawData** field can composed of any data contents.*
4. *In indicators, the module prints hex values in low case. For other places in the document we always use upper case characters for hex values.*

2.2. Examples

Here is some examples show how to use the NCCI commands and indicators.

Ex. 2.1

➔ BC:NM=A Keyb<CR><LF> ← *configure the GAP device name characteristic. The new name is "A Keyb".*

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

Notes:

1. *For the examples in this document, the command sent to the Bluetooth Module will be shown with "➔" at the beginning of the line, while the indicator output by Bluetooth Module will be shown with "←" at the beginning of the line.*
2. *For the examples in this document, the comments will start with a "←" and be written in italic.*
3. *For the examples in this document, only the characters in grey background color are the real content of a command or indicator.*

For a full example using the module with iOS devices, please check 7.1.

3. Command List

All the available NCCI commands are listed and briefly described in the tables below. The detailed description of each command is given in chapter 5.

Table 3.1 NCCI for BLE Command List

Command	Short Description
UD	Query or configure the GATT service **U**UI**D**128 the module
AD	Query the Bluetooth **ad**dress of the module
AP	Query or configure the GAP **Ap**pearance characteristic
FT	Query or configure the **f**ea**t**ures of the Bluetooth Module.
CC	Query or configure the peripheral preferred **c**onnection parameter **c**haracteristic
NM	Query or change the GAP device **n**a**m**e characteristic.
BR	Query or change the UART **b**aud **r**ate.
FD	Configure all setting to **f**actory **d**efault values
MD	**M**ake the Bluetooth Module **d**iscoverable/connectable, in BLE the peripheral start/stop the advertising after got this command.
DC	**D**is**c**onnect with remote BLE central.
DT	Send **d**a**t**a packet to the connected BLE central.
RT	Query the current **r**adio s**t**atus.

RI	Query the RSSI for the current connection.
PD	**P**ower **d**own the module, put it into dormant or hibernate state, or warm reset it.
CP	**C**lear the **p**aired Bluetooth device list.

4. Indicator List

All the available NCCI indicators are listed and briefly described in the tables below. The detailed description of each indicator is given in chapter 6.

Table 4.1 NCCI for BLE Indicator List

Indicator	Short Description
OK	Indicates a command was adopted by the Bluetooth Module.
ER	Indicates there is an **er**ror detected in the command sent by the host.
UD	Report the GATT service **U**UI**D**128 the module
AD	Report the Bluetooth **ad**dress of the Module
AP	Report the GAP **Ap**pearance characteristic
FT	Reports the configured **f**ea**t**ures of the Bluetooth Module.
CC	Report the peripheral preferred **c**onnection parameter
NM	Reports the device **n**a**m**e of the Bluetooth Module.
BR	Reports the UART **b**aud **r**ate.
DT	Received **d**a**t**a packet from the connected BLE central.
RT	Report the **r**adio **s**tate.
RI	Report the RSSI for the current connection.

5. Description of ASCII Commands

5.1. General Information Commands

5.1.1. UD—GATT Service UUID128

5.1.1.1. Description:

The module works as a BLE peripheral and it implements a proprietary service which can be taken as a virtual SPP channel between the peripheral and connected central. This command can be used to set a unique UUID128 for the service. Once configured, the UUID128 will take effect when the module reset or power up for next time. The Bluetooth module stores the UUID128 in its non-volatile memory so the value won't change until be set again.

5.1.1.2. Syntax:

Synopsis:
BC:UD[=Value Index,Value]<CR><LF>

5.1.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Value Index	The UUID128 has 128bits, which is represented by 8 16-bits words. The 8 words are stored in a lowest word first order. Index 0 is the lowest word of the UUID128. Value: 0000-0007	M	
Value	The value of the word. Should be 4 hex char. The word value should be in a lowest byte first order. Refer to the following example.	O	If value is not given, the current value of the index is output as the response of this command.

5.1.1.4. Examples:

Ex. 5.1. To query the UUID128 of Bluetooth Module:

➔ BC:UD=0000<CR><LF> ← *query the value of word 0 in the UUID128 (lowest word).*

← UD=0000,7166<CR><LF> ← *report the value of the word 0 in the UUID128, it's 0x6671 (in lowest byte first order).*

Ex. 5.2. To set the UUID128 of the Bluetooth Module to "C5161D82-AAB0-4E55-8D96-C59D816E6671" (an ASCII string represents the UUID128):

➔ BC:UD=0000,7166<CR><LF> ← *set the word 0 in the UUID128 to 0x6671.*

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

➔ BC:UD=0001,6e81<CR><LF> ← *set the word 1 in the UUID128.*

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

➔ BC:UD=0002,9dc5<CR><LF> ← *set the word 2 in the UUID128.*

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

➔ BC:UD=0003,968d<CR><LF> ← *set the word 3 in the UUID128.*

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

➔ BC:UD=0004,554e<CR><LF> ← *set the word 4 in the UUID128.*

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

➔ BC:UD=0005,b0aa<CR><LF> ← *set the word 5 in the UUID128.*

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

➔ BC:UD=0006,821d<CR><LF> ← *set the word 6 in the UUID128.*

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

➔ BC:UD=0007,16c5<CR><LF> ← *set the word 7 in the UUID128.*

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

5.1.2. AD—Bluetooth Address

5.1.2.1. Description:

This command can query the Bluetooth address of local module. The response also contains the remote device address if it's bonded. Once the Bluetooth Module adopted this query request, it will report its Bluetooth address by the Indicator AD.

5.1.2.2. Syntax:

Synopsis:
BC:AD<CR><LF>

5.1.2.3. Parameter Description:

None.

5.1.2.4. Examples:

Ex. 5.3. To query the Bluetooth address of local module:

➔ BC:AD<CR><LF> ← *query the Bluetooth address of local module.*

← AD=00,00189600abcd<CR><LF> ← *the Bluetooth Module is using a public Bluetooth address, 00:18:96:00:AB:CD.*

5.1.3. AP—GAP Appearance Characteristic

5.1.3.1. Description:

This command can query or configure the GAP device appearance character value of Bluetooth module. Once configured, the configuration will take effect immediately and until the next time the module is configured by this command or the module is rebooted. It means the Bluetooth Module will not remember the configuration, and after the Bluetooth module has been powered off, the configuration will be lost.

If the parameter is not presented, the Bluetooth Module will report current appearance by the Indicator AP.

The appearance is also used in the advertising. When updated, the value will be used in next time the module start advertising. Standard definitions of appearance could be found in Table 7.1.

5.1.3.2. Syntax:

Synopsis:
BC:AP[=Appearance Value]<CR><LF>

5.1.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Appearance Value	The GAP appearance characteristic of the Bluetooth Module. Value: 0000h—FFFFh as defined by Bluetooth Assigned Number, refer to Table 7.1 Default: per firmware version	O	

5.1.3.4. Examples:

Ex. 5.4. To query current appearance of Bluetooth Module:

➔ BC:AP<CR><LF> ← *query current appearance value.*

← AP=03c1<CR><LF> ← *report current appearance value: 03C1 as a keyboard.*

Ex. 5.5. To configure the appearance of Bluetooth Module:

➔ BC:AP=0301<CR><LF> ← *configure the module appearance value: 0301.*

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

5.1.4. FT—Features

5.1.4.1. Description:

This command can query or configure the features of Bluetooth Module. Once configured, the configuration will take effect immediately. The Bluetooth module stores the value in its non-volatile memory so the value won't change until be set again. If the parameter is not presented, the Bluetooth Module will report current feature configuration by the Indicator FT. If the user wants to configure the features, all of the parameters should be given together.

5.1.4.2. Syntax:

Synopsis:
FT[=Features]<CR><LF>

5.1.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Features	A 16bits value as a bit mask, for example, 8000 means the bit 15 is 1 and all other bits is 0. The features are defined with the 16 bits mask, 1 is enable. Bit 15: automatically start advertising when power up. Bit 14: reserved, always set to 1. Bit 13: reserved, always set to 1. Bit 12: use privacy, which means the module only connects to bonded devices (need to clear the paring information before connect a new device) Bit 11-0: Not defined yet.	O	

Notes:

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

5.1.4.4. Examples:

Ex. 5.6. To query current feature configuration of Bluetooth Module:

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

← FT=e000<CR><LF> ← *report current feature configuration.*

5.1.5. CC—Preferred Connection Parameters Characteristic

5.1.5.1. Description:

This command can query or configure the GAP preferred connection parameter characteristic of Bluetooth Module. Once configured, the configuration will take effect at the next time when a Bluetooth connection is being established. The Bluetooth module stores the value in its non-volatile memory so the value won't change until be set again.

If the parameter is not presented, the Bluetooth Module will report current configuration by the Indicator CC.

5.1.5.2. Syntax:

Synopsis:
BC:CC[=Min Interval,Max Interval,Latency,Supervision Timeout]<CR><LF>

5.1.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Min Interval	The minimum value for the connection interval. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default depends on firmware	O	
Max Interval	The maximum value for the connection interval. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default depends on firmware	O	
Latency	The slave latency for the connection in number of connection events. Hex Value: xxxx (0000 to 03E8) Default depends on firmware	O	
Supervision Timeout	The connection supervisor timeout multiplier as a multiple of 10ms. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default depends on firmware	O	

5.1.5.4. Examples:

Ex. 5.7. To query current preferred connection parameters of the Bluetooth Module:

➔ BC:CC<CR><LF> ← *query the current preferred connection parameters.*

← CC=0010,0050,0004,0258<CR><LF> ← *report the current preferred connection parameters.*

Ex. 5.8. To set the preferred connection parameters of the Bluetooth Module:

➔ BC:CC=0010,0050,0004,0258<CR><LF> ← *Set new connection parameters.*

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

5.1.6. NM—GAP Device Name Character Value

5.1.6.1. Description:

This command can query or change the GAP device name character value of the Bluetooth Module. It takes effect immediately. The Bluetooth Module stores the value in its non-volatile memory so the value won't change until be set again. If the parameter is not presented, the Bluetooth module will report current friendly name by the Indicator NM.

5.1.6.2. Syntax:

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

5.1.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Name	The new device name of the Bluetooth Module. Length: 1—19 characters Default: Per software version.	O	

5.1.6.4. Examples:

Ex. 5.9. To query current device name of Bluetooth Module:

➔ BC:NM<CR><LF> ← *query current GAP device name character value.*

← NM=NVC_iGATE<CR><LF> ← *report current GAP device name character value, it's "NVC_iGATE".*

Ex. 5.10. To change the device name of Bluetooth Module:

➔ BC:NM=NVC_BLE<CR><LF> ← *change the GAP device name character value to "NVC_BLE"*

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

5.1.7. BR—UART Baud Rate

5.1.7.1. Description:

This command can query or change the UART baud rate of Bluetooth Module. Once changed, the new baud rate will take effect immediately. The Bluetooth module stores the value in its non-volatile memory so the value won't change until be set again.

If the parameter is not presented, the Bluetooth Module will report current baud rate by the Indicator BR.

5.1.7.2. Syntax:

Synopsis:
BC:BR[=Baud Rate]<CR><LF>

5.1.7.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Baud Rate	The new baud rate of the Bluetooth Module. Value: 00h—0Ch 00: 2400 01: 9600 02: 19200 03: 38400 04: 57600 05: 115200 06: 230400 07: 460800 08: 921600 09: 1382400 0A: 1843200	O	The default baud rate may not be 2400 per software version. **There's an advantage using default 2400 baud rate – no need to assert (pull up to high logic level) the WAKE pin to wake up the module from deep sleep before transmitting to it.**

	0B: 2764800 0C: 3686400 Default: 00 (2400)		

Warning:

1. *Please do NOT try to change to a new baud rate if you don't have a host which can work in that baud rate, for there is no other way to reset it except for UART port.*
2. *If baud rate other than 2400 is used, please keep pulling up the WAKE pin during sending data to the module.*

5.1.7.4. Examples:

Ex. 5.11. To query the baud rate of Bluetooth Module:

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

← BR=00<CR><LF> ← *report the baud rate, it's 2400.*

Ex. 5.12. To change the baud rate of Bluetooth Module:

➔ BC:BR=01<CR><LF> ← *change the baud rate to 9600.*

← ← *There's no response for the set command. The new baud rate is effective immediately after receive the command. And the following commands and indicators are in new baud rate.*

5.1.8. FD—Configure All Setting to Factory Default Values

5.1.8.1. Description:

This command can be used to configure all settings to the factory default values.

5.1.8.2. Syntax:

Synopsis:
BC:FD<CR><LF>

5.1.8.3. Parameter Description:

None.

5.1.8.4. Examples:

Ex. 5.13. To reset all settings of the Bluetooth Module:

➔ BC:FD<CR><LF> ← *Configure all settings to factory default values.*

← ← *There's no response of the command because the UART baud rate is also reset to default.*

5.2. Connection and Data Transfer Commands

5.2.1. MD—Discoverable

5.2.1.1. Description:

This command can start or stop the BLE advertising. The module is discoverable/connectable only when the module is advertising.

5.2.1.2. Syntax:

Synopsis:
BC:MD{=Flag}<CR><LF>

5.2.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Flag	Switch on/off advertising. Value: 00h or 01h 00: Stop advertising. 01: Start advertising.	M	

5.2.1.4. Examples:

Ex. 5.14. To make Bluetooth Module discoverable:

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

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

5.2.2. DC—Disconnect from BLE Central

5.2.2.1. Description:

This command disconnects the Bluetooth Module from the connected Bluetooth BLE central.

5.2.2.2. Syntax:

Synopsis:
BC:DC<CR><LF>

5.2.2.3. Parameter Description:

None.

5.2.2.4. Examples:

Ex. 5.15. To disconnect with the connected devices:

➔ BC:DC<CR><LF> ← *disconnect with current connected BLE central.*

← RT=00<CR><LF> ← *the Bluetooth Module is now disconnected.*

5.2.3. DT—Send Data Packet to BLE Central

5.2.3.1. Description:

This command is used to send a data packet to connected BLE central.

5.2.3.2. Syntax:

Synopsis:
BC:DT{=Data String}<CR><LF>

5.2.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Data String	The data string to be send. **(keep the length of string not more than 20, not including the** <CR><LF>)	M	

Notes:

1. *The <CR><LF> sequence is treated as the terminator of a command line. If the user want to send these special characters to remote device, precede them with a backslash character "\". (So send "\\" for "\").*

5.2.3.4. Examples:

Ex. 5.16. To send data:

➔ BC:DT=1234567890<CR><LF> ← *send a data packet.*

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

5.2.4. RT—Radio Status

5.2.4.1. Description:

This command is used to query the status of the local BLE radio.

5.2.4.2. Syntax:

Synopsis:
BC:RT<CR><LF>

5.2.4.3. Parameter Description:

None.

5.2.4.4. Examples:

Ex. 5.17. To query the state of BLE radio:

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

← RT=05,01,00189600abcd<CR><LF> ← *the Bluetooth Module is now bonded to the BLE central device whose random address is 00:18:96:00:AB:CD.*

5.2.5. RI—RSSI

5.2.5.1. Description:

This command is used to query the RSSI (radio signal strength indication) of the current connection.

5.2.5.2. Syntax:

Synopsis:
BC:RI<CR><LF>

5.2.5.3. Parameter Description:

None.

5.2.5.4. Examples:

Ex. 5.18. To query the RSSI of the current connection:

➔ BC:RI<CR><LF> ← *query the RSSI of the current connection.*

← RI=bb<CR><LF> ← *the RSSI of the current connection is -69dBm.*

5.3. Power Mode Commands

5.3.1. PD—Power Down the Module

5.3.1.1. Description:

This command can power down the module into dormant or hibernate state. In dormant state the module can only be wake up by the pull high the WAKE pin. In hibernate state it will automatically wake up after a given timeout. This command can also be used to warm reset the module.

There's no indicator or response from the UART when the module is power down.

5.3.1.2. Syntax:

Synopsis:
BC:PD[=Mode,Timeout Value]<CR><LF>

5.3.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Mode	The power down mode. Value: 00: Dormant, the module power down immediately. 01: Hibernate, the module wake up after the timeout 02: warm reset	M	
Timeout	7 hex value, the timeout unit is 2^20 microseconds, about	Should not	

Value	1second. So the valid range is 1s to 8.9 years.	use with Dormant or Warm reset. M for hibernate	

Notes:

1. *When in dormant, the module can only wake up by pull WAKE pin to logical 1 (high logical voltage).*

5.3.1.4. Examples:

Ex. 5.19. To put the module into hibernate and make it automatically wake up after 16 seconds:

➔ BC:PD=01,0000020<CR><LF> ← *Put the module into hibernate, and ask it automatically wake up after 32 seconds.*

Ex. 5.20. To put the module into dormant state:

➔ BC:PD=00<CR><LF> ← *Put the module into dormant, can only wake up it by WAKE pin.*

Ex. 5.21. To warm reset the module:

➔ BC:PD=02<CR><LF> ← *Reset the Bluetooth Module.*

5.4. Security Commands

5.4.1. CP—Clear the Paired Bluetooth Device Information

5.4.1.1. Description:

This command can clear the paired device information stored in the Bluetooth Module. All paired devices are stored in the module as a list. This command will clear all the information in the list.

5.4.1.2. Syntax:

Synopsis:
BC:CP<CR><LF>

5.4.1.3. Parameter Description:

None.

5.4.1.4. Examples:

Ex. 5.22. To clear the paired device list:

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

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

6. Description of ASCII Indicators

6.1. General Indicators

6.1.1. OK—Command is Adopted by the Bluetooth Module

6.1.1.1. Description:

This indicator indicates a command was adopted by the Bluetooth Module successfully.

6.1.1.2. Syntax:

Synopsis:
OK<CR><LF>

6.1.1.3. Parameter Description:

None.

6.1.1.4. Examples:

Ex. 6.1. To make Bluetooth Module discoverable:

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

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

6.1.2. ER—Error

6.1.2.1. Description:

This indicator indicates there is an error detected in the command sent by the host.

6.1.2.2. Syntax:

Synopsis:
ER{=Error Code}<CR><LF>

6.1.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Error Code	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	

6.1.2.4. Examples:

Ex. 6.2.

➔ RT=01 ← *The module is advertising now.*

➔ BC:MD=01<CR><LF> ← *Send command to make Bluetooth Module discoverable*

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

Ex. 6.3.

➔ BC:XX<CR><LF> ← *Send a command "XX" which is not in the supported command list.*

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

6.1.3. UD—GATT Service UUID128

6.1.3.1. Description:

This indicator reports the GATT service UUID128 of the Module.

6.1.3.2. Syntax:

Synopsis:
UD{=Value Index,Value}<CR><LF>

6.1.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Value Index	The UUID128 has 128bits, which is represented by 8 16-bits words. The 8 words are stored in a lowest word first order. Index 0 is the lowest word of the UUID128. Value: 0000-0007	M	
Value	The value of the word. The word value is print in a lowest byte first order. Refer to the example in 5.1.1.4.	M	

6.1.3.4. Examples:

Refer to the examples in section 5.1.1.4.

6.1.4. AD—Bluetooth Address of the Module

6.1.4.1. Description:

This indicator reports the Bluetooth address of the Module.

6.1.4.2. Syntax:

Synopsis:
AD{=Bluetooth Address Type,Bluetooth Address}<CR><LF>

6.1.4.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Bluetooth Address Type	The type of the connected device Bluetooth address. Value: 00h-01h 00: Public device address 01: Random device address	O	Only available when there's a connection existed (state = 04).
Bluetooth Address	The Bluetooth address of the connected BLE central.	O	Only available when there's a connection existed (state = 04).

6.1.4.4. Examples:

Refer to the examples in section 5.1.2.4.

6.1.5. AP—GAP Appearance Characteristic

6.1.5.1. Description:

This indicator indicates the GAP appearance characteristic.

6.1.5.2. Syntax:

Synopsis:
AP{=Appearance Value}<CR><LF>

6.1.5.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Appearance Value	The GAP appearance characteristic of the Bluetooth Module. Value: 0000h—FFFFh as defined by Bluetooth Assigned Number, refer to Table 7.1 Default: per firmware version	M	

6.1.5.4. Examples:

Refer to 5.1.3.4

6.1.6. FT—Feature Configuration

6.1.6.1. Description:

This indicator will report current feature configuration of the Bluetooth Module.

6.1.6.2. Syntax:

Synopsis:
FT{=Features}<CR><LF>

6.1.6.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Features	Refer to 5.1.4.3.	M	

Notes:

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

6.1.6.4. Examples:

Refer to the examples in section 5.1.4.4.

6.1.7. CC—Preferred Connection Parameter Characteristic

6.1.7.1. Description:

This indicator report the GAP preferred connection parameter characteristic of Bluetooth Module.

6.1.7.2. Syntax:

Synopsis:
CC{=Min Interval,Max Interval,Latency,Supervision Timeout}<CR><LF>

6.1.7.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Min Interval	The minimum value for the connection interval. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default depends on firmware	O	
Max Interval	The maximum value for the connection interval. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default depends on firmware	O	
Latency	The slave latency for the connection in number of connection events. Hex Value: xxxx (0000 to 03E8) Default depends on firmware	O	
Supervision Timeout	The connection supervisor timeout multiplier as a multiple of 10ms. Hex Value: xxxx (0006 to 0C80, FFFF means no specific define) Default depends on firmware	O	

6.1.7.4. Examples:

Refer to the examples in section 5.1.5.4.

6.1.8. NM—GAP Device Name Characteristic

6.1.8.1. Description:

This indicator will report current GAP device name characteristic of the Bluetooth Module.

6.1.8.2. Syntax:

Synopsis:
NM{=Name}<CR><LF>

6.1.8.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Name	Refer to 5.1.6.3.	M	

6.1.8.4. Examples:

Refer to the examples in section 5.1.6.4.

6.1.9. BR—UART Baud Rate

6.1.9.1. Description:

This indicator will report current UART baud of the Bluetooth Module.

6.1.9.2. Syntax:

Synopsis:
BR{=Baud Rate}<CR><LF>

6.1.9.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Baud Rate	Refer to 5.1.7.3.	M	

6.1.9.4. Examples:

Refer to the examples in section 5.1.7.4.

6.2. Connection and Data Packet Indicators

6.2.1. RT—Radio State

6.2.1.1. Description:

This indicator reports the radio state.

6.2.1.2. Syntax:

Synopsis:
RT{=Radio State},[Remote Bluetooth Address Type],[remote Bluetooth Address]<CR><LF>

6.2.1.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Radio State	The state of radio connection. State Value: 00h-05h 00: idle and no connection exists. 01: the module is in fast advertising. 02: the module is in slow advertising. (connecting speed is slower, but power consumption is much lower than fast advertising) 03: the module is connected with a BLE central. 04: the module is trying to disconnect, this is a temporary state. 05: a security channel is made with the connected BLE central	M	
Remote Bluetooth Address Type	The type of the connected BLE central address. Value: 00h-01h 00: Public device address 01: Random device address	O	Only available when there's a secure connection existed (state = 05).
Remote Bluetooth Address	The Bluetooth address of the connected BLE central.	O	Only available when there's a secure connection existed (state = 05).

6.2.1.4. Examples:

Refer to the examples in section 5.2.4.4.

6.2.2. DT—Data packet received from remote device

6.2.2.1. Description:

This indicator indicates there is a data packet received from a remote device.

6.2.2.2. Syntax:

Synopsis:
DT{=Data String}<CR><LF>

6.2.2.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
Data String	The received data. The raw data is output as what they are encoded in the remote sender. So the parser should handle the case that <CR> and <LF> is mixed in the data string.	M	

6.2.2.4. Examples:

Ex. 6.4. A data packet is received:

← DT=1234567890<CR><LF> ← *a data string received from remote device.*

6.2.3. RI—RSSI

6.2.3.1. Description:

This indicator reports the RSSI of the current connection.

6.2.3.2. Syntax:

Synopsis:
RI{=RSSI}<CR><LF>

6.2.3.3. Parameter Description:

Parameter	Description	Mandatory or Optional	Comments
RSSI	The RSSI of the current connection. The value is in dBm and the accuracy is 6dB. The value is in two's complement signed format. So the 2 hex characters have a range of +127 to -128.	M	

6.2.3.4. Examples:

Refer to the examples in section 5.2.5.4.

6.3. Power Mode Indicators

The module goes directly to the power mode which "PD" command request, it does not send any indicator when power mode changes.

6.4. Security Indicators

Currently no security indicator is supported. The module uses “Just work” simple pairing to bond with a BLE central.

7. Appendix A

7.1. Work with the iOS Devices

To simplify the integration of the module into an iOS accessory, NovaComm also supplies library on iOS to help customer’s apps communicating with the NovaComm BLE modules. NovaComm has another user manual for the iOS library.

Two examples of the common commands and indicators log when work with iOS devices at the module side are listed below. **The iOS app should use the same UUID128 as described in 5.1.1 to make the connection.**

Ex. 7.1, automatically advertising after power up:

➔ iGate V1.2<CR><LF> ← *The module powers up and print its version number 1.2.*

➔ RT=01<CR><LF> ← *The module start advertising automatically after power up.*

← RT=03<CR><LF> ← *A BLE central is connecting with the module.*

← RT=05,01,64200c2c462a<CR><LF> ← *Indicate that the module has been connected with an iOS device whose random address is 64:20:0C:2C:46:2A.*

➔ BC:DT=Hello iOS central<CR><LF> ← *Send “Hello iOS central” to connected iOS app.*

← DT=Hello NovaComm BLE<CR><LF> ← *Got a message “Hello NovaComm BLE” from iOS app.*

……

Ex. 7.2, start advertising by command:

➔ BC:MD=01<CR><LF> ← *When the module is in “RT=00” status.*

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

← RT=03<CR><LF> ← *A BLE central is connecting with the module.*

← RT=05,01,64200c2c462a<CR><LF> ← *Indicate that the module has been connected with an iOS device whose random address is 64:20:0C:2C:46:2A.*

……

7.2. Code of GAP Appearance

Table 7.1 Code of GAP Appearance

Value(DEC)	Device Type	Category
0	Unknown	None
64	Generic Phone	Generic category
128	Generic Computer	Generic category
192	Generic Watch	Generic category
193	Watch: Sports Watch	Watch subtype
256	Generic Clock	Generic category

320	Generic Display	Generic category
384	Generic Remote Control	Generic category
448	Generic Eye-glasses	Generic category
512	Generic Tag	Generic category
576	Generic Keyring	Generic category
640	Generic Media Player	Generic category
704	Generic Barcode Scanner	Generic category
768	Generic Thermometer	Generic category
769	Thermometer: Ear	Thermometer subtype
832	Generic Heart rate Sensor	Generic category
833	Heart Rate Sensor: Heart Rate Belt	Heart Rate Sensor subtype
896	Generic Blood Pressure	Generic category
897	Blood Pressure: Arm	Blood Pressure subtype
898	Blood Pressure: Wrist	Blood Pressure subtype
960	Human Interface Device (HID)	HID Generic
961	Keyboard	HID subtype
962	Mouse	HID subtype
963	Joystick	HID subtype
964	Gamepad	HID subtype
965	Digitizer Tablet	HID subtype
966	Card Reader	HID subtype
967	Digital Pen	HID subtype
968	Barcode Scanner	HID subtype
1024	Generic Glucose Meter	Generic category
1088	Generic: Running Walking Sensor	Generic category
1089	Running Walking Sensor: In-Shoe	Running Walking Sensor subtype
1090	Running Walking Sensor: On-Shoe	Running Walking Sensor subtype
1091	Running Walking Sensor: On-Hip	Running Walking Sensor subtype
1152	Generic: Cycling	Generic category
1153	Cycling: Cycling Computer	Cycling subtype
1154	Cycling: Speed Sensor	Cycling subtype
1155	Cycling: Cadence Sensor	Cycling subtype
1156	Cycling: Power Sensor	Cycling subtype
1157	Cycling: Speed and Cadence Sensor	Cycling subtype

7.3. Control Packet Format for iLight (PWM) Extension

The NVC-MDCS71 Bluetooth module has four PWM output IOs (PIO3, PIO4, PIO9, PIO10). With proper firmware, special control packet can be send from the connected iOS device to the Bluetooth Module to control the four PWM ports.

Length of the packet (10, not include itself)	Command ID (0x00 for LED control)	LED index (0x00 ~ 0x03)	LED switch (00: off 01: on 02: dim)	LED dull on	LED dull off	LED dull hold	LED bright on	LED bright off	LED bright hold	LED ramp rate

11 bytes total

LED index: 00~03 map to pio 3,4,9,10 of the module
LED switch:
00: always off
01: full on
02: dim pattern defined as following,
dull_on_time the amount of time, in units of ~30us, for which the LED should be on during the dullest part of the flash sequence.
dull_off_time the amount of time, in units of ~30us, for which the LED should be off during the dullest part of the flash sequence.
dull_hold_time the amount of time, in units of ~16ms, for which the LED should be held in the dullest part of the flash sequence.
bright_on_time the amount of time, in units of ~30us, for which the LED should be on during the brightest part of the flash sequence.
bright_off_time the amount of time, in units of ~30us, for which the LED should be off during the brightest part of the flash sequence.
bright_hold_time the amount of time, in units of ~16ms, for which the LED should be held in the brightest part of the flash sequence.
ramp_rate the ramp rate for ramping between brightness levels, in units of ~30us per step with 0 being instantaneous (no ramp).

Figure 7.1 Packet Format of iLight (PWM)

And here's an example code in Xcode to send a iLight command to the connected Bluetooth BLE module.

```
- (IBAction)lightLum:(id)sender
{
    if([self.swLight isOn])
    {
        NSMutableData *valData;
        Byte cmd[11];
        cmd[0]=10;
        cmd[1]=00; // cmd id
        cmd[2]=02; // led index
        cmd[3]=02; //02; // switch off / on / breathing
        cmd[4]=(Byte)(self.slideLight.value*2); // 0; // dull on (unit 30us)
        cmd[5]=(Byte)((100-self.slideLight.value)*2); //200; // dull off (unit 30us)
        cmd[6]=62; // dull hold   (unit 16ms)
        cmd[7]=(Byte)(self.slideLight.value*2); // bright on (unit 30us)
        cmd[8]=(Byte)((100-self.slideLight.value)*2); // bright off (unit 30us)
        cmd[9]=62; // bright hold   (unit 16ms)
        cmd[10]=166; // ramp rate (unit 30us)
        valData = [NSData dataWithBytes:cmd length:11];
        [iGate sendData:valData];
    }
}
```