DataExchanger-AT

Command Reference

Version 1.3

Issue 1

[2017-03-26]

Change History

Issue	Date	Changes	Author
1	2017-03-26	1. Based on Version 1.2. 2. Added more DataExchanger Central Role specific commands. 3. Modified some existing command to support multiple connections (for Central Role). 4. Minor corrections	ML

New In This Release

New Features

New commands are added to support DataExchanger Central Role.

- `AT+LCN, AT+LCN?`
- `AT+LCONQ, AT+LCONQ?`
- `AT+LAC, AT+LAC?`

Changes

Some of the existing commands/notifications are modified:

Command / Notification	**Change**	**Notes**
`AT+EC`	New response parsing mode	To support AT response parsing
`AT+I2CRW`	New timestamp parameter in the response	Running clock from MCU recorded when the I2C command is issued on the driver.
`AT+LSND`	Send to specific connection `AT+LSND=<connIdx>,<bytesToSend>`	Since Central Role supports multiple connections, the new `connIdx` parameters will allow users to send data to a specific connection.
`+LRCV`	Indicate which connection the data is coming from `+LRCV:<connIdx>,bytesSent>`	Since Central Role supports multiple connections, the new `connIdx` parameters will indicate users from which specific connection the data are received.
`AT+LSCAN`	Deprecated the “mode” Input Parameter.	RSSI will always be returned.
`AT+LRL?`	Deprecated	
`AT+DL?`	Deprecated	
`AT+LCH` `AT+LCH?`	Changed to: `AT+LFM and AT+LFM?`	No change in functionality

Table of Contents

Introduction

This document provides the list of all DataExchanger-AT commands, responses, and notifications. Please note each DataExchanger-AT firmware may support only a subset of this list. Please refer to the corresponding firmware release note.

Definitions

The following syntactical definitions apply:

<...> - name enclosed in angle brackets is a syntactical element.

- <CR> - Carriage Return, ASCII code 13 in decimal.
- <LF> - Line Feed, ASCII code 10 in decimal
- <errCode> - see Error Code Table
- for others, see their parameters definitions

[...] - optional single (or group of) parameter(s) of a command

Documentation Format

Each command is described using a table with entries like below:

AT+Command

Description	Brief description of the command
Syntax	Syntax of the command
Parameters	Parameter definitions
Response	Command response
Role	Central or Peripheral roles
Default	Default value for the set variables
Persistent	Whether the variables are stored in the flash or not
Example	`Usage Examples`
Note	Detail description, usage notes, etc
Related	Related commands

General Syntax

AT commands always starts with prefix "AT+" and terminates with either <CR><LF> or <LF> only character. Here are the types of commands:

Command Type	Example
Set Command	AT+UART=115200,8,1,0,0
Query Command	AT+UART?
Execute Command	AT+LSTOP

There are three types of commands:

- Set Command:
  - set variables with the input parameters
    - multiple parameters are separate with a comma.
  - always contains a '=' character in between the command name and the parameter list
  - commands can be executing a function or querying some data or just setting parameters.
- Query Command:
  - query the variables stored for the command function
  - always ended with a '?' Character
  - support verbose or non verbose query response. See the Verbose section for details.
- Execute Command:
  - execute a function

A response is always returned when a command is executed. Some responses are simply OK or ERR with error code. Some responses (like all from query commands) will return specific data. Those responses that return specific data are called query responses. Query response is always prefixed with "+" and followed by a name that in most cases is the originated command name. All responses are terminated with <CR><LF>.

Command Type	Response Example
Set Command	>AT+DTP=1,1 ERR=6
Execute Command	>AT+LSTOP OK
Query Command	>AT+UART? +UART:115200,8,1,0,0 OK

Notification is a special query response without being triggered by a command – i.e. notification is unsolicited. Notification shares the same format of query response. In fact some notifications and query responses are exactly the same.

Notification	Example
Connection Status Change	+LCONN:1,0,01:02:03:04:05:06
User Data Received	+LRCV:0,10
Entered Command Mode	+CMDM

Please note: space is not allowed in the AT commands.

Termination

All commands should be terminated with either <CR><LF> or <LF> only character for the system parser to recognize the completion of command input. All responses and notifications returned by the system are terminated with <CR><LF>.

Verbose

Verbose is an option to display machine-optimized and human readable responses. When the Verbose option is turned off, only the machine-optimized responses will be returned, otherwise both machine-optimized and human readable response will be returned. For example:

Command	Response (Verbose Off)	Response (Verbose On)

AT+UART?	`+UART:115200,8,1,0,0`	`+UART:115200,8,1,0,0`
		`Baud [112500]`
		`Data [8bit]`
		`StopBits [1bit]`
		`Parity [None]`
		`FlowCtrl [No]`

Verbose option can be turned on using "`AT+VB=1`" command. The default is off. Due to memory limitation, some firmware may not support verbose mode. Please check the firmware release note for details.

Echo

Echo is an option to allow the input characters be echoed to the output interface. This is useful for human users to know what they have typed when constructing the command. On the other hand, it would be better to turn it off for machine users to simplify parsing. Echo option can be turned off using "`AT+EC=0`" command. The default is on (`AT+EC=1`).

Three additional special echo modes (2, 3 and 4) are introduced to support AT response parsing. The following table summarized this command options:

Modes	Echo Off	Prefixed "AT+"	Suppressed ERR	Suppressed OK
`AT+EC=0`	Yes	No		
`AT_EC=1`	No			
`AT+EC=2`	Yes		No	No
`AT+EC=3`			Yes	No
`AT+EC=4`			Yes	Yes

Startup String

When the system powers up or has been through a system reset, a "+RDY" string will be sent out to indicate the system is ready for command and operation. This string will only be sent when the system is in the command line mode at the time of startup.

Test Command

User can send "AT" to test the command parser. The system will response "OK" to indicate that it is ready for next command.

Error Codes

The following table is the error codes and their definitions

Error Code	Definitions
0	Success
1	Invalid Command
2	Invalid Parameter
3	Conversion Overflow
4	Conversion Error
5	Insufficient Parameters
6	Too Many Parameters
7	Not Supported
8	Not Allocated
9	Has Been Occupied
10	Mask Conflict
11	Internal Error
12	Parser Buffer Overflow
13	Pin Allocation Conflict
14	Parameters Out Of Range
15	Wake Up Pin Not Assigned
16	Invalid pin ID
17	Already Connected
18	Invalid index
19	Connection Full
20	Busy
21	Insufficient Memory
22	Read Length Exceed Maximum

Error Code	Definitions
23	Bus Fault
24	Missing Parameters
25	Port Opened
26	Port Closed
27	Flash Operation Failed
28	Advertising in progress
29	Advertising in progress
30	IO Expander Conflict (7x7 package detected)
31	IO Expander I2C not open (AT+I2CC must be called 1st)
32	IO Expander I2C Error
33	IO Expander Interrupt Pin Allocated
34	Pin conflict
35	Out of Range

Command Editing

Backspace <BS> character is supported. User can use backspace to delete the last input characters.

Description of Commands

This section describes the list of AT commands and their responses. The commands are listed in alphabetic order.

Command	Central Role Device	Peripheral Role Device
AT+DTP=, AT+DTP?	Yes	Yes
AT+EC=, AT+EC?	Yes	Yes
AT+GIOC=, AT+GIOC?	Yes*	Yes*
AT+GIOD=, AT+GIOD?	Yes*	Yes*
AT+GIOE=, AT+GIOE?	Yes*	Yes*
AT+GIOI?	Yes*	Yes*
AT+GIOO=, AT+GIOO?	Yes*	Yes*
AT+I2CC=, AT+I2CC?	Yes*	Yes*
AT+I2CRW=	Yes*	Yes*
AT+LADR?	Yes*	Yes*
AT+LAS=	Yes	No
AT+LAVP=, AT+LAVP?	No	Yes
AT+LAVQ=	Yes	No
AT+LAC=, AT+LAC?	Yes	No
AT+LCM=, AT+LCM?	No	Yes
AT+LCN=, AT+LCN?	Yes	No
AT+LCON?	Yes	Yes
AT+LCONA=	Yes	No
AT+LCONI=	Yes	No
AT+LCONN[=]	No	Yes
AT+LCONN?	Yes	Yes
AT+LCONP=, AT+LCONP?	No	Yes
AT+LCONQ=, AT+LCONQ?	Yes	No
AT+LFM=, AT+LFM?	Yes	No
AT+LSND=	Yes	Yes
AT+LSCAN[=], AT+LSCAN?	Yes	No
AT+LSTOP[=]	Yes	Yes
AT+LSTOPC	No	Yes
AT+NM?	Yes	Yes
AT+ORGL	Yes	Yes
AT+PWMC=, AT+PWMC?	Yes*	Yes*

AT+PWM=, AT+PWM?	Yes*	Yes*
AT+RBUF?	Yes	Yes
AT+RST	Yes	Yes
AT+SLP[=]	No	Yes
AT+SYSP, AT+SYSP?	Yes	Yes
AT+UART=, AT+UART?	Yes	Yes
AT+VB=, AT+VB?	Yes*	Yes*
AT+VS?	Yes	Yes

* - Optional check firmware release note.

AT+DTP=

Description	Select user data presentation mode
Syntax	AT+DTP=<mode>
Parameters	<mode>: 0 - binary 1 - compact hex 2 - formatted hex
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	Compact Hex (1)
Persistent	No
Example	`>AT+DTP=2  (formatted hex mode)` `OK`
Note	Data presentation mode determines how user data following +LRCV notification are presented. See +LRCV notification for details.
Related	+LRCV

AT+DTP?

Description	Query user data presentation mode
Syntax	AT+DTP?
Response	+DTP:<mode>
Parameters	<mode>: 0 - binary 1 - compact hex 2 - formatted hex
Role	Central or Peripheral

Example 1	`>AT+DTP?  (verbose off)`
	`+DTP:1`
	`OK`

Example 2	`>AT+DTP?  (verbose on)`
	`+DTP:1`
	`Compact Hex`
	`OK`

Note	Data presentation mode determines how user data following +LRCV notification are presented. See +LRCV notification for details.

AT+EC=

Description	Turn echo on or off
Syntax	AT+EC=<mode>
Parameters	<mode>: 0 - echo off, no response prefix, keep +OK/+ERR 1 - echo on, no response prefix, keep +OK/+ERR 2 – echo off, "AT+" response prefix, keep OK/ERR 3 – echo off , "AT+" response prefix, keep OK, suppress ERR 4 – echo off, "AT+" response prefix, suppress OK/ERR
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	Echo On (1)
Persistent	No

Example 1	`> (typing AT+EC?)`
	`+EC:0`
	`OK`
	`> (typing AT+EC=1)`
	`OK`
	`>AT+EC? (typing AT+EC?)`
	`+EC:1`
	`OK`

Example 2	`>AT+EC? (typing AT+EC?)`
	`+EC:1`
	`OK`
	`>AT+EC=2 (typing AT+EC=2)`
	`AT+OK`
	`> (typing AT+EC?)`
	`AT+EC:2`
	`AT+OK`
	`> (typing AT+EC=4)`
	`> (typing AT+EC?)`
	`AT+EC:3`

Note	When echo is turned on, each input character will be echoed to the output interface immediately. When echo is turned off, no input character will be echoed. For `mode>=2`, echo is set off, and all subsequent responses will be prefixed with "AT+". For `AT+EC=3`, `AT+ERR` will be suppressed in addition. For `AT+EC=4`, both `AT+OK` and `AT+ERR` will be suppressed.

AT+EC?

Description	Query echo mode

Syntax	AT+EC?
Response	+EC:<mode>
Parameters	<mode>: 0 - echo off with +OK/+ERR 1 - echo on with +OK/+ERR 2 – echo off with "AT+" prefix in response and with +OK/+ERR 3 – echo off with "AT+" prefix in response but without +OK/+ERR
Role	Central or Peripheral

Example 1	`> (typing AT+EC?)`
	`+EC:0`
	`OK`
	`> (typing AT+EC=1)`
	`OK`
	`>AT+EC? (typing AT+EC?)`
	`+EC:1`
	`OK`

Example 2	`>AT+EC? (typing AT+EC?)`
	`+EC:1`
	`OK`
	`>AT+EC=2 (typing AT+EC=2)`
	`AT+OK`
	`> (typing AT+EC?)`
	`AT+EC:2`
	`AT+OK`
	`> (typing AT+EC=3)`
	`> (typing AT+EC?)`
	`AT+EC:3`

Note	When echo is turned on, each input character will be echoed to the output interface immediately. When echo is turned off, no input character will be echoed. For `AT+EC=2` and `AT+EC=3`, echo is set off, and all subsequent responses will be prefixed with "AT+". For `AT+EC=3` only, `AT+OK` or `AT+ERR` will be suppressed.

AT+FLSC=

Description	Enable or disable the external SPI flash subsystem
Syntax	AT+FLSC=<enable>
Parameters	< enable>: • 0 - disable • 1 - enable
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	Enabled

Persistent	No
Example 1	`>AT+FLSC=0 (Disable the external flash subsystem)` `OK`
Note	This command is used to enable or disable the external SPI flash subsystem. Once the subsystem is disabled, the SPI pins could be allocated for other purposes. Please note, if the SPI pins were allocated for some other purpose before re-enabling the subsystem (`AT+FLSC=0`), this command will return error.
Related	AT+FLSC?

AT+FLSC?

Description	Query the state of the external SPI flash subsystem
Syntax	AT+FLSC?
Response	+FLSC:<enable>
Parameters	< enable>: • 0 – disable • 1 - enable
Role	Central or Peripheral

Example 1	`>AT+FLSC?`
	`+FLSC:1`
	`OK`

Note	This command is used to check whether the external flash subsystem is enabled or disabled.
Related	AT+FLSC

AT+GIOC=

Description	Allocate or remove GPIO pins
Syntax	AT+GIOC=<enableMask>[,<disableMask>[,debounceTime]]

Parameters	<enableMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	  • I/O expander is not allowed.
	• Bit=1 means to enable the respective GPIO line
	• Bit=0 means ignore
	<disableMask>:
	• 32 bits unsigned integer

	• For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	• For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means to disable the respective GPIO line
	• Bit=0 means ignore
	• [This parameter is optional]
	<debounceTime>:
	• 16 bits unsigned integer
	• Range is 0 – 65535ms
	• Default is 100ms
	• [This parameter is optional]

Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	No pin will be allocated for GPIO
Persistent	No
Example 1	`>AT+GIOC=0xFFFFFFFF` *(Enable all allocatable GIO lines)* `OK`
Example 2	`>AT+GIOC=2,1` *(Enable DIO1 and disable DIO0)* `OK`
Example 3	`>AT+GIOC=2,2` *(Enable DIO1 and disable DIO1)* `ERR=10`
Note	This command is used to allocate or remove pins for GPIO function. There are few rules to follow: • Pins must be allocated first using this command for further GPIO commands to work. • If some pins have already been allocated for GPIO and are set in <enableMask>, those pins requested for allocation will be ignored and no error will be returned. `AT+GIOC?` can be used to verify the result. • If some pins have not been allocated for GPIO, but are set in <disableMask>, those pins requested for deallocation will be ignored and no error will be returned. • If some pins have been allocated for other systems, e.g. the UART Tx and Rx pins, and set in either <enableMask> and <disableMask>, those pins for allocation or deallocation will be ignored and no error will be returned. • the same pin cannot be set in both <enableMask> and <disableMask>. Otherwise, `ERR=10` will be returned for mask conflict. All pins are allocated as floating input.
Related	AT+GIOD, AT+GIOE, AT+GIOI, AT+GIOO, +GIO

AT+GIOC?

Description	Query GPIO pin allocation status
Syntax	AT+GIOC?
Response	+GIOC:<enableMask>[,<disableMask>]

Parameters	<enableMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means the pin is requested for enabling
	• Bit=0 means the pin is not requested and should leave alone
	<disableMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means the pin is requested for disabling
	• Bit=0 means the pin is not requested and should leave alone

Role	Central or Peripheral

Example 1	`>AT+GIOC=0xFFFF (allocated all available pins)`
	`OK`
	`>AT+GIOC?`
	`+GIOC:0x7FF0 (only DIO4 – DIO14 has been allocated)`
	`OK`

Note	In the examples above, DIOo-DIO3 were occupied by UART subsystem.
Related	AT+GIOD, AT+GIOE, AT+GIOI, AT+GIOO, +GIO

AT+GIOD=

Description	Set GPIO direction and interrupt detection mask
Syntax	AT+GIOD=<pinMask>,<ioDir>[,<posEdge>,<negEdge>[,pullUpMode]]

Parameters	<pinMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package

	• Bit 0 - 30 represents DIO0 – DIO30 • Bit=1 means selected for action • Bit=0 means ignore <ioDir>: • 0 - input, 1 - output <posEdge>: • 0 - clear for positive edge interrupt detection • 1 - set for positive edge interrupt detection • [only apply to GPIO input pins (i.e. ioDir = 0)] <negEdge>: • 0 - clear for negative edge interrupt detection • 1 - set for negative edge interrupt detection • [only apply to GPIO input pins (i.e. ioDir = 0)] <pullUpMode>: • 0 – tri-state • 1 – pull down • 2 – pull up • default is 1 (pull down) • [only apply to GPIO input pins (i.e. ioDir = 0)]
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	All allocated pin are defaulted to input pins and float
Persistent	No

Example 1	`(Set all GPIO lines as output)`
	`>AT+GIOD=0xFFFF,1`
	`OK`

Example 2	`>(Set DIO0 and DIO1 lines as input, and turn on both positive and negative edge interrupt, and pull up)`
	`>AT+GIOD=3,0,1,1,2`
	`OK`

Note	This command is used to set direction of the allocated GPIO pins and to enable interrupt for the input pins. There are few rules to follow: • Pins must be allocated first using `AT+GIOC` command. • If some other pins specified in pinMask that are not allocated for GPIO, they will be ignored and no error will be returned. • If <ioDir> is set, <posEdgeMask> and <negEdgeMask> will be ignored if specified.
Related	AT+GIOC, AT+GIOE, AT+GIOI, AT+GIOO, +GIO

AT+GIOD?

Description	Query GPIO direction and interrupt detection for GPIO pins
Syntax	AT+GIOD?
Response	+GPIOD:<pinMask>, <ioDirMask>, <posEdgeMask>, <negEdgeMask>, <pullDownMask>, <pullUpMask>

Parameters	<pinMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means the pin is allocated for GPIO
	• Bit=0 means the pin is not allocated for GPIO
	<ioDirMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means the pin is set for output
	• Bit=0 means the pin is set for input
	• [only apply to GPIO pins]
	<posEdgeMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit 1 means positive edge interrupt detection is enabled
	• Bit 0 means not enable or not relevant
	<negEdgeMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit 1 means negative interrupt detection is enabled
	• Bit 0 means not enable or not relevant
	<pullDownMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit 1 means set for pull-down
	• Bit 0 means not relevant

	<pullUpMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit 1 means set for pull-up
	• Bit 0 means not relevant

Role	Central or Peripheral

Example 1	`>AT+GIOC=0xFFFF`
	`OK`
	`>AT+GIOC?`
	`+GIOC:0xFFF0` *(DIO0-DIO3 were preoccupied)*
	`OK`
	`>AT+GIOD=0x7000,1 (set to output)`
	`OK`
	`>AT+GIOD=0x0FFF,0,1,1` *(set to input, both edge trigger, and float as default)*
	`OK`
	`>AT+GIOD?`
	`+GIOD:0x00007FF0,0x00007000,0x00000FF0,0x00000FF0,0x00000000, 0x00000000`
	`OK`

Note	In the example above, DIOo-DIO3 were occupied by the UART subsystem.
Related	AT+GIOC, AT+GIOE, AT+GIOI, AT+GIOO, +GIO

AT+GIOE=

Description	Enable/disable I2C IO expander
Syntax	AT+GIOE=<enable>,<i2cAddr>,<maxPins>,<localIntrPinId>,<pullUp>
Parameters	<enable>: • 0 – disable • 1 – enable < i2cAddr >: • i2c address of IO expander < maxPins >: • max number of pins that the IO expander supports < localIntrPinId >: • native IO pin used for IO expander interrupt • Range: 0 – 14, 0xFF (no interrupt) < pullUp >: • 0 – local interrupt pin floating • 1 – local interrupt pin pull down

	• 2 – local interrupt pin pull up
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	None
Persistent	No
Example	`>AT+GIOE=1,0x23,16,8,2` `OK`
Note	This command is used to enable or disable the IO expander. Current it supports only PCA6515 I2C IO expander from Texas Instrument. Please note, for hardware that use 7x7 package of CC2640 cannot enable IO expander.
Related	AT+GIOC, AT+GIOD, AT+GIOI, AT+GIOD, +GIO

AT+GIOE?

Description	Query IO expander state
Syntax	AT+GIOE?
Response	+GIOE=<enable>,<i2cAddr>,<maxPins>,<localIntrPinId>,<pullUp>
Parameters	<enable>: • 0 – disable • 1 – enable < i2cAddr >: • i2c address of IO expander < maxPins >: • max number of pins that the IO expander supports < localIntrPinId >: • native IO pin used for IO expander interrupt • Range: 0 – 14, 0xFF (no interrupt) < pullUp >: • 0 – local interrupt pin floating • 1 – local interrupt pin pull down • 2 – local interrupt pin pull up
Role	Central or Peripheral

Example 1	`>AT+GIOE=1,0x23,16,8,2`
	`OK`
	`>AT+GIOE?`
	`+GIOE:1,35,16,8,2`
	`OK`

Note	This command is used to query the IO expander setting
Related	AT+GIOC, AT+GIOD, AT+GIOI, AT+GIOO, +GIO

AT+GIOI?

Description	Query GPIO input pin state
Syntax	AT+GIOI?
Response	+GIOI:<inPinMask>,<valMask>

Parameters	<inPinMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means the pin is allocated for GPIO input
	• Bit=0 means the pin is not allocated for GPIO input
	<valMask>:
	• 32 bits unsigned integer
	  • For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	  • For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means the pin is read high
	• Bit=0 means the pin is read low
	• [only apply to GPIO input pins]

Role	Central or Peripheral

Example 1	`>AT+GIOC=0xFFFF`
	`OK`
	`>AT+GIOD=0xFFF,0`
	`OK`
	`>AT+GIOI?  (verbose off)`
	`+GIOI:0x00000FF0,0x00000420`
	`OK`

Note	This command is used to query the input pin state.
Related	AT+GIOC, AT+GIOD, AT+GIOE, AT+GIOO, +GIO

AT+GIOO=

Description	Set GPIO output pin state
Syntax	AT+GIOO=<pinMask>,<state>,<pulseTime>
Parameters	<pinMask>: • 32 bits unsigned integer • For 5x5 package

	• Bit 0 -14 represents DIO0 – DIO14 • Bit 15 is NULL • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins) • For 7x7 package • Bit 0 - 30 represents DIO0 – DIO30 • Bit=1 means selected for GPIO • Bit=0 means ignore <state>: • 0 - output low, 1 - output high, 2 – pulse <pulseTime>: • 1ms to 500ms • default is 150ms • This is optional.
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	All output pins are set with output low
Persistent	No

Example	`>AT+GIOC=0xFFFF`
	`OK`
	`>AT+GIOD=0xF000,1`
	`OK`
	`>AT+GIOO=0x6000,1` *(Turn on)*
	`OK`
	`>AT+GIOO=0x6000,2` *(Turn off then back on)*
	`OK`

Note	This command is used to set state of the allocated GPIO output pins.
Related	AT+GIOC, AT+GIOD, AT+GIOE, AT+GIOI, +GIO

AT+GIOO?

Description	Query GPIO output pin state
Syntax	AT+GIOO?
Response	+GIOD:<outPinMask>,<stateMask>
Parameters	<outPinMask>: • 32 bits unsigned integer • For 5x5 package • Bit 0 -14 represents DIO0 – DIO14 • Bit 15 is NULL • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins) • For 7x7 package • Bit 0 - 30 represents DIO0 – DIO30 • Bit=1 means the pin is allocated for GPIO output • Bit=0 means the pin is not allocated for GPIO output < stateMask >: • 32 bits unsigned integer

	• For 5x5 package
	  • Bit 0 -14 represents DIO0 – DIO14
	  • Bit 15 is NULL
	  • Bit 16 – Bit 31 is NULL or expander I/O (max 16 pins)
	• For 7x7 package
	  • Bit 0 - 30 represents DIO0 – DIO30
	• Bit=1 means the pin is set for output high
	• Bit=0 means the pin is set for output low
	• [only apply to GPIO output pins]

Role	Central or Peripheral
Example 1	>AT+GIOC=0xFFFF OK >AT+GIOD=0xF000,1 OK >AT+GIOO=0x6000,1 OK >AT+GIOO? +GIOO:0x00007000,0x00006000 OK
Note	This command is used to query the state of the output pin.
Related	AT+GIOC, AT+GIOD, AT+GIOE, AT+GIOI, +GIO

AT+I2CC=

Description	Enable or disable I2C port
Syntax	AT+I2CC=<portId>,<enable>,[<fastSpeed>,<sdaPin>,<sclPin>]
Parameters	<portId>: • I2c port id <enable>: • 0 – disable • 1 – enable <fastSpeed>: • 0 – low speed 100KHz • 1 – high speed 400KHz • [This parameter is optional] < sdaPin >: • SDA pin id • Range 0 to max GPIO – 1 • [This parameter is optional] < sclPin >: • SCL pin id • Range 0 to max GPIO – 1 • [This parameter is optional]
Response	OK or ERR=<errCode>
Role	Central or Peripheral

Default	All I2C ports are disabled
Persistent	No
Example 1	`>AT+I2CC=0,1  (Enable I2C port 0)` `OK`
Example 2	`>AT+I2CC=1,0  (Disable I2C port 1)` `OK`
Example 3	`>AT+I2CC=0,1,1,4,5 (Enable I2C port 0 with high speed and using pin4 for SDA and pin5 for SCL)` `OK`
Note	This command is used to enable or disable I2C function. I2C port must be enabled before AT+I2CRW command can be used.
Related	AT+I2CC?, AT+I2C

AT+I2CC?

Description	Query I2C port status
Syntax	AT+I2CC?
Response	+I2CC:<portId>,<enable>,<sdaPin>, <sclPin>[;...]
Parameters	<portId>: • i2c port id <enable>: • 0 – disable • 1 – enable < sdaPin >: • SDA pin id < sclPin >: • SCL pin id
Role	Central or Peripheral

Example 1	`>AT+I2CC?`
	`+I2CC:0,0,255,255`
	`OK`

Note	.

AT+I2CRW=

Description	Issue I2C command
Syntax	AT+I2C=<portId>,<chipAddr>,<readLen>[,<writeString>]
Parameters	<portId>: • I2c port id <chipAddr>: • I2c chip device address

	<readLen>: • Number of bytes to read • [0 means no read] • max 32 bytes <writeString> • hex codes • optional space in between codes; if no space, each code must be two hex digits. • [This is an optional parameter]
Response 1	OK
Response 1	+I2CRW:<timeStamp>,<readByteStr> OK
Response 2	ERR=<errCode>
Role	Central or Peripheral
Default	No
Persistent	No
Example 1	`>AT+I2CRW=0,23,0,ABCD (Write 0xAB 0xCD)` `OK`

Example 2	`>AT+I2CRW=0,23,3,DE BD (Write 0xDE 0xBD, then read 3 bytes)`
	`+I2CRW:0x756FB,01 02 03`
	`OK`

Example 3	`>AT+I2CRW=0,23,4 (Read 4 bytes)`
	`+I2CRW:0xAB678,01 02 03 04`
	`OK`

Note	This command is used to issue read/write I2C commands
Related	AT+I2CC, AT+I2CC?

AT+LAC=

Description	Set active connection (Central Role)
Syntax	AT+LAC=< connIdx >
Parameters	< connIdx >: • connection index
Response	OK or ERR=<errCode>
Role	Central
Default	`connIdx=0` when the first connection is established
Persistent	No
Example 1	`>AT+LAC=0` `OK`
Example 2	`>AT+LAC=1` `ERR=18  (connIdx=1 is not connected)`

Note	This command is used to switch the active connection before entering to data mode. When UART is set (via the CLI/Data Mode Switch control pin) to data mode, there is only one connection the UART can attach and communicate on. While in CLI mode, the send (AT+LSND) and receive (+LRCV) are already channelized, therefore this command is unnecessary.
Related	AT+LAC?

AT+LAC?

Description	Query active connection (Central Role)
Syntax	AT+LAC?
Response	+LAC:<connIdx>
Parameters	< connIdx >: • connection index
Role	Central

Example 1	`>AT+LAC?`
	`+LAC:0`
	`OK`

Note	This command is used to query the active connection.
Related	AT+LAC

AT+LADR?

Description	Query local Bluetooth address
Syntax	AT+LADR?
Response	+LADR:<btaddr>
Parameters	< btaddr >: • Bluetooth address
Role	Central or Peripheral

Example 1	`>AT+LADR?`
	`+LADR:C4:BE:84:7E:AA:00`
	`OK`

Note	This command is used to query local Bluetooth address.

AT+LAS=

Description	Enable/Disable Auto-Scan
Syntax	AT+LAS=<enable>

Parameters	< enable >: • 0 – disable auto-scan • 1 – enable auto-scan
Response	OK or ERR=<errCode>
Role	Central
Default	Auto Scan at startup can be set via AT+LCN command
Persistent	No.
Example	`>AT+LAS=0 (disable auto-scanning now)` `OK`
Note	This command is used to change the current auto-scanning policy. Please note this command is not persistent. For the power-up auto-scan policy, please use AT+LCN command instead.

AT+LAVP=

Description	Set connection advertising parameters
Syntax	AT+LAVP=<advIntervalFast>,<advIntervalSlow>, <advDurationFast>,<advDurationSlow>,<connectable>,<advPayload>
Parameters	<advIntervalFast> • advertising interval in fast cycle • Range: 20ms to 10240ms • Unit: millisecond <advIntervalSlow> • advertising interval in slow cycle • Range: 20ms to 10240ms • Unit: millisecond <advDurationFast> • Duration for fast advertising cycle • Range: 0ms to 65534ms; 65535 (0xFFFF) indicates non-stop • Unit: millisecond <advDurationSlow> • Duration for slow advertising cycle • Range: 0ms to 65534ms; 65535 (0xFFFF) indicates non-stop • Unit: millisecond <connectable> • 0 – non-connectable advertising • 1 – connectable advertising <advPayload> • max 31 bytes of advertising packet payload 2 digits hex string • Optional
Response	OK or ERR=<errCode>
Role	Peripheral
Default	Advertising Interval Fast Cycle = 30ms Advertising Interval Slow Cycle = 100ms

	Advertising Duration Fast Cycle = 30000ms (30s) Advertising Duration Slow Cycle = Inf (non stop) Connectable = 1 [These value may be changed in different firmware application. Please check the firmware release note]
Persistent	Yes
Example 1	`>AT+LAVP=0,30,100,30000,65535,1` `OK`
Example 2	`>AT+LAVP=0,100,100,30000,65535,0,0201061AFF4C000215B9407F30F5F8466EAFF925556B57FE6D0049000AC5` `OK`
Note 1	This command is used to set advertising parameters. Advertising cycle is the stage to invite other devices to make connections. The system supports two cycles – fast and slow. Fast advertising cycle is the first cycle to run. Slow advertising cycle may run after fast cycle is completed. Each cycle is governed by two properties – advertising interval and advertising duration. Advertising interval determines how often advertising packets are sent, and advertising duration determines how long the adverting cycle is last.
Note 2	Advertising interval cannot be greater than advertising duration.
Note 3	When advertising duration is set to zero, the cycle will be disabled.
Note 4	Advertising payload can be specified in a sequence of two digits hex string to represent max 31 bytes.

AT+LAVP?

Description	Query connection advertising parameters
Syntax	AT+LAVP?
Response	+LAVP:<advIntervalFast>, <advIntervalSlow>, <advDurationFast>, <advDurationSlow>, <connectable>, <advPayload>
Parameters	<advIntervalFast> • advertising interval in fast cycle • Range: 20ms to 10240ms • Unit: millisecond <advIntervalSlow> • advertising interval in slow cycle • Range: 20ms to 10240ms • Unit: millisecond <advDurationFast> • Duration for fast advertising cycle • Range: 0ms to 65534ms; 65535 (0xFFFF) indicates non-stop • Unit: millisecond <advDurationSlow> • Duration for slow advertising cycle • Range: 0ms to 65534ms; 65535 (0xFFFF) indicates non-stop

	• Unit: millisecond <connectable> • 0 – non-connectable advertising • 1 – connectable advertising <advPayload> • max 31 bytes of advertising payload in 2 digits hex string
Role	Peripheral
Example 1	>AT+LAVP? *(verbose off)* +LAVP:100,100,30000,65535,0,02 01 06 1A FF 4C 00 02 15 B9 40 7F 30 F5 F8 46 6E AF F9 25 55 6B 57 FE 6D 00 49 00 0A C5 OK
Example 2	>AT+LAVP? *(verbose on)* +LAVP:100,100,30000,65535,0,02 01 06 1A FF 4C 00 02 15 B9 40 7F 30 F5 F8 46 6E AF F9 25 55 6B 57 FE 6D 00 49 00 0A C5 Advertising Fast Interval [100ms] Advertising Slow Interval [100ms] Advertising Fast Duration [30000ms] Advertising Slow Duration [Inf] Connectable [0] Advertising Payload [02 01 06 1A FF 4C 00 02 15 B9 40 7F 30 F5 F8 46 6E AF F9 25 55 6B 57 FE 6D 00 49 00 0A C5] OK
Note	This command is used to query the advertising parameters.

AT+LAVQ=

Description	Request advertising packet dump
Syntax	AT+LAVQ=<scanIdx>
Response 1	+LAVQ:<scanIdx>,<advPktBytes> OK
Response 2	ERR=35 (out of range)
Parameters	< scanIdx >: • scan index from AT+LSCAN? < advPktBytes >: • 31 bytes advertsing packet bytes
Role	Central
Example	>AT+LAVQ=0 +LAVQ:0,02 01 06 1A … OK
Note	This command is used to get the scanned advertising packet bytes.

AT+LCM=

Description	Set peripheral role connection policy
Syntax	AT+LCM=<autoConnect>,<whitelistPolicy>, <connectAfterDisconnect>
Parameters	< autoConnect >: • 0 - don't connect after power up • 1 - connect after power up < whitelistPolicy >: • 0 – scan & connect to any device [ALL] • 1 – scan whitelisted device only but connect to any device [SCAN] • 2 – scan any device but connect to whitelist device only [CONN] • 3 – scan and connect to whitelist device only [SCAN+CONN] < connectAfterDisconnect > • 0 – don't connect after disconnect by remote • 1 – connect after disconnect by remote • [this applies to remote disconnect but not to local disconnect using the AT+LSTOP command]
Response	OK or ERR=<errCode>
Role	Peripheral
Default	Auto connect after power up = Yes Whitelist Policy = Connect To Any Device Connect After Disconnect = Yes
Persistent	Yes
Example	`>AT+LCM=1,0,1` `OK`
Note	This command is used to set peripheral role connection policy including power up auto-connect, whitelist policy, and connect after remote disconnect.
Related	AT+LCM?

AT+LCM?

Description	Query peripheral role policy
Syntax	AT+LCM?
Response	+LCM:<autoConnect>,<whitelistPolicy>, < connectAfterRemoteDisconnect>
Parameters	< autoConnect >: • 0 - don't connect after power up • 1 - connect after power up < whitelistPolicy >: • 0 – connect to any device

	• 2 – connect to whitelist device only < connectAfterDisconnect > • 0 – don't connect after disconnect by remote • 1 – connect after disconnect by remote • [this applies to remote disconnect but not to local disconnect using the AT+LSTOP command]
Role	Peripheral
Example 1	>AT+LCM=1,0,1 OK >AT+LCM? *(verbose off)* +LCM:1,0,1 OK
Example 2	>AT+LCM=1,0,1 OK >AT+LCM? *(verbose on)* +LCM:1,0,1 Power On Connect [Yes] Whitelist Policy [All] Connect After Disconnect [Yes] OK
Note	This command is used to set peripheral role connection policy including power up auto-connect, whitelist policy, and connect after remote disconnect.
Related	AT+LCM

AT+LCN=

Description	Set central role connection policy
Syntax	AT+LCN=<autoContScan>,<whitelistPolicy>,<multiConnPolicy>, <connectAfterDisconnect>,< discoverUsingServiceUUID>, <pwrToAutoConnect>
Parameters	< autoContScan >: o 0 - don't continuously scan after power up • 1 – continuously scan after power up < whitelistPolicy >: • 0 – scan any device • 1 – scan whitelist device only < multiConnPolicy >: • 0 – don't switch to next valid connection • 1 – switch to next valid connection < connectAfterDisconnect > • 0 – don't connect after disconnect by remote • 1 – connect after disconnect by remote • [this applies to remote disconnect but not to local disconnect using the AT+LSTOP command]

	< discoverUsingServiceUUID >: • 0 – discovere using any service UUID • 1 – discover using the default UUID < pwrToAutoConnect >: • minimum RSSI to auto connect after scanning • range -127 to 0
Response	OK or ERR=<errCode>
Role	Central
Default	Auto scan after power up = Yes [1] Whitelist Policy = Scan Any Device [0] Channel Policy = Switch To Next Valid Channel [1] Connect After Disconnect = Yes [1] Discover Using Service UUID = Yes [1] Power To Auto-Connect = -50 (dBm)
Persistent	Yes
Example	`>AT+LCN=1,0,1,1,1,-50` `OK`
Note	This command is used to set central role connection policy including power up continuous scan, whitelist policy, multi-connection policy, connect after remote disconnect, and power to auto connect. When Power-Up-Continuous-Scan is set, it will start scanning upon power up. When Multi-Connection-Policy is set, it will attempt to switch to the next valid connection if the current connection is about disconnected. When Connect-After-Remote-Disconnect is set, it will attempt to reconnect to the same device after the connection to that device is disconnected. Power-To-Auto-Connect is the minimum power level to determine whether the connection should be made with the scanned device.
Related	AT+LCN?

AT+LCN?

Description	Query central role connection policy
Syntax	AT+LCN?
Response	+LCN=<autoContScan>,<whitelistPolicy>,< multiConnPolicy >, <connectAfterDisconnect>,< discoverUsingServiceUUID>, <pwrToAutoConnect>
Parameters	< autoContScan >: o 0 - don't continuously scan after power up • 1 – continuously scan after power up < whitelistPolicy >: • 0 – scan any device • 1 – scan whitelist device only

	< multiConnPolicy >: • 0 – don't switch to next valid connection • 1 – switch to next valid connection < connectAfterDisconnect > • 0 – don't connect after disconnect by remote • 1 – connect after disconnect by remote • [this applies to remote disconnect but not to local disconnect using the AT+LSTOP command] < discoverUsingServiceUUID >: • 0 – discovere using any service UUID • 1 – discover using the default UUID < pwrToAutoConnect >: • minimum RSSI to auto connect after scanning • range -127 to 0
Role	Central

Example 1	`>AT+LCN?`
	`+LCN:1,0,1,1,1,-50`
	`OK`

Note	This command is used to set central role connection policy including power up continuous scan, whitelist policy, multi-connection policy, connect after remote disconnect, and power to auto connect. When Power-Up-Continuous-Scan is set, it will start scanning upon power up. When Multi-Connection-Policy is set, it will attempt to switch to the next valid connection if the current connection is about disconnected. When Connect-After-Remote-Disconnect is set, it will attempt to reconnect to the same device after the connection to that device is disconnected. Power-To-Auto-Connect is the minimum power level to determine whether the connection should be made with the scanned device.
Related	AT+LCN

AT+LCON?

Description	Query connection state
Syntax	AT+LCON?
Response	+LCON:<state_1>[,<state_2>[,<state_3>]] ...
Parameters	<state_#>, where # is the connection index.: • 0 - idle • 1 - connecting • 2 – connected • 3 – connected ready (central role only)
Role	Central or Peripheral
Example 1	`>AT+LCON? (Peripheral Role with 1 connection)` `+LCON:1`

	OK
Example 2	>AT+LCON? (Central Role with 3 connections) +LCON:1,0,0 OK
Note	This command is used to query the connection states for both central and peripheral roles connection

AT+LCONA=

Description	Start connecting in Central Role using BT address
Syntax	AT+LCONA=<btAddr>[,<timeout>]
Parameters	<btAddr>: • BT Address <timeout>: • duration to connect • Unit: millisecond • Default: forever
Response	OK or ERR=<errCode>
Role	Central
Example 1	>AT+LCONA=C4BE847EAA00,30000 (try to connect to device with C4:BE:84:7E:AA:00 BT address for 30s) OK
Example 2	>AT+LCONI= C4BE847EAA00 (try to connect to device with C4:BE:84:7E:AA:00 BT address forever) OK
Note	This command is used to start connection in Central Role using device BT address.
Related	AT+LCN, AT+LCN?, AT+LCON?, AT+LCONI, AT+LCONQ, AT+LCONQ?, AT+LSTOP

AT+LCONI=

Description	Start connecting in Central Role using scan list index
Syntax	AT+LCONI=<scanIdx>[,<timeout>]
Parameters	<scanIdx>: • index from AT+LSCAN? Results • Range 1 to number of scanned devices <timeout>: • duration to connect • Unit: millisecond • Default: forever
Response	OK or ERR=<errCode>

Role	Central
Example	>AT+LSCAN OK +LSCAN:2,1 >AT+LSCAN? +LSCAN2,1 +LSCANI:0, C4:BE:84:7E:AA:00,-65 >AT+LCONI=0,30000 (try to connect the 1st device in the scanned list for 30s) OK
Note	This command is used to start connection in Central Role using index of scan device list.
Related	AT+LCN, AT+LCN?, AT+LCON?, AT+LCONA, AT+LCONQ, AT+LCONQ?, AT+LSTOP

AT+LCONN[=]

Description	Start connecting in peripheral role
Syntax	AT+LCONN[=<timeout>]
Parameters	<timeout>: • duration to connect • Unit: millisecond
Response	OK or ERR=<errCode>
Role	Peripheral
Example 1	>AT+LCONN=30000 (try to connect for 30s) OK
Example 2	>AT+LCONN (using the default advertising parameters) OK
Note	This command is used to start connection in Peripheral Role with optional parameter. If timeout is not provided, the current advertising duration will be used.
Related	AT+LCM, AT+LCON?, AT+LCONP, AT+LCONP?, AT+LSTOP, AT+LSTOPC, AT+LAVP

AT+LCONN?

Description	Query live connection status
Syntax	AT+LCONN?
Response	+LCONN:<state>,<connIdx>,<btAddr>
Parameters	<state>: • 0 - idle • 1 - connecting

	• 2 – connected • 3 – connected ready (central role only) <connIdx>: • connection index <btAddr>: • remote BLE address
Role	Central or Peripheral

Example 1	`>AT+LCONN? (Peripheral Role)`
	`+LCONN:2,0,01:02:03:04:05:06`
	`OK`

Example 2	`>AT+LCONN? (Central Role)`
	`+LCONN:3,0,01:02:03:04:05:06`
	`+LCONN:1,1,0A:0B:0C:0D:0E:0F`
	`OK`

Note	This command is used to query the current state of each connection.
Related	AT+LCON?, AT+LCONA, AT+LCONI, AT+LCONN, AT+LSTOP, AT+LSTOPC, +LCONN

AT+LCONP=

Description	Set peripheral role connection parameters
Syntax	AT+LCONP=<ciInMSec>,<slaveLatency>,<timeout>,<shouldUpdate>
Parameters	<ciInMSec>: • Connection interval • Unit: millisecond • Range: 7 to 4000 <slaveLatency>: • Slave latency • Range: 0 to 4 <timeout>: • Time wait before declaring disconnect for no activity • Unit: millisecond • Range: 100 to 32000 • Default: 6000 <shouldUpdate>: • Request for connection parameter update • 0 - no, 1 – yes
Response	OK or ERR=<errCode>
Role	Peripheral
Default	Connection Interval = 40ms Slave Latency = 0 Connection Timeout = 6000ms Update Parameters = Yes [These value may be changed in different firmware application. Please check

	the firmware release note]
Persistent	Yes
Example	`>AT+LCONP=40,0,6000,1` `OK`
Note	This command is used to set connection parameters. Connection Interval is 1.25ms base slot. If input is not multiple of 1.25ms, it will always be rounded up to the next slot. This commands applies to peripheral role only.
Related	AT+LCM, AT+LCON?, AT+LCONN, AT+LCONP?, AT+LSTOP, AT+LSTOPC, AT+LAVP

AT+LCONP?

Description	Query peripheral role connection parameters
Syntax	AT+LCONP?
Response	+LCONP:<ciInMSec>,<slaveLatency>,<timeout>,<shouldUpdate>
Parameters	<ciInMSec>: • Connection interval • Unit: millisecond • Range: 7 to 4000 <slaveLatency>: • Slave latency • Range: 0 to 4 <timeout>: • Time wait before declaring disconnect for no activity • Unit: millisecond • Range: 100 to 32000 • Default: 6000 <shouldUpdate>: • Request for connection parameter update • 0 - no, 1 – yes
Role	Peripheral

Example 1	`>AT+LCONP?` *(verbose off)*
	`+LCONP:40,0,6000,1`
	`OK`

Example 2	`>AT+LCONP?` *(verbose on)*
	`+LCONP:40,0,6000,1`
	`Interval [40ms]`
	`Latency [0]`
	`Timeout [6000ms]`
	`Update [Yes]`
	`OK`

Note	This command is used to query the peripheral role connection parameters.

Related	AT+LCM, AT+LCON?, AT+LCONN, AT+LCONP, AT+LSTOP, AT+LSTOPC, AT+LAVP

AT+LCONQ=

Description	Set central role connection parameters
Syntax	AT+LCONQ=<ciInMSec>,<slaveLatency>,<timeout>,<shouldUpdate>
Parameters	<ciInMSec>: • Connection interval • Unit: millisecond • Range: 7 to 4000 <slaveLatency>: • Slave latency • Range: 0 to 4 <timeout>: • Time wait before declaring disconnect for no activity • Unit: millisecond • Range: 100 to 32000 • Default: 6000 <shouldUpdate>: • Request for connection parameter update • 0 - no, 1 – yes
Response	OK or ERR=<errCode>
Role	Central
Default	Connection Interval = 40ms Slave Latency = 0 Connection Timeout = 6000ms Update Parameters = Yes [These value may be changed in different firmware application. Please check the firmware release note]
Persistent	Yes
Example	`>AT+LCONQ=40,0,6000,1` `OK`
Note	This command is used to set connection central role parameters. Connection Interval is 1.25ms base slot. If input is not multiple of 1.25ms, it will always be rounded up to the next slot. This commands applies to central role only.
Related	AT+LCN, AT+LCN?, AT+LCON?, AT+LCONA, AT+LCONI, AT+LCONQ?, AT+LSTOP

AT+LCONQ?

Description	Query central role connection parameters

Syntax	AT+LCONQ?
Response	+LCONQ:<ciInMSec>,<slaveLatency>,<timeout>,<shouldUpdate>
Parameters	<ciInMSec>: • Connection interval • Unit: millisecond • Range: 7 to 4000 <slaveLatency>: • Slave latency • Range: 0 to 4 <timeout>: • Time wait before declaring disconnect for no activity • Unit: millisecond • Range: 100 to 32000 • Default: 6000 <shouldUpdate>: • Request for connection parameter update • 0 - no, 1 – yes
Role	Central

Example 1	`>AT+LCONQ?  (verbose off)`
	`+LCONP:40,0,6000,1`
	`OK`

Example 2	`>AT+LCONQ?  (verbose on)`
	`+LCONP:40,0,6000,1`
	`Interval [40ms]`
	`Latency [0]`
	`Timeout [6000ms]`
	`Update [Yes]`
	`OK`

Note	This command is used to query the central role connection parameters.
Related	AT+LCN, AT+LCN?, AT+LCON?, AT+LCONA, AT+LCONI, AT+LCONQ, AT+LSTOP

AT+LFM=

Description	Change forwarding mode
Syntax	AT+LFM=<mode>
Parameters	< mode >: • 1 – forward to remote data channel • 2 – forward to remote command channel • 0 – forward to remote data channel with command channel disabled
Response	OK or ERR=<errCode>
Role	Central
Default	2 – forward to remote command channel. When connected, all bytes received via UART will be forwarded to the remote command channel – i.e. accessing

	remote CLI
Persistent	No
Example 1	[Purpose: access remote CLI] [Verify for local access] >AT+NM? +NM:DXM,TIEM,B0B448DA488F,GT-tronics OK >AT+LFM=2 OK >+AT+LSCAN OK +LSCAN:2,1 >AT+LSCAN? +LSCAN:2,1 +LSCANI:0, 01:02:03:04:05:06,-45 OK >AT+LCONI=0 OK +LCONN:1,0,01:02:03:04:05:06 +LCONN:2,0,01:02:03:04:05:06 +LCONN:3,0,01:02:03:04:05:06 [Set CLI and Data mode pin to low] +DATM [Any bytes input here will go to the peripheral's AT command parser – i.e. remote CLI] [Verify for remote access] AT+NM? +NM:DXS,TIEM,B0B448DA3777,GT-tronics OK [Continue on Example 2 …]
Example 2	[Purpose: send data to remote data channel] [Set CLI and Data mode pin to high] +CMDM [back to local CLI] AT+LFM=1 [Set CLI and Data mode pin to low] +DATM [any bytes input here will go to the peripheral's data channel (UART)] [Continue on Example 3 …]
Example 3	[In Example 1 and 2, the peripheral CLI is locked out for the Central Role device until the connection is closed. Therefore, the peripheral UART only can forward to data channel] [Purpose: allow the peripheral to access its local CLI] [Set CLI and Data mode pin to high] +CMDM

	[back to local CLI] >AT+LSTOP (need to stop connection 1st) +LCONN:4,0,01:02:03:04:05:06 +LCONN:5,0,01:02:03:04:05:06 >AT+LFM=0 >AT+LCONI=0 (reconnect) +LCONN:1,0,01:02:03:04:05:06 +LCONN:2,0,01:02:03:04:05:06 +LCONN:3,0,01:02:03:04:05:06 [Set CLI and Data mode pin to low] +DATM [any bytes input here will go to the peripheral's data channel (UART). However, the peripheral device can choose its local data or command channel]
Note	This command is used to decide how the bytes received from UART should be forwarded on a Central Role device. Please see the examples above for details.
Related	AT+LFM?

AT+LFM?

Description	Query forwarding mode on data channel
Syntax	AT+LFM?
Response	+LFM:<mode>
Parameters	< mode >: • 1 – forward to remote data channel • 2 – forward to remote command channel • 0 – forward to remote data channel with command channel disabled
Role	Central
Example	>AT+LFM? +LFM:2 OK
Note	This command is used to query the connection forwarding mode.
Related	AT+LFM

AT+LSND=

Description	Send user data
Syntax	AT+LSND=<connIndex>,<byteToSend>
Parameters	< connIndex >: • connection to which the user data will send <byteToSend>:

	• number of bytes to send
Response	OK or ERR=<errCode>
Role	Central or Peripheral

Example	`>AT+LSND=0,20`
	`OK`
	`> ... (send 20 raw bytes here)`
	`+LSND:0,20`

Note	This command is used to send user data bytes to the data connection under command line. Once the response of "`OK`" returned, user can send the exact amount of bytes specified in the command before. The system will not return to parsing mode until the exact amount of sent bytes are received. After the exact amount of byte received, the system will response with "`+LSND:<connIdx>,<byteSent>`" notification. Please note: all the bytes sent after AT+LSND command but before OK will be ignored. Therefore, it is important to wait for the OK response before sending any user data.

AT+LSCAN[=]

Description	Scan LE devices
Syntax	AT+LSCAN[=<maxDev>,<timeout>]
Parameters	<maxDev>: • Maximum number of devices to be scanned • Range: 1 to 20 • Default: 20 <timeout>: • Maximum period of scanning if maxDev is not reached • Unit: millisecond • Range: 1 to 65535 • Default: 5000ms (5s)
Response	OK or ERR=<errCode>
Role	Central

Example 1	`>AT+LSCAN`
	`OK`
	`(after scanning for 5s ...)`
	`+LSCAN:2,0 (no device found)`
	`>AT`
	`OK`

Example 2	`>AT+LSCAN=2,10000`
	`OK`
	`(after 2 devices scanned ...)`
	`+LSCAN:2,2 (two devices found)`
	`>AT`
	`OK`

Note	This command is used to start scanning LE devices.
Related	AT+LSCAN?, +LSCAN, +LSCANI

AT+LSCAN?

Description	Query scanned LE devices and current scanning status
Syntax	AT+LSCAN?
Response	+LSCAN:<state>,<scanItems> +LSCANI:<idx>,<btAddr>,<rssi> ...
Parameters	+LSCAN <state>: • 0 – Idle • 1 – Scanning • 2 – Scan Completed <scanItems> • number of scanned items +LSCANI <idx> • scan index <btAddr> • remote Bluetooth address <rssi> • receive power (dBm)
Role	Central
Example 1	>AT+LSCAN OK *(after scanning for 5s ...)* +LSCAN:2,0 *(notification)* *>AT+LSCAN?* +LSCAN:0,0 >AT OK
Example 2	>AT+LSCAN=3,2 OK *(after 2 devices scanned ...)* +LSCAN:2,2 *(notification)* >AT+LSCAN? +LSCAN:0,2 +LSCANI:0,06:05:04:03:02:01,-64 +LSCANI:1,01:02:03:04:05:06,-89 >AT OK

Note	This command is used to retrieve the scanned device list from the last AT+LSCAN command or the current scanning status. The list will only be refreshed after a new AT+LSCAN is issued or after system reset.
Related	AT+LSCAN, +LSCAN, LSCANI

AT+LSTOP[=]

Description	Stop connecting or disconnect
Syntax	AT+LSTOP[=<connIdx>]
Parameters	< connIdx >: • Connection index • Default: 0xFE (stop all connections)
Response	OK or ERR=<errCode>
Role	Central or Peripheral

Example 1	`>AT+LSTOP (peripheral role)`
	`OK`
	`+LCONN:3`

Example 2	`>AT+LSTOP=1 (central role)`
	`OK`
	`+LCONN:4,1,01:02:03:04:05`
	`+LCONN:5,1,01:02:03:04:05`
	`>AT+LCON?`
	`+LCON:3,5,3`
	`OK`

Note	This command is used to stop connecting, disconnect connections, or scanning.
Related	AT+LCON?, AT+LCONA, AT+LCONI, AT+LCONN, AT+LSCAN, AT+LSCAN?, AT+LSTOPC, +LCONN

AT+LSTOPC

Description	Disconnect and reconnect
Syntax	AT+LSTOPC
Parameters	none
Response	OK or ERR=<errCode>
Role	Peripheral

Example	`>AT+LSTOPC`
	`OK`
	`+LCONN:3,0,01:02:03:04:05:06`
	`+LCONN:1,0,0a:0b:0c:0d:0e:0f`

Note	This command is used to disconnect the current connection and put itself

	in reconnecting mode. This feature would allow a Central Role device to make disconnection on the Peripheral Role device.
Related	AT+LCON?, AT+LCONN, AT+LSTOP, +LCONN

AT+NM?

Description	Query local device info
Syntax	AT+NM?
Response	+NM:<dxRole>,<modelNo>,<devId>,<mfgName>
Parameters	< dxRole >: • DXM – DataExchanger Master (Central Role) • DXS – DataExchanger Slave (Peripheral Role) < modelNo>: • Model Number < devId >: • Device Id < mfgName >: • Manufacture Name
Role	Central or Peripheral

Example	`>AT+NM?`
	`+NM:DXM,TIEM,010203040506,GT-tronics`
	`OK`

Note	This command is used to query the local device info.

AT+ORGL

Description	Factory reset
Syntax	AT+ORGL
Parameters	none
Response	+RDY
Role	Central or Peripheral
Example	`>AT+ORGL` `+RDY`
Note	This command is used to revert all parameters to the original factory installed setting. This command will be followed with a system reset immediately.
Related	AT+RST

AT+PWMC=

Description	Enable or disable PWM port
Syntax	AT+PWMC=<portId>,<enable>
Parameters	<portId>: • I2c port id <enable>: • 0 – disable • 1 – enable
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	All PWM ports are disabled
Persistent	No
Example 1	`>AT+PWMC=0,1  (Enable PWM port 0)` `OK`
Example 2	`>AT+PWMC=1,0  (Disable PWM port 1)` `OK`
Note	This command is used to enable or disable PWM function. PWM port must be enabled before AT+PWM command can be used.
Related	AT+PWMC?, AT+PWM, AT+PWM?

AT+PWMC?

Description	Query PWM port status
Syntax	AT+PWMC?
Response	+PWMC:<portId>,<enable
Parameters	<portId>: • pwm port id <enable>: • 0 – disable • 1 – enable
Role	Central or Peripheral

Example 1	`>AT+PWMC?`
	`+PWMC:0,0 (PWM port 0 is disabled)`
	`OK`

Note	This command is used to query the PWM port status.
Related	AT+PWMC, AT+PWM, AT+PWM?

AT+PWM=

Description	Issue PWM command
Syntax	AT+PWM=<portId>,<period>,<dutyCycle>
Parameters	<portId>: • pwm port id <period>: • 32 bits unsigned integer • unit in Hz <dutyCycle> • 32 bits unsigned integer • unit in percentage • 0xFFFFFFFF = 100% • 0x000000000 = 0%
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	No
Persistent	No
Example 1	`>AT+PWM=0,40000,0x7FFFFFFF (40KHz period @ 50% duty cycle)` `OK`
Example 2	`>AT+PWM=0,1000000,0x19999998 (1MHz period @ 10% duty cycle)` `OK`
Note	This command is used to set PWM on a port. Please note: it is invalid to change the PWM setting using this command prior to enable the PWM port.
Related	AT+PWMC, AT+PWMC?, AT+PWM?

AT+PWM?

Description	Get PWM port status command
Syntax	AT+PWM?
Response	+PWM:<portId>,<period>,<dutyCycle>[;...]
Parmameters	<portId> • pwm port id <period>: • 32 bits unsigned integer • unit in Hz <dutyCycle> • 32 bits unsigned integer • unit in percentage • 0xFFFFFFFF = 100%

	• 0x000000000 = 0%
Role	Central or Peripheral
Example 1	>AT+PWM? +PWM:0,40000,0x80000000;… (40KHz, 50% duty cycle) OK
Note	This command is used to query the current PWM settings
Related	AT+PWMC, AT+PWMC?, AT+PWM

AT+RBUF?

Description	Query receive data free buffer size
Syntax	AT+RBUF?
Response	+RBUF:<freeBytes>
Parameter	<freeBytes>: • free buffer size in bytes
Role	Central and Peripheral
Example	>AT+RBUF? +RBUF:270 OK
Note	This command will return the free receive buffer size in the system that AT+LSND can accept without overflow. It can be used to send user data when UART flow control is not possible.
Related	AT+LSND

AT+RST

Description	Perform a system reset
Syntax	AT+RST
Parameters	none
Role	Central and Peripheral
Response	+RDY
Role	Central or Peripheral
Example	>AT+RST +RDY
Note	This command is used to reset the hardware.
Related	AT+ORGL

AT+SLP[=]

Description	Set system sleep state
Syntax	AT+SLP[=<sleepMode>]
Parameters	<sleepMode>: • 0 – Wakeup (only valid for Remote CLI) • 1 – Power Down • 2 - Standby • Default: 2 (standby) if not specified
Response	OK or ERR=<errCode>
Role	Peripheral

Example 1	`>AT+SLP`
	`OK`
	`(Wakeup Pin Triggered)`
	`+WKUP`

Example 2	`>AT+SLP=1`
	`(No "OK" coming back)`
	`(Wakeup Pin Triggered)`
	`+RDY`

Example 3	`>AT+SLP`
	`ERR=15`
	`(Error because wakeup pin not assigned)`

Note	This command will set the system sleep states – Wakeup, Standby Sleep or Power Down Sleep. The difference between Standby Sleep and Power Down Sleep is that Standby Sleep will allow connection or advertising be maintained while sleeping, but Power Down state will shut down everything. Both sleep states can be waked up by triggering the wakeup pin. But only Standby Sleep can be waked up by issuing command through remote CLI (`AT+SLP=0`). While in any of the sleep state, UART interface will be shutdown. Waking up from Power down state will go through a hardware reset.

AT+SYSP=

Description	Set System I/O Pin Assignment
Syntax	AT+SYSP=<connStatusPinId>,<cliDataModeSwitchPinId>, <deepSleepWakeupPinId>
Parameters	<connStatusPinId>: • Pin ID for connection status • Range: 0 – 14, and 0xFF • 0 to 14 represents DIO0 to DIO14, 255 represents unassigned <cliDataModeSwitchPinId>: • Pin ID for cli/data mode switch • Range: 0 – 14, and 0xFF

	• 0 to 14 represents DIO0 to DIO14, 255 represents unassigned <deepSleepWakeupPinId>: • Pin ID for deep sleep wakeup • Range: 0 – 14, and 0xFF • 0 to 14 represents DIO0 to DIO14, 255 represents unassigned
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	Check firmware release notes
Persistent	Yes
Example	`>AT+SYSP=14,6,255` `OK`
Note	This command is used to assign three system I/O pins for connection status (output), cli/data mode switch (input), and deep sleep wakeup (input). Setting the pin id to zero can null these pins assignment. Reassignment can fail if the pin has been allocated for something else - e.g. GPIO. In this case, user must chooses another pin or remove the pin in the other subsystem first.

AT+SYSP?

Description	Query System I/O Pin Assignment
Syntax	AT+SYSP?
Response	+SYSP:<connStatusPinId>,<cliDataModeSwitchPinId>, <deepSleepWakeupPinId>
Parameters	<connStatusPinId>: • Pin ID for connection status • Range: 0 – 14, and 0xFF • 0 to 14 represents DIO0 to DIO14, 255 represents unassigned <cliDataModeSwitchPinId>: • Pin ID for cli/data mode switch • Range: 0 – 14, and 0xFF • 0 to 14 represents DIO0 to DIO14, 255 represents unassigned <deepSleepWakeupPinId>: • Pin ID for deep sleep wakeup • Range: 0 – 14, and 0xFF • 0 to 14 represents DIO0 to DIO14, 255 represents unassigned
Role	Central or Peripheral

Example 1	`>AT+SYSP?` *`(verbose off)`*
	`+SYSP:14,6,255`
	`OK`

Example 2	`>AT+SYSP?` *`(verbose on)`*
	`+SYSP:14,6,255`
	`ConnStatus Pin [14] OK`

	CLI/DATA Switch Pin [6] OK Wakeup Pin [255] N/A OK
Note	This command is used to query the system I/O pin assignment settings.

AT+UART=

Description	Set UART connection parameters
Syntax	AT+UART=<baud>,<dataLen>,<stopBit>,<parity>,<flowCtrl>
Parameters	<baud>: • Baud rate • Supported: 9600, 19200, 38400, 57600, 115200 <dataLen>: • Data length in bits • Range:5 to 8 bits <stopBit>: • Stop bits • 1 or 2 bits <parity>: • Parity • 0 – None, 1 – even, 2 – odd <flowCtrl>: • Flow Control • 0 – off, 1 – on
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	Baud Rate = 115200 Data Length = 8 bits Stop Bit = 1 bit Parity = None Flow Control = Off [These value may be changed in different firmware application. Please check the firmware release note]
Persistent	Yes
Example	>AT+UART=115200,8,1,0,1 OK
Note	This command is used to set UART connection parameters. The new settings will not be effective after hardware reset.

AT+UART?

Description	Query UART connection parameters

Syntax	AT+UART?
Response	+UART:<baud>,<dataLen>,<stopBit>,<parity>,<flowCtrl>
Parameters	<baud>: • Baud rate • Supported: 9600, 19200, 38400, 57600, 115200 <dataLen>: • Data length in bits • Range:5 to 8 bits <stopBit>: • Stop bits • 1 or 2 bits <parity>: • Parity • 0 – None, 1 – even, 2 – odd <flowCtrl>: • Flow Control • 0 – off, 1 – on

Example 1	`>AT+UART?  (verbose off)`
	`+UART:112500,8,1,0,0`
	`OK`

Example 2	`>AT+UART?  (verbose on)`
	`+UART:115200,8,1,0,0`
	`Baud [115200]`
	`Data [8bit]`
	`Stop [1bit]`
	`Parity [None]`
	`FlowCtrl [No]`
	`OK`

Note	This command is used to query the UART connection parameters.

AT+VB=

Description	Set verbose mode
Syntax	AT+VB=<mode>
Parameters	<mode>: 0 - verbose off 1 - verbose on
Response	OK or ERR=<errCode>
Role	Central or Peripheral
Default	Verbose On
Persistent	No
Example	`>AT+VB=0  (verbose off)` `OK`

Note	This command is used to set the verbose mode. Verbose mode applies at system level. When verbose mode is set, all query commands will be responded with both machine-optimized and human readable info. If cleared, only the machine-optimized will be responsed.

AT+VB?

Description	Query verbose mode
Syntax	AT+VB?
Response	+VB:<mode>
Parameters	<mode>: 0 - verbose off 1 - verbose on
Role	Central or Peripheral

Example 1	`>AT+VB?  (verbose off)`
	`+VB:0`
	`OK`

Example 2	`>AT+VB?  (verbose on)`
	`+VB:1`
	`Verbose On`
	`OK`

Note	This command is used to query the verbose mode.

AT+VS?

Description	Query firmware and hardware version
Syntax	AT+VS?
Response	+VS:<hardVerStr>,<softVerStr>,<serialNo>,<capabilityStr>
Parameters	<hardVerStr> • hardware version string <softVerStr> • software version string < serialNo > • serial number < capabilityStr > • capability string • used for OAD firmware upgrade.
Role	Central or Peripheral

Example 1	`>AT+VS?`
	`+VS:2.0,1.0.436,123456,TIEM`
	`OK`

Note	This command is used to query the system hardware and firmware versions

Description of Notifications

This section describes the list of notifications. All notifications are unsolicited. Below is the list of notifications.

Notification	Central Role Device	Peripheral Role Device
+CMDM	Yes	Yes
+DATM	Yes	Yes
+GIO	Yes	Yes
+LCLI	No	Yes
+LCONN	Yes	Yes
+LSND	Yes	Yes
+LRCV	Yes	Yes
+LSCAN	Yes	No
+RCLI	No	Yes
+RDY	Yes	Yes
+WKUP	No	Yes

+CMDM

Description	Notify when changed to command line mode
Syntax	+CMDM
Parameters	None
Role	Central or Peripheral
Example	(Toggle CLI/Data Mode Switch Pin) +CMDM (Toggle CLI/Data Mode Switch Pin again) +DATM (Toggle CLI/Data Mode Switch Pin again) +CMDM >AT OK
Note	This notification is sent whenever the system changes to command line mode by toggling the CLI/Data Mode Switch control pin. User should wait for this notification before sending next command.

+DATM

Description	Notify when changed to data mode
Syntax	+DATM
Parameters	None
Role	Central or Peripheral
Example	(Toggle CLI/Data Mode Switch Pin) +DATM (Toggle CLI/Data Mode Switch Pin again) +CMDM (Toggle CLI/Data Mode Switch Pin again) +DATM >AT OK
Note	This notification is sent whenever the system changes to data mode by toggling the CLI/Data Mode Switch control pin. User should wait for this notification before starting to send user data to BT radio.

+GIO:

Description	Notify connection status changed
Syntax	+GIO:<pinID>,<pinState>
Parameters	<pinID> • Pin Index • Range 0 – 14 represents DIO0 to DIO14 <pinState> • Pin state • 0 – low, 1 - high
Role	Central or Peripheral
Example	>AT+GIOC=0xFFFF OK >AT+GIOC? +GIOC:0x1083 OK >AT+GOIOD=0xFFFF,0,1,1 OK >AT+GIOD? +GIOD:0x00001083,0x00000000,0x00001083,0x00001083,0x00000000 OK (Toggle Pin #1) +GIO=1,1 +GIO=1,0 (Toggle Pin #12) +GIO=12,1 +GIO=12,0

Note	This notification is sent whenever a GPIO interrupt is detected.
Related	AT+GIOC, AT+GIOD

+LCONN:

Description	Notify connection status changed
Syntax	+LCONN:<connStatus>,<connIdx>,<btAddr>
Parameters	<connStatus>: connection status • 0 - Idle • 1 - Connecting • 2 - Connected • 3 – Connected Ready (Central Role only) • 4 – Disconnecting (Transitional, Central Role only) • 5 – Disconnected (Transitional) • 5 – Disconnected By Remote (Transitional) • 6 - Connecting Timeout (Transitional) • 7 – Connecting Error (Transitional) <connIdx> • connection index <btAddr> • remote Bluetooth address
Role	Central or Peripheral
Example 1	>AT+LCONN OK +LCONN:1,0,01:02:03:04:05:06 +LCONN:2,0,01:02:03:04:05:06
Example 2	>AT+LSTOP=1 (Central Role) OK +LCONN:4,1,01:02:03:04:05:06 +LCONN:5,1,01:02:03:04:05:06 >AT+LCON? +LCON:0,0,0 OK
Note	This notification is sent whenever there is a connection status changed. It works in conjunction with AT+LCONA, AT+LCONI, AT+LCONN and AT+LSTOP. All transitional status above will go to Idle state immediately.
Related	AT+LCONN, AT+LSTOP, AT+LSTOPC

+LSND:

Description	Notify when user data to radio are all accepted

Syntax	+LSND:<connIdx>,<byteSent>
Parameters	< connIdx >: • connection index <byteSent>: • user bytes sent • Range 1 to 16383
Role	Central or Peripheral
Example	>AT+LSND=0,10 OK (User sends "0123456789") +LSND:0,10 >AT OK
Note	This notification is sent whenever all user bytes are completely accepted after AT+LSND command.
Related	AT+LSND, +LRCV

+LRCV:

Description	Notify when user data are received from BT radio
Syntax	+LRCV:<connIdx>,<byteRcv>
Parameters	< connIdx >: • connection index <byteRcv>: • user bytes received • Range 1 to 16383
Role	Central or Peripheral
Example	>AT+DTP=1 OK (sending some from radio ...) +LRCV:0,10 30 31 32 33 34 35 36 37 38 39 (Compact Hex) >AT OK
Note	This notification is sent whenever a packet of data is received from BT radio.
Related	AT+LSND, AT+DTP

+LSCAN:

Description	Notify when scanning is completed
Syntax	+LSCAN:<state>,<NumOfDevScanned>

Parameters	<state>: • Scanning state • 2 – completed, 3 – interrupted • 0 – Idle and 1 – Scanning are reserved for AT+SCAN? <numOfDevScanned>: • Number of devices scanned • Range 0 to maxDev set with AT+SCAN
Role	Central

Example 1	`>AT+LSCAN=3,5`
	`OK`
	`(after 3s of scanning ...)`
	`+LSCAN:2,3 (3 devices scanned)`
	`>AT`
	`OK`

Example 2	`>AT+LSCAN=3,2`
	`OK`
	`(after 2s of scanning ...)`
	`+LSCAN:2,2 (2 devices scanned)`
	`>AT`
	`OK`

Note	This notification is sent whenever scanning of device is completed or interrupted. Please note that the device scan list will not show with this notification. User is required to issue AT+LSCAN? in order to retrieve the device scan list.
Related	AT+LSCAN, AT+LSTOP, AT+LSTOPC

+RCLI

Description	Notify when entering remote parsing mode
Syntax	+RCLI
Parameters	None
Role	Peripheral
Example 1	`+RCLI`
Note	This notification is sent to the local UART interface when the slave device accepts a reroute request of its AT command parser to the Bluetooth radio interface. When the parser is rerouted to the radio interface, the UART interface will lose it CLI control temporarily in command mode. Any data sent through the UART will be forwarded to radio interface as in data mode.
Related	+LCLI

+LCLI

Description	Notify when leaving remote parsing mode
Syntax	+LCLI
Parameters	None
Role	Peripheral
Example 1	`+LCLI`
Note	This notification is sent to the local UART interface when the slave device reroutes its AT command parser back to its local UART interface.
Related	+RCLI

+RDY

Description	Notify after system reset
Syntax	+RDY
Parameters	None
Role	Central or Peripheral

Example 1	`(Power up)`
	`+RDY`
	`>AT`
	`OK`

Example 2	`AT+RST`
	`+RDY`
	`>AT`
	`OK`

Example 3	`AT+SLP=1`
	`(System goes into power down state)`
	`(Toggle wake up pin)`
	`+RDY`
	`>AT`
	`OK`

Note	This notification is sent whenever the system is reset after a physical power up, sending `AT+RST` or sending `AT+SLP=1` command.
Related	AT+SLP, AT+RST, AT+ORGL

+WKUP

Description	Notify when system just wakes up from standby

Syntax	+WKUP
Parameters	None
Role	Peripheral
Example	AT+SLP=0 OK (System goes to standby state) (Toggle the wake up pin) +WKUP >AT OK
Note	This notification is sent whenever the system wakes up from standby state.
Related	AT+SLP