BLUETOOTH SERIAL PORT ADAPTER AT COMMANDS GATT

Document Revision
Document number: 15925256
Release: May 13, 2014 11:37
Document version: 2

1 Abstract

This document describes AT commands for the Generic Attribute Profile(GATT) that is part of the Bluetooth Low Energy/Smart technology.

2 Table of Content

3 Related documents

[1] Bluetooth Low Energy Serial Port Adapter - Getting Started: Describes the basic concepts for the Bluetooth Serial Port Adapter. This document is written for the classical Bluetooth Serial Port Adapter but the concepts are the same also for the Bluetooth Low Energy solutions.

[2] Bluetooth Low Energy Serial Port Adapter - Getting Started: Describes how to use the connectBlue *Bluetooth Low Energy Serial Port Adapter* modules.

4 GATT mode

The GATT AT commands are enabled when Low Energy is enabled. Indications for service changed might be received any time if the remote device supports the Serial port service. Before GATT AT commands can be used an ACL connection must exist to the remote device. Use the extensions for ADCP, and ADCC described below to connect and disconnect. Note that an ACL connection and Serial Port connection can not be made in parallel to the same device.

5 AT commands reference

5.1 Low Energy ACL connection

5.1.1 ACL connection to a remote device (AT*ADCP)

AT Command	Description
AT*ADCP=<bd_addr>, 17, 0, 0<CR>	Make an ACL connection to a remote device supporting Low Energy. The connection shall be used in data mode. When the host connects to a service on a remote device it implicitly registers to receive the "Connection Data Mode Closed" event.

Responses	Description
<CR><LF>*ADCP:<acl_connection_handle><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the device to connect to.
acl_connection_handle	integer	The connection handle identifies the connection. The connection handle is used when closing the connection. Note that this handle might collide with the Serial Connection handle used when connecting to another device with classic Serial port or Low Energy Serial Port service. When using multi-point the host should keep track of whether the connection was made as a pure ACL connection or as a Serial port connection together with the handle.

Model	Constraint
cB-OEMSPA3xx, cB-OBS410, cB-OLS425/426	Not supported
cB-OBS421, cB-RBE221s, cB-RBS421	Supported for Client-Central and Server-Central configuration.The command is not supported for the Client-Peripheral and Server-Peripheral configuration

5.1.2 Close ACL connection (AT*ADCC)

AT Command	Description
AT*ADCC=<connection_handle>,1<CR>	Close an existing ACL connection

Responses	Description
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value

connection_handle	integer	The connection handle identifies the connection.

Model	Constraint
cB-OLS425/426, cB-OEMSPA3xx, cB-OBS410	Not supported.
cB-OBS421, cB-RBE221s, cB-RBS421	Supported in Low Energy mode.

5.1.3 ACL connection closed (*ADCCO)

Event	Description
*ADCCO:<connection_handle>,<reason>,1<CR><LF>	A connection to a remote device has been disconnected.

Event Parameters	Type	Value
connection_handle	integer	Identifies the connection.
reason	enumerator	0: Disconnected by command 1: Disconnected by link loss 255: Reason unknown

Model	Constraint
cB-OLS425/426	Not supported.
cB-OBS421, cB-RBE221s, cB-RBS421	Supported in Low Energy mode.

5.2 GATT AT commands

5.2.1 Discover All Primary Services(AT*GATTDP)

AT Command	Description
AT*GATTDP=<handle><CR>	Discovers all primary services on the remote device.

Responses	Description
<CR><LF>*GATTDP: <handle>,<start>,<end>,<uuid>	This response is sent for every service found
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
start	integer	Start handle of the service
end	integer	End handle of the service
uuid	hex string	UUID of the service. This can be either 16 bit or 128 bit e.g. 2A00

5.2.2 Discover Primary Services by UUID(AT*GATTDPU)

AT Command	Description
AT*GATTDPU=<handle>,<uuid>CR>	Discovers all primary services by UUID on the remote device.

Responses	Description
<CR><LF>*GATTDPU: <handle>,<start>,<end>	This response is sent for every service found
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
start	integer	Start handle of the service
end	integer	End handle of the service
uuid	hex string	UUID of the service. This can be either 16 bit or 128 bit e.g. 2A00

5.2.3 Find Included Services(AT*GATTFI)

AT Command	Description
AT*GATTFI=<handle>,<start_handle>,<end_handle><CR>	Find all included services on the remote device between start handle and end handle.

Responses	Description
<CR><LF>*GATTFI: <handle>,<attr_handle>,<start>,<end>,<uuid>	This response is sent for every service found
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
attr_handle	integer	Attribute handle of the included service
start	integer	Start handle of the service
end	integer	End handle of the service
uuid	hex string	UUID of the service. This can be either 16 bit or 128 bit e.g. 2A00

5.2.4 Discover All Characteristics of Service(AT*GATTDCS)

AT Command	Description
AT*GATTDCS=<handle>,<start>,<end>CR>	Discovers all characteristics of a service.

Responses	Description
<CR><LF>*GATTDCS: <handle>,<attr_handle>,<properties>,<value_handle>,<uuid>	This response is sent for every characteristic found
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
start	integer	Start handle of the service
end	integer	End handle of the service
attr_handle	integer	Attribute handle of the characteristic
properties	hex byte	Bit mask describing the properties of the characteristic Bit 0: Broadcast Bit 1: Readable Bit 2: Writable with no response Bit 3: Writable Bit 4: Notify Bit 5: Indicate Bit 6: Authenticated signed write Bit 7: Extended property available
value_handle	integer	Attribute handle of the characteristic value
uuid	hex string	UUID of the service. This can be either 16 bit or 128 bit e.g. 2A00

5.2.5 Discover All Characteristic Descriptors(AT*GATTDCD)

AT Command	Description
AT*GATTDCD=<handle>,<val_handle>,<service_end_handle>CR>	Discovers all descriptors of a characteristic

Responses	Description
<CR><LF>*GATTDCD: <handle>,<char_handle>,<desc_handle>,<uuid>	This response is sent for every descriptor found
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
char_handle	integer	Handle for the characteristic
service_end_handle	integer	End handle of the service which the characteristic belongs to
desc_handle	integer	Handle of the descriptor
value_handle	integer	Handle of the characteristic value
uuid	hex string	UUID of the service. This can be either 16 bit or 128 bit e.g. 2A00

5.2.6 Read Characteristic(AT*GATTR)

AT Command	Description
AT*GATTR=<handle>,<val_handle>CR>	Reads the characteristic. It will read all bytes in the characteristic.

Responses	Description
<CR><LF>*GATTR: <handle>,<val_handle>,<hex_data>	This response is sent if read data is found. If all bytes do not fit on one response line the data will continue as a new response but with the same value handle.
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
hex_data	hex string	The read data as hex string e.g. 070809AABBCC

5.2.7 Read Characteristic by UUID(AT*GATTRU)

AT Command	Description
AT*GATTRU=<handle>,<start>,<end>,<uuid>CR>	Reads all characteristics by UUID. It will read all bytes in each characteristic.

Responses	Description
<CR><LF>*GATTRU: <handle>,<val_handle>,<hex_data>	This response is sent if read data is found. If all bytes do not fit on one response line the data will continue as a new response but with the same value handle.
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
start	integer	Start handle
end	integer	End handle
uuid	hex string	UUID of the service. This can be either 16 bit or 128 bit e.g. 2A00
hex_data	hex string	The read data as hex string e.g. 070809AABBCC

5.2.8 Read Multiple Characteristics(AT*GATTRM)

AT Command	Description
AT*GATTRM=<handle>,<attrHandleList><CR>	Reads all characteristics in the attrHandleList. Note that it will not be possible to distinguish individual values in the response. Thus the length of all attributes except the last one must be known beforehand.

Responses	Description
<CR><LF>*GATTRM: <handle>,<val_handle>,<hex_data>	This response is sent if read data is found. If all bytes do not fit on one response line the data will continue as a new response but with the same value handle.
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the first characteristic value that is read
attrHandleList	hex string	The attribute handles as a list e.g. 00010002
hex_data	hex string	The read data as hex string e.g. 070809AABBCC

5.2.9 Write Characteristic(AT*GATTW)

AT Command	Description
AT*GATTW=<handle>,<val_handle>,<hex_data><CR>	Write characteristic. Note that this should only be used for small data transfers i.e. < 20 bytes. For larger writes use AT*GATTWL.

Responses	Description

<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
hex_data	hex string	The data as hex string e.g. 070809AABBCC

5.2.10 Write Characteristic Configuration(AT*GATTWC)

AT Command	Description
AT*GATTWC=<handle>,<val_handle>,<config><CR>	Write characteristic configuration.

Responses	Description
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
config	integer	The client configuration as integer: 0: None 1: Enable notifications 2: Enable indications 3: Enable notification and indications The server configuration as integer: 0: None 1: Enable broadcasts

5.2.11 Write Characteristic with No Response(AT*GATTWN)

AT Command	Description
AT*GATTWN=<handle>,<val_handle>,<hex_data><CR>	Write characteristic with no response. Note that this should only be used for small data transfers i.e. < 20 bytes. For larger writes use AT*GATTWL.

Responses	Description
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
hex_data	hex string	The data as hex string e.g. 070809AABBCC

5.2.12 Write Long Characteristic(AT*GATTWL)

AT Command	Description
AT*GATTWL=<handle>,<val_handle>,<hex_data><reliable>,<flag>,<offset><CR>	Write long characteristic. This is used to write a characteristic longer than 20 bytes or when a reliable write is required.

Responses	Description
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
hex_data	hex string	The data as hex string e.g. 070809AABBCC
reliable	integer	Send the data as reliable or not. If using reliable the returned data will be checked. 0: Not reliable 1: Reliable
flag	integer	Optional flag which is used when sending several packets or when data is canceled. If sending several packets all but the last packet should set the flag to more data. The last data packet should set the flag to final. 0: Final data, (default) 1: More data 2: Cancel data writing

offset	integer	Optional offset of the data to write(0 is default). Is used when several packets need to be sent to write a complete data value.

5.2.13 Notification(*GATTN)

AT Event	Description
*GATTN:<handle>,<val_handle>,<hex_data><CR>	This event is received when the remote side sends a notification

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
hex_data	hex string	The data as hex string e.g. 070809AABBCC

5.2.14 Indication(*GATTI)

AT Event	Description
*GATTI:<handle>,<val_handle>,<hex_data><CR>	This event is received when the remote side sends an indication.

Parameters	Type	Value
handle	integer	Handle of the connected device
val_handle	integer	Handle of the characteristic value
hex_data	hex string	The data as hex string e.g. 070809AABBCC