



Serial Port Adapter™

2nd Generation

AT Commands

connectBlue

Serial Port Adapter™
2nd Generation

AT Commands

Copyright © 2003 connectBlue AB.

The contents of this document can be changed by connectBlue AB without prior notice and do not constitute any binding undertakings from connectBlue AB. connectBlue AB is not responsible under any circumstances for direct, indirect, unexpected damage or consequent damage that is caused by this document.

All rights reserved.

Release: 0309

Document version: 2.0

Document number: cBProduct-0211-04 (4)

Printed in Sweden.

Trademarks

The BLUETOOTH trademarks are owned by their proprietor and used by connectBlue under license.

Third party brands and names are the property of their respective owners.

Contents

1. Introduction	8
1.1 Related Documents	8
2. Data Mode and AT Mode	9
3. Baud Rate	10
4. Syntax	11
4.1 Command Line Format	11
4.2 Data Types	11
String	12
Integer	12
Enumerator	12
Bd_Addr.....	12
5. Generation 2 vs. Generation 1	13
6. Writing Bluetooth Applications	14
6.1 Configuration	14
Bluetooth Settings.....	14
6.2 Searching for Other Bluetooth Devices	14
6.3 Searching for Services.....	14
6.4 Creating Serial Connections and Sending Data	15
How to Select What Profile to Use	15
Client or Server.....	15
Wireless Multidrop™	15
Connection Establishment – Server	15
Connection Establishment – Client.....	15
6.5 Master/Slave Handling.....	16
6.6 Bluetooth Security.....	17
Security Modes	17
Bonding and Pairing	17
7. AT Commands Over Bluetooth	18
8. AT Commands Reference	19
8.1 Standard AT Commands	19
Command “AT”	19
Command “AT*”	19
Command “ATZ”	19
Command “AT&F”	19
Command “ATS2”	20
Command “ATE”	20
8.2 GAP Commands.....	20
Read_Discoverability_Mode (AT*AGDM?)	20
Write_Discoverability_Mode (AT*AGDM=).....	21

	Read_Connectability_Mode (AT*AGCM?).....	21
	Write_Connectability_Mode (AT*AGCM=).....	22
	Read_Pairing_Mode (AT*AGPM?).....	22
	Write_Pairing_Mode (AT*AGPM=).....	22
	Read_Security_Mode (AT*AGSM?).....	23
	Write_Security_Mode (AT*AGSM=).....	23
	Name_Discovery (AT*AGND=).....	24
	Device_Discovery (AT*AGDD=).....	24
	Inquiry (AT*AGI=).....	25
	Bond (AT*AGB=).....	26
	Un_Bond (AT*AGUB=).....	26
	Read_Bonded_Devices (AT*AGBD?).....	26
	Read_Fixed_PIN (AT*AGFP?).....	27
	Write_Fixed_PIN (AT*AGFP=).....	27
	Read_Local_Name (AT*AGLN?).....	28
	Write_Local_Name (AT*AGLN=).....	28
	Read_Local_COD (AT*AGLC?).....	29
	Write_Local_COD (AT*AGLC=).....	29
	Get_MasterSlaveRole (AT*AGGMSR=).....	30
	Change_MasterSlaveRole (AT*AGCMSR=).....	30
	Read_MasterSlaveRole_Policy (AT*AGMSP?).....	31
	Write_MasterSlaveRole_Policy (AT*AGMSP=).....	31
	Get_RSSI (AT*AGRSS=).....	32
8.3	Service Search Commands.....	32
	Service_Search (AT*ARSS=).....	32
8.4	Data Mode Commands.....	34
	Data_Mode (AT*ADDM).....	34
	Connect_To_Serial_Service_Data_Mode (AT*ADCP=).....	34
	Close_Serial_Connection_Data_Mode (AT*ADCC=).....	35
	Read_Default_Client_Profile (AT*ADDCP?).....	35
	Write_Default_Client_Profile (AT*ADDCP=).....	36
	Read_Default_Server_Profile (AT*ADDSP?).....	36
	Write_Default_Server_Profile (AT*ADDSP=).....	37
	Read_Max_No_Of_Remote_Peers (AT*ADM RP?).....	37
	Read_No_Of_Remote_Peers (AT*ADN RP?).....	38
	Write_No_Of_Remote_Peers (AT*ADN RP=).....	38
	Read_Default_Remote_Peer (AT*ADRDRP=).....	39
	Write_Default_Remote_Peer (AT*ADWDRP=).....	39
	Read_Inactivity_Tick (AT*ADIT?).....	40
	Write_Inactivity_Tick (AT*ADIT=).....	40
	Read_Wireless_Multidrop_Configuration (AT*ADWM?).....	41
	Write_Wireless_Multidrop_Configuration (AT*ADWM=).....	42
8.5	Informational Commands.....	42
	Read_Local_BD_ADDR (AT*AILBA?).....	42
	Read_Local_Version_Information (AT*AILVI?).....	42
8.6	Miscellaneous Commands.....	43
	Read_RS232_Settings (AT*AMRS?).....	43
	Write_RS232_Settings (AT*AMRS=).....	44
	Read_Serial_Interface_Type (AT*AMSIT?).....	44
	Write_Serial_Interface_Type (AT*AMSIT=).....	45
	Read_Favorites (AT*ACF?).....	45
	Add_Change_Favorite (AT*ACACF=).....	46
	Delete_Favorite (AT*ACDF=).....	46
	Read_Allow_Configuration_Over_Bluetooth (AT*ACCB?).....	46
	Write_Allow_Configuration_Over_Bluetooth (AT*ACCB=).....	47
	Write_Controller_Power_Save_Mode (AT*AMPM=).....	47
	Write_Max_Output_Power (AT*AMMP=).....	48
	Write_Esc_Sequence_Timing (AT*AMET=).....	48
8.7	Events.....	49

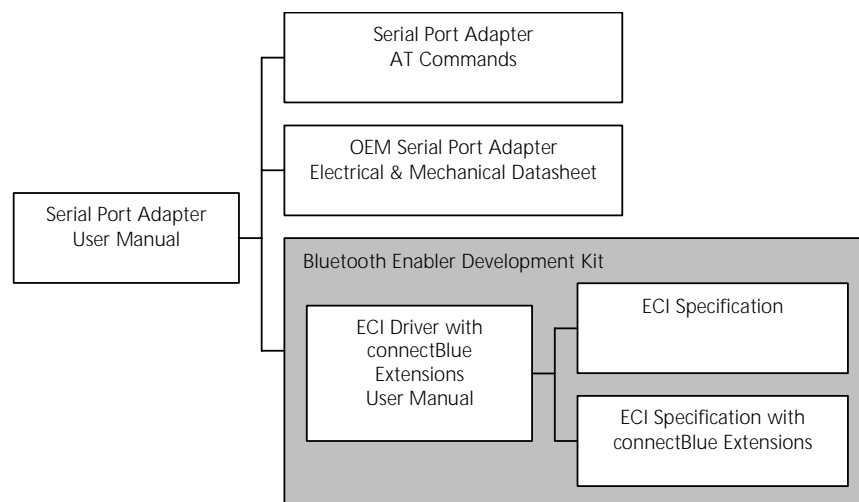
Serial_Connection_Data_Mode_Closed (*ADCCO)	49
---	----

Chapter 1

Introduction

1.1 Related Documents

- The **Serial Port Adapter User Manual** contains information on how to use the Serial Port Adapter. Study this document before moving on to the others.
- The **Serial Port Adapter AT Commands** document, this document, contains a description of the AT commands supported in the Serial Port Adapter. It also contains information on how to use the AT commands to create Bluetooth applications.
- The **OEM Serial Port Adapter Electrical & Mechanical Datasheet** contains important information about the OEM Serial Port Adapter. Read this document if you are using the OEM Serial Port Adapter.
- The **Bluetooth Enabler Development Kit** is required when using the ECI functionality embedded in the Serial Port Adapter.
 - The **ECI Driver with connectBlue Extensions User Manual** contains a description of the ECI functionality available in the Serial Port Adapter. It also contains information on how to use the ECI Driver to create Bluetooth applications. If you are going to use the ECI protocol in your application, study this document.
 - The **ECI Specification** and the **ECI Specification with connectBlue Extensions** contain detailed descriptions of the ECI protocol. These documents should be used as a reference when reading the ECI Driver with connectBlue Extensions User Manual.



Chapter 2

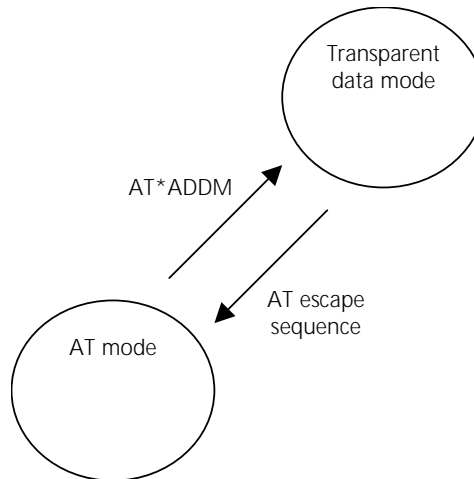
Data Mode and AT Mode

The Serial Port Adapter can be in two different modes AT mode and data mode. The Serial Port Adapter starts up in data mode and can be requested to move to AT mode by sending an escape sequence. The default escape sequence consists of three consecutive forward slash characters '/'. The escape sequence character can be changed using the ATS2 command. The default escape character is restored by pressing the restore-button on the Serial Port Adapter. For more information consult the manual of your Serial Port Adapter.

The following criteria must be met for the Serial Port Adapter to interpret the sequence as a valid escape sequence:

- Before the escape sequence there must be silence for 1 second. This time can be changed using the AT*AMET command.
- After the escape sequence there must be silence for 1 second. This time can be changed using the AT*AMET command.
- The entire escape sequence must be sent within 200 ms.

To move from AT mode to data mode, use the "AT*ADDMM" command.



Chapter 3

Baud Rate

The Serial Port Adapter does not support auto baud rate. The baud rate is set using the "Write_RS232_Settings" command.

The default RS232 settings are 57600 baud, 8 data bits, no parity, 1 stop bit, hardware flow control. Consult the manual for your Serial Port Adapter to find out how to restore the default RS232 settings.

Chapter 4

Syntax

4.1 Command Line Format

Each command line sent from the DTE to the DCE is made up of a prefix, body and terminator. As prefix for the Serial Port Adapter AT commands, only "AT" (ASCII 065, 084) and "at" (ASCII 097, 116) can be used. There is no distinction between upper and lower case characters. The body is a string of characters in the range ASCII 032-255. Control characters other than <CR> (carriage return; ASCII 013) and <BS> (back space; ASCII 008) in a command line are ignored. The terminator is <CR>.

Commands denoted with a "*" character are extended AT commands, i.e. Serial Port Adapter specific AT commands.

S-registers are not used and not supported, except for the S2 register.

Multiple commands in the same command line are not supported. Each command has to be terminated by a <CR> before a new command can be sent. A command must not be larger than 300 characters.

A command can either be:

- Read commands without parameters:
AT<command>?<CR>
- Read and write commands with parameters:
AT<command><CR>
- Write commands without parameters:
AT<command>=<parameter1>, parameter2>, ...<parameterN><CR>

Responses are sent back to the host and can be any of the following:

- Successful final message:
<CR><LF>OK<CR><LF>
- Successful intermediate/final message with parameters follows an OK message in some commands. In these cases the OK message works as a confirm message only.
<CR><LF><result_response>:<parameter1>, parameter2>, ...<parameterN>
- Error message:
<CR><LF>ERROR<CR><LF>

4.2 Data Types

The definition of each command specifies the data types used for values associated with the command.

There are four data types:

- String
- Integer
- Enumerator

- Bluetooth Device Address

String

A string shall consist of a sequence of displayable characters from the ISO 8859-1 (8-bit ASCII) character set, except for characters “\” and “ ” and characters below 32 (space). A string constant shall be delimited by two double-quote (“ ”) characters, e.g. “Donald Duck”. If the double-quote character (“ ”) is to be used within a string, e.g. “My friend “Bono” is a singer”, they have to be represented as “\22”. If the back-slash character (“\”) is to be used within a string constant, it has to be represented as “\5C”. An empty string is represented by two adjacent delimiters, “ ”.

Integer

An integer value consists of a sequence of characters all in the range {0..9}. Numeric constants are expressed in decimal format only.

Enumerator

An enumerator value is actually an integer, where all its possible values are specified in each case. Only the defined values are accepted for the command in question.

Bd_Addr

This type is used to represent the Bluetooth Device Address. The type is composed of six fields, each representing a hexadecimal value using two characters. For example, the Bluetooth Device Address 0x112233AABBCC is represented as 112233AABBCC or 112233aabbcc. Note that the Bluetooth Device Address is **not** delimited with by two double-quote (“ ”) characters.

Chapter 5

Generation 2 vs. Generation 1

If you are migrating from using the first generation of Serial Port Adapters from connectBlue to generation 2 the following AT commands/events have been changed, added or removed:

AT Command/Event	Comment
Read_Local_COD	New parameter type and values in the response.
Write_Local_COD	New parameter type and values in the command.
Get_MasterSlaveRole	Command name has changed from AT*AGMSR? to AT*AGGMSR=. Command has a new parameter.
Change_MasterSlaveRole	Command name has changed from AT*AGMSR= to AT*AGCMSR=. Command has a new parameter.
Service_Search	Command has a new parameter.
Connect_To_Serial_Profile	Old command, AT*ASCP=, has been replaced by the command Connect_To_Serial_Service_Data_Mode, AT*ADCP=.
Close_Serial_Connection	Old command, AT*ASCC=, has been replaced by the command Close_Serial_Connection_Data_Mode, AT*ADCC=.
Read_Max_No_Of_Remote_Peers	New command.
Read_No_Of_Remote_Peers	New command.
Write_No_Of_Remote_Peers	New command.
Read_Default_Remote_Peer	Command name has changed from AT*ADDRP? to AT*ADRDRP=. Parameters have changed.
Write_Default_Remote_Peer	Command name has changed from AT*ADDRP= to AT*ADWRP=. Parameters have changed.
Read_Wireless_Multidrop_Configuration	New command.
Write_Wireless_Multidrop_Configuration	New command.
Read_RS232_Settings	Parameters have changed.
Write_RS232_Settings	Parameters have changed.
Serial_Connection_Closed event	Old event, *ASCCO, has been replaced by the new event Serial_Connection_Data_Mode_Closed, *ADCCO.
Write_Controller_Power_Save_Mode	New command.
Inquiry	New command.
Write_Max_Output_Power	New command.
Write_Esc_Sequence_Timing	New command.
Get_RSSI	New command.

Chapter 6

Writing Bluetooth Applications

This chapter gives some guidelines on how to write Bluetooth applications using AT commands.

6.1 Configuration

There are several request packets that can be used to configure the Serial Port Adapter. Many of these request packets take a boolean parameter called `<store_in_startup_database>`. If this parameter is set to 1 the setting will be applied immediately and also when the Serial Port Adapter starts up in the next power cycle. If this parameter is set to 0 the setting will be applied immediately but it will not be applied when the Serial Port Adapter starts up in the next power cycle.

Bluetooth Settings

A Bluetooth device can be in several different operation modes. The operation mode determines whether or not a device can be connected to and whether or not other devices performing searches can discover a device. Use the `"Write_Discoverability_Mode"` and `"Write_Connectability_Mode"` commands to set the operation mode.

All Bluetooth devices have a user-friendly name. Use the `"Write_Local_Name"` command to set the local device name.

All Bluetooth devices have a 'class-of-device' indicator that can be discovered by other devices when they are performing searches. Use the `"Write_Local_COD"` command to set the 'class-of-device'.

6.2 Searching for Other Bluetooth Devices

Two commands are available to search for other devices:

- The `"Device_Discovery"` command returns the Bluetooth device address, the class of device and the name of all the devices in the vicinity that are in discoverable mode.
- The `"Inquiry"` command returns the Bluetooth device address and the class of device of all the devices in the vicinity that are in discoverable mode.

The `"Inquiry"` command is faster than the `"Device_Discovery"` command.

6.3 Searching for Services

It is possible to search for services on remote devices. A service search is performed using the `"Service_Search"` command.

6.4 Creating Serial Connections and Sending Data

Serial connections are Bluetooth connections based on the Serial Port Profile, the Dial-up Networking Profile and the LAN Access Profile.

How to Select What Profile to Use

It is important that the same profile is used on both devices wanting to communicate. If the remote device is not a Serial Port Adapter, refer to the documentation of that device to determine what profile it uses or to find out how to select what profile to use.

Client or Server

First decide if your device is supposed to act as a client (initiator of connections), a server (acceptor of connections) or both.

Wireless Multidrop™

The Wireless Multidrop™ feature allows the Serial Port Adapter to simultaneously communicate with several devices even when it is in data mode. If Wireless Multidrop™ is used all data sent to the Serial Port Adapter via the serial interface will be forwarded to all connected devices. All data received from any of the connected devices will be forwarded on the serial interface in the order that the data was received.

For more information please consult the Serial Port Adapter user manual.

Connection Establishment – Server

In AT mode, use the "Write_Default_Server_Profile" command to select what profile to use when acting as a server.

If you want to use the Wireless Multidrop™ feature and allow several devices to simultaneously connect to your device you must enable Wireless Multidrop™ using the "Write_Wireless_Multidrop_Configuration" command. In addition, the Serial Port Adapter must be configured to perform a master/slave switch every time a device connects to it. This is done using the "Write_MasterSlaveRole_Policy" command.

After configuring the Serial Port Adapter for server operation, move to data mode.

Connection Establishment – Client

For clients there are two methods that can be used to create connections to a remote device:

1. Establish the connection in AT mode using the "Connect_To_Serial_Service_Data_Mode" command. Then move to data mode and utilize the connection as a transparent data mode connection. It is possible to utilize the Wireless Multidrop™ feature and connect to several devices. Note that Wireless Multidrop™ must be enabled to be able to connect to more than one device.
2. Configure the Serial Port Adapter in AT mode then move to data mode.
 - a. Select the number of devices to connect to using the "Write_No_Of_Remote_Peers" command. Select 1 for point-to-point

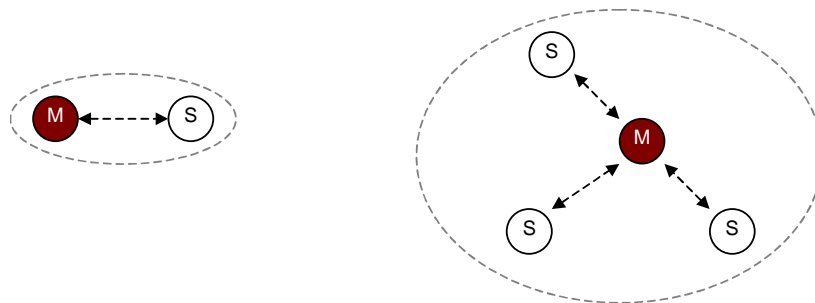
operation and more than 1 for Wireless Multidrop™ operation. Note that Wireless Multidrop™ must be enabled to be able to use a value higher than 1. The maximum number of remote peers can be determined by calling the "Read_Max_No_Of_Remote_Peers" command.

- b. Select what device(s) to communicate with using the "Write_Default_Remote_Peer" command.
- c. Then use the "Write_Default_Client_Profile" command to select what profile to use when connecting to the selected default remote peer.
- d. Finally move to data mode and send data. The Serial Port Adapter will connect to the selected remote peer(s) using the selected profile according to the connect scheme setting (connect on data, always connected or connect on external signal).

Specify zero number of remote peers using the "Write_No_Of_Remote_Peers" command if you do not want to act as a client.

6.5 Master/Slave Handling

When a device communicates with several other devices it is called the master. The other devices are called slaves.



A master can:

- Setup a connection to another device.
- Perform searches.
- Accept connections from other devices.

A slave cannot:

- Connect to another device.
- Perform searches.
- Accept connections from other devices.

There are a few settings and parameters that affect how the roles master and slave are assigned to the devices communicating.

First, it is possible for a device to request to become the master when another device connects to it. This setting is called the master/slave role policy. Use the "Write_MasterSlaveRole_Policy" command to set the policy to either:

- 0, the device will request to become the master every time another device tries to connect.
- 1, the device will become the slave every time another device tries to connect.

Second, when connecting to another device using the "Connect_To_Serial_Service" command the parameter <must_be_master> indicates whether or not the connecting device allows the server to become the master of the connection.

NOTE: If `<must_be_master>` is 1 and the server has set master/slave role policy to 0 (become master) the connection will *not* be established.

If you do not specifically need to become the master the `<must_be_master>` parameter should be set to 0 to allow the server to select which role to take.

6.6 Bluetooth Security

Bluetooth has support for security. The Bluetooth security is based on authentication during connection establishment and encryption of sent and received data.

Security Modes

A Bluetooth device can be in two different security modes, security enabled (authentication and encryption turned on) and security disabled (authentication and encryption turned off). If at least one of the two devices wanting to communicate has security enabled, security will be used. Use the `"Write_Security_Mode"` command to set the security mode for the device.

Bonding and Pairing

To be able to communicate if security has been enabled, bonding has to be performed. The bonding procedure creates a link key, valid between two devices, which is used during the authentication procedure. Once bonded the two devices can establish connections with each other using security enabled.

During bonding both devices must be in pairable mode (able to accept bonding). Use the `"Write_Pairing_Mode"` command to set the device(s) in pairable mode.

During bonding a pin code is used. Use the `"Write_Fixed_PIN"` command to set the PIN code to use.

To perform bonding use one of the following methods:

- On one device use the `"Bond"` command.
- Create a connection. If authentication or encryption is turned on, on either device, bonding will be performed automatically.

It is possible for the Serial Port Adapter to store link keys for several devices at the same time. Use the `"Read_Bonded_Devices"` command to get a list of the currently bonded devices.

It is also possible to remove a device from the list of bonded devices. To do this, use the `"Un_Bond"` command.

Chapter 7

AT Commands Over Bluetooth

It is possible to configure and control a remote Serial Port Adapter via Bluetooth using AT commands.

Criteria for using AT commands over Bluetooth:

- A Bluetooth connection to the remote device (the device to configure) must exist. The Bluetooth connection can e.g. be established using another Serial Port Adapter, a device with a PC-card/compact flash or a device with embedded Bluetooth support.
- The remote Serial Port Adapter must have turned on the support for configuration over Bluetooth, for more information see the “Write_Allow_Configuration_Over_Bluetooth” command.
- Once the connection has been established, the escape sequence must be sent over the Bluetooth connection to the remote Serial Port Adapter to move it into AT mode.
 - The same escape sequence rules apply to AT over Bluetooth as for AT over cable, for more information see the section about Data Mode and AT Mode.
 - Note that the same escape sequence is used for cable and Bluetooth. Therefore, if you are using two Serial Port Adapters one of the Serial Port Adapters must have its escape sequence changed using the ATS2 command. This is to keep the first Serial Port Adapter to enter AT mode instead of the remote Serial Port Adapter.

Chapter 8

AT Commands Reference

8.1 Standard AT Commands

Command "AT"

AT Command	Description
AT<CR>	Attention command determining the presence of a DCE, i.e. the Serial Port Adapter.

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

Command "AT*"

AT Command	Description
AT*<CR>	Lists the supported AT commands.

Responses	Description
<CR><LF><command>	This response is sent to the host for every supported command.
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Command "ATZ"

AT Command	Description
ATZ<CR>	Does nothing.

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

Command "AT&F"

AT Command	Description
------------	-------------

AT&F<CR>	Does nothing.
----------	---------------

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

Command "ATS2"

AT Command	Description
ATS2=<esc_char><CR>	Changes the escape character to esc_char.

Command Parameters	Type	Description
esc_char	integer	esc_char is the ASCII value of the new escape character. E.g. 47 equals '/

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

Command "ATE"

AT Command	Description
ATE<echo><CR>	Set whether or not the Serial Port Adapter shall echo incoming characters.

Command Parameters	Type	Description
echo	integer	0: Incoming characters will not be echoed. 1: Incoming characters will be echoed.

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

8.2 GAP Commands

Read_Discoverability_Mode (AT*AGDM?)

AT Command	Description
AT*AGDM?<CR>	This command reads the current GAP discoverability mode.

Responses	Description
<CR><LF>*AGDM:<discoverability_mode><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
discoverability_mode	enumerator	1: GAP non-discoverable mode 2: GAP limited discoverable mode 3: GAP general discoverable mode

Write_Discoverability_Mode (AT*AGDM=)

AT Command	Description
AT*AGDM=<discoverability_mode>,<store_in_startup_database><CR>	This command writes the current GAP discoverability mode.

Command Parameters	Type	Value
discoverability_mode	enumerator	1: GAP non-discoverable mode 2: GAP limited discoverable mode 3: GAP general discoverable mode
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Connectability_Mode (AT*AGCM?)

AT Command	Description
AT*AGCM?<CR>	This command reads the current GAP connectability mode.

Responses	Description
<CR><LF>*AGCM:<connectability_mode><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Result Parameters	Type	Value
connectability_mode	enumerator	1: GAP non-connectable mode 2: GAP connectable mode

Write_Connectability_Mode (AT*AGCM=)

AT Command	Description
AT*AGCM=<connectability_mode>,<store_in_startup_database><CR>	Writes the GAP connectability mode.

Command Parameters	Type	Value
connectability_mode	enumerator	1: GAP non-connectable mode 2: GAP connectable mode
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Pairing_Mode (AT*AGPM?)

AT Command	Description
AT*AGPM?<CR>	Reads the pairing mode.

Responses	Description
<CR><LF>*AGPM:<pairing_mode><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Result Parameters	Type	Value
pairing_mode	enumerator	1: GAP non-pairing mode 2: GAP pairing mode

Write_Pairing_Mode (AT*AGPM=)

AT Command	Description
AT*AGPM=<pairing_mode>,<store_in_startup_database><CR>	Writes the GAP pairing mode.

Command Parameters	Type	Value
pairing_mode	enumerator	1: GAP non-pairable mode. 2: GAP pairable mode.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Security_Mode (AT*AGSM?)

AT Command	Description
AT*AGSM?<CR>	Reads the GAP security mode.

Responses	Description
<CR><LF>*AGSM:<security_mode><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Result Parameters	Type	Value
security_mode	enumerator	1: Link level authentication and encryption disabled (GAP security mode 1 with encryption disabled). 2: Link level authentication and encryption enabled (GAP security mode 3 with encryption enabled).

Write_Security_Mode (AT*AGSM=)

AT Command	Description
AT*AGSM=<security_mode>,<store_in_startup_database><CR>	Writes the GAP security mode.

Command Parameters	Type	Value
--------------------	------	-------

security_mode	enumerator	1: Link level authentication and encryption disabled (GAP security mode 1 with encryption disabled). 2: Link level authentication and encryption enabled (GAP security mode 3 with encryption enabled).
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Name_Discovery (AT*AGND=)

AT Command	Description
AT*AGND=<bd_addr><CR>	Retrieves the device name of a remote device given its Bluetooth device address.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the device from which to retrieve the name.

Responses	Description
<CR><LF>*AGND:<device_name><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
device_name	string	Null terminated string of maximum 240 characters (8-bit ASCII).

Device_Discovery (AT*AGDD=)

AT Command	Description
AT*AGDD=<inquiry_Type>,<inquiry_length><CR>	Performs device discovery.

Command Parameters	Type	Description
inquiry_type	enumerator	1: Limited inquiry 2: General inquiry

inquiry_length	integer	Maximum amount of time specified before the inquiry is halted. Range: 8-48 Time = inquiry_length*1.28 seconds Range in seconds: 10.24-61.44
----------------	---------	--

Responses	Description
<CR><LF>*AGDD: <no_of_devices><CR><LF>OK<CR><LF>	Successful response
*AGDDE:<bd_addr>, <cod>, <device_name_valid>, <device_name><CR><LF>	This response is sent for every found device.
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
no_of_devices	integer	Value in range {0..255}. Number of devices discovered during the inquiry procedure.
bd_addr	Bd_Addr	Bluetooth device address of a discovered device.
cod	integer	See Read_Local_COD command.
device_name_valid	enumerator	1: device_Name parameter valid. 0: Device was discovered, but its name could not be retrieved. device_name is parameter invalid and should be ignored.
device_name	string	Name of discovered device. ASCII represented string of maximum 240 bytes.

Inquiry (AT*AGI=)

AT Command	Description
AT*AGI=<inquiry_Type>,<inquiry_length>,<max_no_of_devices_to_find><CR>	Performs an inquiry procedure to find any discoverable devices in the vicinity.

Command Parameters	Type	Description
inquiry_type	enumerator	1: Limited inquiry 2: General inquiry
inquiry_length	integer	Maximum amount of time specified before the inquiry is halted. Range: 1-48 Time = inquiry_length*1.28 seconds Range in seconds: 1.28-61.44
max_no_of_devices_to_find	integer	0: No limitation on the number of devices to find. 1-255: Maximum number of devices to find.

Responses	Description
<CR><LF>*AGI:<bd_addr>,<cod>	This response is sent for every found device.
<CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of a found device.
cod	integer	See Read_Local_COD command.

Bond (AT*AGB=)

AT Command	Description
AT*AGB=<bd_addr><CR>	Performs a GAP bond procedure with another Bluetooth device. During the bonding procedure the fixed PIN code is used, see the Write_Fixed_PIN and Read_Fixed_PIN commands. Note that to be able to perform bonding the remote device must be in pairable mode.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the device to bond with.

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

Un_Bond (AT*AGUB=)

AT Command	Description
AT*AGUB=<bd_addr><CR>	This command un-bonds a previously bonded device.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the device subject to un-bond.

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

Read_Bonded_Devices (AT*AGBD?)

AT Command	Description
AT*AGBD?<CR>	Read the bonded devices.

Responses	Description
<CR><LF>*AGBD:<no_of_devices><CR><LF>OK<CR><LF>	Successful response
*AGBDE:<bd_addr>,<device_name_valid>,<device_name><CR><LF>	This response is sent for every found device.
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
no_of_bonded_devices	integer	Number of bonded devices.
bd_addr	Bd_Addr	Bluetooth device address of the device from which to retrieve the name.
device_name_valid	enumerator	0: device_Name parameter valid. 1: Device is bonded but its name is not available. Device_Name is parameter invalid.
device_name	string	Name of discovered device. Null terminated ASCII represented string.

Read_Fixed_PIN (AT*AGFP?)

AT Command	Description
AT*AGFP?<CR>	Read the fixed PIN code used by the Serial Port Adapter during bond and pairing.

Responses	Description
<CR><LF>*AGFP:<pin_code><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
pin_code	string	The PIN code is a string of maximum 16 characters in range {0..9}, e.g. "0145673829029827".

Write_Fixed_PIN (AT*AGFP=)

AT Command	Description
AT*AGFP=<pin_code>,<store_in_startup_database><CR>	Writes the fixed PIN code used by the Serial Port Adapter during bond.

Command Parameters	Type	Value
pin_code	string	The PIN code is a string of maximum 16 characters in range {0..9}, e.g. "0145673829029827".

store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.
---------------------------	------------	--

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

Read_Local_Name (AT*AGLN?)

AT Command	Description
AT*AGLN?<CR>	Reads the local Bluetooth device name.

Responses	Description
<CR><LF>*AGLN:<device_name><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
device_name	string	Max 240 characters.

Write_Local_Name (AT*AGLN=)

AT Command	Description
AT*AGLN=<device_name>, <store_in_startup_database><CR>	Writes the local Bluetooth device name.

Command Parameters	Type	Value
device_name	string	Max 240 characters.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Local_COD (AT*AGLC?)

AT Command	Description
AT*AGLC?<CR>	Reads the Local Class Of Device code.

Responses	Description
<CR><LF>*AGLC:<cod><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
cod	integer	<p>Valid values for this parameter are specified in the Bluetooth Assigned Numbers Document, www.bluetooth.com. The parameter has been divided into three segments, a service class segment, a major device class segment and a minor device class segment (bits 2-7).</p> <p>Extract from the Bluetooth Assigned Numbers Document:</p> <p>Service class (bit mask, bits 13-23): Bit 16: Positioning (Location identification) Bit 17: Networking (LAN, Ad hoc, etc) Bit 18: Rendering (Printing, Speaker, etc) Bit 19: Capturing (Scanner, Microphone, etc) Bit 20: Object Transfer (v-Inbox, v-Folder, etc) Bit 21: Audio (Speaker, Microphone, Headset service, etc) Bit 22: Telephony (Cordless telephony, Modem, Headset service) Bit 23: Information (WEB-server, WAP-server, etc)</p> <p>Major device class (number, bits 12-8): 00000: Miscellaneous 00001: Computer (desktop, notebook, PDA, etc) 00010: Phone (cellular, cordless, modem, etc) 00011: LAN/Network Access point 00100: Audio/Video (headset, speaker, stereo, video display, VCR) 00101: Peripheral (mouse, joystick, keyboards) 00110: Imaging (printing, scanner, camera, etc) 11111: Uncategorized, specific device code not specified</p> <p>For the minor device class field please refer to [1].</p>

Write_Local_COD (AT*AGLC=)

AT Command	Description
AT*AGLC=<cod>, <store_in_startup_database><CR>	Writes the Local Class Of Device code.

Command Parameters	Type	Value
cod	integer	See Read_Local_COD command.
store_in_startup_database	enumerator	0: The setting will only be valid for the current

Command Parameters	Type	Value
cod	integer	See Read_Local_COD command.
		power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Get_MasterSlaveRole (AT*AGGMSR=)

AT Command	Description
AT*AGGMSR=<bd_addr><CR>	Read the local master-slave role. Returns the role of the Serial Port Adapter, master or slave, for the connection between the Serial Port Adapter and the remote device identified by the 'bd_addr' parameter.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Identifies a device that the Serial Port Adapter is currently communicating with.

Responses	Description
<CR><LF>*AGGMSR:<role><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
role	enumerator	0: Slave 1: Master

Change_MasterSlaveRole (AT*AGCMSR=)

AT Command	Description
AT*AGCMSR=<bd_addr>,<role><CR>	Changes the master-slave role. Changes the role of the Serial Port Adapter, master or slave, for the connection between the Serial Port Adapter and the remote device identified by the 'bd_addr' parameter. Note that the remote device does not have to accept the master/slave switch.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Identifies a device that the Serial Port Adapter is currently communicating with. The role will be changed on the connection to this device.
role	enumerator	0: Slave 1: Master

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

Read_MasterSlaveRole_Policy (AT*AGMSP?)

AT Command	Description
AT*AGMSP?<CR>	Reads the role policy of the device.

Responses	Description
<CR><LF>*AGMSP:<role_policy><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
role_policy	enumerator	0: Always attempt to become master on incoming connections. 1: Always let the connecting device select master/slave role on incoming connections.

Write_MasterSlaveRole_Policy (AT*AGMSP=)

AT Command	Description
AT*AGMSP=<role_policy>,<store_in_startup_database><CR>	Writes the role policy of the device.

Command Parameters	Type	Value
role_policy	enumerator	0: Always attempt to become master on incoming connections. 1: Always let the connecting device select master/slave role on incoming connections.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Get_RSSI (AT*AGRSS=)

AT Command	Description
AT*AGRSS=<bd_addr><CR>	This request returns the current received signal strength, RSSI, for the connection between the ECI Controller and the remote device identified by the 'bd_addr' parameter. Note that this command can only be used on an existing connection that has been established using the Connect_To_Serial_Service_Data_Mode command and only while still in AT mode.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Identifies a device that the Serial Port Adapter is currently communicating with.

Responses	Description
<CR><LF>*AGRSS:<rssi><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
rssi	integer	< 128: The received signal strength is 128-RSSI dB below the optimal signal range. 128: The received signal strength is within the optimal signal range. >128: The received signal strength is RSSI-128 dB above the optimal signal range.

8.3 Service Search Commands

Service_Search (AT*ARSS=)

AT Command	Description
AT*ARSS=<bd_addr>,<role_and_profile>,<max_nbr_of_results><CR>	Search for services on a remote device.

Command	Type	Value
---------	------	-------

Parameters		
bd_addr	Bd_Addr	Bluetooth device address of the device on which to search for services.
role_and_profile	enumerator	0: DevB role, Serial Port Profile 1: Gateway role, Dial-Up Networking Profile 2: LAN Access Point role, LAN Access Profile
max_nbr_of_results	integer	The maximum number of services to be collected.

Responses	Description
<CR><LF>*ARSS:<matching_service_records><CR><LF>OK<LR><CR>	Successful response
*ARSRSP:<bd_addr>, <rfcomm_server_chan>, <service_name_valid>, <service_name><CR><LF>	This response is the result of a device B serial port profile service search.
*ARSRDUN:<bd_addr>, <rfcomm_server_chan>, <service_name_valid>, <service_name>, <audio_feedback_support_valid>, <audio_feedback_support><CR><LF>	This response is the result of a GW dial-up networking profile service search.
*ARSRLAN:<bd_addr>, <rfcomm_server_chan>, <service_name_valid>, <service_name>, <service_description_valid>, <service_description>, <service_availability_valid>, <service_availability>, <ip_subnet_valid>, <ip_subnet><CR><LF>	This response is the result of a LAP LAN access profile service search.
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the device on which to search for services.
matching_service_records	integer	The number of matching service records. Range 0 to 255.
rfcomm_server_chan	integer	RFCOMM server channel number on which this service can be found. It is used when connecting to a profile. Range 1 to 30.
service_name_valid	enumerator	0: The service_name parameter could not be retrieved from the remote device and the value is not valid. 1: The service_name parameter could be retrieved from the remote device and the value is valid.
service_name	string	Service name.
audio_feedback_support_valid	enumerator	0: The audio_feedback_support parameter could not be retrieved from the remote device and the value is not valid. 1: The audio_feedback_support parameter could be retrieved from the remote device and the value is valid.
audio_feedback_support	enumerator	0: No, device does not support audio feedback. 1: Yes, device supports audio feedback.
service_description_valid	enumerator	0: The service_description parameter could not be retrieved from the remote device and the value is not valid. 1: The service_description parameter could be retrieved from the remote device and the value is valid.

service_description	string	Manufacturer description of the services a product can provide.
service_availability_valid	enumerator	0: The service_availability parameter could not be retrieved from the remote device and the value is not valid. 1: The service_availability parameter could be retrieved from the remote device and the value is valid.
service_availability	integer	The service availability parameter available in the LAN Access Profile service record.
ip_subnet_valid	enumerator	0: The ip_subnet parameter could not be retrieved from the remote device and the value is not valid. 1: The ip_subnet parameter could be retrieved from the remote device and the value is valid.
ip_subnet	string	IP subnet mask to reach the device on the LAN.

8.4 Data Mode Commands

Data_Mode (AT*ADDM)

AT Command	Description
AT*ADDM<CR>	Request the Serial Port Adapter to move to data mode. After a successful response the Serial Port Adapter will leave AT-mode and enter data mode.

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

Connect_To_Serial_Service_Data_Mode (AT*ADCP=)

AT Command	Description
AT*ADCP=<bd_addr>, <role_and_profile>, <rfcomm_server_channel>, <must_be_master><CR>	Connect to a serial service enabled on a remote device. This request is used to connect to profiles based on the Serial Port Profile. 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 Serial_Connection_Data_Mode_Closed event.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the device to connect to.
role_and_profile	enumerator	0: DevB role, Serial Port Profile 1: Gateway role, Dial-Up Networking Profile 2: LAN Access Point role, LAN Access Profile

rfcomm_server_chan	enumerator	RFCOMM server channel number on which this service can be found. 0: Service search will be performed automatically and the first available <role_and_profile> will be connected. 1-30: RFCOMM server channel number. This server channel number can be retrieved using a service search operation. 31-255: Invalid values.
must_be_master	enumerator	0: The remote device may choose to become master or slave. 1: This device must be master of the new connection.

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

Response Parameters	Type	Value
connection_handle	integer	The connection handle identifies the connection. The connection handle is used when closing the connection.

Close_Serial_Connection_Data_Mode (AT*ADCC=)

AT Command	Description
AT*ADCC=<connection_handle><CR>	Close an existing data mode connection.

Command Parameters	Type	Value
connection_handle	integer	The connection handle identifies the connection.

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

Read_Default_Client_Profile (AT*ADDCP?)

AT Command	Description
AT*ADDCP?<CR>	This command reads the default client profile. The default client profile is the profile that the Serial Port Adapter uses when it establishes a connection, in data mode, to the default remote peer(s).

Responses	Description
<CR><LF>*ADDCP:<role_and_profile><CR><LF>OK<CR><LF>	Successful response.

<CR><LF>ERROR<CR><LF>	Error response.
-----------------------	-----------------

Response Parameters	Type	Value
role_and_profile	enumerator	0: DevB role, Serial Port Profile 1: Gateway role, Dial-Up Networking Profile 2: LAN Access Point role, LAN Access Profile 255: No profile

Write_Default_Client_Profile (AT*ADDCP=)

AT Command	Description
AT*ADDCP=<role_and_profile>,<store_in_startup_database><CR>	This command reads the default client profile. The default client profile is the profile that the Serial Port Adapter uses when it establishes a connection, in data mode, to the default remote peer(s).

Command Parameters	Type	Value
role_and_profile	enumerator	0: DevB role, Serial Port Profile 1: Gateway role, Dial-Up Networking Profile 2: LAN Access Point role, LAN Access Profile 255: No profile
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Default_Server_Profile (AT*ADDSP?)

AT Command	Description
AT*ADDSP?<CR>	This command reads the default server profile. The default server profile is the profile that other devices can connect to when the Serial Port Adapter is in data mode. The default server profile is activated when the Serial Port Adapter is moved to data mode if no connection exists. The default server profile is deactivated when the Serial Port Adapter is moved from data mode to AT mode.

Responses	Description
<CR><LF>*ADDSP:<role_and_profile><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
role_and_profile	enumerator	0: DevB role, Serial Port Profile 1: Gateway role, Dial-Up Networking Profile 2: LAN Access Point role, LAN Access Profile 255: No profile

Write_Default_Server_Profile (AT*ADDSP=)

AT Command	Description
AT*ADDSP=<role_and_profile>,<store_in_startup_database><CR>	This command writes the default server profile. The default server profile is the profile that other devices can connect to when the Serial Port Adapter is in data mode. The default server profile is activated when the Serial Port Adapter is moved to data mode if no connection exists. The default server profile is deactivated when the Serial Port Adapter is moved from data mode to packet mode.

Command Parameters	Type	Value
role_and_profile	enumerator	0: DevB role, Serial Port Profile 1: Gateway role, Dial-Up Networking Profile 2: LAN Access Point role, LAN Access Profile 255: No profile.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Max_No_Of_Remote_Peers (AT*ADM RP?)

AT Command	Description
AT*ADM RP?<CR>	For some Serial Port Adapters it is possible to have more than one remote peer defined. This command reads the maximum number of allowed remote peers.

Responses	Description
<CR><LF>*ADM RP:<max_no_of_remote_peers><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
max_no_of_remote_peers	integer	The maximum number of allowed remote peers.

Read_No_Of_Remote_Peers (AT*ADNRP?)

AT Command	Description
AT*ADNRP?<CR>	For some Serial Port Adapters it is possible to have more than one remote peer defined. This command reads the number of remote peers defined.

Responses	Description
<CR><LF>*ADNRP:<no_of_remote_peers><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
no_of_remote_peers	integer	The number of remote peers currently defined.

Write_No_Of_Remote_Peers (AT*ADNRP=)

AT Command	Description
AT*ADNRP=<no_of_remote_peers>,<store_in_startup_database><CR>	This command writes the number of remote peers defined. The number of remote peers may not be greater than the number returned by the Read_Max_No_Of_Remote_Peers command. After writing the number of remote peers the host must use the Write_Default_Remote_Peer command to write all the remote peers to the Serial Port Adapter. For more information see Read_No_Of_Remote_Peers.

Command Parameters	Type	Value
no_of_remote_peers	integer	The number of remote peers.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

<CR><LF>ERROR<CR><LF>	Error response.
-----------------------	-----------------

Read_Default_Remote_Peer (AT*ADRRP=)

AT Command	Description
AT*ADRRP=<peer_id ><CR>	This command reads the Bluetooth device address and device name of the selected default remote peer (peer id).

Command Parameters	Type	Value
peer_id	integer	The peer ID can be between 0 and the value written by the Write_No_Of_Peers command -1 or read by the Read_No_Of_Peers command -1.

Responses	Description
<CR><LF>*ADRRP:<bd_addr>,<connect_scheme>,<update_remote_peer_on_incoming>,<device_name><CR><LF>OK<CR><LF>	Successful response.
<CR><LF>ERROR<CR><LF>	Error response.

Response Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the default remote peer.
connect_scheme	integer	This parameter is a bit field. Bit 0 is the least significant bit. Each bit is defined as follows: Bit 0: Try to connect to default remote peer on data traffic. Bit 1: Always try to be connected to the default remote peer when in data mode. Bit 2: Try to connect to default remote peer on external signal. The external signal is implementation specific, e.g. some Serial Port Adapters might have a button. Bit 3-7: Reserved for future use.
update_remote_peer_on_incoming	enumerator	1: Every time a remote device connects to the selected DefaultServerProfile, update the remote peer device address to the device address of the connecting device. The new remote peer device address will be stored in the startup database. Only one of all the remote peers can use this feature. 0: Do not update the remote peer device address on incoming connections.
device_name	string	Maximum 240 characters.

Write_Default_Remote_Peer (AT*ADWDRP=)

AT Command	Description
AT*ADWDRP=<peer_id>,<bd_addr>,<connect_scheme>,<update_remote_peer_on_incoming>,<device_name>,<store_in_startup_database><CR>	This command writes the Bluetooth device address, connect scheme and device name of the currently selected default remote peer.

Command Parameters	Type	Value
peer_id	integer	The peer ID can be between 0 and the value written by the Write_No_Of_Peers command -1 or read by the Read_No_Of_Peers command -1.
bd_addr	Bd_Addr	Bluetooth device address of the default remote peer.
connect_scheme	integer	See Read_Default_Remote_Peer.
update_remote_peer_on_incoming	enumerator	See Read_Default_Remote_Peer.
device_name	string	See Read_Default_Remote_Peer.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Inactivity_Tick (AT*ADIT?)

AT Command	Description
AT*ADIT?<CR>	This command reads the current inactivity tick setting. If there is no data activity between two consecutive ticks the Serial Port Adapter will automatically disconnect the current data mode connection(s).

Responses	Description
<CR><LF>*ADIT:<inactivity_tick><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
inactivity_tick	integer	0: no inactivity tick. 1-255: the period (in minutes) of the inactivity timer.

Write_Inactivity_Tick (AT*ADIT=)

AT Command	Description
AT*ADIT=<inactivity_tick>,<store_in_startup_database><CR>	This command writes a new inactivity tick setting. If there is no data activity between two consecutive ticks the Serial Port Adapter will automatically disconnect the current connection(s).

Command	Type	Value
---------	------	-------

Parameters		
inactivity_tick	integer	0: no inactivity tick. 1-255: the period (in minutes) of the inactivity timer.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Wireless_Multidrop_Configuration (AT*ADWM?)

AT Command	Description
AT*ADWM?<CR>	<p>This request returns whether or not the Wireless Multidrop™ feature has been enabled.</p> <p>When the Wireless Multidrop™ has been enabled; all data sent to the Serial Port Adapter in data mode will be forwarded to all connected devices. Data received from a remote device will be forwarded to the host. If the Auto_Forward parameter is set to TRUE is will also forward all received data to all the other connected devices.</p> <p>Connections to remote devices can be established using three methods:</p> <ul style="list-style-type: none"> Let the Serial Port Adapter connect to the desired devices when it is in data mode. The host uses the request Write_No_Of_Remote_Peers and Write_Default_Remote_Peer to tell the Serial Port Adapter how many devices to connect to, which devices to connect to and when to connect to the defined devices. A server has been enabled using the request Write_Default_Server_Profile and one or several devices connect to this server. One or several connections are established in packet mode using the request Connect_To_Serial_Service_Data_Mode. After all desired connection have been setup by the host it moves to data mode. <p>When the Wireless Multidrop™ has been disabled; only one connection at a time can be established. A maximum of one remote peer can be defined.</p>

Responses	Description
<CR><LF>*ADWM:<enabled>, <auto_forward><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
enabled	enumerator	0: Wireless Multidrop™ disabled. 1: Wireless Multidrop™ enabled.

auto_forward	enumerator	0: Data received from a connected device will only be forwarded to the host. 1: Data received from a connected device will be forwarded to the host and all the other connected devices.
--------------	------------	---

Write_Wireless_Multidrop_Configuration (AT*ADWM=)

AT Command	Description
AT*ADWM=<enable>, <auto_forward>, <store_in_startup_database><CR>	See Read_Wireless_Multidrop_Configuration.

Command Parameters	Type	Value
enabled	enumerator	See Read_Wireless_Multidrop_Configuration.
auto_forward	enumerator	See Read_Wireless_Multidrop_Configuration.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

8.5 Informational Commands

Read_Local_BD_ADDR (AT*AILBA?)

AT Command	Description
AT*AILBA?<CR>	Reads the Bluetooth Device Address of the local device.

Responses	Description
<CR><LF>*AILBA:<bd_addr>,<CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
bd_addr	Bd_Addr	Local Bluetooth device address.

Read_Local_Version_Information (AT*AILVI?)

AT Command	Description
AT*AILVI?<CR>	This command reads the local version information to

	the Serial Port Adapter.
--	--------------------------

Responses	Description
<CR><LF>*ALVL:<eci_controller_man>, <eci_controller_sw_ver>, <host_stack_sw_ver>, <link_manager_sw_ver>, <bluetooth_hardware_manufacturer><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
eci_controller_man	string	Serial Port Adapter manufacturer.
eci_controller_sw_ver	string	Serial Port Adapter software version.
host_stack_sw_ver	string	Bluetooth host stack version.
link_manager_sw_ver	string	Bluetooth link manager version.
bluetooth_hardware_manufacturer	string	Bluetooth hardware manufacturer.

8.6 Miscellaneous Commands

Read_RS232_Settings (AT*AMRS?)

AT Command	Description
AT*AMRS?<CR>	This command reads current RS232 settings from the Serial Port Adapter.

Responses	Description
<CR><LF>*AMRS:<baud_rate>, <data_bits>, <stop_bits>, <parity>, <flow_control><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
baud_rate	enumerator	1: 300 2: 1200 3: 2400 4: 4800 5: 9600 6: 19200 7: 38400 8: 57600 9: 115200 10: 230400 11: 460800 12: 921600
data_bits	enumerator	1: 8 data bits 2: 7 data bits 3: 6 data bits 4: 5 data bits
stop_bits	enumerator	1: 1 stop bit 2: 2 stop bits

parity	enumerator	1: no parity 2: odd parity 3: even parity
flow_control	enumerator	1: CTS/RTS used for flow control 2: CTS/RTS not used.

Write_RS232_Settings (AT*AMRS=)

AT Command	Description
AT*AMRS=<baud_rate>, <data_bits>, <stop_bits>, <parity>, <flow_control>, <change_after_confirm>, <store_in_startup_database><CR>	This command applies new RS232 settings to the Serial Port Adapter. If 5, 6 or 7 data bits are selected the Serial Port Adapter will not change its RS232 settings until the next power cycle

Command Parameters	Type	Value
baud_rate	enumerator	See Read_RS232_Settings.
data_bits	enumerator	See Read_RS232_Settings.
stop_bits	enumerator	See Read_RS232_Settings.
parity	enumerator	See Read_RS232_Settings.
flow_control	enumerator	See Read_RS232_Settings.
change_after_confirm	enumerator	0: The Serial Port Adapter will not change RS232 settings until after the next power cycle. 1: The Serial Port Adapter will change RS232 settings after it has sent the OK to the ECI Host.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Serial_Interface_Type (AT*AMSIT?)

AT Command	Description
AT*AMSIT?<CR>	This command reads the serial interface type currently used.

Responses	Description
<CR><LF>*AMSIT:<serial_interface_type> <CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
serial_interface_type	enumerator	1: RS232 2: RS422 3: RS485 4-255: Reserved for future use.

Write_Serial_Interface_Type (AT*AMSIT=)

AT Command	Description
AT*AMSIT=<serial_interface_type>,<store_in_startup_database><CR>	This command writes the serial interface type currently used. Note that the Serial Port Adapter does not change serial interface type until the next time it is restarted.

Command Parameters	Type	Value
serial_interface_type	enumerator	See Read_Serial_Interface_Type.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Read_Favorites (AT*ACF?)

AT Command	Description
AT*ACF?<CR>	Read the stored favorites.

Responses	Description
<CR><LF>*ACF:<no_of_stored_favorites><CR><LF>OK<CR><LF>	Successful response
*ACFD:<bd_addr>,<favorite_name><CR><LF>	This response is sent for every found favorite device.
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
no_of_stored_favorites	integer	Number of stored favorite devices.
bd_addr	string	Bluetooth device address of the stored favorite.
favorite_name	string	Name of favorite.

Add_Change_Favorite (AT*ACACF=)

AT Command	Description
AT*ACACF=<bd_addr>,<favorite_name><CR>	Add or change a favorite. If the favorite, identified by its Bluetooth device address does not exist, it will be created.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the favorite.
favorite_name	string	Maximum 240 characters (8-bit ASCII).

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

Delete_Favorite (AT*ACDF=)

AT Command	Description
AT*ACDF=<bd_addr><CR>	Deletes a stored favorite.

Command Parameters	Type	Value
bd_addr	Bd_Addr	Bluetooth device address of the stored favorite.

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

Read_Allow_Configuration_Over_Bluetooth (AT*ACCB?)

AT Command	Description
AT*ACCB?<CR>	Determine whether or not configuration over Bluetooth is allowed. Configuration can either be performed using AT commands or using the ECI protocol.

Responses	Description
<CR><LF>*ACCB:<allow_configuration_over_bluetooth><CR><LF>OK<CR><LF>	Successful response
<CR><LF>ERROR<CR><LF>	Error message.

Response Parameters	Type	Value
allow_configuration_over_bluetooth	enumerator	0: Configuration over Bluetooth is not allowed. 1: Configuration over Bluetooth is allowed.

Write_Allow_Configuration_Over_Bluetooth (AT*ACCB=)

AT Command	Description
AT*ACCB= <allow_configuration_over_bluetooth>, <store_in_startup_database><CR>	See the Read_Allow_Configuration_Over_Bluetooth command.

Command Parameters	Type	Value
allow_configuration_over_bluetooth	enumerator	See the Read_Allow_Configuration_Over_Bluetooth command.
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Write_Controller_Power_Save_Mode (AT*AMPM=)

AT Command	Description
AT*AMPM=<mode>, <store_in_startup_database><CR>	This command writes a new power save mode setting to the Serial Port Adapter. Please study the Serial Port Adapter manual for details on how to utilize the power save modes.

Command Parameters	Type	Value
mode	enumerator	1: Allow only online mode. The Serial Port Adapter will never enter sleep or stop mode to save power. 2: Allow sleep mode (default value). The Serial Port Adapter will enter sleep mode when possible to save power. 3: Allow sleep mode and stop mode. The Serial Port Adapter will enter sleep or stop mode when possible to save power. 4 – 255: Power modes reserved for future use.

store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.
---------------------------	------------	--

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

Write_Max_Output_Power (AT*AMMP=)

AT Command	Description
AT*AMMP=<max_output_power>,<store_in_startup_database><CR>	Set the maximum output power to be used by the Serial Port Adapter when communicating. Note that not all Serial Port Adapter models support output power control.

Command Parameters	Type	Value
max_output_power	integer	255: Use the highest output power supported by the Serial Port Adapter as the maximum output power (default). 128-m: -m dBm (m<30) 128: 0 dBm 128+n: n dBm (n<30)
store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.

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

Write_Esc_Sequence_Timing (AT*AMET=)

AT Command	Description
AT*AMET=<min_before_time>,<min_after_time>,<store_in_startup_database><CR>	For an escape sequence to be valid, a period of no data activity is required before and after the escape sequence. This command sets the minimum time of no data activity required before and after the escape sequence.

Command Parameters	Type	Value
min_before_time	integer	50-500 ms.
min_after_time	integer	50-500 ms.

store_in_startup_database	enumerator	0: The setting will only be valid for the current power cycle. 1: The Serial Port Adapter will remember the setting between power cycles. The settings database in the Serial Port Adapter will be updated.
---------------------------	------------	--

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

8.7 Events

Serial_Connection_Data_Mode_Closed (*ADCCO)

Event	Description
*ADCCO:<connection_handle>,<reason><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