AT Command Set

Contents

1. Introduction

This document describes the protocol used to control and configure the following TDK Systems Bluetooth devices:

- blu^{2i} Module
- blu^{2i} PCMCIA Adaptor
- blu^{2i} RS-232 Adaptor
- blu^{2i} Universal RS-232 Adaptor
- go blue Activator

The protocol is similar to the industry standard Hayes AT protocol used in telephony modems due to the fact that both types of devices are connection oriented. Appropriate AT commands have been provided to make the blu^{2i} device perform the two core actions of a Bluetooth device, which is make/break connections and Inquiry. Many other AT commands are also provided to perform ancillary functions, such as, pairing, trusted devices database management and S Register maintenance.

Just like telephony modems, the blu^{2i} device powers up into an unconnected state and will only respond via the serial interface. In this state the blu^{2i} device will not even respond to Bluetooth Inquiries. Then, just like controlling a modem, the host can issue AT commands which map to various Bluetooth activities. The command set is extensive enough to allow a host to make connections which are authenticated and/or encrypted or not authenticated and/or encrypted or any combination of these.

The device has a serial interface which can be configured for baud rates from 1200 up to 921600, and an RF communications end point. The latter has a concept of connected and unconnected modes and the former will have a concept of command and data modes. This leads to the matrix of states shown below.

	RF Unconnected	**RF Connected**
Local Command Mode	**OK**	**OK**
Remote Command Mode	**ILLEGAL**	**OK**
Data Mode	**ILLEGAL**	**OK**

The combinations, ‘Data and RF Unconnected Mode’ and ‘Remote Command and RF Unconnected Mode’ do not make sense and will be ignored.

Navigation between these states is done using the AT commands which are described in detail in subsequent sections.

2. AT Command Set

2.1 Assumptions

The CSR (Cambridge Silicon Radio) BC2 chipset in blu^{2i} devices is memory resource limited. Therefore it is NOT proposed that there be full implementation of the AT protocol as seen in modems. The claim made for this device is that it will have a protocol ***similar*** to an AT modem. In fact, the protocol is similar enough so that existing source code written for modems, can be used with very little modification with a blu^{2i} device.

Therefore the following assumptions are made:-

1 All commands will be terminated by the carriage return character 0x0D, which is represented by the string <cr> in descriptions below and cannot be changed.

2 All responses from the blu^{2i} device will have carriage return and linefeed characters prepending and appending the response. These dual character sequences have the values 0x0D and 0x0A respectively and shall be represented by the string <cr,lf> and cannot be changed.

3 All Bluetooth addresses shall be represented by a fixed 12 digit hexadecimal string, case insensitive.

4 All Bluetooth Device Class codes shall be represented by a fixed 6 digit hexadecimal string, case insensitive.

5 All new Bluetooth specific commands shall be identified by the string +BT**x**, where **x** is generally a mnemonic of the intended functionality.

2.2 Commands

This section describes all available AT commands. Many commands require mandatory parameters and some take optional parameters. These parameters are either integer values, strings, Bluetooth addresses or device classes. The following convention is used when describing the various AT commands.

<bd_addr>	A 12 character Bluetooth address consisting of ASCII characters '0' to '9', 'A' to 'F' and 'a' to 'f'.
<devclass>	A 6 character Bluetooth device class consisting of ASCII characters '0' to '9', 'A' to 'F' and 'a' to 'f'.
n	A positive integer value.
m	An integer value which could be positive or negative, which can be entered as a decimal value or in hexadecimal if preceded by the '$' character. E.g. the value 1234 can also be entered as $4D2
<string>	A string delimited by double quotes. E.g. "Hello World". The " character MUST be supplied as delimiters.
<uuid>	A 4 character UUID number consisting of ASCII characters '0' to '9', 'A' to 'F' and 'a' to 'f'.

2.2.1 ^^^ {Enter Local Command Mode}

When in data and connected mode, the host can force the device into a command and connected mode so that AT Commands can be issued to the device. The character in this escape sequence is specified in the S2 register, so can be changed. In addition, the escape sequence guard time is specified by S Register 12. By default the guard time is set to 100 milliseconds. Please refer to Section 5: Dropping Connections for more related information.

In modems this escape sequence is usually "+++" and "^^^" is specified to avoid confusion when the module is providing access to a modem.

Response: <cr,lf>**OK**<cr,lf>

2.2.2 !!! {Enter Remote Command Mode}

When in data and connected mode, the host can force the remote device into a command and connected mode so that AT Commands can be issued to the device remotely. The escape sequence guard time is specified by S Register 12 and is the same as per the ^^^ escape sequence. By default the guard time is set to 100 milliseconds. The remote device issues ATO as normal to return to data mode.

For this command to be effective S Register 536 must be set to 1.

Response: <cr,lf>**OK**<cr,lf>

2.2.3 AT

Response: <cr,lf>**OK**<cr,lf>

2.2.4 ATA {Answer Call}

Accept an incoming connection, which is indicated by the unsolicited string <cr,lf>**RING 123456789012**<cr,lf> every second. **123456789012** is the Bluetooth address of the connecting device.

Response: <cr,lf>**CONNECT 123456789012**<cr,lf>

2.2.5 ATD<bd_addr>,<uuid> {Make Connection}

Make a connection to device with Bluetooth address <bd_addr> and profile <uuid>. The <uuid> is an optional parameter which specifies the UUID of the profile server to attach to, and if not supplied then the default UUID from S Register 101 is used. As this is a blu^{2i} device which utilises the RFCOMM layer as described in the Bluetooth specification, it necessarily implies that only profiles based on RFCOMM can be connected to.

Authentication and Encryption is as per S registers 500 and 501. The timeout is specified by S register 505.

Response: <cr,lf>**CONNECT 123456789012**<cr,lf>
Or <cr,lf>**NO CARRIER**<cr,lf>

Due to a known issue in the Bluetooth RFCOMM stack, it is not possible to make more that 65525 outgoing connections. Therefore if that number is exceeded, then the connection attempt will fail with the following response:-

Response: <cr,lf>**CALL LIMIT**
Or <cr,lf>**NO CARRIER**<cr,lf>

In that case, issuing an ATZ to reset the device will reset the count to 0 and more connections are possible.

The following RFCOMM based UUIDs are defined in the Bluetooth Specification:-

Profile Name	UUID
Serial Port	1101
LAN Access Using PPP	1102
Dialup Networking	1103
IrMC Sync	1104
OBEX Object Push	1105
OBEX File Transfer	1106
IrMCSyncCommand	1107
Headset	1108
Cordless Telephony	1109
Intercom	1110
Fax	1111
Audio Gateway	1112
WAP	1113
WAP_CLIENT	1114

2.2.6 ATDU<bd_addr>,<uuid> {Make Authenticated Connection}

Make an authenticated connection to device with Bluetooth address <bd_addr>. Encryption is disabled. <uuid> is an optional parameter and usage is as described for the ATD variant of the command.

Response: <cr,lf>**CONNECT 123456789012 A**<cr,lf>
Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.7 ATDY<bd_addr>,<uuid> {Make Encrypted Connection}

Make an encrypted connection to device with Bluetooth address <bd_addr>. Authentication is disabled. <uuid> is an optional parameter and usage is as described for the ATD variant of the command.

Response: <cr,lf>**CONNECT 123456789012 E**<cr,lf>
Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.8 ATDUY<bd_addr>,<uuid> {Make Authenticated/Encrypted Connection}

Make an authenticated and encrypted connection to device with Bluetooth address <bd_addr>. The order of U and Y is not significant. <uuid> is an optional parameter and usage is as described for the ATD variant of the command.

Response: <cr,lf>**CONNECT 123456789012 AE**<cr,lf>
Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.9 ATD<UY>L {Remake Connection}

Make a connection with the same device and service as that specified in the most recent ATD command. The <UY> modifiers are optional. An error will be returned if the ‘L’ modifier is specified AND a Bluetooth address.

If both ‘L’ and ‘R’ modifiers are specified then an error will be returned.

Response: <cr,lf>**CONNECT 123456789012 AE**<cr,lf>
Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.10 ATD<UY>R {Make Connection to peer specified in AT+BTR}

Make a connection with the device address specified in the most recent AT+BTR command. The service is as that specified in S Register 101. The <UY> modifiers are optional. An error will be returned if the ‘R’ modifier is specified AND a Bluetooth address.

If both ‘R’ and ‘L’ modifiers are specified then an error will be returned.

Response: <cr,lf>**CONNECT 123456789012 AE**<cr,lf>
Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.11 ATEn {Enable/Disable Echo}

This command enables or disables the echo of characters to the screen. A valid parameter value will be written to S Register 506.

E0 Disable echo.
E1 Enable echo.

All other values of n will generate an error.

Response: <cr,lf>**OK**<cr,lf>
Or
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.12 ATH {Drop Connection}

Drop an existing connection or reject an incoming connection indicated by unsolicited RING messages.

Response: <cr,lf>**NO CARRIER**<cr,lf>

2.2.13 ATIn {Information}

This will return information about the blu^{2i} device.

I0 Reports the product name/variant.
I1 The CSR firmware build number is returned.
I2 The TDK firmware build number is returned. For internal use only.
I3 The TDK firmware revision is returned.
I4 A 12 digit hexadecimal number corresponding to the Bluetooth address of the blu^{2i} device is returned.
I5 The manufacturer of this device is returned.
I6 The maximum size of trusted device database is returned.
I7 The manufacturer of the Bluetooth chipset is returned.
I8 The chipset format is returned.
I9 Returns 0 if not in a connect state and 1 if in a connect state.
I11 Returns the reason why a “NO CARRIER” resulted in the most recent attempt at making an outgoing connection. Where the response values are as follows:

0 = No prior connection
1 = Connection timeout
2 = Connection attempt cancelled
3 = Normal disconnection
4 = Peer device has refused connection
5 = Service profile <uuid> requested not available on remote device
6 = Connection has failed
32 = ATH was entered
33 = Incoming connection aborted because too many rings
34 = Unexpected incoming connection
35 = Invalid address
36 = DSR is not asserted
37 = Call limit of 65531 connections has been reached
38 = Pairing in progress
39 = No link key
40 = Invalid link key
255 = Unknown Reason

I12 Returns the last ERROR response number.
I13 The Sniff status is returned as follows:-

Response: <cr,lf>**a:b,c,d,e**<cr,lf>**OK**<cr,lf>

Where ‘a’ = 0 when not online and 1 when online and Sniff has been enabled, ‘b’ is the Sniff Attempt parameter, ‘c’ is the Sniff timeout parameter, ‘d’ is the minimum sniff interval and ‘e’ is the maximum sniff interval. All parameters ‘b’, ’c’, ’d’ and ‘e’ are given as Bluetooth slots which are 625 microseconds long converted from values of S Registers 561, 562, 563 and 564 respectively.

I42 State information is returned. Where the response values are as follows:

13 = NotOpen
14 = OpenIdle
15 = Ringing
16 = OnlineCommand
172 to 177 = waiting for connectable and/or discoverable where the lowest significant digit equates to the value stored in S Register 512 or 555.

Note when n=16, ATI9 will return 1.

For recognised values of n. All other values of n will generate an error.

Response: <cr,lf>**As Appropriate**<cr,lf>**OK**<cr,lf>
or
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.14 ATO {Enter Data Mode} (letter ‘o’)

Return to data mode. Assume in data mode after OK is received. Responds with an error if there is no Bluetooth connection.

Response: <cr,lf> **CONNECT 123456789012**<cr,lf>
or
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.15 ATSn=m {Set S Register}

There is a concept of registers which are used to store parameters, such as escape sequence character, inquiry delay time etc, as listed in detail below.

The value part ‘m’ can be entered as decimal or hexadecimal. A hexadecimal value is specified via a ‘$’ leading character. For example $1234 is a hexadecimal number.

When S register values are changed, the changes are **not** stored in non-volatile memory UNTIL the AT&W command is used. Note that AT&W does not affect S registers 520 to 525, 1000 to 1010 as they are updated in non-volatile memory when the command is received.

Register	Default	Range	Comment
S0	1	-1..15	Number of RING indication before automatically answering an incoming connection. A value of 0 disables autoanswer. If -1, then autoanswer on one RING and do NOT send RING/CONNECT response to the host. This emulates a serial cable replacement situation Setting values >= 0, resets S Register 504 to 0 and <0 forces 504 to 1. If S0 <> 0 and S100 <> 0 then S0 must be < S100. If a value is entered which violates this rule, then ERROR 29 is sent in response. If S504 =1 then this register will return -1, regardless of the actual value stored in non-volatile memory.
S2	0x5E	0x20..0x7E	Escape sequence character. It is not '+' by default as a Bluetooth serial link can be used to connect to a mobile phone which exposes an AT command set, which will in turn use '+' as default. So if both used '+' there will be confusion. 0x5e is the character '^'.
S12	100	40..5000	Escape sequence guard time in milliseconds, with a granularity of 20ms. New values are rounded down to the nearest 20ms multiple
S100	15	0..15	Number of RING indications before an auto disconnection is initiated. A value of 0 disables this feature. If S0 <> 0 and S100 <> 0 then S0 must be < S100. If a value is entered which violates this rule, then ERROR 29 is sent in response.
S101	$1101	0..$ffff	UUID of default SPP based profile when not specified explicitly in the ATD command.
S102	1	1..7	Defines a set of masks for enabling profile servers. 1 is Serial Port Profile 2 is Headset 4 is DUN
S500	0	0..1	Authentication for outgoing connections. Set to 1 to Enable Authentication.
S501	0	0..1	Encryption for outgoing connections. Set to 1 to Enable Encryption.
S502	0	0..1	Authentication for incoming connections. Set to 1 to Enable Authentication.
S503	0	0..1	Encryption for incoming connections. Set to 1 to Enable Encryption.
S504	0	0..1	Setting to 1 will force S0 to -1 and will suppress messages arising from connections or pairing. E.g CONNECT, NO CARRIER, RING, PAIR etc. Suppressing connection based messaged allows the blu^{2i} device to be configured in cable replacement mode.
S505	5	2..120	Minimum delay before abandoning connection attempt as a master. Referenced by ATD. In units of seconds. See S Registers 530 and 543 also. Please note that as disconnection time can vary, this register only guarantees the minimum delay. Note that for invalid addresses specified in the ATD command, the "NO CARRIER" response will be immediate. See S register 560 for specifying disconnect max timeout.
S506	1	0..1	Enable/Disable echoes. The ATEn command also affects this.
S507	0	0..2	When set to 0, a connection can be dropped using ^^^ escape sequence only and the state of DSR line is ignored. When set to 1 a connection can be dropped using EITHER the ^^^ escape sequence OR the DSR handshaking line. Finally when set to 2, a connection can only dropped using a deassertion of DSR. Mode 2 provides for the highest data transfer rate. If the status of the DSR line is to be conveyed to the remote device as a low bandwidth signal then this register MUST be set to 0, otherwise a deassertion of DSR will be seen as a request to drop the Bluetooth connection. This register affects S Register 536 – see details of 536 For the Go blue Activator variant this can only be set to 0.
S508	640	10..2550	Page Scan Interval in milliseconds. Minimum is 11.25ms so 10/11ms will give 11.25ms.
S509	320	10..2550	Page Scan Window in milliseconds. Minimum is 11.25ms so 10/11ms will give 11.25ms.
S510	640	10..2550	Inquiry Scan Interval in milliseconds. Minimum is 11.25ms so 10/11ms will give 11.25ms.
S511	320	10..2550	Inquiry Scan Window in milliseconds. Minimum is 11.25ms so 10/11ms will give 11.25ms.
S512	1	0..7	Specify power up state. When set to 0, AT+BTO is required to open the device for Bluetooth activity. When set to 1, it proceeds to a state as if AT+BTO was entered. When set to 2, it will be discoverable only, similar to issuing AT+BTQ. When set to 3, it will be connectable but not discoverable e.g. AT+BTG When set to 4, it will be connectable and discoverable e.g. AT+BTP. When set to 5, it will be like 2, but all UART RX traffic is discarded in absence of a connection while DSR is asserted. If DSR is not asserted, then it behaves exactly as per mode 2. When set to 6, it will be like 3, but all UART RX traffic is discarded in absence of a connection while DSR is asserted. If DSR is not asserted, then it behaves exactly as per mode 3. When set to 7, it will be like 4, but all UART RX traffic is discarded in absence of a connection while DSR is asserted. If DSR is not asserted, then it behaves exactly as per mode 4. Note that by implication, a change to this can only be seen after a power cycle AND if AT&W is **actioned** prior to the power cycle. If S Reg 554 is non-zero and this register is between 2 and 7 inclusive, then the value of S554 specifies the time in seconds that the device will remain in the specified mode after power up. On timeout, the device will fall back to the mode specified in S Register 555.

			S512 continued…. In some firmware builds, S Registers 565 to 569 inclusive are visible, which allows the startup mode to depend on the state of RI line (Setting S Reg 565 forces the RI pin to be configured as an input). For this feature to be active, SReg 565 should be set to 1. In that case, on startup, if RI is asserted, then the startup mode is defined by S Reg 566 and if deasserted then S Reg 567.
S513	1	0..1	Pairing Authentication, 1 = Enable
S514	10	1..60	Pairing Timeout in seconds. This includes the time a host takes to supply the PIN number when PIN? messages are indicated.
S515	0x001F00	0.. 0xFFFFFF	Default Device Class Code to be used with AT+BTO when it is not explicitly specified. When queried, the value is always printed as a hexadecimal number. To change the device class of the module, after AT+BTO, use the command AT+BTC.
S516	0x000000	0..0x2FFFFF	Default Device Class filter to be used with AT+BTI when it is not explicitly specified. When queried the value is always printed as a hex number. The seventh most significant digit, can be 0,1 or 2, and is used to specify the type of device class filter. When 0, it specifies no filtering. When 1, it specifies an AND mask and all 24 bits are relevant When 2, it specifies a filter to look for devices with matching major device class which occupies a 5 bit field from bits 8 to 12 inclusive (assuming numbering starts at bit 0). All other 19 bits MUST be set to 0.
S517	20	2..61	Inquiry Length in units of seconds. This parameter is referenced by the AT+BTI command
S518	8	0..255	Maximum number of responses from an inquiry request. This parameter is reference by the AT+BTI command. If this number is set too high, then AT+BTI will return ERROR 27. For a particular firmware revision, determine the effective maximum value by trial and error. That is, set to a high value, send AT+BTI and if ERROR 27 is returned, then retry with a smaller value. This effective max value will remain unchanged for that particular firmware build.
S519	500	100..6000	When S507>0, and in a connection, DSR can be used to change from data to command state by deasserting the DSR line for less than the time specified in this register. This value is rounded down to the nearest 100ms
S520	See Comment	1200..115200	Change baud rate. The effect is immediate and in fact the OK will be sent at the new baud rate. Only one of the following baud rates are accepted: 1200,2400,4800,9600,19200,28800,38400,57600,115200. If S register 525=1, then the maximum baud rate is limited to 115200 The default is 9600 for the blu^{2i} Module and 115200 for other blu^{2i} devices. For the Go blue Activator variant of the module this register is read only See S Register 526 for further information.
S521	See Comment	1200..921600	Change baud rate. The effect is immediate and in fact the OK will be sent at the new baud rate. Due to the integer arithmetic involved, the actual baud rate is different. To inspect the actual baud rate, do ATS521? If the actual baud rate is more than 2% offset from the desired, then an ERROR will be returned and the old baud rate will prevail. If S Register 525=1, then the max baud rate is limited to 115200 **In the event that a non-standard baud rate is requested, it is entirely possible that the host is not capable of generating such a baud rate and in that circumstance the blu^{2i} device cannot be communicated with.** **There is a procedure to recover from this situation and it is described in section titled "Factory Default Mode"** The default is 9600 for the blu^{2i} Module and 115200 for other blu^{2i} devices. For the Go blue Activator variant of the module this register is read only See S Register 526 for further information
S522	1	1	1 = CTS/RTS hardware handshaking enabled For the Go blue Activator variant of the module this register is read only. See S Register 526 for further information.
S523	1	1..2	Number of Stop bits For the Go blue Activator variant of the module this register is read only. See S Register 526 for further information.
S524	0	0..2	Parity. 0=None, 1=Odd, 2=Even For the Go blue Activator variant of the module this register is read only. See S Register 526 for further information.
S525	See Comment	0..1	Apply multiplier of 8 to baud rate internally. This is set to 0 (disabled) by default for the blu^{2i} Module/RS-232 Adaptor/Universal RS-232 Adaptor, and set to 1 (enabled) by default for the blu^{2i} PC Card. It is required in the PC Card because the UART chip on the PC Card is driven by a 14.7456MHZ crystal instead of 1.8432MHz. This means that when a host asks for a baud rate, in reality it gets a baud rate which is 8 times faster. If S Register 521 > 115200 then this register cannot be set to 1. For the Go blue Activator variant of the module this register is read only. See S Register 526 for further information.

S526	3	1..3	This register specifies a 2 bit mask used to qualify how S Registers 520 to 525 are actioned. When bit 0 is 1, the new comms parameter affects the UART immediately. When bit 1 is 1, the new comms parameter is stored in non-volatile memory So for example, to change comms parameters, but have them come into effect only after subsequent power cycles, then this register should be set to 2, and likewise to affect immediately and yet not have it persist over a power cycle, the value should be set to 1.
S530	1000	100..15000	Reconnect delay when configured as master in pure-cable-replacement mode. This value is rounded down to the nearest 100ms. See S Register 505 and 543 also
S531	0	0..3	Specifies the mode on connection establishment. 0 = Normal, that data is exchanged between UART and RF 1 = LOCAL_COMMAND. UART input is parsed by the AT interpreter and RF data is discarded 2 = REMOTE_COMMAND. RF input is parsed by the AT interpreter and UART data is discarded. 3=LOCAL_COMMAND. UART input is parsed by the AT interpreter and incoming RF data is sent to the host using the **RX<string>** asynchronous response. If S Reg 536 is not 1 then this register cannot be set to 2 and an ERROR will be returned
S532	0	0..3	If non zero then on every connection, and SCO channel (audio) will be initiated. If value =1 then HV1, if =2 then HV2 and finally if =3 then HV3
S533	1	0..2	If set to 1 then left LED follows RI state, if set to 2 then it follows the state of DSR and if 0 then neither and GPIO5 is available as a user I/O. This register will not necessarily be effective immediately after changing the value. It must be saved to non-volatile memory using AT&W and will operate as expected after an ATZ or a power cycle.
S534	1	0..1	If set to 1 then right LED follows DCD state, otherwise GPIO4 is available as user i/o. This register will not necessarily be effective immediately after changing the value. It must be saved to non-volatile store using AT&W and will operate as expected after an ATZ or a power cycle.
S535	20	0..41	Link Supervision Timeout. If units go out of range, then a NO CARRIER message will be sent to the host after the time specified here
S536	0	0..1	When set to 1, a remote device can 'capture' the AT parser of this unit by it sending this module an escape "!!!" sequence. The inter character timing is set via S Register 12. If S Register 507 is >= 2, then reading this register will always return 0 and writing 1 will result in ERROR 33.
S537	X	X..X	This register is no longer available – see 551,552,553 instead It only exists in firmware version 1.1.12 to 1.1.47 The functionality it controlled is now defined by registers 551,552 and 553
S538	0	0..1	If 1, then when a successful pairing occurs, it is automatically saved in the trusted device database – if it has room to store it.
S539	0	0..1	When set to 1, in idle mode (S512=1), UART Rx characters are discarded if DSR is deasserted.
S540	0	0 48..127	Sets the MTU in L2CAP configuration negotiations. The value of 0 is a special value which is taken to mean that the current value should remain.
S541	6	-25..6	This sets the power level in dBm when inquiring or paging. Reading this register returns the value stored in non-volatile memory.
S542	6	-25..6	As per S541, however reading this register returns the current power level as set in the base band. The read can be different from S541because the actual power is set using a lookup table and the base band rounds **down** to the nearest value in the table.
S543	0	0..1	If this is set to 1, then incoming pairing attempts will be accepted (if a pin code has been pre-entered using AT+BTK) while in the wait phase of auto connect cycle initiated by the AT+BTR command. In addition to accepting pairing attempts, if the pairing is successful, then the new device is automatically set as the peer address for automatic connections (as if an explicit AT+BTR command was entered). See S Register 505 and 530 also
S544	1	0..1	Configure the UART for either low latency or maximum throughput. A setting of 1 gives maximum throughput.
S551	0x3211	0xFFFF	This register specifies in each 4 bit nibble, how the outgoing modem status bits to the remote peer gets its value. Bluetooth allows for RTR, RTC, DV and IC bits to be exchanged over an RFCOMM connection. Nibble 0..3 specifies the source for RTC 4..7 specifies the source for RTR 8..11 specifies the source for DV (i.e. DCD) 12..15 specifies the source for IC (i.e. RI) Each nibble can take the following value:- 0 Always set to 0 1 Always set to 1 2 If DCD (pin 8 on module connector) is output then always 1 If DCD is input then 1 if DCD is asserted otherwise 0 3 If RI (pin 6) is output then always 0 If RI is input then 1 if RI is asserted otherwise 0 4 If DSR (pin 10) is asserted then 1 otherwise 0

			In the event that a nibble specifies DSR as the source of its state, be aware that if, S Register 507 is anything other than 0, a de-assertion of DSR will cause the Bluetooth connection to be dropped. If bits 0..3 and 4..7 are set to 0, then some Bluetooth devices will use that as a signal to stop sending any data back. For example, Nokia 6310 stops responding. If this register is changed while in command and connected mode, then on going back online using the ATO command, a fresh signal will be sent to the peer to update the bits.
S552	0x0122	0x0FFF	This register specifies in each 4 bit nibble, how the DTR, DCD, RI output pins are controlled when in a Bluetooth connection Nibble 0..3 specifies the source for DTR 4..7 specifies the source for DCD 8..11 specifies the source for RI Each nibble can take the following value:- 0 Do NOT touch the I/O 1 Always deassert 2 Always assert 3 If RTC bit in CONTROL_IND is 1 then assert otherwise deassert 4 If RTR bit in CONTROL_IND is 1 then assert otherwise deassert 5 If DV bit in CONTROL_IND is 1 then assert otherwise deassert 6 If IC bit in CONTROL_IND is 1 then assert otherwise deassert If this register is changed while in command and connected mode, then on going back online using the ATO command, the modem output lines will get refreshed.
S553	0x0201	0x0FFF	This register specifies in each 4 bit nibble, how the DTR,DCD,RI output pins are controlled when NOT in a Bluetooth connection Nibble 0..3 specifies the source for DTR 4..7 specifies the source for DCD 8..11 specifies the source for RI In addition it also refers to S Register 552 to see if the relevant pin is an input or not to be touched. If the nibble in 552 is 0, then the relevant pin is an input. Each nibble can take the following value:- 0 Always deassert 1 Always assert 2 Assert if RING is being sent to the host The default for the Universal RS-232 Adaptor is $0200.
S554	0	0..900	If S Register 512>=2 and <=7 then this register specifies a time in seconds for which the device will stay in the S512 mode after power up or reset. On timeout, it will abort the discoverable and/or connectable and fall back into S512=1 mode, when it is deaf and dumb. Note that if AT+BTR has been used to specify a peer device, then on reverting to mode 1, it will attempt to make a connection to that peer device. A power cycle, reset via BREAK or ATZ is required to see effects of a change.
S555	1	1..7	If S Register 554 is nonzero, then after the post reset window expires, the mode will revert to the mode specified in this register. This allows, for example, the device to be discoverable and connectable on power up (mode 4 or 7) and on window timer expiry to revert to connectable only (mode 3 or 6). A power cycle, reset via BREAK or ATZ is required to see effects of a change. In some firmware builds, S Registers 565 to 569 inclusive are visible, which allows the startup mode to depend on the state of RI line (Setting S Reg 565 forces the RI pin to be configured as an input). For this feature to be active, SReg 565 should be set to 1. In that case, on startup, if RI is asserted, then the startup mode is defined by S Reg 568 and if deasserted then S Reg 569.
S558	0	0..1	When 1, the following responses; “RING”, “NO CARRIER” and “CONNECT” are replaced by “BTIN”, “BTDOWN” and “BTUP” respectively. This will eliminate ambiguity when the module has a Bluetooth connection to a AT modem which also gives these responses.
S559	0	0..3	This specifies a mask. When Bit 0 is 1, the response word “ERROR“ is replaced by “BTERR” and “OK” is replaced by “ok”. When Bit 1 is 1, then error responses do not include the error number and instead the error number can be retrieved using ATI12.
S560	15	15..120	Disconnect timeout in seconds. This timer specifies how long to wait for confirmation from the peer device and/or the underlying stack that the connection has been successfully torn down. There can be instances where a confirmation does not arrive and so in this case this timer is used to ‘close off’ the procedure and put the state machine back into a proper mode for new operations.

			Time is specified with 15 seconds intervals.
S561	0	0..1000	Sniff Attempt Time in units of 100 milliseconds. 0 means disable See section “Power Consumption and Reset” in the user guide for more details.
S562	0	0..1000	Sniff timeout Time in units of 100 milliseconds. 0 means disable See section “Power Consumption and Reset” in the user guide for more details.
S563	0	0..1000	Sniff Minimum Interval in units of 100 milliseconds. 0 means disable See section “Power Consumption and Reset” in the user guide for more details.
S564	0	0..1000	Sniff Maximum Interval in units of 100 milliseconds. See section “Power Consumption and Reset” in the user guide for more details.
S565	0	1	If set to 1, RI (Ring Inicate) line is configured as an input and forces the startup mode (SReg512) and post-timeout on Startup mode (SReg555) to be dependent on the state of RI. The RI conditional modes are defined by SRegs 566 to 569 inclusive.
S566	1	7	If S565=1, and RI is asserted then this is the mode the device will start up in.
S567	1	7	If S565=1, and RI is deasserted then this is the mode the device will start up in.
S568	1	7	If S565=1, and RI is asserted then this is the mode the device will assume after the post-startup timeout defined in SReg 554 instead of mode defined in SReg555
S569	1	7	If S565=1, and RI is deasserted then this is the mode the device will assume after the post-startup timeout defined in SReg 554 instead of mode defined in SReg555
S610	0	0..31	Set direction of digital I/O lines. This is a mask made up of 5 bits. Setting a bit to 1 makes that I/O line an output. GPIO1 is bit 0, GPIO2 is bit 1, up to bit 4 for GPIO5.
S611	0	1	Set to 1 to invert the logic of GPIO outputs. For example, ATS621=1 will set the output pin to low and vice versa.
S620	n/a	0..31	Read/Write to all 8 Digital lines in one atomic step. The value is returned as a 4 digit hexadecimal value with trailing 0s.
S621	n/a	0..1	Read/Write to GPIO1
S622	n/a	0..1	Read/Write to GPIO2
S623	n/a	0..1	Read/Write to GPIO3
S624	n/a	0..1	Read/Write to GPIO4
S625	n/a	0..1	Read/Write to GPIO5
S626	n/a	0	Read/Write to GPIO6 - Not Available so 0 always returned
S627	n/a	0	Read/Write to GPIO7 - Not Available so 0 always returned
S628	n/a	0	Read/Write to GPIO8 - Not Available so 0 always returned
S631	n/a	0..65535	When GPIO1 is configured as an input, low to high transitions are counted. There is no software debouncing. External RC circuit may be required. The counter wraps to 0 when it overflows beyond 65535.
S632	n/a	0..65535	When GPIO2 is configured as an input, low to high transitions are counted. There is no software debouncing. External RC circuit may be required. The counter wraps to 0 when it overflows beyond 65535.
S641	n/a	0..65535	As per 631, but the action of reading the value will reset the count to 0.
S642	n/a	0..65535	As per 632, but the action of reading the value will reset the count to 0.
S701	n/a	0..65535	Read/Write to Analogue Line 0, when reading value is returned in decimal
S702	n/a	0..65535	Read/Write to Analogue Line 1, when reading value is returned in decimal
S711	n/a	0000..FFFF	Read/Write to Analogue Line 0, when reading value is returned in hexadecimal
S712	n/a	0000..FFFF	Read/Write to Analogue Line 1, when reading value is returned in hexadecimal
S721	0	0	Set direction of Analogue Line 0
S722	0	0	Set direction of Analogue Line 1
S1001 to S1010		0.. 2^32	10 General Purpose 32 bit Registers for use by host. These are stored in non-volatile memory.

2.2.16 ATSn? {Read S Register Value}

This will return the current value of register n.

For recognised values of n
Response: <cr,lf>**As Appropriate**<cr,lf>**OK**<cr,lf>

For unrecognised values of n
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.17 ATSn=? {Read S Register – Valid Range}

This will return the valid range of values for register n.

For recognised values of n
Response: <cr,lf>**Sn:(nnnn..mmmm)**<cr,lf>**OK**<cr,lf>

For unrecognised values of n
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.18 ATX<string> {Send Data in Local Command and Connected Mode}

This command is used to send data to the remote device when in local command and connected mode.

The parameter <string> is any string not more than 24 characters long. If a non-visual character is to be sent then insert the escape sequence \hh where hh are two hexadecimal digits. The 3 character sequence \hh will be converted into a single byte before transmission to the peer.

Response: <cr,lf>**OK**<cr,lf>

2.2.19 ATZ {Hardware Reset}

Forces the device through a hardware reset which means it will eventually come alive in the local command and unconnected mode. This allows changes to the PS store to take effect. Allow for about 2 seconds for the device to start responding to AT commands again. The best way of determining that the device is alive again, is to keep sending it AT<cr> until it responds with an OK response.

Response: <cr,lf>**OK**<cr,lf> *note OK is returned before the RESET*

2.2.20 AT&Fn {Set S Register Defaults}

This command will only work when the device is in local command and unconnected mode. Depending on the value of 'n' it installs S Register values appropriate for various power modes, ranging from minimum power consumption to maximum.

Legal values of 'n' are as per the following table. All other values of n will generate a syntax error response. If 'n' is not specified then a default value of 0 is assumed where the baud rate is NOT changed.

&F0 (Default)	Medium power consumption, UART baud rate unchanged, Left LED off, Right LED = DCD
&F1	Minimum power consumption, UART baud rate set to 9600, Left and Right LED off
&F2	Minimum power consumption, UART baud rate set to 38400, Left and Right LED off
&F3	Minimum power consumption, UART baud rate set to 115200, Left and Right LED off
&F4	Medium power consumption, UART baud rate set to 115200, Left LED off, Right LED = DCD
&F5	Maximum power consumption, UART baud rate set to 115200, Left LED=DSR, Right LED = DCD
&F6	Maximum power consumption, UART baud rate set to 115200, Left LED=DSR, Right LED = DCD Explicitly set higher baud rates using ATS521=n

Please refer to the "Power Consumption" chapter in the relevant blu^{2i} device user guide, for more detailed infromation of power usage.

The new values are NOT updated in non-volatile memory until the AT&W command is sent to the blu^{2i} device.

Response: <cr,lf>**OK**<cr,lf>
Or
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.21 AT&F* {Clear Non-volatile Memory}

The AT&F* variant of the command installs values in S registers as per command AT&F4 and then all other user parameters in non-volatile memory are erased. This means that the trusted device database is cleared, and so are parameters related to the following commands:- AT+BTR, AT+BTN, AT+BTS.

Response: <cr,lf>**OK**<cr,lf>
Or
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.21 AT&F+ {Clear Non-volatile Memory}

This command erases all user parameters in non-volatile memory except S Registers 520 to 525. This means that the trusted device database is cleared, and so are parameters related to the following commands:- AT+BTR, AT+BTN, AT+BTS.

Response: <cr,lf>**OK**<cr,lf>
Or
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.22 AT&W {Write S Registers to Non-volatile Memory}

Writes current S Register values to non-volatile memory so that they are retained over a power cycle.

Response: <cr,lf>**OK**<cr,lf>
Or
Response: <cr,lf>**ERROR nn**<cr,lf>

2.2.23 AT+BTAn {Control Audio Channel}

Once a Bluetooth connection is active, **and assuming the peer device is a TDK blu^{2i} device**, this command is used to start/stop a SCO channel which will connect the PCM interfaces of the two peer devices. This means that if a codec is attached to the PCM pins, then 2-way audio can be established.

+BTA0 Switch off the channel.
+BTA1 Switch on the channel.

On receipt of the command, the following response immediately follows.

Response: <cr,lf>**OK**<cr,lf>

The lower layers then go through the process of setting up the SCO channel, and as soon as a SCO link is established, the following response is asynchronously sent to the host.

Response: <cr,lf>**AUDIO ON**<cr,lf>

Or if the SCO failed to be established.

Response: <cr,lf>**AUDIO FAIL**<cr,lf>

On the peer device, the host will asynchronously get ….

Response: <cr,lf>**AUDIO ON**<cr,lf>

2.2.24 AT+BTC<devclass> {Set Device Class Code}

This command is used to set the device class code which will be sent in subsequent inquiry responses. It can be read back using the AT+BTC? Command, as described below.

<devclass> is a 6 digit hexadecimal number derived as per section "1.2 The Class of Device/Service Field" of the Bluetooth specification "Bluetooth Assigned Numbers".

The 24 bits are made of 4 fields briefly described as follows (bit 0 corresponds to the least significant bit):-

Bits 0-1:	Format Type. This field currently only has a value of 00 (i.e. format type 1).
Bits 2-7:	These 6 bits define the Minor Device Class and the value is interpreted differently based on the Major Device class stored in the next 5 bits.
Bits 8-12:	These 5 bits define the Major Device Class as per Table 1.3 in "Bluetooth Assigned Numbers".
Bits 13-23:	This is an 11 bit field used as a mask to define the Major Service Class, as per Table 1.2 in "Bluetooth Assigned Number".

blu^{2i} devices do not map to any predefined Major Service Class or Major Device Class and so the default devclass as shipped is 001F00, which means no Major Service Class and "Unclassified" Major Device class.

Other examples of device class codes are follows:-

Code (Hexadecimal)	Name	Major Service	Major Device	Minor Device
001F00	Unclassified	None	Unclassified	n/a
200404	Headset	Audio	Audio	Headset

Response: <cr,lf>**OK**<cr,lf>

Or for an invalid <devclass> value (usually a value which is not 6 hexadecimal characters long).

Response: <cr,lf>**ERROR 08**<cr,lf>

2.2.25 AT+BTC? {Read Device Class Code}

This command is used to read the current device class code.

Response: <cr,lf>**123456**
<cr,lf>**OK**<cr,lf>

2.2.26 AT+BTD<bd_addr> {Remove Trusted Device}

This command is used to remove the specified device from the list of trusted devices in the non-volatile database. If the device is not in the database then the response will still be an OK.

Response: <cr,lf>**OK**<cr,lf>

2.2.27 AT+BTD* {Remove All Trusted Devices}

This command is used to remove all devices from the list of trusted devices in the non-volatile database. No confirmation will be asked for. So beware!!!

***WARNING**: If you make an authenticated connection, the link key gets cached in the underlying stack. So if you subsequently delete the key using AT+BTD* and immediately request an authenticated connection to the same device, then the connection will be established. To ensure this does not happen, either send ATZ after the AT+BTD* OR send AT+BTD<bd_addr> for each item in the trusted device database.*

Response: <cr,lf>**OK**<cr,lf>

2.2.28 AT+BTF=<string> {Set Friendly Name}

This sets the friendly name of this device as seen by other devices

Response: <cr,lf>**OK**<cr,lf>

2.2.29 AT+BTG<bd_addr> {Enable Cautious Page Scanning ONLY}

Enable page scanning and wait for a connection from device with Bluetooth address <bd_addr>. If the specified address is 000000000000 then incoming connections are accepted from any device, is as per AT+BTP without an address. Inquiry Scans are disabled.

This command also has variants which allow authentication and encryption to be explicitly specified. For example:-

AT+BTGU123456789012
AT+BTGY123456789012
AT+BTGUY123456789012
AT+BTGYU123456789012

Response: <cr,lf>**OK**<cr,lf>

2.2.30 AT+BTG {Enable Promiscuous Page Scanning ONLY}

Enable page scanning only and wait for a connection from any device. Inquiry scans are disabled. Authentication and Encryption is as per S registers 502 and 503.

Response: <cr,lf>**OK**<cr,lf>

2.2.31 AT+BTGU {Enable Promiscuous Page Scanning ONLY}

Enable page scanning only and wait for a connection from any device. Inquiry scans are disabled. Authentication is enabled and encryption is disabled.

Response: <cr,lf>**OK**<cr,lf>

2.2.32 AT+BTGY {Enable Promiscuous Page Scanning ONLY}

Enable page scanning only and wait for a connection from any device. Inquiry scans disabled. Authentication is disabled and encryption is enabled.

Response: <cr,lf>**OK**<cr,lf>

2.2.33 AT+BTGUY {Enable Promiscuous Page Scanning ONLY}

Enable page scanning only and wait for a connection from any device. Inquiry scans are disabled. Authentication and encryption are both enabled. The order of U and Y is not significant.

Response: <cr,lf>**OK**<cr,lf>

2.2.34 AT+BTI<devclass> {Inquire}

This will make the device perform an inquiry for device class code for **delay** milliseconds and **max** number of unique responses, where **delay** is specified by S register 517 and **max** is specified by S register 518.
The <devclass> is an optional parameter where the value specifies either a 6 digit device class code or a 2 digit major device class. If it is not specified, then the value is taken from S register 516.

When <devclass> is 6 hexadecimal characters long, it specifies an AND mask which is used to filter inquiry responses. When <devclass> is 2 hexadecimal characters long, it forces the inquiry to filter responses to devices that match their major device class code to this value – which can only be in the range 00 to 1F.

Response: <cr,lf>**12346789012**
<cr,lf>**12345678914**
<cr,lf>**OK**<cr,lf>

If the module is waiting for an incoming connection, (entered via AT+BTP, AT+BTG, AT+BTQ), then it will respond with ERROR 14. To perform the inquiry, send AT+BTX to put the module back into idle mode.

Response: <cr,lf>**ERROR 14**<cr,lf>

ERROR RESPONSE

A Bluetooth inquiry process is such that for a single inquiry request a device could respond many times. To ensure that an address is sent to the host only once for a particular AT+BTI, an array of addresses is created at the start of each AT+BTI and is filled as responses come in. This array of addresses is stored in dynamic memory and as such if the memory allocation fails then the inquiry procedure is aborted and in that case an error response is sent to the host.
*To clarify, a single AT+BTI will **never** return the same Bluetooth address more than once, but as long as the responding device is active, all AT+BTI commands will always return it.*

Response: *<cr,lf>**ERROR 27**<cr,lf>*

2.2.35 AT+BTIV<devclass> {Inquire}

As per AT+BTI but the response includes the device class code for all inquiry responses. Please refer to the 'ERROR RESPONSE' note in the description for AT+BTI<devclass>.

Response: <cr,lf>**12346789012,123456**
<cr,lf>**12345678914,123456**
<cr,lf>**OK**<cr,lf>

2.2.36 AT+BTIN<devclass> {Inquire}

As per AT+BTI but the response includes the device class code and friendly name for all inquiry responses. Please refer to the 'ERROR RESPONSE' note in the description for AT+BTI<devclass>. The friendly name strings are in UTF-8 format as per the Bluetooth specification.

Response: <cr,lf>**12346789012,123456,"TDK AT DONGLE 1"**
<cr,lf>**12345678914,123456, "TDK blu2i RS232"**
<cr,lf>**OK**<cr,lf>

2.2.37 AT+BTK=<string> {Set Passkey}

This command is used to provide a passkey when **PIN? 12345678** indications are received asynchronously. If a pairing is not in progress then the pin is written to non-volatile memory for future use. Specifying an empty string deletes the key from the non-volatile memory.
The string length must be in the range 0 to 8, otherwise an error will be returned.

Response: <cr,lf>**OK**<cr,lf>

2.2.38 AT+BTM<bd_addr> {Set Incoming Peer Address}

This command is used to store a peer address for incoming connections in non-volatile memory. A value of 000000000000 has the special meaning of invalid peer address.

When S register 512 = 3, 4, 6 or 7 then it will wait for an incoming connection from the peer address specified. If the peer address is not 000000000000, then it waits for a connection from the specified master, otherwise will connect to anyone.

Response: <cr,lf>**OK**<cr,lf>

2.2.39 AT+BTM {Delete Incoming Peer Address}

This command is used to delete the peer address previously stored using AT+BTR<bd_addr>.

Response: <cr,lf>**OK**<cr,lf>

2.2.40 AT+BTM? {Read Incoming Peer Address}

This command is used to display the peer address stored in non-volatile memory, used to put the module in pure cable replacement mode.

Response: <cr,lf>**12346789012**
<cr,lf>**OK**<cr,lf>

If the location is empty the response is as follows.

Response: <cr,lf>**00000000000**
<cr,lf>**OK**<cr,lf>

2.2.41 AT+BTN=<string> {Set Friendly Name in Non-volatile Memory}

This sets the default friendly name of this device as seen by other devices. It will be stored in non-volatile memory. Use AT+BTF to make the name visible to other devices. Use AT+BTN? To read it back. An empty string (“”) will delete the string from non-volatile memory which will force the default name to be used.

Response: <cr,lf>**OK**<cr,lf>

2.2.42 AT+BTN? {Read Friendly Name from Non-volatile Memory}

Read the default friendly name from non-volatile memory.

Response: <cr,lf>**"My FriendlyName"**<cr,lf>
<cr,lf>**OK**<cr,lf>

2.2.43 AT+BTO<devclass> {Open and make Unit Detectable}

After power up and ATZ, this command is sent so that RFCOMM is initialised and opened and the service name as specified in AT+BTN is exposed via the SDP registry.

The <devclass> value specifies an optional fixed length hexadecimal device class code. If it is not specified, then the device class code is taken from S Register 515.

For this command to be effective, S Register 512 must be set to 0.

Response: <cr,lf>**OK**<cr,lf>

2.2.44 AT+BTP<bd_addr> {Enable Cautious Page/Inquiry Scanning}

Enable page scanning and wait for a connection from device with Bluetooth address <bd_addr>. If the specified address is 000000000000 then incoming connections are accepted from any device, is as per AT+BTP without an address. Inquiry scanning is also enabled.

This command also has variants which allow authentication and encryption to be explicitly specified. For example:-

AT+BTPU123456789012
AT+BTPY123456789012
AT+BTPUY123456789012
AT+BTPYU123456789012

Response: <cr,lf>**OK**<cr,lf>

2.2.45 AT+BTP {Enable Promiscuous Page/Inquiry Scanning}

Enable page scanning and wait for a connection from any device. Inquiry scanning is also enabled. Authentication and Encryption is as per S registers 502 and 503.

Response: <cr,lf>**OK**<cr,lf>

2.2.46 AT+BTPU {Enable Promiscuous Page/Inquiry Scanning}

Enable page scanning and wait for a connection from any device. Inquiry scanning is also enabled. Authentication is enabled and encryption is disabled.

Response: <cr,lf>**OK**<cr,lf>

2.2.47 AT+BTPY {Enable Promiscuous Page/Inquiry Scanning}

Enable page scanning and wait for a connection from any device. Inquiry scanning is also enabled. Authentication is disabled and encryption is enabled.

Response: <cr,lf>**OK**<cr,lf>

2.2.48 AT+BTPUY {Enable Promiscuous Page/Inquiry Scanning}

Enable page scanning and wait for a connection from any device. Inquiry scanning is also enabled. Authentication and encryption are both enabled. The order of U and Y is not significant.

Response: <cr,lf>**OK**<cr,lf>

2.2.49 AT+BTQ {Enable Inquiry Scans ONLY}

When inquiry scan is enabled, it implies that this device will respond to inquiries from other devices. Use AT+BTX to disable inquiries.

Response: <cr,lf>**OK**<cr,lf>

2.2.50 AT+BTR<bd_addr> {Set Outgoing Peer Address}

This command is used to store a peer address for outbound connections in non-volatile memory. A value of 000000000000 has the special meaning of invalid peer address.
This command is used to set up a module in pure cable replacement mode.

If S register 512 = 1 and the peer address is NOT 000000000000, then it will periodically (time specified via S register 505) attempt to connect to the peer address specified. In this circumstance all commands from the host are buffered in the

receive buffer, until a Bluetooth connection is established with the peer device and it then sends0 the buffer across. This means that if the peer device is not in the vicinity and will never be there, the device effectively becomes useless, as in this circumstance a host would want to get attention of the AT parser to send it new commands – probably one to delete the peer device.
In this circumstance, a recovery is possible, by one of two methods. The first method assumes that the DTR from the host is connected to the DSR line of the module and the second method assumes that this connection is absent. In the first method it is enough to deassert the DTR line from the host and that will abort the autoconnect cycle. The second method is initiated by resetting the device and then ensuring that the text string “AT+BT&BISM&<cr>” is sent (where <cr> is the carriage return character). There is special code which looks out for this magic command and terminates the autoconnect cycle if it sees it and confirms to the host of that fact by sending an “OK” response.

Response: <cr,lf>**OK**<cr,lf>

2.2.51 AT+BTR {Delete Outgoing Peer Address}

This command is used to delete the peer address previously stored using AT+BTR<bd_addr>.

Response: <cr,lf>**OK**<cr,lf>

2.2.52 AT+BTR? {Read Outgoing Peer Address}

This command is used to display the peer address stored in non-volatile memory, used to put the blu^{2i} device in pure cable replacement mode.

Response: <cr,lf>**12346789012**
<cr,lf>**OK**<cr,lf>

If the location is empty the response is as follows.

Response: <cr,lf>**00000000000**
<cr,lf>**OK**<cr,lf>

2.2.53 AT+BTS=<string> {Set Service Name}

This writes the name to non-volatile memory. It will be used after ATZ, power cycle or AT+BTO if it has not been issued yet. Use AT+BTS? to read it back from non-volatile memory. An empty string (“”) will delete the string from non-volatile memory which will force the default service to be used.

Response: <cr,lf>**OK**<cr,lf>

If the service name cannot be set for any reason then an error response **ERROR 11** is returned.

2.2.54 AT+BTS? {Read Service Name from Non-volatile Memory}

Reads the default service name from non-volatile memory.

Response: <cr,lf>**"My ServiceName"**<cr,lf>

<cr,lf>**OK**<cr,lf>

2.2.55 AT+BTT {Add Trusted Device}

This command is used to store the cached link key in the non-volatile database. If the database is full it will respond with an ERROR. If the device is already in the database, then the key is replaced.
If the link key cache is empty, that is, a pairing has not been performed since the device was powered, then the response will be an ERROR.

Response: <cr,lf>**OK**<cr,lf>
or
Response: <cr,lf>**ERROR**<cr,lf>

2.2.56 AT+BTT? {List Trusted Device}

This command is used to list the contents of the trusted device database. The link key is NOT displayed so the response is as shown below. If the list is empty then just the OK response is sent otherwise an OK is used to terminate the list. Use the command ATI6 to read the maximum size of the trusted device database.

Response: <cr,lf>**12346789012**
<cr,lf>**12345678913**
<cr,lf>**12345678914**
<cr,lf>**OK**<cr,lf>

2.2.57 AT+BTW<bd_addr> {Initiate Pairing}

This initiates pairing with a device whose Bluetooth address is <bd_addr>. An OK response is sent and when the PIN is required, asynchronous indications will be sent to the host in the form **PIN? <bd_addr>** where the address confirms the device with which the pairing is to be performed. To supply a PIN, use the AT+BTK command.

For a successful pairing, the link key is stored in a volatile cache which is overwritten everytime a new pairing is initiated using this command. The link key can be stored in a non-volatile database within the device. The list of trusted devices is managed using commands AT+BTT?, AT+BTT and AT+BTD. The AT+BTT? command produces a list of trusted Bluetooth addresses (link key is NEVER displayed) and AT+BTT is used to store the cached link key. The command AT+BTD123456789012 is used to remove the specified device from the database.

The “OK” response is sent immediately on receipt of the AT+BTW command. On pairing completion, an unsolicited message will be sent to the host which will be in the form PAIR n <bd_addr>. See section 3.7 for more details.

If AT+BTI or AT+BTP or AT+BTG or AT+BTQ or ATD is issued between the AT+BTW command and the subsequence PAIR asynchronous response, then an ERROR response will be sent to those commands as the device is not in a mode from where such commands can be actioned.

Response: <cr,lf>**OK**<cr,lf>

2.2.58 AT+BTW? {List Cached Trusted Device}

This command is used to list the cached trusted device.

Response: <cr,lf>**12346789012**
<cr,lf>**OK**<cr,lf>

If the cache is empty the response is as follows.

Response: <cr,lf>**OK**<cr,lf>

2.2.59 AT+BTX {Disable Page/Inquiry Scanning}

Disable page/inquiry scanning. This means it will not accept incoming connections or inquiry requests. In fact this negates the effect of AT+BTQ, AT+BTG and AT+BTP commands.

Response: <cr,lf>**OK**<cr,lf>

3. Unsolicited Responses

The 'AT' Protocol is a command/response type of protocol. This means that the blu^{2i} device will normally only respond to AT commands.

Under special circumstances, unsolicited responses will be sent to the host. They are described in the following subsections.

3.1 RING

This string is sent to the host when a remote device is initiating a serial port connection. The fully qualified string is in the form RING 012345678901 where 012345678901 is a 12 digit hexadecimal number which corresponds to the remote device's Bluetooth address. This response is sent to the host every 2 seconds until the host either accepts the connection using the ATA command or rejects it using the ATH command.

3.2 PIN?

This response is sent to the host during a pairing negotiation.

The fully qualified string is PIN? 012345678901 where 012345678901 is the Bluetooth address of the peer device. In response, the host must supply a pin code which is entered using the AT+BTK command.

If the peer address does not supply the address in the message exchange, then the address is specified as 000000000000 – and the paring will proceed as normal.

3.3 AUDIO ON

This response is sent to the host when a SCO channel has been established.

3.4 AUDIO OFF

This response is sent to the host when an existing SCO channel has been closed.

3.5 AUDIO FAIL

This response is sent to the host when a SCO channel setup fails.

3.6 ERROR 27

This response is sent to the host on power up if the firmware is unlicensed.

3.7 PAIR n <bd_addr>

This response is sent to the host on termination of a pairing process. If pairing was successful then 'n' = 0, if a timeout occurred then 'n'=1 and for all other unsuccessful outcomes the value will be 2.
The parameter <bd_addr> is the address of the peer device if available.

3.8 PAIR 0 <bd_addr> MM

This response is sent to the host on termination of a successful pairing process. The optional MM is sent only if S Register 538 is set to 1 to automatically save the link key. The value MM indicates the result of the save operation and a value of 00 implies success, otherwise the value corresponds to an error code.

3.9 RX<string>

This response is sent to the host when the unit is in online-command mode and S Register 531 is set to 3 and data arrives from a peer.

If the data from the string contains non-visual characters (for example ASCII 0 to 31 and ASCII 128 to 255), then those characters are translated into a 3 character escape sequence starting with '\'. For example and embedded <cr><lf> sequence would be sent as the 6 character string \0D\0A.

If the data contains the character '"' then it is sent as \22.
If the data contains the character '\' then it is sent as \5C

4. Incoming Connections

The blu^{2i} device can be configured using the AT+BTP or AT+BTG command so that it will scan for incoming connections from other Bluetooth devices. It can also be configured via S Register 512 to be in this mode by default on power up.

When the lower layers detect an incoming call, a **RING 123456789012** string is sent to the host every second. The command ATA is used to accept the connection and ATH to reject it.

On connection, if the S0 Register is >=0 then confirmation to the host is in the form:-

CONNECT 123456789012
CONNECT 123456789012 A
CONNECT 123456789012 E
CONNECT 123456789012 AE

Where ‘A’ means authenticated connection and ‘E’ means encryption has been enabled.

When S0 register is -1, neither RING nor CONNECT is sent to the host and the connection is silently accepted.

If the S 100 register is non-zero, then after the ring indications specified by this register have been sent to the host, and the host has failed to accept or reject the incoming connection, then an automatic ‘hangup’ is initiated.

5. Dropping Connections

In a conventional telephony modem, a call is normally terminated by first sending a +++ character sequence enveloped by an escape sequence guard time (of the order of 100 to 1000 milliseconds) to enter local command and connected mode and then the ATH command.

This device will provide a couple of ways of dropping a connection. One method will be similar to the above, but instead a ^^^ character sequence is used, this is to eliminate ambiguity when a data call is in progress via a mobile phone which was established using the mobile phone's Bluetooth AT modem. The second method will involve the host dropping the DTR (DSR from the module's viewpoint) handshaking line.

*Being able to drop a connection using the escape sequence ^^^ has a **severe** penalty on data throughput, in fact, the data rate is of the order of 85kbps instead of about 200kbps. To cater for this performance hit, the device's connection drop capability is configurable to be in one of two modes.*

One mode allows for a connection to be dropped using either method, and the other mode allows for a connection drop using the DTR method only. By default, the device is in former mode. This mode is selected using the S507 register. See S register table described in an earlier section.

To reiterate, the escape sequence is as follows:-

<Guard time><Esc Chr><Guard time><Esc Chr><Guard time><Esc Chr><Guard time>

This means that even when a file transfer is occurring and it happens to be full of <Esc Chr> characters then it is not going to drop into command mode because, when transferring a file it is going to happen as fast as possible and so the inter character gap is going to be significantly shorter than the <Guard time>.

The <Esc Chr> character can be changed via the S2 register and the <Guard time> interval can be specified via the S12 register.

6. Pairing and Trusted Devices

When authentication is enabled via S register 500 or when using the 'u' modifier in the ATD and AT+BTP commands, a connection attempt will require a link key for the peer device. The link key can be obtained prior to connection by invoking the AT+BTW and AT+BTK commands. A new link key can be obtained as often as required and is stored in a volatile cache. At any time, this cached link key can be added to the trusted devices database using the AT+BTT command. A trusted device can be deleted using the AT+BTD command. To view a list of trusted device issue the command AT+BTT?.

In addition, if S Register 538 is set to 1, then on a successful pairing, the link key will be automatically saved to the trusted device database. In that case, the asynchronous message PAIR 0 <bd_addr> has an error code appended at the end to convey the result of the save operation.

When a connection attempt requires a link key, the trusted device database will be searched automatically and if one exists will be provided without host interaction. If the link key is not present, then the connection attempt will be terminated and a NO CARRIER response will be given to the ATD command.

A typical session to pair, say an Ericsson T68i, to a serial module would be …

- Make the T68i discoverable and send AT+BTI to the serial module. This will result in inquiry responses from all devices. Make a note of the Bluetooth address of the phone e.g. 123456789012
- On the T68i start pairing procedure by selecting "Phone accepts" in the relevant Bluetooth menu.
- Send command AT+BTW123456789012 to the serial module
- Confirm that you get an OK response and then PIN? responses on a 2 second interval.
- Enter a pin code on the phone. Say it is 1234
- Then enter the command AT+BTK="1234".
- The phone will confirm success and likewise the serial module will respond with OK
- On success the serial module will send an unsolicited message in the form of PAIR 0 <bd_addr>
- Send AT+BTT to the serial module so that the pairing information is stored in the non-volatile database.
- Confirm that the link key has been stored by sending the command AT+BTT?. This will result in a list of all devices paired with the module.

If two blu^{2i} devices need to be paired, then it can be accomplished as follows:-

- To device 1 send ATI4, it will respond with the local Bluetooth address. E.g. 123456789001
- To device 1 send AT+BTP. It will become discoverable and connectable.
- To device 2 send AT+BTW123456789001 and it will respond with OK
- Then on both devices you will see PIN? asynchronous responses

- To both modules send AT+BTK=”1234”
- On success the serial module will send an unsolicited message in the form of PAIR 0 <bd_addr>
- The pairing link key, is at this stage, in volatile memory, so send AT+BTT to both.
- The two units now have pairing information which will survive a power cycle.

7. Error Responses

All error responses from the blu^{2i} device will be in the form <cr,lf>**ERROR nn**<cr,lf>, where nn will be a number in the range 00 to 99.

Error	Description
01	Register not recognised
02	Value for register is out of range
03	Incoming call NOT pending
04	No call to connect to. This error code has meaning for ATO only
05	Syntax Error
06	Empty String
06	Device Class could not be stored
08	Invalid Device Class Code
09	Invalid Bluetooth Address
10	Could not set Service or Friendly name
11	PS Store Write
12	PS Store Read
13	Not Idle
14	Incorrect Mode
15	Already Scanning
16	Pairing is already in progress
17	Not USED
18	Not USED
19	Not USED
20	Not safe to write to Non-volatile Store - Ongoing Bluetooth Connection
21	Link Key Cache is Empty
22	Link Key Database is Full
23	Malloc returned NULL - Resource Issue
24	Remote Address same as Local Address
25	Connection Setup Fail, DSR Not asserted
26	Unauthenticated licence
27	Max Responses (See S Register 518) too high. Memory allocation error
28	The length of Pin in AT+BTK is too long
29	Invalid Ring count specified for S Register 0 or 100. If S0<>0 and S100<>0 then S0 must be < S100
30	ADC Error
31	Analogue Value cannot be read as it is set for output
32	Analogue Value cannot be written as it is set for input
33	S Register Value is invalid
34	Both L and R modifier cannot be specified in ATD command
35	Invalid Major Device Class – valid value in range 0x00 to 0x1F inclusive
36	Pairing in progress – Command cannot be actioned – try again later
37	Invalid Sniff parameter specified. E.g. new Attempt value greater than MinInterval. Solution is to first increase MinInterval and re-enter the Attempt value.

8. Factory Default Mode

blu^{2i} devices are capable of operating at a very wide range of baud rates. S Registers 520 and 521 allow the baud rate to be set very easily. The baud rate clock generator in the blu^{2i} device is more versatile that that available in a standard 16550 UART commonly available in PCs.

In fact, as long as the equation BAUDRATE * 0.004096 produces an integer value, then there will be 0% error in clocking for that baud rate.

So it is possible to set a baud rate that a PC cannot cope with, and in that circumstance it is virtually impossible to communicate with it.

To cater for this circumstance, the blu^{2i} device will come out of reset using 9600,N,8,1 comms settings for exactly 750 milliseconds and then revert to the comms parameters as per the S Registers.

If the host sends the string **!<BISM>!<cr>** where <cr> is the carriage return character within that 750ms period, then the module will remain at 9600,N,8,1 and will also configure itself using factory default S Register values.