

Bluetooth Serial Module AT Command Set

Prepared by :	Mahendra Tailor	Authorised by:	
Signature :		Signature :	

© 2003 COPYRIGHT TDK Systems Europe Ltd

This document is issued by TDK Systems Europe Limited (hereinafter called TDK) in confidence, and is not to be reproduced in whole or in part without the prior written permission of TDK. The information contained herein is the property of TDK and is to be used only for the purpose for which it is submitted and is not to be released in whole or in part without the prior written permission of TDK.

Change History

Issue	Description	Author	Date
1.0	First Draft	M Tailor	4-Oct-2002
1.1	*) Added S registers 504,530,531 *) Added commands AT+BTR & AT+BTA *) Audio capability *) Added Chapter "Asynchronous Responses"	M Tailor	16-Oct-02
1.2	*) Added S Registers 533 and 534	M Tailor	31-Oct-02
1.3	*) Updated Pin Designation on 40 way connector	M Tailor	1 Nov 02
1.4	*) Added S Register 535	M Tailor	12 Nov 02
1.5	*) Error in AT Command sequence in the pure cable replacement chapter corrected *) Removed paragraph regarding unilateral acceptance of incoming call – not an issue anymore *) Added the "Frequently Asked Questions" Chapter *) Added NickH's comments	M Tailor	6 Dec 02
1.6	*) Updated the "pure cable replacement" chapter *) Added the AT&F* command for completely erasing the nvram	M Tailor	9 Dec 02
1.7	*) AT+BTI now returns ERROR 27 if S Reg 518 is set too high *) ERROR 17,18 and 19 no longer reported *) New Asynchronous response => PAIR n <bd_addr> *) Version = 0.3.6 *) S Reg 533 now takes value 0 to 2 inclusive *) Added S Reg 525	M Tailor	2 Jan 03
1.8	*) Added the Motherboard Chapter	M Tailor	14 Jan 03
1.9	*) Corrected details for SReg 508 to 511 inclusive *) AT&F command modified *) Added Power Consumption Chapter	M Tailor	3 Feb 03

Contents

1.	Introduction	5
2.	AT Command Set	6
2.1	Assumptions	6
2.2	Commands	6
2.2.1	AT	6
2.2.2	ATZ {Hardware Reset}	6
2.2.3	ATEn {Enable/Disable Echoes}	7
2.2.4	AT&Fn {Set S Reg Defaults}	7
2.2.4	AT&F* {Clear NonVolatile Storage}	7
2.2.5	AT&W {Write S Regs to Non-Volatile Memory}	7
2.2.6	ATIn {Information}	8
2.2.7	ATSn=m {Set S Register}	9
2.2.6	Future S Registers	10
2.2.8	ATSn? {Query S Register}	11
2.2.9	ATD<bd_addr> {Make Connection}	11
2.2.10	ATDU<bd_addr> {Make Authenticated Connection}	11
2.2.11	ATDY<bd_addr> {Make Encrypted Connection}	11
2.2.12	ATDUY<bd_addr> {Make Encrypted/Authenticated Connection}	11
2.2.13	ATDYU<bd_addr> {Make Encrypted/Authenticated Connection}	11
2.2.14	+++ {Enter Command Mode}	11
2.2.15	^^^ {Enter Command Mode}	12
2.2.16	ATA {Answer Call}	12
2.2.17	ATH {Drop Connection}	12
2.2.18	ATO {Enter Online Mode} (letter 'o')	12
2.2.19	AT+BTO<devclass> {Open and make Unit Detectable}	12
2.2.20	AT+BTI<devclass> {Inquire}	12
2.2.21	AT+BTIV<devclass> {Inquire}	13
2.2.22	AT+BTIN<devclass> {Inquire}	13
2.2.23	AT+BTX {Disable Page/Inquiry Scanning}	13
2.2.24	AT+BTQ {Enable Inquiry Scans ONLY}	13
2.2.25	AT+BTG {Enable Promiscuous Page Scanning ONLY}	13
2.2.26	AT+BTGU {Enable Promiscuous Page Scanning ONLY}	13
2.2.27	AT+BTGY {Enable Promiscuous Page Scanning ONLY}	14
2.2.28	AT+BTGUY {Enable Promiscuous Page Scanning ONLY}	14
2.2.29	AT+BTGYU {Enable Promiscuous Page Scanning ONLY}	14
2.2.30	AT+BTG<bd_addr> {Enable Cautious Page Scanning ONLY}	14
2.2.31	AT+BTP {Enable Promiscuous Page Scanning + Inquiry Scans}	14
2.2.32	AT+BTPU {Enable Promiscuous Page Scanning + Inquiry Scans}	14
2.2.33	AT+BTPY {Enable Promiscuous Page Scanning + Inquiry Scans}	15
2.2.34	AT+BTPUY {Enable Promiscuous Page Scanning + Inquiry Scans}	15
2.2.35	AT+BTPYU {Enable Promiscuous Page Scanning + Inquiry Scans}	15
2.2.36	AT+BTP<bd_addr> {Enable Cautious Page Scanning + Inq Scans}	15
2.2.37	AT+BTN=<string> {Set Friendly Name in NonVol Mem}	15
2.2.38	AT+BTN? {Read Friendly Name from NonVol Mem}	15
2.2.39	AT+BTF=<string> {Set Friendly Name}	15
2.2.40	AT+BTS=<string> {Set Service Name}	16
2.2.41	AT+BTS? {Read Service Name from NonVol Mem}	16
2.2.42	AT+BTW<bd_addr> {Initiate pairing}	16
2.2.43	AT+BTK=<string> {Set Passkey}	16
2.2.44	AT+BTT {Add Trusted Device}	16
2.2.45	AT+BTT? {List Trusted Devices}	17
2.2.46	AT+BTW? {List Cached Trusted Device}	17

2.2.47	AT+BTD<bd_addr> {Remove Trusted Device}	17
2.2.48	AT+BTD* {Remove All Trusted Device}	17
2.2.50	AT+BTR<bd_addr> {Set Peer Address}	17
2.2.49	AT+BTR? {Display Peer Address}	18
2.2.51	AT+BTAn {Control Audio Channel}	18
3.	Unsolicited Responses	19
3.1	RING	19
3.2	PIN?	19
3.3	AUDIO ON	19
3.4	AUDIO OFF	19
3.5	AUDIO FAIL	19
3.6	ERROR 27	19
3.7	PAIR n <bd_addr>	19
4.	Incoming Connections	20
5.	Dropping Connections	21
6.	Pairing and Trusted Devices	22
7.	Error Responses	23
8.	Pin Out	24
9.	RS232 Signal Pin Mapping	25
10.	Power Consumption and Reset	26
10.1	Reset	26
10.2	Power Consumption	26
11.	Pure Cable Replacement Mode	28
11.1	Data Cable	28
11.2	Audio Cable	29
12.	Frequently Asked Questions	30
13.	Development Motherboard	32
14.	Use Cases	35
14.1	Example 1: Light Switch Controller	35
14.1.1	Background	35
14.1.2	AT Command Sequence	35
14.2	Example 2: Remote Data Logger	36
14.2.1	Background	36
14.2.2	AT Command Sequence	36
14.3	Example 3: Trusted Device Management	37
14.3.1	Background	37
14.3.2	Obtain a new Link Key	37
14.3.3	Store link key in trusted device database	37
14.3.4	List trusted device database	37
14.3.5	Store link key in trusted device database (full)	38

1. Introduction

This document describes the protocol used to control and configure TDK's Bluetooth Serial Module. 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 module 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, S Register maintenance etc.

Just like telephony modems, the serial module powers up into an unconnected state and will only respond via the serial interface. In this state the module 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 have neither.

The device has a serial interface which can be configuration for baudrates 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 Online mode. This leads to a 2x2 matrix of four states, as shown below.

	RF Unconnected	RF Connected
Command Mode	OK	OK
Online Mode	ILLEGAL	OK

One combination, 'RF unconnected' and 'Online Mode' does 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 the TDK Modules 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 for use with this serial module.

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 module 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 +BTx, 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. Eg the value 1234 can also be entered as \$4D2
- <string>** A string delimited by double quotes. Eg "Hello World". The " character MUST be supplied as delimiters.

2.2.1 AT

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

2.2.2 ATZ {Hardware Reset}

Forces the device through a hardware reset which means it will eventually come alive in the Command mode and Bluetooth 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 surest 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

2.2.3 ATEn {Enable/Disble Echoes}

It affects S Register 506. Set n=0 to disable echoes and n=1 to enable. All other values will generate an error.

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

2.2.4 AT&Fn {Set S Reg Defaults}

This command will only work when the device is in idle mode. Depending on the value of 'n' it installs S Reg 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.

'n'	Description
0 (Default)	Medium power consumption, UART baudrate unchanged, Left LED off, Right Led = DCD
1	Minimum power consumption, UART baudrate set to 9600, Both LEDs off
2	Minimum power consumption, UART baudrate set to 38400, Both LEDs off
3	Minimum power consumption, UART baudrate set to 115200, Both LEDs off
4	Medium power consumption, UART baudrate set to 115200, Left LED off, Right Led = DCD
5	Maximum power consumption, UART baudrate set to 115200, Left LED=DSR, Right Led = DCD
6	Maximum power consumption, UART baudrate set to 460800, Left LED=DSR, Right Led = DCD

If 'n' is not specified then a default value of 0 is assumed where the baudrate is NOT changed.

Please refer to the "Power Consumption" chapter further on, for a more detailed discussion of power usage.

The new values are NOT updated in nvram until the AT&W command is sent to the module

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

Or

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

2.2.4 AT&F* {Clear NonVolatile Storage}

This command will only work when the device is in idle mode. It installs values in S registers as per command AT&F4 and then all other user parameters in non-volatile storage is 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>

2.2.5 AT&W {Write S Regs to Non-Volatile Memory}

Writes current S Register values to non-volatile memory so that they are retained over a power cycle. This command only works if not in ONLINE_COMMAND mode. This is due to the fact that writing to non-volatile storage is not allowed while a Bluetooth connection is active.

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

Or

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

2.2.6 ATIn {Information}

This will return information about the module. For example, version numbers, manufacturer etc

For n=0, the response describes the product.

Response: <cr,lf>**TDK Serial Module**<cr,lf>**OK**<cr,lf>

For n=1, the CSR firmware build number is returned, for example

Response: <cr,lf>**520**<cr,lf>**OK**<cr,lf>

For n=2, the TDK Image number is returned, for example

Response: <cr,lf>**06G**<cr,lf>**OK**<cr,lf>

A letter suffix implies an engineering build for internal use.

For n=3, the version number of the vm app is returned, for example

Response: <cr,lf>**1.01**<cr,lf>**OK**<cr,lf>

For n=4, a 12 digit hex number corresponding to the Bluetooth address of the device is returned

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

For n=5, the Manufacturer of this device is returned

Response: <cr,lf>**TDK Systems Europe Ltd, UK**<cr,lf>**OK**<cr,lf>

For n=6, the maximum size of trusted device database is returned

Response: <cr,lf>**4**<cr,lf>**OK**<cr,lf>

For n=7, the manufacturer of the Bluetooth chipset is returned

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

For n=8, the chipset designation is returned

Response: <cr,lf>**BC2-EXT**<cr,lf>**OK**<cr,lf>

2.2.7 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 hex number.

When S register values are changed, the changes are **not** stored in non-volatile storage 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 storage when the command is received.

Register 'n'	Default	Range	Comment
0	1	-1..15	Number of RING indication before auto answering an inbound 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.
2	^	0..255	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
12	100	50..5000	Escape sequence guard time in milliseconds.
500	0	0..1	Default Authentication for outgoing connections, Set to 1 to Enable Authentication
501	0	0..1	Default Encryption for outgoing connections, Set to 1 to Enable Encryption
502	0	0..1	Default Authentication for incoming connections, Set to 1 to Enable Authentication
503	0	0..1	Default Encryption for incoming connections, Set to 1 to Enable Encryption
504	0	0..1	Set to 1 for suppression of CONNECT message to host for outbound connections
505	5	2..120	Delay before abandoning connection attempt as a master. Referenced by the ATD. In units of seconds
506	1	0..1	Enable/Disable echoes. The ATEn command also affects this.
507	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 be dropped using a deassertion of DSR. Mode 2 provides for the highest data transfer rate.
508	640	10..2550	Page Scan Interval in units millisecond (10ms will give 11.25ms)
509	320	10..2550	Page Scan Window in units millisecond (10ms will give 11.25ms)
510	640	10..2550	Inquiry Scan Interval in units of millisecond (10ms will give 11.25ms)
511	320	10..2550	Inquiry Scan Window in units of millisecond (10ms will give 11.25ms)
512	1	0..4	Specify power up state. If set to 0 then 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 scan for inquiries only, similar to issuing AT+BTQ. When set to 3, it will page for a connection but not be discoverable eg AT+BTG and finally when set to 4, it will be like AT+BTP. 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.
513	1	0..1	Pairing Authentication, 1 = Enable
514	10	1..60	Pairing Timeout in seconds
515	0x000000	0..0xFFFFFFFF	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 hex number.
516	0xFFFFFFFF	0..0xFFFFFFFF	Default Device Class Code to be used with AT+BTI when it is not explicitly specified. When queried the value is always printed as a hex number.
517	20	2..61	Inquiry Length in units of seconds. This parameter is referenced by the AT+BTI command
518	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.
519	500	100..6000	When S507>0, and in a connection, DSR can be used to change from online 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
520	115200	300..921600	Change baudrate. The effect is immediate and in fact the OK will be sent at the new baudrate. Due to the integer arithmetic involved, the actual baudrate is different. To inspect the actual baudrate, do AT\$521? AT\$520? will return the baudrate to the nearest 100.

			If Sreg 525=1, then the max baudrate is limited to 115200
521	115200	300.. 921600	Change baudrate. The effect is immediate and in fact the OK will be sent at the new baudrate. Due to the integer arithmetic involved, the actual baudrate is different. To inspect the actual baudrate, do ATS521? If Sreg 525=1, then the max baudrate is limited to 115200
522	1	1	1 = CTS/RTS hardware handshaking enabled
523	1	1..2	Number of StopBits
524	0	0..2	Parity. 0=None, 1=Odd, 2=Even
525	See Comment	0..1	Apply multiplier of 8 to baudrate internally. This is set to 0 (disabled) by default for the serial module, and set to 1 (enabled) by default for the pc card variant. 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 baudrate, in reality it gets a baudrate which is 8 times faster. If S Reg 520 > 115200 then this register cannot be set to 1.
530	1000	100.. 15000	Reconnect delay when configured as master in pure-cable-replacement mode. This value is rounded down to the nearest 100ms
531	0	0..1	If Set to 1 then on connection, remain in Command Mode.
532	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
533	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 store using AT&W and will operate as expected after an ATZ or a power cycle.
534	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.
535	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
540	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.
1001 to 1010		0.. 2^32	10 General Purpose 32 bit Registers for use by host. These are stored in non-volatile memory.
610	0	0..31	Set direction of digital i/o lines
620	n/a	0..31	Read/Write to all 8 Digital lines in one atomic step. The value is returned as a 4 digit hex value with trailing 0s.
621	n/a	0..1	Read/Write to GPIO1
622	n/a	0..1	Read/Write to GPIO2
623	n/a	0..1	Read/Write to GPIO3
624	n/a	0..1	Read/Write to GPIO4
625	n/a	0..1	Read/Write to GPIO5
626	n/a	0	Read/Write to GPIO6 - Not Available so 0 always returned
627	n/a	0	Read/Write to GPIO7 - Not Available so 0 always returned
628	n/a	0	Read/Write to GPIO8 - Not Available so 0 always returned

2.2.6 Future S Registers

The following registers are planned for the future.

Register	Default	Range	Comment
601	n/a	TBD	Read/Write to Analogue Line 1
602	n/a	TBD	Read/Write to Analogue Line 2
603	0	0..1	Set direction of Analogue Line 1
604	0	0..1	Set direction of Analogue Line 2

Response: <cr,>OK<cr,>

If the register is not recognised or the value specified is out of range then

Response: <cr,>ERROR nn<cr,>

2.2.8 ATSn? {Query S Register}

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>

2.2.9 ATD<bd_addr> {Make Connection}

Make a connection to device with Bluetooth address <bd_addr>. 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>

2.2.10 ATDU<bd_addr> {Make Authenticated Connection}

Make an authenticated connection to device with Bluetooth address <bd_addr>. Encryption is disabled.

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

Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.11 ATDY<bd_addr> {Make Encrypted Connection}

Make an encrypted connection to device with Bluetooth address <bd_addr>. Authentication is disabled

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

Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.12 ATDUY<bd_addr> {Make Encrypted/Authenticated Connection}

Make an encrypted and authenticated connection to device with Bluetooth address <bd_addr>.

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

Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.13 ATDYU<bd_addr> {Make Encrypted/Authenticated Connection}

Make an encrypted and authenticated connection to device with Bluetooth address <bd_addr>.

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

Or <cr,lf>**NO CARRIER**<cr,lf>

2.2.14 +++ {Enter Command Mode}

Usually this is the escape sequence for forcing a modem into command mode when in online state. Since this device could be used to connect to a mobile phone modem over a Bluetooth serial port profile, it could confuse matters considerably, therefore the sequence ^^^ is used as described next.

2.2.15 ^^^ {Enter Command Mode}

When in online 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.

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

2.2.16 ATA {Answer Call}

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

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

2.2.17 ATH {Drop Connection}

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

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

2.2.18 ATO {Enter Online Mode} (letter 'o')

Return to online data mode. Assume online after OK is received. Will repond with ERROR if there is no Bluetooth connection

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

or

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

2.2.19 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 a device class code mask and is a fixed length hex digit optional string. 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.

2.2.20 AT+BTI<devclass> {Inquire}

This will make the device perform an inquiry for device class code **cccccc** 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> value specifies a device class code and is a fixed length hex digit optional string. If it is not specified, then the device class code is taken from S register 516. <DevClass> must be exactly 6 hex characters long.

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

2.2.21 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>

2.2.22 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>.

Response: <cr,<lf>12346789012,123456,"TDK AT DONGLE 1"
 <cr,<lf>12345678914,123456,"TDK AT DON7GLE 3"
 <cr,<lf>OK<cr,<lf>

2.2.23 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>

2.2.24 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>

2.2.25 AT+BTG {Enable Promiscuous Page Scanning ONLY}

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

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

2.2.26 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>

2.2.27 AT+BTGY {Enable Promiscuous Page Scanning ONLY}

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

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

2.2.28 AT+BTGUY {Enable Promiscuous Page Scanning ONLY}

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

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

2.2.29 AT+BTGYU {Enable Promiscuous Page Scanning ONLY}

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

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

2.2.30 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>

2.2.31 AT+BTP {Enable Promiscuous Page Scanning + Inquiry Scans}

Enable page scanning and wait for a connection from any device. Inquiry scanning is also enabled. Authentication and Encyprion is as per defaults in S registers

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

2.2.32 AT+BTPU {Enable Promiscuous Page Scanning + Inquiry Scans}

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

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

2.2.33 AT+BTPY {Enable Promiscuous Page Scanning + Inquiry Scans}

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

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

2.2.34 AT+BTPUY {Enable Promiscuous Page Scanning + Inquiry Scans}

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

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

2.2.35 AT+BTPYU {Enable Promiscuous Page Scanning + Inquiry Scans}

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

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

2.2.36 AT+BTP<bd_addr> {Enable Cautious Page Scanning + Inq Scans}

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>

2.2.37 AT+BTN=<string> {Set Friendly Name in NonVol Mem}

This sets the default friendly name of this device as seen by other devices. It will be stored in a non volatile storage area. Use AT+BTF to make the name visible to other devices. Use AT+BTN? to read it back

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

2.2.38 AT+BTN? {Read Friendly Name from NonVol Mem}

Reads the default friendly name from non-volatile store.

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

2.2.39 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>

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

This writes the name to non-volatile storage area. 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 store.

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

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

2.2.41 AT+BTS? {Read Service Name from NonVol Mem}

Reads the default fservice name from non-volatile store.

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

2.2.42 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 detailed next.

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+BTB. 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+BTB123456789012 is used to remove the specified device from the data base.

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.

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

2.2.43 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 store for future use. Specifying an empty string deletes the key from the non-volatile store. The string length must be in the range 0 to 8, otherwise an error will be returned.

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

2.2.44 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: <lfcrlf>**OK**<cr,lf> or <lfcrlf>**ERROR**<cr,lf>

2.2.45 AT+BTT? {List Trusted Devices}

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 AT+I6 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>

2.2.46 AT+BTW? {List Cached Trusted Device}

This command is used to list the cached trusted device and the response is as follows.

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

If the cache is empty the the response is as follows.

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

2.2.47 AT+BTDBD<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: <lfc<cr>OK<cr,<lf>

2.2.48 AT+BTDD* {Remove All Trusted Device}

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 !!!

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

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

This command is used to store a peer address in non-volatile storage. 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.

If S register 512 = 3 or 4 then it will wait for an inbound 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: <lfc<cr>OK<cr,<lf>

2.2.49 AT+BTR? {Display Peer Address}

This command is used to display the peer address stored in non-volatile storage, 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 the response is as follows.

Response: <cr,>lf>**000000000000**
 <cr,>lf>**OK**<cr,>lf>

2.2.51 AT+BTAn {Control Audio Channel}

Once a Bluetooth connection is active, **and assuming the peer device is a TDK Serial Module**, 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.

The parameter is 0 to switch off the channel and it is 1 to switch it off.

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 seting 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>

3. Unsolicited Responses

The 'AT' Protocol is a command/response type of protocol. This means that the TDK Serial Module 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 hex 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 pairing 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.

4. Incoming Connections

The serial module can be configured using the AT+BTP or AT+BTG command so that it will scan for incoming connections from other Bluetooth devices.

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.

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) 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+BDT command. To view a list of trusted device issue the command AT+BTT?.

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 key 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 serial modules need to be paired, then it can be accomplished as follows:-

- To module 1 send AT+I4, it will respond with the local Bluetooth address. Eg 123456789001
- To module 1 send AT+BTP. It will become discoverable and connectable.
- To module 2 send AT+BTW123456789001 and it will respond with OK
- Then on both modules 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 device will be in the form <cr,<lf>**ERROR nn**<cr,<lf>, where nn will be a number in the range 00 to 99. There shall be leading 0s if the error value is less than 10.

The error values shall be designated as follows:-

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 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	ATD Remote Address same as Local Address
25	Connection Setup Fail, DSR Not asserted
26	Unauthenticated licence
27	Max Responses (See S Reg 518) too high. Memory allocation error
28	The length of Pin in AT+BTK is too long

8. Pin Out

The TDK Bluetooth Serial Module is physically accessed via a 40 way mini-pci connector. The pin allocation is as follows:-

PinNo	Signal	Description	PinNo	Signal	Description
1	RESERVED	Do Not Connect	2	GPIO1	I/O for Host.
3	RESERVED	Do Not Connect	4	GPIO2	I/O for Host
5	SPI_MOSI	SPI bus serial in I/P	6	UART_RI	Output
7	SPI_CSB	SPI bus chip select I/P	8	UART_DCD	Output
9	SPI_CLK	SPI bus clock I/P	10	UART_DSR	Input
11	GND		12	GPIO3	I/O for Host
13	RESET	Reset I/P	14	GPIO4	I/O for Host (Right LED)
15	GND		16	GPIO5	I/O for Host (Left LED)
17	SPI_MISO	SPI bus serial out O/P	18	GND	
19	UART_CTS	Clear to Send I/P	20	PCM_CLK	PCM Clock I/P
21	UART_TX	Transmit Data O/P	22	PCM_IN	PCM Data I/P
23	UART_RTS	Request to Send O/P	24	PCM_SYNC	PCM Sync I/P
25	UART_RX	Receive Data I/P	26	PCM_OUT	PCM Data O/P
27	VCC_3V3	3.3V Output	28	N/C	
29	USB_5V	3.6V < VIN < 6.0V	30	GND	
31	N/C		32	RESERVED	Do Not Connect
33	N/C		34	RESERVED	Do Not Connect
35	GND		36	GND	
37	N/C		38	GND	
39	VCC_1V8	1.8V Output	40	N/C	

Notes:

- UART_RX, UART_TX, UART_CTS, UART_RTS, UART_RI, UART_DCD, UART_DSR are all 3.3v level logic. For example, when RX and TX are idle they will be sitting at 3.3V. Conversely for handshaking pins CTS, RTS, RI, DCD, DSR a 0v is treated as an assertion.
- Pin 6 (UART_RI) has inverse logic. It is normally 3.3v. When a remote device initiates a connection, this pin goes low. This means that when this pin is converted to RS232 voltage levels it will have the correct voltage level for assertion.
- Pin 8 (UART_DCD) has inverse logic. It is normally 3.3v. When a connection is live this pin is low. This means that when this pin is converted to RS232 voltage levels it will have the correct voltage level for assertion.
- Pin 10 (UART_DSR) is an input, again with inverse logic. It should be connected to the DTR output of the host. When the serial module is in high speed mode (See definition for S reg 512), this pin should be asserted by the host to ensure that the connection is maintained. A deassertion is taken to mean that the connection should be dropped, or an online command mode is being requested.
- The GPIO pins can be accessed using S Registers 621 to 625
- GPIO4 and GPIO5 are also connected to LEDs on the module. If these i/o pins are set for input, then the LED will be driven by the host and appropriate drive current requirements must be satisfied. By default GPIO4 is used to drive the right led which indicates connection status.

9. RS232 Signal Pin Mapping

Just as a telephony modem has control and status lines, the TDK serial module also provides for 3 control and status lines as per the table below. The direction column is as seen from the module's viewpoint.

Direction	Function
OUT	CI also known as RI (Ring Indicate)
OUT	DCD (Data Carrier Detect)
IN	DSR (Data Set ready)

All three lines are under program control and as such require GPIO pins. The mapping will be as per the table below

BC2_PIO Pin	Direction	Connector Pin Label	Function
0	IN/OUT	GPIO1	General Purpose I/O
1	IN/OUT	GPIO2	General Purpose I/O
2	OUT	UART_RI	Output from module
3	OUT	UART_DCD	Output from module
4	IN	UART_DSR	Input to Module
5	IN/OUT	GPIO3	General Purpose I/O
6	IN/OUT	GPIO4	General Purpose I/O (Right LED)
7	IN/OUT	GPIO5	General Purpose I/O (Left LED)

Notes:

- BC2_PIO4 (DSR) is used by the serial module to sense that the host is connected, and is intricately linked with connections. For outgoing calls, if this line is not asserted then an error is immediately. Similarly for AT+BTP and AT+BTG.
While in a call, for appropriate modes, a deassertion means fall into command state. If the deassertion exists for longer than the period specified in S register 519 then the connection is dropped as if an ATH command was received.
- BC2_PIO2 (RI), is normally deasserted. When an incoming connection is detected it will be asserted, until the connection is either answered or rejected using ATA and ATH respectively.
Unlike normal telephony modem, this line does not follow the state of RING tone. Basically it is asserted for incoming call setup to allow for a host to wakeup etc.
As this pin is connected to an LED, there will be visual indication when RING is being indicated.
- BCS_PIO3 (DCD) will be deasserted when the device is in the unconnected state. Asserted when a connection is active. **In addition since this pin also controls an LED on our device, there is visual indication of a connection.**

GPIO Pins 1,2,3,4 and 5 are available for general purpose use.

BREAK

If the host sends a break signal of duration greater than 100ms, then the serial module is configured to treat that as a signal to perform a hardware reset.

10. Power Consumption and Reset

10.1 Reset

The module can be reset by the host without the need of any i/o using a BREAK signal. The module has been configured to reset when the RX line detects a break condition for durations greater than 100 milliseconds.

10.2 Power Consumption

The current drain from the Vcc power input line is dependent on various factors. The three most significant factors are the voltage level at Vcc, UART Baudrate and the operating mode.

The hardware specification for the serial module allows for a voltage range of 3.6 to 6.0v at Vcc. Tests have shown that there is no significant difference in current draw when Vcc is 5 or 6V. Therefore the data presented below, pertains to Vcc levels of 3.6 and 5v only. Tests have shown that where power drain is an issue, it is best to keep Vcc at the lower end of the range.

The UART baudrate has a bearing on power drain because as is normal for digital electronics, the power requirements increase linearly with increasing clocking frequencies. Hence higher baudrates result in a higher current drain.

Finally with regards to operating mode the significant modes are; idle, waiting for a connection, inquiring, initiating a connection and connected. With connected mode, it is also relevant to differentiate between data being transferred and when data is being transferred at the maximum rate possible.

The operating mode can best be described by stating the AT commands required to enter that mode. In addition, there are certain S Registers which have a direct impact on power consumption, which are described next.

The serial module has 2 Leds which can be configured to display connection status. One led is used to display connection status, while the other is used to either display 'Ring Indicate' status or follow the state of the incoming DSR line on the UART interface. Tests have shown that these Leds can consume up to 5.3mA which is more than double the current draw when in Idle mode. Therefore S registers 533 and 534 can be used to completely disable these indications.

Finally S Registers 508 to 511, which specify the page and inquiry scan intervals and windows, can be used to adjust the **average** current drain when in discoverable and or connectable modes. Registers 508 and 509 specify the interval and window for page scans and registers 510 and 511 specify the interval and window for inquiry scans. Register pairs 508/509 and 510/511 describe duty cycles when the serial module goes into scan modes. It is while scanning that the highest current draw occurs. The average current draw is determined by simple arithmetic using the values stored in the 508/509 and 510/511 register pairs.

The operating modes described above are entered using AT commands as follows:-

Idle	On power up, with S Reg 512 = 1
Wait for Connection	AT+BTG (100% page scan duty cycle)
Discoverable Only	AT+BTQ (100% inquiry scan duty cycle)
Connecting	ATD
Connected	No Data

Connected

Max data transfer

All current consumption values in the table below assume that the connection status indication functionality of the Leds has been disabled by setting S Registers 533 and 534 to 0.

All current values are in milliamps.

		Baudrate			
		9600	38400	115200	460800
Current per Led	3.6v	3.20	3.20	3.20	3.20
	5.0v	5.30	5.30	5.30	5.30

		Baudrate			
		9600	38400	115200	460800
Idle Mode, S512=1	3.6v	1.60	1.80	1.96	3.00
	5.0v	2.00	2.10	2.30	3.40
Wait for Connection Or Discoverable Mode, AT+BTP S508=S510=640, S509=S511=320	3.6v	59.00	59.00	59.00	59.00
	5.0v	65.00	65.00	65.00	65.00
Wait for Connection Or Discoverable Mode, AT+BTP S508=S510=1000, S509=S511=11	3.6v	2.75	2.94	3.10	4.12
	5.0v	3.26	3.36	3.55	4.63
Inquiring Mode, AT+BTI	3.6v	50.00	50.00	50.00	50.00
	5.0v	54.00	54.00	54.00	54.00
Connecting Mode (ATDxxx)	3.6v	50.00	50.00	50.00	50.00
	5.0v	54.00	54.00	54.00	54.00
Connected Mode (No Data Transfer)	3.6v	6.00	6.10	6.40	7.20
	5.0v	7.20	7.20	7.40	8.20
Connected Mode (Max Data Transfer)	3.6v	21.50	22.50	24.50	32.50
	5.0v	24.50	26.00	28.00	36.00

As can be seen, the current drain while waiting for a connection or discoverable mode is about 30 times higher than in idle mode. This is when the page/inquiry scan duty cycle is 100%. These modes gives the quickest response to a page or inquiry request from a remote peer.

It is possible to reduce the duty cycle down to as low as 0.5% at the expense of response time. The response time can be specified via S Reg 508 and 510 for page and inquiry respectively, where the worst case response time can be as high as 2.5 seconds. Then the duty cycle can be varied by changing the value of S Regs 509 and 511 appropriately.

For example, if SReg 508 and 510 are both set to 1000ms and SReg 509 and 511 are both set to 11ms then the duty cycle is reduced to 1%, this means that average current drain at 5.0v will be 2% of 65mA plus the normal idle mode current, that is, it is as low as 2.75mA. However, in this case, it can take up to 1 second to establish a connection.

11 Pure Cable Replacement Mode

11.1 Data Cable

The module has the capability of being preset into a pure 5-wire data cable replacement mode. The 5 wires being RX, TX, CTS, RTS and GND. This mode requires no changes to a host application since the Bluetooth connection is automatically setup on power up and will retry when the connection drops.

By implication, two devices are needed to replace a cable. One device is pre-configured to always be a master and the other, a slave.

Assuming the bluetooth address of the master to be <bdaddr_m> and that of the slave to be <bdaddr_s>, the master module is configured by sending it the following AT commands.

```
AT&F
ATS512=1
ATS504=1
ATS507=2
ATS530=2000
AT&W
AT+BTR<bdaddr_s>
```

Where the ATS507=2 setting puts the device in DSR drop mode only. This means that when the device needs to be reconfigured, deasserting the DSR line will ensure that the module responds quickly to AT commands. This further means that in stand alone mode, the DSR input line MUST be asserted e.g 0v in TTL signal mode.

The slave is configured by,

```
AT&F
ATS512=4
ATS0=-1
AT&W
AT+BTR<bdaddr_m>
```

Where <bdaddr_m> is optional. If it is not specified, then the slave unit will accept connections from any device. If specified then only connections from the device specified will be accepted.

If it is desired that the slave unit not be discoverable (the master is by default not discoverable), then the configuration commands are,

```
AT&F
ATS512=3
ATS0=-1
AT&W
AT+BTR<bdaddr_m>
```

Where <bdaddr_m> is optional. If it is not specified, then the slave unit will accept connections from any device. If specified then only connections from the device specified will be accepted.

When the units are next power cycled, the slave unit will wait for the master to connect to it and the master will continually look for the slave. If a connection attempt fails, the master will wait for 2 seconds before reattempting a connection. This 2 second delay can be varied by issuing it an ATS530 command with an appropriate value in the range 100ms to 15000ms.

IMPORTANT NOTE: When S Reg 507 = 0, the DSR input to the module **MUST** be asserted for the auto connection to succeed. When operating at TTL levels a 0V is seen as an assert state. When operating at RS232 levels and voltage greater than 3V is seen as assert. It is usual to connect the DTR line of the host to the DSR line of this device.

11.2 Audio Cable

With a pair of these modules it is possible to replace a mono audio cable with two way traffic. That is, a setup where a microphone is connected to a speaker at the remote end and vice versa. So this mode effectively replaces two audio cables.

Assuming the bluetooth address of the master to be <bdaddr_m> and that of the slave to be <bdaddr_s>, the master module is configured by sending it the following AT commands.

```
AT&F
ATS512=1
ATS504=1
ATS530=2000
ATS532=1
AT&W
AT+BTR<bdaddr_s>
```

And the slave is configured by,

```
AT&F
ATS512=4
ATS0=-1
AT&W
AT+BTR<bdaddr_s>
```

12. Frequently Asked Questions

1. I know my mobile phone's Bluetooth address and have successfully paired as detailed in Chapter 6. When I try to make a connection using ATD it always fails?

Mobile phones are very security conscious, so always insist on authenticated connections. So to make a connection, use the ATDU<bd_addr> variant of the command.

2. I have paired with a mobile phone and configured the device in cable replacement mode as detailed in Chapter 10, but I keep getting "NO CARRIER" responses.

Set S register 502 to 1 so that the default for outgoing connection will ask for authenticated connections. For security reasons most phones will only allow authenticated Bluetooth connections.

3. When my module is configured for cable replacement, on connection I want to suppress the "CONNECT" response sent to the host, so that the first character my host sees is the first character from my actual data traffic.

Set S register 504 to 1

4. I cannot get data transfer rates greater than 92kbps.

You probably have the uart baudrate set to 115200. ($115200 * 8 / 10 = 92160$) where the *8/10 factor arises from the start and stop bit overhead.

Set the uart baudrate to say 230400 or 460800 via S register 520, assuming your host can cope with the higher baudrates. Setting the buadrate to 921600 will guarantee that the UART is never responsible for the bottleneck.

5. I have done as suggested in 4 above and I still don't get the 200kbps claimed in the data sheet.

Set S register 507 to 2 and ensure that the DSR input to the module is asserted. If DSR is not asserted in this mode, then all connections attempts will be aborted.

6. I modified S Registers to put the device in cable replacement mode and nothing happens after a power cycle.

Ensure that the modified S registers are stored in non-volatile storage by sending the command AT&W

7. Can I suppress character echoes when in command mode?

Yes. Set S Register 506 to 0

8. When I establish a connection, I want an SCO channel (Audio) to be started automatically.

Set S Register 532 to 1

9. After a successful connection, I want to remain in command mode so that I can send more AT commands and then when done, go to online mode by issuing the ATO command

Set S Register 531 to 1

10. When in connected and online mode, I want to drop into command mode using the more usual +++ escape sequence instead of ^^

Set S Register 2 to 43

11. When I perform an inquiry using AT+BTI I only get 8 responses, yet I know I have more than 8 devices in the neighbourhood.

Set S Register 518 to a larger value than the default 8 before issuing the AT+BTI command

12. When doing an inquiry I want to get the friendly names as well, along with the devices class codes.

Instead of the AT+BTI command, issue the variant AT+BTIN

13. When my connected device goes out of range, it takes too long for a "NO CARRIER" response from the device.

Set S Register 535 as desired. It modifies the baseband link timeout appropriately.

14. The cable between my host and the serial module has just RX,TX,GND. I get no responses from the serial module.

Pull the CTS line of the serial module to the asserted state (0V).

15. Can I disable CTS/RTS handshaking and use XON/XOFF handshaking?

No. If your host does not have CTS/RTS handshaking capability, we suggest you set the UART baudrate to something less than 38400 and link the CTS pin of the module to a permanently asserted state. The higher bit rate capability on the RF side should ensure that the receive buffer of the serial module will remain empty.

16. I have changed S Register 512 so that on power up it comes in mode 4. But nothing seems to happen, in fact after power cycle, the value is as if I had never changed it.

After modifying the S register, enter the command AT&W so that the new value is stored in non-volatile storage.

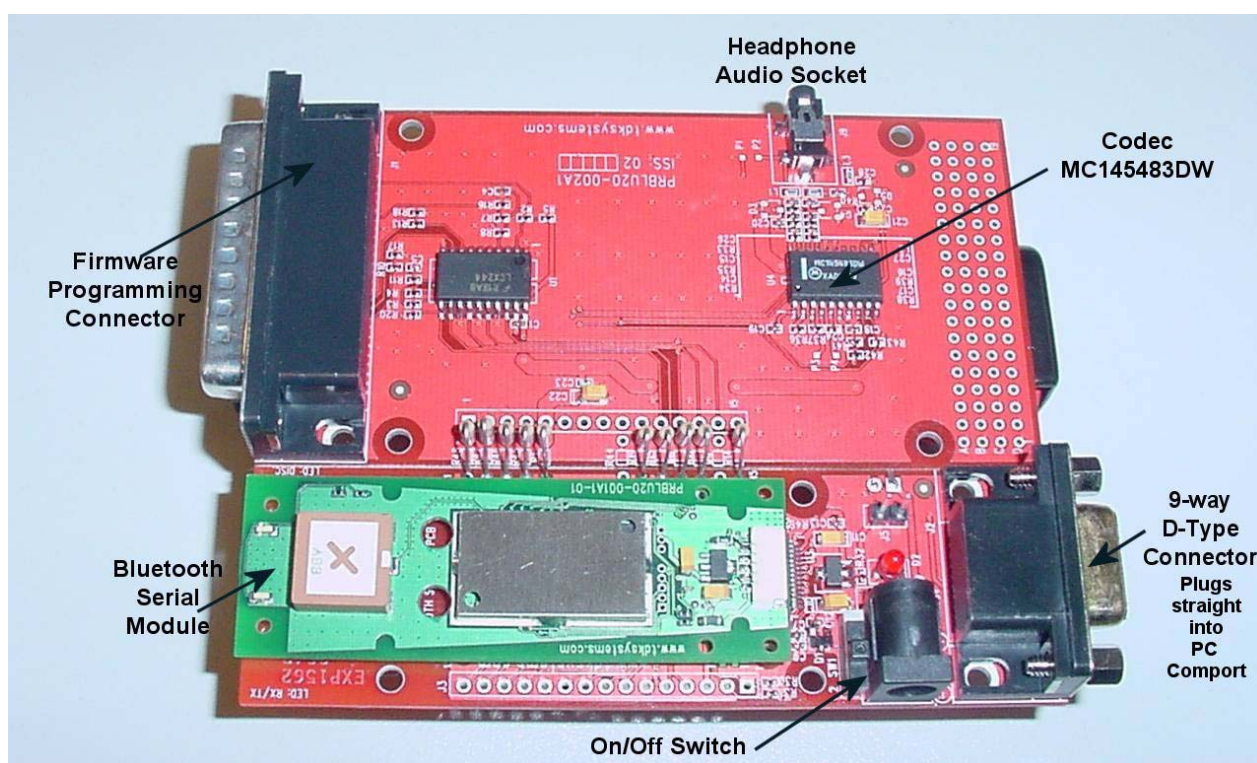
17. I have set S Reg 507 to 1 and specified a peer address using AT+BTR, but I still don't get an auto connection.

Ensure that DSR of the device is asserted (ie at 0v when operating at TTL levels or > +3V when operating at RS232 levels .

13. Development Motherboard

TDK have developed a motherboard which allows the serial module to be connected directly to a PC. This motherboard provides RS232 level conversion and a standard 9 way D-Type connector. In addition it also has a 25 way D-Type connector which gives access to the SPI bus on the serial module for firmware updates and there is a headphone socket which allows for experimentation with the serial modules audio capability.

A picture of the serial module mounted on the motherboard is presented below.



The 9 way D-Type connector is a socket, and the pin-out is such that it can be plugged straight into a PC serial port. Alternately, if access to the serial port of the PC is restricted, an extension cable with all 9 wires can be used to connect the two.

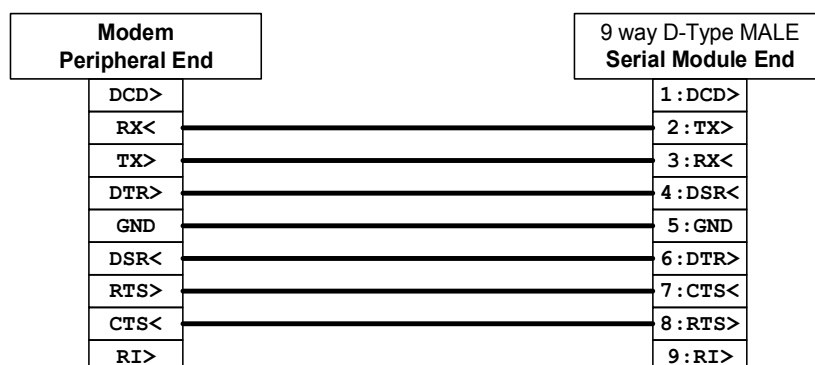
Similarly, for firmware updates, assuming the PC has been loaded with the appropriate tools (CSR's Bluelab), the 25 way D-Type connector, can be plugged straight into the LPT port of the PC.

The pin-out for the 9 way serial connector is as per the table below, and the direction is as seen from the serial modules point of view.

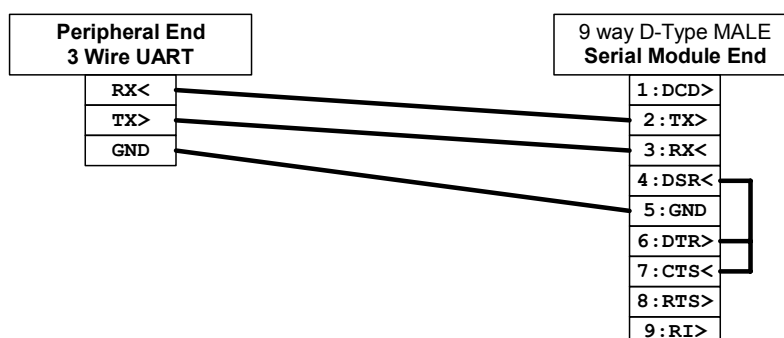
Pin No.	Description		Direction
1	Data Carrier Detect	DCD	Output
2	Transmit	TX	Output
3	Receive	RX	Input
4	Data Set Ready	DSR	Input
5	Ground	GND	---
6	Data Terminal Ready	DTR	Output
7	Clear To Send	CTS	Input
8	Ready To Send	RTS	Output
9	Ring Indicate	RI	Output

One of the main features of the TDK serial module is that it enables the replacement of a serial cable between a PC and a peripheral such as a modem.

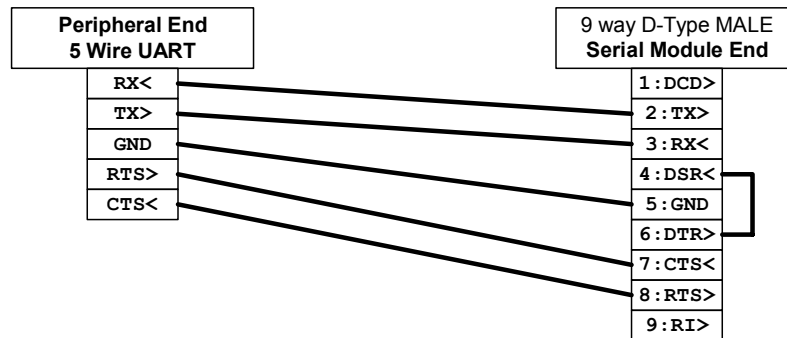
The pin-out on modems is as per the table above, so if the serial cable between a peripheral like a modem and a PC is to be replaced with 2 serial modules, then the motherboard will plug into the PC as is. However, the pin-out and the gender at the modem end are not one to one compatible with the motherboard. So to facilitate a connection, a null modem cable is required as per the diagram below



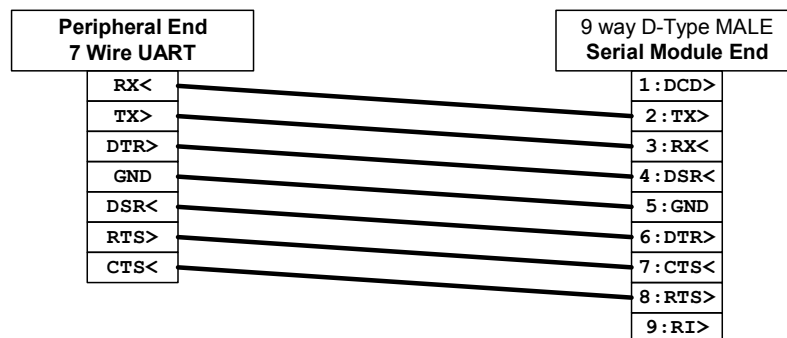
In the case of a peripheral which only has a 3 wire uart interface consisting of RX, TX, and GND. Where it relies on software handshaking, then the cabling between it and the serial module should be as shown in the diagram below. The XON/XOFF handshaking will be observed by the peer NOT the serial module.



In the case of a peripheral which has a 5 wire uart interface consisting of RX,TX, and GND and the hardware handshaking flow control lines CTS and RTS. The cabling between it and the serial module should be as shown in the diagram below. The RTS/CTS flow control will be handled by the serial module.



Finally in the case of a peripheral which has a 7 wire uart interface consisting of RX,TX, and GND and the hardware handshaking flow control lines CTS and RTS plus DSR and DTR. The cabling between it and the serial module should be as shown in the diagram below. The RTS/CTS flow control will be handled by the serial module. In this configuration, an intelligent peripheral will be able to take advantage of the full 200kbps datarate capability where it utilises a brief deassertion of DTR to change from online to online-command mode.



14. Use Cases

14.1 Example 1: Light Switch Controller

14.1.1 Background

A small microcontroller controlling a single light switch. The microcontroller's serial port is connected to the serial module and a digital output line drives a relay which controls a light. The remote host sends the text "ON" to switch on the relay and "OFF" to switch it off.

14.1.2 AT Command Sequence

AT Command	Response	Comment
ATZ	OK	Resets the device and sets all S Register values as per the values in non-volatile memory database.
ATS0=1	OK	Auto answer incoming connection after 1 ring
AT+BTP	OK	Will accept incoming connection from any device and will also be discoverable
	CONNECT 123456789012	Call has been answered automatically, and the Bluetooth address of the peer device is provided.
At this point the light switch protocol takes in commands from the remote. For example ON, OFF etc		
	NO CARRIER	If the remote end drops the connection
OR		
^^^	OK	Puts the device in to command mode
ATH	OK	Drops the call.

14.2 Example 2: Remote Data Logger

14.2.1 Background

A data logger on a remote site gathers data which is then transferred to a central site via a Bluetooth enabled phone. Assume that the remote site is accessed via the telephone 02089388609 and the PC connected to the modem logs the data to a file, so all the data logger has to do is open a connection and then send textual data corresponding to the gathered data.

Assume that the data logger knows the BT address of the phone and assume it is 123456789012

Further assume that the Bluetooth Phone exposes a serial port profile through which an AT modem can be accessed.

14.2.2 AT Command Sequence

AT Command	Response	Comment
ATZ	OK	Resets the device and sets all S Register values as per the values in non-volatile memory database.
ATD123456789012	CONNECT 123456789012	
ATZ	OK	The reponse is coming from the Bluetooth phone
ATD02089388609	CONNECT	
The data logger sends data which is captured at the remote end.		
+++	OK	Puts the Bluetooth Phone Modem into command mode
ATH	OK	Drops the call to 02089388609
^^^	OK	Puts the device into command mode
ATH	OK	Drops the Bluetooth connection

14.3 Example 3: Trusted Device Management

14.3.1 Background

Bluetooth provides for secure connections through the use of link keys. The link keys are 128 bit entities which are uniquely created for each BT device via the device's Bluetooth address. Since all BT addresses are unique, this results in unique keys.

The link keys are 'private' objects and as such are never exposed via the AT interface. They at all times remain within the serial device. Therefore the serial module allows link key management by using Bluetooth addresses as handles.

The device has a database of link keys. Each record in the database has two fields. One field contains the Bluetooth address and the other contains the 128 bit link key. **Only the address field is viewable.**

The following subsections describe AT command sequence for typical pairing functions.

14.3.2 Obtain a new Link Key

This process will obtain a new link key for a remote device whose address is 123456789012 and whose pin code is known to be 12345.

AT Command	Response	Comment
AT+BTW123456789012	OK	
	PIN? 123456789012	This is sent to the host every 2000ms
	PIN? 123456789012	
AT+BTK="12345"	OK	Pairing complete. The link key is stored in a volatile cache
	PIN 0 123456789012	Pairing successful

14.3.3 Store link key in trusted device database

This process assumes that the procedure described in "Obtain a new link key" has been done and there exists a link key in the cache, and that the database is not full.

AT Command	Response	Comment
AT+BTT	OK	

14.3.4 List trusted device database

This process assumes that the procedure described in "Obtain a new link key" has been done and there exists a link key in the cache, and that the database is not full.

AT Command	Response	Comment
AT+BTT?	123456789012	
	123456789013	
	123456789014	
	123456789015	
	123456789016	
	OK	

14.3.5 Store link key in trusted device database (full)

This process assumes that the procedure described in “Obtain a new link key” has been done and there exists a link key in the cache, and that the database is full.

AT Command	Response	Comment
AT+BTT	ERROR	The data base is full
ATI6	8	The maximum size of trusted devices database
	OK	
AT+BTT?	123456789015	
	123456789016	
		
	123456789015	
	OK	
AT+BTD123456789016	OK	Deletes the key associated with 123456789016
AT+BTT	OK	The store will work