BTM510/511 MULTIMEDIA MODULE

User Guide
Version 6.4

global solutions: local support ™

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852-2923-0610
www.lairdtech.com/bluetooth

Revision History

Revision	Date	Description	Approved By
1.0	3/01/2012	Initial Release	Jonathan Kaye
2.0	3/30/2012	Updates and correct to firmware	Jonathan Kaye
3.0	4/13/2012	v18.1.3.0 General reformatting edits	Jonathan Kaye
4.0	10/16/2012	Fixed Search bug. Fixed general formatting and hyperlink issues	Jonathan Kaye
5.0	1/10/2013	Update FCC and IC statements Updated mechanical drawings	Jonathan Kaye
6.0	3/22/2013	Updated document for new firmware: v18.1.4.0 / BTM51x - 08	Jonathan Kaye
6.1	13 Jan 2014	Separated document into two documents: Hardware Integration Guide and User Guide	Sue White
6.2	25 Feb 2014	Removed references to BTM52x	Jonathan Kaye
6.3	09 Dec 2014	PSKEY_PCM_CONFIG32 updated to PSKEY_DIGITAL_AUDIO_CONFIG	Jonathan Kaye
6.4	02 Feb 2015	Updated guide for Firmware v22_2_5_0	Jonathan Kaye

Contents

1 OVERVIEW

The BTM510 and BTM511 are low-power Bluetooth® modules designed for adding robust audio and voice capabilities. Based on the market-leading Cambridge Silicon Radio BC05 chipset, these modules provide exceptionally low power consumption with outstanding range. Supporting Bluetooth v2.1+EDR specification, these modules provide the important advantage of secure simple pairing that improves security and enhances easy use.

At only 14 mm x 20 mm, the compact size of the BTM510 is ideal for battery-powered or headset form factor audio and voice devices. With a 16-bit stereo codec and microphone inputs to support both stereo and mono applications, these modules also contain a full, integrated Bluetooth-qualified stack along with SPP, HFP 1.6, HSP, AVRCP, and A2DP profiles. Customers using these modules may list and promote their products on the Bluetooth website.

The BTM510/511 modules include an embedded 32-bit, 64-MIPS DSP core within the BC05. This is integrated with the Bluetooth functionality which allows designers to add significant product enhancements including features such as echo cancellation, noise reduction, and audio enhancement using additional soft codecs. The availability of the 16MB of flash memory in the module allows complex functionality to be included. DSP routines can be licensed through a number of specialist partners. Typical applications for these modules include Bluetooth stereo headsets, VoIP phones, and wireless audio links.

To speed product development and integration, Laird Technologies has developed a comprehensive AT command interface that simplifies application development, including support for audio and headset functionality. Access to GPIO pins allows mapping for direct connection to actuator buttons on headsets. Combined with a low-cost development kit, Laird Technologies' Bluetooth® modules provide faster time to market.

1.1 Features

- Fully featured Bluetooth® multimedia module
- Bluetooth® v3.0+EDR
- Supports mono and stereo headset applications
- Adaptive Frequency Hopping to cope with interference from other wireless devices
- 32-bit Kalimba DSP for enhanced audio applications
- Support for Secure Simple Pairing
- External or internal antenna options
- HSP, HFP, A2DP, and AVRCP audio profiles
- 16-bit stereo codec and microphone input
- Integrated audio amplifiers for driving stereo speaker
- Comprehensive AT interface for simple programming
- Bluetooth END product qualified
- Compact size
- Class 2 output – 4 dBm
- Low power operation
- Wi-Fi co-existence hardware support

1.2 Application Areas

- High-quality stereo headsets
- Mono voice headsets
- Hands-free devices
- Wireless audio cable replacement
- MP3 and music players
- Phone accessories
- VoIP products
- Cordless headsets
- Automotive

2 AT COMMAND SET REFERENCE

2.1 Introduction

This document describes the protocol used to control and configure the BTM Bluetooth device.

The protocol is similar to the industry standard Hayes AT protocol used in telephony modems. It is appropriate for cable replacement scenarios, as both types of devices are connection oriented.

Just like telephony modems, the Laird Technologies device powers up in an unconnected state and only responds via the serial interface. In this state the device does not even respond to Bluetooth Inquiries. The host can then issue AT commands which map to various Bluetooth activities. The configuration of the device can be saved so that on the next power up the device is automatically discoverable or connectable.

The device has a serial interface which can be configured for baud rates from 1200 up to 921600 (default is 9600) and an RF communications end point. The latter operates in terms of connected and unconnected modes and the former operates in command and data modes. This leads to the matrix of states shown below.

Table 2-1: RF Communications end point states

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

The combinations Data Mode + RF Unconnected and Remote Command Mode + RF Unconnected are invalid and are ignored.

Navigation between these states is done by issuing AT commands, described in detail in subsequent sections.

2.2 Glossary of Terms

Table 2-2: Glossary

Term	Definition
A2DP	Advanced Audio Distribution Profile (unidirectional stereo audio)
ACL	Asynchronous Connection-Oriented Link
ACR	Auto Connect Record
ACS	Auto Connect Service
ADC	Analogue to Digital Converter
AGHFP	Audio Gateway Hands-Free Profile
AT	Command prefix, 'Attention'
AVRCP	Audio/Video Remote Control Profile
BISM	Bluetooth Intelligent Serial Module
CoD	Class Of Device (also referred to as "device class")
Codec	Device capable of encoding / decoding an analogue / digital signal
DAC	Digital to Analogue Converter
DREG	Dynamic Register
DSP	Digital Signal Processor
DUN	Dialup Network Profile
EIR	Extended Inquiry Response
eSCO	Enhanced Synchronous Connection Oriented Link (bidirectional mono audio for speech transmission)

Term	Definition
FTP	File Transfer Profile
GOEP	Generic Object Access Exchange Profile
GPIO	General Purpose Input Output
HCI	Host Controller Interface
HF	Hands-free Role of Hands-free Profile (Hands-free Unit)
HFG	Audio Gateway Role of Hands-free Profile (Hands-free Gateway)
HFP	Hands Free Profile
HID	Human Interface Device Profile
HS	Headset Role of Headset Profile ("Headset")
HSG	Audio Gateway Role of Headset Profile ("Headset Gateway")
HSP	Headset Profile
I2S	Inter IC (integrated circuit) Sound
I/O (IO)	Input/Output
Mic	Microphone
MITM	Man In The Middle
OPP	Object Push Profile
PBAP	Phone Book Access Profile
PT	PASSS THROUGH Command
PWM	Pulse Width Modulation
SBC	Sub Band Codec
SCO	Synchronous Connection Oriented Link (for bidirectional mono audio for transmission of speech)
SLC	Service Level Connection
SPP	Serial Port Profile
SSO	Serial Stream Oriented
SSP	Secure Simple Pairing
SUI	SUBUNIT INFO Command
Sxxx	S-Register No. xxx
TDL	Trusted Device List
UART	Universal Asynchronous Receiver / Transmitter
UI	UNIT INFO Command
VRA	Voice Recognition Activation
WBS	Wideband Speech

2.3 Overview of the BTM Product Family

Table 2-3: BTM product family

BTM410 / BTM 411	
Chipset	CSR BC4-Ext
Bluetooth version	2.1
Features	SSP, EIR, SCO (1), eSCO (1), 4 GPIOs
Profiles	SPP
(1) external codec required	
BTM510 / BTM 511	
Chipset	CSR BC5MM-Ext

BTM410 / BTM 411	
Bluetooth version	3.0
Features	SSP, EIR, SCO, eSCO, 4 GPIOs, APTx, AAC (sink), CVC
Profiles	SPP, A2DP, AVRCP,HSP,HFP

2.4 BTM - AT Command Set

This section describes the AT Command Set for a BTM module. This section divides AT commands into functional groups, including module configuration, Bluetooth profiles, hardware units and miscellaneous functions.

2.4.1 Assumptions

- All commands are terminated by the carriage return character 0x0D, which is represented by the string <cr> in descriptions below. This cannot be changed.
- All responses from the module have carriage return and linefeed characters preceding and appending the response. These dual character sequences have the values 0x0D and 0x0A respectively and shall be represented by the string <cr,lf>.
- All Bluetooth addresses are represented by a fixed 12-digit hexadecimal string, case insensitive.
- All Bluetooth Device Class codes are represented by a fixed six digit hexadecimal string, case insensitive.
- All profile-specific commands are identified by the prefix shown in Table 3-2.

Table 2-4: AT command prefix for profiles

Profile	Term	AT-Command Prefix
Serial Port Profile	SPP	AT+SP…
Advanced Audio Distribution Profile	A2DP	AT+AP…
Audio/Video Remote Control Profile	AVRCP	AT+AV…
Headset Profile	HSP	AT+HS…
Hands-Free Profile	HFP	AT+HF…

2.4.2 Command Syntax

The following syntax is used in this document to describe optional or mandatory parameters for AT commands.

<bd_addr>	A 12 character Bluetooth address made of ASCII characters '0' to '9', 'A' to 'F' and 'a' to 'f'.
<devclass>	A 6 character Bluetooth device class made 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 is 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'.

2.5 Module Configuration

2.5.1 General AT Commands

2.5.1.1 AT

Used to check if the module is available.

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

2.5.1.2 ATEn {Enable/Disable Echo}

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

E0 … Disable echo.
E1 … Enable echo.

All other values of n generate an error.

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

Or

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

2.5.1.3 ATZ<n> {Hardware Reset and emerge into boot mode 'n'}

Forces the device through a hardware reset so that it eventually comes alive in local command and unconnected mode. This allows changes to the non-volatile memory to take effect. After the reset is complete and the module is ready to receive commands, it issues the OK response.

ATZ and ATZ0 initiate a reset followed by loading into the current boot mode (see command ATI14). ATZ1 to ATZ4 instructs the module to reset and then emerge into their corresponding boot mode. Note that S Register 103 specifies the boot mode from cold.

Boot modes are required to configure some low level device settings which cannot be configured by S registers and AT commands. Currently there are predefined settings defining the PCM data format to be used with certain codec ICs (applies mainly to BC04).

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

2.5.1.4 AT+BTC<devclass$_{hex}$> {Set Device Class Code Temporarily}

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

<devclass> is a six digit hexadecimal number derived as per "Bluetooth Assigned Numbers" [8].

The 24 bits are 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:	Minor Device Class: The value of these 6 bits is interpreted differently based on the Major Device Class stored in the next 5 bits.
Bits 8-12:	Major Device Class: 5 bits, see Figure 1 and Table 3 in "Bluetooth Assigned Numbers" [8].
Bits 13-23:	Major Service Class: 11 bit field, used as a mask to define service classes, refer to Figure 1 and Table 2 in "Bluetooth Assigned Numbers" [8].

Laird Technologies 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. Profile specifications define certain mandatory flags to be set in the device class code. These can usually be found in the section named "Link Controller (LC) Interoperability Requirements" in the appropriate profile specification.

Other examples of device class codes are listed in Table 2-5.

Table 2-5: Device class codes

Code (Hexadecimal)	Name	Major Service	Major Device	Minor Device
0x001F00	Unclassified	None	Unclassified	n/a
0x200404	Headset	Audio	Audio/Video	Wearable Headset Device
0x200408	Hands-free device	Audio	Audio/Video	Hands-free Device

A free tool, Class of Device Generator, is available online to create a particular device class code: see [9].

A device class set by AT+BTC becomes visible immediately but is lost on the next power cycle.

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>

2.5.1.5 ATS515=<devclass$_{hex}$> {Set Device Class Code Permanently}

S Register 515 sets the device class code permanently. Use AT&W to save the setting to non-volatile memory. The new value becomes visible on next power cycle which can be initiated by ATZ. Refer to AT+BTC<devclasshex> {Set Device Class Code Temporarily} for more about the device class code.

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

2.5.1.6 AT+BTC? {Read Device Class Code}

This command reads the current device class code.

Response: <cr,lf>123456

<cr,lf>OK<cr,lf>

2.5.1.7 AT+BTF="<string>" {Set Friendly Name Temporarily}

This sets the friendly name of this device as seen by other devices. The new name becomes immediately visible. Any name set by this command is lost on next power cycle. Refer to S Register Table 3-1.

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

2.5.1.8 AT+BTN="<string>" {Set Friendly Name Permanently}

This sets the default friendly name of this device as seen by other devices. It is stored in non-volatile memory. The new name becomes visible to other devices on next power cycle. Use AT+BTF to make the name visible immediately. Use AT+BTN? To read it back. An empty string ("") deletes the string from non-volatile memory which enables the default name, "Laird BTM 789012". The digits in the default friendly name represent the last 6 digits of the local Bluetooth address.Refer to S593 in Table 3-1. If a new value must be retained permanently, save it to non-volatile memory by "AT&W".

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

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

Read the default friendly name from non-volatile memory.

Response: <cr,lf>My Friendly Name<cr,lf>
<cr,lf>OK<cr,lf>

2.5.1.10 AT+BTF<bd_addr> {Get Remote Friendly Name}

This command gets the remote friendly name of the peer specified.

Response: <cr,lf><bd_addr>,Friendly Name
<cr,lf>OK<cr,lf>

2.5.1.11 AT+BTP {Make Device Discoverable and Connectable}

Make the device discoverable and connectable and wait for a connection from any device.

The setting remains valid until next reset or power cycle (unless not changed by any other AT command subsequently). For permanent discoverable/connectable settings, refer to S Register 512.

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

2.5.1.12 AT+BTQ {Make Device Discoverable}

Make the device discoverable but not connectable. Being discoverable implies that this device responds to inquiries from other devices (inquiry scans enabled).

The setting remains valid until next reset or power cycle (unless not changed by any other AT command subsequently). For permanent discoverable/connectable settings, refer to S Register 512.

Use AT+BTX to make the device not discoverable.

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

2.5.1.13 AT+BTG {Make Device Connectable}

Make the device connectable but not discoverable and wait for a connection from any device.

The setting remains valid until next reset or power cycle (unless not changed by any other AT command subsequently). For permanent discoverable/connectable settings, refer to S Register 512.

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

2.5.1.14 AT+BTV<bd_addr>,<uuid> {SDP Query for Service }

This command interrogates the SDP database of the peer device <bd_addr> for the service <uuid>. It results in an ACL connection and then an SDP transaction.

If the <uuid> service is present then it returns:

Response: <cr,lf>0
<cr,lf>OK<cr,lf>

If the <uuid> service is not present then it returns:

Response: <cr,lf>1
<cr,lf>OK<cr,lf>

If the device < bd_addr > cannot be reached, or is in non-connectable mode then it returns:

Response: <cr,lf>2
<cr,lf>OK<cr,lf>

If the SDP database is corrupt or invalid then it returns:

Response: <cr,lf>3
<cr,lf>OK<cr,lf>

If the device is not in idle mode then it returns:

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

In this case, the command AT+BTX may put the device into the correct idle mode.

2.5.1.15 ATIn {Information}

This returns the information about the Laird Technologies device and its status. Refer to Table 3-2 (Appendix) for a complete list of supported ATIn parameters.

For recognised values of n:

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

For unrecognised values of n.

Response: <cr,lf>**Laird Technologies Inc, UK, ©2014**<cr,lf>

2.5.1.16 AT+SIT<n> {play a pre-defined tone}

<n> = index of tone, 1..94 (dec); volume of tone is controlled by S387

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

2.5.1.17 AT+UC<x> {enable/disable/query unrecognised command notification message '…UC'}

<x> = 1/0/? corresponding to enable/disable/query;

2.5.1.18 AT+RC<x> {enable/disable/query SLC/ACL spy message '…RX'}

<x> = 1/0/? corresponding to enable/disable/query;

2.5.2 AT Commands for S Registers

As with modems, the Bluetooth module employs a concept of registers which are used to store parameters, such as escape sequence character or inquiry delay time.

For a list of general S registers refer to General S Registers and Table 3-1.

S registers associated with a particular profile or specific functions, are described in the appropriate profile section of this document. The following AT commands edit the values of S registers.

2.5.2.1 ATSn=m {Set S Register}

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 or 1000 to 1010 as they are updated in non-volatile memory when the command is received.

2.5.2.2 ATSn? {Read S Register Value}

This returns 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>

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

This returns 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>

2.5.2.4 AT&Fn {Set S Register Defaults}

AT&Fn (n=0…8) allows you to set pre-defined configurations for maximum, medium, and low power consumption. This allows you to set up the trade-off between power consumption and data latency without having to deal with details of the various parameters and S-registers.

This command only works when the device is in local command and unconnected mode.

Legal values of **n** are as per Table 2-6. All other values of **n** generate a syntax error response. If **n** is not specified, a default value of 0 is assumed and the baud rate is not changed.

Table 2-6: Pre-defined power settings

AT&F*n*	UART Baud Rate	Scan Power Setting (discoverable/connectable)	Sniff Mode/ SSR Power Setting
0 (default)	Unchanged	Medium	Maximum
1	9600	Minimum	Minimum
2	38400	Minimum	Minimum
3	115200	Minimum	Minimum
4	115200	Medium	Medium
5	115200	Maximum	Maximum
6	115200	Maximum	Maximum
7	Unchanged	Medium	Medium
8	Unchanged	Minimum	Minimum

The new values are not updated in non-volatile memory until the AT&W command is sent to the device.

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

Or

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

Table 2-7: Pre-defined Min/Med/Max power setting parameters

<table>
<tr><th>Scan Power Setting</th><th>Minimum</th><th>Medium</th><th>Maximum</th></tr>
<tr><td>S508 (page scan interval, ms)</td><td>2500</td><td>640</td><td>640</td></tr>
<tr><td>S509 (page scan interval, ms)</td><td>11</td><td>160</td><td>320</td></tr>
<tr><td>S510 (inq. scan interval, ms)</td><td>2500</td><td>640</td><td>640</td></tr>
<tr><td>S511 (inq. scan window, ms)</td><td>11</td><td>160</td><td>320</td></tr>
<tr><th>Sniff Mode/SSR Power Setting</th><th>Minimum</th><th>Medium</th><th>Maximum</th></tr>
<tr><td>S348 (SSR max. remote latency, 0.1s)</td><td>7</td><td>3</td><td rowspan="2">0 (disabled)</td></tr>
<tr><td>S349 (SSR remote timeout, 0.1)</td><td>10</td><td>20</td></tr>
</table>

S350 (SSR local timeout, 0.1)	10	20	
S561 (SM attempt, ms)	2	2	0 (disabled)
S562 (SM timeout, ms)	2	30	
S563 (SM min. interval, ms)	50	30	
S564 (SM max. interval, ms)	100	50	

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

The AT&F* variant of the command installs values in S registers as per command AT&F1 and then erases all user parameters in non-volatile memory. The trusted device database is cleared, as are parameters related to AT+BTR, AT+BTN, and AT+BTS.

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

Or

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

2.5.2.6 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, as are parameters related to AT+BTR, AT+BTN, and AT+BTS.

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

Or

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

2.5.2.7 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>

2.5.3 General S Registers

Refer to Appendix, Table 3-1 for a list of supported S Registers.

The main purpose of S Registers is to make the device configuration persistent. All S Registers can be saved to non-volatile memory by AT&W.

In some cases, an AT command and an S register exist for one and the same setting. In the majority of those cases the AT command's setting is lost on next power cycle. The S register can be saved and is still available after power cycle. This rule applies to many but not all of those cases.

2.5.4 AT Commands for Inquiry

2.5.4.1 AT+BTI<devclass> {Inquire}

This initiates an inquiry for **delay** seconds and **max** number of unique responses, where **delay** is defined 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, the value is taken from S register 516.

When <devclass> is 6 hexadecimal characters long, it specifies an AND mask which filters 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.

The response format to AT+BTI is defined by S330 by bitmask. This is device address, device class, friendly name, receiver strength indicator and extended inquiry data. Refer to Table 2-8 and Figure 2-1.

For S330=1:

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

In the Bluetooth inquiry process, a device could respond many times for a single inquiry request. 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. If the memory allocation fails, the inquiry procedure is aborted and an error response is sent to the host. To clarify, a single AT+BTI does not return the same Bluetooth address more than once. As long as the responding device is active, all AT+BTI commands always return it.

As the inquiry process is driven by randomness, it is not guaranteed that each discoverable device is always found on the first attempt. Sometimes more than one inquiry processes might be necessary to find a particular device. The probability also depends on the inquiry scanning intervals of the device being searched for.

The inquiry process can be speed up if the friendly name is not required (flag not set in S330) as part of the inquiry response or if a <dev_class> filter is used.

Although it is very convenient to have the friendly name displayed in the inquiry response, this option can significantly lengthen the inquiry process. In areas with a large number of discoverable Bluetooth devices it might become nearly impossible to find a particular device.

An optimal solution would be a first inquiry scan without friendly name and <dev_class> filter. In a second run, the friendly name is queried by AT+BTF<BdAddr> for each device found.

Bit	7	6	5	4	3	2	1	0
	Reserved for future usage			EIRD	RSSI	FN	COD	ADR
Default	0	0	0	0	0	0	0	1

Figure 2-1: Register 330 controlling inquiry response format

Table 2-8: Field Descriptions for S Register 330

Field	Description
0 – ADR	Bluetooth device address 1 – display Bluetooth device address on inquiry result.

Field	Description
	0 – do not display Bluetooth device address on inquiry result. If any further bit is set, a comma is inserted as separator.
1 – COD	Class of device 1 – display class of device on inquiry result. 0 – do not display class of device on inquiry result. If any further bit is set, a comma is inserted as separator.
2 – FN	Friendly name 1 – display friendly name on inquiry result 0 – do not display friendly name on inquiry result. If any further bit is set, a comma is inserted as separator.
3 – RSSI	Receiver signal strength indicator (RSSI) 1 – display RSSI value on inquiry result. 0 – do not display RSSI value on inquiry result. If any further bit is set, a comma is inserted as separator.
4 – EIRD	Extended inquiry response data 1 – display EIRD on inquiry result. 0 – do not display EIRD on inquiry result.

2.5.4.2 Inquiry Response format

The format of an inquiry result is:

<cr,lf><bd_addr>,<dev_class>,<friendly_name>,<rssi>,<eir_data><cr,lf>

<bd_addr>	= 12 digit, hexadecimal;
<dev_class>	=6 digit, hexadecimal;
<friendly_name>	= printable ASCII character, enclosed by ‘ “ ‘
<rssi>	= signed 2 digits decimal
<eir_data>	= printable ASCII character whenever possible, otherwise a byte is displayed as 2 digit hexadecimal with preceding ‘\’, enclosed by ‘ “ ‘

For example the hexadecimal data block 01 41 42 43 44 02 03 34 35 36 04 0A 0D is presented as \01ABCD\02\03456\04\0A\0D

No validation is performed on incoming EIR data.

If a higher significant flag is set and a lower significant bit is not set in S330, for each disabled item a comma is printed.

Example: S330 = 9 (ADDR enabled, COD and FN disabled, RSSI enabled)

Inquiry Response:

<cr,lf>123456789012,,,-54
<cr,lf>123456789014,,,-54
<cr,lf>OK<cr,lf>

2.5.4.3 AT+BTIV<devclass> { Inquire }

As per AT+BTI but the response comprises for all inquiry responses:

- Bluetooth device address
- Device class code

S register 330 is not referenced.

2.5.4.4 AT+BTIN<devclass> { Inquire }

As per AT+BTI but the response comprises for all inquiry responses:

- Bluetooth device address
- Device class code
- Friendly name

S register 330 is not referenced.

2.5.4.5 AT+BTIR<devclass> { Inquire }

As per AT+BTI but the response comprises for all inquiry responses:

- Bluetooth device address
- Device class code
- Friendly name
- RSSI (receiver signal strength indicator)

S register 330 is not referenced.

2.5.4.6 AT+BTIE<devclass> { Inquire }

As per AT+BTI but the response comprises for all inquiry responses:

- Bluetooth device address
- Device class code
- Friendly name
- RSSI (receiver signal strength indicator)
- Extended inquiry data

S register 330 is not referenced.

2.5.5 AT Commands for Extended Inquiry Response Data

Bluetooth 2.1 specification allows up to 240 Bytes of extended inquiry data. On BTM5xx modules, this data is limited to a maximum length of 112 Bytes due to internal memory restrictions. Extended inquiry data may be utilised to transmit information such as the friendly name, UUIDs of supported profiles or user defined data within the inquiry process and without creating a Bluetooth connection.

The architecture for managing EIR data is composed of three buffers and a set of AT commands around them:

- Baseband (EIR data visible to inquiring devices)
- RAM buffer (allows accumulation of data)
- EIR persistent store (non-volatile buffer, copied to baseband at boot time)

As the input buffer length for one AT command is limited, there is a RAM buffer to accumulate several short data packets. The accumulated data of the RAM buffer can be copied to the Baseband where it becomes visible to other inquiring devices immediately. The content of the RAM buffer can also be copied to the EIR persistent store. If the EIR persistent store contains data, it is copied to the Baseband automatically at boot time.

This allows a flexible usage of extended inquiry data. For example, data with a low data rate (e.g. temperature) can be transmitted without creating a Bluetooth connection. This method sacrifices encryption and authentication.

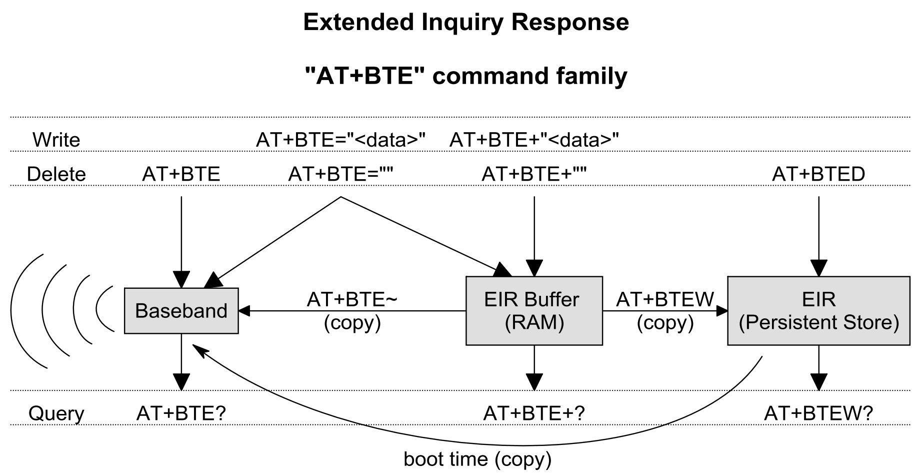

Figure 2-2: Extended Inquiry Response – command overview

2.5.5.1 EIR Data Format

When passing EIR data (<data>) to AT commands (AT+BTE=<data>/ AT+BTE+<data>), each byte should be presented by its ASCII representation whenever it is a printable character. Each non-printable ASCII character must be presented as 2 hex digits with a preceding '\'. For example, a byte of decimal value 5 would be presented as \05 because the ASCII character of 05d is not printable. A decimal value of 43 should be presented as + because + is the ASCII character representing 43d. The module would also accept "\2B" (the hexadecimal presentation of 43d) but at the price of two redundant characters.

Exceptions:

'"' (quotation mark) ***must*** be presented as \22

'\' (backslash) ***must*** be presented as \5C

When querying the content of any buffer (Baseband / RAM / Persistent Store), non-printable ASCII characters is presented by 2 hex digits with preceding '\'.

Exceptions:

'"' (quotation mark) is presented as \22

'\' (backslash) is presented as \5C

',' (comma) is presented as \2C

Data passed to the baseband must match the format defined in the Bluetooth Specification Version 2.1 + EDR [1], vol3, Part C – Generic Access Profile, 8 Extended Inquiry Response Data Format (page 1305 in the *.pdf file). The AT command interpreter does not perform any checks on the baseband data format.

2.5.5.2 AT+BTE+<data>{Accumulate data in RAM buffer}

This command adds <data> to the content of the RAM buffer. The maximum number of characters for <data> is 25 due to the limited AT command input buffer.

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

Or: <cr,lf>**ERROR 05**<cr,lf>

2.5.5.3 AT+BTE=<EIR-data> {Write EIR data to baseband and RAM buffer}

This command writes EIR (extended inquiry response) data to the baseband and to the RAM buffer. The maximum number of characters for <EIR-data> is 25 due to the limited AT command input buffer. See AT+BTE+<data>{Accumulate data in RAM buffer} for more information.

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

Or: <cr,lf>ERROR 05<cr,lf>

2.5.5.4 AT+BTE~ {Copy RAM buffer to baseband}

This command copies all data from the RAM buffer to the baseband. The data passed to the baseband must match the EIR data format as specified in the BT2.1 specification (page 1305 in the *.pdf file). S See AT+BTE+<data>{Accumulate data in RAM buffer} for more information.

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

2.5.5.5 AT+BTEW {Copy RAM buffer to EIR persistent store}

This command copies all data from the RAM buffer to the (non-volatile) persistent store. If the EIR persistent store contains any data at boot time, this data is copied to the baseband at boot time automatically. Therefore, copying data to the EIR persistent store makes it visible to inquiring devices from next power cycle onwards. Data passed to the baseband must match the EIR data format as specified in the BT2.1 specification (page 1305 in the *.pdf file). See AT+BTE+<data>{Accumulate data in RAM buffer} for more information.

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

2.5.5.6 AT+BTE+? {Query data from RAM buffer}

This command prints the data that is currently stored in the RAM buffer.

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

2.5.5.7 AT+BTE? {Query outgoing EIR data from baseband}

This command prints the outgoing EIR data that is currently set up in the local baseband. Some interpretation on the EIR data format is done here. If the leading byte of a data block contains information of the wrong length, then some unexpected output may appear, e.g. \00 is appended.

Response: <cr,lf><EIR-data><cr,lf>OK<cr,lf>

2.5.5.8 AT+BTEW? {Query data from RAM buffer}

This command prints the data that is currently stored in the EIR persistent store.

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

2.5.5.9 AT+BTE {Delete EIR data from baseband}

This command deletes the EIR data in the baseband.

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

2.5.5.10 AT+BTE="" {Delete EIR data from baseband and RAM buffer}

This command deletes the EIR data in the baseband and deletes any data from the RAM buffer.

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

2.5.5.11 AT+BTE+"" {Delete RAM buffer}

This command deletes all data from the RAM buffer.

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

2.5.5.12 AT+BTEW {Delete EIR persistent store}

This command deletes the EIR persistent store.

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

2.5.6 Secure Simple Pairing (SSP)

Secure Simple Pairing (SSP) has been introduced since Bluetooth 2.1 + EDR. It aims to increase the security provided by a Bluetooth link whilst making the pairing process more user friendly.

There are whitepapers about SSP available through the internet (provided by the Bluetooth SIG and other companies), explaining the background and mechanisms of SSP. They can be found by searching the internet for topics such as *Bluetooth Secure Simple Pairing*. Read these to better understand SSP and the following settings.

2.5.6.1 Security Level (S320)

The security level is defined in the BT2.1+EDR specification [1], vol3, Generic Access Profile (Table 5.7). The specification provides four levels of security, shown in Table 2-9.

Table 2-9: Security Levels

Security Level	Characteristics	Comment
Level 3	MITM protection (Man in the Middle attack) Encryption User interaction	High security
Level 2	No MITM protection Encryption	Medium Security
Level 1	No MITM protection (No) Encryption (1) Minimal user interaction	Low Security
Level 0	No MITM protection No Encryption Minimal user interaction	Permitted only for service discovery

(1) Although encryption is not necessary for security level one, encryption is always enabled because this specification mandates encryption for all services other than SDP (service discovery).

The security level is defined by S Register 320 and is referenced at boot time only. Hence the register must be saved by AT&W and the module must be power cycled (or ATZ) subsequently.

S320 = 3 overrules the setting of S Register 322 (enable MITM protection).

The security level remains the same until next power cycle and is valid for all profiles and services of the module. For SDP (service discovery profile), security level 0 is always assigned internally.

2.5.6.2 IO-capability (S321)

S-Register 321 defines the IO-capability of the device. The setting is used for IO-capability negotiations prior to SSP in order to identify whether the IO-capabilities of both devices are sufficient for MITM protection (if required). Table 2-10 displays possible values.

Table 2-10: IO capabilities

S321	IO-Capability	Comment
0	Display only	The device is able to display / communicate a six digit decimal number.
1	Display yes no	The device can display or communicate a six digit decimal number and has a mechanism that allows the user to indicate either *yes* or *no* (pressing a single yes button before a timer expires or two buttons for yes and no)
2	Keyboard only	The device has a numeric keyboard that can input numbers 0 through 9 and a confirmation. The device has at least two buttons that can be easily mapped to 'yes' and 'no' or a mechanism whereby the user can indicate either 'yes' or 'no' (e.g. pressing a button within a certain time limit)
3	No input no output	The device does not have the ability to indicate 'yes' or 'no', and the device cannot display or communicate a 6 digit decimal number.
4	Reject IO-Cap requests	IO-capability requests prior to SSP are rejected.

2.5.6.3 Force Man-in-the-middle protection (MITM, S322)

S322 can enable protection against MITM-attacks. This S-Register only applies if the security level (S320) is less than 3. For security level (S320) = 3, MITM protection is always enabled and this S 322 is ignored.

- A new value written to S322 applies immediately. No power cycle is required.
- A link key created with MITM protection is named **authenticated link key**.
- A link key created without MITM protection is named **unauthenticated link key**.

2.5.6.4 Disable Legacy Pairing (S323)

If the remote device is a legacy device (BT2.0 or earlier), legacy pairing with usage of PIN codes is used. Legacy Pairing can be disabled by S-Register 323 = 1, but then pairing with legacy devices always fails.

2.5.6.5 SSP Timeout (S324)

The SSP timeout [s] is defined by S-Register 324. The timeout must be at least 60 s to meet the BT specification requirements [1]. This time must be long enough for the user to compare or read and input a 6 digit number. A time of 90 seconds is recommended, which is the default value.

2.5.6.6 SSP Input commands

Table 2-11 lists all AT commands related to SSP input operations.

Table 2-11: SSP Input commands

AT Command	Operation	Comment
AT+BTBY	Accept pairing request	Representing 'yes' input
AT+BTBN	Reject pairing request	Representing 'no' input
AT+BTB012345	Enter 6 digit passkey displayed by remote device	Representing keyboard input

2.5.6.7 AT+BTW<bd_addr> {Initiate SSP}

This command initiates secure simple pairing (dedicated bonding) with a device whose Bluetooth address is <bd_addr>. The Bluetooth 2.1+EDR specification's term for this is "Dedicated Bonding".

Dedicated bonding means the exchange of link keys (pairing) without creating a connection to a particular profile or service immediately.

The remote device must have Bluetooth 2.1 or later, otherwise legacy pairing occurs automatically if S323=0. For legacy pairing refer to AT Commands for Legacy Pairing.

The module immediately sends "OK" on receipt of AT+BTW. Depending on the devices' collective IO-capabilities, an asynchronous message may appear during pairing. See Table 2-13 for the required actions.

On pairing completion, an unsolicited message in the form PAIR n <bd_addr> <nn> <lktype> is sent to the host.

2.5.6.8 S Registers for Secure Simple Pairing

Table 2-12 lists all S Registers for Secure Simple Pairing. For details on the S Registers refer to their descriptions above.

Table 2-12: S-Registers for Secure Simple Pairing (SSP)

Register	Default	Range	Comment
S320	2	1..3	Security Level: see [1], vol3, Generic Access Profile - Table 5.7 needs subsequent 'AT&W' and power cycle to take effect value = 3 overwrites S322
S321	1	0..4	Set IO capability: 0 – display only 1 – display yes no 2 – keyboard only 3 – no input no output 4 – reject IO-cap requests
S322	0	0..1	Force man-in-the-middle-protection (MITM): 0 – disabled 1 – enabled referenced only if security level (S320) < 3
S323	0	0..1	Disable legacy (pre-BT2.1) Pairing: 0 – legacy pairing enabled 1 – legacy pairing disabled
S324	90	1..255	Secure Simple Pairing timeout in s This value must be at least 60 in order to meet the recommendation of BT2.1 specification

2.5.6.9 Asynchronous SSP messages

Table 2-13 lists asynchronous messages which occur if MITM is enabled. The sent message depends on the combination of the IO capabilities of both ends. The combination of IO capabilities of both devices may be insufficient for MITM protection. In that case the pairing fails (PAIR 2 <BdAddr>). Refer to Table

5.6 in BT2.1+EDR specification [1], vol3, Generic Access Profile for sufficient combinations of IO-capabilities for MITM (=authenticated link key).

Table 2-13: Asynchronous messages for SSP

Message	Action / Comment
PAIR ? <BdAddr>,<friendlyname>,<Passkey> Example: PAIR ? 0016A4000002,Laird BTM 000002,863611	Passkey compare request. Prompts the user to compare the passkey displayed on both ends and to confirm a match by AT+BTBY at both ends or reject by AT+BTBN to deny a match.
PASSKEY ? <BdAddr>,<friendlyname> Example: PASSKEY ? 0016A4000001,Laird BTM 000001	Passkey request. Prompts the user to enter the passkey displayed by the remote device. Use AT+BTB<passkey>, example: AT+BTB012345, *see(1) below
PAIR N <BdAddr>,<friendlyname>,<Passkey> Example: PASSKEY N 0016A4000002,Laird BTM 000002,164585	Passkey notification. Display BdAddr, friendly name and passkey to user; Prompts the user to enter the passkey from this message at the remote device's numeric keyboard.
PAIR 0 <BdAddr> <nn> <lktype>	Successfully paired with device of <BdAddr>. <nn> (optional) indicates the status of automatic storage to trusted device list. Value 0 = success; Settings are controlled by S325 to S328. Refer to Automatic storage of link keys. <lktype> indicates the link key type: 0 – no link key 1 – legacy link key (BT2.0 and earlier) 2 – debug link key (should never occur in practice) 3 – unauthenticated link key (no protection against MITM attack) 4 – authenticated link key (protection against MITM attack) 5 – link key changed In practice only values of 1,3,4 should play a role
PAIR 1 <BdAddr>	Pairing timeout, see (2)
PAIR 2 <BdAddr>	Pairing failed, see (2)
PAIR 3 <BdAddr>	Pairing failed (too many repeat attempts)
PAIR 4 <BdAddr>	Pairing rejected by remote device
PAIR 5 <BdAddr>	Pairing failed (unit keys not supported)
PAIR 6 <BdAddr>	Pairing failed (SSP not supported)
PAIR 7 <BdAddr>	Pairing failed (already busy with pairing)

(1) If both devices have a "KeyboardOnly" capability, no pass key can be displayed. In that case, the user is required to invent and enter the identical 6 digit numeric passkey at both ends.

(2) it was observed that scenarios which would have returned "PAIR 2 …" (pairing failed) with f/w v18.1.4.0, do return "PAIR 1 …" (pairing timeout) with f/w v22.2.5.0. It may be necessary to review existing host controller source code to avoid issues due to f/w upgrade.

2.5.7 AT Commands for Legacy Pairing

2.5.7.1 AT+BTW<bd_addr>{Initiate Pairing}

Provided the remote device is a Bluetooth 2.0 device or earlier and legacy pairing is not disabled (S323 = 0), this command initiates legacy pairing with the device with <bd_addr>. Legacy pairing refers to the mechanism of entering an identical PIN key on both ends.

If the PIN is required (if not set earlier by AT+BTK="<PIN>"), asynchronous indications are sent to the host in the form PIN? <bd_addr>. The address confirms the pairing device. To supply a PIN, use AT+BTK.

For a successful pairing, the link key is stored in a volatile cache which is overwritten every time a new pairing is initiated using this command. If S register 325=1, the link key is automatically saved to the non-volatile trusted device list. Otherwise (S325=0) the link key can be added to the trusted device list by AT+BTT. Refer to AT Commands managing Trusted Devices for further AT commands related to trusted device list.

The OK response is sent immediately on receipt of the AT+BTW command. After pairing, an unsolicited message is sent to the host in the form PAIR n <bd_addr> <nn> <lktype>.

If AT+BTI or AT+BTP or AT+BTG or AT+BTQ or ATD is issued between the AT+BTW command and the subsequent PAIR asynchronous response, then an ERROR response is sent to those commands. They cannot be executed in this mode.

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

2.5.7.2 AT+BTK="<string>" {Set Passkey}

This command provides a PIN passkey. The PIN is stored in non-volatile memory for future use. If this command is used as response to a PIN? 12345678 asynchronous message, the PIN provided by this command is not stored in non-volatile memory.

Specifying an empty string deletes the PIN from the non-volatile memory. The string length must be in the range 0 to 8 or an error is returned.

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

2.5.7.3 Legacy Pairing – Asynchronous Messages

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 does not supply the address in the message exchange, then the address is specified as 000000000000 – and the paring proceeds as normal.

PAIR n <bd_addr> <lktype>
This response is sent to the host on termination of a pairing process. If pairing is successful then 'n' = 0. If a timeout occurs then 'n'=1. For all other unsuccessful outcomes 'n' >= 2. The parameter <bd_addr> is the address of the peer device if available. <lktype> indicates the link key type. For legacy pairing the value should be 1.

PAIR 0 <bd_addr> <nn> <lktype>
This response is sent to the host on termination of a successful pairing process. The optional <nn> is sent only if the according S Register 325..328 is set to 1, automatically saving the link key. The value <nn> indicates the result of the save operation. A value of 00 implies success, otherwise <nn> is an error code. <lktype> indicates the link key type. For legacy pairing the value should be 1.