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 ✓RoHS

- 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

Scan Power Setting	Minimum	Medium	Maximum
S508 (page scan interval, ms)	2500	640	640
S509 (page scan interval, ms)	11	160	320
S510 (inq. scan interval, ms)	2500	640	640
S511 (inq. scan window, ms)	11	160	320
Sniff Mode/SSR Power Setting	**Minimum**	**Medium**	**Maximum**
S348 (SSR max. remote latency, 0.1s)	7	3	0 (disabled)
S349 (SSR remote timeout, 0.1)	10	20	

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.

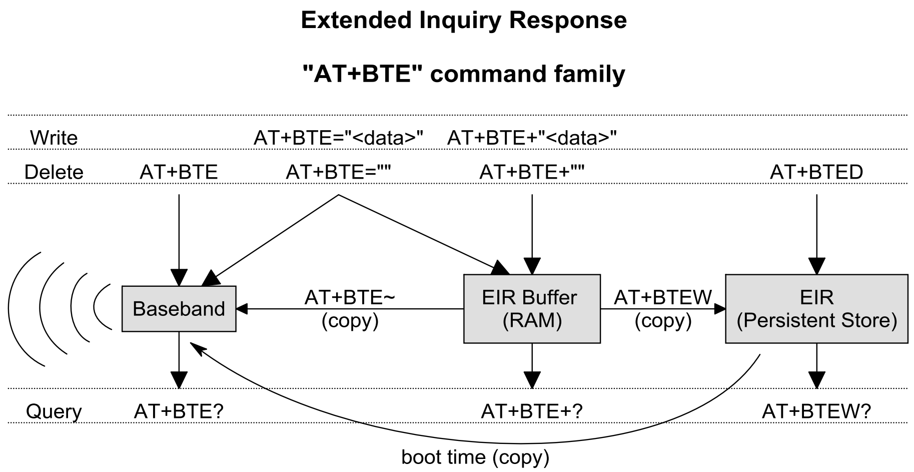


Figure 2-2: Extended Inquiry Response – command overview

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.

2.5.8 AT Commands Managing Trusted Devices

2.5.8.1 AT+BTT? {List Trusted Device}

This command lists the contents of the trusted device database. The link key is NOT displayed but the link key type is appended to each line of the response as shown below. If the list is empty then just the OK response is sent. Otherwise an OK terminates the list. Use the command ATI6 to read the maximum size of the trusted device database.

```
Response: <cr,lf>12346789012,4
          <cr,lf>123456789013,3
          <cr,lf>123456789014,3
          <cr,lf>OK<cr,lf>
```

<bdaddr>,<lktype>

Possible values for <lktype> are displayed in Table -2-14.

Table -2-14: Link key type values

<lktype> Value	Description
0	No link key
1	Legacy link key (BT 2.0 and earlier)
2	Debug link key (should never occur in practice)
3	Unauthenticated link key (no MITM protection)
4	Authenticated link key (MITM protection)
5	Link key changed

Note: In practice, only the values of 1, 3, and 4 should play a role.

Examples for pairing messages with link key type appended:

Unauthenticated link key	PAIR 0 0016A4000001 00 3
Authenticated link key	PAIR 0 0016A4000001 00 4

2.5.8.2 AT+BTT {Add Trusted Device}

This command stores the cached link key in the non-volatile database. If the database is full it responds with an ERROR. If the device is already in the database, then the key is replaced. If the link key cache is empty (a pairing has not been performed since powering) this responds with an ERROR.

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

Or

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

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

This command removes the specified device from the list of trusted devices in the non-volatile database. The response is OK even if the device is not in the database.

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

2.5.8.4 AT+BTD {Remove All Trusted Devices}*

This command removes all devices from the trusted device list (TDL) in the non-volatile database. The device does not ask for confirmation. Use with caution.

WARNING: If you make a connection, the link key gets cached in the underlying stack. So if you subsequently delete the key using AT+BTD* and immediately request a connection to the same device, then the connection is established. To ensure this does not happen, send ATZ after the AT+BTD*.

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

2.5.8.5 AT+BTW? {List Cached Trusted Device}

This command lists the cached trusted device.

Response: <cr,lf>12346789012

<cr,lf>OK<cr,lf>

If the cache is empty the response is as follows.

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

2.5.9 AT Commands for Serial Stream Oriented Profiles (SSO)

The Serial Port Profile (SSP) belongs to the group of Serial Stream Oriented profiles (SSO).

When activated, an SSO profile claims one UART for its data stream and assumes all data at the UART to be transmitted over or received from RF 1:1. Hence, as there is only one UART available on a BTM device, the UART is not available for other profiles, services or module control purposes.

One approach to managing data and control over UART is to configure local command mode with S531=3. In this mode, incoming RF data is presented by the asynchronous message RX<string>. Outgoing data is sent by ATX<string> or ATY<string>.

With this approach you may manage several non-SSO connections (e.g. A2DP, AVRCP) and one SSO connection (SSP). An attempt to simultaneously connect a second SSO profile returns Error 65.

Any incoming SSO connection request is also rejected if one SSO is already connected.

The following section describes AT- commands related to SSO-profiles.

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

This command sends data to the remote device when in local command and connected mode.

The parameter <string> is any string not more than 29 characters long whereby a non-printable character (\hh, see below) counts 3 characters. This restriction results from the maximum AT command length which is 34 (query by ATI15). The difference of 5 is caused by ATX (3 characters) and the enclosing quotation marks (2 characters).

If the maximum string length is exceeded, this generates ERROR 05 (syntax error).

If a non-visual character is to be sent then insert the escape sequence \hh where hh is two hexadecimal digits. The 3 character sequence \hh is converted into a single byte before transmission to the peer.

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

Or

<cr,lf>ERROR 05<cr,lf>   (e.g. <string> too long)

2.5.9.2 ATY<string> {Send Data in Local Command and Connected Mode}

This command is similar to ATX in syntax and functionality, except that the string is only copied to the output RF buffer. When an empty string is presented, all pending data in the output RF buffer is flushed.

The parameter <string> is any string not more than 29 characters long whereby a non-printable character (\hh, see below) counts 3 characters. This restriction results from the maximum AT command length which is 34 (query by ATI15). The difference of 5 is caused by ATX (3 characters) and the enclosing quotation marks (2 characters).

If the maximum string length is exceeded, ERROR 05 (syntax error) occurs.

If a non-visual character is to be sent then insert the escape sequence \hh where hh are two hexadecimal digits. The 3 character sequence \hh is converted into a single byte before transmission to the peer.

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

Or

<cr,lf>ERROR 05<cr,lf>   (e.g. <string> too long)

2.5.9.3 ^^^ {Enter Local Command Mode}

When in data and connected mode and when S 507 is set to 0 or 1, 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, which may be changed. In addition, the escape sequence guard time is specified by S 12. By default the guard time is set to 100 milliseconds.

Leaving data mode using "^^^"severe impacts data throughput, because each incoming character needs to be checked for '^' with respect to the guard time.

Alternatively, a de-assertion of the DTR/DSR line can be used as the only trigger to leave data mode (S507=2). This gives a significant higher data throughput because data is passed directly between UART and RF without character checking. Refer to Dropping SSO Connections for more information.

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

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

2.5.9.4 !!!{Enter Remote Command Mode}

When in data and connected mode, the host can force the remote device into connected command mode so that AT Commands can be remotely issued to the device. The escape sequence guard time is specified by S Register 12 and is the same as per the ^^^ escape sequence. By default the guard time is set to 100 milliseconds. The remote device issues ATO as normal to return to data mode (Refer to 0). For this command to be effective S Register 536 must be set to 1.

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

2.5.9.5 ATO {Enter Data Mode} (letter 'o')

Returns to data mode. This command assumes that the module is in data mode after OK is received. It responds with an error if there is no Bluetooth SSO connection.

Response:

<cr,lf> CONNECT 123456789012,<<cr,lf> (incoming connection)

<cr,lf> CONNECT 123456789012,><cr,lf> (outgoing connection)

Or

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

2.5.9.6 Dropping SSO Connections

In a conventional telephony modem, a call is normally terminated by first sending a +++ character sequence. It is enveloped by an escape sequence guard time (between 100 to 1000 milliseconds). This places the receiving device into local command and connected mode, whereupon the sender issues the ATH command.

The Laird modules provide multiple ways to drop a connection. One method is similar to the above, but instead a ^^^ character sequence is used. This is to eliminate ambiguity when a data call via a mobile phone is in progress when the call is established using the phone's Bluetooth AT modem. The second method involves the host dropping the DTR (DSR from the module's viewpoint) modem control line.

Dropping a connection using the escape sequence ^^^ severely impacts throughput, reducing the data rate from about 300 kbps to around 85 kbps. To cater for this performance hit, the device's connection drop capability is configurable to be in one of two modes.

The first mode allows a connection to be dropped using either method. This is the default. The second allows only the DTR drop mode. Enable this mode using the S507 register (AppendixTable 3-1).

The escape sequence is as follows:

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

When a file transfer is occurring which contains multiple <Esc Chr> characters, the module does not drop into command mode. Thi is because file transfer occurs quickly, and the gap between characters gap is much 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 (Appendix, Table 3-1).

2.5.9.7 SSO - Asynchronous Messages

RX<string>

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

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

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

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

2.5.9.8 SSO – S Registers

Table 2-15 lists S registers for SSO profiles.

Table 2-15: S Registers for SSO profiles

Register	Default	Range	Description

Register	Default	Range	Description
S2	94	32..126	Escape sequence character. It is not '+' by default, due to potential confusion when serial linked to a mobile phone, which exposes an AT command set and also uses '+' as default. If both use '+' there is confusion. 94 is the character '^'.
S12	100	40..5000	Escape sequence guard time in milliseconds, with a granularity of 20 ms. New values are rounded down to the nearest 20 ms multiple.
S507	0	0..2	When set to 0, a connection can be dropped using ^^^ escape sequence only and the state of DSR line is ignored. When set to 1 a connection can be dropped using EITHER the ^^^ escape sequence OR the DSR modem control line. When set to 2, a connection can only dropped by de-asserting DSR. Mode 2 provides for the highest data transfer rate. If the DSR line's status is to be conveyed to the remote device as a low bandwidth signal, this register MUST be set to 0. Otherwise de-asserting DSR is seen as a request to drop the Bluetooth® connection. This register affects S Register 536 – see details of 536.
S531	0	0..4	On SPP connect mode: Specifies the mode on SPP connection establishment. 0 Normal. Data is exchanged between UART and RF. 1 LOCAL_COMMAND. UART input is parsed by the AT interpreter and RF data is discarded. 2 REMOTE_COMMAND. RF input is parsed by the AT interpreter and UART data is discarded. If S Reg 536 is not 1 then this register cannot be set to 2. An ERROR is returned. 3 LOCAL_COMMAND. UART input is parsed by the AT interpreter. Incoming RF data is sent to the host using the RX<string> asynchronous response. 4 LOCAL_COMMAND. On the RF side, the GPIO is automatically sent when there is a change in input (digital I/O cable replacement mode).
S536	0	0..1	When set to 1, a remote device can 'capture' the AT parser of this unit by sending this module an escape "!!!" sequence. Inter-character timing is set via S Register 12. If S Register 507 is >= 2, reading this register always returns 0. Writing 1 results in ERROR 33.

2.5.10 AT Commands for a Selected Peer Device

This section describes AT commands to make the BTM Bluetooth device connectable for one specific remote device only or to automatically connect to one particular remote device on reset or power cycle.

Prior to firmware v18.1.3.9, the AT commands of this section applied to serial port profile (SPP) only. Beginning with v18.1.3.9 (including v18.1.4.0 and later) these commands apply to all profiles supported by BTM51x.

2.5.10.1 AT+BTP<bd_addr> {Make Device Discoverable and Selectively Connectable}

Make the module discoverable (for all devices) and connectable for only the device with the Bluetooth address <bd_addr>. Connection requests from any other devices are rejected.

If <bd_addr> is 000000000000, incoming connections are accepted from any device. This is as per AT+BTP, but without an address.

The setting is valid until the next reset or power cycle (unless not changed by any other AT command subsequently). For permanent discoverable/connectable settings, refer to S512 and AT+BTM<bd_addr>.

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

2.5.10.2 AT+BTG<bd_addr> {Make Device Selectively Connectable Only}

Make the module connectable only to the device with the Bluetooth address <bd_addr>. Connection requests from other devices are rejected.

If the specified address is 000000000000, incoming connections are accepted from any device. This is as per AT+BTP without an address.

The module is not discoverable.

The setting is valid until next reset or power cycle (unless changed by any other AT command thereafter). For permanent discoverable/connectable settings, refer to S Register 512 and AT+BTM<bd_addr>

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

2.5.10.3 AT+BTM<bd_addr> {Set Incoming Peer Address (non-vol. mem.)}

This command stores a peer address for incoming connections in non-volatile memory. Only the device with Bluetooth address <bd_addr> may connect to the module. Other devices are rejected.

The new setting applies immediately and remains over a power cycle, because it is copied from non-volatile memory to the incoming peer address variable at boot time. Any subsequent AT+BTP/G overwrites or clears the incoming peer address.

When S 512 = 3, 4, 6, or 7 then the module waits for an incoming connection from the peer address specified. If <bd_addr> is 000000000000, then incoming connections from any devices are permitted.

Note: AT+BTM<BdAddr> only makes sense if also ATS512=3, 4, 6 or 7 (plus AT&W and ATZ) is set. If the BTM51x is not discoverable and not connectable at boot time (ATS512=1), then AT+BTP or AT+BTG would clear the current incoming peer address

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

2.5.10.4 AT+BTM {Delete Incoming Peer Address (non-vol. mem.)}

This command deletes the peer address previously stored using AT+BTM<bd_addr>. If the module is connectable for the selected device before this command, it is connectable for any device afterward.

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

2.5.10.5 AT+BTM? {Read Incoming Peer Address (non-vol. mem.)}

This command displays the peer address stored in non-volatile memory, placing the module in pure cable replacement mode.

Response: <cr,lf>123456789012

<cr,lf>OK<cr,lf>

If the location is empty the response is as follows.

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

2.5.10.6 ATI75 {Read Current Incoming Peer Address}

ATI75 displays the currently valid incoming peer address. It may originate from non-volatile memory at boot time (AT+BTM<bd_addr>; S512>=3) or be set by AT+BTG/P<bd_addr>.

Response: <cr,lf>123456789012

<cr,lf>OK<cr,lf>

If no incoming peer address is currently set, the response is as follows:

Response: <cr,lf>000000000000

<cr,lf>OK<cr,lf>

2.5.10.7 AT+BTR<bd_addr> {Set Outgoing Peer Address for SPP, legacy SPP auto connect}

This command stores a peer address for outbound SPP connections in non-volatile memory.

This command sets up a module in pure cable replacement mode. If S512 = 1 and the peer address is NOT 000000000000, then it periodically (time specified via S505) attempts to connect to the specified peer address. All data from the host are then buffered in the receive buffer, until a Bluetooth connects and then sends the buffer across. This means if the peer device is not and will not be available and S507=1 or 2, the module effectively becomes useless and does not listen for commands arriving on the UART.

If this happens, two recovery methods are available. The first assumes that the DTR from the host is connected to the DSR line of the module and S507=1. The second assumes that this connection is absent and S507=1 or 2.

In the first method, deasserting the DTR line from the host aborts the autoconnect cycle. No “OK” is sent in response. The host must send a character regularly (e.g. one per second) until the module echoes all buffered characters to the host (provided echo is enabled) signifying it is in command mode.

The second method is to reset the device and ensure that the text string “AT+BT&BISM&<cr>” is sent (where <cr> is the carriage return character). There is special code which waits for this command and then terminates the autoconnect cycle. This function then sends an “OK” response.

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

2.5.10.8 AT+BTR {Delete Outgoing Peer Address}

This command deletes the peer address previously stored using AT+BTR<bd_addr>.

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

2.5.10.9 AT+BTR? {Read Outgoing Peer Address}

This command displays the peer address stored in non-volatile memory for placing the device in pure cable replacement mode.

Response: <cr,lf>12346789012

<cr,lf>OK<cr,lf>

If the location is empty the response is as follows:

Response: <cr,lf>00000000000

<cr,lf>OK<cr,lf>

2.6 Bluetooth Profiles

This section covers S-Registers and AT-Commands related to supported Bluetooth Profiles on BTM.

2.6.1 Profile Activation

To activate available profiles and advertise them to potential clients, S102 is used. Per default, only SPP is activated (value=1). Other supported profiles can be activated by setting the appropriate Flag in S102. Once S102 is written, the value must be saved to non-volatile memory (AT&W). Subsequently, a reset (ATZ) or power cycle is required. AT&W saves all S Registers to non-volatile memory.

2.6.2 SPP (Serial Port Profile)

The serial port profile (SPP) enables bidirectional serial data transmission with a remote device, like a wireless replacement for a serial cable.

SSP belongs to the group of serial stream oriented profiles (SSO) so refer to AT Commands for Serial Stream Oriented profiles (SSO) as well.

In order to use SPP, enable the profile in S102 (value=1). If it is not enabled prior, set S102 and then issue AT&W followed by ATZ.

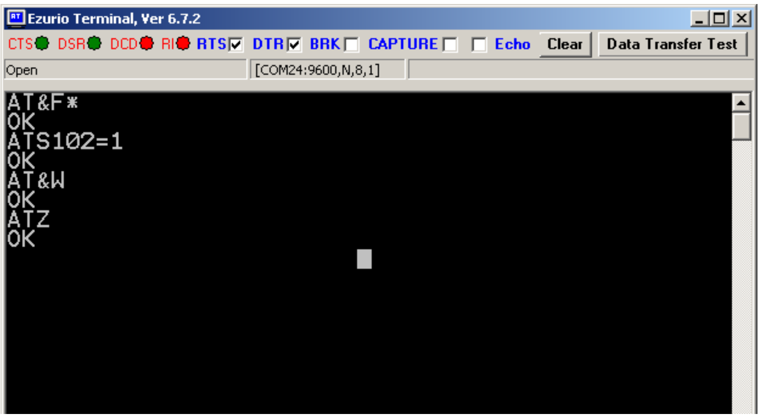
2.6.2.1 SPP example

This section explains how to make an SPP connection between two Laird BTM devices. This assumes devices A and B are connected to a terminal program e.g. Ezurio Terminal on a PC. The example sequence of AT commands is listed in Table 2-16. Figure 2-3: SPP example - Preparation of Device A - Figure 2-6 show the Ezurio Terminal.

Table 2-16: SPP Example Command Sequence

Phase	Dev.	AT Command	Comment
Preparation	A	AT&F*	Restore factory default settings
		ATS102=1	Enable Serial Port Profile (SPP)
		AT&W	Store settings
		ATZ	Reset
Preparation	B	AT&F*	Restore factory default settings
		ATS102=1	Enable Serial Port Profile (SPP)
		ATS0=1	Automatic response after one “RING”
		AT&W	Store settings
		ATZ	Reset
		AT+BTP	Make device temporary connectable and discoverable
		ATI4	Query Bluetooth device address of local device <BdAddr_DevB>

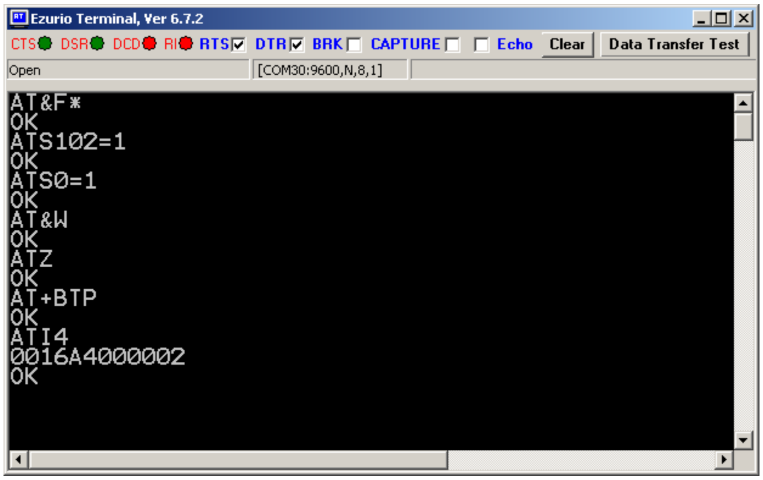
Phase	Dev.	AT Command	Comment
Initiate connection	A	AT+SPD<BdAddr_DevB>	Initiate SPP connection from device A to device B. Asynchronous messages: “PAIR 0…” (pairing successful, A and B) “RING…” (B only) “CONNECT…” (connected, A and B)
Connected	A,B	<data>	Any character entered on one end is displayed at the other end.
Enter command mode	A or B	^^^	Response OK : command mode confirmed, now AT commands are expected at the UART; UART data from host is not sent across to remote device
Disconnect		AT+SPH	Response NO CARRIER… (A and B): disconnection confirmed


Ezurio Terminal, Ver 6.7.2
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo Clear Data Transfer Test

Open [COM24:9600,N,8,1]

```
AT&F*
OK
ATS102=1
OK
AT&W
OK
ATZ
OK
```

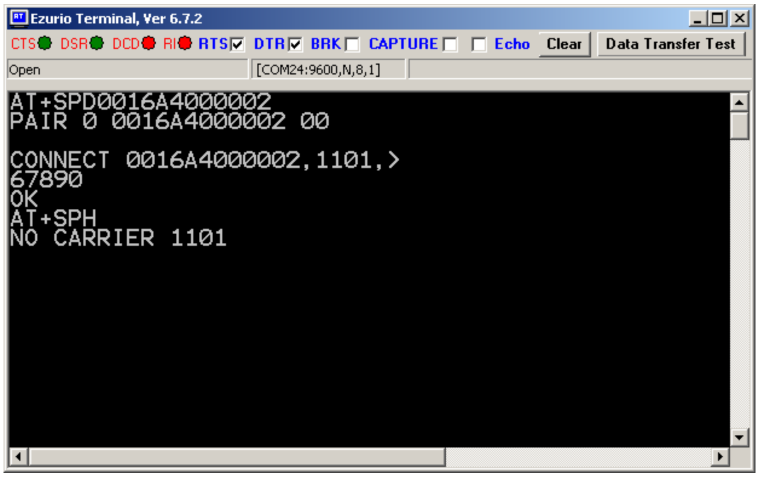
Figure 2-3: SPP example - Preparation of Device A


Ezurio Terminal, Ver 6.7.2
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo Clear Data Transfer Test

Open [COM30:9600,N,8,1]

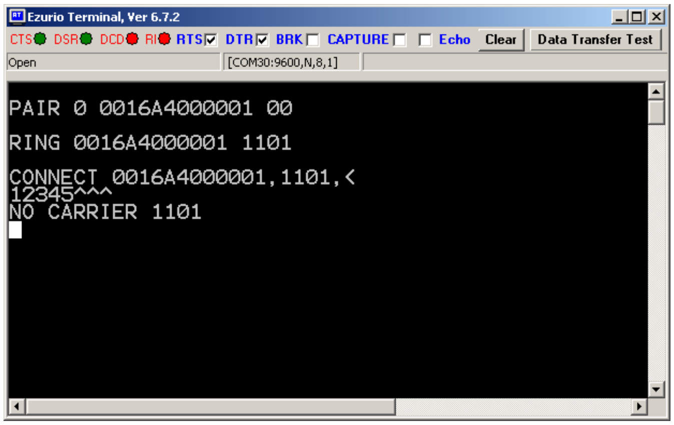
```
AT&F*
OK
ATS102=1
OK
ATS0=1
OK
AT&W
OK
ATZ
OK
AT+BTP
OK
ATI4
0016A4000002
OK
```

Figure 2-4 SPP example – Preparation of Device B


Ezurio Terminal, Ver 6.7.2
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo Clear Data Transfer Test
Open
[COM24:9600,N,8,1]

```
AT+SPD0016A4000002
PAIR 0 0016A4000002 00

CONNECT 0016A4000002,1101,>
67890
OK
AT+SPH
NO CARRIER 1101
```

Figure 2-5: SPP example Device A - initiate connection, receiving data, command mode, disconnect


Ezurio Terminal, Ver 6.7.2
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo Clear Data Transfer Test
Open
[COM30:9600,N,8,1]

```
PAIR 0 0016A4000001 00

RING 0016A4000001 1101

CONNECT 0016A4000001,1101,<
12345^^^
NO CARRIER 1101
```

Figure 2-6: SPP example Device B - incoming connection, receiving data, disconnection

2.6.2.2 ATA {Accept incoming SPP connection request}

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

Response: <cr,lf>CONNECT 123456789012,1101,<<cr,lf>

2.6.2.3 AT+SPD<bd_addr> {Make Outgoing SPP Connection}

Initiate a SPP connection to device with Bluetooth address <bd_addr> and SPP profile. The timeout is specified by S register 505.

For backward compatibility, the following command fulfils the same purpose: ATD<bd_addr>.

Response: <cr,lf>CONNECT 123456789012,1101,><cr,lf>

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

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

Response: <cr,lf>CALL LIMIT

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

In that case, issuing ATZ to reset the device resets the count to 0. More connections are made available.

2.6.2.4 AT+SPDL {Remake Connection}

Make a SPP connection with the same device as that specified in the most recent AT+SPD command. An error is returned if the 'L' modifier AND a Bluetooth address are specified.

For backward compatibility, the following command fulfils the same purpose: ATDL

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

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

2.6.2.5 AT+SPDR {Make SPP Connection to peer specified in AT+BTR}

Make a SPP connection with the device address specified in the most recent AT+BTR command. An error is returned if the 'R' modifier AND a Bluetooth address are specified.

For backward compatibility, the following command fulfils the same purpose: ATDR

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

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

2.6.2.6 AT+SPH {Drop SPP Connection}

Drop an existing SPP connection or reject an incoming connection indicated by unsolicited RING messages. For backward compatibility, the command ATH fulfils the same purpose.

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

2.6.2.7 SPP – Incoming Connections

The module can be configured using AT+BTP or AT+BTG to scan for incoming connections from other Bluetooth devices. It can also be configured via S512 to enter this mode by default on power up.

When the lower layers detect an SPP connection request, a RING 123456789012 string is sent to the host every second. The command ATA accepts the connection and ATH rejects the request.

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

CONNECT 123456789012,1101,<

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

If the S 100 register is non-zero, after ring indications are sent to the host and the host fails to accept or reject the connection, an automatic 'hangup' is executed.

2.6.2.8 SPP – Inquiry exception

Inquiry exception is a feature which allows to start an inquiry (AT+BTI…) if an SPP connection is established and in connected command mode or remote command mode. Normally this would not be permitted because it can lead to internal state issues.

However, now that this feature has been added, a number of possible issue cases were identified, fixed and manually tested (e.g. a disconnection occurs while an inquiry is running), but there may be untested cases left which can cause internal state issues or merged prints at the UART. This is why this feature is disabled per default and must be enabled by ATS383=1 (plus subsequent AT&W, ATZ).

2.6.2.9 SPP – Asynchronous Messages

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

CONNECT 123456789012,1101,<
An SPP connection with Bluetooth device 123456789012 is established successfully. The connection is initiated by the remote device (incoming).

CONNECT 123456789012,1101,>
An SPP connection has with Bluetooth device 123456789012 is successfully established. The connection is initiated by the local device (outgoing).

2.6.2.10 SPP – S Registers

S Registers for SPP are summarized in Table 2-17.

Table 2-17: S Registers for SPP

Register	Default	Range	Description
S0	0	-1..15	Number of RING indications before automatically answering an incoming connection. A value of 0 disables autoanswer. A value of -1 enables autoanswer on the first RING and disables sending a RING/CONNECT response to the host. This emulates serial cable replacement. Values >= 0 reset S Register 504 to 0. Values <0 set S504 to 1. If S0 <> 0 and S100 <> 0 then S0 must be set to < S100. If a value is entered which violates this rule, ERROR 29 is sent in response. If S504 =1, this returns -1, regardless of the value in non-volatile memory.
S100	15	0..15	Number of RING indications before auto disconnection. A value of 0 disables this feature. If S0 <> 0 and S100 <> 0 then S0 must be < S100. If a value is entered that violates this rule, ERROR 29 is sent in response.
S383	0	0..1	Enable inquiry exception for SPP link

2.6.3 A2DP (Advanced Audio Distribution Profile)

The "Advanced Audio Distribution Profile" is used for unidirectional transmission of high quality stereo audio streams between two Bluetooth devices. A2DP must be enabled by setting S102= 128. Also, an A2DP role must be assigned to either the source or sink module. Lastly, the settings must be stored by AT&W followed by ATZ.

An incoming A2DP connection request is accepted automatically if a valid link key for the paging device exists. If no link key is available, Secure Simple Pairing (SSP, BT2.1) or legacy pairing (BT2.0 or earlier) is carried out, depending on the Bluetooth version of the paging device.

After A2DP connection is established, the module remains in AT Command mode. S Register 531 is ignored for A2DP connections.

Version 1.2 of A2DP is supported.

The A2DP service record contains an optional field which indicates the A2DP supported features of a device. This feature field can be set by S312 for a sink and by S313 for a source. Refer to Table 2-24 for details.

Table 2-18 provides an overview on supported A2DP features on a module.

Table 2-19 lists supported A2DP codec types.

Table 2-18: A2DP Supported Features on BTM5xx

A2DP Feature		Support in SRC (Audio Source)		Support in SNK (Audio Sink)	
		Spec.	BTM5xx	Spec.	BTM5xx
1.	Audio Streaming	M	Yes	M	Yes
M: mandatory					

Table 2-19: A2DP Supported Codec Types on BTM5xx

A2DP Codec Type		Support in SRC (Audio Source)		Support in SNK (Audio Sink)	
		Spec.	BTM5xx	Spec.	BTM5xx
1.	SBC	M	Yes	M	Yes
2.	MPEG-1,2 Audio	O	No	O	No
3.	MPEG-2,4 AAC	O	No	O	No
4.	ATRAC family	O	No	O	No
M: O:	Mandatory Optional				

2.6.3.1 A2DP Example 1

This section gives an example of an A2DP connection between a Laird module as Audio Sink (wireless speaker/ wireless headphones) and a PC with a built in Bluetooth device and Toshiba Bluetooth Stack 2.1 as Audio Source.

The PC must support A2DP. If it is a different stack, the procedure must be similar and follow these steps:

1. Device discovery
2. Device Selection
3. Pairing
4. Connection establishment (e.g. initiated by PC)

If you have a Bluetooth 2.0 or earlier stack on your PC, legacy pairing with PIN occurs. In that case use AT+BTK="<PIN>" to enter the PIN on the module.

This assumes the BTM5xx device is connected to a terminal program e.g. Ezurio Terminal on a PC. The sequence of AT commands and the instructions for the PC side are listed in Table 2-20. Figure 2-7 through Figure 2-12 are screenshots of the process.

Table 2-20: A2DP Example 1 Command Sequence

Phase	Dev.	AT Command	Comment

Phase	Dev.	AT Command	Comment
Preparation	BTM5xx	AT&F*	Restore factory default settings A2DP profile is enabled per default in S102 Audio Sink role is enabled per default in S300
		ATS515=$040400	Set A2DP device class
		ATS512=4	Make device connectable and discoverable (permanent setting)
		AT&W	Store settings
		ATZ	Reset
Preparation / Connection setup	PC	n/a	1. Open "Bluetooth Settings" from the taskbar icon – Figure 2-7: A2DP example 1 – PC Bluetooth settings
			2. Click on "New Connection" - Figure 2-7: A2DP example 1 – PC Bluetooth settings
			3. Click Next (Discover BT Devices) - Figure 2-8
			4. Select "Laird BTMM…" if not listed: click "Refresh" - Figure 2-9
			5. Click "Yes" on "Allow this Device to connect?" - Figure 2-10.
			6. Connection is initiated by PC - Figure 2-11
			7. Open a player (e.g. Windows Media Player) to play some music
			8. Connect a Headphone/Speaker to the Audio Output of the BTM device
			Troubleshooting: If there is no audio, ensure that a Bluetooth Audio Device has appeared in the Device Manager. Also check that the Bluetooth Audio Device is selected as speaker in your player application. Sometimes a player needs to be closed and restarted in order to send its audio output to a new speaker device (which is the Bluetooth Audio Device here).
Connected	BTM5xx	n/a	Audio can be heard on the speakers/headphones
Adjust Volume		AT+GOU	Increment volume (audio output gain) by one
		AT+GOD	Decrement volume (audio output gain) by one
Disconnect		AT+APH	Response "NO CARRIER 110D": disconnection confirmed

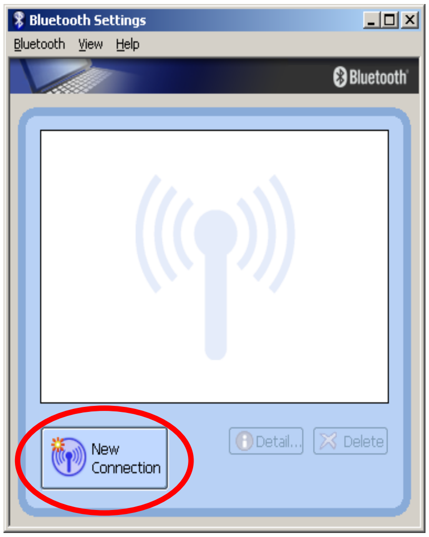
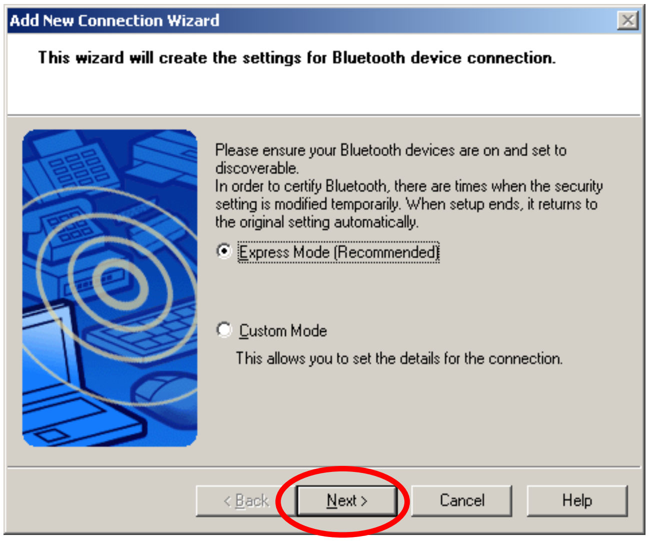


Figure 2-7: A2DP example 1 – PC Bluetooth settings



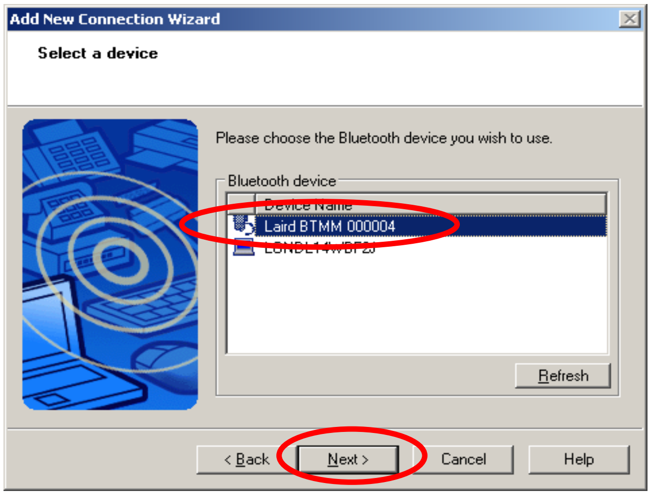
This wizard will create the settings for Bluetooth device connection.

Please ensure your Bluetooth devices are on and set to discoverable.
In order to certify Bluetooth, there are times when the security setting is modified temporarily. When setup ends, it returns to the original setting automatically.

● Express Mode (Recommended)

○ Custom Mode
This allows you to set the details for the connection.

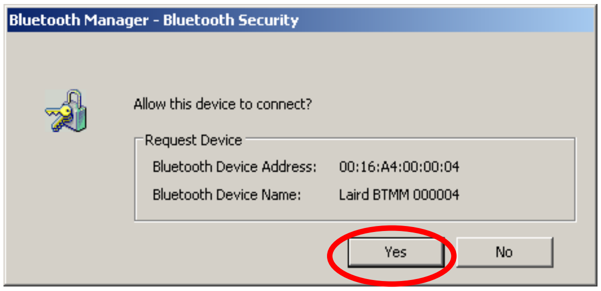
Figure 2-8: A2DP example 1 – Start discovery of Bluetooth devic



Please choose the Bluetooth device you wish to use.

Laird BTMM 000004

Figure 2-9: A2DP example 1 – Select Bluetooth device (Click "Refresh" if Laird BTMM is not listed)

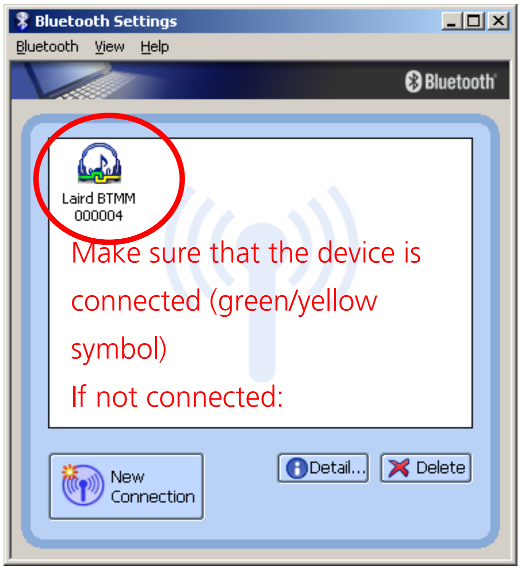


Allow this device to connect?

Bluetooth Device Address: 00:16:A4:00:00:04

Bluetooth Device Name: Laird BTMM 000004

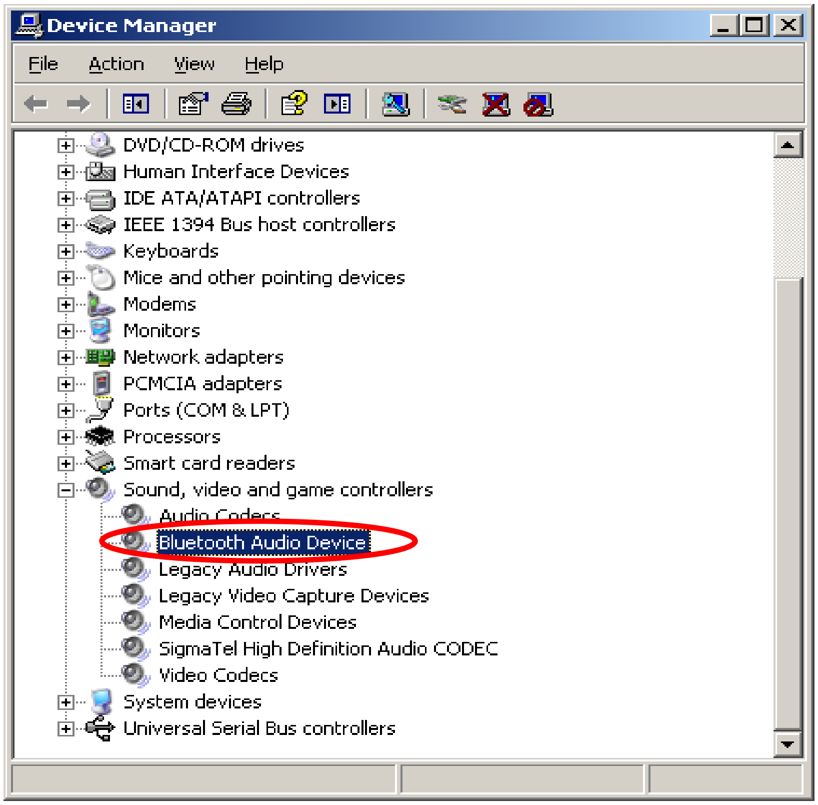
Figure 2-10: A2DP example 1 – Confirm pairing (here: Secure Simple Pairing, no PIN required)



Make sure that the device is connected (green/yellow symbol)

If not connected:

Figure 2-11: A2DP example 1: Connection established



- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Keyboards
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- PCMCIA adapters
- Ports (COM & LPT)
- Processors
- Smart card readers
- Sound, video and game controllers
  - Audio Codecs
  - Bluetooth Audio Device
  - Legacy Audio Drivers
  - Legacy Video Capture Devices
  - Media Control Devices
  - SigmaTel High Definition Audio CODEC
  - Video Codecs
- System devices
- Universal Serial Bus controllers

Figure 2-12: A2DP example 1: New Bluetooth Audio Device in the Device Manager

2.6.3.2 A2DP Example 2

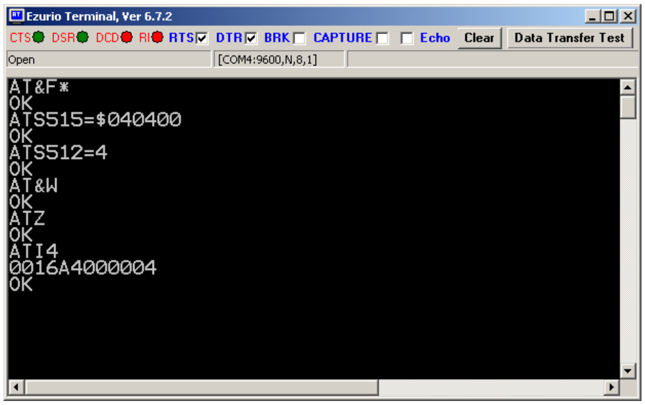
This section gives an example of an A2DP connection between a Laird BTM5xx device (A) as Audio Sink (wireless speaker/ wireless headphones) and another Laird BTM5xx device (B) as Audio source.

Both BTM5xx devices are assumed to be connected via UART to a terminal program e.g. Ezurio Terminal on a PC. The required instructions along with the sequence of AT commands are listed in Table 2-21. Figure 2-13 through Figure 2-16 show appropriate screenshots.

Table 2-21: A2DP Example 2 Command Sequence and Instructions

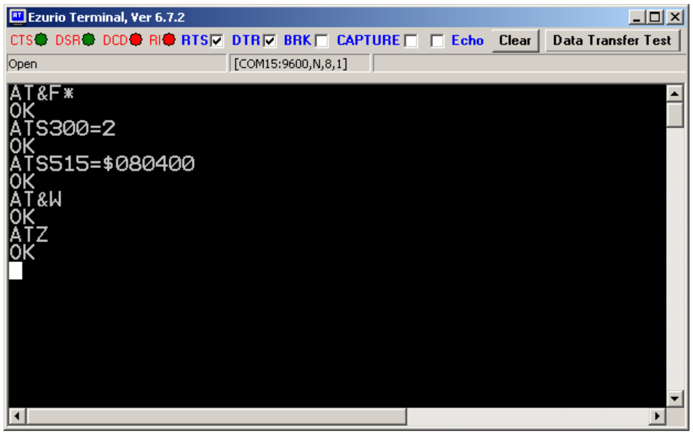
Phase	Dev.	AT Command	Comment
Preparation	ext. audio equipment.		1. Connect an audio source (e.g.MP3 player) to the stereo audio input (line in) of Device B.
			2. Connect a headphone or a speaker to the stereo audio output (line out) of device A
Preparation	A	AT&F*	Restore factory default settings A2DP profile is enabled per default in S102 Audio Sink role is enabled per default in S300
		ATS515=$040400	Set A2DP device class (Sink)
		ATS512=4	Make device connectable and discoverable (permanent setting)
		AT&W	Store settings
		ATZ	Reset
		ATI4	Query Bluetooth device address of Dev. A <BdAddr_DevA>
Preparation	B	AT&F*	Restore factory default settings A2DP profile is enabled per default in S102
		ATS300=2	Enable Audio Source role
		ATS515=$080400	Set A2DP device class (source)
		AT&W	Store settings
		ATZ	Reset
Initiate Connection	B	AT+APD<BdAddr_DevA>	**Response:** PAIR 0… (pairing successful, A and B) CONNECT… (connected, A and B)
Connected			Play music from the audio source Music should be audible on the headphones / speakers. If not, check the audio output device (e.g. connect headphones directly to audio source for a test)
Adjust Volume	A	AT+GOU	Increment volume (audio output gain) by one
		AT+GOD	Decrement volume (audio output gain) by one
	B	AT+GIU	Increment volume (audio input gain) by one
		AT+GID	Decrement volume (audio input gain) by one
			Note: the audio input gain (A2DP source) is a critical setting because the optimal setting cannot be verified by the module objectively.

Phase	Dev.	AT Command	Comment
Disconnect	A/B	AT+APH	Response NO CARRIER 110D: disconnection confirmed


Ezurio Terminal, Ver 6.7.2
CTS DSR DCD RI RTS☑ DTR☑ BRK□ CAPTURE□ □ Echo Clear Data Transfer Test
Open
[COM4:9600,N,8,1]

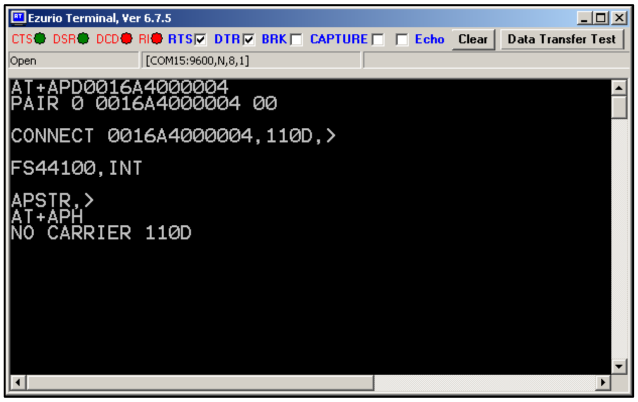
```
AT&F*
OK
ATS515=$040400
OK
ATS512=4
OK
AT&W
OK
ATZ
OK
ATI4
0016A4000004
OK
```

Figure 2-13: A2DP Example 2 – Preparation of Device A (Sink)


Ezurio Terminal, Ver 6.7.2
CTS DSR DCD RI RTS☑ DTR☑ BRK□ CAPTURE□ □ Echo Clear Data Transfer Test
Open
[COM15:9600,N,8,1]

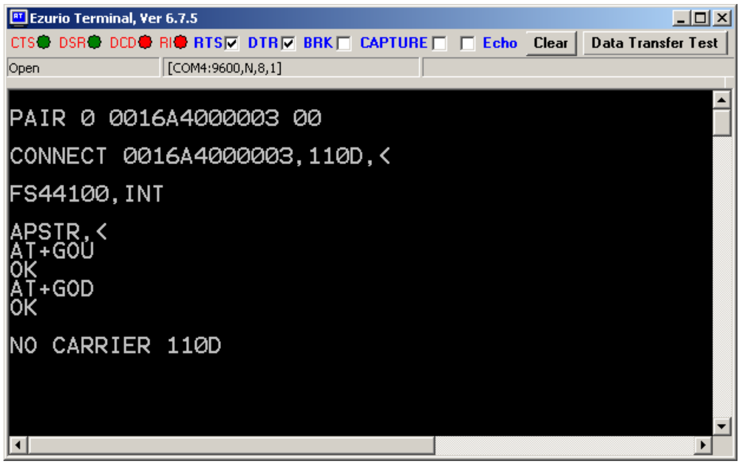
```
AT&F*
OK
ATS300=2
OK
ATS515=$080400
OK
AT&W
OK
ATZ
OK
```

Figure 2-14: A2DP Example 2 – Preparation of Device B (Source)


Ezurio Terminal, Ver 6.7.5
CTS DSR DCD RI RTS☑ DTR☑ BRK□ CAPTURE□ □ Echo Clear Data Transfer Test
Open
[COM15:9600,N,8,1]

```
AT+APD0016A4000004
PAIR 0 0016A4000004 00

CONNECT 0016A4000004,110D,>

FS44100,INT

APSTR,>
AT+APH
NO CARRIER 110D
```

Figure 2-15: A2DP Example 2 –Initiate and Release Connection from Device B (Source)


Ezurio Terminal, Ver 6.7.5
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo
Clear
Data Transfer Test
Open
[COM4:9600,N,8,1]

```
PAIR 0 0016A4000003 00

CONNECT 0016A4000003,110D,<

FS44100,INT

APSTR,<
AT+GOU
OK
AT+GOD
OK

NO CARRIER 110D
```

Figure 2-16: A2DP Example 2 – Accepting Connection and Volume Adjustment Device A (Sink)

2.6.3.3 Enable A2DP

The advanced audio distribution profile (A2DP) is enabled by issuing ATS102=128. After verifying that a role is set (S300 != 0) the S registers must be saved by AT&W, and then issue ATZ to make the new settings effective. If the A2DP flag in S102 (0x80) is not set after these steps, then S300 was = 0 at boot time, which is invalid for A2DP.

2.6.3.4 Select A2DP Role

The A2DP role is selected by S register 300 as follows:

0 – No role selected
1 – A2DP Sink (default)
2 – A2DP Source

The setting must be saved by AT&W and takes effect on next power cycle (ATZ).

CAUTION: If S300=0 at boot (no role selected) S102 is set with the A2DP flag (0x80), the A2DP flag is cleared automatically. A2DP cannot be enabled without a role selected.

2.6.3.5 Set A2DP Device Class

ATS515=$<device_class$_{hex}$>; AT&W; ATZ

For compliance with the A2DP specification [2] (and hence for successful interoperability to other devices) you must set up a valid device class code. The default device class code of a module is 0x001F00, which is invalid for A2DP.

The A2DP specification [2] mandates the following requirements for the device class of an A2DP device:

- Mandatory to set the 'Rendering' bit for the SNK and the 'Capturing' bit for the SRC in the Service Class field.
- Recommended to set 'Audio/Video' as Major Device class both for the SNK and the SRC.
- Select the appropriate Minor Device class as defined in the Bluetooth Assigned Numbers [8].

There is a tool available online named Class of Device generator, made for creating a particular device class code: refer to [9].

With the assumptions above (Major Device class = Audio/Video), Table 8 in [8] gives the complete list of codes for the minor device class. If you are not sure about the minor device class, use the row marked with n/a for the minor device (e.g. 0x040400 for sink or 0x080400 for source).

Table 2-22 provides examples of device class codes for A2DP devices.

Table 2-22: A2DP device class code – examples

Device Class Code	Major Service Class	Major Device Class	Minor Device	A2DP Role
0x040400	Rendering (1)	Audio/Video (3)	n/a	Sink
0x040414	Rendering (1)	Audio/Video (3)	Loudspeaker	Sink
0x040418	Rendering (1)	Audio/Video (3)	Headphones	Sink
0x04041C	Rendering (1)	Audio/Video (3)	Portable Audio	Sink
0x040420	Rendering (1)	Audio/Video (3)	Car audio	Sink
0x080400	Capturing (2)	Audio/Video (3)	n/a	Source
0x080410	Capturing (2)	Audio/Video (3)	Microphone	Source
0x080428	Capturing (2)	Audio/Video (3)	Hi-Fi Audio Device	Source

(1) the "rendering" flag (0x040000) is mandatory for an A2DP SNK
(2) the "capturing" flag (0x080000) is mandatory for an A2DP SRC
(3) "Audio/Video" major device class (0x000400) is recommended for an A2DP device
Refer to A2DP 1.2 specification, section 5.5.1 [2].

The device class is written to the module using ATS515=$<device_class$_{hex}$> where <device_class$_{hex}$> is the 6 character device class code without leading 0x. Use subsequent AT&W and ATZ to commit the new value. Also refer to AT+BTC<devclasshex>   {Set Device Class Code Temporarily}.

2.6.3.6 Initiate A2DP Connection

AT+APD<bd_addr$_{hex}$>

Initiate A2DP connection to Bluetooth address <bd_addr$_{hex}$>. The remote device must support the complementary role to the local device. If link keys are missing in one or both devices, pairing either occurs automatically or must be initiated by AT+BTW<bd_addr$_{hex}$>. This depends on factors like the combination of local and remote IO capabilities or the Bluetooth version of the remote device.

Response:

<cr><lf>PAIR 0 <bd_addr> 00 3<cr><lf>   (first time only, auto pairing without MITM authent.)
<cr><lf>CONNECT <bd_addr>,110D,><cr><lf>
Or:
<cr><lf>NO CARRIER 110D<cr><lf>   (not successful)

2.6.3.7 Output Gain Settings – A2DP Sink

AT+GOU / AT+GOD / AT+GO<n>

AT+GOU – Increment audio output gain (volume).
AT+GOD – Decrement audio output gain (volume).
AT+GO<n> – Set absolute output gain <n>, range = 0..15 (dec)

Response:

<cr><lf>OK<cr><lf>
<cr><lf>ERROR 57<cr><lf> – Maximum gain level reached
<cr><lf>ERROR 58<cr><lf> – Minimum gain level reached

S register 589 can set the output gain level directly. Alternatively, S register 689 may set the required overall output gain in dBr multiplied by 10. Refer to Onboard Codec Gain.

When a DSP application is running (typically for audio links), the maximum output gain is 15. This is due to DSP operation applying to all audio links.

2.6.3.8 Input Gain Settings – A2DP Source

AT+GIU / AT+GID / AT+GI<n>

AT+GIU – Increment audio input gain.
AT+GID – Decrement audio input gain.
AT+GI<n> – Set absolute input gain <n>, range = 0..15 (dec)

Response:

<cr><lf>OK<cr><lf>
<cr><lf>ERROR 57<cr><lf> – Maximum gain level reached
<cr><lf>ERROR 58<cr><lf> – Minimum gain level reached

S Register 590 can set the input gain level directly. Alternatively, S register 690 can set the required overall input gain in dBr multiplied by 10. Refer to Onboard Codec Gain.

2.6.3.9 Release A2DP Connection

AT+APH / ATH110D

Release an A2DP connection by AT+APH. Alternatively, you may use ATH110D. (110D presents the UUID for the A2DP profile). Refer to Disconnecting Profiles for more information.

2.6.3.10 Suspend A2DP Stream

AT+APU

When creating an A2DP link, the module enters the streaming state normally. Streaming can be manually suspended using AT+APU. The A2DP link is retained. The asynchronous message APSUS,> confirms that streaming is suspended and that it was initiated locally ('>'). Refer to A2DP streaming state.

Response:

<cr><lf>OK<cr><lf>	(command accepted)
<cr><lf>APSUS,><cr><lf>	(A2DP stream suspended, locally initiated)
or	
<cr><lf>ERROR 77<cr><lf>	(wrong A2DP state)

2.6.3.11 Resume A2DP Stream

AT+APR

A suspended A2DP stream can be resumed manually by AT+APR. The asynchronous message APSTR,> confirms that streaming is resumed and that it was initiated locally ('>'). Refer to A2DP streaming state.

Response:

<cr><lf>OK<cr><lf>	(command accepted)
<cr><lf>APSTR,><cr><lf>	(resuming A2DP stream, locally initiated)
or	
<cr><lf>ERROR 77<cr><lf>	(wrong A2DP state)

2.6.3.12 Supported Features – A2DP Sink

S312

Use S Register 312 to set supported features in the A2DP sink that are advertised in the A2DP service record. Refer to Table 2-24. The S register needs to be saved (AT&W) and the module needs to be power cycled (ATZ) for a new value to become effective.

2.6.3.13 Supported Features – A2DP Source

S313

Use S Register 313 to set the A2DP source supported features that are to be advertised in the A2DP service record. Refer to Table 2-24. The S register must be saved (AT&W) and the module must be power cycled (ATZ) for a new value to become effective.

2.6.3.14 A2DP Audio Stream Routing

S314

By default, A2DP audio is routed to the internal codec of the BTM5xx module, i.e. its onboard analogue audio inputs or outputs. S314 permits you to use a digital audio bus, such as I2S for A2DP audio, as follows:

S314: 0=internal (default), 1=I2S master, 2=I2S slave

Refer to Table 2-24 and Digital Audio Interface (including important information regarding sampling rate capabilities of devices connected to the I2S bus).

2.6.3.15 A2DP Codecs: SBC vs. APTx / AAC

The standard codec for A2DP is the Sub Band Codec (SBC). Here, "codec" refers to DSP algorithms which encode an audio stream for transmission over a limited bandwidth link and which decode the compressed data on the receiver side [1].

One advantage of SBC is wide support by all A2DP devices, because SBC is a mandatory feature of the A2DP specification. One disadvantage is the significant latency (around 1s) needed to encode, transmit

[1] A different definition of the term *Codec* is an integrated circuit (IC) composed of ADC and/or DAC, combined with a sampling rate clock and anti-aliasing filters. This definition is referred to when talking about "external audio codec" or "internal codec" in terms of analogue-to-digital conversion or digital-to-analogue conversion of an audio stream.

and decode an audio stream. Due to this, SBC is not ideal for applications that require synchronised audio, e.g. wireless headphones for TV.

However, there are optional codecs to overcome this limitation of SBC, such as APTx. APTx claims low latency and much better audio transparency than SBC. This may of course be evaluated by measurements, but the judgment is ultimately subjective and up to the listener.

Another optional codec is AAC. AAC is implemented for an A2DP sink only (decoder). It has the preferred codec for the iPhone.

An optional codec must be supported by both ends of an A2DP link. At the beginning of each A2DP link a negotiation takes place. In the result of this negotiation both devices agree on the codec and the sampling rate to be used for the time of the connection. If no optional codec matches on both ends, SBC is used. Multiple optional codecs can be enabled.

2.6.3.16 Enabling optional codecs (APTX, AAC)

S307

S307 enables optional codecs. Bit 0 (val=1) enables APTX (source or sink) and bit 1 (val=2) enables AAC (sink only). If AAC is enabled but the module is configured for A2DP source, the AAC bit is ignored. Multiple optional codecs can be enabled, e.g. S307=3 enables AAC and APTX.

S register 307 is queried by firmware at boot time. For a new setting to become effective, you must send AT&W (save SRegs) and ATZ (reset) to the module. APTx or AAC must be enabled on both ends of the A2DP link or it falls back to SBC. After connection, the current codec may be verified by ATI26.

2.6.3.17 Discoverable/Connectable while A2DP streaming

In a scenario where both devices of an A2DP link are discoverable and connectable while streaming audio, interrupts in the audio stream are likely to occur. Bandwidth which would be required for the A2DP stream seems to be consumed by inquiry-scanning (discoverable) and page-scanning (connectable) processes. Therefore the number of slots available for the A2DP link is too low and the audio stream is interrupted.

One workaround is to make the module neither discoverable nor connectable when in an A2DP connection using AT+BTX. It has been observed that sometimes only disabling either connectable or discoverable creates an interrupt free audio stream. AT commands to achieve this are AT+BTQ (discoverable only) and AT+BTG (connectable only).

S308 provides an automatic workaround solution: whenever entering an A2DP connection, it disables connectable and discoverable mode, like AT+BTX. Whenever exiting an A2DP connection, connectable and discoverable states are restored according to the value of S512. Each action can be enabled/disabled in S308, refer to Table 3-1.

2.6.3.18 A2DP streaming state

When initiating an A2DP link successfully, the streaming state is normally entered immediately to transmit audio. But there are scenarios where the A2DP link should be retained but streaming should be suspended, because the audio resources are temporarily needed for other purposes, e.g. an incoming call over HFP. A number of asynchronous messages indicate any change of the A2DP streaming state (APSTR/ APSUS). Refer to A2DP - Asynchronous messages. Query the status of an A2DP connection with ATI61. Refer to Table 3-2.

2.6.3.19 Simultaneous A2DP and HFP

One scenario of particular interest is a headset scenario with A2DP (sink) enabled for receiving stereo audio and HF-unit enabled for receiving/making calls. The A2DP source and the HFP gateway can be combined in one device (e.g. a mobile phone), but they could also be different devices (e.g. an MP3 player as A2DP source and a mobile phone as HFP gateway). For the first case, we assume that a mobile phone will suspend the A2DP stream before enabling the SCO connection for speech. Once the call is finished, the mobile phone is expected to initiate resumption of A2DP streaming. The only user action required is to answer the call at the BTM5xx module by AT+HFCA (or an appropriate GPIO button assigned to the HF green button). Asynchronous messages related to suspend/resume actions are listed below (HF messages are not shown here):

APSUS,<	(A2DP stream suspended, initiated remotely)
FS8000,INT	(8kHz sampling rate on internal codec)
HF"AU1"	(SCO connection)
…	(ongoing call)
HF"AU0"	(SCO disconnected)
FS44100,INT	(44.1 kHz sampling rate on internal codec)
APSTR,<	(resuming A2DP stream, initiated remotely)

In the second scenario, where the audio source and the HF gateway are different devices, you must take more precautions. Neither device knows if the audio resource of the A2DP sink / HF unit is in use when one requests an audio link. For example, if the mobile phone signals an incoming call and requests a SCO link while A2DP is being streamed from another device, the mobile phone is unaware of A2DP stream and so cannot request its suspension. In this case, the audio stream must be suspended manually by a host microcontroller (AT+APU) before accepting the call. Once the call finishes, the audio stream should be resumed manually by a host microcontroller (AT+APR).

S-register 355 ("audio resource override") helps to address and automate this scenario. It controls whether incoming/outgoing requests for A2DP or SCO connections are accepted by requiring that a link of another type be suspended (A2DP) or closed (SCO) before accepting/initiating the new link. See Table 3-1 for details of S355.

In an A2DP / HFP simultaneous scenario, it is likely that the user wants different gain settings for SCO and A2DP links. ATI22 (for output) and ATI23 (for input) query the last gain settings used for a SCO connection of the internal codec. ATI24 (for output) and ATI25 (for input) query the last gain settings used for an A2DP connection. S register 356 restores last SCO gain settings when entering a SCO connection. S-register 357 restores the last A2DP gain settings when entering an A2DP connection. Refer to Table 3-1 for details.

2.6.3.19.1 Issue/Workaround with Apple Devices

When initiating A2DP from BTM51x with AAC enabled to an iPhone, we have observed regular short drops that can be resolved by changing the HCI role so that the iPhone becomes master of the link. The setting to request slave role for BTM51x on each A2DP and AVRCP connection is:

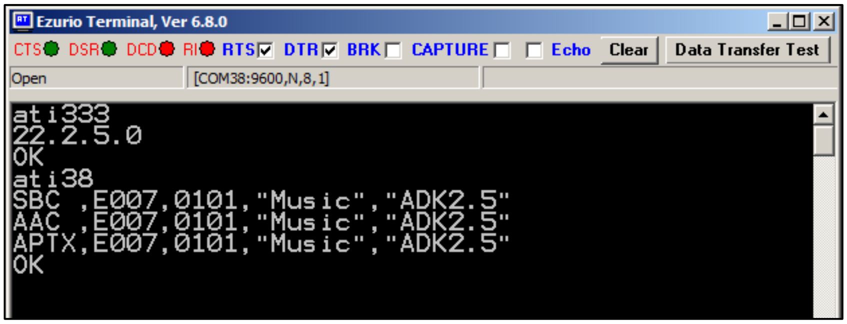
```
ATS368=$0180

AT&W

ATZ
```

See AT+SR<role>,<pm> {Setting HCI role (master / slave)} for more details on HCI roles.

2.6.3.20 A2DP decoder post processing (MusicManager)

As of firmware v18.1.4.0 all A2DP decoders (SBC, AAC, APTX) support DSP post processing to set up parametric equalizers, 3D stereo enhancement, compander, BassBoost etc. CSR's tool UniversalFrontEnd (UFE) / Music Manager GUI allows to set parameters and hearing changes immediately. It requires an SPI connection to BTM51x via CSR's USB-SPI interface or LPT port interface as well as the correct version of UFE/Music Manager.

Check ATI38 to find the correct version (Figure 2-17). The last part ADK2.0 indicates that the correct version of UFE/Music Manager for these decoders is contained in CSR ADK2.0. Laird may provide this tool under certain conditions.


Ezurio Terminal, Ver 6.8.0
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo
Clear
Data Transfer Test
Open
[COM38:9600,N,8,1]

```
ati333
22.2.5.0
OK
ati38
SBC ,E007,0101,"Music","ADK2.5"
AAC ,E007,0101,"Music","ADK2.5"
APTX,E007,0101,"Music","ADK2.5"
OK
```

Figure 2-17: ATI38

2.6.3.21 Connecting to UFE

Create an A2DP connection with BTM5xx as sink to start streaming, e.g. playing music from a phone. As soon as "APSTR,<" appears on the module's UART, you may connect to the DSP application loaded. UFE queries the SysID and BuildNo of the DSP application over SPI. You may only connect if the SysId is correct (0xE006).

After UFE has successfully connected, in the upper left corner MusicManager… appears followed by SysId ("vE006" – hexadecimal), BuildNo (b101 – decimal) and current sampling frequency. Enable monitoring mode by either clicking **Menu > DSP > Monitor DSP** or the glasses button.

In monitoring mode, parameters are polled from the DSP at regular intervals. This allows monitoring of MIPS performance of the decoder (Decoder MIPS) and MIPS of post processing functions (Function MIPS). In addition, you may monitor signal peaks at four points. Another function allows you to set the current mode.

2.6.3.22 Post processing modes

One out of five modes can be selected by UFE or by AT commands. As outlined in Table 2-23, only Modes 1 (Pass Thru) and 2 (Full) are sensible to A2DP stereo applications (highlighted in table 3-16). However, modes 3 and 4 can be utilised for custom post processing algorithms.

Table 2-23: A2DP decoder post processing modes

Mode Index (AT+APMx)	Mode (UFE)	Comment
0	Stand By	Not useful for A2DP stereo application
1	PassThru	Stereo, post processing disabled, default mode for SBC/AAC decoder
2	Full	Stereo, post processing enabled, such as PEQ

Mode Index (AT+APMx)	Mode (UFE)	Comment
		(parametric equalizer),Stereo 3D enhancement, Compander etc., stereo
3	Mono	Mono, not useful for A2DP stereo application, can be replaced by custom post processing algorithms
4	MonoPassThru	Mono, not useful for A2DP stereo application, can be replaced by custom post processing algorithms

2.6.3.23 AT+ACS? {Query post processing mode}

AT+APM? – query current A2DP decoder post processing mode

Response:

- 1 = pass-thru mode (no post processing).
- 2 = full post processing mode (CSR music example / Music Manager).
- ERROR 93 – A2DP not in streaming state or not in A2DP sink role (decoder).
- ERROR 94 – command not supported by current decoder plugin.

2.6.3.24 AT+APMx {Set post processing mode}

AT+APMx – set current A2DP decoder post processing mode, with x='0'..'5':

- 1 = pass-thru mode (no post processing).
- 2 = full post processing mode (CSR music example).
- Values other than listed above (0,3,4,5) should not be set because no useful function is assigned.
- The command is only accepted in A2DP streaming state; otherwise ERROR 93 is returned.

2.6.3.25 S363 {Set default post processing mode}

S363 – set decoder post processing default mode [1..3]:

Whenever A2DP enters the streaming state (i.e. a decoder is loaded on the DSP), the post processing mode is set automatically as per S363.

Values:

- 1 = pass-thru mode (no post processing)
- 2 = full post processing mode (CSR music example)
- 3 = reserved for custom post processing algorithms

Default value: 1 (pass-thru, no post processing)

If a loaded A2DP decoder doesn't support post processing, S363 is ignored. As of v18.1.4.0, all A2DP decoders (SBC, AAC, APTX) support post processing.

2.6.3.26 A2DP - Asynchronous messages

CONNECT 123456789012,110D,<

An A2DP connection with Bluetooth device address 123456789012 is established. The connection was initiated by the remote device (incoming)

CONNECT 123456789012,110D,>

An A2DP connection with Bluetooth device address 123456789012 is established. The connection was initiated by the local device (outgoing)

FS44100,INT

The internal codec is configured with a sampling frequency of 44100 Hz.

FS44100,I2S_M

The I2S bus is configured as I2S master with a sampling frequency of 44100 Hz.

FS44100,I2S_S

The I2S bus is configured as I2S slave with a sampling frequency of 44100 Hz.

APSUS,<

Transition to A2DP suspended state, initiated by remote device (incoming).

APSUS,>

Transition to A2DP suspended state, initiated by local device (outgoing).

APSTR,<

Transition to A2DP streaming state, initiated by remote device (incoming).

APSTR,>

Transition to A2DP streaming state, initiated by local device (outgoing).

NO CARRIER 110D

An existing A2DP connection has been terminated or an A2DP connection attempt has failed.

Table 2-24: A2DP – S Registers and AT-Commands

Task	AT-Command / SRegister	Comment
Enable A2DP profile	S102	128 = A2DP, Error 46 if A2DP role has not been set (see S300). Needs subsequent AT&W and ATZ to become effective
Set A2DP role	S300 [0..2]	0 Feature not set 1 A2DP Sink (default) 2 A2DP Source Needs subsequent AT&W and ATZ to become effective
Initiate outgoing A2DP connection	AT+APD<bd_addr>	Response if accepted CONNECT 0123456789012,110D,> Response if rejected NO CARRIER 110D
close only A2DP connection	"AT+APH" or "ATH110D"	**Response:** If connection has existed and S329 = 0 NO CARRIER 110D If connection has not existed and S329 = 0 NO CARRIER
close all connections	ATH*	**Response:** For each profile that was previously connected NO CARRIER <profileUUID> See Disconnecting Profiles.
Set gain level	S589 [0..22],	Set codec output gain level (applies to sink).

Task	AT-Command / SRegister	Comment	
	default = 12 or: AT+GO<n>	Range for <n> = 0…15 (dec)	
	S590 [0..22], default = 12 Or: AT+GI<n>	Set codec input gain level (applies to source). Range for <n> = 0…15 (dec)	
Set overall gain (dBr * 10)	S689 [-450..215]	Set codec output gain in dBr * 10 (applies to sink) Default: 0	
	S690 [-450..215]	Set codec input gain in dBr * 10 (applies to source), Default: 0	
Increment Gain	AT+GOU	Increment codec output gain by 1 step in gain table (refer to A2DP Example 1).	
	AT+GIU	Increment codec input gain by 1 step in gain table (refer to A2DP Example 1).	
Decrement Gain	AT+GOD	Decrement codec output gain by 1 step in gain table (refer to A2DP Example 1).	
	AT+GID	Decrement codec input gain by 1 step in gain table (refer to A2DP Example 1).	
Auto-BTX – Automatic control of discoverable/connectable mode when entering and exiting various profiles (Auto-BTX)	S308 [0..15]	0	Make the module not discoverable and not connectable when entering a connection.
		1	Restore discoverable/connectable state as defined by S512 when exiting a connection.
		3,2	Multi-profile policy: ▪ 00b – A2DP only: Bits 0 and 1 apply only for A2DP connections (backward compatible) ▪ 01b – Enter auto-BTX when ALL profiles of S102 are connected, restore to S512 as soon as ONE profile is disconnected. ▪ 10b – Enter auto-BTX when ONE profile is connected, restore to S512 as soon as ALL profiles are disconnected (default) ▪ 11b – Enter auto-BTX when ALL profiles of S102 are connected, restore S512 as soon as ALL profiles are disconnected.
		Note: The S308 default value of 11 makes the BTM51x not discoverable and not connectable after the first profile is connected. If more than one incoming connection is expected, then S308 must be changed appropriately. Also refer to Discoverable/Connectable while A2DP Streaming.	

Task	AT-Command / SRegister	Comment
Set A2DP Sink supported features bit mask	S312 [1..15]	Bitmask - sink supported features: 0 Headphone 1 Speaker (default) 2 Recorder 3 Amplifier Subsequent AT&W plus ATZ required for a new value to become effective
Set A2DP Source supported features bit mask	S313 [1..15]	Bitmask - source supported features: 0 Player 1 Microphone (default) 2 Tuner 3 Mixer Subsequent AT&W plus ATZ required to commit value.
Enable optional A2DP codec	S307 [0..3]	0 No optional codec (default) 1 APTx (Bit 0) 2 AAC (Bit 1, sink only) 3 Both APTx and AAC enabled Subsequent AT&W plus ATZ required to commit value.
Configure audio resource override	S355 [0..3]	0 No override allowed 1 Incoming/outgoing SCO request is accepted while A2DP is streaming. The module initiates A2DP suspend automatically (default). 2 Incoming/outgoing A2DP start/resume request is accepted while SCO is active. The module automatically releases the SCO link. 3 Both 1 and 2 are enabled. Refer to Simultaneous A2DP and HFP as well.
Enable restoring of last gain used for SCO	S356 [0..3]	0 Disable gain restoring for SCO 1 Enable output gain restoring for SCO 2 Enable input gain restoring for SCO 3 Enable input and output gain restoring for SCO (default)
Enable restoring of last gain used for A2DP	S357 [0..3]	0 Disable gain restoring for A2DP 1 Enable output gain restoring for A2DP 2 Enable input gain restoring for A2DP 3 Enable input and output gain restoring for A2DP (default)
Check codec in use	ATI26	**Response:** If S333 = 1 NONE/SBC/APTX If S333 = 0 0 – None 1 – SBC 2 – APTX

Task	AT-Command / SRegister	Comment	
Check current discoverable / connectable status (scan state)	ATI27	0	Not discoverable, not connectable (not scanning)
		1	Discoverable (inquiry scanning)
		2	Connectable (page scanning)
		3	Discoverable and connectable (inquiry and page scanning)
Query A2DP status	ATI61	0	A2DP not connected
		1	A2DP connected and streaming
		2	A2DP connected but streaming suspended
Query SCO token	ATI21	Index of the profile instance owning an active SCO connection:	
		0	None
		1	HS
		2	HSG
		3	HF
		4	HFG
		5	SPP_BTA
Query last SCO output gain	ATI22	For values [0..22] refer to Table 2-68.	
Query last SCO input gain	ATI23	For values [0..22] refer to Table 2-68.	
Query last A2DP output gain	ATI24	For values [0..22] refer to Table 2-68.	
Query last A2DP input gain	ATI25	For values [0..22] refer to Table 2-68.	
Query current post-processing mode	AT+APM?	As of v18.1.4.0 for A2DP decoder (SBC/AAC/APTX) in streaming state).	
Set current post-processing mode	AT+APMx	Range: x=['0'..'6'] For A2DP decoder in streaming state only.	
Set default post processing mode	S363 [1..3]	Whenever an A2DP decoder, capable of post processing modes is loaded, the mode defined by S363 is entered. Refer to S363 {Set default post processing mode} Range: [1..3], default value = 1	
Query available A2DP decoders capable of post processing	ATI38	Display Type, SysID, BuildNo, FriendlyName and CSR SDK of all available A2DP decoders capable of post processing.	

2.6.4 AVRCP 1.0 (Audio Video Remote Control Profile)

The Audio/Video Remote Control profile remotely controls audio or video streaming devices. A device must be defined as either control (CT) or target (TG). Furthermore, you must assign a device to one of four categories (Player/Recorder, Monitor/Amplifier, Tuner, Menu).

Version 1.0 of AVRCVP is supported for TG role. CT role also supports AVRCP v1.5 which is backward compatible with version 1.0. All AT-commands and messages for AVRCP v1.5 – CT are described in the separate section <AVRCP1.5>

The AVRCP specification [3] adopts the AV/C Digital Interface Command Set (AV/C command set, defined by the 1394 Trade Association) device model and control message format. In this device model a remote control target consists of one or more subunits. At least the subunit PANEL must exist. Remote control commands are passed to a subunit with the PASS THROUGH command. A BTM device configured as target supports one PANEL-subunit.

Table 2-25 provides an overview on supported AVRCP features on a module.

Table 2-25: AVRCP Supported Features on BTM5xx

AVRCP Feature		Support in CT (Control)		Support in TG (Target)	
		Spec.	**BTM5xx**	**Spec.**	**BTM5xx**
1.	Connection establishment for control	M	Yes	O	Yes
2.	Release connection for control	M	Yes	M	Yes
3.	Sending UNIT INFO command	O	Yes	X	No
4.	Receiving UNIT INFO command	X	No	M	Yes
5.	Sending SUBUNIT INFO command	O	(1)	X	No
6.	Receiving SUBUNIT INFO command	X	No	M	Yes
7.	Sending VENDOR DEPENDENT command	O	No	X	No
8.	Receiving VENDOR DEPENDENT command	X	No	O	No
9.	Sending PASS THROUGH command	M	Yes	X	No
10.	Receiving PASS THROUGH command	X	No	M	Yes

M – Mandatory   O – Optional   X – Excluded   (1) – Incomplete

2.6.4.1 AVRCP Example 1

This section illustrates an AVRCP connection between a Laird BTM5xx device as AVRCP Controller and a PC with a built in Bluetooth device and Toshiba Bluetooth Stack 2.1 as AVRCP Target.

For any other Bluetooth Stack with AVRCP (target role) support, the setup should follow these steps:

1. Identify the Bluetooth Device Address of the PC.
2. Enable the AV Remote Control Service.
3. Select the player and/or setup display of incoming remote control commands.

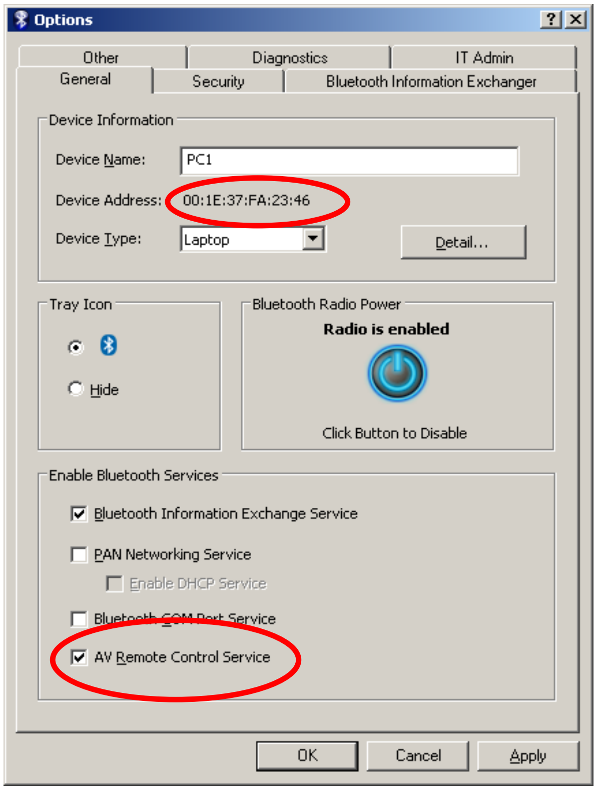
The BTM5xx device is assumed to be connected to a terminal program e.g. Ezurio Terminal on a PC. The sequence of AT commands and the instructions for the PC side are listed in Table 2-26. Figure 2-18 to Figure 2-23 show appropriate screenshots.

This example can be combined with the A2DP Example 1 (A2DP Example 1). Then the AVRCP connection should be initiated after A2DP is connected.

Table 2-26: AVRCP Example 1 Command Sequence

Phase	Dev.	AT Command	Comment
Preparation	BTM5xx	AT&F*	Restore factory default settings ▪ AVRCP profile is enabled per default in S102 ▪ AVRCP Control role is enabled per default in S301
		ATZ	Reset
Preparation	PC	n/a	1. Select **Options…** from the Bluetooth icon in the

<table>
<tr><th>Phase</th><th>Dev.</th><th>AT Command</th><th>Comment</th></tr>
<tr><td></td><td></td><td></td><td>taskbar (Figure 2-18).
2. Identify the PC's Bluetooth address <BdAddr_PC> in the General tab of Bluetooth Options (Figure 2-18).
3. Enable *AV Remote Control Service* (Figure 2-18).
4. Go to the Other tab of Bluetooth Options and click **AV Player** (Figure 2-19).
5. Select TopRight at the Display Position drop-down menu (Figure 2-20).
6. Close each window by clicking **OK**.</td></tr>
<tr><td>Initiate Connection</td><td>BTM5 xx</td><td>AT+AVD <BdAddr_PC></td><td>**Response:** (Figure 2-21)<table><tr><td>If AVRCP is connected</td><td>CONNECT <BdAddr_PC>,110E,></td></tr></table></td></tr>
<tr><td rowspan="3">Connected, send remote control commands</td><td rowspan="3">BTM5 xx</td><td>AT+AVC44</td><td>Send **Play** (Figure 2-22, Figure 2-23)
Response:<table><tr><td>If the command is accepted</td><td>OK</td></tr><tr><td>If a confirmation is received from the PC</td><td>AVPTC 0,44,0</td></tr><tr><td>If a confirmation is received from the PC</td><td>AVPTC 0,44,1</td></tr></table>Reception of command should be displayed in top-right corner of PC screen (Figure 2-23), Player should start playing.</td></tr>
<tr><td>AT+AVC46</td><td>Send **Pause** (Figure 2-22, Figure 2-23)
Response:<table><tr><td>If the command is accepted</td><td>OK</td></tr><tr><td>If a confirmation is received from the PC</td><td>AVPTC 0,46,0</td></tr><tr><td>If a confirmation is received from the PC</td><td>AVPTC 0,46,1</td></tr></table>Reception of command should be displayed in top-right corner of PC screen (Figure 2-23). Player should pause.</td></tr>
<tr><td>AT+AVC45</td><td>Send **Stop** (Figure 2-23)
Response:<table><tr><td>If the command is accepted</td><td>OK</td></tr><tr><td>If a confirmation is received from the PC</td><td>AVPTC 0,45,0</td></tr><tr><td>If a confirmation is received from the PC</td><td>AVPTC 0,45,1</td></tr></table>Reception of command should be displayed in top-right corner of PC screen (Figure 2-23), Player should stop.
Refer to Table 2-28 for more operations</td></tr>
<tr><td>Disconnect</td><td>BTM5 xx</td><td>AT+AVH</td><td>**Response:** (Figure 2-22)<table><tr><td>If the disconnection is confirmed</td><td>NO CARRIER 110E</td></tr></table></td></tr>
</table>



Device Information

Device Name: PC1

Device Address: 00:1E:37:FA:23:46

Device Type: Laptop

Tray Icon

●

○ Hide

Bluetooth Radio Power

Radio is enabled

Click Button to Disable

Enable Bluetooth Services

☑ Bluetooth Information Exchange Service

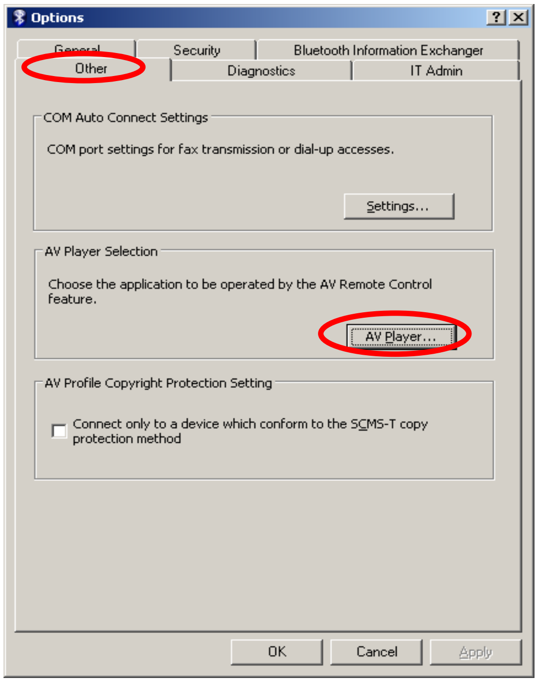
☐ PAN Networking Service

☐ Enable DHCP Service

☐ Bluetooth COM Port Service

☑ AV Remote Control Service

Figure 2-18: AVRCP Example 1 – Bluetooth Address of PC and AV Remote Control Service



COM Auto Connect Settings

COM port settings for fax transmission or dial-up accesses.

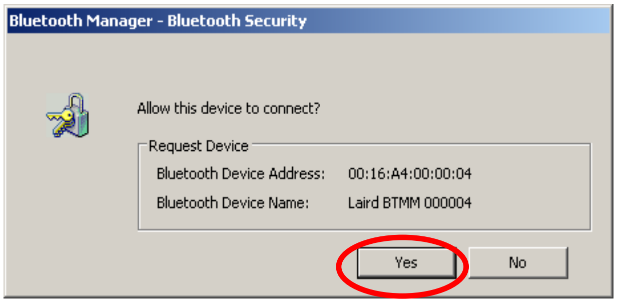
AV Player Selection

Choose the application to be operated by the AV Remote Control feature.

AV Profile Copyright Protection Setting

☐ Connect only to a device which conform to the SCMS-T copy protection method

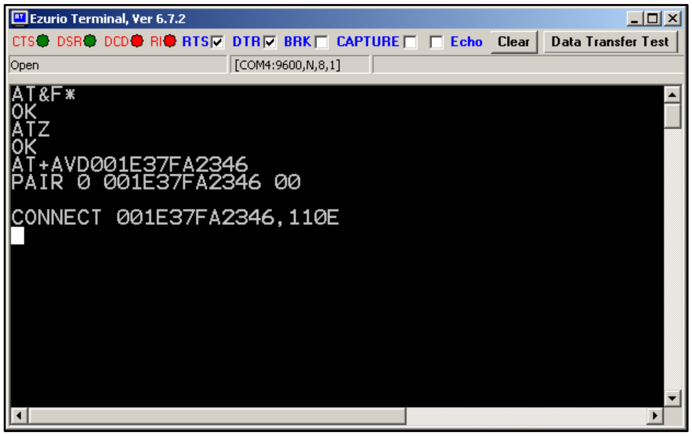
Figure 2-19: AVRCP Example 1 – Player Selection and Receiving Commands Display Setup



Allow this device to connect?

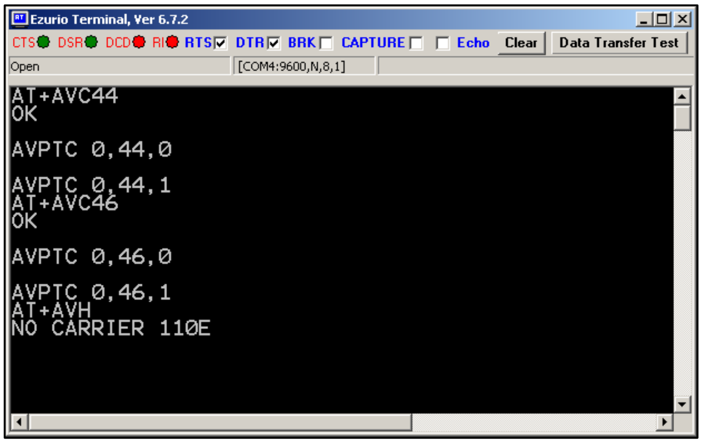
Request Device
Bluetooth Device Address: 00:16:A4:00:00:04
Bluetooth Device Name: Laird BTMM 000004

Figure 2-20: AVRCP Example 1 – Secure simple pairing dialogue


Ezurio Terminal, Ver 6.7.2
CTS DSR DCD RI RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo Clear Data Transfer Test
Open [COM4:9600,N,8,1]

```
AT&F*
OK
ATZ
OK
AT+AVD001E37FA2346
PAIR 0 001E37FA2346 00

CONNECT 001E37FA2346,110E
```

Figure 2-21: AVRCP Example 1 – BTM5xx Preparation and Connection Setup


Ezurio Terminal, Ver 6.7.2
CTS DSR DCD RI RTS ☑ DTR ☑ BRK □ CAPTURE □ □ Echo Clear Data Transfer Test
Open [COM4:9600,N,8,1]

```
AT+AVC44
OK

AVPTC 0,44,0

AVPTC 0,44,1
AT+AVC46
OK

AVPTC 0,46,0

AVPTC 0,46,1
AT+AVH
NO CARRIER 110E
```

Figure 2-22: AVRCP Example 1 – BTM5xx Sending Commands and Connection Release

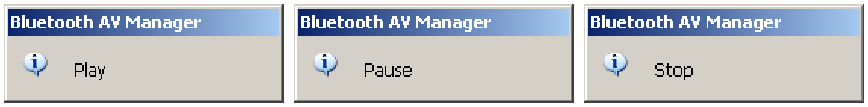


Figure 2-23: AVRCP Example 1 – Incoming AVRCP commands (top-right corner of screen)

2.6.4.2 AVRCP - Control (CT) and Target (TG)

This section describes AT Commands and S registers which are common to the BTM5xx AVRCP Controller and AVRCP Target roles.

2.6.4.2.1 Initiate AVRCP connection

AT+AVD<bd_addr$_{hex}$>

Initiate AVRCP control connection to Bluetooth address <bd_addr$_{hex}$>. The module must be configured as AVRCP Control by S register 301 = 1. Furthermore, a category must be selected in S register 302.

Response:

<cr,lf>CONNECT 123456789012,110E,><cr,lf>

Or: <cr,lf>NO CARRIER 110E<cr,lf>

Or: <cr,lf>ERROR 47<cr,lf>

Or: <cr,lf>ERROR 48<cr,lf>

After an AVRCP connection has been established, the module remains in AT command mode. S Register 531 is ignored for AVRCP connections.

2.6.4.2.2 Release AVRCP control connection

AT+AVH

Release AVRCP control connection.

Response :
<cr,lf>NO CARRIER 110E<cr,lf>

2.6.4.3 AVRCP - Control (CT)

This section describes AT Commands and S registers relevant to the BTM5xx as an AVRCP Controller (S301=1 or 3).

2.6.4.3.1 Send UNIT INFO Request

AT+AVU

Send a Unit Info request to a connected AVRCP target.

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

On command completion:

<cr,lf>AVUR <n>,<unit_id$_{hex}$>,<unit_type$_{hex}$>,<company_id$_{hex}$><cr,lf>

<n> = status$_{dec}$: 0 – success
1 – fail
4 – timeout

For unit_type see Table 2-29.

If <n> is greater than zero (unsuccessful outcome), <unit_type$_{hex}$> and <company_id$_{hex}$> is not sent. Examples of unsuccessful responses are:

<cr,lf>AVUR 1<cr,lf> or

<cr,lf>AVUR 4<cr,lf>

2.6.4.3.2 Send SUBUNIT INFO Request (incomplete*)

AT+AVS<page$_{dec}$>

Send a Subunit Info request to a connected AVRCP target.

Response: <cr,lf>OK<cr,lf> (immediately)

And: <cr,lf>AVSR <n>,<page$_{dec}$>,<pagedata$_{hex}$> <cr,lf> (after command completion)

<n> = status$_{dec}$: 0 – success
1 – fail
4 – timeout

<page$_{dec}$> : requested page [0..31]

<pagedata $_{hex}$>: 1st word of requested page

If <n> is greater than zero (unsuccessful outcome), <page$_{dec}$> and <pagedata $_{hex}$> is not sent. Examples for unsuccessful response are:

<cr,lf>AVSR 1<cr,lf> or

<cr,lf>AVSR 4<cr,lf>

*) incomplete because only the first word of the requested page is being displayed in the AVSR asynchronous message

2.6.4.3.3 Send remote control command

AT+AVC<operation_id$_{hex}$>,<button_state>

Send a remote control command to a connected AVRCP target. Internally, a PASS THROUGH command is created and sent to the PANEL subunit of the AVRCP target.

<operation_id$_{hex}$> is the value for the remote control command. Values are specified in Table 2-27.

<button_state> represents "Button pushed" (0) or "Button released" (1). If <button_state> is not specified, two PASS THROUGH commands, each with button_state=0 and button_state=1, are consecutively created and sent.

The OK response is sent immediately on receipt of AT+AVC command. On completion, an unsolicited message is sent to the host in the form AVPTC <n>,<bd_addr>, <button_state>. AVPTC means AVrcp Pass Through Confirmation. Parameter 'n' indicates the command's status:
'n'=0: successful, command confirmation received from target
'n'=1: timeout, target has not sent confirmation within the specified maximum time
'n'=2: all other unsuccessful outcomes

Parameters:
<operation_id$_{hex}$> (mandatory): see Table 2-27.
<button_state> (optional): 0 - Button pushed
1 - Button released
Response: <cr,lf>OK<cr,lf> (immediately)
And: <cr,lf>AVPTC <n>,<operation_id$_{hex}$>,<button_state><cr,lf>
(after command completion)
If status 'n' indicates an unsuccessful outcome,<operation_id$_{hex}$> and <button_state> are omitted.

2.6.4.4 AVRCP – Target (TG)

This section describes AT Commands and S registers for when BTM5xx is configured as an AVRCP Target (S301=2). In this mode, BTM5xx supports one subunit PANEL (see [3]).

2.6.4.4.1 Incoming AVRCP Connection Request

An incoming AVRCP connection request is accepted automatically if a valid link key for the paging device exists. If no link key is available, Secure Simple Pairing (SSP, BT2.1) or legacy pairing (BT2.0 or earlier) is carried out, depending on the Bluetooth Version of the paging device.

After an AVRCP connection has been established, the module remains in AT command mode. S Register 531 is ignored for AVRCP connections.

2.6.4.4.2 UNIT INFO Response

It is mandatory to respond to a UNIT INFO command if configured as AVRCP target. Required response parameters are IEEE Company ID and a Unit Type.

The IEEE Company ID is a 24 bit integer which can be set via S Register 303.

The response is sent automatically with the company ID as per S303 and a fixed unit type of 0x09 ("Panel")

2.6.4.4.3 SUBUNIT INFO Response

It is mandatory to respond to a SUBUNIT INFO command if configured as AVRCP target. Required response parameters are Subunit type and MaxSubUnitId.

The response is sent automatically with a fixed value of 0x09 ("Panel") for parameter Subunit type and a fixed value of 0x00 for parameter MaxSubUnitId (only one subunit exists, which is panel).

2.6.4.4.4 PASS THROUGH Indication

An incoming PASS THROUGH command is indicated by an unsolicited message:

AVPTI <subunit_id$_{hex}$>,<operation_id$_{hex}$>,<button_state>

subunit_id$_{hex}$: should always be zero as a BTM supports only one subunit.

For operation_id$_{hex}$ see Table 2-28.

<button_state>: '0' is Button pushed
'1' is Button released

2.6.4.4.5 PASS THROUGH Response

AT+AVR<avrc_response_type$_{hex}$>

If S register 310 == 0, a Pass Through (PT) response is required from the host. The response is sent using the command:

AT+AVR<avrc_response_type$_{dec}$>

Parameter:
<avrc_response_type$_{dec}$>:

Table 2-31, write-value.
If S register 310 == 1, a Pass Through response is sent automatically with an <avrc_response_type> defined by S register 311. In this case, the host is not required to respond.

Table 2-27: AVRCP – S Registers and AT Commands

Task	AT-Command/ S Register	Comment
Enable AVRCP profile	S102	256 = AVRCP, Error 47 if AVRCP role has not been set (see S301); Error 48 if S301== 2 and Category has not been set (see S302) Needs subsequent AT&W and ATZ to become effective.
Set AVRCP role	S301 [0..3]	0 Disabled 1 Control (CT) 2 Target (TG) 3 Controller and target, enabling AVRCP1.5 CT - default Needs subsequent AT&W and ATZ to become effective.
Set AVRCP category	S302 [0..4]	0 Feature disabled 1 Player/recorder – default 2 Monitor/amplifier 3 Tuner 4 Menu Needs subsequent AT&W and ATZ to become effective.
Initiate outgoing AVRCP control connection	AT+AVD<bd_addr>	**Response**: If accepted: CONNECT 0123456789012,110E,> If rejected: NO CARRIER 110E
Close only AVRCP connection	AT+AVH or ATH110E	**Response**: If a connection has existed and S329=0: NO CARRIER 110E If a connection has not existed and S329=0: NO CARRIER
Close all connections	ATH*	**Response:** For each profile that was previously connected: NO CARRIER <profileUUID> See Disconnecting Profiles.
Send remote control command (Control)	AT+AVC<operation_id$_{hex}$>,<state>	<operation_id$_{hex}$>: see Table 2-27 <state> (optional): 0 Button pushed 1 Button released **Response on command completion:** AVPTC <n>,<operation_id$_{hex}$>,<state>

Task	AT-Command/ S Register	Comment	
Send a Unit Info request (Control)	AT+AVU	**Response on command completion:** AVUR <n>,<unit_id$_{hex}$>,<unit_type$_{hex}$>,<company_id$_{hex}$> Successful if <n> = 0.	
Send a Subunit Info request (Control)	AT+AVS (incomplete*)	**Response on command completion:** AVSR <n>,<page$_{dec}$>,<pagedata$_{hex}$> <cr,lf> Successful if <n> = 0. * Only the first word of the pagedata is being displayed in the AVSR response message.	
Set Company Id (Target)	S303 [0..0xFFFFFF]	IEEE Company ID, 24bit hexadecimal, Required for UNIT INFO Response in AVRCP target mode, default value is 0xFFFFFF.	
Enable Unit Info Response (Target)	S305 [0..1]	0	Reject incoming Unit Info Requests
		1	Accept incoming Unit Info Requests and send response automatically (default) with Company ID as per S303 and unit type=0x09 (Panel, fixed)
Enable Subunit Info Response (Target)	S306 [0..1]	0	Reject incoming Subunit Info Requests
		1	Accept incoming Subunit Info Requests and send response automatically (default) with Subunit type = 0x09 (Panel, fixed) and MaxSubUnitId = 0x00 (fixed)
Configure PASS THROUGH (PT) Response (Target)	S310 [0..1]	0	Host is required to respond to PT-Indication, see AT+AVR.
		1	Enable automatic PT-response, response type is read from S311 (default).
Set automatic PT response type (Target)	S311 [0.. 7]	This value is queried for automatic PT-Response, see Table 2-31. Default value is "accepted" 1w/ 9r **Note:** If this value is set to **reject** (2w/10r), then incoming Pass Through commands are not forwarded to the host processor (no AVPTI message is sent to the host.)	
Respond to incoming Pass Through command (Target)	AT+AVR<avrc_response_type$_{hex}$>	<avrc_response_type$_{hex}$>: see Table 2-31 If S 310 == 1, response from host is not required.	
Suppress AVRCP direction indicator	S362 [0..1]	0	AVRCP direction indicator enabled (default).
		1	AVRCP direction indicator suppressed.

Table 2-28: AVRCP – Operation IDs for Remote Control Commands

Command	Operation ID	Command	Operation ID
Select	0x00	Previous channel	0x32

Command	Operation ID
Up	0x01
Down	0x02
Left	0x03
Right	0x04
Right up	0x05
Right down	0x06
Left up	0x07
Left down	0x08
Root menu	0x09
Setup menu	0x0A
Contents menu	0x0B
Favourite menu	0x0C
Exit	0x0D
0	0x20
1	0x21
2	0x22
3	0x23
4	0x24
5	0x25
6	0x26
7	0x27
8	0x28
9	0x29
Dot	0x2A
Enter	0x2B
Clear	0x2C
Channel up	0x30
Channel down	0x31
Sound select	0x33
Input select	0x34
Display information	0x35
Help	0x36
Page up	0x37
Page down	0x38
Power	0x40
Volume up	0x41
Volume down	0x42
Mute	0x43
Play	0x44
Stop	0x45
Pause	0x46
Record	0x47
Rewind	0x48
Fast forward	0x49
Eject	0x4A
Forward	0x4B
Backward	0x4C
Angle	0x50
Sub picture	0x51
F1	0x71
F2	0x72
F3	0x73
F4	0x74
F5	0x75
Vendor unique	0x7e

Table 2-29: AV/C Unit/Subunit Types

Unit/Subunit Type	Value
Monitor	0x00
Audio	0x01
Printer	0x02
Disc	0x03
Tape recorder player	0x04
Tuner	0x05
CA	0x06
Camera	0x07
Reserved	0x08
Panel	0x09
Bulletin board	0x0A
Camera storage	0x0B
Vendor unique	0x1C
Reserved for all	0x1D
Extended	0x1E
Unit	0x1F

Table 2-30: AVRC Respons Types

AVRC Response Type	Read-Value (S311, sent in response)	Write-Value (S311) or AT+AVR<parameter>
Not implemented	8	0
Accepted	9	1
Rejected	10	2
Note: Incoming pass through commands (AVPTI) are not displayed to the host if this is the value of S311		
In transition	11	3
Stable	12	4
Changed	13	5
Interim	15	6
Bad profile	16	7

Table 2-31: AVRCP Unsolicited Messages

Message	Comment		
CONNECT 123456789012,110E,>	Outgoing AVRCP connection established		
		123456789012	bd_addr of peer device
		110E	UUID indicating AVRCP
CONNECT 123456789012,110E,<	Incoming AVRCP connection established		
		123456789012	bd_addr of peer device
		110E	UUID indicating AVRCP
NO CARRIER 110E	AVRCP connection rejected or closed		
NO CARRIER	All Connections closed		
AVUR 0,<unit_id>,<unit_type$_{hex}$>,<company_id$_{hex}$>	Successful response to AT+AVU (UNIT INFO Request), indicates completion of command		
		<unit_id$_{hex}$>	Unit ID
		<unit_type>	See Table 2-29
		<company_id>	IEEE Company ID
AVUR <n> (n>0)	Unsuccessful response to AT+AVU (UNIT INFO Request), indicates completion of command.		
		<n> = status$_{dec}$	1 – Fail 4 – Timeout
AVSR 0,<page$_{dec}$>,<pagedata$_{hex}$>	Successful response to AT+AVS (SUBUNIT INFO Request), Indicates completion of command		
		<page$_{dec}$>	requested page [0..31]
		<pagedata$_{hex}$>	1st word of requested page
AVSR <n> (n>0)	Unsuccessful response to AT+AVS (SUBUNIT INFO Request), indicates completion of command		
		<n> = status$_{dec}$	1 – Fail 4 – Timeout
AVPTI <subunit_id$_{hex}$>,<operation_id$_{hex}$>,<state>	Indication of incoming Pass Through command		
		<subunit_id$_{hex}$>	Subunit id
		<operation_id$_{hex}$>	See Table 2-29
		<state>	0 – Button pushed 1 – Button pushed
AVPTC <n>,<operation_id$_{hex}$>,<state>	Confirmation of AT+AVC (Control Command Request)		
		<n> = status$_{dec}$	0 – Success 1 – Fail 4 - Timeout 7 – Operation not supported Any value in range [1…9] – Not successful
		<operation_id$_{hex}$>	See Table 2-27
		<state>	0 – Button pushed 1 – Button released

2.6.4.5 AVRCP GPIO Mapping

GPIOs can be mapped to AVRCP Commands (operations) with GPIO Configuration Registers S 651 to 658. If a GPIO is configured appropriately as input (See GPIO (General Purpose Input/Output)) with a valid av_operation_id assigned (see Table 2-27: AVRCP – S Registers and AT Commands) and if the module is configured as AVRCP Control (S 301), a rising edge causes the appropriate command request to be sent to the connected AVRCP target. This is as if AT+AVC is issued with <state>=0 (Button pushed). A trailing edge on this GPIO causes the same command to be sent but with <state>=1 (Button released).

The logical level of a GPIO can be inverted by setting the appropriate flag INV in the appropriate GPIO configuration register.

If configured as AVRCP Target, the direction flag (DIR) in the GPIO Configuration Register must be set to 1 (output) in order to indicate received commands at a digital output. A write operation to a GPIO has no effect if that GPIO is mapped to AVRCP.

The inversion Flag INV of the GPIO configuration register also applies to AVRCP targets.

GPIOs which are mapped to an alternative function, e.g. modem control line, cannot be used for this purpose.

Refer to Table 2-83 and section GPIO – AVRCP operation ID).

2.6.5 AVRCP 1.5 (Audio Video Remote Control Profile, CT role)

This section contains updated documentation about AVRCP1.5 on the BTM51x.

2.6.5.1 AVRCP1.5 AT Command Overview

- AT+AVQ – Command family: **query** AVRCP related data (parameters/items/attributes, etc.). Refer to Table 2-32 for more details on query commands.

Table 2-32: AVRCP1.5 Query commands

Query Command	Description
AT+AVQAP	Query list of available media players of connected target (TG)
AT+AVQCL	Query current play list
AT+AVQFS	Browse virtual file system
AT+AVQFSN	Browse virtual file system (next window)
AT+AVQMI	Query media item attributes
AT+AVQNP	Query now playing track attributes
AT+AVQRE	Query registered events (notification)
AT+AVQRF	Query remote device AVRCP features
AT+AVQXT	Query remote device AVRCP extensions (profile version, metadata)
AT+AVQEC	Query remote device AVRCP event capabilities
AT+AVQSP	Query addressed (‘selected’) player
AT+AVQBP	Query browsed player
AT+AVQPI	Query TG player ID
AT+AVQPA	Query TG player application setting attributes
AT+AVQPV	Query TG player application setting values (for all attributes)
AT+AVQPE	Query TG player application setting attribute value(s) text
AT+AVQPF	Query TG player features
AT+AVQPP	Query TG play status of addressed (selected) player
AT+AVQPT	Query TG player application settings attribute text
AT+AVQPR	Query TG player application settings attribute range

- AT+AVS – Command family: **set** AVRCP related data (player/browsing path/etc). Refer to Table 2-33 for more details on set commands.

Table 2-33: AVRCP1.5 Set commands

Set Command	Short Description
AT+AVSAI	Add item to now playing list
AT+AVSBP	set browsed player
AT+AVSG	set remote gain
AT+AVSSP	set addressed (selected) player
AT+AVSPU	set path upwards one level
AT+AVSPUQ	set path upwards one level and query
AT+AVSPRQ	set path upwards to root and query
AT+AVSPR	set path upwards to root
AT+AVSPI	play item (selected by browsing)
AT+AVSPQ	set path downwards and query
AT+AVSP	set path downwards
AT+AVSV	set player application value (e.g. repeat, shuffle etc.)

- Messages

The following table (Table 2-34) lists all AVRCP1.5 messages in alphabetical order.

Table 2-34: AVRCP1.5 messages in alphabetical order

Message / Tag	Type
AVAPC	Available players changed notification
AVAPI	Available player item
AVAPL	Available players list start / end
AVBS	Battery status of remote TG changed notification
AVCPL	Current play list start / end
AVFC	Folder (path) changed confirmation
AVFSI	File system item (folder)
AVFSL	File system list start / end
AVMEA	Media element attribute (browsing)
AVMEI	Media element item (track)
AVNPA	Now playing track attribute
AVNPC	Now playing content has changed notification
AVNPT	Now playing track attributes start / end
AVPA	Player application setting attribute only
AVPC	Player application setting changed list start / end (indication of value change)

Message / Tag	Type
AVPE	Player application setting attribute value text
AVPI	Play item confirmation
AVPL	Player application setting response list start / end (response to AT+AVQPV/A)
AVPP	Playback status/parameters response (AT+AVQPP)
AVPPC	Playback position changed notification
AVPS	Play status changed notification
AVPT	Player attribute text
AVPV	Player application setting attribute/value pair
AVSBP	Set browsed player confirmation
AVSPC	Addressed ('selected') player changed notification
AVSS	System status of remote TG changed notification
AVSSP	Set addressed ('selected') player confirmation
AVTC	Track changed notification
AVTE	Track reached end notification
AVTS	Track reached start notification
AVUC	UIDs changed notification
AVVC	Absolute volume changed indication

2.6.5.2 AVRCP1.5 Quick Start Example

For this quick start example, a TG device that supports AVRCP 1.4 or higher and A2DP SRC is required. Note that AVRCP 1.4 or higher allows the absolute volume feature to work. This example was tested against an iPhone as the TG but any recent smartphone that supports these Bluetooth profiles and versions should work.

AT Command/Message	Phone	Comment
Preparation		
AT&F*		Factory default settings
ATS515=$200400		Set device class
ATS512=4		Make connectable and discoverable after reset
AT&W		Save settings
ATZ		Reset

AT Command/Message	Phone	Comment	
	Search for BT device **Laird BTMM…** and connect	Pairing should occur followed by A2DP and AVRCP connection. If pairing or connection fails, ensure that any previous pairing with BTM51x is deleted (i.e. 'ignore device') and try again.	
Operation on Phone			
	Open the music player application		
Message: AVPS 01"Playing" AVNPT $0007 AVNPA0 01"Title",006A,000F,"<title_name>” … AVNPT OK	Chose a track and start playing	AVPS	**P**layback **S**tatus changed. New status = 1, hint string "Playing" (which can be disabled by S390 bit0 = 0
		AVNPT	**N**ow **P**laying track attributes presented as list with start/end tag AVNPT
		AVNPA 0	Track attribute items (packet type '0') Hint strings for attribute type
Message: … AVVC $4F,$09 …	Change volume on phone	AVVC – **V**olume **C**hanged message which indicates that absolute volume is received and corresponding output gain which is applied immediately. Can be double- checked using AT+GO?.	
Message: AVTC <UID> AVNPT $0007 AVNPA0 01"Title",006A,000F,"<title_name>” … AVNPT OK	Jump to next track	AVTC	**T**rack **C**hanged indication with UID of new track
		AVNPT	Start/end tag for **N**ow **P**laying Track attributes
		AVNPA 0	Attribute items Hint strings for attribute type (packet type=0)
Message: AVPS 02"Paused"	Stop playback	AVPS – **P**layback **S**tatus changed, new status=2, hint string "Paused" (can be disabled by S390 bit0=0)	

AT Command/Message	Phone	Comment	
Message: AVPC $0002 AVPV 02"Repeat",02"Single track" AVPV 03"Shuffle",01"Off" AVPC OK	Change repeat mode, e.g. ‘repeat track’	AVPC <n>	Player application settings Changed, followed by <n> att/val items.
		AVPV	Attribute/value pairs of player application settings.
		AVPC OK	End tag.
Operation on BTM51x			
Command: AT+AVQFS *Response:* AVFSL $0007,$00… AVFSI[0] <UID>, 05"Playlists",0,… AVFSI[1] <UID>, 03"Artists",0,… AVFSI[2] <UID>, 01"Titles",0,… … AVFSL OK OK		Query folder items. Initial path should be root.	
		AVFSL	Start tag of file system list; seven items in this list: <start offset>, <end offset>, <total items in folder>
		AVFSI[<n>]	File system item; n= UidCacheIndex Hint strings after folder type (Playlists/Artists).
		AVFSL OK	List’s end tag.
		OK	Command terminator
Command: AT+AVSPQ2 *Response:* AVFSL $0008, … AVMEI[0] <UID$_0$>, 00”Audio”,01,… AVMEI[1] <UID$_1$>, 00”Audio”,01,… … AVFSL OK OK		Change path to item indicated by UidCachIndex [2] in previous list. Should be list of “Titles”, UidCacheIndex of “Titles” depends on TG and may differ on TG devices; In addition query the new path like AT+AVQFS (combined command)	
		AVFSL	File system start tag; eight items in this list: <start offset>, <end offset>, <total items in folder>
		AVMEI[<n>]	Media element item; n=UidCacheIndex
		AVFSL OK	List’s end tag.
		OK	Command terminator

AT Command/Message	Phone	Comment	
Command: AT+AVSPI<UID$_1$> or AT+AVSPI1 Response – AVPI $0000 OK ***Subsequent messages:*** (event notifications may vary depending on TG status) **AVNPC** AVNPT $0007 … AVNPT OK **AVTC <UID>** AVNPT $0007 … AVNPT OK **AVPS 01"Playing"** AVNPT $0007 … AVNPT OK	Observe: Now playing track change Playback status change	AT+AVSPI	Play item identified by <UID$_1$> of previous list or by UidCacheIndex [1] of previous list
		AVPI $0000	Play item success confirmation
		OK	Command terminator
		AVNPC	Now playing content changed, triggers request of now playing track attributes (if S390 Bit6=0).
		AVNPT	List of now playing track attributes triggered by AVNPC
		AVTC <UID>	Track changed notification, UID of new track, triggers request of now playing track attributes (if S390 Bit5=0).
		AVNPT	List of now playing track attributes triggered by AVTC
		AVPS 01"Playing "	Playback status changed notification, new status=01 (playing), Hint string for playing status; This event triggers a now playing attributes request (if S390 Bit4=0).
		AVNPT	list of now playing track attributes triggered by AVPS
Command: AT+GOU or AT+GOD or AT+GO<n> *Response:* OK	Observe: indication of changed volume (e.g. slider position)	Change BTM51x local output gain. <n> – absolute output gain, range=[0..15] OK – command terminator	

AT Command/Message	Phone	Comment	
Command: AT+AVSVR3 *Response:* AVSV OK OK *Subsequent messages:* AVPC $0002 AVPV 02"Repeat",03"All tracks" AVPV 03"Shuffle",01"Off" AVPC OK	Make sure that player application is displayed on screen and observe: Change of repeat mode to “Repeat All” (or similar)	AT+AVSVR 3	Change repeat mode to 3 “All tracks” (set player application setting)
		AVSV OK	Player application setting change success confirmation
		OK	Command terminator
		AVPC	List of current player application settings (repeat, shuffle etc.), triggered by the change request
		$0002	Two items to follow
		AVPV	Attribute/value pair of player application setting with hint strings
		AVPC OK	List end

2.6.5.3 AVRCP1.5 Application Notes

In contrast to previously supported AVRCP1.0, AVRCP1.5 introduces a number of advanced features for control of a remote audio player, such as:

- Browsing the virtual file system on the TG (media player), change path, select a media element item (i.e. a track) for play back
- Query information from the TG data base (media player) such as track details (attribute list), list folder content
- Control of player application settings (shuffle, repeat etc.)
- Registration for reception of event notifications
- Absolute volume control

If information of this document appears insufficient, we recommend that you refer to the AVRCP1.5 specification document for clarification and more background information.

This initial implementation has been developed using a phone as TG for testing. Some AVRCP1.5 features are not currently supported because they were either not supported by the test phone or could not be tested. These include the following:

- Search
- Add to now playing list
- Group navigation
- Battery status changed

Note: Only the CT role of AVRCP1.5 is supported by this firmware. Details of the feature map are listed in **Table 2-25**. TG role should still be functional for AVRCP1.0 as in earlier versions.

2.6.5.4 AVRCP1.5 Initialisation, Role Configuration

At boot time, AVRCP functionality is initialised according to S-Registers S301 (AVRCP role). In order to enable functionality of AVRCP1.5, S301 must be set to 3 (new default value) which corresponds to *target and controller* role. The only target functionality required here is to receive volume change commands from the remote device (phone). When volume is changed at the phone, then the relevant message causes the BTM51x's audio output gain to be set accordingly, providing an A2DP link exists with the same device and the BTM51x is audio sink. This is the only scenario where the phone acts as CT and BTM51x is acting as TG. Given that in all other scenarios BTM51x is acting as CT and a phone is acting as TG, the volume case is neglected when stating that only AVRCP1.5 CT role is supported.

In addition, for the volume case to be functional, BTM51x TG features are set to category 2 internally which is hardcoded.

We observed during development that S302 must be set to 1 (category 1). If S301=3 (target and controller), then S302 value stands for CT functionality. If S302 has a different value while S301=3, then features may not work as described or not work at all. Category 1 is the focused-upon use case for this release.

The new default values for S301 (=3) and S302 (=1) reflect these findings.

2.6.5.5 AVRCP connection setup

When the BTM51x connects to a remote TG device by AVRCP1.5, information is exchanged in the background between TG and CT. For example: AVRCP features, profile extensions (beyond AVRCP1.0), and event capabilities. These TG characteristics are buffered in BTM51x and can be queried using AT+AVQRF, AT+AVQXT, AT+AVQEC. These are mainly used for BTM51x internal purposes such as rejecting unsupported requests.

Subsequently to this exchange, BTM51x attempts to register as many event capabilities as possible in order to receive relevant notifications from TG. The outcome of this can be checked using AT+AVQRE, which returns flags set for all events that have been successfully registered.

Typically, the TG (phone) registers for notifications on local BTM51x volume change events. In this case, the volume change notification is sent each time the local output gain is changed by AT+GOU/AT+GOD/etc. The phone uses this value for awareness of current output gain setting and also to update its display.

2.6.5.6 AVRCP operation

Once AVRCP has successfully connected and registered for events, typical notification messages are forwarded to UART such as play status changed (start/pause), now playing track changed, player settings (repeat, shuffle etc.) changed, available players, and so forth.

For a number of notifications, BTM51x subsequently and automatically queries more data related to the new status – such as now playing track attributes (Title, Artist, Album etc.) – and prints it to UART. We have observed that, in some circumstances, multiple events occurring virtually at the same time may cause multiple lists of now playing track attributes, for example. If this is interfering with a host controller, S390 also allows the prevention of automatic queries for a number of notifications (see S390 bits 4/5/7). Queries for now playing track attributes can also be carried out by AT command at any time (AT+AVQNP).

2.6.5.7 UID cache

In the virtual file system, folders and media element items (track) are uniquely identified by a 64 bit UID (unique identifier). This UID is always contained in relevant UART messages (list items). Given that handling 64bit values may be inconvenient for resource constrained host microcontrollers or for human beings during evaluation, BTM51x can cache up to 10 UIDs of an item list in an internal array. For a subsequent command, a listed item can then be referred by its UidCacheIndex, which is between 0 and 9, rather than the UID which is 16 characters long.

UID cache feature is enabled by default and can be disabled by setting S391=0. The quick start example above also covers this feature.

2.6.5.8 Browsing

The browsing feature allows a CT device to navigate, view, and launch media content on the TG. A browsing channel is created automatically on top of an existing AVRCP link when needed. After five seconds of no browsing activity, the browsing channel is closed by BTM51x. This browsing channel handling happens in the background, invisible to the user or host controller; no messages concerning browsing channels are sent to UART.

Media content on the target is structured in a tree oriented virtual file system. BTM51x provides AT commands to change the current path and to list items in the current path/folder. When listing folder items (AT+AVQFS), the number of items being returned may be limited by AVRCP packet size. This means that only a partial window of the entire folder list is returned by one command. Because of this, a start offset parameter <s> can be appended to AT+AVQFS<s>; it must be less than or equal to the total number of items in the current folder. If omitted then <s> defaults to zero meaning that the list is returned from top (first item).

The total number of items is indicated in the change path/folder confirmation (AVFC $<**nItems**>) and also in the start tag of a file system list (AVFSL $<nItems>,$<nStartOffs>,$<nEndOffs>,$<**nTotalItems**>).

AT+AVQFSN (N=next) permits it to list the next window of folder items by calculating the following start offset dynamically from the last printed list (start_offset + nItems actually returned). Hence, multiple calls of AVQFSN print all chunks of an item list without having to manage start offsets for each window by the host controller.

Table 2-35 provides an overview of AT commands for navigating the virtual file system. More detail about each command can be found in Table 2-39 and Table 2-40.

Table 2-35: Standard navigation AT commands

Standard Operations for File System Navigation	
AT Command	Short Description
AT+AVQFS[<s>[,<e>]]	List folder items (current path)
AT+AVSP<n>	set path downwards
AT+AVSPU	set path upwards one level
AT+AVSPI<n>	play item (selected by browsing)
AT+AVQMI<n>	query media item attributes
Arguments:	
<n>	UidCacheIndex or UID
<s>	start_offset, optional, set to 0 if not passed
<e>	end_offset, optional, set to <s>+9 if not passed

Beyond standard operations like list folder items, set path up/down, etc. (defined in AVRCP1.5 specification) the BTM51x provides a set of combined commands which facilitate more convenient navigation. See Table 2-36.

Table 2-36: BTM51x combined navigation AT commands

BTM51x Combined Operations

BTM51x Combined Operations		
Navigation Command	Short Description	Comment
AT+AVQFSN	List folder items next window	AT+AVQFS<s> with start offset <s> managed by BTM51x
AT+AVSPR	Set path upwards to root	Multiple calls of AT+AVSPU until root folder is reached
AT+AVSPQ<n>	Set path downwards and query	= AT+AVSP<n> ; AT+AVQFS
AT+AVSPUQ	Set path upwards one level and query	= AT+AVSPU ; AT+AVQFS
AT+AVSPRQ	Set path upwards to root and query	= AT+AVSPR ; AT+AVQFS
Arguments: <n> UidCacheIndex or UID		

2.6.5.9 AVRCP sniff mode delay

We have observed that responses to AVRCP requests (AT commands) are delayed in the range of about 0.5 to 1 second. This is likely to be caused by sniff mode.

2.6.5.10 AVRCP1.5 continuation

This feature caters for the case that media element attributes may not fit into a single packet due to large size (such as a long track name or similar). This is why the AVRCP1.5 specification defines four different packet types:

- Single (0)
- Start (1)
- Continue (2)
- End (3)

If all attribute data fits into a single packet, then the packet type is Single (0). If attribute data is spread over more than one packet, then the first packet type is Start (1), followed by zero or more packets of type Continue (2), followed by a terminating packet of type End (3).

On the BTM51x, this becomes relevant for representation of now playing track attributes – **AVNPA…** message. To cater for this feature, the current packet type is appended immediately to AVNPA<n> as a single decimal character [0…3]. Most cases are 0; however, if the packet type is 1, then more AVNPA<pt>… messages must be expected as described above (pt=2, finally pt=3). For each packet received, a separate AVNPA<pt>… message is sent to UART.

This feature has only been tested against PTS (profile tuning suite) but not against a test TG due to lack of support on the latter (not having a long enough track name) [Ref.4-46].

2.6.6 AVRCP1.5 AT Command Reference

2.6.6.1 S390 – AVRCP1.5 configuration

The flags of S390 are designed so that cleared flags correspond to machine-friendly, automated, and complete configuration, whereas set flags (except bits 4,5,7) correspond to a more human-friendly configuration.

- S390 Range: [0..$FF]
- S390 Default: $7F

S390 details are listed in Table 2-37.

Table 2-37: S390 - AVRCP1.5 configuration

Bit	Mask	Description	Value	
0	0x01	Hint strings	0	Do not append hint strings
			1	Append hint strings to: ▪ Play status (AVPS) ▪ Player application settings (AVPA / AVPV ▪ Folder (AVFSI) ▪ Media type (AVMEI) ▪ Media element attribute (AVMEA) ▪ Now playing track attribute (AVNPA)
1	0x02	Media element attributes (browsing)	Ignored	Available attributes are not appended to media items (AVMEI).
			Reason	On the test device, extra attributes did not provide additional information (just the track name which is already provided). More importantly, the extra attribute data was observed to overrun the UART Tx buffer due to an inflated data format for presentation (start/stop tags, etc.)
2	0x04	ASCII presentation of strings	0	Print displayable names, attribute values, and other strings in hexadecimal format; more machine-friendly than human-friendly.
			1	Print ASCII strings surrounded by double quotes. Non-printable characters are presented hexadecimal as \xx. A character is considered non-printable if it is ≥ 128, double quote (“), backslash (\), space ( ), or comma (,). More human-friendly than machine-friendly but also has more exceptions (such as character set).
3	0x08	Attributes query basic / full	On automatic requests for now playing track attributes query:	
			0	Full set of attributes (title, artist, album, track no., total tracks, genre, playing time)
			1	Basic set of attributes only (title, artist, album)
4	0x10	Action on 'Play status changed' notif. (AVPS…)	On reception of AVPS notification (play status changed) and if new status is playing (0x01):	
			0	Automatically query now-playing track attributes
			1	Do nothing

Bit	Mask	Description	Value
5	0x20	Action on 'Track changed' notif. (AVTC)	On reception of AVTC notification (track changed): 0 Automatically query now-playing track attributes 1 Do nothing
6	0x40	Action on *Now-playing content changed* notification (AVNPC)	Ignored No automatic query of now-playing track is initiated on the reception of AVNPC. This was deemed not useful and confusing due to multiple redundant AVNPC notifications (up to four were observed).
7	0x80	Action on 'Available players changed' notif. (AVAPC)	On reception of AVAPC notification (available players changed): 0 Automatically query available players list (AT+AVQAP) 1 Do nothing **Note:** We observed AVAPC when switching forth and back between the standard player Music and other streaming applications such as a web-radio player or similar.

2.6.6.2 S397 – AVRCP1.5 notification control

S-register 397 allows you to disable certain AVRCP1.5 notifications. Disabling certain notifications can help to simplify requirements for the host controller's parser algorithm. Messages that are currently not covered by S397 can only be suppressed by S504=1, which unfortunately generally prevents any output on the BTM51x UART (silent mode).

- S397 Range: [0..$FF]
- S397 Default: $0C

S397 details are listed in Table 2-38.

Table 2-38: S397 - AVRCP1.5 notification control

Bit	Mask	Description	Value
0	0x01	AVPS… Play status changed notification	On reception of play status changed event: 0 Print AVPS… message to UART 1 Suppress AVPS… message
1	0x02	AVTC… Track changed notification	On reception of track changed event: 0 Print AVTC… message to UART 1 Suppress AVTC… message

Bit	Mask	Description	Value
2	0x04	Redundant AVTC Track changed notification	On reception of redundant track changed event: 0 Print AVTC… message to UART 1 Suppress AVTC… message **Note:** An AVTC message is regarded as redundant if the UID parameters of precedent and current AVTC events are identical. We observed redundant AVTC messages during regression test phase.
3	0x08	AVNPC Now playing content changed notification	On reception of now playing content changed event: 0 Print AVNPC… message to UART 1 Suppress AVNPC… message (default) We observed multiple redundant occurrences of this message during regression test phase. Querying now playing track (automatically) based on this event did not lead to meaningful results (showing the old track). In nearly all cases, AVNPC notification was accompanied by AVTC event. Querying now playing track based on AVTC event led to meaningful results (showing the new track). Based on these observations we recommend that you ignore AVNPC. However, there may be other AVRCP1.5 TG devices which may show a different behavior.
4	0x10	AVVC Absolute volume changed on TG notification.	On reception of absolute volume changed event: 0 Print AVVC… message to UART 1 Suppress AVVC… message (default) Analogue output gain of the BTM51x is always adjusted on this event, regardless of whether or not this message is printed. Current output gain can be queried anytime using AT+GO?.
5	0x20	AVAPC / AVSPC Available players changed / selected [addressed] player changed notification.	On reception of available players changed event or selected/addressed player changed event: 0 Print relevant message AVAPC or AVSPC… to UART 1 Suppress relevant message AVAPC or AVSPC… to UART

Bit	Mask	Description	Value
6	0x40	AVPC Player application setting changed notification	On reception of player application setting changed event: 0 – Print list of updated attribute/value pairs AVPC… to UART 1 – Suppress AVPC… list message
7	0x80	Other AVRCP1.5 notifications	On reception of any of the following events: AVPPC – Playback position changed event AVBS – TG battery status changed event AVSS – TG system status changed (such as power on/off) AVT[E|S] – Track reached end|start event AVUC – UIDs changed event 0 – Print relevant message to UART 1 – Suppress relevant message

2.6.6.3 Query commands

Table 2-39: AVRCP1.5 query commands

AT-Command	Purpose	Description, Parameters, Etc.	
AT+AVQAP	Query list of available media players of connected target (TG)	**Parameters**	None
		Response	Available players list. See AVPL, AVAP (Table 2-42)
		Terminator	OK/ERROR nnn
AT+AVQCL	Query current play list	**Parameters**	None
		Response	Current play list. See AVCPL, AVMEI (Table 2-42)
		Terminator	OK/ERROR nnn
AT+AVQFS	Browse virtual file system, list items of current path	**Syntax**	AT+AVQFS[<s>[,<e>]]
		Parameters (optional)	<s>: start offset, UI16 decimal/hex with $ <e>: end offset, UI16 decimal/hex with $ If no parameters are passed then <s> defaults to 0 and <e> defaults to <9>. If only <s> is passed then <e> = <s>+9. It can happen that actually less items than requested are returned. This can be caused by a limited buffer size which is negotiated at beginning of an AVRCP connection. AVRCP1.5 spec, p.76 (last paragraph) describes these circumstances.
		Response	List of folder items. See AVFSL/AVFSI/AVMEI (Table 2-42)
		Terminator	OK/ERROR nnn

AT-Command	Purpose	Description, Parameters, Etc.	
AT+AVQFSN	Browse virtual file system, next block (or window)	**Parameters**	None **Note:** This command is like AT+AVQFS<s> whereby start offset <s> is calculated from a previous list so that the subsequent items are requested. With this command, a host controller is not needed to parse all parameters from AVFSL start message. If no previous list exists, then <s> defaults to 0. The end offset <e> is always calculated as <s>+9. This command allows to cycle through an entire list of items iteratively. A reset to <s>=0 can be accomplished by AT+AVQFS.
		Response	List of folder items. See AVFSL/AVFSI/AVMEI (Table 2-41) (Table 2-42)
		Terminator	OK/ERROR nnn
AT+AVQMI	Query media item (track) attributes	**Parameters**	UID (64 bit hex) or UidCacheIndex (UI8, 0..9), UID or UidCacheIndex is retrieved from a previous folder list (AT+AVQFS…)
		Response	Attribute list of item, see AVME, AVMEA (Table 2-42)
		Terminator	OK/ERROR nnn
		Examples: AT+AVQMI75E40538A0744A6A (UID) AT+AVQMI2 (UidCacheIndex)	
AT+AVQNP	Query now playing track attributes	**Parameters**	None
		Response	Now playing track attributes list, see AVNPT, AVNPA (Table 2-42)
		Terminator	OK/ERROR nnn
		Note: This command queries all available attributes (usually seven).	
AT+AVQRE	Query registered events (for notification)	**Parameters**	None
		Response	UI16 hex value indicating all events which have been registered with the connected target in order to receive event notifications from TG. See Table 2-48.
		Terminator	OK/ERROR nnn

AT-Command	Purpose	Description, Parameters, Etc.	
AT+AVQRF	Query remote device AVRCP features	**Parameters**	None
		Response	UI16 hex value indicating features of remote device. Value corresponds to SDP record of remote device. See Table 2-49.
		Terminator	OK/ERROR nnn
AT+AVQXT	Query remote device AVRCP extensions (profile version, metadata)	**Parameters**	None
		Response	UI16 hex value indicating profile version and metadata support of local device. Currently $0007 is returned which indicates the following: AVRCP1.4 + browsing supported
		Terminator	OK/ERROR nnn
AT+AVQEC	Query remote device event capabilities	**Parameters**	None
		Response	UI16 hex value indicating all events the connected remote device (TG) offers the local device (CT) to register for, to receive event notifications. See Table 2-48.
		Terminator	OK/ERROR nnn
AT+AVQSP	Query addressed ('selected') player	**Parameters**	None
		Response	UI16 hex value, player ID of addressed (selected) player.
		Terminator	OK/ERROR nnn
AT+AVQBP	Query browsed player ID	**Parameters**	None
		Response	UI16 hex value, player ID of browsed player. If $FFFF is returned, then no command has been issued to set the browsed player (AT+AVSPB<n>) since power cycle.
		Terminator	OK/ERROR nnn
AT+AVQPI	Query addressed player ID	**Parameters**	None
		Response	UI16 hex value, player ID of browsed player.
		Terminator	OK/ERROR nnn
AT+AVQPA	Query TG player application setting attributes	**Parameters**	None
		Response	List of player application attributes (but not values). See AVPL, AVPA (Table 2-41 and Table 2-42)
		Terminator	OK/ERROR nnn
AT+AVQPV	Query TG player application setting values (for all attributes)	**Parameters**	None
		Response	List of player application attribute/value pairs. See AVPL, AVPA (Table 2-41 and Table 2-42)
		Terminator	OK/ERROR nnn

AT-Command	Purpose	Description, Parameters, Etc.	
AT+AVQPE…	Query TG player application setting attribute **value text**	Syntax	AT+AVQPE[<AttId>[,<nValId>]]
		Parameters	**<AttId>** - Attribute ID to query text for (optional). **<nValId> -** Value ID to query text for (optional). If neither <AttId> nor <nValId> are specified, then the value texts for all possible value IDs of all available attributes are returned. If only <AttId> is specified, then value texts for all possible value IDs of <AttId> are returned.
		Response	List of player application attribute/value pairs. See AVPL, AVPE (Table 2-41 and Table 2-42)
		Terminator	OK/ERROR nnn
AT+AVQPF	Query TG player features	Parameters	None
		Response	Octets 7 and 8 of Player Feature Bitmask of the last player listed in the available players list (AT+AVQAP, AVAPI). See AVRCP1.5 Specification, pages 68-69.
		Terminator	OK/ERROR nnn
AT+AVQPP	Query play status and parameters of addressed (selected) player	Parameters	None
		Response	AVPP <play_status>,<elapsed_ms>,<song_len_ms>
		Terminator	OK/ERROR nnn
AT+AVQPT[<AttId>]	Query TG player application setting **attribute text**	Syntax	AT+AVQPT[<AttId>]
		Parameters	**<AttId> -** Attribute ID to query text for (optional). If <AttId> is not specified, then the text for all available attributes is returned.
		Response	List of player application attribute/value pairs. See AVPL, AVPT (Table 2-41 and Table 2-42)
		Terminator	OK/ERROR nnn
		Note: This was not tested due to lack of support on test TG.	

AT-Command	Purpose	Description, Parameters, Etc.	
AT+AVQPR[<AttId >]	Query TG player application setting **value range**	**Syntax**	AT+AVQPR[<AttId>]
		Parameters	**<AttId> -** Attribute ID to query possible value IDs for (optional). If <AttId> is not specified, then all possible value IDs for all available attributes is returned.
		Response	List of player application attribute/value pairs. See AVPL, AVPR (Table 2-41 and Table 2-42)
		Terminator	OK/ERROR nnn

2.6.6.4 Set commands

Table 2-40: AVRCP1.5 set commands

AT-Command	Purpose	Description, Parameters, Etc.	
AT+AVSAI<n>	Add media item to now playing list	**Parameters**	<n>=UidCacheIndex (0…9) or UID (16 hex characters, \$ prefix not required) The parameter is expected to be chosen from an earlier file system (such as AT+AVQFS). In particular, the UID cache index always refers to the last printed list.
		Note: This was tested against PTS only and not against test TG due to lack of support on test TG.	
AT+AVSBP<n>	Set browsed player	**Parameters**	<n>=player ID, decimal/hex with \$ prefix
		Response	**Success –** AVSBP\$< plid$_{16h}$>,\$<nItems$_{16h}$> **Failure –** AVSBP ERROR \$<asc$_{16h}$> **With:** ▪ <plid>: browsed player ID (confirmation) ▪ <nItems>: number of items in root folder ▪ <asc>: avrcp status code, see Table 2-50. **Examples:** AVSBP \$0001,\$0007 AVSBP ERROR \$0091
		Terminator	OK/ERROR 135
AT+AVSG<n>	Set absolute volume on TG	**Parameters**	<n>=volume, decimal/hex with \$ prefix. Range of <n>: 0..127 The local volume is adjusted appropriately (similar to AT+GO<m> with <m> in [0…15]).
		Note: This was not tested due to lack of support on test TG)	
AT+AVSSP<n>	Set addressed ('selected') player	**Parameters**	<n>=player ID, decimal/hex with \$ prefix
		Response	**Success –** AVSBP\$< plid$_{16h}$>,\$<nItems$_{16h}$>

AT-Command	Purpose	Description, Parameters, Etc.	
			Failure – AVSBP ERROR \$<asc$_{16h}$> **With:** ▪ <plid>: browsed player ID (confirmation) ▪ <nItems>: number of items in root folder ▪ <asc>: avrcp status code, see Table 2-50. **Examples:** AVSBP \$0001,\$0007 AVSBP ERROR \$0091
		Terminator	OK/ERROR 135
AT+AVSPUQ	Set path upwards one level and query	Parameters	None
		Response	**Success** – AVFSL.. / AVFSI… (Table 2-41 and Table 2-42) **Failure** – AVFC ERROR \$<asc$_{16h}$> **With:** ▪ <asc>: avrcp status code, see Table 2-50.
		Terminator	OK/ERROR 135
AT+AVSPU	Set path upwards one level	Parameters	None
		Response	**Success** – AVFSL.. / AVFSI… (Table 2-41 and Table 2-42) **Failure** – AVFC ERROR \$<asc$_{16h}$> **With:** ▪ <asc>: avrcp status code, see Table 2-50.
		Terminator	OK/ERROR 135
AT+AVSPRQ	Set path upwards to root and query	Parameters	None
		Response	**Success** – AVFSL.. / AVFSI… (Table 2-41 and Table 2-42) **Failure** – AVFC ERROR \$<asc$_{16h}$> **With:** ▪ <asc>: avrcp status code, see Table 2-50.
		Terminator	OK/ERROR 135
AT+AVSPR	Set path upwards to root	Parameters	None
		Response	**Success** – AVFC <nItems$_{16h}$>. See Table 2-42 **Failure** – AVFC ERROR \$<asc$_{16h}$> **With:** ▪ <nItems>: number of items in root folder ▪ <asc>: avrcp status code, see Table 2-50.
		Terminator	OK/ERROR 135
AT+AVSPI<n>	Play item (selected by browsing)	Parameters	<n>=UidCacheIndex (0…9) or UID (16 hex characters, \$ prefix not required). The parameter is expected to be chosen from an earlier file system query (such as AT+AVQFS). In particular, UID cache index always refers to the last printed list.
		Response	**Success** – AVPI \$0000

AT-Command	Purpose	Description, Parameters, Etc.	
			Failure – AVPI ERROR \$<asc$_{16h}$> **With:** ▪ <asc>: avrcp status code, see Table 2-50. … plus subsequent automatic queries due to track changed event, play status changed, etc., depending on S390.
		Terminator	OK/ERROR 135
AT+AVSPQ<n>	Set path downwards and query	**Parameters**	<n>=UidCacheIndex (0…9) or UID (16 hex characters, \$ prefix not required). The parameter is expected to be chosen from an earlier file system (such as AT+AVQFS). In particular, UID cache index always refers to the last printed list.
		Response	**Success** – AVFSL… / AVFS… / AVMEI… (Table 2-41 and Table 2-42) **Failure** – AVFC ERROR \$<asc$_{16h}$> **With:** ▪ <asc>: avrcp status code, see Table 2-50. … plus subsequent automatic queries due to track changed event, play status changed, etc., depending on S390.
		Terminator	OK/ERROR 135
AT+AVSP<n>	set path downward	**Parameters**	<n>=UidCacheIndex (0…9) or UID (16 hex characters, \$ prefix not required). The parameter is expected to be chosen from an earlier file system (such as AT+AVQFS). In particular, UID cache index always refers to the last printed list.
		Response	**Success** – AVFC <nItems$_{16h}$> See also **Error! Reference source ot found.** **Failure** – AVFC ERROR \$<asc$_{16h}$> **With:** ▪ < nItems>: number of items in new path ▪ <asc>: avrcp status code, see Table 2-50.
		Terminator	OK/ERROR 135
AT+AVSV	set player application value (e.g. repeat, shuffle etc.)	**Syntax**	AT+AVSV<nAtt$_{dec}$>,<nVal$_{dec}$> AT+AVSVE<nVa$_{dec}$l> [Equaliser,nAtt=1] AT+AVSVR<nVal$_{dec}$> [Repeat,nAtt=2] AT+AVSVS<nVal$_{dec}$> [Shuffle,nAtt=3] AT+AVSVC<nVal$_{dec}$> [Scan,nAtt=4]
		Parameters	<nAtt$_{dec}$>,<nVal$_{dec}$> With: ▪ <nAtt>: attribute ID, see Table 2-44

AT-Command	Purpose	Description, Parameters, Etc.	
			▪ <nVal>: value, see Table 2-44 Option: <nAtt> can be replaced by E/R/S/C as outlined above.
		Response	**Success** – AVSV OK **Failure** – AVSV ERROR \$<asc$_{16h}$> **With:** ▪ <asc>: avrcp status code, see Table 2-50.
		Terminator	OK/ERROR nnn

2.6.6.5 Item lists

A number of different lists has been implemented in the BTM51x in order to present AVRCP1.5 data to a host controller via UART:

▪ Available players	(AT+AVQAP)
▪ Current play list	(AT+AVQCL)
▪ (virtual) File system	(AT+AVQFS)
▪ Now playing track	(AT+AVQNP)
▪ Player application settings	(AT+AVQPV)

Each list begins with a start tag, followed by a variable number if items, and is terminated by another end tag. Start and end tags allow the module to identify the context of a list in cases where item tags can be ambiguous (such as AVMEI).

Table 2-41: AVRCP1.5 - list tags overview

List Type →	Available Players	Current Play List	File System	Now Playing Track	Player Application Settings
Start Tag	AVAPL <nItems>	AVCPL <nItems>	AVFSL …	AVNPT <nAtt>	AVPL <nItems> or AVPC <nItems>
Item Tag	AVAPI …	AVMEI …	AVFSI … *or* AVMEI …	AVNPA …	AVPV … *or* AVPA … *or* AVPR …
End Tag	AVAPL OK *or* AVAPL ERROR …	AVCPL OK *or* AVCPL ERROR...	AVFSL OK *or* AVFSL ERROR…	AVNPT OK *or* AVNPT ERROR…	AVPL OK or AVPL ERROR … *or* AVPC OK

2.6.6.6 AVRCP1.5 message syntax

Table 2-42: AVRCP1.5 message syntax

Message/Tag	Type	Syntax	Comment
AVAPL	Available players list start / end	AVAPL \$<nItems$_{16h}$>	start tag, nItems: number of items to follow, UI16 hex
		AVAPL OK	end tag, success

		AVAPL ERROR \$<asc$_{16h}$>	end tag, error
			asc=avrcp status code, UI16 hex

AVCPL	Current play list start / end	AVCPL \$<nItems>	start tag, nItems: number of items to follow, UI16 hex
		AVCPL OK	end tag, success
		AVCPL ERROR \$<asc$_{16h}$>	end tag, error asc: avrcp status code, UI16 hex
AVFSL	File system list start / end	AVFSL \$<nItems>,\$<nStartOffs>, \$<nEndOffs>,\$<nTotalItems>	start tag nItems: number of items to follow, UI16 hex nStartOffs: index of first item, UI32 hex nEndOffs: index of last item, UI32 hex nTotalItems: total number of items in current folder, UI32 hex
		AVFSL OK	end tag, success
		AVFSL ERROR \$<asc$_{16h}$>	end tag, error asc: avrcp status code, UI16 hex
AVNPT	Now playing track attributes start / end	AVNPT \$<nAtt$_{16h}$>	start tag, nAtt: number of attributes to follow, UI16 hex
		AVNPT OK	end tag, success

		AVNPT ERROR \$<asc$_{16h}$>	end tag, error
			asc=avrcp status code, UI16 hex

AVPL	Player application setting response list	AVPL \$<nItem$_{16h}$>	start tag, nAtt: number of items to follow, UI16 hex
		AVPL OK	end tag, success

Message/Tag	Type	Syntax	Comment

	start / end
	(response to AT+AVQPV/A)
		AVPL ERROR \$<asc$_{16h}$>	end tag, error
			asc=avrcp status code, UI16 hex

AVPC	Player application setting changed list	AVPC \$<nAtt$_{16h}$>	start tag (indication), nAtt: number of attributes to follow, UI16 hex
	start / end *(indication of value change)*	AVPC OK Between start and end message, <ni> items of "AVPV" (attribute values) messages are sent	end tag, success
AVAPI	Available player item	AVAPI <id$_{16h}$>,<mt$_{8h}$>, <st$_{32h}$>,<ps$_{8h}$>,<cs$_{16h}$>,<nl$_{16h}$>, "<dn$_{8s}$>" Options: S390: - Hint string inserted after <ps$_{8h}$> - <dn> printed in hexadecimal format	id: player id, UI16 hex mt: player major type, UI8 hex st: player sub type, UI32 hex ps: play status, UI8 hex cs: character set, UI16 hex nl: length of displayable name, UI16 hex dn: displayable name: string of indicated character set, non-printable characters presented as \xx
	Example:	AVAPI 0001,01,00000000,02,006A,0005,"Music"	
AVFSI	File system item (folder)	AVFSI <uid$_{64h}$>,<ft$_{8h}$>, <ip$_{B}$>,<cs$_{16h}$>,<nl$_{16h}$>,"<dn$_{8s}$>" **Options:** **S390:** ▪ Hint string inserted after <ft$_{8h}$> ▪ <dn> printed in hexadecimal format **S391:** ▪ UidCache index "[n]" inserted after AVFSI	uid: item UID, UI64 hex ft: folder type, UI8 hex, see Table 2-45 ip: flag 'is_playable', BOOL cs: character set, UI16 hex nl: length of displayable name, UI16 hex dn: displayable name: string of indicated character set, non-printable characters presented as \xx; optionally: hexadecimal format (see S390)

Message/Tag	Type	Syntax	Comment
	Examples:	AVFSI[1] 0F8BBB8D26804017,01"Titles",0,006A,000B,"Barry White" AVFSI 0F8BBB8D26804017,01,0,006A,000B,4261727279205768697465	
AVMEI	Media element item (track)	AVMEI <uid$_{64h}$>,<met$_{8h}$>, <na$_{8h}$>,<cs$_{16h}$>,<nl$_{16h}$>,"<dn$_{8s}$>" **Options:** **S390:** ▪ Hint string inserted after <met$_{8h}$> ▪ <dn> printed in hexadecimal format ▪ attributes appended (AVMEA) in subsequent line **S391:** ▪ UidCache index "[n]" inserted after AVMEI	uid: item UID, UI64 hex met: media el. type, UI8 hex, see Table 2-46. na: number of attributes to follow, UI8 hex cs: character set, UI16 hex nl: length of displayable name, UI16 hex dn: displayable name: string of indicated character set, non-printable characters presented as \xx; optionally: hexadecimal format (see S390)
	Examples:	AVMEI D75D439C1A32E511,00,01,006A,0012,"Let The Music Play" AVMEI[3] D75D439C1A32E511,00"Audio",01,006A,0012,"Let The Music Play"	
AVMEA	Media element attribute (browsing)	AVMEA <attid$_{8h}$>,<cs$_{16h}$>,<vl$_{16h}$>, "<av$_{8s}$>" **Options:** **S390:** ▪ Hint string inserted after <attid$_{8h}$> ▪ <dn> printed in hexadecimal format, without '"'	attid: attribute ID, UI8 hex, see Table 2-43. cs: character set, UI16 hex vl: length of attribute value, UI16 hex av: attribute value: string of indicated character set, non-printable characters presented as \xx; optionally: hexadecimal format (see S390)
	Examples:	AVMEA 01"Title",006A,0014,"Just The Way You Are" AVMEA 01,006A,0014,4A757374205468652057617920596F7520417265	

Message/Tag	Type	Syntax	Comment
AVNPA	Now playing track attribute	AVNPA<pt> <attid$_{8h}$>,<cs$_{16h}$>, <vl$_{16h}$>, ”<av$_{8s}$>” **Options:** S390: ▪ Hint string inserted after <attid$_{8h}$> ▪ <av> printed in hexadecimal format, without ‘”’	pt: packet type: 0=single / 1=Start / 2=Continue / 3=End see section 2.6.5.10 attid: attribute ID, UI8 hex, see Table 2-43. cs: character set, UI16 hex vl: length of attribute value, UI16 hex av: attribute value: string of indicated character set, non-printable characters presented as \xx; optionally: hexadecimal format (see S390)
	Examples:	AVNPA0 02"Artist",006A,000B,"Barry White" AVNPA0 02,006A,000B,4261727279205768697465	
AVTC	Track changed notification	AVTC <uid$_{64h}$>	uid: UID of new track, UI64 hex if S390 bit5 == 0: automatically query and display attributes of now playing track (like AT+AVQNP)
AVPV	Player application setting: attribute/ value pair	AVPV <nAttId$_{8h}$>,<nVal$_{8h}$> **Option:** S390 bit0 == 1: Hint strings inserted after <nAttId$_{8h}$> and after <nVal$_{8h}$>	nAttId: attribute ID, UI8 hex, see Table 2-44. nVal: attribute value, UI8 hex, see Table 2-44.
AVPA	Player application setting: attribute only	AVPA <nAttId$_{8h}$> **Option:** S390 bit0 == 1: Hint strings inserted after <nAttId$_{8h}$>	nAttId: attribute ID, UI8 hex, see Table 2-44.

Message/Tag	Type	Syntax	Comment
AVPT	Player application setting attribute text	AVPT <nAttId$_{8h}$>,<nl$_{8h}$>,"<at$_{8s}$>" **Option:** S390 bit0 == 1: Hint strings inserted after <nAttId$_{8h}$>	nAttId: attribute ID, UI8 hex, see Table 2-44. nl: length of displayable name, UI8 hex at: displayable name: string of indicated character set, non-printable characters presented as \xx; optionally: hexadecimal format (see S390)
AVPE	Player application setting attribute value text	AVPE <nAttId$_{8h}$>,<nValId$_{8h}$>, <nl$_{8h}$>,"<vt$_{8s}$>" **Option:** S390 bit0 == 1: Hint strings inserted after <nAttId$_{8h}$> and <nValId8h>	nAttId: attribute ID, UI8 hex, see Table 2-44. nValId: value ID, UI8 hex nl: length of displayable name, UI8 hex vt: value text: string with non-printable characters presented as \xx; optionally: hexadecimal format (see S390)
AVAPC	Available players changed notification	AVAPC	No parameters; -automatic query of available players list if S390, bit 7 (0x80)==0 -message can be supressed by S397 bit 5 (0x20) ==1
AVSPC	Addressed ('selected') player changed notification	AVSPC $<plid$_{16h}$ >,$<ucnt$_{16h}$ >	plid: player ID, UI16 hex ucnt: UID counter, UI16 hex
AVNPC	Now playing content has changed notification	AVNPC No parameters	AVNPC notification can be supressed by S397 bit3 == 1 (0x08)
AVPS	Play status changed notification	AVPS <ps$_{8h}$> **Options:** if S390 bit0 == 1: Hint string inserted after <ps$_{8h}$>	ps: play status, UI8 hex, see Table 2-47. If S390 bit4 == 0 and new status is 'playing' (0x01): automatically query and display attributes of now playing track (like AT+AVQNP)

Message/Tag	Type	Syntax	Comment
AVPP	Play status and parameters, response to AT+AVQPP	AVPP <ps_{8h}>,$<$el_{32h}$ >,$<$len_{32h}$ > **Options:** if S390 bit0==1: Hint string inserted after <ps_{8h}>	ps: play status, UI8 hex, see Table 2-47. el: elapsed time in ms, UI32 hex len: song length in ms, UI32 hex
AVPPC	Playback position changed notification	AVPPC $<$pos_{32h}$>	pos: playback position in ms, UI32 hex
AVVC	Absolute volume changed indication	AVVC $<$av_{8h}$>,$<go_{8h}>	av: absolute volume, UI8 hex, range 0…127 go: corresponding output gain, UI8 hex, range 0…15
AVSBP	Set browsed player confirmation	*Successful:* AVSBP $<$plid_{16h}$>,$<ni_{16h}> *Not successful:* AVSBP ERROR $<$asc_{16h}$> part of 'AT+AVSBP' response, sent while AT parser is suspended and prior to 'OK'	plid: player ID, UI16 hex ni: number of items in browsed player root path, UI16 hex asc: avrcp status code, UI16 hex
AVBS	Battery status of remote TG changed notification	AVBS <nBS_{8h}> Options: if S390 bit0==1: Hint string inserted after <nBS_{8h}>	nBS: battery status
AVSSP	Set addressed ('selected') player confirmation	*Successful:* AVSSP $<$plid_{16h}$> *Not successful:* AVSSP ERROR $<$asc_{16h}$> part of 'AT+AVSSP' response, sent while AT parser is suspended and prior to 'OK'	plid: player ID, UI16 hex asc: avrcp status code, UI16 hex

Message/Tag	Type	Syntax	Comment
AVFC	Folder (path) changed confirmation	*Successful:* AVFC $<ni$_{16h}$> *Not successful:* AVFC ERROR $<asc$_{16h}$> part of "set path" command family ('AT+AVSP…') response, sent while AT parser is suspended and prior to 'OK'	ni: number of items in new folder, UI16 hex asc: avrcp status code, UI16 hex
AVPI	Play item confirmation	*Successful:* AVPI $0000 *Not successful:* AVPI ERROR $<asc$_{16h}$> part of 'AT+AVSPI' response, sent while AT parser is suspended and prior to 'OK'	asc: avrcp status code, UI16 hex
AVSS	System status of remote TG changed notification	AVSS <nSS$_{8h}$> Options: if S390 bit0==1: Hint string inserted after <nSS$_{8h}$>	nSS: system status
AVTE	Track reached end notification	AVTE	No parameters
AVTS	Track reached start notification	AVTS	No parameters
AVUC	UIDs changed notification	AVUC $<cnt$_{16h}$>	cnt: UID counter, UI16 hex

Message/Tag	Type	Syntax	Comment
Notes: (*) All messages of this table are surrounded by <CRLF> (0x0D 0x0A) (*) BOOL / $_{B}$ = '0' or '1' (*) UI8 hex / $_{8h}$ = 2 hexadecimal digits (*) UI16 hex / $_{16h}$ = 4 hexadecimal digits (*) UI32 hex / $_{32h}$ = 8 hexadecimal digits (*) UI64 hex / $_{64h}$ = 16 hexadecimal digits <asc> - avrcp status code, see Table 2-50.			

Table 2-43: Media element attributes

Media Element Attributes	
Attribute ID	Description
0x01	Title
0x02	Artist Name
0x03	Album Name
0x04	Track Number
0x05	Total Number of Tracks
0x06	Genre
0x07	Playing Time (ms)
Note: See also AVRCP1.5 spec, p.132	

Table 2-44: Player application setting attributes

Player Application Setting Attributes		
Attribute ID	Description	Values
0x01	Equaliser ON/OFF status	1=OFF 2=ON
0x02	Repeat Mode status	1=OFF 2=Single Track repeat 3=All track repeat 4=Group repeat
0x03	Shuffle Mode status	1=OFF 2=All tracks shuffle 3=Group shuffle

Player Application Setting Attributes		
0x04	Scan Mode status	1=OFF 2=All tracks scan 3=Group scan
Note: See also AVRCP1.5 spec, p.133		

Table 2-45: Folder types

Value	Description
0x00	Mixed
0x01	Titles
0x02	Albums
0x03	Artists
0x04	Genres
0x05	Playlists
0x06	Years
Note: See also AVRCP1.5 spec, p.70	

Table 2-46: Media element types

Value	Description
0x00	Audio
0x01	Video
Note: See also AVRCP1.5 spec, p.71	

Table 2-47: Play status

Value	Description
0x00	Stopped
0x01	Playing
0x02	Paused
0x03	Fwd Seek
0x04	Rev Seek
Note: See also AVRCP1.5 spec, p.67	

Table 2-48: Event IDs for notification

Event ID Mask	Description

0x0001	Playback status changed
0x0002	Track changed
0x0004	Track reached end
0x0008	Track reached start
0x0010	Playback position changed
0x0020	Battery status changed
0x0040	System status changed
0x0080	Player application setting changed
0x0100	Now playing content changed
0x0200	Available players changed
0x0400	Addressed player changed
0x0800	UIDs changed
0x1000	Volume changed
Note: See also AVRCP1.5 spec, p.135	

Table 2-49: AVRCP TG features (SDP record)

Feature Mask	Description
0x0001	Category 1
0x0002	Category 2
0x0004	Category 3
0x0008	Category 4
0x0010	Player application settings
0x0020	Group navigation
0x0040	Supports browsing
0x0080	Supports media player applications
Note: See also AVRCP1.5 spec, p.91	

Table 2-50: AVRCP status codes

Status Code (hex)	Description
$0000	Success
$0001	Operation failed
$0002	Not enough resources
$0003	Request not supported in current state
$0004	Time out
$0005	Not connected
$0006	Busy

Status Code (hex)	Description
$0007	Request not supported
$0008	Invalid sink
$0009	Link loss occurred
$000A	Operation was rejected
$000B	General failure during browsing channel initialisation
$000C	Browsing channel not connected
$000D	Browsing not supported by remote device
$000E	Browsing timeout
$000F	Interim response success
$0010	Link key missing in remote device
$0080	Operation rejected due to invalid PDU
$0081	Operation rejected due to invalid parameter
$0082	Operation rejected due to invalid content
$0083	Operation rejected due to internal error
$0085	Operation rejected due to UID changed
$0087	Operation rejected due to invalid direction
$0088	Operation rejected due to not a directory
$0089	Operation rejected due to invalid UID
$008A	Operation rejected due to invalid scope
$008B	Operation rejected due to range out of bound
$008C	Operation rejected due to UID is a directory
$008D	Operation rejected due to media in use
$008E	Operation rejected due to now playing list is full
$008F	Operation rejected due to search not supported
$0090	Operation rejected due to search in progress
$0091	Operation rejected due to invalid player ID
$0092	Operation rejected due to player not browseable
$0093	Operation rejected due to player not addressed
$0094	Operation rejected due to no valid search results
$0095	Operation rejected due to no available players
$0096	Operation rejected due to addressed player changed

Table 2-51: AVRCP1.5 supported features (CT only)

Feature		Support in CT (Controller)		Comment
		Spec.	BTM51x	
1	Connection establishment for control	M	Yes	
2	Release connection for control	M	Yes	
3	Connection establishment for browsing	C6	Yes	Background operation
4	Release connection for browsing	C6	Yes	Background operation
5	AV/C Info commands	O	partly	See table in BTM51x UM
6	Category 1: Player/Recorder	C3	Yes	
7	Category 2: Monitor/Amplifier	C3	Yes	abs. volume TG role
8	Category 3: Tuner	C3	untested	Not evaluated
9	Category 4: Menu	C3	untested	Not evaluated
10	Capabilities	O	Yes	
11	Player Application Settings	O	Yes	
12	Metadata Attributes for Current Media Item	O	Yes	
13	Notifications	C2	Yes	
14	Continuation	C2	Yes	Tested by PTS only, not by test TG
15	Basic Group Navigation	O	No	
16	Absolute Volume	C4	Yes	
17	Media Player Selection	O	Yes	
17	1 - Supports Multiple Players	O	Yes	
18	Browsing	O	Yes	
18	1 - Database Aware Players	O	Yes	UID changed notification
19	Search	O	No	
20	Now Playing	C6	Yes	

Feature		Support in CT (Controller)		Comment
20	1 - Playable Folders	O	Yes, untested	Flag 'is_playable' is part of AVFSI message
21	Error Response	X	n/a	
22	PASSTHROUGH operation supporting press and hold	O	Yes	

Note: TG role is still supported on BTM51x with AVRCP version 1.0 features, which is not covered by this table. Please refer to section AVRCP 1.0 of this document

2.6.7 HSP (Headset Profile)

The Headset Profile provides full-duplex audio capability combined with minimal device control commands. Audio bandwidth is limited and deemed to be sufficient for voice links. Table 2-52 lists the feature requirements of the HSP specification [4] and the level of support by BTM5xx.

Version 1.2 of the Headset Profile specification is supported.

In most cases it is preferable to use the more advanced Hands-Free Profile (HFP).

HSP defines the role of the headset (HS) and the role of the audio gateway (AG) which are both supported on BTM5xx. They are enabled by setting the appropriate flag in S-Register 102 (plus subsequent AT&W and atz).

An HSP connection has two states: ACL connected or audio connected. The ACL is initiated by either HS or AG. The audio connection (a SCO link) is always initiated and released by the AG.

A host processor is required, using AT commands to control the BTM5xx module (hosted operation mode).

An AT command beginning with AT+HS… indicates affiliation to the Headset role of HSP.

An AT command beginning with AT+HSG…indicates affiliation to the Audio Gateway role of HSP.

As of f/w version 22.2.5.0, it is not possible any longer to initiate or accept a connection from or to HSP if a link to the 'sister' profile instance (HFP) already exists.

Example: if a link to a local HS profile instance exists, no more links can be initiated/accepted for the local HF profile. Or, if a link exists to a local HFG instance, no more links can be made from/to the local HSG instance and vice versa.

This is caused by the new underlying version of BT stack and library. It has not been evaluated, whether the feature of earlier f/w version can be restored.

Table 2-52: Headset Profile supported features on BTM5xx

HSP Feature		Support in HS		Support in AG	
		Specification	BTM5xx	Specification	BTM5xx
1.	Incoming audio connection	M	Yes	M	Yes
2.	Outgoing audio connection	M	Yes	O	Yes
3.	Audio connection transfer	M	Yes	M	Yes
4.	Remote audio volume control	O	Yes	O	Yes

M – Mandatory O – Optional

2.6.7.1 Headset role (HS)

Headset role is activated by setting flag 0x02 in S102 plus AT&W followed by ATZ.

Figure 2-24 illustrates a HSP link and how a BTM5xx module is integrated in hosted operation mode.

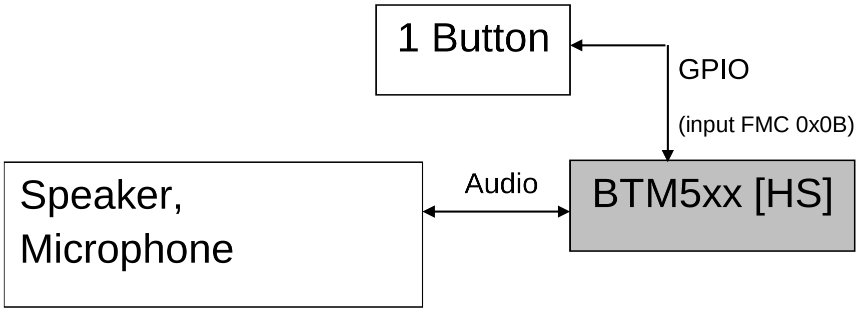


Figure 2-24: Headset block diagram

2.6.7.1.1 Initiate ACL connection from HS

AT+HSD<bd_addr$_{hex}$>

Initiate ACL connection from local headset instance to remote device with <bd_addr$_{hex}$>. The remote device must support the audio gateway role (AG) of the headset profile (HSP).

Response: <cr,lf>CONNECT 123456789012,1112,><cr,lf>

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

Or: <cr,lf>ERROR 59<cr,lf>

Or: <cr,lf>ERROR 63<cr,lf>

After an ACL connection has been established, the module remains in AT command mode. S Register 531 is ignored for HSP connections. An audio connection should be established within short time. The command is AT+HSB.

2.6.7.1.2 Send AT+CKPD=200

AT+HSB

Send "AT+CKPD=200" to connected audio gateway. This indicates a user action on the gateway, e.g. a button press. Depending on the audio connection state, the audio gateway either establishes or releases the audio connection subsequently. Refer to Figure 2-26.

Table 2-53: Outcome of AT+HSB

Connection status	Outcome of AT+HSB ("AT+CKPD=200")
ACL connected (ATI63=1)	Audio link is to be initiated by AG, Referred to as *Audio Connection Transfer from AG to HS* in HSPv1.2
Audio connected (ATI63=2)	Audio link and ACL should be released by the AG, actual outcome depends on AG

2.6.7.1.3 Send custom data over ACL (HS)

AT+HSX"<data>"

Send custom data <data> over ACL from HS instance. Please refer to section 2.9.24 for details on utilizing ACL for custom data.

2.6.7.1.4 Release connection from headset

AT+HSH

Release connection from local Headset instance. Audio connection is released if existing, ACL is released regardless.

A connection release initiated by the Headset is not defined in the Headset Profile specification [4]. A profile compliant disconnection is initiated from Headset by sending "AT+CKPD=200" (use "AT+HSB") to the audio gateway.

This command exists for the sake of completeness and is for testing purposes only.

Response:

On audio disconnection (optional): <cr,lf>HS"AU0"<cr,lf>
On ACL disconnection: <cr,lf>NO CARRIER 1108<cr,lf>

2.6.7.1.5 Headset status

ATI63

Returns the status of the Headset (HSP) instance:

0 = not connected
1 = ACL connected
2 = Audio connected

2.6.7.1.6 Headset audio routing

Audio in the headset profile is always referred to as a SCO connection for bidirectional speech transmission.

On a BTM5xx, SCO audio is routed by default to the internal codec. S315 selects the I2S interface in master or slave mode for SCO connections. Refer to Routing audio streams over I2S.

2.6.7.1.7 Headset asynchronous messages

CONNECT <bd_addr$_{hex}$>,<uuid$_{hex}$>[,<dir>]

An ACL connection to headset has been established.

<bd_addr$_{hex}$>	: Bluetooth address of headset device
<uuid$_{hex}$>	: 1108 (incoming connection) 1112 (outgoing connection)
<dir>	: </>/I/O optionally indicates the direction (incoming/outgoing),

Refer to S331 and UUIDs in CONNECT/NO CARRIER Messages.

HS"RING"

HS has received a RING indication from the connected audio gateway. HS is expected to respond with AT+CKPD=200 (see AT+HSB).

FS8000,INT

The internal codec is configured for a sampling frequency of 8000 Hz.

HS"AU1"

Audio connection (SCO) has been established (= audio on).

HS"AU0"

Audio connection (SCO) has been released (= audio off).

HSRX"<data>"

Spy messages displaying incoming data on the ACL (rfcomm of HSP link).

HSUC"<data>"

Message displaying unrecognized command data from ACL (rfcomm of HSP link).

NO CARRIER 1108

ACL connection to local HS-instance has been released. Note UUIDs in CONNECT/NO CARRIER Messages as well.

2.6.7.1.8 Headset Summary

Table 2-54: Headset role (HSP) – Summary of S Registers and AT Commands

<table>
<tr><th>Task</th><th>AT-Command / S Register</th><th>Comment</th></tr>
<tr><td>Enable HSP-HS role</td><td>S102</td><td>0x02 = HS role of HSP (bitmask),
needs subsequent AT&W and atz to activate.</td></tr>
<tr><td>Initiate ACL from headset</td><td>AT+HSD<bd_addr$_{hex}$></td><td><b>Responses:</b><table><tr><td>If successful</td><td>CONNECT 123456789012,1112,></td></tr><tr><td>If failed</td><td>NO CARRIER</td></tr><tr><td>If profile disabled</td><td>ERROR 59</td></tr><tr><td>If incorrect state</td><td>ERROR 63</td></tr></table></td></tr>
<tr><td>Send “Button pressed” (HS)</td><td>AT+HSB</td><td>Sends “AT+CPKD=200” to the connected gateway.</td></tr>
<tr><td>Disconnect from HS</td><td>AT+HSH / ATH1108 / ATH*</td><td>For test purposes only, because disconnection initiated by HS other than sending AT+CKPD=200 to gateway is not defined in HSP specification.</td></tr>
<tr><td>Send custom data on HS-instance ACL</td><td>AT+HSX”<data>”\r</td><td>Maximum data length – 26 characters
<data> = a string of printable ASCII characters or the two-digit hexadecimal presentation of a non-printable character, preceded by <b>\</b>.
\r must be used to terminate a command or block of data.
This feature can be used to transmit short data sequences or commands but is not suitable for transmitting large amounts of data. Refer to Figure xx</td></tr>
<tr><td>Enable “ATH” for HS</td><td>S332</td><td>Enable HS disconnection ATH1108 and ATH*<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled (default)</td></tr></table>Should only be enabled for test purposes because disconnection initiated by HS other than sending AT+CKPD=200 to gateway is not defined in HSP specification.</td></tr>
</table>

Task	AT-Command / S Register	Comment
Select SCO audio interface (BTM5xx only) Catering for PCM configuration as master or slave	S315	SCO audio interface select: 0 Analogue (on-board audio codec) – Default 1 I2S master 2 I2S slave 3 PCM master (PCM_M) 4 PCM slave (PCM_S) Refer to Routing audio streams over I2S.
Query HS status	ATI63	0 = not connected 1 = ACL connected 2 = Audio connected

2.6.7.2 Audio gateway role (AG-HSP / HSG)

Audio gateway role (for HSP) is activated by setting flag 0x08 in S102 plus "AT&W" plus "atz". Figure 2-25 outlines a block diagram of an audio gateway with a BTM5xx in hosted operation mode.

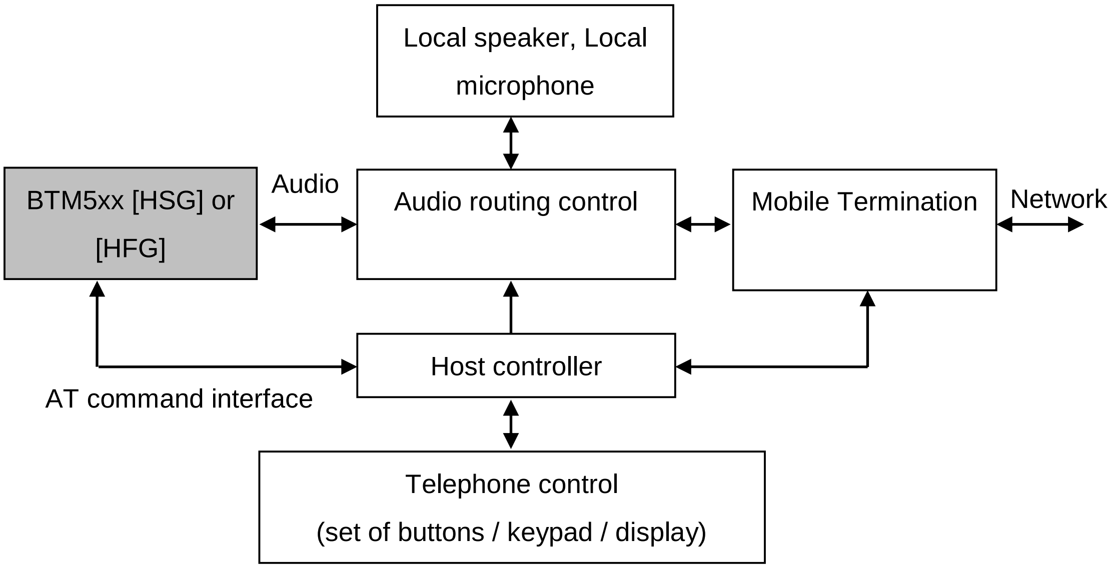


Figure 2-25: Audio Gateway block diagram

2.6.7.2.1 Initiate ACL connection from AG (HSP)

AT+HSGD<bd_addr$_{hex}$>

Initiates an ACL connection from local headset-gateway instance to a remote device, <bd_addr$_{hex}$>. The remote device must support the headset role (HS) of the headset profile (HSP).

Response:	<cr,lf>CONNECT 123456789012,1108,><cr,lf>
Or:	<cr,lf>NO CARRIER<cr,lf>
Or:	<cr,lf>ERROR 59<cr,lf>
Or:	<cr,lf>ERROR 63<cr,lf>

After an ACL connection is established, the module remains in AT command mode. S Register 531 is ignored for HSP connections.

If S-Register 345=1, the audio gateway alerts the headset immediately by sending RING or performing the in-band ringing procedure (depending on S Register 346) and waits for the response AT+CKPD=200.

If S-Register 345=0, an incoming call must be signalized to the HS manually by issuing "AT+HSGC". This either sends the "RING" message to the HS or performs the in-band ringing procedure and waits for AT+CKPD=200 to arrive from the HS.

If the headset does not respond, the host must to terminate the connection by "AT+HSGH".

2.6.7.2.2 Alert HS by sending RING message or in-band ringing

AT+HSGC

When in ACL connected state, the command AT+HSGC alerts the HS either by sending RING (if S346=0) or by in-band ringing (if S346=1). In the latter the AG is assumed to create an in-band ringtone which is audible on the HS through the SCO connection. The in-band ringtone is not created by BTM5xx itself. If in-band ringing is disabled, then BTM51x is capable of playing a ringtone locally on incoming 'RING' messages. Refer to S388 and AT+SIT<n>

In-band ringing is indicated by ATI64 returning 6. If ATI64 returns 5, alerts are indicated with the RING message.

In both cases the response AT+CKDP=200 (Button pressed) is expected from the HS.

2.6.7.2.3 Cancel an ongoing alert process

AT+HSGCH

When the HSG is alerting the HS (ATI64 = 5 / 6) this state can be cancelled by AT+HSGCH. If in-band ringing is active (ATI64 = 6), the audio connection is released.

2.6.7.2.4 Release audio from HSG side / transfer audio from HS to HSG

AT+HSGR

When in Audio connected state, the AG can transfer the audio connection from the headset to itself with AT+HSGR. This releases the audio connection but the ACL connection is retained. In ACL connected state, the headset (HS) is able to transfer the audio connection back to the HS by sending AT+CKPD=200.

2.6.7.2.5 Initiate audio from HSG side / transfer audio from HSG to HS

AT+HSGA

When in ACL connected state, the AG can transfer the audio connection from itself to the headset by AT+HSGA. This establishes an SCO-based audio connection.

2.6.7.2.6 Send custom data over ACL (HSG)

AT+HSGX"<data>"

Send custom data <data> over ACL from HSG instance. Please refer to section 2.9.24 for details on utilizing ACL for custom data.

2.6.7.2.7 Connection release from AG

AT+HSGH

Release existing connection to a HS. If the HS is alerted, alerting is cancelled. If an audio connection exists it is released first. ACL is released in either case.

Response:

On audio disconnection (optional)	<cr,lf>HSG"AU0"<cr,lf>
On SLC disconnection	<cr,lf>NO CARRIER 1112<cr,lf>

2.6.7.2.8 Enable automatic alerting on SLC establishment

S345

S-Register 345 enables automatic alerting on ACL establishment. It contains a bitmask where bit 0 corresponds to outgoing ACL connections and bit 1 corresponds to incoming ACL connections.

If automatic alerting is disabled (S345=0), it can be initiated manually by AT+HSGC.

2.6.7.2.9 Enable in-band ringing

S346

In-band ringing is enabled by S-Register 346 (value =1). If disabled (S346=0), the RING message is send instead to alert the HS.

2.6.7.2.10 Enable automatic SLC release

S347

If the HSG is in Audio connected state and the HS initiates an audio release by AT+CKPD=200 (Button pressed), the ACL is released too if S-Register 347 = 1. If S347=0, it is up to the user to release the ACL manually by AT+HSGH or retain the ACL by doing nothing.

2.6.7.2.11 Query/Set/Increment/Decrement remote gain of HS

AT+HSGG…

Syntax: AT+HSGG<I/O><?/n/U/D> with:

I/O	I – Input (microphone) O – Output (speaker)
?	Query current/cached remote gain value

<n>	Set remote gain value. Valid range: 0..15 (dec)
U	Up – Increments of one (1)
D	Down – Increments of one (1)
ERROR 129	Indicates that a cached gain value is not available. This can happen if HS does not support the remote volume control feature (missing notifications).

Examples:

AT+HSGGOU – increment speaker gain of connected HS

AT+HSGGI? – query current (cached) microphone gain of connected HS

AT+HSGGI13 – set microphone gain of connected HS to 13

2.6.7.2.12 Headset gateway audio routing

Audio in the headset profile is always referred to as a SCO connection which is used for bidirectional transmission of speech.

On a BTM5xx, SCO audio is routed to the internal codec by default. S315 can select the I2S interface in master or slave mode for SCO connections. Refer to Routing audio streams over I2S.

2.6.7.2.13 Receiving commands from SLC

There are two taps to display incoming SLC/ACL data. The first is before the SLC (ACL) parser and the second is the one after the parser if the command is not recognized. The first tap should be used for debugging or learning purposes; the second should be used for customer commands. See Figure xx.

The following explains how to enable the asynchronous tap messages:

Temporary (valid until power cycle or until reverted):

- AT+RX1 – Enable all SLC RX messages (first tap)
- AT_UC1 – Enable all unknown command messages (second tap)

Permanently (valid as of next power cycle):

- S379=1 – Enable all SLC RX messages (first tap)
- S380=1 – Enable all unknown command messages (second tap) plus subsequent AT+W and ATZ.

Asynchronous Message	Description
HSRX"<data>"/HSGRX"<data>"	Display any data received on HSP ACL.
HFRX"<data>"/HFGRX"<data>"	Display any data received on HFP SLC
HSUC"<data>"/HSGUC"<data>"	Display unknown command received on HSP ACL
HFUC"<data>"/HFGUC"<data>"	Display unknown command received on HSP ACL

2.6.7.2.14 AG status

ATI64
Returns the status of the Audio Gateway (HSP) instance:

0	Not connected
1	ACL connected
2	Audio connected

5	Alerting HS by sending RING
6	Alerting HS by in-band ringing

2.6.7.2.15 AG asynchronous messages

CONNECT <bd_addr$_{hex}$>,<uuid$_{hex}$>[,<dir>]	An ACL connection to headset has been established. <bd_addr$_{hex}$> Bluetooth address of headset device <uuid$_{hex}$> 1112 – Incoming connection 1108 – Outgoing connection <dir> <"/">/I/O optionally indicates the direction (incoming/outgoing). Refer to S331 and UUIDs in CONNECT/NO CARRIER Messages.
FS8000,INT	The internal codec is configured for a sampling frequency of 8000 Hz.
HSG"AU1"	Audio connection (SCO) is established – Audio on
HSG"AU0"	Audio connection (SCO) has been released – Audio off
HSG"B"	AT+CKPD=200 (Button pressed) was received from HS
HSG"VGSnn"	AT+VGS=nn was received from HS (speaker gain setting). Valid range for nn: 00..15.
HSG"VGMnn"	AT+VGM=n was received from HS (microphone gain setting. Valid range for n: 00..15.
HSGRX"<data>"	Spy messages displaying incoming data on the ACL (rfcomm of HSP link)
HSGUC"<data>"	Message displaying unrecognized command data from ACL (rfcomm of HSP link)
NO CARRIER 1112	ACL connection to local AG-instance has been released. See UUIDs in CONNECT/NO CARRIER Messages.

2.6.7.2.16 AG - HSP summary (HSG)

Table 2-55: Audio gateway role (HSP) – Summary of S Registers and AT Commands

Task	AT-Command/ S Register	Comment
Enable HSP-AG role	S102	0x08 = AG role of HSP (bitmask), Must issue AT&W and atz to activate.
Initiate SLC from AG (HSP)	AT+HSGD<bd_addr$_{hex}$>	**Response:** If successful CONNECT 123456789012,1108,> If failed NO CARRIER If wrong state ERROR 63 If profile disabled ERROR 59
Alert HS by RING or in-band ringing	AT+HSGC	An existing SLC is required. RING / in-band-ringing is controlled by S346. Automatic alerting is set using S345.

<table>
<tr><th>Task</th><th>AT-Command/
S Register</th><th>Comment</th></tr>
<tr><td>Cancel alerting before answered by HS</td><td>AT+HSGCH</td><td>HSG must be in alerting state (ATI64=5 or =6)
HSG stops sending RING-message or release audio (in-band ringing)</td></tr>
<tr><td>Initiate audio / Transfer audio from AG to HS</td><td>AT+HSGA</td><td>An existing SLC is required.</td></tr>
<tr><td>Release audio connection / Transfer audio from HS to AG</td><td>AT+HSGR</td><td><b>Response:</b>
<table>
<tr><td>If successful</td><td>OK and HSG”AU0”</td></tr>
<tr><td>If wrong state</td><td>ERROR 63</td></tr>
<tr><td>If profile/role disabled</td><td>ERROR 59</td></tr>
</table></td></tr>
<tr><td>Release audio and SLC from AG</td><td>AT+HSGH or
ATH1112 or
ATH*</td><td><b>Response:</b>
<table>
<tr><td>If successful</td><td>NO CARRIER 1112
AG(HSP) was previously connnected</td></tr>
<tr><td>If wrong state</td><td>ERROR 63</td></tr>
<tr><td>If profile/role disabled</td><td>ERROR 59</td></tr>
</table></td></tr>
<tr><td>Enable automatic alerting on ACL establishment</td><td>S345</td><td>
<table>
<tr><td>0</td><td>Disable auto alerting</td></tr>
<tr><td>1</td><td>Auto alerting on outgoing ACL established</td></tr>
<tr><td>2</td><td>Auto alerting on incoming ACL established</td></tr>
<tr><td>3</td><td>Auto alerting on outgoing and incoming ACL established</td></tr>
</table></td></tr>
<tr><td>Enable in-band ringing</td><td>S346</td><td>
<table>
<tr><td>0</td><td>Disable in-band ringing</td></tr>
<tr><td>1</td><td>Enable in-band ringing</td></tr>
</table></td></tr>
<tr><td>Enable automatic ACL release</td><td>S347</td><td>
<table>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>ACL is released automatically when audio is released by the HS</td></tr>
</table></td></tr>
<tr><td>Select SCO audio interface (BTM5xx only)</td><td>S315</td><td>SCO audio interface select:
<table>
<tr><td>0</td><td>Internal codec (Default)</td></tr>
<tr><td>1</td><td>I2S master</td></tr>
<tr><td>2</td><td>I2S slave</td></tr>
<tr><td>3</td><td>PCM master (PCM_M)</td></tr>
<tr><td>4</td><td>PCM slave (PCM_S)</td></tr>
</table>
Refer to Routing audio streams over I2S.</td></tr>
</table>

Task	AT-Command/ S Register	Comment
Query AG status	ATI64	0 Not connected 1 ACL connected 2 Audio connected 5 Ringing (ACL connected) 6 In-band ringing (audio connected)
Send custom data on HSG-instance ACL	AT+HSGX"<data>"	Maximum data length – 25 characters <data> = a string of printable ASCII characters or the two-digit hexadecimal presentation of a non-printable character, preceded by ****. \r must be used to terminate a command or block of data. This feature can be used to transmit short data sequences or commands but is not suitable for transmitting large amounts of data. Refer to Figure xx

2.6.7.3 Remote volume control (HSP)

Remote volume control feature for HSP allows the audio gateway (HSG) to control microphone- and speaker gains of HS-unit. In addition, HS-unit notifies the audio gateway (HSG) about local changes of microphone or speaker gain. When an ACL is established, HS is sending the current gain settings to the HSG (synchronisation). For HSP the feature is always enabled.

From HS, a volume change notification is sent to HSG whenever the analogue gains are changed at HS, i.e. on AT+GO/I… or assignments of S-Registers 589,590,689,690. An incoming microphone/speaker volume control message (+VGM:<gain> / +VGS:<gain>) immediately affects analogue input or output gain of HS.

From HSG a remote volume change notification can be sent by AT+HSGG… command family. It is possible to change the remote gains with an absolute value or incrementally. Incoming notifications about HS gain changes are displayed by UART message and are also cached internally at HSG. The latter allows to query current remote gain settings plus incremental change of the values (up/down).

2.6.8 HFP (Hands-free Profile)

The Hands-free profile (HFP) defines how two devices supporting HFP shall interact with each other on a point-to-point basis. The use case for HFP is a hands-free unit that is connected wirelessly to an audio gateway. The audio gateway is typically a cellular phone. The hands-free unit acts as audio input and audio output of the cellular phone and allows control of typical telephony functions without access to the phone.

We recommend the BTM5xx module be controlled by a host processor using AT commands (hosted operation mode). BTM5xx HFP implementation supports both Hands-free role and Audio gateway role.

An AT command beginning with AT+HF…indicates affiliation to Hands-free role of HFP.

An AT command beginning with AT+HFG…indicates affiliation to Audio Gateway role of

HFP.

As of f/w version 22.2.5.0, it is not possible any longer to initiate or accept a connection from or to HFP if a link to the 'sister' profile instance (HSP) already exists.

Example: if a link to a local HS profile instance exists, no more links can be initiated/accepted for the local HF profile. Or, if a link exists to a local HFG instance, no more links can be made from/to the local HSG instance and vice versa.

This is caused by the new underlying version of BT stack and library. It has not been evaluated, whether the feature of earlier f/w version can be restored.

Version 1.6 of the hands-free profile are supported. Refer to the HFP 1.6 specification for more information. [5].

Table 2-56 shows the feature requirements for this profile and the level of support on BTM5xx. We recommend you download the profile specification [5] in order to understand the procedures related to each AT command. For quick navigation, references to [5] are given in this section, which follow the syntax:

#<Feature>_<Section>

With:

<Feature> = HFP feature no. in Table 2-56 and Table 3.1 of [5]

<Section> = Appropriate Section in [5]

Example:

#3_4.12
feature no. = 3
section = 4.12

Table 2-56: Hands-free Profile supported features on BTM5xx

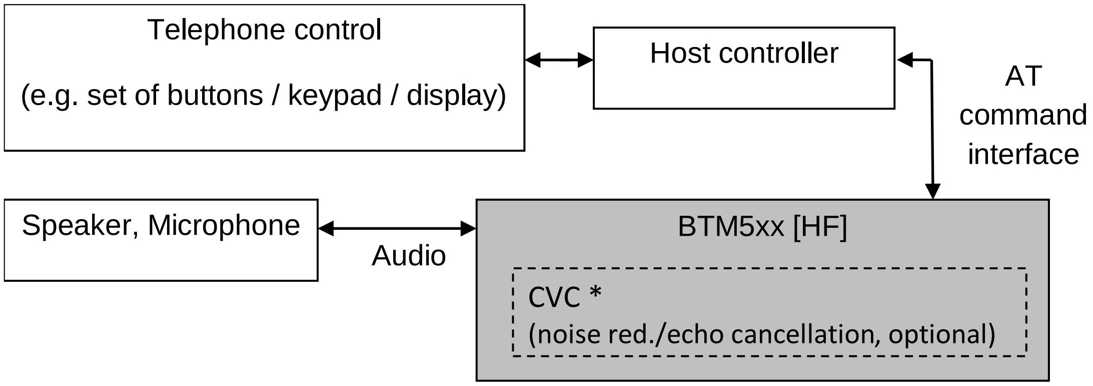
HFP Feature		Support in HF		Support in AG	
		Specification	BTM5xx	Specification	BTM5xx
1.	Connection management	M	Yes	M	Yes
2.	Phone status information	M	Yes	M	Yes
3.	Audio Connection handling	M	Yes	M	Yes
4	Accept an incoming voice call	M	Yes	M	Yes
5.	Reject an incoming voice call	M	Yes	O	Yes
6.	Terminate a call	M	Yes	M	Yes
7.	Audio Connection transfer during an ongoing call	M	Yes	M	Yes
8.	Place a call with a phone number supplied by the HF	O	Yes	M	Yes
9.	Place a call using memory dialing	O	Yes	M	Yes
10.	Place a call to the last number dialed	O	Yes	M	Yes
11.	Call waiting notification	O	Yes	M	Yes
12.	Three way calling	O	(2)	O	(2)
13.	Calling Line Identification (CLI)	O	Yes	M	Yes
14.	Echo cancellation (EC) and noise reduction (NR)	O	(5)	O	(5)
15.	Voice recognition activation	O	Yes	O	Yes

HFP Feature		Support in HF		Support in AG	
		Specification	BTM5xx	Specification	BTM5xx
16.	Attach a Phone number to a voice tag	O	(5)	O	(5)
17.	Ability to transmit DTMF codes	O	(5)	M	(4)
18.	Remote audio volume control	O	Yes	O	Yes
19.	Respond and Hold	O	(2)	O	(2)
20.	Subscriber Number Information	O	Yes	M	Yes
21a.	Enhanced Call Status	O	(2)	M	Yes
21b.	Enhanced Call Controls	O	(2)	O	(2)
22	Individual Indicator Activation	O	Yes	M	Yes
23	Wideband Speech	O	Yes	O	Yes
24	Codec Negotiation	O	Yes	O	Yes

M – Mandatory O – Optional
(1) support planned for future AT firmware release
(2) support planned for future AT firmware release, but with low priority
(3) not planned to be supported for future AT firmware
(4) service level signalization only
(5) planned for future AT firmware release

2.6.8.1 Hands-free unit role (HF)

Hands-free role is activated by setting flag 0x10 in S102 plus AT&W plus atz. Figure 2-26 is a block diagram of a BTM5xx in hosted operation mode.



*license fee applies for CVC algorithms (CSR)

Figure 2-26: Hands-free unit block diagram

2.6.8.1.1 Initiate service level connection (SLC) from HF (#1_4.2)

AT+HFD<bd_addr$_{hex}$>

Initiates service level connection (SLC) from local hands-free instance to remote device with <bd_addr$_{hex}$>. The remote device must support the audio gateway role (AG) of the Hands-free profile (HFP).

Response:

If SLC established	<cr,lf>CONNECT 123456789012,111F,><cr,lf>
If failed	<cr,lf>NO CARRIER<cr,lf> Or: <cr,lf>ERROR 59<cr,lf> Or: <cr,lf>ERROR 63<cr,lf>

After a SLC connection has been established, the module remains in AT command mode. S Register 531 is ignored for HFP connections.

2.6.8.1.2 Initiate audio connection from HF (#3_4.11)

AT+HFA

Initiates audio connection from local Hands-free instance. An existing service level connection is required.

Response:

Audio on	<cr,lf>HF"AU1"<cr,lf>

2.6.8.1.3 Release audio connection from HF (#3_4.12)

AT+HFR

Releases audio connection only. The service level connection is retained.

Response:

Audio on	<cr,lf>HF"AU0"<cr,lf>

2.6.8.1.4 Release entire connection from HF (#1_4.3)

AT+HFH

Releases connection from local Hands-free instance. Audio connection is released if existing. SLC is released anyway.

Response:

On audio disconnection	<cr,lf>HF"AU0"<cr,lf> (only if audio connection exists)
On SLC disconnection	<cr,lf>NO CARRIER 111E<cr,lf>

2.6.8.1.5 Answer incoming call from HF (#4_4.13)

AT+HFCA

Answers an incoming call. ATA is sent to the AG. In return, the audio gateway updates its +CIEV indicators (call=1 and callsetup=0) and sends appropriate messages to the HF. Upon receipt of a +CIEV message, HF notifies its host by a HFI<indicator>,<value> message. Refer to DTMF tone request (#17_4.27).

2.6.8.1.6 Reject incoming call from HF / Terminate call from HF (#5_4.14;#6_4.15)

AT+HFCH

Rejects incoming call / terminate ongoing call. AT+CHUP is sent to the AG. In return, the audio gateway updates its indicator call=0 and sends the appropriate +CIEV message to the HF. HF notifies its host by a HFI<indicator>,<value> message. Refer to Refer to DTMF tone request (#17_4.27).

2.6.8.1.7 Place call with number provided by HF (#8_4.18)

AT+HFC"nnn"

Initiates a new call from HF to the number specified by the number string **nnn**. This command sends ATDnnn to the audio gateway.

2.6.8.1.8 Memory dialling from the HF (#9_4.19)

AT+HFCM"mmm"

Initiates a new call from HF to the number stored in memory location **mmm** of the AG. This command sends ATD>mmm to the audio gateway.

2.6.8.1.9 Re-Dial last number from HF (#10_4.20)

AT+HFCL

Initiates a new call from HF to the number last dialled by AG. AT+BLDN is sent to the AG. In return, the audio gateway updates its indicator callsetup=2 and sends the appropriate +CIEV message to the HF. HF notifies its host by a HFI<indicator>,<value> message. Refer to DTMF tone request (#17_4.27).

2.6.8.1.10 Enable Caller Line Identification Presentation – CLIP (#13_4.23)

The CLIP feature is enabled by setting Bit 2 (value=4) in the HF supported features S register 581. Subsequent AT&W and ATZ is required to take effect. If this bit is set at boot time, the following actions are carried out automatically:

1. The appropriate flag is set in the HF service record.
2. The appropriate flag is set in the BRSF-message on Service Level Connection establishment
3. The SLC message AT+CLIP=1 is sent automatically once the SLC is established

Response to action no. 3 is indicated by either HF CLIP,OK or HF CLIP,ERROR after the CONNECT… message.

2.6.8.1.11 Enable Call Waiting Notification – CCWA (#11_4.21)

The CCWA feature is enabled by setting Bit 1 (value=2) in S581, HF supported features. Subsequent AT&W and ATZ is required to take effect. If this bit is set at boot time, the following actions are carried out automatically:

1. The appropriate flag is set in the HF service record.
2. The appropriate flag is set in the BRSF-message on Service Level Connection establishment.
3. The SLC message AT+CCWA=1 is sent automatically once the SLC is established.

Response to action no. 3 is indicated by either HF CCWA,OK or HF CCWA,ERROR after the CONNECT… message.

2.6.8.1.12 Enable/Disable/Query/Toggle Voice Recognition Activation (#15_4.25)

AT+HFV<x>   with <x>=1/0/?/T

2.6.8.1.13 Query subscriber number from HFG (#20_4.30)

AT+HFS?

Query subscriber numbers from HFG. AT+CNUM is sent to the HFG. HFG sends the subscriber number information, indicated on the HF by HF+CNUM,,,. On termination, OK is sent by HFG, which is indicated by HF CNUM,OK. If the HFG does not support this feature, it should send ERROR, indicated as HF CNUM,ERROR. Refer to Hands-free asynchronous messages.

Response:

Immediately, BTM accepting command	<cr><lf>OK<cr><lf>
Plus one of the following: <cr><lf>HF”+CNUM:<number>,<type>,<service>“<cr><lf> (refer to Hands-free asynchronous messages.) <cr><lf>HF"CNUM,OK"<cr><lf>	

-OR-

<cr><lf>HFX”<SLC_message>”<cr><lf> (original SLC response from HF gateway, but response format not compliant to HFP specification)
<cr><lf>HF"CNUM,OK"<cr><lf>

-OR-

<cr><lf>HF”CNUM,ERROR” <cr><lf> (feature not supported by HF-gateway)

-OR-

Nothing, if the HF-gateway ignores this query for any reason.

2.6.8.1.14 Query Operator Selection (#2_4.8)

AT+HFO?

Query operator selection from HFG. “AT+COPS?” is sent to the HFG. HFG sends the operator selection information, indicated on the HF by HF”+COPS,,,””. An “OK” is sent by HFG on termination, indicated on HF by the asynchronous message HF”COPS,OK”. If the HFG does not support this feature, it should send ERROR, indicated on HF as HF”COPS,ERROR”. Refer to Hands-free asynchronous messages.)

Response:

Immediately, BTM accepting command	<cr><lf>OK<cr><lf>

Plus one of the following:

<cr><lf>HF”+COPS:<mode>,<operator_size>,<operator>”<cr><lf> (refer to Hands-free asynchronous messages.)
<cr><lf>HF"COPS,OK"<cr><lf>

-OR-

<cr><lf>HF"COPS,ERROR"<cr><lf>

2.6.8.1.15 Enable Extended Error Codes in HFG – CMEE (#2_4.9)

AT+HFE1

Enable extended error result codes in HFG. “AT+CMEE=1” is sent to the HFG. HFG responds with “OK”, indicated at the HF by HF”CMEE,OK” or “ERROR” indicated at the HF by HF”CMEE,ERROR”

Response:

Immediately, BTM accepting command	<cr><lf>OK<cr><lf>

Plus one of the following:

<cr><lf>HF"CMEE,OK"<cr><lf> (confirmation from HFG)

-OR-

<cr><lf>HF"CMEE,ERROR"<cr><lf> (error, e.g. not supported by HFG)

2.6.8.1.16 DTMF tone request (#17_4.27)

AT+HFM<x> with <x>=0…9,*,#,A...D

Request a DTMF tone to be created by a connected HFG (e.g. mobile phone). The command AT+VTS=<x> is sent on the SLC. Refer to HFP 1.5 specification, feature 17 / section 4.27. This command has been tested with an iPhone (COD: S515=$200404) and a Samsung E250.

2.6.8.1.17 Enable/disable call waiting notification

AT+HFN<x> <x>=1,0

Causes the HF unit to send AT+CCWA=1 or AT+CCWA=0 across to HFG over SLC.

2.6.8.1.18 Query support for Voice Recognition Activation

AT+HFFV?

Query support for Voice Recognition Activation on local (HF) and remote (HFG) end (based on HFP-supported features.

2.6.8.1.19 Enable/disable/query/toggle Voice Recognition Activation (#15_4.25)

AT+HFV<x> <x>=1,0,?,T

Enable/disable/query/toggle voice recognition activation.

Voice recognition activation (VRA) is a HFP feature which is intended to be used with applications such as "Siri" (iPhone) or other voice recognition applications. Once an SLC connection is established, from a HF unit, the easiest command to start this is the toggle function: "AT+HFVT". This switches VRA on if it is off and vice versa. The toggle function can also be controlled by a push button (GPIO), FMC=0x12.

2.6.8.1.20 As a precondition, both ends must support VRA feature. "AT+HFFV?"

Allows to double check this: If the response is "1,1", then VRA should be working.Enable/Disable/Query for Wideband Speech status

AT+HFW<x> <x>=1, 0, ?

As a precondition for this command, WBS/codec negotiation (bit 7) must be enabled in HF-supported features (S581) at boot time.

2.6.8.1.21 Disable noise reduction/echo cancellation in HFG (#14_4.24)

AT+HFZ

Disable NR/EC in HFG (AT+NREC=0 sent on SLC, Feature #14_4.24 of HFP1.5 spec). This command is only allowed if both HF and HFG have declared NREC support in their HF/HFG supported features (see S581 [HF] and S596 [HFG]). Otherwise, ERROR 89 is returned.

AT+HFF? and AT+HFFN? allow to query currently supported HFP features.

2.6.8.1.22 Query supported features, locally and remote

AT+HFF?

Query local (HF) and remote (HFG) supported features (SF). HFP SLC is required.

Response: <HF_SF>,<HFG_SF>

Refer to S581 for a definition of local (HF) supported features.
Refer to S596 for a definition remote (HFG) supported features.

2.6.8.1.23 Query NREC flag, locally and remote

AT+HFFN?

Query NREC (noise reduction, echo cancellation) flag of local (HF) and remote (HFG) supported features. HFP SLC is required.

Response: <HF_NREC>,<HFG_NREC>

The NREC flag is Bit 0 in the HF supported features (S581) and Bit 1 in HFG supported features (S596).

2.6.8.1.24 Indicators from HFG +CIEV (#2)

HFI"<indicator_string>",<val> (S333=1, default)

HFI<indicator_id>,<val> (S333=0)

Inform the HF-host about a "+CIEV" indicator message received from the connected audio gateway. S-Register 333 enables verbose mode for HFI indicators. Possible indicators are listed in Table 2-30.

Table 2-57: +CIEV indicators in HFI message

+CIEV Audio Gateway Indicator	HFI indicator_string (S333=1)	HFI indicator_id (S333=0)	Section in HFP spec.
Service	service	1	4.4
Call	call	2	All call related sections
Call Setup	callsetup	3	
Call held	callheld	4	
Signal	signal	5	4.5
Roam	roam	6	4.6
Battery Charge	battchg	7	4.7

2.6.8.1.25 Hands-free audio routing

Audio in the Hands-free profile is always referred to as a SCO connection which is used for bidirectional transmission of speech.

On a BTM5xx, SCO audio is routed to the internal codec by default. S315 can select the I2S or PCM interface in master or slave mode for SCO connections. Refer to Routing audio streams over I2S.

2.6.8.1.26 Send custom data over SLC (HF)

AT+HFX"<data>"

Send custom data <data> over SLC from HF instance. Please refer to section 2.9.24 for details on utilizing ACL for custom data.

2.6.8.1.27 Hands-free status

ATI65

Returns the status of the Hands-free (HFP) instance:

0	Not connected
1	SLC connected
2	Audio connected
3	In call, SLC connected
4	In call, audio connected
5	In call setup (incoming/dialling/alerting), SLC connected
6	In call setup (incoming/dialling/alerting), audio connected

2.6.8.1.28 Hands-free asynchronous messages

CONNECT <bd_addr$_{hex}$>,<uuid$_{hex}$>[,<dir>]	A Service level connection to headset has been established and initialized.

<bd_addr$_{hex}$>	Bluetooth address of headset device.
<uuid$_{hex}$>	111E – If it is an incoming connection. 111F – If it is an outgoing connection.
<dir>	<"/">/I/O – Optionally indicates the direction (incoming/outgoing).

Refer to S331 and UUIDs in CONNECT/NO CARRIER Messages.

HF"RING"	HF has received a RING indication from the connected audio gateway. HF is expected to respond with ATA (answer, see AT+HFCA) or AT+CHUP (see AT+HFCH).
HF"ERROR"	HF has received ERROR from the connected audio gateway. This can be due to a request for memory dialling with invalid memory location (AT+HFC>mmm) or a request to redial the last number (AT+HFDL). There is no last number available in the AG if this error appears.
FS8000,INT	The internal codec is configured for a sampling frequency of 8000 Hz (narrowband/CVSD).
FS16000,INT	The internal codec is configured for a sampling frequency of 16000 Hz (WBS).
HF"AU1"	Audio connection (SCO) is established – Audio On.
HF"AU0"	Audio connection (SCO) is released – Audio Off.
HF"CLIP,OK"	HFG replied with OK to reception of AT+CLIP=1 or AT+CLIP=0.
HF"CLIP,ERROR"	HFG replied with ERROR to reception of AT+CLIP<…>.
HF"+CLIP,n,m,1234567"	A Calling Line Identification Notification (+CLIP:1234567,<type>) was received from the HFG. This message is sent by the HFG on incoming calls together with HF RING. The HFG and telephone network must support this feature and the according flag (Bit 2, value = 4) in the HF supported features. S register 581 must be set at boot time.

<table>
<tr><td></td><td>
<table>
<tr><td>n</td><td>Type of number (derived from the <type> in SLC messages, not the original value)</td></tr>
<tr><td>0</td><td>Unknown</td></tr>
<tr><td>1</td><td>International number</td></tr>
<tr><td>2</td><td>National number</td></tr>
<tr><td>3</td><td>Network-specific number</td></tr>
<tr><td>4</td><td>Dedicated access, short code</td></tr>
<tr><td>m</td><td>Length of caller number (number of characters)</td></tr>
<tr><td>1234567</td><td>Caller number</td></tr>
</table>
</td></tr>
<tr><td><b>HF”CCWA,OK”</b></td><td>HFG replied with OK to reception of AT+CCWA=1 or AT+CCWA=0.</td></tr>
<tr><td><b>HF”CCWA,ERROR”</b></td><td>HFG replied with ERROR to reception of AT+CCWA&lt;…&gt;.</td></tr>
<tr><td><b>HF”+CNUM:1234567,ttt,s”</b></td><td>Response to an AT+CNUM request received from HFG:
<table>
<tr><td>1234567</td><td>Phone number in the format specified by <b>ttt</b>.</td></tr>
<tr><td>ttt</td><td>The format of the phone number provided. Can be one of the following values:
▪ <b>Values 128-143</b>: The phone number format may be a national or international format and may contain a prefix and/or escape digits. No change to the number presentation is required.
▪ <b>Values 144-159</b>: The phone number format is an international number and includes a country code prefix. The plus sign (+) is not included as part of the number and is added by the HFG as needed.
▪ <b>Values 160-175</b>: The phone number is a national number. No prefix or escape digits are included.</td></tr>
<tr><td>s</td><td>Indicates the service to which this phone number relates. Is either 4 (voice) or 5 (fax).</td></tr>
</table>
</td></tr>
<tr><td><b>HF”+BSIR:0”</b></td><td>HFG has disabled in-band ring tone capability (+BSIR:0 was received on SLC)</td></tr>
<tr><td><b>HF”+BSIR:1”</b></td><td>HFG has enabled in-band ring tone capability (+BSIR:1 was received on SLC)</td></tr>
<tr><td><b>HF”CNUM,OK”</b></td><td>HFG has replied with OK to reception of AT+CNUM (subscriber number information query).</td></tr>
<tr><td><b>HF”CNUM,ERROR”</b></td><td>HFG has replied with ERROR to reception of AT+CNUM (subscriber number information query). This may be because this feature is not supported by the HFG.z.</td></tr>
</table>

HF”+COPS:m,ss,Operator”	Response to an AT+COPS? request received from HFG: m Contains the current mode and contains no information regarding the name of the operator. ss Number of characters in the operator name. Operator The name of the network operator in alphanumeric format.
HF”COPS,OK”	HFG replied with OK to reception of AT+COPS?.
HF”COPS,ERROR”	HFG replied with ERROR to reception of AT+COPS.
HF”CMEE,OK”	HFG replied with OK to reception of AT+CMEE=1.
HF”CMEE,ERROR”	HFG replied with ERROR to reception of AT+CMEE=1.
HFI<…>	Indicator (+CIEV…) from audio gateway was received. Refer to UUIDs in CONNECT/NO CARRIER Messages.
HFRX”<data>”	Spy message displaying incoming data on the SLC (rfcomm) of the HFP link.
HFUC”<data>”	Message displaying unrecognized command data from the SLC (rfcomm) of the HFP link.
NO CARRIER 111E	Service level connection to local HF-instance has been released. See UUIDs in CONNECT/NO CARRIER Messages as well.

2.6.8.1.29 Hands-free summary

Table 2-58: Hands-free role (HFP) – Summary of S Registers and AT Commands

Task	AT-Command /S Register	Comment
Enable HFP-HF role	S102	0x10 = HF role of HFP (bitmask), needs subsequent AT&W and ATZ to activate
Initiate SLC from HF	AT+HFD<bd_addr$_{hex}$>	**Responses:** If successful CONNECT 123456789012,111F,> If failed NO CARRIER If profile disabled ERROR 59 If incorrect state ERROR 63
Initiate audio connection from HF	AT+HFA	**Responses:** HF”AU1”
Release audio only from HF / Transfer Audio to AG	AT+HFR	Responses: HF”AU0”
Initiate call with number provided by HF	AT+HFC”nnn”	nnn = number string

Task	AT-Command /S Register	Comment
Initiate call with number provided by memory of AG	AT+HFCM"mmm"	mmm = memory location in AG of number requested to dial.
Initiate call to last dialled number	AT+HFCL	**Response:** HFI callsetup,2 – confirmation, call setup in progress -OR- HFG ERROR - last dialled number not available in AG.
Query supported features, locally and remotely	AT+HFF?	Returns supported features for local HF and connected remote HFG
Query NREC flag, locally and remotely	AT+HFFN?	Returns status of NREC flag for local HF and connected remote HFG
Disable noise reduction/echo cancellation of HFG	AT+HFZ	Only works when NREC enabled in both local HF and connected remote HFG, otherwise ERROR 89.
Request DTMF tone from HFG	AT+HFM<x>	with <x>=0…9,*,#,A...D Request a DTMF tone to be created by a connected HFG.
Disconnect SLC from HF (and audio if exists)	AT+HFH / ATH111E / ATH*	See Disconnecting Profiles and UUIDs in CONNECT/NO CARRIER Messages.
Send custom data on HF-instance SLC	AT+HFX<data>\r	Maximum data length – 26 characters <data> = a string of printable ASCII characters or the two-digit hexadecimal presentation of a non-printable character, preceded by ****. \r must be used to terminate a command or block of data. This feature can be used to transmit short data sequences or commands but is not suitable for transmitting large amounts of data. Refer to Figure xx
Enable verbose indicators	S333	0 – display indicator ID only in HFI.. asynchronous message (refer to DTMF tone request (#17_4.27)). 1 – display indicator string in HFI… asynchronous message (refer to DTMF tone request (#17_4.27)).
Enable WBS/16 kHz, also applies to incoming SPP-SCO requests	S376	0 Disable WBS (8kHz) 1 Enable WBS (16kHz, default)

Task	AT-Command /S Register	Comment
HFP indicators enable mask; Individual indicator activation; See also HFP1.6 spec, search for "AT+BIA" command	S381 [0..0x7f]	Bitmask: 0x01 Service indicator 0x02 Signal strength indicator 0x04 Roaming status indicator 0x08 Battery charge indicator 0x10 Call indicator 0x20 Call setup indicator 0x40 Call held indicator
Set HF supported features	S581	Bitmask: Bit 0 Echo cancellation and/or noise reduction Bit 1 Call waiting notification capability and three-way calling Bit 2 CLIP presentation capability Bit 3 Voice recognition activation Bit 4 Remote volume control Bit 5 Enhanced call status Bit 6 Enhanced call control (currently not supported) Bit 7 Codec negotiation **Note:** Bit 0 through Bit 4 are advertised in the Service Record of the HF. Only CLIP presentation capability (Bit 2 = 0x04) is currently supported in the BTM module. After setting a new value to this S register, the commands AT&W and ATZ are required to commit the value.
Select SCO audio interface (BTM5xx only)	S315	SCO audio interface select: 0 Internal codec (default) 1 I2S master 2 I2S slave 3 PCM master (PCM_M) 4 PCM slave (PCM_S) Refer to Routing audio streams over I2S.

Task	AT-Command /S Register	Comment
Inquire HF status	ATI65	0 Not connected 1 SLC connected 2 Audio connected 3 In call – SLC 4 In call – audio 5 In call setup (incoming/dialling/alerting) – SLC 6 In call setup (incoming/dialling/alerting) - audio

2.6.8.2 Audio gateway role (AG-HFP / HFG)

Audio gateway role is activated by setting flag 0x40 in S102 plus AT&W plus atz. Refer to Figure 2-25: Audio Gateway block diagram for a block diagram of an audio gateway with a BTM5xx in hosted operation mode.

Table 2-56 shows the feature requirements for this profile and the support level on BTM5xx. Laird highly recommends you download the profile specification [5] order to understand the procedures related to specific AT commands. For quick navigation, references to [5] are given in this section with the syntax:

#<Feature>_<Section>

With:

<Feature> = HFP feature no. in Table 2-56 and Table 3.1 of [5]
<Section> = Appropriate Section in [5]

Example:

#3_4.12
feature no. = 3 , section = 4.12
Many of the HFG commands refer to call states. These can be queried by AT+HFGC?. A description can be found in HFG audio routing.

2.6.8.2.1 Initiate service level connection (SLC) from AG (#1_4.2)

AT+HFGD<bd_addr$_{hex}$>

Initiate service level connection from local audio gateway instance to device with **<bd_addr$_{hex}$>. The remote device must support the Hands-free role (HF) of the Hands-free profile (HFP).**

Response:

SLC established:	<cr,lf>CONNECT 123456789012,111E,><cr,lf>
Failed:	<cr,lf>NO CARRIER<cr,lf>
Or:	<cr,lf>ERROR 59<cr,lf>
Or:	<cr,lf>ERROR 63<cr,lf>

After an SLC connection is established, the module remains in AT command mode. S Register 531 is ignored for HFP connections.

2.6.8.2.2 Initiate audio connection from AG (#3_4.11)

AT+HFGA

Initiate audio connection from local audio gateway instance. An existing service level connection is required. This command is not permitted to be issued in a call setup state (incoming / dialling / alerting, ATI66 = 5 or 6) and returns with ERROR 63 in that case.

Response:

Audio on: <cr,lf>HFG"AU1"<cr,lf>

2.6.8.2.3 Release audio connection from AG (#3_4.12)

AT+HFGR

Release audio connection only. The service level connection is retained. This command is not permitted to be issued in a call setup state (incoming / dialling / alerting, ATI66 = 5 or 6) and returns with ERROR 63 in that case.

Response:

Audio off: <cr,lf>HFG"AU0"<cr,lf>

2.6.8.2.4 Release entire connection from AG (#1_4.3)

AT+HFGH

Release connection from local audio gateway instance. An audio connection is released if existing, SLC is released anyway. This command is not permitted during a call setup state (incoming/dialling/alerting, ATI66 = 5 or 6) and returns with ERROR 63 in that case.

Response:

On audio disconnection: <cr,lf>HFG"AU0"<cr,lf> (only if audio connected)
On SLC disconnection: <cr,lf>NO CARRIER 111F<cr,lf>
Refer to UUIDs in CONNECT/NO CARRIER Messages.

2.6.8.2.5 Signal incoming call from AG to HF (#4_4.13)

AT+HFGC<number_string>,<type$_{dec}$>

Signal an incoming call by sending sending RING and +CLIP:<number_string>,<type$_{dec}$> to HF periodically. The <number_string> field represents the phone number of the ringing party. The type field specifies the format of the phone number, and can be one of the following values:

- Values 128-143: The phone number format may be a national or international format, and may contain prefix and/or escape digits. No changes on the number presentation are required.
- Values 144-159: The phone number format is an international number, including the country code prefix. If the plus sign (+) is not included as part of the number and shall be added by the AG as needed.
- Values 160-175: National number. No prefix nor escape digits included.

Refer to the Hands-free Profile Specification [5].

The HF is now expected to answer or to reject the call. Optionally, an incoming call can be answered or rejected at the HFG side (AT+HFGCA / AT+HFGCH).

2.6.8.2.6 Signal "call answered at AG" to HF (#4_4.13.3)

AT+HFGCA

AT+HFGCA (call answer) intelligently performs actions based on one of the following situations.

If no call is active (check with AT+HFGC?, refer to HFG audio routing): Signal from HFG to HF that an incoming call has been answered at the HFG. On the SLC the indicators +CIEV:2,1 (call=1) and +CIEV:3,0 (callsetup=0) are sent to the HF subsequently. The status of the incoming call is changed to active.

If one call is currently active and another is incoming, AT+HFGCA places the active call on hold and sets the incoming call to active.

If more than one call is on hold, AT+HFGCA places the active call on hold and makes the other call active.

The states of all current calls can be checked by AT+HFGC?

If none of the above conditions is met, an ERROR 063 is returned.

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

2.6.8.2.7 Signal outgoing call status from AG to HF (#8_4.18; #9_4.19; #10_4.20)

AT+HFGC1["<number_string>",<type$_{dec}$>]

This command shall signal a successful response to a HF initiated call, indicated by one of the following asynchronous messages:

- HFG"Dnnn" (dial number <nnn> given by HF-unit)
- HFG"D>nnn" (dial from AG memory location nnn)
- HFG"L" (redial last number)

"OK" is sent as successful response to the HF's request.

Signal an outgoing call status by sending "+CIEV:3,2" (callsetup=2) indicator to HF. If an audio connection is not present yet, it is initiated.

A call record with "status = dialling" is created and can be checked by AT+HFGC?.

The optional parameters <number_string> and <type> is stored in the call record. Their format is described in Signal incoming call from AG to HF (#4_4.13).

For HFG"Dnnn", AT+HFGC1 is the only possible response the HFG can issue. If the HFG wants to cancel the HF's call request, it shall issue AT+HFGC1 first, followed by AT+HFCH which terminates the call process.

AT+HFGC2

Signal to HF that remote party is reached and is being alerted during an outgoing call setup procedure. "+CIEV:3,3" (callsetup=3) indicator is sent to HF. If the telephone network does not provide an indication of alerting the remote party, the HFG may not send this indication.

The status of the appropriate call record is changed from dialling to alerting.

AT+HFGC3

Signal to HF that remote party has answered the call. "+CIEV:2,1" (call=1) and "+CIEV:3,0" (callsetup=0) indicators is sent to HF subsequently.

The status of the appropriate call record is changed from dialling or alerting to active.

AT+HFGC4["<number_string>",<type$_{dec}$>]

This command is used if a call is initiated at the HFG and not at the HF.

OK is not sent to the HF (in contrast to HFGC1).

Signal an outgoing call status by sending "+CIEV:3,2" (callsetup=2) indicator to HF. If an audio connection does not exist, it is initiated.

A call record with status = dialling is created and can be checked by AT+HFGC?.

The optional parameters <number_string> and <type> are stored in the call record. Their format is described in Signal incoming call from AG to HF (#4_4.13).

2.6.8.2.8 Signal termination or rejection of a call from AG to HF (#6_4.15)

AT+HFGCH (HFG call hang-up)

Terminate or reject all calls in one of the following states: active, incoming, waiting, dialling, alerting. And signal termination of a call by sending +CIEV:2,0 (call=0) indicator to HF.

- If a call is active: the active call is terminated (hang up).
- Else, if a call is incoming (ringing): the incoming call is rejected
- Else, if an outgoing call is being set up (dialling/alerting): the outgoing call is terminated
- Else, if a call is held: the held call is released (hang up)

If none of the above conditions are met, an ERROR 063 is returned.

According to the call status of the terminated or rejected call, HFGT or HFGR is sent to the host.

AT+HFGCHL

Terminate all calls of status *held*. Terminate call signals by sending **+CIEV:4,0** (callheld=0) indicator to HF. The appropriate call record is deleted.

Verify by AT+HFGC?, S354=1. Manual tests show that this command works well on the HFG when no SLC exists. If this command is issued when an HF unit is connected (SLC or SLC and audio), it's not guaranteed that all signalling (HFP indicators) and automatic audio handling works correctly in all circumstances.

AT+HFGCH*

Hang up all calls, clearing all call records on HFG.Verify by AT+HFGC?, S354=1. Manual tests show that this command works well on the HFG when no SLC exists. If this command is issued when an HF unit is connected (SLC or SLC and audio), it's not guaranteed that all signalling (HFP indicators) and automatic audio handling works correctly in all circumstances.

2.6.8.2.9 Sending AG indicators to HF "+CIEV…, #2)

AT+HFGI<indicator_id$_{dec}$>,<value$_{dec}$>

Send an AG indicator to HF in the form +CIEV:<indicator_id>,<value>.

Table 2-59 lists possible indicators and their value range.

If there is no SLC existing and indicator "Service" is set (AT+HFGI1,n), the value is stored in volatile memory to be presented during a future SLC initialization procedure.

For all other indicators, this command is valid only if a SLC exists.

AT+HFGI should only be used with indicator IDs 1 (service), 5 (network signal strength), 6 (roam) and 7 (battery charge level).

For call management purposes, use the appropriate AT+HFGCx command. They send call-related indicators (call, call setup, call held) automatically.

Table 2-59: AG indicators for AT+HFGI.

Indicator Name	indicator_id	Range of Value	Section in HFP Spec. [5]
Service	1	0 - 1	4.4
Call	2	0 - 1	All call related sections
Call setup	3	0 - 3	
Call held	4	0 - 2	
Signal	5	0 - 5	4.5
Roam	6	0 - 1	4.6
Battery Charge	7	0 - 5	4.7

2.6.8.2.10 Send operator string to HF (+COPS…, #2_4.8)

AT+HFGO<operator_string>

Send network operator string to HF by +COPS:0,0,<operator_string> on SLC. This command responds to a AT+COPS? request, indicated by the asynchronous message HFGOP?.

2.6.8.2.11 Send ERROR or +CME ERROR… to HF

AT+HFGE<cme_code>

Send ERROR to HF if no <cme_code> is given. Required as optional response to HFG D>nnn an HFGL to confirm that the dial request was not successful (i.e. invalid memory location or last dialled number not available).

If the optional <cme_code> is appended and if extended AG error result codes were enabled by HF (stored internally), the appropriate extended error result code in the form of +CME ERROR: <cme_code> is sent to HF.

If <cme_code> is appended but extended error codes have not been enabled by HF, <cme_code> is ignored and only ERROR is sent to HF.

2.6.8.2.12 Change In-band ring tone setting when SLC exists (#4_4.13.4)

AT+HFGB0 / AT+HFGB1

As a prerequisite for this command, the in-band ring flag (0x08) must have been set in S596 (HFG supported features) at boot time and a service level connection to the HF shall be established.

Use AT+HFGB0 to indicate to the HF that the HFG will not provide in-band ring tones subsequently. +BSIR:0 is sent over the SLC.

Use AT+HFGB1 to indicate to the HF that the HFG will provide in-band ring tones subsequently. +BSIR:1 is sent over the SLC.

The BTM module does not create ring tones. The actual ringtone signal must be provided by an external circuit and injected to the analogue audio input port of the BTM module.

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

Or: <cr,lf>**ERROR 68**<cr,lf> If no SLC present

Or: <cr,lf>**ERROR 69**<cr,lf> If in-band ring flag was not set in the HFG supported features (S596) at boot time

2.6.8.2.13 Subscriber number records (+CNUM…, #20_4.30)

AT+HFGS=<number_string>,<type>,<service>

This command adds a subscriber number record to the HFG. The subscriber number is the telephone number of the HFG. It may have more than one subscriber numbers. All subscriber numbers available should be stored in the HFG immediately after power up of the system. When an SLC to a HF device is established, the HF can query the HFG's subscriber numbers at any time by sending AT+CNUM to the HFG on the SLC. The available subscriber number records then transfer to the HF automatically. Subscriber number records are stored in volatile memory and are lost on reset or power cycle. At maximum, 4 subscriber number records are allowed.

<number_string> = subscriber number string, characters allowed: 0123456789*#+ABDCTP!W@

<type> = specifies the format of <number_string> provided, can be one of the following values:

- 128-143: The phone number format may be a national or international format, and may contain prefix and/or escape digits. No changes on the number presentation are required.
- 144-159: The phone number format is an international number, including the country code prefix. If the plus sign (+) is not included, it shall be added by the AG as needed.
- 160-175: National number. No prefix nor escape digits included.

<service> = service of this subscriber number, permitted values:

4 – voice
5 – fax

Response:

```
<cr><lf>OK<cr><lf>                (command valid, record added)
OR:
<cr><lf>ERROR 05<cr><lf> (syntax or value range error)
OR:
<cr><lf>ERROR 67<cr><lf> (maximum number of subscriber number records reached)
```

AT+HFGS?

This command returns a list of currently available subscriber number records which are sent to the HF-unit via "AT+CNUM" (issued by HF on the SLC).

Response:

For each subscriber number record (if at least one is available):

```
<cr><lf>[<index>]"<number_string>",<type>,<service>
```

And finally:

```
<cr><lf>OK<cr><lf>                (command finished)
```

With:

<index> = current index of the record, required as parameter n for AT+HFGSD<n>
<number_string> = subscriber number string

<type> = specifies the format of <number_string>. Refer to AT+HFGS=… above.

<service> = service of this subscriber number:

4 – voice
5 – fax

AT+HFGSD

Delete all subscriber number records in the HFG.

Response:

```
<cr><lf>OK<cr><lf>
```

AT+HFGSD<n>

Delete subscriber number record with index <n> in the HFG. <n> refers to the index that is displayed on HFGS?. The value of <n> is assumed to be one digit. The index is not fixed to a record. If there are higher numbered indexes than the one deleted, the remaining records get a new index assigned dynamically on HFGS?.

Response:

```
<cr><lf>OK<cr><lf>
Or:
<cr><lf>ERROR 66<cr><lf> Record with requested index <n> not available.
Or:
<cr><lf>ERROR 05<cr><lf> Syntax error.
```

2.6.8.2.14 HFG – Call Waiting Notification (+CCWA…, #11_4.21, #12_4.22.1)

Call waiting notification is a mandatory feature for the HFG and is therefore supported on the BTM510/511. If call waiting notification was enabled by the HF (4.21), when a call is waiting the call waiting notification +CCWA… and the appropriate call setup indicator +CIEV: callsetup=1 are sent automatically as specified in section 4.22.1 of the HFP specification [5].

2.6.8.2.15 Enable/disable/query/toggle voice recognition (#15_4.25)

AT+HFGV<x>   <x>=1, 0, ?, T

Voice recognition activation (VRA) is a HFP feature which is intended to be used with applications such as "Siri" (iPhone) or other voice recognition applications. Once an SLC connection is established, from a HFG unit, the easiest command to start this is the toggle function: AT+HFGVT. This switches VRA on if it is off and vice versa. The toggle function can also be controlled by a push button (GPIO), FMC=0x13

2.6.8.2.16 Query/Set/Increment/Decrement remote gain of HF (#18_4.28)

AT+HFGG…

Syntax: AT+HFGG<I/O><?/n/U/D> with:

I/O	I – Input (microphone) O – Output (speaker)
?	Query current/cached remote gain value
<n>	Set remote gain value. Valid range: 0..15 (dec)
U	Up – Increments of one (1)
D	Down – Increments of one (1)
ERROR 129	Indicates that a cached gain value is not available. This can happen if HS does not support the remote volume control feature (missing notifications).

Examples:

AT+HFGGOU – increment speaker gain of connected HF

AT+HFGGI? – query current (cached) microphone gain of connected HF

AT+HFGGI13 – set microphone gain of connected HF to 13

2.6.8.2.17 Enable/disable/query wideband speech status

AT+HFGW<x>   <x>=1, 0, ?

As a precondition for this command, WBS/codec negotiation (bit 9) must be enabled in HFG-supported features (S596) at boot time. For AT+HFGW? (query WBS status), the status for the local HFG and the remote HF unit is returned. WBS is used on an HFP audio link only if enabled on either end (for example: A response to AT+HFGW? = 1,1).

2.6.8.2.18 Query HFP-supported features for local (HFG) and remote (HF) end.

AT+HFGF?

Response when HFG connected:

\r\n$<AAAA>,$<BBBB>\r\nOK\r\n

<AAAA>: HFG supported features (local), hexadecimal

<BBBBB>: HF supported features (remote), hexadecimal

Example response:

\r\n$026C,$009C\r\nOK\r\n

2.6.8.2.19 Query support for Voice Recognition Activation on local (HFG) and remote (HF) end

AT+HFGFV?

Response is based on HFP-supported features.

Response when HFG connected:

\r\n<A>,<B>\r\nOK\r\n

<A>: Voice recognition act. support on HFG (local): 0=not supported / 1=supported

<B>: Voice recognition act. support on HF (remote): 0=not supported / 1=supported

Example response (indicating support on either end):

\r\n1,1\r\nOK\r\n

2.6.8.2.20 Query support for Noise Reduction/Echo Cancellation on local (HFG) and remote (HF) end

AT+HFGFN?

Response is based on HFP-supported features.

Response when HFG connected:

\r\n<A>,<B>\r\nOK\r\n

<A>: NREC support on HFG (local): 0=not supported / 1=supported

<B>: NREC support on HF (remote): 0=not supported / 1=supported

Example response (indicating support on remote end only):

\r\n0,1\r\nOK\r\n

2.6.8.2.21 HFG audio routing

Audio in the Hands-free profile is always referred to as a SCO connection, used for bidirectional transmission of speech.

On a BTM5xx, SCO audio is routed to the internal codec by default. S315 can select the I2S or PCM interface in master or slave mode for SCO connections. Refer to Routing audio streams over I2S.

2.6.8.2.22 Send custom data over SLC (HFG)

AT+HFGX"<data>"

Send custom data <data> over SLC from HFG instance. Please refer to section 2.9.24 for details on utilizing ACL for custom data.

2.6.8.2.23 HFG status

ATI66

Returns the status of the HFG instance:

0	Not connected
1	SLC connected
2	Audio connected
3	In call, SLC connected
4	In call, audio connected
5	In call, audio connected
6	In call setup (incoming/dialling.alerting), SLC connected
8	In call, no SLC

2.6.8.2.24 HFG Call Records

The HFG maintains call records for all calls. Even when no HF is attached to the HFG (no SLC existing), the HFG needs to be aware of all ongoing calls in order to send the correct HFP indicators as soon as an HF connects.

Also, it is mandatory for the HFG to respond to a current call list request, issued by the HF (AT+CLCC, #21a_4.31.1). The HFG queries its internal list of current calls and replies automatically to AT+CLCC.

The list of current calls is displayed to the host by AT+HFGC?.   AT commands beginning with AT+HFGC… change the state of a call record.

AT+HFGC?

List all current calls.

S354 controls verbose mode, and is detailed below. The response for each call record, if it exists, is as follows:

<cr,lf>[<idx>],<dir>,<call_status>,<call_mode>,<fmultiparty>,<number_string>,<number_type>

With:

idx : unique index of call, 1 digit.

dir : 0 ('>') for outgoing call, 1 ('<') for incoming call.

call_status	0	Active
	1	Held
	2	Dialling (outgoing)
	3	Alerting (outgoing)
	4	Incoming (incoming and ringing)
	5	Waiting (incoming and waiting whilst another call is active)
call_mode	0	Voice
	1	Data
	2	Fax
fmultiparty	0	Call is point to point (p)
	1	Call is part of multiparty (m)
number_string	Calling or called line identification presentation. Refer to HFG audio routing.	
number_type	Number type for CLIP, refer to HFG audio routing.	

Response at the end of list:

<cr,lf>OK<cr,lf>

S354 – Enable/disable verbose presentation of call records

S register 354 controls verbose presentation of call records (AT+HFGC?)

Value 0 = non-verbose mode (default)

Value 1 = verbose mode

Verbose mode gives better understanding of call states and call records in a human readable style whereas non-verbose mode is the preferred mode for a host microcontroller.

2.6.8.2.25 Call status vs. HFG status

Table 2-60 maps the call states to HFG states. For more information about HFG status refer to section HFG audio routing.

Table 2-60: Mapping of call status and HFG status

Call status (AT+HFGC?)	HFG status, SLC (ATI66)	HFG status, no SLC (ATI66)
incoming (ringing)	5/6	8
dialling	5/6	8
alerting	5/6	8
active, waiting	5/6	8
alerting	3/4	8
held	3	8

2.6.8.2.26 Audio Gateway (HFP) - Asynchronous Messages

CONNECT <bd_addr$_{hex}$>,<uuid$_{hex}$>[,<dir>]

A Service level connection to headset is established and initialized.

<bd_addr$_{hex}$>	Headset device Bluetooth address
<uuid$_{hex}$>	111F – Incoming connection 111E – Outgoing connection
<dir>	</>/I/O

Refer to S331 and UUIDs in CONNECT/NO CARRIER Messages.

HFG"VGS<n>"

Speaker gain setting message was received from HF (+VGS:<n>) with n = gain [0..15]

Note: <n> now displays two digits, padding with a leading 0 if the n value is ≤ 9.

HFG"VGM<n>"

Microphone gain setting message was received from HF (+VGM:<n>) with n = gain [0..15]

Note: <n> now displays two digits, padding with a leading 0 if the n value is ≤ 9.FS8000,INT

The internal codec is configured for a sampling frequency of 8000 Hz (narrowband/CVSD).

FS16000,INT

The internal codec is configured for a sampling frequency of 16000 Hz (WBS).

HFG"AU1"

Audio connection (SCO) has been established (= audio on).

HFG"AU0"

Audio connection (SCO) has been released (= "audio off").

HFG"C"

A call has been established and is ongoing.

HFG"T"

A call has been terminated.

HFG"R"

Incoming call was rejected.

HFG"OP?"

Request for network operator string (AT+COPS?) received from HF. AG is expected to reply with +COPS:0,0,<operator_string> on SLC. Use AT+HFGO<operator_string>.

HFG"Dnnn"

Request from HF to place an outgoing call to phone number provided by HF with <nnn> = number string. The AG is expected to confirm the outgoing call with the command AT+HFGC1. If the call request is not accepted by the HFG, it must send AT+HFGC1 first and then AT+HFGCH.

HFG"D>nnn"

Request from HF to place an outgoing call using memory dialling with <nnn> = memory location. The AG is expected to either confirm the outgoing call with the command AT+HFGC1 or to respond with AT+HFGE if the memory location is invalid.

HFG"L"

Request from HF to place an outgoing call using the last number dialled. The AG is expected to either confirm the outgoing call with the command AT+HFGC1 or to respond with AT+HFGE if the last number dialled is unavailable.

HFG"Mn"

Request from HF to generate DTMF code <n> towards the telephony network.

HFG"NREC0"

Request from HF to disable noise reduction and echo cancellation. This message only appears if noise reduction/echo cancellation was enabled in S596 at boot time (Bit1). The HFG is supposed to disable noise reduction and echo cancellation and respond with an OK. If noise reduction / echo cancellation was not enabled at boot time, no message is displayed and ERROR is sent to the HF silently.

HFG"CLIP1"

Request to activate caller line identification notification in the AG.

HFG"CCWA1"

Request to activate call waiting notification in the AG.

HFG"CMEE1"

Request to activate extended AG error result codes in the form +CME ERROR: <err>.

HFGRX"<data>"

Spy message displaying incoming data on SLC (rfcomm) of the HFP link

HFGUC"<data>"

Message displaying unrecognized command data from SLC (rfcomm) of the HFP link.

NO CARRIER 111F

Service level connection to local HFG-instance has been released. See UUIDs in CONNECT/NO CARRIER Messages as well.

2.6.8.2.27 AG - HFP summary (HFG)

Table 2-61: AG-HFP (HFG) – Summary of S Registers and AT Commands

Task	AT-Command/S Register	Comment
Enable HFP-AG role	S102	0x40 = AG role of HFP (bitmask), needs subsequent AT&W and ATZ to activate
Initiate SLC from HFG	AT+HFGD<bd_addr$_{hex}$>	Responses: Successful:CONNECT 123456789012,111F,> failed: "NO CARRIER" profile disabled: ERROR 59 incorrect state: ERROR 63
Initiate audio connection from AG/Transfer Audio from AG to HF	AT+HFGA	**Response:** HFG"AU1" ERROR 63 is returned if this command is used during call setup (incoming/dialling/alerting).
Release audio only from AG / Transfer Audio from HF to AG	AT+HFGR	**Response:** HFG"AU0" ERROR 63 is returned if this command is used during call setup (incoming/dialling/alerting).
Disconnect SLC from HFG (and audio if exists)	AT+HFGH / ATH111F / ATH*	See sections Disconnecting Profiles and UUIDs in CONNECT/NO CARRIER Messages. ERROR 63 is returned if this command is used during call setup (incoming/dialling/alerting).
Signal incoming call to HF	AT+HFGC<number_string>,<type$_{dec}$>	Sends RING and +CLIP: <number_string>, <type> to HF <type> must be in range 128..175 Refer to Signal incoming call from AG to HF (#4_4.13).

Task	AT-Command/S Register	Comment
Signal incoming call answered at HFG to HF	AT+HFGCA	Intelligently answers a call based on the following situations: ▪ If a call is incoming (ringing): answer the incoming call and change status to 'active' ▪ Else, if a call is waiting (incoming call while another call is active): the active call is put on hold and the waiting call is answered (becoming 'active') ▪ Else, if a call is held: the held call is activated. If an active call exists it is put on hold (swap). If none of the above condition is met, then ERROR 063 is returned.
Signal outgoing call status to HF, update call record	AT+HFGC1[<number_string >,<type$_{dec}$>]	Outgoing call is being initiated by AG as result of a HF request. Use this AT command in response to one of the following asynchronous messages at HFG: HFG"Dnnn" or (1) HFG"D>mmm" or (2) HFG"L" (2) OK is sent to the HF. A call record in state *dialling* is created. <number_string> and <type> are optional and is be stored in the call record if given. <type> must be in range 128..175 Refer to Signal incoming call from AG to HF (#4_4.13)..
	AT+HFGC2	Outgoing call: Remote party has been reached and is being alerted; may not be used if the telephone network does not provide this information. Call record state is changed to *alerting*.
	AT+HFGC3	Outgoing call: Remote party has answered the call; state of call record is changed to active.

Task	AT-Command/S Register	Comment
	AT+HFGC4[<number_string>,<type$_{dec}$>]	Outgoing call: use if a call is initiated on the HFG and was not requested by HF. OK is not sent to the HF. A call record in state *dialling* is created. <number_string> and <type> are optional and is be stored in the call record if given. <type> must be in range 128..175 Refer to Signal incoming call from AG to HF (#4_4.13).
Signal termination of a call or a call setup procedure to HF	AT+HFGCH	Intelligently hangs up call based on the current situation: ▪ If a call is active: the active call is terminated (hang up). ▪ Else, if a call is incoming (ringing): the incoming call is rejected. ▪ Else, if an outgoing call is being set up (dialing/alerting): the outgoing call is terminated ▪ Else, if a call is held: the held call is released (hang up) If none of the above condition is met, then ERROR 063 is returned.
Release all held calls	AT+HFGCHL	Release was initiated either locally or remotely
Query list of current calls of HFG	AT+HFGC?	Returns the list of current calls
Enable/disable verbose mode for call record presentation	S354 [0..1]	Value: 0 =non-verbose mode for AT+HFGC? (default) 1 = verbose mode for AT+HFGC?
Send AG indicator status to HF (+CIEV…)	AT+HFGI<indicator_id$_{dec}$>, <value$_{dec}$>	For <indicator_id> and <value> see Table 2-59. Command is recommended only with indicator IDs 1(service), 5 (signal), 6 (roam) and 7(battery charge level). For all other indicators (2-call, 3-call setup, 4-call held), Laird recommends you use the appropriate call managing AT commands. These commands send the indicators automatically.
Send network operator string to HF	AT+HFGO<operator_string>	Command used to respond to HFGOP?
Send error to HF	AT+HFGE	Sends ERROR to HF. Needed on HFG"D>mmm" if memory location is invalid or on HFGL if last dialled number not available

Task	AT-Command/S Register	Comment
Send extended error result code to HF	AT+HFGE<cme_code$_{dec}$>	+CME ERROR: <cme_code$_{dec}$> is sent to HF if HF did enable this feature earlier. Otherwise a simple ERROR is sent to HF.
Disable / Enable In-Band ring tone	AT+HFGB0/1	+BSIR:0 or +BSIR:1 is sent to HF. SLC must exist and Bit 3 of S596 (HFG supported features) must have been set at boot time. Ringtone signal must be provided by external circuit.
Send custom data on HFG-instance SLC	AT+HFGX<data>\r	Maximum data length – 26 characters <data> = a string of printable ASCII characters or the two-digit hexadecimal presentation of a non-printable character, preceded by ****. \r must be used to terminate a command or block of data. This feature can be used to transmit short data sequences or commands but is not suitable for transmitting large amounts of data. Refer to Figure xx

<table>
<tr><th>Task</th><th>AT-Command/S Register</th><th>Comment</th></tr>
<tr><td>Set HFG supported features</td><td>S596</td><td>Bitmask:
<table>
<tr><td>Bit 0</td><td>Three-way calling</td></tr>
<tr><td>Bit 1</td><td>Echo cancellation and/or noise reduction function</td></tr>
<tr><td>Bit 2</td><td>Voice recognition function</td></tr>
<tr><td>Bit 3</td><td>In-band ring tone capability</td></tr>
<tr><td>Bit 4</td><td>Attach a number to voice tag</td></tr>
<tr><td>Bit 5</td><td>Ability to reject calls</td></tr>
<tr><td>Bit 6</td><td>Enhanced call status</td></tr>
<tr><td>Bit 7</td><td>Enhanced call control</td></tr>
<tr><td>Bit 8</td><td>Extended Error Result Codes</td></tr>
<tr><td>Bit 9</td><td>Codec negotiation</td></tr>
</table>
Note:
Bit 0…Bit 4 of these settings are advertised in the Service Record of the HFG.
The default value is 0x026C:
▪ Bit 2 – Voice recognition (0x04)
▪ Bit 3 – In-band ring (0x08)
▪ Bit 5 – Incoming call reject (0x20)
▪ Bit 6 – Enhanced call status (0x40)
▪ Bit 9 – Codec negotiation (0x200)
After a new value was set, the commands AT&W and ATZ are required for the new value to become effective.

Make sure that Bit 6 is always set – this is a mandatory feature for HFP1.5 – AG and HFP1.6 – AG.</td></tr>
<tr><td>Select SCO audio interface (BTM5xx only)</td><td>S315</td><td>SCO audio interface select:
<table>
<tr><td>0</td><td>Internal codec (default)</td></tr>
<tr><td>1</td><td>I2S master</td></tr>
<tr><td>2</td><td>I2S slave</td></tr>
<tr><td>3</td><td>PCM master (PCM_M)</td></tr>
<tr><td>4</td><td>PCM slave (PCM_S)</td></tr>
</table>
Refer toI2S – Inter Integrated Circuit Sound.</td></tr>
</table>

Task	AT-Command/S Register	Comment	
Inquire HFG status	ATI66	0	Not connected
		1	SLC connected
		2	Audio connected
		3	In call – SLC
		4	In call – audio
		5	Call setup: ringing (incoming call)/dialling or alerting (outgoing call) – SLC
		6	Call setup: ringing (incoming call)/dialling or alerting (outgoing call) – audio
		8	In call but no SLC
Enable WBS/16kHz, applies also to incoming SPP-SCO requests	S376 [0..1]	0	Disable WBS (8kHz)
		1	Enable WBS (16kHz, default)
Close audio on voice recognition stop	S378 [0..1]	0	Disable
		1	Enable (default)

(1) AT+HFGC1 is the only possible response to HFG"Dnnn". To reject this call attempt from the HF, the HFG must issue AT+HFGC1 first and subsequently AT+HFGCH.

(2) HFGE instead may respond to HFG"D>mmm" and HFGL to confirm that the requested operation was not successful (i.e. memory location invalid or last dialled number not available).

2.6.8.3 Wideband Speech

Wideband speech uses a 16 kHz audio sampling rate and a modified SBC codec (mSBC) to compress the voice data for bi-directional transmission over a limited bandwidth eSCO link. The higher sampling rate (compared to 8 kHz for legacy narrowband/CVSD) leads to a more natural sound of the voice.

If WBS is supported, both HFP parties must also support the Codec Negotiation feature. During the codec negotiation, WBS support availability is checked. If either end indicates that WBS is enabled, it is used for subsequent audio links.

WBS can be temporarily disabled by any of the following:

- AT+HFW0
- AT+HFGW0
- ATS376=0

Use one of the following to check the current status of WBS support:

- AT+HFW?
  -or-
- AT+HFGW? – AT+HFGW returns the status for both local (HFG) and remote (HF), separated by a comma)

By default, WBS is enabled. When entering an HFP audio link with FS16000, the sampling frequency of 16 kHz is a strong indication that WBS is in use.

The difference between narrowband and wideband speech can be demonstrated by the following setup/sequence:

assuming devA=HF and devB=HFG, default parameters, SLC established;

1. Create audio: AT+HFA – HF"AU1"; FS16000 (WBS)
2. Listen to the audio link.
3. Close audio: AT+HFR – HF"AU0"
4. Disable WBS: AT+HFW0
5. Re-create audio: AT+HFA – HF"AU1"; FS8000 (narrowband)
6. Listen to the audio link.
7. Close audio: AT+HFR – HF"AU0"
8. Re-enable WBS: AT+HFW1
9. Create audio: AT+HFA – HF"AU1"; FS16000 (WBS)

… and so on

Note: The above if for demonstration purposes only. In practice, WBS should not be disabled unless there are good reasons for doing so.

2.6.8.4 Remote volume control (HFP)

Remote volume control feature for HFP allows the audio gateway (HFG) to control microphone- and speaker gains of HF-unit. In addition, HF-unit notifies the audio gateway (HFG) about local changes of microphone or speaker gain. When an SLC is established, HF is sending the current gain settings to the HFG (synchronisation). For HFP, this feature is enabled by Bit 4 (0x10) of HF supported features (S581, enabled by default).

From HF, a volume change notification is sent to HFG whenever the analogue gains are changed at HF, i.e. on AT+GO/I… or assignments of S-Registers 589,590,689,690. An incoming microphone/speaker volume control message (+VGM:<gain> / +VGS:<gain>) immediately affects analogue input or output gain of HF.

From HFG a remote volume change notification can be sent by AT+HFGG… command family. It is possible to change the remote gains with an absolute value or incrementally. Incoming notifications about HF gain changes are displayed by UART message and are also cached internally at HFG. The latter allows to query current remote gain settings plus incremental change of the values (up/down).

2.6.9 MPS compatibility layer

MPS (multi profile specification) defines procedures which devices must follow if multiple profiles are supported simultaneously. For BTM51x this concerns A2DP, AVRCP and HFP. One typical use case is:

a stereo headset/hands-free device (A2DP-SNK, AVRCP-CT and HF-unit)

and another one (the opposite):

a stereo audio source / HF-gateway (A2DP-SRC, AVRCP-TG, HFG), typical use case of a phone, but also supported by BTM51x.

In a "MPSD" (multi-profile single device) scenario, all profiles are connected from the local device to one single remote device.

In “MPMD” (multi-profile multi device) scenarios, profile A (e.g. HFP) connects to remote device1, profile B (e.g. A2DP, AVRCP) connect to remote device2 and so forth.

MPS addresses use cases in which an active A2DP stream is temporarily suspended in order to serve a call process of HFP. This involves managing audio resources and stream control (A2DP stream vs. HFP SCO).

MPS compatibility is enabled by S386>0. Table 21 1 below compares MPS procedures with non-MPS procedures for various scenarios.

In earlier versions, when MPS was not supported yet, streaming-versus-call scenarios have been addressed by S355. This S-Register allows to assign priority to either A2DP or SCO (or to set them equal). A request of the lower priority entity for audio resources is rejected if at the same time the higher priority entity already owns the audio resource. If audio resource is requested by an equal priority or higher priority entity, the request is accepted, audio of the active entity is suspended (A2DP) or closed (SCO) and audio resource is granted to the requesting entity. The default setting (S355=1, SCO priority) effectuates that an A2DP stream is suspended if SCO audio is required, e.g. for a HFP call.

If MPS is disabled, then the existing S355 rules and related behaviour remains in place.

Table 2-62: MPS procedures

Scenario	MPS disabled (S386=0)	MPS enabled (S386>0)
A2DP signalling connect	BTM51x initiates start of streaming if permitted by S355, no matter if BTM51x is SRC or SNK	BTM51x initiates start of streaming only if the connection was initiated from local device and if configured as SRC (S300=2). Streaming is initiated by AT+APR from either end or by AVRCP “PLAY” command (provided ARCP is connected).
AVRCP-TG, reception of pass through command ‘PLAY’ (0x44)	Notify host controller via UART messages “AVPTI…”	Notify host controller via UART messages “AVPTI…” BTM51x A2DP SRC/HFG: initiate A2DP stream on reception of status “button released”
AVRCP-TG, reception of pass through command ‘PAUSE’ (0x46)	Notify host controller via UART messages “AVPTI…”	Notify host controller via UART messages “AVPTI…” BTM51x A2DP SRC/HFG: initiate suspension of A2DP stream on reception of status “button released”
Incoming call while streaming A2DP	If S355=1 or 3: suspend A2DP streaming, accept the call/SCO request, route the SCO link to audio interface; If S355=0 or 2: reject the SCO request, retain A2DP stream;	BTM51x A2DP SRC/HFG: initiate suspension of A2DP, route SCO link to audio interface as required for the call BTM51x A2DP SNK/HF: ignore

Scenario	MPS disabled (S386=0)	MPS enabled (S386>0)
Call finished (terminated or rejected)	No extra action is taken	BTM51x A2DP SRC/HFG: if A2DP stream was suspended due to the call, then re-start A2DP stream BTM51x A2DP SNK/HF: ignore
MPS service record	No change	A service record according to MPS specification is registered and can be accessed from remote devices when connectable (SDP).

S386 provides three options to enable MPS:

- 1: just enable MPS service record and set S355=1. The user is responsible to enable all required profiles (S102) and roles (S300, S301)
- 2: automatic configuration for A2DP-SNK / AVRCP-CT / HF
- 3: automatic configuration for A2DP-SRC / AVRCP-TG / HFG

Note: With S386=2 or =3 : S-Registers S102, S300 and S301 are modified at boot time according to Table 2-63 below. Unrelated flags of S102 are not touched (e.g. SPP[0x01] / HS[0x02] / HSG[0x08])

Table 2-63: MPS boot time actions

S386	MPS Mnemonic	Boot time actions
0	disable	None
1	enable	S355=1 (SCO priority over A2DP)

2	SNK/CT/HF	S102 set: A2DP,AVRCP,HF (0x190)
		S102 clear: HFG (0x40)
		S300=1 (A2DP-SNK)
		S301=1 (AVRCP-CT)
		S355=1 (SCO priority over A2DP)

3	SRC/TG/HFG	S102 set: A2DP,AVRCP,HFG (0x1C0)
		S102 clear: HF (0x10)
		S300=2 (A2DP-SRC)
		S301=2 (AVRCP-TG)
		S355=1 (SCO priority over A2DP)

2.7 Automation

As of v18.1.4.0, the BTM51x modules provide features that allow more automated or even hostless operation of the Bluetooth device across a number of profiles. New features such as auto-connect records / auto-connect service, status indication and dynamic registers are described in this section.

As automation features are likely to make extensive usage of non-volatile memory, procedures described in Management of Persistent Store should be taken into account.

For a quick start: Hostless operation provides an example for hostless operation, demonstrating automation features on a BTM511 DVK-V04 (BTM51x audio development kit).

2.7.1 Auto Connect Records (ACR)

The auto connect feature of BTM51x addresses two use cases:

- Automatic connection to *known* devices (and profiles) at boot time
- Automatic re-connection following link loss when returning back into range

Given that BTM51x supports multiple profiles and allows simultaneous connections to different devices with different profiles, a database is required which defines values such as the Bluetooth device address, profiles and the order of the automatic (re-) connection attempts. This database is composed of a list of up to five auto-connect records (ACR) stored in persistent store and referred to as ACR table. There are two options to populate or modify the ACR table (which do not exclude each other):

- Manual mode: full control on each AC record by AT-commands
- Automatic/host-less mode: population of ACR table automatically, based on established connections

To cater for manual mode, a new command family AT+AC… was created. Description of these commands can be found below.

For automatic mode (or host-less mode), a new ACR is added as soon as a connection with a new device is established. If the device is already known in the ACR table, the appropriate ACR is updated if required (e.g. profile flags). On *normal* disconnect (not link loss) the ACR is deleted from the table to prevent automatic reconnection. Only on link loss does the record remain in the table, causing a reconnection when returning back into range. Automatic/host-less mode is enabled by setting Bit 2 of S370.

No matter which option is selected for population, the ACR table can be queried by ATI72.

2.7.1.1 Auto connect record fields

An auto connect record is composed of the following fields:

BdAddr	Bluetooth device address of the remote device
ProfileMask	Flags for all profiles to connect by this ACR, multiple profiles allowed
Interval	Interval after which an attempt cycle is started if not all profiles are connected yet
Status	ACR status, see ATI72 for definition
ProfileMaskConnected	Mask where only actually connected profiles are indicated
OptionFlags	Optional flags, see ATI72 for definition

Refer to the following sections for AT commands to modify the field values.

2.7.1.2 AT+AC… {Editing ACR scratch record (manual mode)}

Before an ACR is added to the ACR table (non-volatile memory), it can be edited in RAM (volatile memory) in the so-called scratch record. This section describes appropriate AT commands.

AT+AC?	Query auto connect scratch record. See ATI72 for syntax definition.
AT+ACC	Clear auto connect scratch record. Set all fields to zero.
AT+ACA=<BdAddr>	Set BdAddr. <BdAddr> = remote device Bluetooth address, 12 hex characters.
AT+ACP=<profile_mask>	Set profile mask. <profile_mask> = as defined by S102. See section 14.3.
AT+ACI=<interval>	Set reconnect attempt interval in seconds. <interval> = decimal number, 0-127.
AT+ACO=<option>	Set option flags. <option> = bitmask, see ATI72 <flags> for definition.

Note: When using AT+ACW to write the scratch record to persistent ACR table, it is inserted at the top and obtain highest priority in the ACS processing order. Refer to Example for Host-less Operation for more details.

2.7.1.3 AT+AC… {ACR table in persistent store (non-volatile memory)}

This section describes AT commands for passing the ACR scratch record to persistent memory and vice versa, as well as how to delete ACR table entries:

AT+ACW	Write scratch record to persistent store (insert at top / prepend), highest priority
AT+ACRn	Replace record in persistent store by scratch record n = index of record to be replaced in ACR table, select from ATI72 response
AT+ACLn	Load (copy) record from persistent store to scratch record n = index of record to be loaded from ACR table, select from ATI72 response
AT+ACDn	Delete one record from persistent store n = index of record to be deleted in ACR table, select from ATI72 response
AT+ACD*	Delete all records from persistent store

To modify an existing ACR table entry with index n, copy it to the scratch record first, using AT+ACLn. In the scratch record the index will be set to zero. Modify it, then copy it back to persistent store. Use AT+ACRn to replace the existing record with index n. Use AT+ACW to insert the scratch record at top of the ACR table. The new index is 1 and all subsequent ACR table entries indices are incremented by one.

Up to five ACRs can be stored in the table. One record can contain multiple profiles. When adding a new record to the top of the table (index 1), the index of all other existing records are increased. This can be interpreted like the index representing the age or history of a record. The oldest record (index=5) is deleted when adding a new record to the complete list.

2.7.1.4 ATI72 {Query ACR table (persistent store)}

ATI72 allows to query the current ACR table from persistent store (= non-volatile memory). Each line presents one entry. If the ACR table contains zero items, response is "0" (more precisely, <cr,lf>0<cr,lf>OK<cr,lf>).

ATI72 response syntax:

```
[<index>],<BdAddr>,$<pm>,<interval>,<state>,$<pm_cnct>,$<flags>
```

With:

<index>	Index of the entry, starting with 1
<BdAddr>	Bluetooth device address of remote device

<pm>		Profile mask as per S102, four hexadecimal characters with leading $
<interval>		Interval of reconnection attempts in seconds
<state>		Status of the ACR:
	1	Attempting connection (at every interval seconds)
	2	Partly connected (at least one profile of pm connected, at least one profile of pm not connected, attempting to connect the remaining profiles)
	3	Fully connected (all profiles of the ACR connected)
<pm_cnct>		Profile mask of actually connected profiles, as per S102, four hexadecimal characters with leading $.
<flags>		Option flags, two hexadecimal characters with leading $, bitmask:
	Bit 0	Always reconnect: If this bit is set, a normal disconnection yields reconnect attempts. If bit is cleared, the record is deleted from the ACR table on normal disconnect. Only a link loss leaves the entry in the ACR-table and allows reconnect attempts.
	Bit 2	Always send AVRCP play
	Bit 3	Never send AVRCP play

Example for ATI72 response:

[1],0016A4001793,$0001,005,3,$0001,$00

[2],C09F43887E15,$0180,005,3,$0180,$00

2.7.1.5 S370=4 {Automatic / host-less ACR mode}

If automatic mode is enabled (Bit2 of S370 is set), new records are automatically created and inserted at top of the ACR table. All field values of the new ACR are retrieved from the new connection, except reconnect interval and option flags. The default reconnect interval is fixed to 5s and option flags are copied from S373.

2.7.1.6 S373 {default ACR option flags}

- These flags populate the ACR option field when the auto-add flag is enabled in S370.
- Range = [0…7]
- Bit 0 – always re-connect (0x01)
- Bit 1 – always send AVRCP play (0x02)
- Bit 2 – never send AVRCP play (0x04)
- If neither Bit1 nor Bit2 is set, AVRCP play is sent intelligently on auto-reconnect depending on certain conditions. AVRCP version 1.0 doesn't support updating the play status at the controller end. Therefore, sending AVRCP play may be unexpected in some cases. This feature is experimental and can be disabled by setting Bit2.

2.7.2 Auto Connect Service (ACS)

Auto connects service is a prerequisite for the auto connect feature. Auto connect service (ACS) is an entity in firmware which caters to required actions such as initiating attempt cycles, monitoring connect/disconnect events, updating the ACR table and more. ACS can be started by AT command (AT+ACS1) or, at boot time, by setting Bit0 of S370.

There are preconditions which must be met before ACS may start:

- Not discoverable, not connectable

- No inquiry in progress
- No pairing in progress
- Legacy SPP auto connect disabled (AT+BTR)
- ACR-table not empty (except automatic/host-less mode by S370 Bit2=1)

An appropriate error message identifies if a condition is not met .

ATI73 or AT+ACS? return the status of ACS. Refer to description of ATI73 for values.

2.7.2.1 ACS parser limitation

Once ACS is running, a limited number of AT commands is available:

- All ATI commands
- AT+AC…
- ATO
- AT+HSB

Any other command returns ERROR 109. To re-enable parser functions, ACS must be stopped.

2.7.2.2 S370 {ACS configuration}

S370 allows configuration of auto connect service as follows:

- S370: Auto Connect Service mode, bitmask

Bit 0	Start AC-service at boot time
Bit 1	Stop AC-service by de-asserting DSR
Bit 2	Auto-add new connections to ACR table, reconnect interval = 5s (auto / hostless ACR mode)
S370 range	0..7; default value: 0; unit: bitmask

2.7.2.3 AT+ACSn {Start/Stop ACS}

AT+ACSn: start/stop auto connect service (ACS)

n = 0	Stop service
n = 1	Start service
If n is missing, n = 0 is assumed; hence AT+ACS = Stop service.	

In order to stop ACS, one can either enter AT+ACS0 or de-assert DSR, provided Bit1 in S370 is set. The latter option has turned out to be more convenient when exploring the auto connect feature. Stopping ACS by DSR is confirmed by OK.

If ACS must be stopped when in an attempt cycle, the confirmation can be delayed by ACS waiting for the attempt outcome. In inconvenient circumstances (e.g. pairing is triggered by connect attempt and experiences significant delay, or multiple profiles are enabled in the ACR) a safety timeout is triggered after 10s which forces ACS to stop and return ERROR 105. The error notes the unclean stop of ACS but guarantees ACS is fully stopped and the parser is functional.

2.7.2.4 AT+ACS? {Query ACS status}

AT+ACS? query status of auto connect service (ACS)

Response is identical to ATI73, see ATI73 {Query ACS status} for ACS status definition.

2.7.2.5 ATI73 {Query ACS status}

ATI73: query auto connect service (ACS) status:

0	Service disabled / not running
1	Service enabled, paused between connection attempts, interval timer running
2	Attempting: initiating connections as defined by ACR table
3	Attempt repeat: resuming connection attempts after a profile with transitional state was found and the transitional state was finished
4	All profiles of ACR-table are connected, monitoring disconnect events

2.7.2.6 ATI74 {Query current ACS interval in seconds}

ATI74 – returns actual current reconnect/attempt interval in seconds. In scenarios where the interval parameter varies across multiple ACRs (which have some disconnected profiles), the smallest value of those ACRs applies and displays.

2.7.2.7 Attempt cycle order / rules

When ACS attempts connections it starts with ACR entries with the lowest index (the highest priority). If not all of the profiles specified are connected, the connection attempt starts attempting to connect all profiles in this record. After connection attempt to one or all profiles fails or after successfully connecting, ACS updates the status of the ACR entry and proceeds with the next ACR entry in the order of increasing index.

If a record is found whose profile mask overlaps with a higher priority (lower index) record and if the higher priority record is at least partly connected, the current (lower priority) record is not processed by ACS.

The duration of an attempt cycle can vary significantly, depending on the number of parameters such as number of profiles, number of ACR entries and connection result (success or failure).

The reconnect interval timer starts when ACS finishes a attempt cycle. If the reconnect interval varies across multiple ACR entries, the lowest interval applies. ATI74 returns the actual current interval in seconds.

2.7.3 Dynamic Registers

Dynamic registers allow complex settings configuration by AT commands. Whereas classic S-Registers allow storage of only one value per register, BTM51x dynamic registers are organized as a dynamic array, meaning that multiple values can be stored. In addition, array elements are not just single values but tuples of values instead. Dynamic registers can be regarded as a two-dimensional array with fixed inner length (tuple) and a variable outer length. This structure is visualised by Figure 2-27.

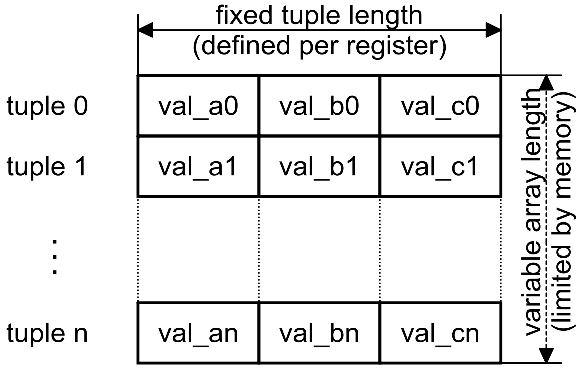


Figure 2-27: Structure of a dynamic register

These registers exist due to automation requirements for host-less operation. It is increasingly necessary to assign conditional actions to events, triggered by a status, a profile etc. The first dynamic registers implemented are S744 and S745 which assign LED blink patterns to module states.

The tuple length is defined per register and can vary from one register to another. The meaning of the particular tuple values and their range is described in the register documentation.

2.7.3.1 Write and Read Syntax

For writing values to dynamic registers, the existing command `ATSxx=<val>` is used with the following extension:

Separator for tuple values: `‘,’`

Separator for array elements: `‘:’`

In order to write three tuples (0,1,2) to dynamic register xx, whereby register xx has a defined tuple length of two values, the following command would be utilised:

```
ATSxx=<val_a0>,<val_b0>:<val_a1>,<val_b1>:<val_a2>,<val_b2>
```

It is also possible to append tuples to an existing dynamic register:

```
ATSxx+=<val_a3>,<val_b3>
```

It is also possible to enter a value in hexadecimal format by prefixing with `‘$’`.

`ATSxx?` reads a dynamic register. The response prints one tuple per line; tuple values are separated by `‘,’`. Tuples themselves (array elements) are separated by carriage return / linefeed (<CRLF>). The response for the above example would look like this:

```
ATSxx?<LF><CRLF>
<val_a0>,<val_b0><CRLF>
<val_a1>,<val_b1><CRLF>
<val_a2>,<val_b2><CRLF>
<val_a3>,<val_b3><CRLF>
OK<CRLF>
```

In contrast to a classic S-register, a dynamic register can be empty. The response for an empty dynamic register is just ‘OK’:

```
ATSxx?<LF><CRLF>
OK<CRLF>
```

2.7.3.2 Dynamic register versus classic S-register

The table below compares significant differences between dynamic and classic S-registers.

Table 2-64: Dynamic S-Registers vs. Classic S-Registers

Characteristic	Dynamic Register	S-Register
Register number range	744…808	0…743
Storage to persistent memory	Automatically, whenever written.	Command: `AT&W`
Delete or restore default value	`ATSxx=` deletes all tuples of the specified dynamic register `AT&F*` deletes all tuples of all dynamic registers and reverts all S-registers to factory default value.	`AT&F*` deletes all tuples of all dynamic registers and reverts all classic S-registers to factory default.
Maximum array length	Maximum number of tuples=31 or limited by persistent store.	n/a

Response to `ATSxx=?` (query range)	`Sxx:(<range_val_a>),(<range_val_b>`
	Example:
	`S744:(1..18),(0..10)`
		`Sxx:(<range_val>)`
		Example:
		`S512:(0..7)`

2.7.4 Status Indication

Status indication means the automatic indication of module states by suitable indicators. Currently (firmware v18.1.4.0), the LEDs of BTM51x are supported as indicators. A number of LED blink patterns exist. LED blink patterns can be tested by AT+SIL0=<PatternID> for LED0 and AT+SIL1=<PatternID> for LED1. Blink pattern IDs are listed in Table 3-5.

For AudioOn status, indication is also possible by a GPIO output, refer to section 16.4.

Motivation for status indication is to achieve host-less operation for typical headset- and wireless speaker use cases.

2.7.4.1 Status definition

There are many states for the BTM51X module with different characteristics and possible parameters. For example, the four scanning states (not connectable, not discoverable / discoverable only / connectable only / discoverable and connectable) are mutually exclusive as only one of these states can be entered at a time. Other states are characterised by particular parameters, e.g. a profile mask. The "connected" state for instance can be determined for each profile. Audio on state can be refined by parameters *audio type* (SCO/A2DP) or *audio direction* (source/sink for A2DP). Furthermore there are transitional states, such as inquiring, pairing, connecting, ringing etc.

Hence, a huge tree of possible states exists, creating the need for simplifications in announcing status indication for ACS. As a result, only the following states were chosen for indication:

- The four scanning states (inquiry-/page scanning = discoverable / connectable)
- Connected (any profile)
- AudioOn (any audio type/direction)

Rather than specifying a profile, the Connected state is regarded as TRUE when any profile is connected. The AudioOn state is regarded as TRUE whenever the audio circuit is turned on, regardless of audio type or direction. See Table 2-65 for a summary.

Table 2-65: ACS Status Summary

Status ID	Status
1	Not connectable, Not discoverable
2	Discoverable
4	Connectable
8	Discoverable, Connectable
16	Connected (any profile)
17	AudioOn (any audio type)
18	Voice recognition on (any audio type)

2.7.4.2 Status priority

The need for simplification also requires definition of a priority scheme. The following illustrates the states we want to indicate over time for a typical use case:

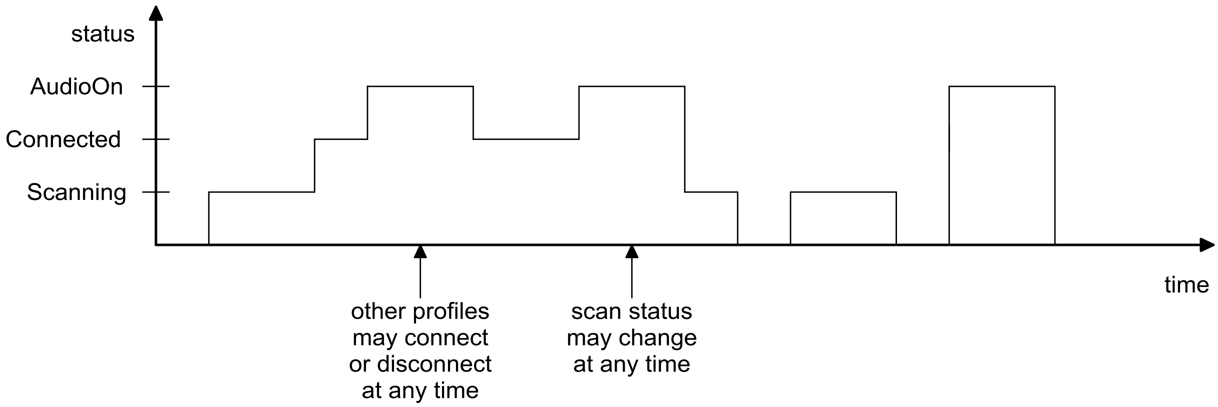


Figure 2-28: Connection States

At first, the module is scanning, meaning it is discoverable and/or connectable. A fast blinking LED is often used to indicate this status. Then, a connection is established, for example an A2DP link initiated from a smartphone. This status is often indicated by a pulsing LED or flashing at long intervals. Later on, the user starts audio streaming by playing music from the smartphone. This could be indicated by an LED which is permanently on. Now, while streaming audio (AudioOn state), another profile could connect (or disconnect), or the scanning status could change. While audio is still on, we don't want to indicate the new "Connected" blink pattern, neither a new "scanning state" blink pattern. Instead it is preferred to stick to indication of the AudioOn state, as long as this state is true.

From this reasoning, we derive the priority rule: in the timeline of a usage scenario, the "innermost" state has the highest priority; the "outermost" state has the lowest priority. In our example the innermost state is "AudioOn" and the outermost state is "Scanning". This leads to the following priority configuration:

Figure 2-29: States defined for indication

Priority	Status ID	Status	Parameters	Comment
1 (low)	1	Not disc., Not con.	no parameters for *scanning states* (mutual exclusive)	ATI27=0
	2	Discoverable only		ATI27=1
	4	Connectable only		ATI27=2

Priority	Status ID	Status	Parameters	Comment
	8	Discoverable, Connectable		ATI27=3
2	16	Connected	none (any profile)	Status regarded as TRUE whenever any profile is connected
3	18	Voice recognition on	none (any audio type)	
4 (high)	17	AudioOn	none (any audio type / direction)	Status regarded as TRUE whenever the audio circuitry is turned on

2.7.4.3 D-Reg 744 / 745 (status to blink pattern mapping)

Dynamic registers 744 (->LED0) and 745 (->LED1) assign an LED blink pattern to a status ID:

- Tuple: <StatusId>,<PatternId>
- Range: (1..17),(0..10)
- Default: none
- For <StatusId> see Table 2-66
- For <PatternId> see Table 2-66

Table 2-66: LED blink pattern IDs

Pattern ID	LED blink pattern
0	LED off
1	LED on
2	Blink fast
3	Blink medium
4	Blink slowly
5	Flash fast
6	Flash medium
7	Flash slowly
8	Pulse fast
9	Pulse medium
10	Pulse slowly

2.7.4.4 AT+SILx=y {Test LED blink pattern}

- AT+SILx=y – test LED blink pattern
  - x = Led-ID: 0 (zero)=LED0, 1=LED1
  - y = blink pattern ID, see Table 2-67.
  - This command can evaluate and find a suitable blink pattern
  - Example: AT+SIL0=9 (pulse medium for LED0)

2.7.4.5 LED indication example

When not discoverable, not connectable, the LED should flash slowly, just to indicate that the module is powered. When any scanning is enabled (discoverable only [2] or connectable only [4] or both [8]), the LED should blink fast (2). Note that scanning status IDs can be added in order to cover multiple scanning states in one tuple. For this example the value is 2+4+8=14. Finally, when any profile is connected (16) the LED shall pulse slowly (10) and when AudioOn (17), the LED shall be permanently on (1). The resulting tuples are summarised in the following table.

Table 2-67: LED status indication example

Status (ID)	Blink Pattern (ID)	Tuple S744 / 745
Not connectable, not discoverable (1)	Flash slowly (7)	1,7
Discoverable/Connectable/Both (14)	Blink fast (2)	14,2
Connected (16)	Pulse slowly (10)	16,10
AudioOn (17)	On permanently (1)	17,1

The resulting command for LED0 is: `ATS744=1,7:14,2:16,10:17,1`

The complete AT script:

AT command	Comment
`AT&F*`	Factory default
`ATS744=1,7:14,2:16,10:17,1`	Set LED0 status - blink pattern assignment
`ATZ`	Reset module (note: dynamic registers are immediately stored to persistent store, AT&W is not needed)
	Verify that LED0 is flashing slowly
`AT+BTQ / AT+BTG / AT+BTP`	Enter various scanning states, LED0 is blinking fast
`AT+BTX`	Enter not discoverable, not connectable, LED0 is flashing slowly

For testing “connected” and “AudioOn” states, see section Hostless operation.

2.7.4.6 Hostless Operation

“Hostless operation” means that no external host microcontroller is needed to control the module. Any input is realized by push buttons and any output is realized by indicators such as LEDs or GPIO outputs.

For hostless operation, ACS is typically started at boot time. The first condition requires the module to be neither discoverable nor connectable at boot time (S512=1). This is why in host-less operation, a GPIO input function mapping code (FMC 0x11) turns the module in discoverable/connectable mode for a time window, specified by S371. This is realized by pressing a button connected to the GPIO input. The remote device is expected to initiate the connection (or pairing) during this time window. LED blink patterns can be configured to indicate the connectable/discoverable status as described in previous sections.

2.7.4.7 Input FMC 0x11 / S371 (enter discoverable/connectable mode with time window)

For input FMC 0x11, the following rules apply:

- A time window can be set by S371 in seconds.
- If S371 is zero, the module remains discoverable and connectable until changed by another AT-command.
- If S371 is greater than zero, the module reverts automatically to not discoverable and not connectable on timeout.
- If auto connect service (ACS) is running while FMC 0x11 is performed (button pressed), ACS is suspended before turning connectable / discoverable. On subsequent window timeout (S371>0) ACS is resumed.
  - S371 range: 0…900; default value: 0; unit: seconds

- Granularity of this S371:
  4 if value is <=60 (round up),
  60 if value is > 60 (round down)

See GPIO function mapping codes (FMC) for details on how to configure GPIO function mapping codes.

2.7.5 Example for Host-less Operation

This section gives an example for host-less operation of a typical headset or wireless speaker use case. It demonstrates the following automation features:

- automatic ACR/ACS mode
- making discoverable/connectable by GPIO
- status indication by LED (dynamic registers)

AT command	Comment
AT&F*	Factory default
ATS515=$200400	Set class of device
ATS370=7	Start AC service at boot time; Stop ACS by DSR; Auto-add new connections to persistent ACR-table
ATS651=$113C	Use GPIO1 to make module discoverable and connectable (BTM51x mini dev kit, all revisions)
ATS371=32	Set discoverable/connectable window to 32 seconds
AT&W	Store settings
`ATS744=1,7:14,2:16,10:17,1`	Set LED0 blink patterns
ATZ	Reset module
AT72	Check ACR table

Now scanning states can be triggered using GPIO_1 (button on BTM511 DVK-V04). When in discoverable and connectable state, connect to the module from a smartphone. Once the connection is established, LED0 should start pulsing slowly. When audio is transmitted from the phone, e.g. by starting playback of music, the LED should turn on permanently as long as audio is activated. If you press the GPIO_1 button again, the blink pattern should not change because a status with higher priority than scanning is being indicated currently. Similarly you may test an incoming SPP connection. Connecting or disconnecting SPP should not change the blinking pattern as long as audio is on or another connection exists (A2DP/AVRCP here).

After audio playback stops some phones leave the A2DP in streaming mode for a few seconds before suspending streaming to save power. On streaming suspend, audio circuitry is disabled and LED0 should fall back to indicate the connected state.

2.8 Hardware Units (BTM510 / 511)

This section covers S-Registers and AT-Commands that are related to hardware units of a BTM510 or BTM511 device. For this section, refer to the Bluecore data sheet [10] for more detailed information.

2.8.1 Analogue Audio Interface

Bluecore BC05 contains an onboard codec, providing analogue input and output capabilities for audio signals. The following sections describe details of this onboard codec.

2.8.1.1 Onboard Codec Gain

Analogue input and output gains (Input Amplifier, Output Amplifier, Figure 2-30) can be set to one of 23 steps called “Gain Level”. To each gain level, an overall gain (dBr) is assigned, according to Table 2-68. The overall gain is formed by an analogue and a digital component as outlined in Table 2-68. Gain values can be specified either as gain level or as overall gain by separate S Registers. Note that a pair of such S-Registers always updates the partner S-Register (e.g. S589 – S689 and S590 – S690).

For S689 and S690 the value must be overall gain in dBr multiplied by ten. If the input value doesn’t match a gain table entry, the nearest possible value is set. The actual value can be checked by reading S689/S690. The value of S689/S690 is printed out multiplied by ten to avoid non integer numbers.

The command class AT+G… enables incremental and decremental gain settings. The increment/decrement command corresponds to one row up/down in the gain table (Table 2-68). The Gain level registers S589/S689 and S590/S690 are not affected by increment/decrement commands. Instead, the current gain level is cached and can be retrieved by AT+G(I|O)?. There are two further commands to restore the cached gain level from S589/S590. They are AT+G(I|O)R) and (AT+G(I|O)S. The latter saves the currently cached gain level to S589/S590.

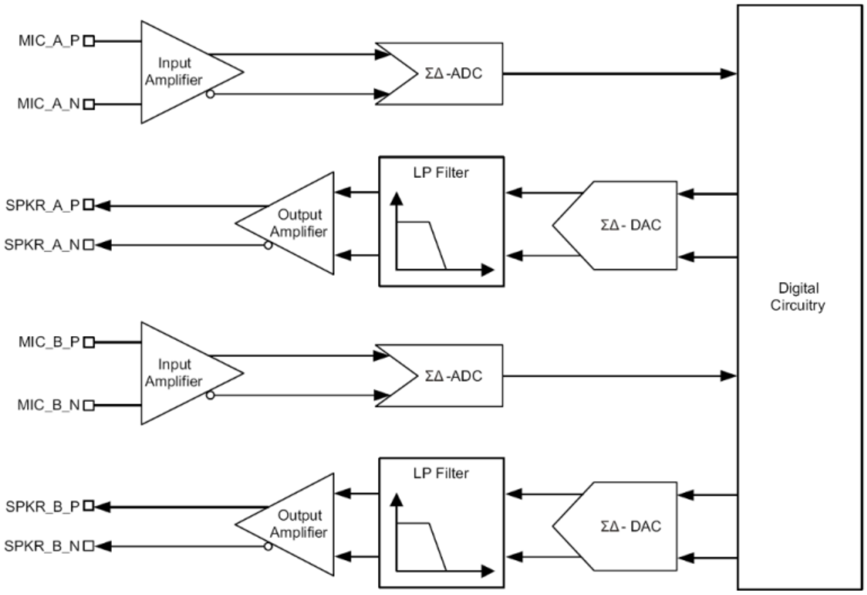


Figure 2-30: BTM51x Stereo Audio Codec Block Diagram

Table 2-68: BTM51x Gain Table

Output Gain Level or Input Gain Level S589 or S590	Overall Gain (dBr) S689 or S690	Digital Component	Analogue Component
22	+21.5	7	7
21	+18.0	6	7
20	+15.5	5	7
19	+12.0	4	7

Output Gain Level or Input Gain Level S589 or S590	Overall Gain (dBr) S689 or S690	Digital Component	Analogue Component
18	+9.5	3	7
17	+6.0	2	7
16	+3.5	1	7
15	0	0	7
14	-3.0	0	6
13	-6.0	0	5
12	-9.0	0	4
11	-12.0	0	3
10	-15.0	0	2
9	-18.0	0	1
8	-21.0	0	0
7	-23.5	15	0
6	-27.0	14	0
5	-29.5	13	0
4	-33.0	12	0
3	-35.5	11	0
2	-39.0	10	0
1	-41.5	9	0
0	-45.0	8	0

Table 2-69: BTM51x Gain Settings AT commands and S-Registers

Task	AT-Command/ SRegister	Comment
Set output gain level	S589 [0..22], default=12	See Gain Table; S689 is also affected.
Set output overall gain (dBr)	S689 [-450..+215], default=-90	See GainTable; value must be entered (and is returned) multiplied by 10; S589 is also affected.
Set input gain level	S590 [0..22], default=12	See Gain Table; S690 is also affected.
Set input overall gain (dBr)	S690 [-450..+215], default=-90	See GainTable; value must be entered (and is returned) multiplied by 10; S590 is also affected.
Increment current output gain level	AT+GOU	Error 57 appears if maximum gain level is reached.

Task	AT-Command/ SRegister	Comment
Decrement current output gain level	AT+GOD	Error 58 appears if minimum gain level is reached.
Set current output gain level	AT+GO<n>	Range for <n> = 0…15 (dec)
Query current output gain level	AT+GO?	
Restore current output gain level from S589	AT+GOR	
Save current output gain level to S589	AT+GOS	
Increment current input gain level	AT+GIU	Error 57 appears if maximum gain level is reached.
Decrement current input gain level	AT+GID	Error 58 appears if minimum gain level is reached.
Set current input gain level	AT+GI<n>	Range for <n> = 0…15 (dec)
Query current input gain level	AT+GI?	
Restore current input gain level from S590	AT+GIR	
Save current input gain level to S590	AT+GIS	

2.8.1.2 Onboard Mic Input Gain

S Register 415 controls a microphone preamplifier, which adds extra 24 dB to input gain. The amplifier is enabled by ATS415=1 and disabled by ATS415=0. Refer to Table 2-72. The first amplifier in Figure 2-31 represents the microphone preamplifier and the second amplifier represents the analogue component of the programmable audio input gain (refer to Table 2-73).

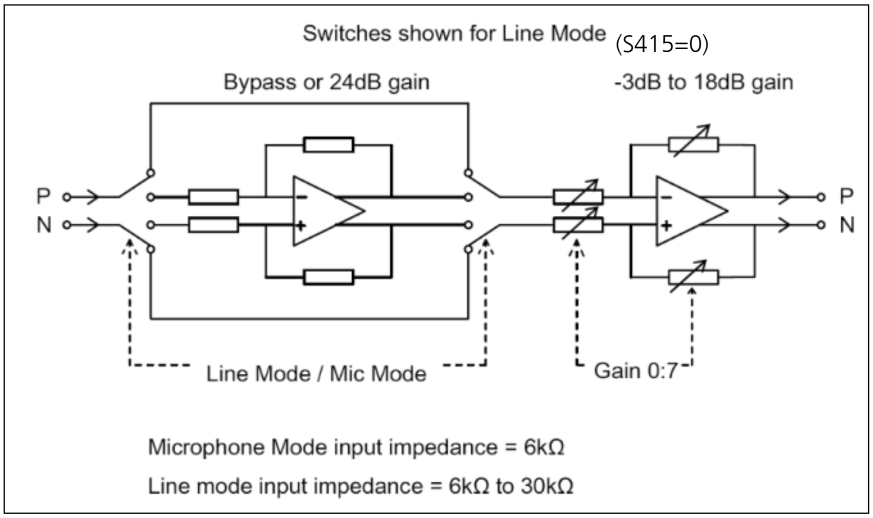

Switches shown for Line Mode
(S415=0)
Bypass or 24dB gain
-3dB to 18dB gain
P
N
P
N
+
+
Line Mode / Mic Mode
Gain 0:7
Microphone Mode input impedance = 6kΩ
Line mode input impedance = 6kΩ to 30kΩ

Figure 2-31: BTM51x ADC Analogue Amplifier Block Diagram

2.8.2 Digital Audio Interface

Bluecore BC05 allows audio streams to be routed to a digital audio interface instead of the internal codec. These sections describe details of the digital audio interface.

2.8.2.1 I2S – Inter Integrated Circuit Sound

I2S is a digital interface to transmit stereo audio data between integrated circuits (IC) in embedded systems. It connects an external audio CODEC or a wave file player/recorder to a BTM module. As most of audio media is available digitally these days, there is no need to convert audio to analogue and then back to digital on the BTM5xx module.

Provided a digital medium is played over A2DP, with a BTM5xx as A2DP source, then a D/A and subsequent A/D conversion between the digital medium and the module (A2DP source) would lose signal quality due to non-optimal settings of DAC / ADC gains and noise picked up in the analogue path.

These issues can be eliminated by using the digital I2S interface rather than the analogue audio input.

An I2S interface consists of at least 3 signals: bus clock (SCK), word select (WS, left/right channel) and serial data (SD_IN / SD_OUT). On a BTM5xx module, there are separate data lines for in input and output, for a total of four I2S signals. I2S signals are mapped to the PCM pins of a BTM51x as outlined in Table 2-70.

When connecting two devices via I2S, one must be the I2S master and the other must be the I2S slave. The I2S master provides SCK and WS signals (output) and the I2S slave synchronises to these signals (WS and SCK = input). S registers S314 (A2DP) and S315 (SCO) select the I2S role for a BTM51x (I2S master / I2S slave).

Table 2-70: BTM51x I2S signal pin mapping

I2S Signal name	BTM51x Signal name	BTM51x module pin no.	Mini devkit PCM header (J5)	Comment
SD_IN	PCM_IN	3	5	I2S Serial data in
SD_OUT	PCM_OUT	4	7	I2S Serial data out
WS	PCM_SYNC	5	6	I2S Word select (right/left ch)
SCK	PCM_CLK	6	4	I2S clock
	GND1/2/3/4	[9,22,13,47]	10	GND

2.8.2.1.1 I2S - MCLK

In addition to the I2S signals mentioned above, many external codec ICs require a high frequency master clock (MCLK). Typically this clock is a multiple integer of the sampling rate f_s, e.g. 128 f_s / 192 f_s / 256 f_s / 384 f_s / 512 f_s and so forth. Unfortunately, BTM51x cannot provide such MCLK signal. Laird recommends you select an external codec IC which has the capability of creating the MCLK internally from the bit clock signal (SCK). Such capability is referred to as “PLL” (phase locked loop) or “FLL” (frequency locked loop) functionality of the codec device.

2.8.2.1.2 I2S example

This section describes an example where the I2S interface of a BTM5xx is used as digital stereo input of an A2DP source.

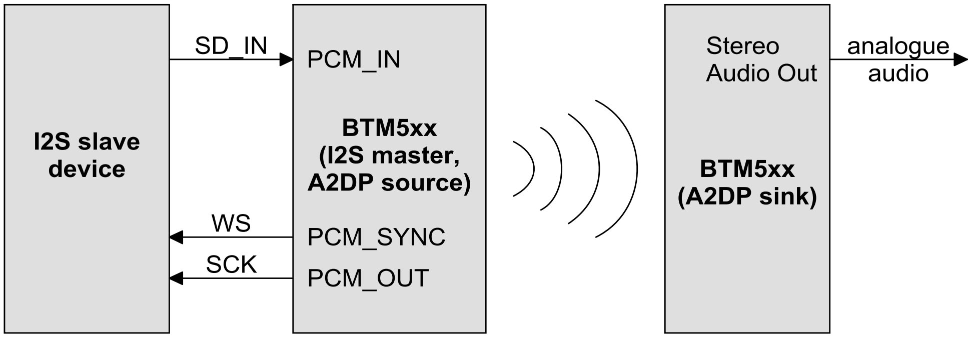


Figure 2-32: BTM51x I2S example block diagram

Connect an I2S (slave) source as outlined in the block diagram (Table 2-56). The SD_OUT connection is not needed for this example. The pin mapping is listed in Table 2-70. A third BTM5xx device, configured in I2S cross loopback mode (AT+BTL4), may be the I2S slave device. It then acts as analogue-to-I2S converter. Table 2-71 shows the required AT command sequence.

Table 2-71: BTM51x I2S example AT-command sequence

Analogue to I2S converter (1)	A2DP Source with I2S input	A2DP Sink with analogue output	Comment
AT&F*	AT&F*	AT&F*	Restore factory default settings
AT+BTL4			Configure I2S (slave) source (1)
	ATS314=1		Select I2S (master) interface at A2DP source
	ATS300=2		Enable Source role
	ATS512=4		Make connectable and discoverable
	AT&W	AT&W	Store S Register configuration
	ATZ	ATZ	Reset
		AT+APD<Bluetooth Address Source>	Initiate connection. Get Bluetooth address of other end by ATI4 on other end or by inquiry (AT+BTI).
1) For demonstration purposes or if no other I2S source is available, a 3rd BTM5xx can be used as analogue to I2S converter in I2S cross loopback mode.			

2.8.2.1.3 Routing audio streams over I2S

S314 selects the audio interface to use when entering an A2DP connection. S315 selects the audio interface for entering an SCO connection. Available options for both S314 and S315 are: 0 - internal codec, 1 - I2S master and 2 - I2S slave.

Table 2-72: BTM51x S Registers Audio Routing

Register	Default	Range	Description

S314	0	0..2	A2DP audio interface select 0 = internal codec (default) 1 = I2S master 2 = I2S slave
S315	0	0..3	SCO audio interface select 0 = internal codec (default) 1 = I2S master (I2S_M) 2 = I2S slave (I2S_S) 3 = PCM master (PCM_M) 4 = PCM slave (PCM_S)

An asynchronous message indicating the current sampling rate and the audio interface is sent to the UART whenever entering a SCO or A2DP connection. The format of this message is:

FSddddd,<interface>
ddddd = sampling frequency in Hz (4 or 5 digits)
<interface> = INT (internal codec) / I2S_M (I2S master) / I2S_S (I2S slave)

S309 or S504 may disable this FS-message.

2.8.2.1.4 A2DP sampling rate capability

When entering an A2DP connection, a negotiation phase finds the best sampling frequency (air rate) that matches both ends. Therefore, each A2DP end must expose its supported sampling frequencies.

When using the BTM5xx internal codec, the supported sampling frequencies are known, so the user does not need to do anything.

But when using the I2S interface, the supported sampling rates of the external circuit are not known by the BTM5xx. The user must specify all sampling rates supported by S Register 316. Check Table 2-72 for details on S316. The negotiated sampling frequency is displayed by the "FSddddd,<interface>" asynchronous message, see Routing audio streams over I2S.

S316 is referenced at boot time for building the sampling rate capability record if S314>0 (I2S interface for enabled A2DP). Before changed values of S316 or S314 become effective, commands AT&W and ATZ must be sent to the module.

Table 2-73: BTM51x A2DP supported sampling rates for I2S

Register	Default	Range	Description
S316	63	1..63	I2S sampling rate capability for A2DP; values can be added for all sampling frequencies supported Register is referenced if S314>0 at boot time, requires AT&W and ATZ for new values to become effective 1 = 48 kHz[(1)] 2 = 44.1 kHz[(1)] 4 = 32 kHz 8 = 24 kHz[(2)] 16 = 22.05 kHz[(2)] 32 = 16 kHz

(1) A2DP Source: at least one of these sampling frequencies (48kHz, 44.1kHz) must be supported;
A2DP Sink : both 48 kHz and 44.1 kHz must be supported (at least)

Register	Default	Range	Description
(2) values ignored as they are not supported by SBC, neither by APTx			

2.8.2.1.5 I2S Data Format

I2S data is transferred in so called "I2S mode". This means left-justified mode with the MSB starting one SCK cycle delayed after transition of the WS signal. The number of bits per sample is set to 24 (=24 SCK cycles between two WS transitions), but the actual sampling bit length is only 16 bit. Hence, 8 SCK cycles are not used for data transmission. Refer to Figure 2-33.

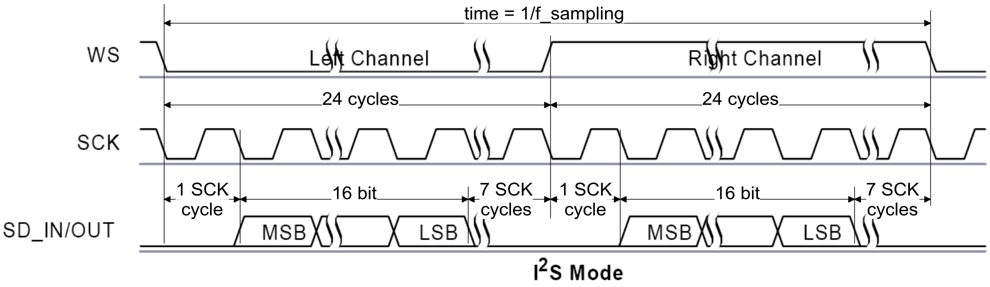


Figure 2-33: BTM51x I2S data format

This standard I2S data format can be modified to customer needs per S-register S384. In contrast to earlier f/w versions, as of v22.2.5.0 it is not required any longer to use a PSKEY for I2S data format modification. All settings of PSKEY_DIGITAL_AUDIO_CONFIG are now replaced by S384. Please refer to the S-Registers Appendix for details on S384.

2.8.2.1.6 I2S Cross Loopback

I2S functions can be tested by a feature called I2S Cross Loopback mode. In this mode, analogue audio input (stereo) is routed to SD_OUT and at the same time SD_IN is routed to analogue audio output (stereo). I2S cross loopback mode is enabled by AT+BTL3 (I2S master role) or by AT+BTL4 (I2S slave role). Figure 2-34 and Figure 2-35 show a block diagram of I2S cross loopback mode. The sampling frequency for this mode can be changed by S-Register 419, as outlined inTable 2-77.

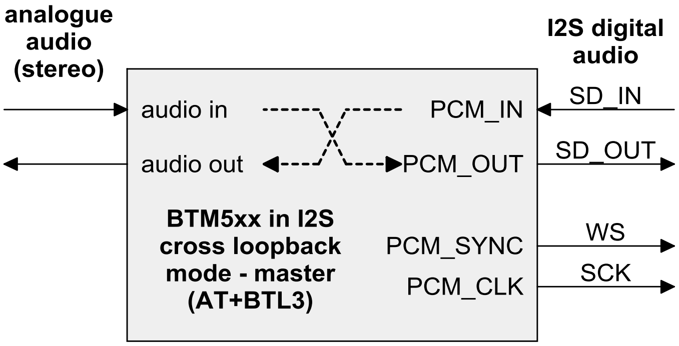


Figure 2-34: BTM51x I2S cross loopback - master

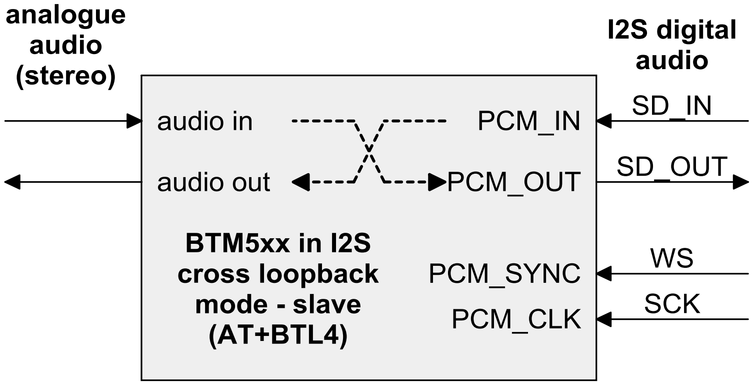


Figure 2-35: BTM51x I2S cross loopback – slave

2.8.2.1.7 I2S Use case: Wave File Player/Recorder

One of the use cases for I2S is the wave file player/recorder. In this use case a wave file is the digital source (player scenario) which is being played back with its recording sampling rate. If the recording sampling rate of the file is different to the playing sampling rate, a sampling rate conversion must be performed by the player entity before data is put on the I2S bus. “Play” in this context means data is shifted out to the I2S bus (SD) with clock timing controlled by only WS and SCK.

Similarly, in the wave file recorder scenario, an incoming audio bit stream is recorded to memory and the timing is controlled only by SCK and WS.

In this use case the SCK and WS signal are fully sufficient to synchronize between the two devices.

2.8.2.1.8 I2S Use case: External Audio CODEC

A second use case for I2S is the external audio codec scenario. This may be needed for higher subjective sound quality granted by dedicated audio codecs with advanced DAC’s / ADCs and advanced audio filters. A system with external audio codec typically requires a high frequency master clock signal in addition to the I2S synchronisation lines SCK and WS. This clock drives the external CODEC’s filter circuits and subsequent internal circuits with precise synchronisation to the master clock. This high frequency master clock is typically in a range of 24 MHz or 12 MHz. Because the BTM5xx cannot synchronise to an external clock as a slave for this purpose, it would have to provide this master clock signal.

BTM51x cannot provide this high frequency master clock signal due to a hardware limitation.

When operating an external CODEC without synchronisation, artefacts are audible in periodic intervals. It has been observed that these artefacts appear at around 15 kHz in the audio spectrum and are audible as noise.

2.8.2.2 PCM – Pulse code modulation interface

Audio data of a SCO link can be routed to/from the PCM interface. The PCM is intended to carry bidirectional voice data at typically 8 kHz sampling frequency, but not stereo audio. The PCM interface is composed of 4 signals as outlined in Table 2-70: PCM_IN, PCM_OUT, PCM_SYNC, PCM_CLK. These signals are shared with the I2S bus, so either I2S or PCM can be used but not both simultaneously.

The PCM interface offers a wide range of configuration options described in the BC05-EXT data sheet [10], chapter 10 (page 50). Unfortunately this configuration cannot be accomplished by AT commands.

Instead, use PSKEYs. Laird Technologies provides the utility PsUpdate.bat which allows certain PSKEYs to be downloaded to a BTM51x module over the UART. The PSKEYS that apply for SCO configuration are:

PSKEY_PCM_CONFIG32	&01B3	(**Note:** For I2S only.)
PSKEY_PCM_FORMAT	&01B6	
PSKEY_PCM_LOW_JITTER_CONFIG	&01BA	

The values for these keys are passed to PsUpdate.bat in a *.psr file.

For more on downloading PSKEYs over UART see CVC on BTM5xx, provided by Laird Technologies under NDA.

2.8.2.3 S/PDIF – Sony/Philips Digital Interconnect Format (not supported)

S/PDIF is another digital audio interface for high quality stereo audio. It is not supported by the current firmware version (22.2.5.0)

2.8.2.4 Fixed sampling rate (PCM / I2S), S419 mapping

As of firmware v22.2.5.0 for PCM and I2S, a fixed sampling rate can be configured. This allows an external codec to run at a defined sampling rate, regardless of the actually negotiated A2DP rate (for example: 44.1 kHz or 48 kHz) or (e)SCO air rate. With this feature, reconfiguring an external codec is no longer required each time depending on the **FS…** message from BTM51x's UART when entering an audio link.

The fixed sampling rate is achieved by utilising appropriate resampling functions on the internal DSP.

- Fixed sampling rate applies to SCO links for all CVC variants and X1M (S318>0) for both narrow band (8 kHz) and wide band (16 kHz).
- Fixed sampling rate also applies to all A2DP decoders (SBC, APTX, and AAC sink).
- Fixed sampling rate does not apply to SCO links if S318=0 (no DSP plugin).
- Fixed sampling rate does not apply to A2DP encoders (SBC, APTX source).

If a fixed sampling rate has been enabled, then the fixed configured rate is appended to the FS-message. If the first FS-value is different from the second, then resampling routines are active on the DSP.

For example:

FS8000,I2S_M,441000

…indicates a narrow band SCO link (8 kHz air rate) with sampling rate converted up to 44.1 kHz for I2S interface, I2S configured as master.

S-register 419 allows enabling and configuring a fixed sampling rate. Values of S419 are mapped to actual sampling rates depending on the context (SCO/A2DP/PCM/I2S) as outlined by

Table 2-74.

Table 2-74: S419 – SCO/A2DP fixed sampling rate

S419 - Fixed Sampling Rate			
S419	SCO audio		A2DP sink
	PCM [S315=3/4]	I2S [S315=1/2]	I2S [S314=1/2]
0	Disabled	Disabled	Disabled
[1,2]	16 kHz	16 kHz	44.1 kHz
[3..5]		32 kHz	
6		44.1 kHz	
7		48 kHz	48 kHz

If fixed sampling rate feature is disabled (S419=0) then no resampling is applied. The sampling rate at the digital interface corresponds to the air rate (such as 8/16 kHz for SCO or 44.1/48 kHz for A2DP).

S419 is not only used to configure a fixed sampling rate for SCO or A2DP; it also controls the sampling rate for audio loopback mode. This loopback mode is intended for testing purposes and is mutual exclusive to SCO and A2DP audio. Because of this, the context-dependent mapping of S419 should not present an issue.

In audio loopback mode, analogue and I2S interface can be configured for a few more sampling rates than the fixed rates available. The following table shows the sampling rate mapping of S419 in loopback mode.

Table 2-75: S419 - loopback mode sampling rate

S419 - Sampling Rate for Loopback Mode			
S419	Analogue [AT+BTL1]	PCM cross LB [AT+BTL7/8]	I2S cross LB [AT+BTL3/4]
0	8 kHz	8 kHz	8 kHz
1	11.025 kHz		11.025 kHz
2	16 kHz	16 kHz	16 kHz
3	22.05 kHz		22.05 kHz
4	24 kHz		24 kHz
5	32 kHz		32 kHz
6	44.1 kHz		44.1 kHz
7			48 kHz (1)

(1) Only for path I2S_in > analogue_out (DAC); BTM51x analogue in (ADC) does not support 48 kHz sampling rate.

2.8.2.4.1 Fixed Sampling Rate, Example A

In this first example we assume the following:

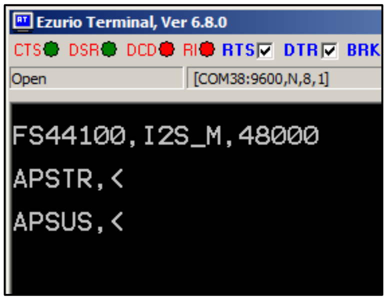
- External codec with I2S interface
- Fixed sampling rate = 48 kHz.
- Use for A2DP sink and for SCO (HF)

A2DP Sink, I2S Output	Comment
AT&F*	Restore factory default settings
ATS102=$90	Enable profiles: A2DP, HF
ATS314=1	Configure I2S (master) for A2DP
ATS315=1	Configure I2S (master) for SCO
ATS419=7	Configure fixed sampling rate for A2DP and SCO: 48kHz
ATS512=4	Make discoverable and connectable at boot time
ATS515=$200400	Set device class to be found by A2DP sources
AT&W	Save settings
ATZ	Reset

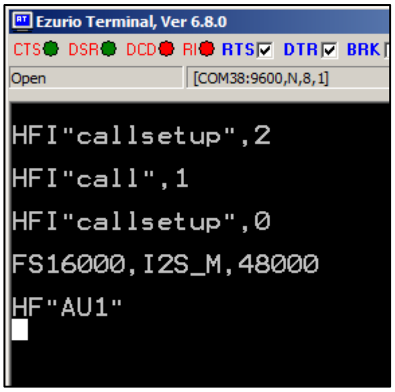
After this configuration we use an iPhone to find the BTM51x and connect to it. Pairing should automatically occur and A2DP and HF will connect. Next we start playing music on the iPhone. The audio output is routed to the connected BTM51x (A2DP) and we can see the FS message from :

```
FS44100,I2S_M,48000
```

This shows that the air rate is 44.1 kHz, audio being routed to I2S in master role (*I2S_M*) and the output rate is 48 kHz. The fact that air rate and I2S rate are different shows that a sampling rate conversion is active.



```
FS44100,I2S_M,48000
APSTR,<
APSUS,<
```



```
HFI"callsetup",2
HFI"call",1
HFI"callsetup",0
FS16000,I2S_M,48000
HF"AU1"
```

Figure 2-36: FS Message

Next, we stop the playback on the phone. After a few seconds the phone will suspend the A2DP stream (*APSUS,<*). Now we create a SCO link. One option is to use Siri for this purpose: keeping the phone main button pressed for a few seconds will launch Siri and create an eSCO link as shown in the second screenshot (right) with the following FS message:

```
FS16000,I2S_M,48000
```

This shows that a channel at 16kHz *(FS16000)* sampling rate has been created (which must be wideband eSCO), that audio is routed to I2S in master role (*I2S_M*) and that the stream is being up-sampled to 48 kHz *(,48000)*.

Finally we disconnect iPhone, e.g. by disabling Bluetooth on the phone. Now we connect a different device which will negotiate an A2DP air rate of 48 kHz. This can be achieved by a PC with Bluetooth and Windows 7. BTM51x should be visible when discovering/adding a new device. We need to confirm the 6 digit pass key on PC end. On BTM51x we need to enter 'AT+BTBY' in response to the passkey compare request. After windows has installed the drivers and has configured the new device we are offered to connect the 'Stereo Headset Operations' as shown in Figure 2-37.

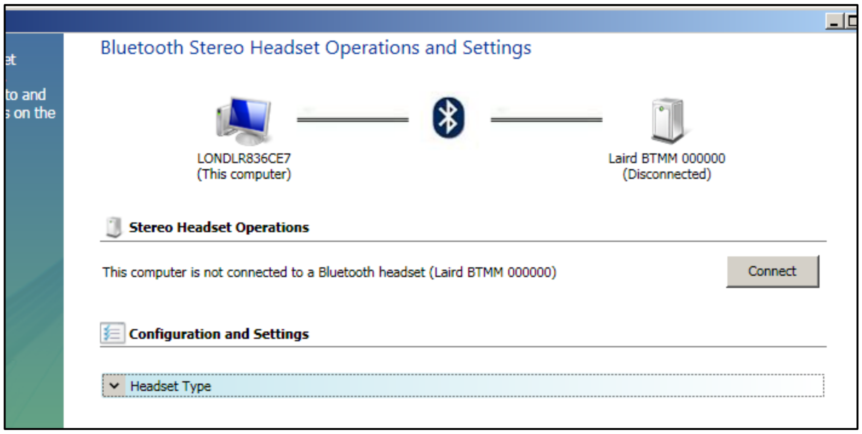
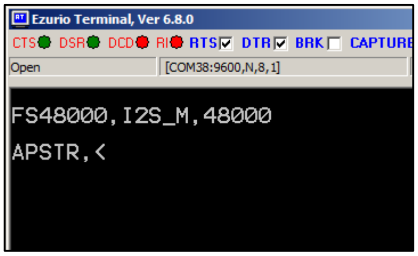

et
to and
s on the
Bluetooth Stereo Headset Operations and Settings
LONDLR836CE7
(This computer)
Laird BTMM 000000
(Disconnected)
Stereo Headset Operations
This computer is not connected to a Bluetooth headset (Laird BTMM 000000)
Connect
Configuration and Settings
Headset Type

Figure 2-37: "Connect" under Stereo Headset Operations

After doing so we can see the following FS message:

```
    FS48000,I2S_M,48000
```


Ezurio Terminal, Ver 6.8.0
CTS DSR DCD RI RTS☑ DTR☑ BRK□ CAPTURE
Open
[COM38:9600,N,8,1]

```
FS48000,I2S_M,48000
APSTR,<
```

Figure 2-38: FS message

This shows that the A2DP air rate is 48 kHz, audio being routed to I2S (master role) and I2S rate is 48 kHz. Now that air rate and I2S rate are identical (48 kHz), no resampling takes actually place. The second ‘48000’ still indicates that this fixed sampling rate has been configured by S419 (=7).

2.8.2.4.2 Fixed Sampling Rate, Example B

In this second example we assume the following:

- External codec with PCM interface
- Fixed sampling rate – 16 kHz
- SCO connections only (HF profile)

Device A: HF PCM for SCO	Device B: HFG Helper Device	Comment
AT&F*	AT&F*	Restore factory default settings
ATS102=$10	ATS102=$40	Enable profiles: HF for DevA, HFG for DevB
ATS315=3		Configure PCM (master) for SCO on DevA
ATS419=2		Configure fixed sampling rate for SCO: 16 kHz

Device A: HF PCM for SCO	Device B: HFG Helper Device	Comment
ATS512=4		Make discoverable and connectable at boot time
AT&W	AT&W	Save settings
ATZ	ATZ	Reset
ATI4		Query BdAddr of device A
	AT+HFGD<BdAddrA)	Initiate connection from device B
AT+HFA		Initiate audio from device A
AT+HFR		Release audio from device A
	AT+HFGW0	Disable WBS on device B, next SCO is narrow band/8 kHz
AT+HFA		Initiate audio from device A
AT+HFH		Release audio and SLC from device A

We use two BTM51x which are configured as HF (device A) and HFG (device B, helper device). After configuration according to the above table, we initiate the connection from device B. On the first “AT+HFA” we see the following FS message:

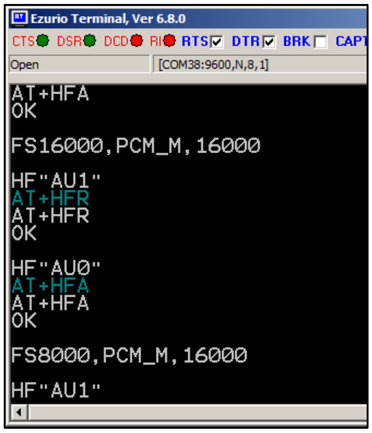
```
FS16000,PCM_M,16000
```

…meaning that this is a WBS eSCO link (‘FS16000’) which is routed to PCM (master role) at 16kHZ (‘,*16000*’). Although in this case sampling rate conversion is not required ([virtual] air rate is identical to PCM rate), the second *16000* still indicates that a fixed rate is configured by S419, which will apply if the air rate changes.

After releasing the audio from device A (AT+HFR), we disable WBS on device B so that any subsequent SCO link will be at 8kHz (narrow band). Initiating audio again on device A shows indeed an air rate of 8kHz:

```
FS8000,PCM_M,16000
```

Now the difference between air rate (8kHz) and PCM rate (16kHz) indicates that the relevant up-sampling is taking place. Finally we close audio and SLC. The screenshot below shows both FS messages.


Ezurio Terminal, Ver 6.8.0
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR ☑ BRK □ CAPT
Open
[COM38:9600,N,8,1]

```
AT+HFA
OK

FS16000,PCM_M,16000

HF"AU1"
AT+HFR
AT+HFR
OK

HF"AU0"
AT+HFA
AT+HFA
OK

FS8000,PCM_M,16000

HF"AU1"
```

2.8.3 Audio Routing Options for SCO

Table 2-76 provides an overview of combinations between profiles and audio interfaces that are supported. *SCO audio* refers to bidirectional mono audio for transmission of speech with 8 kHz or 16 kHz sampling rate.

The PCM pins of the BTM51x share support for PCM, I2S and SPDIF. S315 selects one of these interfaces or the internal codec ("internal codec" refers to the on-board audio ADC/DAC, clock generator and antialiasing filters, hence the onboard analogue interface).

SPDIF is currently not supported but seems feasible (1). For I2S (normally used for stereo audio) ensure the sampling rate is configured to 8 kHz if used with SCO. Otherwise the sampling rate mismatch produces distorted audio.

If the fixed sampling rate feature is enabled for I2S, then the above requirement is obsolete. Please refer to Fixed sampling rate (PCM/I2S), S419 mapping.

Table 2-76: SCO audio routing options

	HF [S102=$10] HS [S102=2]	HFG [S102=$40] HSG [S102=8]	SPP-SCO [AT+BTAx]
Internal Codec [S315=0]	✓	✓	✓
PCM [S315=3]	✓	✓	✓
I2S [S315=1/2]	✓ (8kHz)	✓ (8kHz)	✓ (8kHz)
SPDIF	⊘1	⊘1	⊘1

(1) SPDIF not supported currently, but seems feasible.

Internal Codec – Analogue, all other interfaces are digital external audio interfaces

2.8.4 SCO Input/Output Channel

If SCO (which is mono) is routed over an audio interface, which normally supports stereo, the left channel is selected for the input signal.

	Input Channel (SCO)	Output Channel (SCO)
Internal Codec	A (left)	A and B (left and right)
I2S	Left	Left and Right

Note: On previous versions of the BTM51x mini dev. kit the stereo input connector, left and right channels are swapped (tip=right, ring=left, sleeve=ground,) The correct assignment for the jack is: tip=left, ring=right, sleeve=ground.

This issue is resolved as of the BTM511-DVK-V04.

2.8.5 Audio Loopback Mode

For testing purposes, an audio loopback mode is available. Audio loopback mode is controlled by the AT command AT+BTLm with m = mode [0..4].

In mode 1, audio input (ADC) and audio output (DAC) are connected directly. In mode 2, the stereo audio input signal is fed through the Kalimba DSP with SBC codec running (encoder, decoder) and is directed back to the audio stereo output.

Table 2-77: BTM51x Audio Loopback AT-commands and S-Registers

Task	AT-Command/ SRegister	Comment
Set audio loopback mode	AT+BTL<Mode>	Mode: 0 Off 1 On, ADC > DAC 3 On, I2S cross loopback, master 4 On, I2S cross loopback, slave 7 On, PCM cross loopback, master 8 On, PCM cross loopback, slave

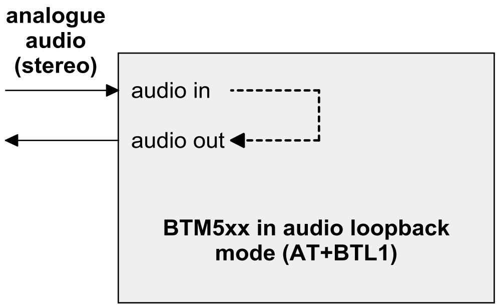


Figure 2-39: BTM51x Audio Loopback Modes 1 and 2

2.8.6 LED Control

The module provides two dedicated output pins for LEDs (LED0, LED1). The following modes are supported: LED_OFF, LED_ON, LED_PWM, and LED_PULSE.

In LED_PWM mode, the parameters Duty Cycle and PWM Period can be specified via S-Registers. This enables either to dim the brightness of an LED (PWM Period=0) or to specify blinking with defined on-time in a defined period (PWM Period > blinking visible for the eye).

In LED_PULSE mode, the brightness of an LED is modulated. Modulation speed is defined by parameter Pulse Rate and maximum brightness is defined by parameter Duty Cycle.

Table 2-78: BTM51x LED S-Registers

Task	AT-Command/ SRegister	Comment
Set LED 0 mode	S335 [0..3], default=0	Mode: 0 LED_OFF 1 LED_ON 2 LED_PWM 3 LED_PULSE
Set LED 1 mode	S340 [0..3], default=0	
Set LED 0 Duty Cycle	S336 [0..4095], default = 2048	referenced if LED mode = LED_PWM or LED_PULSE
Set LED 1 Duty Cycle	S341 [0..4095], default = 2048	
Set LED 0 PWM Period	S337 [0..15], default = 0	referenced if LED mode = LED_PWM
Set LED 1 PWM Period	S342 [0..15], default = 0	
Set LED 0 Pulse Rate	S338 [0..15], default = 0	referenced if LED mode = LED_PULSE
Set LED 1 Pulse Rate	S343 [0..15], default = 0	

2.8.7 Hardware Units – S Registers

Table 2-79 provides an overview on S Registers for hardware units except GPIO. For GPIO Registers refer to Table 2-83.

Table 2-79: BTM51x S Registers for Hardware Units

Register	Default	Range	Description
S309	1	0..1	Enable asynchronous FS-message (presenting sampling frequency and audio interface) 0 Disable 1 Enable (default)
S314	0	0..2	A2DP audio interface select 0 = internal codec (default) 1 I2S master 2 I2S slave

Register	Default	Range	Description
S315	0	0..3	SCO audio interface select 0 Internal codec (default) 1 I2S master 2 I2S slave 3 PCM master (PCM_M) 4 PCM slave (PCM_S)
S316	63	1..63	I2S sampling rate capability for A2DP; values can be added for all sampling frequencies supported Register is referenced if S314>0 at boot time, requires AT&W and ATZ for new values to become effective 1 48 kHz(1) 2 44.1 kHz(1) 4 32 kHz 8 24 kHz(2) 16 22.05 kHz(2) 32 16 kHz (1) A2DP Source: at least one of these sampling frequencies (48kHz, 44.1kHz) must be supported; A2DP Sink : both 48 kHz and 44.1 kHz must be supported (at least) (2) values ignored as they are not supported by SBC, neither by APTx
S415	0	0..1	Enable Microphone Input Gain, adds extra 24 dB to input gain

<table>
<tr><th>Register</th><th>Default</th><th>Range</th><th>Description</th></tr>
<tr><td>S419</td><td>0</td><td>0..7</td><td>Set sampling rate for Audio Loopback and Fixed sampling rate:
<table>
<tr><th colspan="4">Loopback Mode Sampling Rate</th></tr>
<tr><th>S419</th><th>Analogue
[AT+BTL1]</th><th>PCM cross LB
[AT+BTL7/8]</th><th>I2S cross LB
[AT+BTL3/4]</th></tr>
<tr><td>0</td><td>8 kHz</td><td rowspan="2">8 kHz</td><td>8 kHz</td></tr>
<tr><td>1</td><td>11.025 kHz</td><td>11.025 kHz</td></tr>
<tr><td>2</td><td>16 kHz</td><td rowspan="6">16 kHz</td><td>16 kHz</td></tr>
<tr><td>3</td><td>22.05 kHz</td><td>22.05 kHz</td></tr>
<tr><td>4</td><td>24 kHz</td><td>24 kHz</td></tr>
<tr><td>5</td><td>32 kHz</td><td>32 kHz</td></tr>
<tr><td>6</td><td rowspan="2">44.1 kHz</td><td>44.1 kHz</td></tr>
<tr><td>7</td><td>48 kHz[1]</td></tr>
</table>
(1) Only for path I2S_IN –> analogue_out (DAC). Analogue in (ADC) does not support a 48 kHz sampling rate.
<table>
<tr><th colspan="4">SCO/A2DP Fixed Sampling Rate</th></tr>
<tr><th rowspan="2">S419</th><th colspan="2">SCO Audio</th><th>A2DP Sink</th></tr>
<tr><th>PCM
[S315=3/4]</th><th>I2S [S315=1/2]</th><th>I2S [S314=1/2]</th></tr>
<tr><td>0</td><td>Disabled</td><td>Disabled</td><td>Disabled</td></tr>
<tr><td>[1,2]</td><td rowspan="4">16 kHz</td><td>16 kHz</td><td rowspan="3">44.1 kHz</td></tr>
<tr><td>[3,5]</td><td>32 kHz</td></tr>
<tr><td>6</td><td>44.1 kHz</td></tr>
<tr><td>7</td><td>48 kHz</td><td>48 kHz</td></tr>
</table></td></tr>
<tr><td>S589</td><td>12</td><td>0..22</td><td>Codec output gain level (index of gain table)</td></tr>
<tr><td>S590</td><td>12</td><td>0..22</td><td>Codec input gain level (index of gain table)</td></tr>
<tr><td>S689</td><td>-90</td><td>-450..215</td><td>Set codec output gain in dBr * 10
(applies to sink), default = 0</td></tr>
<tr><td>S690</td><td>-90</td><td>-450..215</td><td>Set codec input gain in dBr * 10
(applies to source), default = 0</td></tr>
<tr><td>S335</td><td>0</td><td>0..3</td><td>Set LED0 mode
<table>
<tr><td>0</td><td>LED_OFF</td></tr>
<tr><td>1</td><td>LED_ON</td></tr>
<tr><td>2</td><td>LED_PWM</td></tr>
<tr><td>3</td><td>LED_PULSE</td></tr>
</table></td></tr>
<tr><td>S336</td><td>2048</td><td>0..4095</td><td>Set LED 0 Duty Cycle,
referenced if LED mode = LED_PWM or LED_PULSE</td></tr>
<tr><td>S337</td><td>0</td><td>0..15</td><td>Set LED 0 PWM Period, referenced if LED mode = LED_PWM</td></tr>
<tr><td>S338</td><td>5</td><td>0..15</td><td>Set LED 0 Pulse Rate, referenced if LED mode = LED_PULSE</td></tr>
</table>

Register	Default	Range	Description
S340	0	0..3	Set LED1 mode 0 LED_OFF 1 LED_ON 2 LED_PWM 3 LED_PULSE
S341	2048	0..4095	Set LED 1 Duty Cycle, referenced if LED mode = LED_PWM or LED_PULSE
S342	0	0..15	Set LED 1 PWM Period, referenced if LED mode = LED_PWM
S343	5	0..15	Set LED 1 Pulse Rate, referenced if LED mode = LED_PULSE

2.8.8 GPIO (General Purpose Input/Output)

On a BTM module a number of digital I/Os can be used for general purposes. To each GPIO pin an S-Register is assigned (S651 to S658) which allows configuration of GPIO settings and single pin read/write. Configuration of a GPIO comprises pin direction (input/output), inversion enable, notification enable, function mapping select and function mapping code/av_operation_id. Refer to Figure 2-40: BTM51x GPIO configuration register and Table 2-80 for details.

To enable single pin read/write mode, set S650 to 1. This disables write and read access to configuration bit fields but allows read/write access to the pin state flag (0x01). To enable configuration mode, set S650 to 0. This allows read/write access to the full GPIO configuration register, including the pin state flag.

All logical GPIO lines can be read/written in one atomic step by new S-Register 670 at any time.

When a GPIO is configured as input, a weak internal pull-up or pull-down resistance is enabled by default. A strong pull-up/down (strong bias) can be enabled by setting the appropriate flag for a GPIO in S669. The direction of the resistance (up/down) is determined by the pin state flag of the configuration register. For example, if DIR=0 and PS=1, the input pin status is pulled up, if DIR=0 and PS=0, then the input pin is pulled down. There are no resistors switched inside the chipset, but strong and weak bias are defined by particular currents on the input pin, refer to Table 2-80: BTM51x Bias Currents for GPIO inputs. Note that an external (real) pull-up/pull-down resistor can easily override the internal settings depending on its value. To see the effects of internal pull-up/down clearly, remove external resistors (e.g. on a development kit).

Table 2-80: BTM51x Bias Currents for GPIO inputs

	Min	Typical	Max	Unit
Strong pull-up	-100	-40	-10	µA
Strong pull-down	10	40	100	µA
Weak pull-up	-5.0	-1.0	-0.2	µA
Weak pull-down	0.2	1.0	5.0	µA

Some GPIOs can have an alternative function assigned. If the alternative function is enabled, the appropriate I/O Pin is no longer available as GPIO. Modem control functions (DCD, DSR, RI, DTR) are assigned to GPIO pins as outlined in Table 2-81. This assignment is fixed and configuration settings are read only. Hence, IO pins of modem control lines cannot be configured as other GPIOs. Wi-Fi coexistence functions are not used. Table 2-81 recommends which pins should be reserved when planning a new product with Wi-Fi coexistence support. Wi-Fi coexistence functions cannot be configured by S-Registers. Therefore, contact Laird if Wi-Fi coexistence is required.

The following table lists GPIOs and their alternative functions.

Table 2-81: BTM51x GPIO Alternative Functions

GPIO Pin (BTM510/511)	Alternative Function	
	Modem Control Line [(1)]	Wi-Fi Coexistence [(2)]
GPIO1	-	BT_Active [(3)]
GPIO2	DCD	-
GPIO3	DSR	-
GPIO4	RI	-
GPIO5	-	BT_State/BT_Priority
GPIO6	-	Wlan_Active
GPIO7	-	Rf_Active
GPIO8	DTR	-

(1) Alternate function for modem control lines is fixed.
A modem control line cannot function as GPIO.

(2) Recommended pin assignment, not configurable by S-Registers.
Contact Laird if coexistence is required.

(3) BT_Active = RxEnable OR TxEnable

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	function mapping code / av_operation_id								reserved		FMS		NEN	INV	DIR	PS
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2-40: BTM51x GPIO configuration register

Table 2-82: BTM51x GPIO Configuration Register Field Descriptions

Bit Field	Description
0 – PS	I/O Pin State – returns the current state of the I/O pin (0/1) when read. A write affects the I/0 pin directly if DIR=1 and FMS=0 (FMS_NONE)
1 – DIR	Direction – controls if the I/O pin is an input or an output. 0 Input 1 Output
2 – INV	Inversion – controls if the I/O pin is inverted or not. Applies to both pin directions (read and write). 0 Not inverted 1 Inverted
3 – NEN	Notification enable – sends a notification to the host via UART on each status change if the GPIO is configured as input. If changed, AT&W and ATZ are required for the new setting to become effective. Notification format: \r\nGPIOx,y\r\n With x = GPIO index [1..8], y = new pin state [0..1] with INV flag applied 0 Disable notification 1 Enable notification

Bit Field	Description
[4:5] – FMS	Function Mapping Select – Select function mapping code (FMC) or av_operation_id (see Table 2-28). FMC or av_operation_id is set in bit fields [8:15]. 0 | FMS_NONE | Function mapping disabled 1 | FMS_MODEMCTRL | Use GPIO as modem control line (Read only, Table 2-81) 2 | FMS_AVRCP_OP | Bits [8:15] specify AVRCP operation ID (Table 2-28) 3 | FMS_MAPCODE | Bits [8:15] specify Function Mapping Code (Table 2-84 and Table 2-85)
[6:7]	Reserved
[8:15] function mapping code / av_operation_id	A Function Mapping Code (FMC, (Table 2-84 and Table 2-85) or an av_operation_id (Table 2-28) is set in this bit field. The pin is mapped to a profile specific function (A2DP,HFP) or to an AVRCP remote control operation and it carries out the assigned operation. A mapped function does not affect any other flag of the GPIO configuration register. So, e.g. DIR and INV must be set manually according to the selected function and hardware requirements. Valid only if FMS>=2. FMS selects function mapping code or av_operation_id. See also Table 2-53: BTM51x GPIO S Registers.

Table 2-83: BTM51x GPIO S Registers

Register	GPIO	Default	Range	Comment
S650		0	0..1	Mode for GPIO Config Registers: 0 | No mask 1 | Enable I/O pin state mask
S651	GPIO1	Depending on alternative GPIO usage and wiring	0..0xFFFF	GPIO Configuration Registers S650 must be set to 0 to enable configuration access. Configures: pin state, pin direction, pin inversion, notification enable, function mapping select and function mapping code / av_operation_id. See Table 2-82.
S652	GPIO2			
S653	GPIO3			
S654	GPIO4			
S655	GPIO5			
S656	GPIO6			
S657	GPIO7			
S658	GPIO8			
S669	GPIO1..8	0x0000	0..0xFF	Enable strong bias (=strong pull up / strong pull down) mask if GPIO is configured as input. 0x0001 | GPIO1 0x0002 | GPIO2 0x0004 | GPIO3 0x0008 | GPIO4 0x0010 | GPIO5 0x0020 | GPIO6 0x0040 | GPIO7 0x0080 | GPIO8

Register	GPIO	Default	Range	Comment
S670	GPIO1..8	0x0000	0..0xFF	Read/Write all GPIOs in one atomic step (Write operation only affects GPIOs configured as outputs) 0x0001 GPIO1 0x0002 GPIO2 0x0004 GPIO3 0x0008 GPIO4 0x0010 GPIO5 0x0020 GPIO6 0x0040 GPIO7 0x0080 GPIO8

2.8.8.1 GPIO function mapping codes (FMC)

By using function mapping codes, GPIOs may trigger certain functions by a push button where normally an AT command would be required. Examples include Volume Up/Down, answer a call (HFP), reject a call (HFP) etc. Also, you may use a GPIO for digital cable replacement. Refer to section 2.9.3. Table 2-84 and Table 2-85 both list available function mapping codes.

Examples for **input** FMCs (inversion enabled to work on BTM511 DVK-V04):

- Volume down multiple after short press, GPIO6, notification enabled:

  ATS656=$073C

- Volume up multiple after short press, GPIO7, notification disabled:

  ATS657=$0834

- Headset button (HS must be enabled), GPIO5, notification enabled:

  ATS655=$0B3C

- Button to toggle voice recognition activation from HF, GPIO1:
  ATS651=$123C
- Button to toggle voice recognition activation from HFG, GPIO1:
  ATSS651=$133C

Examples for **output** FMCs:

- Indicate status "AudioOn", GPIO7, notification enabled (e.g. control of external audio amplifier):

  ATS657=$053A

- Indicate status "Connected", GPIO7, notification disabled:

  ATS657=$0632

Subsequent AT&W and ATZ are required before new settings work as expected.

- Indicate voice recognition status for HF (FMC-out=7, GPIO7):
  ATS657=$073A
- Indicate voice recognition status for HFG (FMC-out=8. GPIO7):

ATS657=$083A

2.8.8.2 GPIO – AVRCP operation ID

Another GPIO option, particularly for AVRCP, is the assignment of AVRCP operation IDs (e.g. play, stop, pause etc.) to GPIOs. This allows a module in AVRCP controller role to send a pass through command triggered by a push button instead of an AT-command, and respectively, an AVRCP target can signalize incoming pass through commands on GPIO outputs.

If AVRCP functionality is not required in the application, it can alternatively serve as a digital IO cable replacement. As an advantage, it does not need an SPP link, and does not require a confirmation on the UART about the success or failure of a pass through command. See Cable Replacement for this purpose.

Table 2-84: BTM51x GPIO Function Mapping Codes - Input

Value	Function Mapping Code – Input	Comment
0x00	Cable Replacement TX	Cable replacement via SPP data
0x01	RFC_RTC_TX [(1)]	Cable replacement via RFC control signals
0x02	RFC_RTR_TX [(1)]	Cable replacement via RFC control signals
0x03	RFC_IC_TX [(1)]	Cable replacement via RFC control signals
0x04	RFC_DV_TX [(1)]	Cable replacement via RFC control signals
0x05	Volume down single step	Decrease output gain by one, see AT+GOD
0x06	Volume up single step	Increase output gain by one, see AT+GOU
0x07	Volume down multiple after short press	AT+GOD , S411
0x08	Volume up multiple after short press	AT+GOU , S411
0x09	Volume down multiple after medium press	AT+GOD , S411 + S412
0x0A	Volume up multiple after medium press	AT+GOU , S411 + S412
0x0B	Headset Button	AT+HSB, Headset profile (HS-unit)
0x0C	HF green button	AT+HFCA, accept incoming calls (HF-unit)
0x0D	HF red button	AT+HFCH, reject incoming call, hang up call (HF-unit)
0x0E	SSP YES button	Equivalent to command AT+BTBY, positive confirmation of pairing request
0x0F	SSP "NO" button	Equivalent to command AT+BTBN, negative confirmation or cancellation of pairing request
0x10	Cancel inquiry	Allows immediate cancellation of a running inquiry by GPIO / button
0x11	Enter discoverable/connectable state	A time window can be set by S371 in seconds. Refer to section 2.7.4.7.
0x12	Toggle voice recognition activation	Equivalent to AT+HFVT (HF)
0x13	Toggle voice recognition activation	Equivalent to AT+HFGVT (HFG)

Value	Function Mapping Code – Input	Comment
0x14	HF green button with redial	Answer incoming call by pressing the button. If no call is incoming and the button is pressed, then re-dial the last number. Parameter S389: 0: re-dial immediately 1: re-dial after short button press (S411, default=0.4s [ATI411]) 2: re-dial after medium button press (S411+S412, default=2.9s [ATI412]) 3: re-dial after long button press (S411+S412+S413, default=4.9s [ATI413])
(1) currently (v22.2.5.0) not available, reserved for future		

Table 2-85: BTM51x GPIO Function Mapping Codes - Output

Value	Function Mapping Code – Output	Comment
0x00	Cable Replacement RX	Cable replacement via SPP data
0x01	RFC_RTC_RX (1)	Cable replacement via RFC control signals
0x02	RFC_RTR_RX (1)	Cable replacement via RFC control signals
0x03	RFC_IC_RX (1)	Cable replacement via RFC control signals
0x04	RFC_DV_RX (1)	Cable replacement via RFC control signals
0x05	Indicate status "Audio On"	Active whenever the audio circuit is enabled
0x06	Indicate status "Connected"	Active whenever at least one profile is connected
0x07	Indicate voice recognition status	HF
0x08	Indicate voice recognition status	HFG
(1) currently (v22.2.5.0) not available, reserved for future		

2.9 Miscellaneous

2.9.1 CVC - Noise Reduction and Echo Cancellation

Clear Voice Capture (CVC) is a technology provided by Cambridge Silicon Radio (CSR) which utilises the BC05 internal DSP in order to improve the audio quality of SCO links (speech). This includes noise reduction and echo cancellation.

CVC is supported by BTM5xx from firmware version 18.1.2.3 onwards.

CSR provide different CVC images for different usage scenarios, which are incorporated in the firmware:

- Headset, 1 microphone
- Headset, 2 microphones
- Hands-free (1 microphone)

AT commands, S-Registers and asynchronous messages for CVC support are described in the application note CVC on BTM5xx [13], available from Laird Technologies under NDA.

2.9.1.1 CVC 7th generation

CVC algorithms in firmware v22.2.5.0 have been upgraded to version '7th generation', which comes with CSR ADK2.5.1. For parameter tuning, UFE of ADK2.5.1 is required.

7th generation of CVC works with SPI2USB interface when tuning CVC parameters using UFE. For previous CVC versions the only option for tuning parameters was an LPT parallel interface on PC side.

Still there is a known issue with garbling/scrambling sound in the send path for the WBS variants.

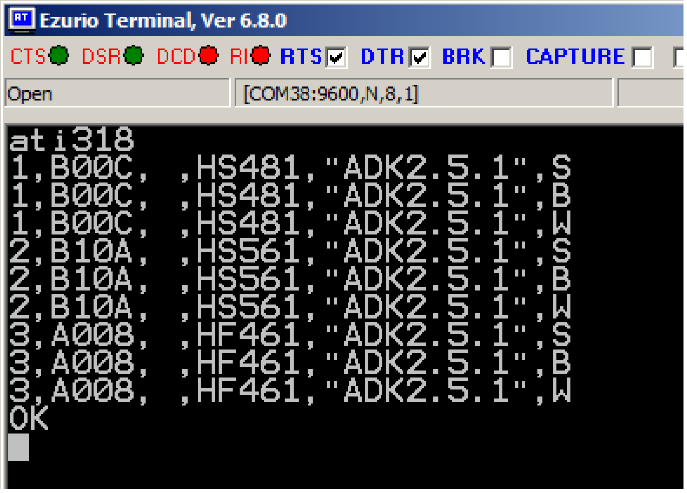
For each of the three CVC flavours (headset 1mic "HS1M", headset 2 mic "HS2M", hands-free "HF1M") a variant exists for narrowband (8kHz) and wide band (16kHz). The 8kHz variant is referred to as "standard" / "STD", the wide band variant as "wide band speech" /"WBS".

In addition, a third variant exists, the so called 'BEX' or 'bandwidth expanded'. This variant is intended for 8kHz air rate with CVC processing at 16 kHz. Therefore, a sampling rate conversion takes place between the 8kHz air interface and 16 kHz CVC processing. The CVC processing rate of 16 kHz is also the sampling rate for the selected audio interface (S315). It has not been evaluated how far BEX provides benefits in terms of audio quality or similar. It is provided in the hope to be useful and can be enabled by S377=1. It will then only be loaded for a narrow band (8kHz) SCO link (WBS disabled). However, BEX breaks the approach of controlling fixed sampling rate by S419 and therefore presents some kind of exception.

ATI318 and ATI35 allow to check availability of all CVC images. All images (flavours and its variants) are being loaded to the DSP and the outcome is printed to the UART. For ATI318, the outcome is printed in detail for each single image, nine in total. Variants are identified by an appended uppercase character:

S	Standard (narrow band)
B	BEX
W	Wide band speech (WBS)

Example for ATI318:


Ezurio Terminal, Ver 6.8.0
CTS● DSR● DCD● RI● RTS☑ DTR☑ BRK□ CAPTURE□
Open
[COM38:9600,N,8,1]

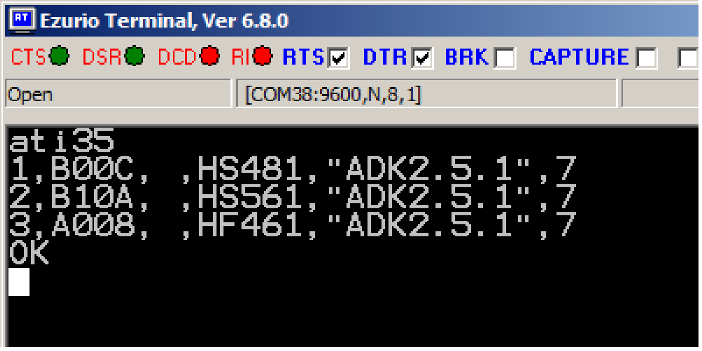
```
ati318
1,B00C, ,HS481,"ADK2.5.1",S
1,B00C, ,HS481,"ADK2.5.1",B
1,B00C, ,HS481,"ADK2.5.1",W
2,B10A, ,HS561,"ADK2.5.1",S
2,B10A, ,HS561,"ADK2.5.1",B
2,B10A, ,HS561,"ADK2.5.1",W
3,A008, ,HF461,"ADK2.5.1",S
3,A008, ,HF461,"ADK2.5.1",B
3,A008, ,HF461,"ADK2.5.1",W
OK
```

ATI35, in contrast, is a compact presentation of available CVC images. A group of variants is summarised in one line. An extra decimal number appended to each line allows to verify that each variant of a group/flavour has been loaded and released successfully, represented by its appropriate flag:

Bit0 =STD (01d); Bit1=BEX (02d); Bit2=WBS (04d)

So, in order to verify success, the extra number should always be ',7'

Example for ATI35:


Ezurio Terminal, Ver 6.8.0
CTS● DSR● DCD● RI● RTS☑ DTR☑ BRK□ CAPTURE□
Open
[COM38:9600,N,8,1]

```
ati35
1,B00C, ,HS481,"ADK2.5.1",7
2,B10A, ,HS561,"ADK2.5.1",7
3,A008, ,HF461,"ADK2.5.1",7
OK
```

For more information on CVC refer to the application note "CVC on BTM51x".

2.9.2 SCO / eSCO Audio Link (SPP-SCO)

BTM modules provide an AT command to establish an SCO / eSCO audio connection between a pair of BTM modules (or BISM2). This enables the user to create bidirectional audio links independently from a particular Bluetooth profile. The only prerequisite is the existence of a Rfcomm link (serial port profile, SPP) between the modules. If the SPP link doesn't exist, it can be created using AT+SPD<BdAddr>. Refer to SPP (Serial Port Profile).

A SCO/eSCO link is intended for bidirectional transmission of speech. The sampling rate is normally fixed to 8 kHz, meaning a usable bandwidth of 3.5 kHz. An exception to this is WBS (wideband speech), using a sampling rate of 16 kHz and therefore leading to a usable audio bandwidth of up to 7.5 kHz.

For SCO there are 3 packet types defined in the Bluetooth specification [1]: HV1, HV2, HV3. Each occupies one slot. They differ in the level of bit error checking. Laird recommends you enable all three packet types for SCO links. This passes the final decision down to the baseband. There is no retransmission of erroneous SCO packets.

For eSCO and basic data rate, there are 3 packet types defined in the Bluetooth specification [1]: EV3, EV4, EV5. EV3 occupies one slot, EV4 and EV5 can occupy up to three slots each. They differ in the level of bit error checking. It is recommended to enable all three packet types for eSCO links. This passes the final decision down to the baseband. eSCO packets involve a CRC code and retransmission of erroneous eSCO packets.

Packet types and link types (SCO or eSCO) are negotiated on link setup. A BTM can accept either incoming SCO (S584=0) or eSCO links (S584=1), but not both SCO and eSCO at one time*. If the initiating side requests an unsupported link type, the audio link fails. The initiating BTM module is supposed to request the remaining link type in that case.

*) As of f/w v22.2.5.0, S584=2 allows to accept an incoming SPP-SCO request, regardless of the SCO/eSCO packet type being requested by the peer device. A new request from the peer device is not needed because the first attempt should be successful.

Likewise, as of f/w v22.2.5.0, SPP-SCO can work with 16kHz sampling frequency, similar to WBS (wideband speech). Relevant commands work as follows:

AT+BTA2000…2007: The range 2000 to 2007 indicates that 16kHz/WBS shall be used. On the remote device, S376 must be enabled (=1). Otherwise the audio attempt will fail (no audio)

AT+BTA8: this command queries internally S-registers to decide which SCO link type (SCO/eSCO) and packets to request. Whereas in previous versions S584 was queried to determine if all eSCO packets should be requested (S584>0), now S376 is queried first. This means if S376=1, eSCO will be requested and WBS codec will be loaded. If S376=0 and S584>0, then all eSCO packets will be requested and the narrowband DSP application will be loaded when the audio link is established. If S584=0, then just SCO packets will be requested.

Please note that the result may be pure noise if this is feature is used with another device which doesn’t implement this assumption. Table 2-86: All AT commands and S-Registers for SCO/eSCO links.

Check link type of current SCO connection (SCO/eSCO)	ATI37	0 = unknown/no SCO link 1 = SCO 2 = eSCO

Table 2-87: SCO/eSCO AT-commands and S-Registers

Task	AT-Command/ S-Register	Comment
Initiate SCO link	AT+BTAx	x = packet type bitmask, recommended = 7 1 HV1 2 HV2 4 HV3
Initiate eSCO link	AT+BTA100x	x = packet type bitmask, recommended = 7 1 EV3 2 EV4 4 EV5
Initiate eSCO link with 16kHz sampling rate	AT+BTA200x	x = packet type bitmask, recommended = 7

Task	AT-Command/ S-Register	Comment
(similar WBS, only between two BTM51x)		1 EV3 2 EV4 4 EV5
Release SCO/eSCO link	AT+BTA0/ AT+BTA	
Initiate SCO/eSCO link	AT+BTA8	Link type (SCO/eSCO) and packet types defined by S584.
Enable WBS/16kHz, applies also to incoming SPP-SCO requests	S376 [0..1]	0 Disable WBS (8kHz) 1 Enable WBS (16kHz, default)
Set SCO / eSCO link type for incoming requests and for AT+BTA8	S584 [0..2]	0 SCO (HV1, HV2, HV3) enabled 1 eSCO (EV3, EV4, EV5) enabled 2 SPP-SCO, incoming audio request: accept requested packet type (SCO/eSCO) Only one link type can be enabled at one time.
Initiate SCO/eSCO link automatically on each SPP link	S532 [0..7]	The recommended value to enable this feature is 7. Value = bitmask for packet type. The link type (SCO/eSCO) is defined by S584. 0 Feature disabled 1 HV1 (S584 = 0) or EV3 (S584 = 1) 2 HV2 (S584 = 0) or EV4 (S584 = 1) 3 HV3 (S584 = 0) or EV5 (S584 = 1)
Check link type of current SCO connection (SCO/eSCO)	ATI37	0 = unknown / no SCO link 1 = SCO 2 = eSCO

2.9.2.1 SCO /eSCO Asynchronous Messages

The following asynchronous messages apply to SCO/eSCO connections:

FS8000,INT

The internal codec is configured for a sampling frequency of 8000 Hz (8 kHz) if narrowband/CVSD is used.

FS16000,INT

The internal codec is configured for a sampling frequency of 16000 Hz (16 kHz) when WBS is used.

SPP”AU1”

This response is sent to the host when a SCO/eSCO channel has been established. To check the actual type, use ATI37 (0=unknown, 1=SCO, 2=eSCO)

SPP”AU0”

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

SPP"AU2"

This response is sent to the host when a SCO channel setup fails. This might be caused by the fact that the peer only accepts eSCO connections but a SCO connection was requested or vice versa. Try to initiate the SCO connection with the remaining link type.

2.9.2.2 SCO / eSCO more information

More information on SCO can be found in the following sections of this document:

BTM51x: PCM – Pulse code modulation interface /SCO Input/Output Channel

2.9.3 Cable Replacement

There are different types and terms for cable replacement scenarios.   A distinction exists between serial cable replacement and digital I/O cable replacement.

- In serial cable replacement a serial (RS232) cable is replaced by a wireless SPP link. In particular, this context refers to TxD and RxD lines, not to handshaking or modem control signals. A stream of serial data is transmitted typically bidirectional with UARTs involved at both ends.
- In digital I/O cable replacement, one or more wire(s), each transmitting a binary signal in one direction, is replaced by a wireless link.
- In mixed cable replacement, digital I/O data and serial data is being transmitted simultaneously. This is needed to replace a real world RS232 cable by a wireless link. Handshaking signals (RTS, CTS) and modem control signals (DSR, DTR, DCD, RI), i.e. I/O-data, need to be sent and received at the same time that serial data (TxD, RxD) is being transmitted over the wireless link. The RFCOMM specification 0 (basis for SPP) caters for this by defining the appropriate procedures. In addition it assigns modem control signals to corresponding TS 07.10 Signals (RTC, RTR, IC, DV).

A BTM51x module allows creation of digital I/O cable replacement in a number of different ways. The following sections describe the characteristics of available options.

2.9.3.1 Digital Cable Replacement by GPIO-FMC

In this digital I/O cable replacement variant, a GPIO x is configured as input with the function mapping code (FMC) = Cable Replacement TX (0x00) and the corresponding GPIO x on the peer device is configured as output with FMC = Cable Replacement RX (0x00). An SPP connection is required to transmit status changes of GPIO inputs to the peer device. For this purpose S531 must be set to 4 before creating the SPP link and S506 must be 0. The SPP connection cannot be used for serial data, and the module remains in local command mode. The GPIO index must be within 1..12. There is no confirmation available if a status change was transmitted successfully to the peer. Ensure that the devices are not at the end of the maximum distance range. The direction of several GPIOs can be mixed, but one GPIO is always unidirectional. For example, GPIO1 direction is input on device A and output on device B. Then GPIO2 could be output on device A and input on device B.

Note: Due to inherent latency of Bluetooth transmission, expect the change of a state to be delayed. This value is typically 100ms and can be much more if the link quality is bad and requires many retries.

GPIOs mapped to the alternative function *modem control* cannot be used for this variant of digital I/O cable replacement. See Digital Cable Replacement by RFC-Modem Control Signals.

2.9.3.2 Digital Cable Replacement by GPIO-AVRCP

This digital I/O cable replacement variant uses the AVRCP profile. It is suitable if all GPIOs have the same direction and if AVRCP is not used for its primary purpose or with a non-BTM5xx module. GPIOs on the AVRCP controller are configured as inputs and GPIOs on the AVRCP target are configured as outputs. An AVRCP operation ID is selected on the controller side and on target side for each GPIO. This allows a free assignment of inputs and outputs, not fixed to the index of a GPIO pin. Even identical AVRCP operation IDs are allowed for different GPIOs, enabling e.g. one GPIO input on the controller to drive two or more outputs on the target or vice versa. An SPP link is not required but an AVRCP link is. Furthermore, it may be confirmed on the UART if a transition was successfully received by the target. See the AVRCP section for details. If an additional SPP connection is required for serial data transmission beside the GPIO, this is possible with the GPIO-AVRCP approach.

GPIOs mapped to the alternative function *modem control* cannot be used for this variant of digital I/O cable replacement. See Digital Cable Replacement by RFC-Modem Control Signals.

2.9.3.3 Digital Cable Replacement by RFC-Modem Control Signals

This digital I/O cable replacement variant uses signals defined in the RFCOMM specification 0 (RTC, RTR, IC, DV). The primary purpose of these signals is the transmission of digital I/O signals simultaneously with serial data over the same wireless link. These digital I/O signals can be categorized as handshaking signals (managing send and receive buffers of serial data: RTS, CTS) and modem control signals (managing the status of a modem: DTR, DSR, DCD, RI). The handshaking signals should not be touched, but modem control signals can be used for digital I/O cable replacement if not otherwise needed. In fact, modem control signals already control or signal many states on a BTM module (particularly in SPP connections). These uses include changing from connected to command mode, dropping an SPP link by DSR or indicating an SPP connection by DCD (see S507). If features like these are not needed for the application, RFC control signals can be used for cable replacement.

Configuration takes place by S551 and S552. Refer to the S Register reference table in 3.1. An SPP link is required for this to be working, and the SPP link can still transmit serial data.

2.9.3.4 Digital Cable Replacement by mixed variants

The fixed mapping of modem control lines to GPIOs reduces the number of free configurable GPIOs by four. But it is possible to mix the RFC approach with GPIO-AVRCP or GPIO-FMC to achieve a higher number of GPIOs to be utilized for digital cable replacement.

2.9.3.5 Pure Cable Replacement

The term *pure cable replacement*, used in this document, refers to automatic creation of an SPP link to a pre-defined peer device whenever the local module is powered up. Refer to the AT+BTR command.

2.9.4 Link Key Management

On a BTM device, link keys are managed by the AT firmware. Appropriate AT commands are described in AT Commands Managing Trusted Devices. There is a range of S Registers defining the behaviour of automatic link key storage on incoming/outgoing and dedicated/general bonding.

2.9.4.1 Dedicated Bonding

In BT2.1 specification, *dedicated bonding* is defined as the exchange of link keys between two devices without the intention of establishing a connection immediately.

Dedicated bonding is initiated by AT+BTW<BdAddr> (initiation of pairing).

2.9.4.2 General Bonding

In BT2.1 specification, *general bonding* is defined as the exchange of link keys between two devices with the intention of establishing a connection immediately. This is the case if a device tries to connect to another device without existing link key. Hence, pairing (authentication and exchange of link keys) is initiated automatically prior to the connection.

General bonding is initiated by a connection-requesting AT command if there is no link key for the peer device existing. Such AT commands are:

> AT+SPD<BdAddr>, AT+APD<BdAddr>, AT+AVD<BdAddr>, AT+HSD<BdAddr>, AT+HSGD<BdAddr>, AT+HFD<BdAddr>, AT+HFGD<BdAddr>, AT+DUD<BdAddr>

2.9.4.3 Automatic storage of link keys

Four S Registers define the automatic storage of link keys in the trusted device list, depending on incoming/outgoing and general/dedicated bonding. See Table 2-88: Automatic storage of link keys.

Table 2-88: Automatic storage of link keys

Task	S-Register	Comment	
Automatic link key storage on dedicated bonding outgoing (DBO)	S325 [0..1]	0	Do not store (cache only)
		1	Store automatically (default)
		Identical with S538	
Automatic link key storage on general bonding outgoing (GBO)	S326 [0..1]	0	Do not store (cache only)
		1	Store automatically (default)
Automatic link key storage on dedicated bonding incoming (DBI)	S327 [0..1]	0	Do not store (cache only)
		1	Store automatically (default)
Automatic link key storage on general bonding incoming (GBI)	S328 [0..1]	0	Do not store (cache only)
		1	Store automatically (default)

2.9.5 Profile Connection Status

The connection status of a profile can be queried by an ATI-Command. This might be helpful in order to decide whether to disconnect all connected profiles (via ATH*) or a specific one. For details see Table 2-89.

Table 2-89: Profile connection status

Task	AT Command	Comment	
Get connection status of SPP	ATI60	0	Not connected
		1	Connected (local command mode)
		2	Connected (remote command mode)
		Identical with ATI9	
Get connection status of A2DP	ATI61	0	Not connected
		1	Connected and streaming
		2	Connected but not streaming (suspended)
Get connection status of AVRCP	ATI62	0	Not connected
		1	Connected

Task	AT Command	Comment	
Get connection status of HSP-Headset	ATI63	0	Not connected
		1	ACL connected
		2	Audio connected
Get connection status of HSP-AG	ATI64	0	Not connected
		1	ACL connected
		2	Audio connected
Get connection status of HFP-HF	ATI65	0	Not connected
		1	SLC connected
		2	Audio connected
		3	In call, SLC
		4	In call, audio
		5	In call setup (incoming/dialling/alerting) – SLC
		6	In call setup (incoming/dialling/alerting) – audio
Get connection status of HFP-AG	ATI66	0	Not connected
		1	SLC connected
		2	Audio connected
		3	In call, SLC
		4	In call, audio
		5	In call setup (incoming/dialling/alerting) – SLC
		6	In call setup (incoming/dialling/alerting) – audio
		8	In call but no SLC

2.9.6 Disconnecting Profiles

A connection to a profile can be released by ATH<Profile-UUID>. For A2DP and AVRCP this is a second way of disconnecting

The response on a disconnect command is usually NO CARRIER <profileUUID> if a connection has existed and S329=0. If no connection has existed and S329=0, no profileUUID is appended.

If all connections are to be released, ATH* may be used. See Table 2-90 (Profile Release Commands).

Table 2-90: Profile release commands

Task	AT-Command	Comment
Disconnect SPP	ATH1101 or AT+SPH or ATH	Single ATH retained for backward compatibility, response NO CARRIER or NO CARRIER 1101 depending on S329 and if a SPP connection has existed previously.
Disconnect A2DP	ATH110D or AT+APH	If A2DP connection released: response = NO CARRIER 110D; If no A2DP connection has existed: response = NO CARRIER
Disconnect AVRCP	ATH110E or AT+AVH	If AVRCP connection released: response = NO CARRIER 110E; If no AVRCP connection has existed: response = NO CARRIER

Task	AT-Command	Comment
Disconnect HSG	ATH1112 or AT+HSGH	If AG(HSP) connection released: response = NO CARRIER 1112; If no HSP connection has existed: response = NO CARRIER
Disconnect HS	ATH1108 or AT+HSH	Must be enabled by S332, otherwise it results in a behaviour not defined in HSP specification. If HS(HSP) connection released: response = NO CARRIER 1108; If no HSP connection has existed: response = NO CARRIER
Disconnect HFG	ATH111F or AT+HFGH	If AG(HFP) connection released: response = NO CARRIER 111F; If no HSP connection has existed: response = NO CARRIER
Disconnect HF	ATH111E or AT+HFH	If HF(HFP) connection released: response = NO CARRIER 111E; If no HSP connection has existed: response = NO CARRIER
Disconnect all profiles listed in this table	ATH*	**Response:** NO CARRIER <ProfileUUID> for each previously connected profile or NO CARRIER if no existing connection found or HS connected but S332=0

2.9.7 Legacy Response Format (BISM2)

Some BISM2 responses have been slightly changed on BTM modules to provide enhanced functionality. If required, a BISM2-compatible response format can be enabled by S Register 329. Table 2-91 and Table 2-92 shows the implications of enabled/disabled legacy response format.

Table 2-91: Enabling/Disabling legacy response format

Task	S-Register	Comment	
Enable legacy response format (BISM2 compatible)	S329 [0..1]	0	Disabled (default)
		1	Enabled

Table 2-92: Implications of S329

Command	Legacy response enabled (S329=1)	Legacy response format disabled (S329=0)
AT+SPH;ATH1101	Response: NO CARRIER	If SPP was connected, response = NO CARRIER 1101 If SPP was not connected, response =NO CARRIER
AT+APH;ATH110D	Response: NO CARRIER	If A2DP was connected, response = NO CARRIER 110D If A2DP was not connected, response =NO CARRIER
AT+AVH;ATH110E	Response: NO CARRIER	If AVRCP was connected, response = NO CARRIER 110E If AVRCP was not connected, response =NO CARRIER
AT+HSH;ATH1108	Response: NO CARRIER	If HS instance was connected, response = NO CARRIER 110E If HS instance was not connected, response = NO CARRIER
AT+HSGH;ATH1112	Response: NO CARRIER	If HSG instance was connected, response = NO CARRIER 1112 If HSG instance was not connected, response = NO CARRIER

Command	Legacy response enabled (S329=1)	Legacy response format disabled (S329=0)
AT+HFH;ATH111E	Response: NO CARRIER	If HF instance was connected, response = NO CARRIER 111E If HF instance was not connected, response = NO CARRIER
AT+HFGH;ATH111F	Response: NO CARRIER	If HFG instance was connected, response = NO CARRIER 111F If HFG instance was not connected, response = NO CARRIER

2.9.8 UUIDs in CONNECT/NO CARRIER Messages

In profiles where functionality and command set differs between both possible roles (asynchronous profiles), role-indicating UUIDs are used in the "CONNECT" and "NO CARRIER" messages. HSP and HFP are asynchronous profiles.

A2DP and AVRCP are treated as synchronous profiles because only one profile instance with one role selected can be initialised currently. Therefore a common UUID is used indicating the profile only but not the role.

In a message "CONNECT <bd_addr$_{hex}$>,<UUID>", UUID is the profile/role the connection is made **to**. Hence for an outgoing connection, UUID indicates the remote device's profile/role. If it is an incoming connection, UUID indicates the local device's profile/role UUID. Hence, role-indicating UUIDs presented in a "CONNECT" message are equal on both ends of a connected pair.

In a message "NO CARRIER <UUID>", UUID represents the **local** profile/role **UUID**. Hence for an asynchronous profile, role-indicating UUIDs differ on both ends of a previously connected pair.

Table 2-93 provides an overview of currently used UUIDs on BTM devices.

Table 2-93: UUIDs used in BTM5xx CONNECT/NO CARRIER messages

UUID (hex)	Profile/Role	Role indication	Comment
1101	SPP	No	Synchronous profile
1108 (1)	HSP – HS	Yes	Headset role (Headset profile)
110D	A2DP	No	Common UUID for source and sink role
110E	AVRCP	No	Common UUID for controller and target role
1112	HSP – AG	Yes	Audio gateway role (Headset profile)
111E	HFP – HF	Yes	Hands-free role (Hands-free profile)
111F	HFP – AG	Yes	Audio gateway role (Hands-free profile)

There is an option to append a direction indicator to a connect message. This enables the host to distinguish an incoming connection from an outgoing connection based on the "CONNECT" message. The direction indicator **only** applies to **role indicating UUIDs** if **S329=0**. The direction indicator is configured by S register 331; refer to Table 2-94.

Table 2-94: Examples for direction indication in "CONNECT" messages

S331	Meaning	Examples	
		Incoming connection	Outgoing connection
0	Disabled	CONNECT	CONNECT 0123456789AB,1108

		0123456789AB,1108	
1	Character style, I or O	CONNECT 0123456789AB,1108,I	CONNECT 0123456789AB,1108,O
2 (default)	Symbol style, < or >	CONNECT 0123456789AB,1108,<	CONNECT 0123456789AB,1108,>

2.9.9 UUIDs in Service Records of HSP

This section provides some background information on UUIDs and their usage in service record of the Headset profile.

In the Bluetooth Assigned Numbers document [8] in the table Service classes, there are three UUIDs assigned for the HSP:

Table 2-95: Bluetooth Assigned Numbers for HSP

Mnemonic	UUID size	UUID	Profile [Ref #5]
HSP	uuid16	0x1108	See Headset Profile, Bluetooth SIG
Headset - AG	uuid16	0x1112	See Headset Profile, Bluetooth SIG
Headset - HS		0x1131	*Bluetooth* Headset Profile, Bluetooth SIG

In the service record for HSP 1.2 - Headset role ([4] page 21), the following values are assigned:

Table 2-96: Selected items of HSP - headset role service record

Item	Value	UUID
ServiceClass0	Headset	0x1131
Profile0	Headset Profile	0x1108

In the service record for HSP 1.2 – Audio gateway role ([4], page 22), the following values are assigned:

Table 2-97: Selected items of HSP - audio gateway role service record

Item	Value	UUID
ServiceClass0	Headset Audio Gateway	0x1112
Profile0	Headset Profile	0x1108

The tables above show that UUID 0x1108 identifies the Headset **profile** whereas the **role** is explicitly identified by the ServiceClass0 UUID (0x1131=HS / 0x1112=HSG).

Although UUID 0x1108 appears in the service records of both Headset unit AND Audio Gateway, 0x1108 is still used as identifier for the Headset role in CONNECT and NO CARRIER messages.

In HSP1.0 - headset role, 0x1108 was assigned to both ServiceClass0 and Profile0, i.e. it appeared twice in the service record of a HS-unit and once in the service record of the audio gateway.

Predecessors of the BTM510/511 (e.g. BISM2) have used 0x1108 as identifier for the Headset role. This is inherited on BTM510/511 and is the reason why 0x1108 is still used in BTM510/511 CONNECT and NO CARRIER messages.

The logically correct UUID for the HS-role would be 0x1131.

When using the AT+BTIV<BdAddr>,UUID$_{hex}$ to query UUIDs of a remote device in order to determine supported profile and roles, the user must be aware of the circumstances described here. (E.g. do not assume the remote device supports the headset role if UUID 0x1108 is found in the service records. Check for UUID 0x1112).

2.9.10 Low Power Operation

In order to decrease power consumption, a couple of options should be considered:

- Page/inquiry scanning – When not connected but in page scanning (=connectable) or inquiry scanning (=discoverable) mode, then the scanning windows can be reduced and scan intervals can be increased in order to lower current consumption at the cost of probably longer time to find the device and longer time to initiate a connection.
- Entering a connection – When in a connection and not expecting any more incoming connections for any remaining profile, then page scanning (being connectable) and inquiry scanning (being discoverable) should be disabled. This leads to lower power and lower bandwidth consumption. As of this firmware version, BTM51x provides a feature to change the scanning modes automatically upon connection of one or all profiles and restoring scanning modes on disconnection of one or all profiles. See Auto-BTX / S308
- Sniff mode – when connected but no data being exchanged, then sniff mode helps to reduce power consumption at the cost of higher latency for the first data being transmitted after a longer idle time. In sniff mode, master and slave agree that the master will not poll the slave for a certain interval. Thus the slave does not have to enable its power hungry radio circuit in each possible slot. The same applies to the master, provided there are no other slaves on the piconet which he controls. Sniff mode is well suited for Rfcomm based profiles and is supported for SPP, HSP and HFP as of this firmware version (22.1.4.2).
- Sniff sub-rating (SSR) – This sub mode of sniff mode has been introduced with BT2.1/EDR specification. It can be understood as permission for the remote device to extend the current sniff interval by a whole numbered multiple. SSR is supported for BTM51x as of f/w version v22.2.5.0.
- Baud rate – even when no data being exchanged, a high baud rate corresponds to higher static current consumption than a comparable low baud rate. The cost of a lower baud rate is lower data throughput obviously.

2.9.10.1 Page Scan/Inquiry Scan Interval and Window

Page scanning means the module is connectable. Inquiry scanning means the module is discoverable. With the following S registers the power consumption of the BTM can be influenced at the cost of longer connection establishment time and longer time until a BTM is discovered.

The page scan window defines the time for the module to look out for incoming connection requests (paging). The inquiry scan window defines the time for the module to look out for incoming inquiry requests (device discovery). If the module is both connectable and discoverable (512=4 or AT+BTP issued), it mutually performs page scanning and inquiry scanning as shown in Figure 2-41. If connectable only, the module performs page scanning only (repeatedly) and if discoverable only, then the module performs page scanning only.

S register 508 defines the page scan interval in ms, range is [11..2250].
S register 509 defines the page scan window in ms, range is [11..2250].
S register 510 defines the inquiry scan interval in ms, range is [11..2250].
S register 511 defines the inquiry scan window in ms, range is [11..2250].

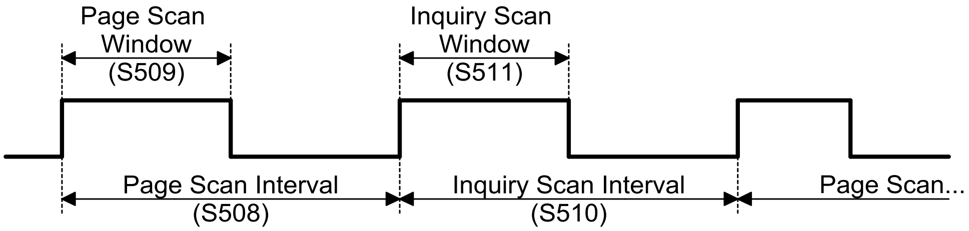

Page Scan Window (S509)
Inquiry Scan Window (S511)
Page Scan Interval (S508)
Inquiry Scan Interval (S510)
Page Scan...

Figure 2-41: Page and Inquiry Scan Intervals and Windows

2.9.10.2 Multi-profile Auto BTX (S308)

S308 has been extended by a multi profile policy. The term 'Auto-BTX' means the automatic change to not discoverable not connectable after entering a connection, and restoring scanning mode according to S512 when leaving a connection. The relevant AT-command for this purpose would be AT+BTX, this is where the term 'Auto-BTX' comes from.

In previous firmware versions, S308 had applied to A2DP only. Two more bits were added to S308 for defining a ONE/ALL multi profile policy:

S308:

Bit	3	2	1	0
	MPP		EXIT	ENTER
Default	1	0	1	1

Table 2-98: S308

Field	Description
0 – ENTER	When entering a connection: 0 = do not change discoverable/connectable mode 1 = make the module not discoverable and not connectable (default)
1 – EXIT	When leaving a connection: 0 = do not change discoverable/connectable mode 1 = restore discoverable/connectable mode according to S512 (default)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>[3:2] – MPP</td><td>Multi profile policy: controls in which cases ENTER/EXIT operations are carried out.
MPP=00b caters for backward compatibility, i.e. only A2DP operation is covered.
Default : 10b
<table>
<tr><th>MPP (bin)
[3:2]</th><th>Carry out ENTER operation when … profile(s) of S102 is/are connected</th><th>Carry out EXIT operation when … profile(s) is/are disconnected</th></tr>
<tr><td>00</td><td>Only A2DP</td><td>Only A2DP</td></tr>
<tr><td>01</td><td>ALL</td><td>ONE</td></tr>
<tr><td>10</td><td>ONE</td><td>ALL</td></tr>
<tr><td>11</td><td>ALL</td><td>ALL</td></tr>
</table>
</td></tr>
</table>

Note: The new default MPP of 10b makes BTM51x not discoverable not connectable as soon as one profile has connected. If more incoming connections are expected, S308 must be modified appropriately.

2.9.10.3 Sniff Mode

Bluetooth connections are master/slave in nature. A master sends packets and a slave has to acknowledge that packet in the next timeslot. Timeslots in Bluetooth are 625 microseconds wide. This implies that a master always knows when packets are to be sent and received, which further means it is able to optimise power usage by switching on power-hungry circuitry only when needed.

A slave does not have prior knowledge of when a packet is to be received and assumes a packet is scheduled to arrive for each receive slot. This means that it must sustain its receiving circuitry for most of the receive slot duration. The result is high power consumption for the slave. In general, a slave draws about 5 times the current of a master. This problem was identified very early in the evolution of Bluetooth (especially since headsets are always slaves in a Bluetooth connection) and it is solved with Sniff mode, with appropriate lower layer negotiating protocol.

Sniff mode during connection is basically an agreement between the slave and its master that data packets are only be exchanged for N timeslots every M slots. The slave can then assume that it will never be contacted during M-N slots, and so can switch its power hungry circuitry off. The specification further specifies a third parameter called ‘timeout’ (T) which specifies ‘extra’ timeslots that the slave agrees to listen for after receiving a valid data packet. If a data packet is received by the slave, then it knows that it MUST carry on listening for at least T more slots. If within that T slot time period another data packet is received, the timer is restarted. This ensures low idle power consumption at the expense of latency. When there is a lot of data to be transferred, it acts as if sniff mode were not enabled.

During sniff mode, a slave listens for N slots every M slots. The Bluetooth specification states that a master can have up to 7 slaves attached with all slaves having requested varying sniff parameters. It may be impossible to guarantee that each slave has its M parameter granted. In light of this, the protocol for enabling sniff mode specifies that a requesting peer specify the M parameter as a minimum and maximum value. This allows the master to interleave the sniff modes for all slaves attached.

For this reason, sniff parameters are specified in the BTM module via four S registers. S Register 561 specifies ‘N’, S Register 562 specifies ‘T’ and S Registers 563/564 specify minimum ‘M’ and maximum ‘M’ respectively.

Although the specification defines these parameters in terms of timeslots, the S register values must be specified in milliseconds and the firmware converts it to timeslots.

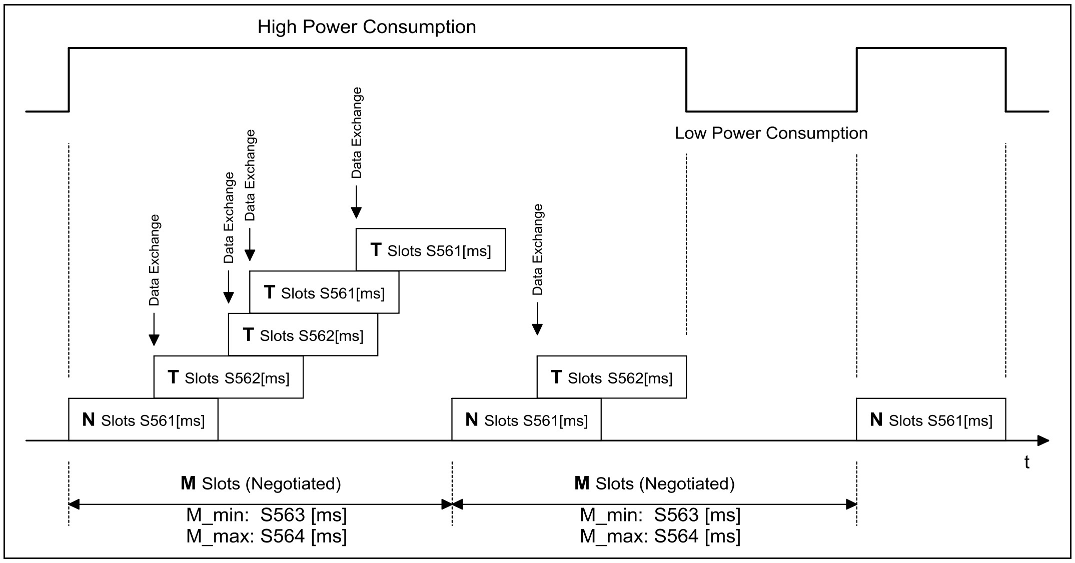


Figure 2-42: Sniff Mode Example

A low power application note exists for BTM41x, which explains plenty of details about sniff mode (SM) and sniff sub-rating (SSR). SM/SSR registers of BTM51x were designed to be identical to relevant BTM41x registers. BTM41x low power application note is highly recommended for reference. This section only covers the most important facts briefly.

In f/w version v22.2.5.0, sniff mode can be configured to be actively initiated for SPP, HSP and HFP. Previously, only passive mode was supported, i.e. remote requests for sniff mode were accepted but not actively requested.

When sniff mode is enabled, the same parameters will apply for all of the three profiles (SPP / HSP / HFP) and to all remote devices connected (if multiple). A differentiation on a per device basis is currently not supported. Sniff mode configuration S-Registers S561…S564 have been re-worked for v22.2.5.0, resulting in a mapping between milliseconds and slots as outlined in Table <**please insert cross reference**> below. When writing to S-Registers 561 to 564, begin with S564 (sniff interval) in order to prevent ERROR 037 (internal plausibility checker).

Further new S-registers related to sniff mode:

S364	Delayed Sniff mode, experimental. Refer to the BTM41x Low Power Application Note for additional information. It can be accessed from the Laird Embedded Wireless Support Site BTM41x product page under the Documentation tab.
S365	Set first row of LP_power_table active. Refer to the BTM41x Low Power Application Note for additional information. It can be accessed from the Laird Embedded Wireless Support Site BTM41x product page under the Documentation tab.

Table 2-99: Sniff register mapping S [561…564]

Write	Read	Number of	Actual	Write	Read	Number of	Actual
(ms)	(ms)	Slots	ms	(ms)	(ms)	Slots	ms

[0, 1]	0	0	0	[95, 149]	100	160	100
[2, 3]	2	4	2.5	[150, 249]	200	320	200
[4, 6]	5	8	5	[250, 349]	300	480	300
[7, 8]	7	12	7.5	[350, 449]	400	640	400
[9, 11]	10	16	10	[450, 549]	500	800	500
[12, 13]	12	20	12.5	[550, 649]	600	960	600
[14, 16]	15	24	15	[650, 749]	700	1120	700
[17, 18]	17	28	17.5	[750, 849]	800	1280	800
[19, 24]	20	32	20	[850, 949]	900	1440	900
[25, 34]	30	48	30	[950, 1000]	1000	1600	1000
[35, 44]	40	64	40				
[45, 54]	50	80	50				
[55, 64]	60	96	60				
[65, 74]	70	112	70				
[75, 84]	80	128	80				
[85, 94]	90	144	90				

2.9.10.4 Sniff Sub-rating (SSR)

Like for sniff mode, it is highly recommended to refer to BTM41x low power application note also for SSR.

Registers related to SSR:

S348: SSR (maximum) remote interval:

The maximum amount of time which the remote device is allowed to be absent in SSR mode,

in 0.1s, Range=[0..170], default=0

S349: SSR (minimum) remote timeout:

Minimum time with no data exchange in sniff mode before remote device is allowed to enter SSR

in 0.1s, Range=[0..170], default=0

S350: SSR (minimum) local timeout:

Minimum time with no data exchange in sniff mode before local device is allowed to enter SSR

in 0.1s, Range=[0..170], default=0

For S348…S350 refer to Table 2-100 for a mapping between register value, slots and seconds.

SSR parameters S348 and S349 can be understood as a permission to the remote device to be absent for whole numbered integer multiple of the current sniff interval. Thus it only benefits the remote device in terms of decreased power consumption. S350 gives control on the local SSR timeout if the remote device has given permission to enter SSR.

Refer to the BTM41x Low Power Application Note for additional information. It can be accessed from the Laird Embedded Wireless Support Site BTM41x product page under the Documentation tab.

Table 2-100: Sniff sub-rating (SSR) register mapping S [348...350]

Write (0.1 sec)	Read (0.1 sec)	Number of slots	Actual Seconds	Write (0.1 s)	Read (0.1 s)	Number of slots	Actual Seconds
[0]	0	0	0	[38,42]	40	6400	4
[1]	1	160	0,1	[43,47]	45	7200	4,5
[2]	2	320	0,2	[48,52]	50	8000	5
[3]	3	480	0,3	[53,57]	55	8800	5,5
[4]	4	640	0,4	[58,64]	60	9600	6
[5]	5	800	0,5	[65,74]	70	11200	7
[6]	6	960	0,6	[75,84]	80	12800	8
[7]	7	1120	0,7	[85,94]	90	14400	9
[8]	8	1280	0,8	[95,104]	100	16000	10
[9]	9	1440	0,9	[105,114]	110	17600	11
[10,12]	10	1600	1	[115,124]	120	19200	12
[13,17]	15	2400	1,5	[125,134]	130	20800	13
[18,22]	20	3200	2	[135,144]	140	22400	14
[23,27]	25	4000	2,5	[145,154]	150	24000	15
[28,32]	30	4800	3	[155,164]	160	25600	16
[33,37]	35	5600	3,5	[165,170]	170	27200	17

2.9.11 Maximum RF-Tx Power Level

S541 and S542 control the maximum RF transmit power level for all operation states (inquiring/connecting/in connection).

2.9.12 Manufacturing Info String

A string with manufacturing information can be retrieved by ATI200.

2.9.13 Bluetooth Version

The Bluetooth version can be queried by ATI18.

2.9.14 Legacy Issues (BT2.0)

There are some special cases if a legacy device (BT2.0 or earlier, e.g.BISM2) requests a connection to a module (BT2.1).

General bonding does not work if initiated by the legacy device. Instead, the legacy device must initiate dedicated bonding first (=pairing, BISM2: AT+BTW<BdAddr>). After successful pairing, the connection can be initiated by the legacy device (BISM2: ATD<BdAddr>).

2.9.15 Factory Default UART Baud Rate

BTM devices may operate at a very wide range of baud rates. S Registers 520 and 521 set the baud rate.

As long as BAUDRATE * 0.004096 produces an integer, there is 0% error in clocking for that baud rate.

It is possible to set a baud rate that a PC can't utilize, rendering it nearly incommunicable.

To cater for this, the module comes out of reset using 9600,N,8,1 comms settings for exactly 750 ms and then reverts to the communication parameters as per its S Registers.

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

If connected to a PC using Ezurio Terminal, the module can be reset to the factory default baud rate as follows:

Right click in the Ezurio Terminal window, then click Factory Default > Via BREAK/CMD @ 9600.

(Tested with version 6.7.2 of Ezurio Terminal)

2.9.16 RI-Dependent Start-up Mode

The UART_RI line can be configured as an input and on power up its state can force the device into one of two modes, defining discoverability and connectability. See description for S Registers 565 to 569 inclusive for more details.

For example, the feature may allow a device to make an outgoing connection if RI is in one state, and await an incoming connection in the other state.

2.9.17 Reset via BREAK

The module can be reset by sending a BREAK signal. A BREAK signal exists when the module's UART_RX input is in a non-idle state (0v) for more than 125 milliseconds.

Ezurio Terminal provides a BREAK capability which can reset a connected module by ticking and un-ticking the BRK field. See Figure 2-43.

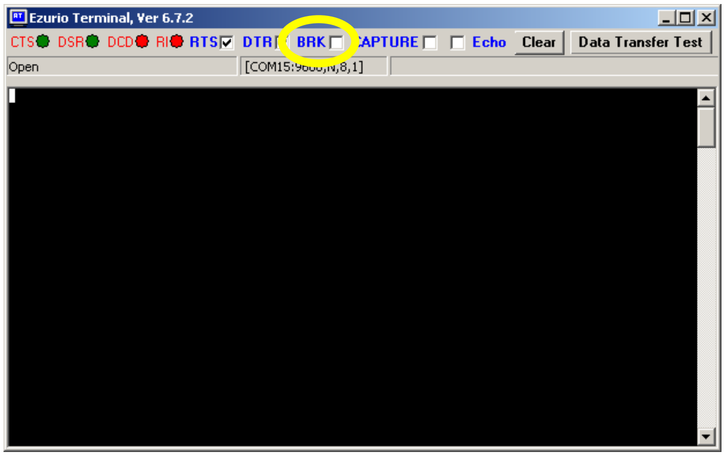

Ezurio Terminal, Ver 6.7.2
CTS ● DSR ● DCD ● RI ● RTS ☑ DTR □ BRK □ CAPTURE □ □ Echo
Clear
Data Transfer Test
Open
[COM15:9600,N,8,1]

Figure 2-43: BREAK capability in Ezurio Terminal

2.9.18 Append Bluetooth Address to Friendly Name

If S Reg 593 is set to 1, the last 6 hex digits of the Bluetooth address are automatically appended to the friendly name. This differentiates multiple devices with the same name in a neighbourhood.

2.9.19 Three Digit Error Responses

As of firmware version 18.1.4.0 (BTM51x-08), error codes are presented as 3 decimal digits with leading zeroes by default. For backward compatibility, clear bit 2 (0x04) of S559 to use legacy two-digit codes. In this case, ERROR codes >=100 are presented as ERROR 99 and ATI12 retrieves the actual error code.

2.9.20 ATI 54/55/56 {status information for all profiles}

As of firmware v18.1.4.0, ATI commands are added to query status information such as remote Bluetooth device address, RSSI (receiver signal strength indicator) and local HCI role for connected profiles:

- ATI54: print remote device's Bluetooth addresses of connected profiles

  Response (one line for per profile):

  ```
  $<profile_mask_4_hex_digits>,<BdAddr>
  with <BdAddr> = 12 hex-digit Bluetooth device address;
  ```

  Example:

  ```
  $0001,0016A40009C8      //connection of SPP profile with device B
  $0080,C09F42934E02      //connection of A2DP profile with device C
  $0100,C09F42934E02      //connection of AVRCP profile with device C
  ```

  If no profile is connected, response is: '0' (zero)

- ATI55: print RSSI (receiver strength indicator) for connected profiles

  Response (one line for per profile):

  ```
  $<profile_mask_4_hex_digits>,<RSSI>
  with <RSSI> = signed decimal value of RSSI, variable length
  ```

  Example:

  ```
  $0001,-11        //RSSI of device with SPP connection (device B)
  $0080,-4              //RSSI of device with A2DP connection (device C)
  $0100,-4              //RSSI of device with AVRCP connection (device C)
  ```

  If no profile connected, response is: '0' (zero)

- ATI56: print local HCI role for connected profiles

  Response (one line for per profile):

  ```
  $<profile_mask_4_hex_digits>,M     if master
  $<profile_mask_4_hex_digits>,S     if slave
  ```

  Example:

  ```
  $0001,M              //master in the link with device of SPP (device B)
  ```

```
$0080,S              //slave in the link with device of A2DP (device C)
$0100,S              //slave in the link with device of AVRCP (device C)
```

If no profile connected, response is: ‘0’ (zero)

2.9.21 Management of Persistent Store

Extensive use of the persistent store (non-volatile memory) can lead some AT commands or functions to unsuccessfully finish and causing an error message. This section explains the background of persistent store and suggests strategies to avoid these scenarios.

2.9.21.1 Persistent store characteristic

The persistent store of BTM51x is made of flash memory, which has the typical characteristic that a single bit can be written only once. For deletion of data, a larger area, a so called segment must be deleted in common. This means that all data of the segment would be lost. So in order to delete a small amount of data but retaining all other data of the segment, the data to be deleted is not actually deleted but is invalidated by internal flash memory pointers. Similarly, overwriting does not actually delete old data but stores the new data in the remaining space of the segment and declares the old location invalid by pointing to the new location. The flash segment fills up with each write (or delete) operation to persistent store. At some point the segment becomes full and write/delete operations fail, causing error messages (e.g. ERROR 011 or ERROR 102). The firmware has a built-in mechanism to recover from this state on a power cycle / reset (ATZ): If the remaining free space of the current segment is below a certain limit, the flash segment is defragmented and copied to a free segment. Due to this defragmentation, which clears out all invalidated data, free space in the segment becomes available. From now on, the new segment is being used for all operations (read/write/delete). Finally, the old segment is deleted to prepare for the next defragmentation/copy cycle.

2.9.21.2 AT+NVQ/F {commands to manage persistent store}

BTM51x firmware provides AT commands allowing management of persistent store:

- AT+NVQ query the remaining space in current segment
- AT+NVF flood the remaining space of current segment. On next power cycle (ATZ)a defragmentation cycle occurs.

2.9.21.3 Commands implying persistent store usage

The following operations and commands use persistent store:

- Write/delete auto connect records (AT+ACW, AT+ACRn, AT+ACDn, AT+ACD*)
- Write dynamic registers (S744=…, S745=…)
- Write EIR data (AT+BTEW)
- Save S-Registers (AT&W)
- …

2.9.21.4 Strategy to prevent persistent store write errors

If an application makes extensive use of persistent store (PS) operations, then the PS consumption of the application should be considered and evaluated.

AT+NVQ helps to analyse the consumption of each relevant operation by querying the free space before and after. It also helps monitor the free space over a longer application period by polling the NVQ value regularly. This should give an idea on the persistent store consumption of an application.

Persistent memory is automatically defragmented on a reset if the NVQ value is less than 300. Although this seems to work well, an additional AT+NVF is always recommended before a reset for the purpose of defragmentation as a precaution.

To prevent write operation errors (e.g. ERROR 011 or ERROR 102) Laird recommends you first analyse the persistent store consumption of the application. If in the course of the application it is likely that no reset will occur over long time and that the NVQ value will decrement down to a critical level, then the host controller should foresee flooding (AT+NVF) and reset (ATZ) in situations where it doesn't interfere (e.g. no connection) and when the NVQ value is getting too low.

2.9.21.5 S374 / Host-less operation

In a hostless operation scenario, no host controller is available which can monitor the NVQ value and reset the module if needed. In addition, with automatic ACR mode, persistent store consumption depends largely on the number of connections to different devices and is difficult to predict.

For this scenario BTM51x provides two options of automatic reset, controlled by flags of S374:

- Bit 0 – Automatic reset on disconnect if no more profile is connected and if NVQ < 300. This is preventive, but the conditions may not be met before ERROR 102 occurs.
- Bit 1 – Immediate reset on ERROR 102. This is a robust recovery method, but it may lose existing connections. Combining both flags (S374=3) seems ideal but it is up to the user to select the best solution for the application.

2.9.22 AT+SR<role>,<pm> {Setting HCI role (master / slave)}

Usually the Bluetooth device which initiates a connection becomes master of the link, and the accepting device becomes slave. Some devices initiate a role change (swapping master and slave role) once a connection is established. This is observed when initiating an SPP link from BTM51x to a PC (Windows 7): reading back the local HCI role (ATI56) has indicated 'slave' for SPP.

Normally the application doesn't need to address the HCI role. However, in some cases changing HCI role can resolve issues. Such a scenario is an A2DP/AVRCP connection, initiated by BTM51x to an iPhone with AAC enabled. If BTM51x remains master, regular drops in the audio stream are observed. After requesting slave role for BTM51x, the issue is solved.

The following AT command allows control of HCI role:

- AT+SR<role>,<profile_mask> – request local HCI role for one given profile
  - <role> = M for master / S for slave
  - Example: AT+SRM,$0001 request local master for SPP profile
  - Example: AT+SRS,$0001 request local slave for SPP profile
  - As the role change request can be rejected by the remote device, there is no guarantee that the request will be successful
  - Check the outcome by ATI56 (100ms delay between AT+SRx and ATI56 is recommended)
  - ERROR 04: profile is not connected
  - ERROR 05: more than one flag is set in the profile mask or other syntax error
  - It is also possible to configure automatic request of master or slave role as soon as a profile has connected:
- S368: set SLAVE role on connect:
  - Parameter: profile mask (like S102). For each profile enabled in S368, whenever the profile connects, local slave role is requested, regardless of the connection's direction (incoming/outgoing).
  - When writing 1 to a profile flag which is already set in S369, the flag is cleared in S369.
  - Check local HCI role for each profile: ATI56

  - Range: 0…$1FF
- S369: set MASTER role on connect:
  - Parameter: profile mask (like S102). For each profile enabled in S369, whenever the profile connects, master role for BTM51x is requested, regardless of the connection's direction (incoming/outgoing).
  - When writing 1 to a profile flag which is already set in S368, the flag is cleared in S368.
  - Check local HCI role for each profile: ATI56
  - Range: 0…$1FF

2.9.23 Encryption

If the remote device is BT2.1 (and later) all connections except SDP (service discovery profile) are encrypted. Encryption key length is 56 bit.

If the remote device is BT2.0 or earlier, an encryption request is sent to the remote device once the connection is established, regardless of the profile which has connected (active encryption request by BTM51x). If required, this request can be disabled by S375, differentiated by connection direction (incoming/outgoing).

2.9.23.1 S375 {disable active encryption request to BT2.0 devices}

- S375 – inhibit active encryption request for remote devices BT2.0 or earlier
  - **Bit 0 –** disable active request of encryption for incoming connections (any profile)
  - **Bit 1 –** disable active request of encryption for outgoing connections (any profile)
  - Default value = 1
  - Values can be ORed

2.9.24 SLC/ACL Features

Transmitting custom commands over SLC/ACL

Headset profile (HSP) and Hands-free profile (HFP) are each based on an Rfcomm link to send and receive the AT commands defined in the appropriate profile specification. For HSP the Rfcomm link is called 'ACL' and for HFP the Rfcomm link is called 'SLC' (service level connection).

As of firmware version 22.2.5.0, some commands were added allowing to send and receive custom commands over the SLC/ACL.

Sending SLC commands

AT Command	Max data length (characters)	Description
AT+HSX<data>	26	Send custom data on HS-instance ACL
AT+HFX<data>	26	Send custom data on HF-instance SLC
AT+HSGX<data>	26	Send custom data on HSG-instance ACL
AT+HFGX<data>	26	Send custom data on HFG-instance SLC

<data> can be a string of printable ASCII characters or the two digit hexadecimal presentation of a non-printable character, preceded by '\', e.g. "hello\0D". The maximum length of <data> is limited by the input buffer size for AT commands which is 36 (including terminator \r).

Please note that '\0D' must be utilised to terminate a command or block of data. Otherwise the data will be buffered on the remote end until 0x0D is found. Once the remote SLC parser finds the command terminator 0x0D, it will start parsing and eventually free the RF input buffer of the remote device.

This shows that this feature can be used to transmit short data sequences or short commands. But it is not suitable to transmit large amount of data. Please refer also to **Error! Reference source not found.**.

Obviously, these commands allow to send AT commands straight to the SLC which would normally be sent only by the BTM51x internal processing, triggered by particular BTM51x AT commands or interaction with a remote device. So, whenever a BTM51x AT command exists for a particular purpose, this command should be used rather than the direct method. Only if a feature may be missing, this command may be used. However, it is still possible that internal state issues may occur in such scenarios and such usage is entirely at the users risk.

Custom commands should be designed in a way that there is no ambiguity with existing AT commands as defined in the HFP specification. One criteria is that the remote parser doesn't recognise the command and issues an UC-message (unknown command) as explained in the following section.

Receiving commands from SLC

There are two taps to display incoming SLC/ACL data. The first is before the SLC (ACL) parser and the second is the one after the parser if the command is not recognized. The first tap should be used for debugging or learning purposes; the second should be used for customer commands. See Figure 2-44.

The following explains how to enable the asynchronous tap messages:

Temporary (valid until power cycle or until reverted):

- AT+RX1 – Enable all SLC RX messages (first tap)
- AT+UC1 – Enable all unknown command messages (second tap)

Permanently (valid as of next power cycle):

- S379=1 – Enable all SLC RX messages (first tap)
- S380=1 – Enable all unknown command messages (second tap)
  plus subsequent AT+W and ATZ.

Asynchronous Message	Description
HSRX<data>/HSGRX<data>	Display any data received on HSP ACL.
HFRX<data>/HFGRX<data>	Display any data received on HFP SLC
HSUC<data>/HSGUC<data>	Display unknown command received on HSP ACL
HFUC<data>/HFGUC<data>	Display unknown command received on HSP ACL

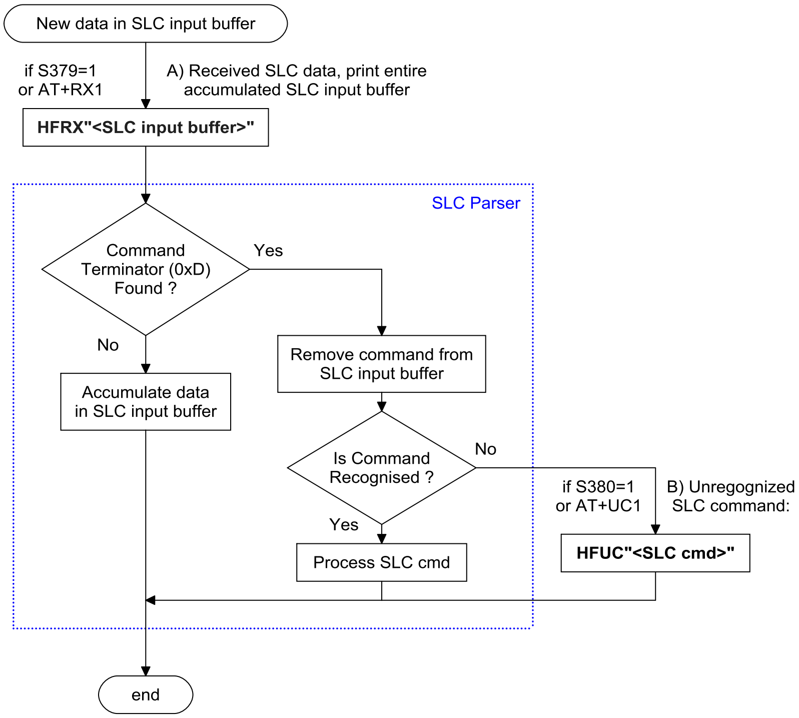


Figure 2-44: Receiving commands from SLC

2.9.25 Ringtones

A set of 94 pre-defined tones has been added which can be played at any time using AT+SIT<n> whereby <n> specifies the index [1..94]. Tone volume is controlled by S387 [0..15]. Some tones are suitable as ring tones, others are preferred for status indications. The only automatic trigger occurs when HF or HS receive the "RING" message on SLC/ACL and if HFG has disabled in-band ringing. In this case, the tone selected by S388 (>0) is played.

Note: GPIO function mapping code (FMC) AudioOn (0x05) is currently not working for tones. See the Release Notes Known Issues for more information.

If AT+SIT<n> is issued during an active A2DP or SCO link with audio routed to I2S or PCM interface, then the tone is played back to the active audio interface. In all other cases, the tone is played back to the analogue audio output.

For A2DP sink with I2S interface at fixed rate of 48kHz (S419=7), a strange sound of some ring tones was observed. It is suspected that this is caused by the internal resampling process.

2.9.26 Tolerant pairing policy / Android compatibility

It has been observed that during initial pairing Android devices require an authenticated link key / MITM protection. If BTM51x I/O capabilities are set to no input/no output (S321=3) then a confirmation of the 6 digit passkey is not possible and hence the pairing is rejected by BTM51x. In other words, a link key is not created because combined I/O capabilities of Android device and BTM51x are insufficient to create an authenticated link key.

It has further been observed that for the actual connection to a profile (or service), the Android device does not require an authenticated link key and that an unauthenticated link key is sufficient, which is contrary to requirements of the initial pairing. If the actual profile connection or services do not require an authenticated link key then initial pairing should not set such requirement either because it rules out devices which strictly adhere to the Bluetooth specification v2.1+EDR with regards to Secure Simple Pairing.

In order to work around this, a feature has been added which allows BTM51x to tolerate and continue a pairing process even if I/O capabilities are insufficient to create an authenticated link key (MITM protection). Obviously, the resulting link key type will be unauthenticated (=3) rather than authenticated (=4) . The link key type can be double checked using AT+BTT?. It is appended to the list of paired devices.

BTM51x's tolerant-pairing-policy feature is enabled by S382=1. It is not enabled by default because, from Laird's point of view, although tolerant pairing may not break the Bluetooth specification, it is weakening the approach to security measures introduced by Secure Simple pairing (BT Core Spec V2.1+EDR).

3 APPENDIX

3.1 S Registers

The following table lists all S Registers.

Table 3-1: BTM - General S Registers

Register	Default	Range	Category	Description
S0	0	-1..15	SPP	Number of RING indication before automatically answering an incoming connection. A value of 0 disables autoanswer. If -1, then autoanswer on one RING and do NOT send RING/CONNECT response to the host. This emulates a serial cable replacement situation Setting values >= 0, resets S Register 504 to 0 and <0 forces 504 to 1. If S0 <> 0 and S100 <> 0 then S0 must be < S100. If a value is entered which violates this rule, then ERROR 29 is sent in response. If S504 =1 then this register returns -1, regardless of the actual value stored in non-volatile memory.
S2	94	32..126	n/a	Escape sequence character. It is not '+' by default as a serial link to a mobile phone exposes the phone's AT command set, which uses '+' as default. If both use '+' there is confusion. 0x5e is the character '^'.
S12	100	40..5000	n/a	Escape sequence guard time in milliseconds, with a granularity of 20ms. New values are rounded down to the nearest 20ms multiple
S100	15	0..15	SPP	Number of RING indications before an auto disconnection is initiated. A value of 0 disables this feature. If S0 <> 0 and S100 <> 0 then S0 must be < S100. If a value is entered which violates this rule, then ERROR 29 is sent in response.
S101	$1101	$0..$ffff	n/a	UUID of default SPP based profile when not specified explicitly in the ATD command.
S102	$0181	$1..$1ff	n/a	Defines a set of bits masks for enabling profiles. Values can be ORed. A profile can be enabled only if it is supported by the BTM variant. Issue AT&W and ATZ in order to make the new setting effective 0x001 Serial Port Profile (SPP) 0x002 Headset (HS)

Register	Default	Range	Category	Description
				0x004 reserved 0x008 Audio Gateway (Headset, HSG) 0x010 Hands-free (FH) 0x020 OBEX FTP 0x040 Audio Gateway (Hands-free, HFG) 0x080 A2DP 0x100 AVRCP
S103	1	1..4	n/a	Boot Mode on cold boot 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). 1 Normal 2..4 For future customization of the module
S300	1	0..2	A2DP	Set A2DP role: 0 Feature not set 1 A2DP Sink (default) 2 A2DP Source Needs subsequent AT&W and ATZ to become effective
S301	3	0..3	AVRCP	Set AVRCP role: 0 Disabled 1 Control (CT) 2 Target (TG) 3 Controller and target, enabling AVRCP1.5 CT – Default Needs subsequent AT&W and ATZ to become effective
S302	1	0..4	AVRCP	Set AVRCP category: 0 Feature disabled 1 Player/recorder – Default 2 Monitor/amplifier 3 Tuner 4 Menu Needs subsequent AT&W and ATZ to become effective

Register	Default	Range	Category	Description
S303	$FFFFFF	$000000.. $FFFFFF	AVRCP (TG)	Set Company Id IEEE Company ID, 24bit hexadecimal, required for UNIT INFO Response in AVRCP target mode, default value is 0xFFFFFF.
S305	1	0..1	AVRCP (TG)	Enable Unit Info Response: 0 Reject incoming Unit Info Requests 1 Accept incoming Unit Info Requests and send response automatically (default) with Company ID as per S303 and unit type = 0x09 (Panel, fixed)
S306	1	0..1	AVRCP (TG)	Enable Subunit Info Response: 0 Reject incoming Subunit Info Requests 1 Accept incoming Subunit Info Requests and send response automatically (default) with Subunit type = 0x09 (Panel, fixed) and MaxSubUnitId = 0x00 (fixed)
S307	0	0..3	A2DP	Enable optional A2DP codec: 0 No optional codec – Default 1 APTx (Bit 0) 2 AAC (Bit 1, sink only) 3 Both APTx and AAC enabled Subsequent AT&W plus ATZ required for a new value to become effective
S308	11	0..15	A2DP/ all profiles	Automatic control of discoverable/connectable mode when entering and exiting A2DP connection (Auto-BTX): **Bit** **Description** 0 Make the module not discoverable and not connectable when entering a connection 1 When exiting an A2DP connection: restore discoverable and connectable state according to S512 3, 2 Multi-profile policy: 00b A2DP only – Bits 0 and 1 apply only for A2DP connections (backward compatible) 01b Enter auto-BTX when **all** profiles of S102 are connected. Restore to S512 when **one** profile is disconnected. 10b Enter auto-BTX when **one** profile is connected. Restore to S512 when **all** profiles are disconnected (default) 11b Enter auto-BTX when **all** profiles of

Register	Default	Range	Category	Description
				S102 are connected. Restore S512 when **all** profiles are disconnected. **Note:** The default value of S308 = 11 makes the BTM51x not discoverable and not connectable after the first profile is connected. If more than one incoming connection is expected, then S308 must be changed appropriately. Refer to section 20.4.8. Default value – 11 (1011b – multi-profile policy = 10b). The current scanning state (discoverable/ connectable) can be queried by ATI27. Also refer to Release A2DP Connection.
S309	1	0..1	Audio Inter-face	Enable asynchronous FS-message (presenting sampling frequency and audio interface) 0 Disable 1 Enable – Default
S310	1	0..1	AVRCP (TG)	Configure PASS THROUGH (PT) Response: 0 Enable automatic PT response; response type is read from S311 – Default 1 Host is required to respond to PT indication. See AT+AVR.
S311	1w/9r	0..7	AVRCP (TG)	Set automatic PT response type This value is queried for automatic PT-Response, see Table 2-30. Default value is "accepted" 1w/ 9r **Note:** if this value is set to "reject" (2w/10r), then incoming Pass Through commands are not forwarded to the host processor (no AVPTI message is sent to the host.)
S312	1	1..15	A2DP (SNK)	Set A2DP Sink supported features, Bitmask: 0 Headphone – Default 1 Speaker 2 Recorder 3 Amplifier Subsequent AT&W plus ATZ required for a new value to become effective.

Register	Default	Range	Category	Description
S313	1	1..15	A2DP (SRC)	Set A2DP Source supported features bit mask Bit 0 Player – Default Bit 1 Microphone Bit 2 Tuner Bit 3 Mixer Subsequent AT&W plus ATZ required for a new value to become effective.
S314	0	0..2	A2DP	A2DP audio interface select: 0 Analogue (on-board audio codec) 1 I2S master 2 I2S slave
S315	0	0..	SCO	eSCO/SCO audio interface select: 0 Analogue (on-board audio codec) 1 I2S master 2 I2S slave 3 PCM master (PCM_M) 4 PCM slave (PCM_S)
S316	63	1..63	A2DP	I2S sampling rate capability for A2DP; values can be added for all sampling frequencies supported. Register is referenced if S314>0 at boot time, requires AT&W and ATZ for new values to become effective: 1 48 kHz(1) 2 44.1 kHz(1) 4 32 kHz 8 24 kHz(2) 16 22.05 kHz(2) 32 16 kHz (1) A2DP Source: at least one of the sampling frequencies 48kHz, 44.1kHz must be supported; A2DP Sink : both 48 kHz and 44.1 kHz must be supported (at least) (2) values ignored as they are not supported by SBC or APTx
S318	4 (X1M)	0..4	CVC	SCO DSP image select: 0 Disable CVC (DSP is not used on an SCO link) 1 CVC Headset, one microphone 2 CVC Headset, two microphones

Register	Default	Range	Category	Description
				3 CVC Hands-free, one microphone 4 X1M (DSP application for resampling/WBS, one microphone input, no CVC) For details on CVC refer to the application note CVC on BTM5xx 0, provided by Laird Technologies under NDA.
319	0	0..1	Misc.	SPP smart disconnect: 0 Disable – Default 1 Enabled With this setting, BTM51x detects if there is any data pending in its internal buffers on an incoming disconnect notification. If so, then BTM51x delays the disconnection until all pending data has been delivered to the UART first and then signals the disconnection on UART (NO CARRIER) and on DCD line. This is an experimental feature which may have side effects in certain situations. It was ported from BTM41x as the result of fixing bug [Ref. 3-21 / 285]
S320	2	1..3	SSP	Security Level: see [1], vol3, Generic Access Profile - Table 5.7Z Needs subsequent AT&W and power cycle to take effect. value = 3 overwrites S322
S321	1	0..4	SSP	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	SSP	Force man-in-the-middle-protection (MITM): 0 Disabled 1 Enabled Referenced only if security level (S320) < 3
S323	0	0..1	SSP	Disable legacy (pre-BT2.1) Pairing: 0 Legacy pairing enabled 1 Legacy pairing disabled
S324	90	1..255	SSP	Secure Simple Pairing timeout in s This value must be at least 60 in order to meet the

Register	Default	Range	Category	Description
				recommendation of BT2.1 specification
S325	1	0..1	n/a	Store link key automatically on dedicated bonding outgoing (DBO)
S326	1	0..1	n/a	Store link key automatically on general bonding outgoing (GBO)
S327	1	0..1	n/a	Store link key automatically on dedicated bonding incoming (DBI)
S328	1	0..1	n/a	Store link key automatically on general bonding incoming (GBI)
S329	0	0..1	n/a	Enable legacy (BISM2) response format
S330	1	1..31	n/a	Configure inquiry response of AT+BTI (Bitmask): 0 Show device address 1 Show class of device 4 Show friendly name 8 Show extended inquiry data Values can be ORed
S331	2	0..2	n/a	Direction indication style for CONNECT messages: 0 Disabled 1 Character style – Append **I** to incoming and **O** to outgoing CONNECT message. Separated by a comma. 2 Symbol style – Append **<** to incoming and **>** to outgoing CONNECT message. Separated by a comma. Applies only to role indicating UUID (e.g.HSP/HFP) and if S329=0
S332	1	0..1	HSP	Enable HS disconnection ATH1108 and ATH*: 0 Disabled 1 Enable – Default Should only be enabled for test purposes, because disconnection initiated by HS other than sending AT+CKPD=200 to gateway is not defined in HSP specification.
S333	1	0..1	HFP	Enable verbose indicators : 0 Display indicator ID only in HFI.. asynchronous message. Refer to DTMF tone request (#17_4.27)

Register	Default	Range	Category	Description
				1 Display indicator string in HFI… asynchronous message. Refer to DTMF tone request (#17_4.27)
S334	0	0..1	n/a	Enable Extended Sdp Error Codes: 0 Disabled 1 Enable
S335	0	0..3	LED	LED 0 mode: 0 LED_OFF 1 LED_ON 2 LED_PWM 3 LED_PULSE
S336	2048	0..4095	LED	LED 0 Duty Cycle – Referenced in LED mode = LED_PWM or LED_PULSE
S337	0	0..15	LED	LED 0 PWM Period – Referenced in LED mode = LED_PWM
S338	5	0..15	LED	LED 0 Pulse Rate – Referenced in LED mode = LED_PULSE
S340	0	0..3	LED	LED 1 mode, see S335
S341	2048	0..4095	LED	LED 1 Duty Cycle, see S336
S342	0	0..15	LED	LED 1 PWM Period, see S337
S343	5	0..15	LED	LED 1 Pulse Rate, see S338
S345	1	0..3	HSG	Enable automatic alerting on ACL establishment: 0 Disable auto-alerting 1 Auto-alerting on outgoing ACL established 2 Auto-alerting on incoming ACL established 3 Auto-alerting on outgoing and incoming ACL established
S346	0	0..1	HSG	Enable in-band ringing: 0 Disable 1 Enable
S347	0	0..1	HSG	Enable automatic ACL release: 0 Disable 1 ACL is released automatically when audio is released by the HS.
S348	0	0..170	SSR	SSR (maximum) remote interval – The maximum amount

Register	Default	Range	Category	Description
				of time that the remote device is allowed to be absent in SSR mode. Applies to SLC/ACL only, i.e. no audio link being present
S349	0	0..170	SSR	SSR (minimum remote timeout – The minimum time with no data exchange in sniff mode before a remote device is allowed to enter SSR. Applies to SLC/ACL only, i.e. no audio link being present
S350	0	0..170	SSR	SSR (minimum) local timeout – The minimum time with no data exchange in sniff mode before a local device is allowed to enter SSR. Applies to SLC/ACL only, i.e. no audio link being present
S354	0	0..1	HFG	Enable/disable verbose mode for call record presentation: 0 Non-verbose mode for AT+HFGC? – Default 1 Verbose mode for AT+HFGC?
S355	1	0..3	A2DP+HFP / HSP	Configure audio resource override: 0 No override allowed 1 Incoming/outgoing SCO request is accepted while A2DP is streaming. The module initiates A2DP suspend automatically – Default 2 Incoming/outgoing A2DP start/resume request is accepted while SCO is active. The module releases the SCO link automatically. 3 Both 1 and 2 enabled Refer to Simultaneous A2DP and HFP as well.
S356	3	0..3	SCO	Enable restoring of last gain used for SCO: 0 Disable gain restoring for SCO 1 Enable output gain restoring for SCO 2 Enable input gain restoring for SCO 3 Enable input and output gain restoring for SCO – Default
S357	3	0..3	A2DP	Enable restoring of last gain used for A2DP: 0 Disable gain restoring for A2DP 1 Enable output gain restoring for A2DP 2 Enable input gain restoring for A2DP 3 Enable input and output gain restoring for A2DP – Default
S362	0	0..1	AVRCP	Suppress AVRCP direction indicator:

Register	Default	Range	Category	Description
				0 AVRCP direction indicator enabled 1 AVRCP direction indicator suppressed
S363	1	1..3	A2DP	Set default post processing mode: 1 Pass through (no post-processing) 2 Full processing 3 Custom mode
S364	0	0..31	Sniff Mode	Delayed sniff mode – The delay (in seconds) before local sniff parameters (S561..S564) are requested after entering a connection.
S365	0	0..1		Initial power mode after entering a connection: 0 Passive (default) 1 Active
S368	$0000	$0..$1FF	Misc	Request HCI **slave** role on connect for profile flags set Profile mask as per S102 (Table 3-7)
S369	$0000	$0..$1FF	Misc	Request HCI **master** role on connect for profile flags set Profile mask as per S102 (Table 3-7)
S370	0	0..7	ACS	Configure Auto Connect Service mode Bitmask: Bit 0 Start AC service at boot time Bit 1 Stop AC service by de-asserting DSR Bit 2 Auto-add new connections to ACR table (automatic/host-less ACR mode)
S371	0	0..900	HLO	Time window in seconds for connectable / discoverable if entered by FMC 0x11 Granularity: 4 If value is ≤ 60 (round up) 60 If value is > 60 (round down) See Input FMC 0x11 / S371 for details.
S372	0	0..1	Misc.	Control of automatic re-pairing if link key missing on remote device but local link key is available. 0 Automatic repairing occurs on connection attempt 1 Automatic repairing is inhibited; connection attempt fails.
S373	0	0..7	ACR	Default ACR option flags for automatic/host-less mode, bitmask:

Register	Default	Range	Category	Description
				Bit 0 Always reconnect (0x01) Bit 1 Always send AVRCP play (0x02) Bit 2 Never send AVRCP play (0x04) See S373 {default ACR option flags} for more detail.
S374	0	0..3	NVM	Management of non-volatile memory in host-less operation; bitmask: Bit 0 Automatic reset on disconnect if no other profiles are connected and if NVQ < 300 Bit 1 Immediate reset on ERROR 102 See Management of Persistent Store for more detail.
S375	1	0..3	Misc.	Inhibit active encryption request **for remote devices BT2.0 or earlier**; bitmask: Bit 0 Disable encryption request for incoming connections (any profile) Bit 1 Disable encryption request for outgoing connections (any profile)
S376	1	0..1	WBS, mSBC	Enable WBS/mSBC. We don't recommend a change while HFP is connected, but it works for demo purposes between two BTM51x modules. 0 Disable 1 Enable (default)
S377	1	0..1		Enable CVC-BEX. CVC support is not currently available; this S register can be ignored until further notice.
S378	1	0..1	HSG, HFG	Close audio on voice recognition stop. 0 Disable 1 Enable (default)
S379	0	0..1	HFP, HSP	Enable RX message for HFP (SLC spy) and HSP (ACL spy) 0 Disable (default) 1 Enable
S380	0	0..1	HFP, HSP	Enable UC (unrecognized command) message for HFP and HSP. 0 Disable (default) 1 Enable
S381	0x7f	0..0x7f	HFP	HFP indicators enable mask; Individual indicator activation; See also HFP1.6 spec, search for "AT+BIA" command Bitmask:

<table>
<tr><th>Register</th><th>Default</th><th>Range</th><th>Category</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td><table>
<tr><td>0x01</td><td>Service indicator</td></tr>
<tr><td>0x02</td><td>Signal strength indicator</td></tr>
<tr><td>0x04</td><td>Roaming status indicator</td></tr>
<tr><td>0x08</td><td>Battery charge indicator</td></tr>
<tr><td>0x10</td><td>Call indicator</td></tr>
<tr><td>0x20</td><td>Call setup indicator</td></tr>
<tr><td>0x40</td><td>Call held indicator</td></tr>
</table></td></tr>
<tr><td>S382</td><td>0</td><td>0..1</td><td>SSP</td><td>Enable tolerant pairing policy / Android compatibility<table>
<tr><td>0</td><td>Disable (default)</td></tr>
<tr><td>1</td><td>Enable</td></tr>
</table></td></tr>
<tr><td>S383</td><td>0</td><td>0..1</td><td>SPP</td><td>Enable inquiry exception<table>
<tr><td>0</td><td>Disable (default)</td></tr>
<tr><td>1</td><td>Enable</td></tr>
</table></td></tr>
<tr><td>S384</td><td>$0006</td><td>$0..$4ff</td><td>I2S</td><td>I2S format configuration which corresponds to PSKEY_DIGITAL_AUDIO_CONFIG (&01D9)<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>Justify Format</td><td>0 = Left justified
1 = Right justified</td></tr>
<tr><td>1</td><td>Left Justify Delay</td><td>For left justified formats:
▪ 0 – MSB of SD data occurs in the first SCLK period following WS transition
▪ 1 – MSB of SD data occurs in the second SCLK period.</td></tr>
<tr><td>2</td><td>Channel Polarity</td><td>For 0, SD data is left channel when WS is high.
For 1, SD data is right channel.</td></tr>
<tr><td>3</td><td>Audio Attenuation</td><td>For 0, 17 bit SD data is rounded down to 16 bits.
For 1, the audio attenuation defined in AUDIO_ATTEN_VALUE is applied over 24 bits with saturated rounding.
Requires 16_BIT_CROP_EN (bit 10) to be 0.</td></tr>
<tr><td>7:4</td><td>Justify Resolution</td><td>Attenuation in 6 dB steps.</td></tr>
<tr><td>9:8</td><td>Justify Resolution</td><td>Resolution of data on SD_IN:
▪ 00 = 16 bit
▪ 01 = 20 bit
▪ 10 = 24 bit
▪ 11 = Reserved.</td></tr>
</table></td></tr>
</table>

Register	Default	Range	Category	Description
				This is required for right justified format and with left justified LSB first.
				10 – 16-bit Crop Enable – For 0, 17 bit SD_IN data is rounded down to 16 bits. For 1, only the most significant 16 bits of data are received.
S385	0	0..1		Append channel information to unsolicited FS…INT message: 0 – Do not change FS… INT message 1 – Append _L (left), _R (right), and _D (dual/stereo) according to current configuration. Example: FS16000,INT_D
S386	0	0..3	MPS	S386 provides three options for enabling MPS: 0 – Disable 1 – Enable MPS service record and set S355 to 1. The user is responsible for enabling all required profiles (S102) and roles (S300, S301). 2 – Automatic configuration for A2DP-SNK/ AVRCP-CT / HF 3 – Automatic configuration for A2DP-SRC/ AVRCP-TG / HFG
S387	15	0..15		Set tone volume (analogue output only) For HF/HS ringtone or AT+SIT<n>
S388	31	0..94		Ringtone select 0 – Disable 1..94 – HF/HS: play selected tone on incoming ‘RING’ messages on SLC/ACL [for HF: only if HFG has disabled in-band ringing] Tones can be tested using AT+SIT<n>
S389	0	0..31	FMC	Input function mapping codes parameter. The value is interpreted depending on FMC. Currently, only input FMC 0x14 refers to this S-Register.
S390	$00F7	$0..$ff	AVRCP	**AVRCP1.5 configuration** Bit – Mask – Descr. – Value

Register	Default	Range	Category	Description			
				0	0x01	Hint strings	0 – Do not append hint strings 1 – Append hint strings to: ▪ Play status (AVPS) ▪ Player application settings (AVPA / AVPV ▪ Folder type (AVFSI) ▪ Media type (AVMEI) ▪ Media element attribute (AVMEA) ▪ Now playing track attribute (AVNPA)
				1	0x02	Media element attributes (browsing)	Ignored: available attributes are **not** appended to media items (AVMEI). Reason: on the test device extra attributes did not provide additional information (just the track name which is already provided), but more importantly, the extra attribute data was observed to overrun the UART Tx Buffer due to inflated data format for presentation (start/stop tags etc.)
				2	0x04	ASCII presentation of strings	0: print displayable names, attribute values and other strings in hexadecimal format, more machine friendly than human friendly 1: print ASCII strings, surrounded by double quotes; non-printable characters are presented hexadecimal as '\xx'. A character is regarded non-printable if it is >=128 or double quote (") or backslash (\) or space ( ) or comma (,). More human friendly than machine friendly but also more exceptions (e.g. character set)

Register	Default	Range	Category	Description			
				3	0x08	Attributes query basic / full	On automatic requests for now playing track attributes query: 0: full set of attributes (Title, Artist, Album, Track no., Total tracks, Genre, Playing time) 1: only basic set of attributes (Title, Artist, Album)
				4	0x10	Action on 'Play status changed' notif. (AVPS…)	On reception of AVPS notification (play status changed) and if new status is 'playing' (0x01): 0: automatically query now playing track attributes 1: do nothing
				5	0x20	Action on 'Track changed' notif. (AVTC)	On reception of AVTC notification (track changed): 0: automatically query now playing track attributes 1: do nothing
				6	0x40	Action on 'Now playing content changed' notif. (AVNPC)	Ignored: **No** automatic query of now playing track is initiated on reception of 'AVNPC'. It has turned out that this was A) not useful and B) confusing due to multiple redundant AVNPC notifications (up to four observed)
				7	0x80	Action on 'Available players changed' notif. (AVAPC)	On reception of AVAPC notification (available players changed): 0 – Automatically query available players list (*AT+AVQAP*) 1 – Do nothing **Note:** AVAPC has been observed when switching forth and back between the standard player 'Music' and other streaming applications, e.g. web-radio player or similar.
S391	1	0..1	UID	Enable UID cache UID cache is designed to facilitate the handling of 64-bit UIDs. Up to 10 UIDs of a list (AT+AVQFS) are cached			

<table>
<tr><th>Register</th><th>Default</th><th>Range</th><th>Category</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>when the list is printed.</td></tr>
<tr><td>S396</td><td>1</td><td>0..1</td><td>SPP</td><td>Unregister SPP service record once connected.
<table>
<tr><td>0</td><td>Leave the SPP service record unchanged once SPP is connected.</td></tr>
<tr><td>1</td><td>Unregister the SPP record once SPP is connected (default).</td></tr>
</table>
Note: This S-register is queried at boot time only. In order to view the effect, AT&W and ATZ are also required.</td></tr>
<tr><td>S397</td><td>$000C</td><td>$0..$ff</td><td>AVRCP</td><td>AVRCP1.5 notification control – Allows the disabling of selected AVRCP1.5 notification messages.</td></tr>
<tr><td>S411</td><td>400</td><td>200..1600</td><td>GPIO</td><td>Short press duration in milliseconds. 200 ms granularity</td></tr>
<tr><td>S412</td><td>2500</td><td>500..4000</td><td>GPIO</td><td>Component of medium press duration in milliseconds, 500 ms granularity; actual value is this value plus S411; actual value is returned by ATI412</td></tr>
<tr><td>S413</td><td>2000</td><td>500..4000</td><td>GPIO</td><td>Component of long press duration in milliseconds, 500 ms granularity; actual value is this value plus S411 + S412; actual value is returned by ATI413</td></tr>
<tr><td>S415</td><td>0</td><td>0..1</td><td>MicGain</td><td>Enable Microphone Input Gain, adds extra 24 dB to input gain</td></tr>
<tr><td>S416</td><td>0</td><td>0..1</td><td></td><td></td></tr>
<tr><td>S417</td><td>0</td><td>0..15</td><td></td><td></td></tr>
<tr><td>S418</td><td>0</td><td>0..15</td><td></td><td></td></tr>
<tr><td>S419</td><td>0</td><td>0..7</td><td>Sampling rates</td><td>Set sampling rate for Audio Loopback and Fixed sampling rate:
<table>
<tr><th colspan="4">Loopback Mode Sampling Rate</th></tr>
<tr><th>S419</th><th>Analogue
[AT+BTL1]</th><th>PCM cross LB
[AT+BTL7/8]</th><th>I1S cross LB
[AT+BTL3/4]</th></tr>
<tr><td>0</td><td>8 kHz</td><td rowspan="2">8 kHz</td><td>8 kHz</td></tr>
<tr><td>1</td><td>11.025 kHz</td><td>11.025 kHz</td></tr>
<tr><td>2</td><td>16 kHz</td><td rowspan="6">16 kHz</td><td>16 kHz</td></tr>
<tr><td>3</td><td>22.05 kHz</td><td>22.05 kHz</td></tr>
<tr><td>4</td><td>24 kHz</td><td>24 kHz</td></tr>
<tr><td>5</td><td>32 kHz</td><td>32 kHz</td></tr>
<tr><td>6</td><td rowspan="2">44.1 kHz</td><td>44.1 kHz</td></tr>
<tr><td>7</td><td>48 Hz[1]</td></tr>
</table>
(1) Only for path I2S_IN –> analogue_out (DAC). Analogue in (ADC) does not support a 48 kHz sampling</td></tr>
</table>

<table>
<tr><th>Register</th><th>Default</th><th>Range</th><th>Category</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>rate.
<table>
<tr><th colspan="4">SCO/A2DP Fixed Sampling Rate</th></tr>
<tr><th rowspan="2">S419</th><th colspan="2">SCO Audio</th><th>A2DP Sink</th></tr>
<tr><th>PCM
[S315=3/4]</th><th>I2S
[S315=1/2]</th><th>I2S [S314=1/2]</th></tr>
<tr><td>0</td><td>Disabled</td><td>Disabled</td><td>Disabled</td></tr>
<tr><td>[1,2]</td><td rowspan="4">16 kHz</td><td>16 kHz</td><td rowspan="3">44.1 kHz</td></tr>
<tr><td>[3,5]</td><td>32 kHz</td></tr>
<tr><td>6</td><td>44.1 kHz</td></tr>
<tr><td>7</td><td>48 kHz</td><td>48 kHz</td></tr>
</table>
(1)</td></tr>
<tr><td>S504</td><td>0</td><td>0..1</td><td>n/a</td><td>Enable silent operation:
Setting to 1 forces S0 to -1 and suppresses messages arising from connections or pairing. E.g. CONNECT, NO CARRIER, RING, PAIR etc.
Suppressing connection based messaged allows the device to be configured in cable replacement mode</td></tr>
<tr><td>S505</td><td>10</td><td>2..120</td><td>n/a</td><td>Minimum delay before abandoning connection attempt as a master.
Referenced by ATD. In units of seconds. See S Registers 530 and 543 also. Note that since disconnection times vary, this register only guarantees the minimum delay. Note that for invalid addresses specified in the ATD command, the NO CARRIER response is immediate. See S register 560 for specifying disconnect max timeout</td></tr>
<tr><td>S506</td><td>0</td><td>0..1</td><td>n/a</td><td>Enable/Disable echoes. The ATEn command also affects this.</td></tr>
<tr><td>S507</td><td>0</td><td>0..2</td><td>n/a</td><td>
<table>
<tr><td>When set to 0</td><td>A connection can be dropped using ^^^ escape sequence only and the state of the DSR line is ignored.</td></tr>
<tr><td>When set to 1</td><td>A connection can be dropped using either the ^^^ escape sequence or the DSR modem control line.</td></tr>
<tr><td>When set to 2</td><td>A connection can only be dropped using a de-assertion of DSR. Mode 2 provides for the highest data transfer rate.</td></tr>
</table>
If the status of the DSR line is to be conveyed to the remote device as a low bandwidth signal then this register MUST be set to 0, otherwise a de-assertion of DSR is seen as a request to drop the Bluetooth® connection. This register affects S Register 536 – see details of 536</td></tr>
</table>

Register	Default	Range	Category	Description
S508	640	11..2550	n/a	Page Scan Interval in milliseconds. Minimum is 11.25 ms so 10/11ms gives 11.25 ms, refer to Page Scan/Inquiry Scan Interval and Window.
S509	160	11..2550	n/a	Page Scan Window in milliseconds. Minimum is 11.25 ms so 10/11ms gives 11.25 ms, refer to Page Scan/Inquiry Scan Interval and Window.
S510	640	11..2550	n/a	Inquiry Scan Interval in milliseconds. Minimum is 11.25 ms so 10/11ms gives 11.25 ms, refer to Page Scan/Inquiry Scan Interval and Window.
S511	160	11..2550	n/a	Inquiry Scan Window in milliseconds. Minimum is 11.25 ms so 10/11ms gives 11.25 ms, refer to Page Scan/Inquiry Scan Interval and Window.
S512	1	0..7	n/a	Specify power up state: When set to 0 — AT+BTO is required to open the device for Bluetooth activity. When set to 1 — It proceeds to a state as if AT+BTO was entered. When set to 2 — It is discoverable only, similar to issuing AT+BTQ. When set to 3 — It is connectable but not discoverable (e.g. AT+BTG) When set to 4 — It is connectable and discoverable (e.g. AT+BTP) When set to 5 — It is like 2 but all UART RX traffic is discarded in absence of a connection while DSR is asserted. If DSR is not asserted, then it behaves exactly as per mode 2. When set to 6 — It is like 3 but all UART RX traffic is discarded in absence of a connection while DSR is asserted. If DSR is not asserted, then it behaves exactly as per mode 3. When set to 7 — It is like 4 but all UART RX traffic is discarded in absence of a connection while DSR is asserted. If DSR is not asserted, then it behaves exactly as per mode 4. **Note:** By implication, a change to this can only be seen after a power cycle AND if AT&W is sent to the module.
S514	10	1..60	n/a	Pairing Timeout in seconds. This includes the time a host takes to supply the PIN number when PIN? messages are

<table>
<tr><th>Register</th><th>Default</th><th>Range</th><th>Category</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>indicated.</td></tr>
<tr><td>S515</td><td>$001F00</td><td>$000000..
$FFFFFF</td><td>n/a</td><td>Default Device Class Code. When queried, the value is always printed as a hexadecimal number.
To change the device class of the module temporary and immediately without power cycle, use the command AT+BTC.
To change the device class of the module permanently, write the new value to this S Register (ATS515=$<devclass$_{hex}$>), save the setting (AT&W) and initiate a power cycle (ATZ).</td></tr>
<tr><td>S516</td><td>$000000</td><td>0..
$50331647</td><td>n/a</td><td>Default Device Class filter to be used with AT+BTI when it is not explicitly specified. When queried the value is always printed as a hex number.
The seventh most significant digit, can be 0, 1, or 2, and specifies the type of device class filter.<table><tr><td>When 0</td><td>It specifies no filtering</td></tr><tr><td>When 1</td><td>It specifies an AND mask and all 24 bits are relevant</td></tr><tr><td>When 2</td><td>It specifies a filter to look for devices with matching major device class which occupies a 5-bit field from bits 8 to 12 inclusive (assuming numbering starts at bit 0). All other 19 bits must be set to 0.</td></tr></table></td></tr>
<tr><td>S517</td><td>20</td><td>2..61</td><td>n/a</td><td>Inquiry Length in units of seconds. This parameter is referenced by the AT+BTI command.</td></tr>
<tr><td>S518</td><td>8</td><td>0..255</td><td>n/a</td><td>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 returns 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 remains unchanged for that particular firmware build.</td></tr>
<tr><td>S519</td><td>500</td><td>100..3000</td><td>n/a</td><td>When S507>0, and in a connection, DSR can move the module from data to command state by de-asserting the DSR line for less than the time specified in this register. This value is rounded down to the nearest 100 ms.</td></tr>
<tr><td>S520</td><td>9600</td><td>1200..
..115200</td><td>n/a</td><td>Change to a standard baud rate. The effect is immediate and in fact the OK is sent at the new baud rate. Only one of the following baud rates are accepted: 1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600, 115200.</td></tr>
</table>

Register	Default	Range	Category	Description
				If S register 525=1, then the maximum baud rate is limited to 115200.
S521	9521	1200.. ..921600	n/a	Change baud rate to non-standard value. BTM modules support any baud rate. The only limitation is the integer arithmetic involved, which may adjust the applied rate slightly. If the internally computed baud rate is more than 2% offset from the desired input value, then an ERROR is returned and the old baud rate will prevail. To inspect the actual baud rate, use ATS521? S521 should only be used for non-standard baud rates. For standard baud rates use S520. The effect is immediate and in fact the OK is sent at the new baud rate. If S Register 525=1, then the max baud rate is limited to 115200. In the event that a non-standard baud rate is requested, it is entirely possible that the host is not capable of generating such a baud rate. In this case the module cannot be communicated with. If this happens, there is a procedure to recover from this situation which is described in Factory Default UART Baud Rate.
S523	1	1..2	n/a	Number of Stop bits See S Register 526 for further information.
S524	0	0..2	n/a	Parity: 0 | None 1 | Odd 2 | Even For the Go blue Activator variant of the module this register is read only. See S Register 526 for further information.
S525	0	0..1	n/a	Apply multiplier of 8 to baud rate internally. This is set to 0 (disabled) by default. If S Register 521 > 115200 then this register cannot be set to 1. See S Register 526 for further information.
S526	3	1..3	n/a	This register specifies a 2 bit mask used to qualify how S Registers 520 to 525 are actioned. If bit 0 is 1 | The new communication parameters affect the UART immediately. If bit 1 is 1 | The new communication parameters are stored in non-volatile memory.

Register	Default	Range	Category	Description
				For example, to change communication parameters, but have them come into effect only after subsequent power cycles, then this register should be set to 2, and likewise to affect immediately and yet not have it persist over a power cycle, the value should be set to 1. Must be set before the baud rate change.
S530	1000	100.. 15000	n/a	Reconnect delay when configured as master in pure-cable-replacement mode. This value is rounded down to the nearest 100ms. See S Register 505 also.
S531	0	0..4	n/a	On SPP connect mode: specifies the mode on SPP connection establishment. 0 Normal. Data is exchanged between UART and RF. 1 LOCAL_COMMAND. UART input is parsed by the AT interpreter and RF data is discarded. 2 REMOTE_COMMAND. RF input is parsed by the AT interpreter and UART data is discarded. If S Register 536 is not 1 then this register cannot be set to 2 and an ERROR is returned. 3 LOCAL_COMMAND. UART input is parsed by the AT interpreter and incoming RF data is sent to the host using the RX<string> asynchronous response. 4 LOCAL_COMMAND and on the RF side, the GPIO is automatically sent when there is a change in input. (digital I/O cable replacement mode)
S532	0	0..7	n/a	If non zero then on every connection, a SCO channel (audio) is initiated. Bit 0 HV1 Bit 1 HV2 Bit 2 HV3 When the connection is lost, the SCO channel disappears along with it.
S535	5	0..41	n/a	Link Supervision Timeout. If units go out of range, then a NO CARRIER message is sent to the host after the time specified here (as of v18.1.4.0: changed default value from 20 -> 5; applies to outgoing connections of all profiles)
S536	0	0..1	n/a	When set to 1, a remote device can 'capture' the AT parser of this unit by it sending this module an escape "!!!" sequence. The inter character timing is set via S Register 12. If S Register 507 is >= 2, then reading this register

Register	Default	Range	Category	Description
				always returns 0. Writing 1 results in ERROR 33.
S538	1	0..1	Pairing	If 1, then when a successful pairing occurs, it is automatically saved in the trusted device database - if it has room to store it.
S539	0	0..1	n/a	When set to 1, in idle mode (S512=1), UART Rx characters are discarded if DSR is de-asserted.
S541	20	-43..20	n/a	This sets the power level in dBm when inquiring or paging. Reading this register returns the value stored in non-volatile memory.
S542	4	-43..20	n/a	As per S541, however reading this register returns the current power level as set in the base band. The read can be different from S541because the actual power is set using a lookup table and the base band rounds down to the nearest value in the table.
S543	0	0..1	n/a	If this is set to 1, then incoming pairing attempts are accepted (if a pin code has been pre-entered using AT+BTK) while in the wait phase of auto connect cycle initiated by the AT+BTR command. In addition to accepting pairing attempts, if the pairing is successful, then the new device is automatically set as the peer address for automatic connections (as if an explicit AT+BTR command was entered). See S Register 505 and 530 also
S544	1	0..1	UART	Configure UART for either high throughput or low latency: 0 Low latency, low throughput 1 High latency, high throughput

<table>
<tr><th>Register</th><th>Default</th><th>Range</th><th>Category</th><th>Description</th></tr>
<tr><td>S551</td><td>$3211</td><td>$0..$ffff</td><td>n/a</td><td>
Bluetooth® RFCOMM specification allows digital I/O signals to be exchanged over an RFCOMM connection (RTR, RTC, DV, IC). This register specifies in each 4 bit nibble, how the outgoing modem status bits to the remote peer gets its value.
<table>
<tr><td>Nibble 0..3</td><td>Specifies the source for RTC (i.e., DSR/DTR)</td></tr>
<tr><td>Nibble 4..7</td><td>Specifies the source for RTR (i.e., RTS/CTS)</td></tr>
<tr><td>Nibble 8..11</td><td>Specifies the source for DV (i.e., DCD)</td></tr>
<tr><td>Nibble 12..15</td><td>Specifies the source for IC (i.e., RI)</td></tr>
</table>
Each nibble can take the following value:
<table>
<tr><td>0</td><td>Always set to 0</td></tr>
<tr><td>1</td><td>Always set to 1</td></tr>
<tr><td>2</td><td>If DCD is output, then always 1.
If DCD is input (S552), then follow DCD; 1 if DCD is asserted, otherwise 0.</td></tr>
<tr><td>3</td><td>If RI is output, then always 0.
If RI is input (S552), then follow RI; 1 if RI is asserted, otherwise 0.</td></tr>
<tr><td>4</td><td>If DSR is asserted, then 1. Otherwise, 0.</td></tr>
</table>
In the event that a nibble specifies DSR as the source of its state, be aware that if, S Register 507 is anything other than 0, a de-assertion of DSR causes the Bluetooth connection to be dropped.
If bits 0..3 and 4..7 are set to 0, then some Bluetooth devices may use that as a signal to stop sending any data. For example Nokia 6310 stops responding.
If this register is changed while in command and connected mode, then on going back online using the ATO command, a fresh signal is sent to the peer to update the bits.
</td></tr>
</table>

<table>
<tr><th>Register</th><th>Default</th><th>Range</th><th>Category</th><th>Description</th></tr>
<tr><td>S552</td><td>$0122</td><td>$0..$fff</td><td>n/a</td><td>This register specifies in each 4 bit nibble how the DTR, DCD and RI output pins are controlled by RFCOMM control bits (RTC,RTR,DV,IC) when in a Bluetooth connection.
<table>
<tr><td>Nibble 0..3</td><td>Specifies the source for DTR</td></tr>
<tr><td>Nibble 4..7</td><td>Specifies the source for DCD</td></tr>
<tr><td>Nibble 8..11</td><td>Specifies the source for RI</td></tr>
</table>
Each nibble can take the following value:
<table>
<tr><td>0</td><td>Do NOT touch the I/O. Configure DCD or RI as input at boot time.</td></tr>
<tr><td>1</td><td>Always de-assert.</td></tr>
<tr><td>2</td><td>Always assert.</td></tr>
<tr><td>3</td><td>If RTC bit in CONTROL_IND is 1, then assert. Otherwise, de-assert.</td></tr>
<tr><td>4</td><td>If RTR bit in CONTROL_IND is 1, then assert. Otherwise, de-assert.</td></tr>
<tr><td>5</td><td>If DV bit in CONTROL_IND is 1, then assert. Otherwise, de-assert.</td></tr>
<tr><td>6</td><td>If IC bith in CONTROL_IND is 1, then assert. Otherwise, de-assert.</td></tr>
</table>
If this register is changed while in command and connected mode, then on going back online using the ATO command, the modem output lines are refreshed.</td></tr>
<tr><td>S553</td><td>$0201</td><td>$0..$fff</td><td>n/a</td><td>This register specifies in each 4 bit nibble how the DTR,DCD and RI output pins are controlled when NOT in a Bluetooth connection.
<table>
<tr><td>Nibble 0..3</td><td>Specifies the source for DTR</td></tr>
<tr><td>Nibble 4..7</td><td>Specifies the source for DCD</td></tr>
<tr><td>Nibble 8..11</td><td>Specifies the source for RI</td></tr>
</table>
In addition it also refers to S Register 552 to see if the relevant pin is an input or not to be touched. If the nibble in 552 is 0, then the relevant pin is an input.

Each nibble can take the following value:
<table>
<tr><td>0</td><td>Always de-assert</td></tr>
<tr><td>1</td><td>Always assert.</td></tr>
<tr><td>2</td><td>Assert if RING is being sent to the host.</td></tr>
</table></td></tr>
<tr><td>S554</td><td>0</td><td>0..900</td><td>n/a</td><td>Post Reset Window: If S Register 512>=2 and <=7 then this register specifies a time in seconds for which the device stays in the S512 mode after power up or reset. On timeout, it aborts the discoverable and/or connectable and fall back into S512=1 mode, when it is</td></tr>
</table>

Register	Default	Range	Category	Description
				not connectable or discoverable. Note that if AT+BTR has been used to specify a peer device, then on reverting to mode 1, it attempts to make a connection to that peer device. A power cycle, reset via BREAK or ATZ is required to see the effects of change. Granularity: 4 | If value is ≤ 60 (round up). 60 | If value is > 60 (round down).
S555	1	1..7	n/a	If S Register 554 is nonzero, after the post reset window expires (defined by S554), the mode reverts to the mode specified in this register. This allows, for example, the device to be discoverable and connectable on power up (mode 4 or 7) and on window timer expiry to revert to connectable only (mode 3 or 6). A power cycle, reset via BREAK or ATZ is required to see effects of a change. In some firmware builds, S Registers 565 to 569 inclusive are visible, which allows the start-up mode to depend on the state of RI line (Setting S Reg 565 forces the RI pin to be configured as an input). For this feature to be active, set SReg 565 to 1. In that case, if RI is asserted on start-up, the start-up mode is defined by S Reg 568 and if de-asserted then S Reg 569.
S558	0	0..1	n/a	When 1, RING, NO CARRIER, and CONNECT responses are replaced by BTIN, BTDOWN, and BTUP respectively. This eliminates ambiguity when the module is connected to an AT modem which also gives these responses.
S559	4	0..7	n/a	This specifies a mask. When Bit 0 is 1, the response word ERROR is replaced by BTERR and OK is replaced by ok. When Bit 1 is 1, error responses do not include the error number and instead the error number can be retrieved using ATI12. As of firmware v18.1.4.0, when Bit 2 (0x04) is set, three-digit error code presentation is enabled (which is the new default setting).
S560	15	15..120	n/a	Disconnect timeout in seconds. This timer specifies how long to wait for confirmation from the peer device and/or the underlying stack that the connection has been successfully torn down. In the event that a confirmation never arrives, this timer 'closes off' the procedure and puts the machine back into a proper mode for new operations. Time is specified with 15 seconds intervals.

Register	Default	Range	Category	Description
S561	0	0..1000	n/a	Sniff Attempt Time in units of milliseconds. 0 means disable. See Sniff Mode and Figure 2-42.
S562	0	0..1000	n/a	Sniff Timeout Time in units of milliseconds. 0 means disable. See Sniff Mode and Figure 2-42.
S563	0	0..1000	n/a	Sniff Minimum Interval in units of milliseconds. 0 means disable. See Sniff Mode and Figure 2-42.
S564	0	0..1000	n/a	Sniff Maximum Interval in units of milliseconds. 0 means disable. See Sniff Mode and Figure 2-42.
S565	0	0..1	n/a	If set to 1, RI (Ring Indicate) line is configured as an input and forces the start-up mode (SReg512) and post-timeout on Start-up mode (SReg555) to be dependent on the state of RI. The RI conditional modes are defined by S Registers 566 to 569 inclusive.
S566	1	1..7	n/a	If S565=1, and RI is asserted then the device boots into this mode.
S567	1	1..7	n/a	If S565=1, and RI is de-asserted then the device boots into this mode.
S568	1	1..7	n/a	If S565=1, and RI is asserted then the device assume this mode after the post-start-up timeout (SReg 554) instead of mode defined in SReg555.
S569	1	1..7	n/a	If S565=1, and RI is de-asserted, the device boots into this mode after the post-start-up timeout (SReg 554) instead of the mode defined in SReg555
S581	0x9C	0..0xFF	HFP (HF)	Set HF supported features, Bitmask: Bit 0 Echo cancellation and/or noise reduction. Bit 1 Call waiting notification capability and three-way calling. Bit 2 CLIP presentation capability. Bit 3 Voice recognition activation. Bit 4 Remote volume control. Bit 5 Enhanced call status. Bit 6 Enhanced call control (not currently supported). Bit 7 Codec negotiation Default value: ▪ Codec negotiation – 0x80 ▪ Remote volume control – 0x10

Register	Default	Range	Category	Description
				▪ Voice recognition – 0x08 ▪ CLI presentation – 0x04 **Note:** Bit 0 - Bit 4 of these settings are advertised in the Service Record of the HF. Only CLIP presentation capability (Bit 2 = 0x04) is supported in the BTM module. After setting a new value to this S register, the commands AT&W and ATZ are required for new value to become effective.
S584	2	0..2	n/a	Enable/Disable SCO/eSCO When changing the unit returns ERROR 014, the device is either in or waiting for a connection. The new value cannot be accepted. If in a connection, drop it and then issue AT+BTX and then set the new value. If waiting, issue AT+BTX prior to setting the register. Value: 0: enable SCO 1: enable eSCO 2: for SPP-SCO incoming audio request: accept requested packet type (SCO/eSCO)
S588	0	0..1	n/a	After a disconnection, perform a cold reset.
S589	12	0..22	Audio	Set output gain level: See Gain Table; S689 is also affected.
S590	12	0..22	Audio	Set input gain level: See Gain Table; S690 is also affected.
S591	$0000	$0..$1ff		
S592	0	0..1	n/a	Set this to 1 to reduce the trusted device database to just one record when pairing auto-save is enabled via S reg 538.
S593	0	0..1	n/a	Automatically append last six digits of local Bluetooth address to the Friendly name set via AT+BTN or AT+BTF.

Register	Default	Range	Category	Description
S596	0x026C	$0..$3FF	HFP (HFG)	Set HFG supported features, Bitmask: Bit 0 – Three-way calling Bit 1 – Echo cancellation and/or noise reduction function Bit 2 – Voice recognition function Bit 3 – In-band ring tone capability Bit 4 – Attach a number to voice tag Bit 5 – Ability to reject call Bit 6 – Enhanced call status Bit 7 – Enhanced call control Bit 8 – Extended Error Result codes Default value: ▪ Codec negotiation – 0x200 ▪ Enhanced call status – 0x40 ▪ Incoming call reject – 0x20 ▪ In-band ring – 0x08 ▪ Voice recognition – 0x04 **Note:** Bit 0…Bit 4 of these settings are advertised in the Service Record of the HFG. The default value is 0x68 (Bit 3, Bit 5 and Bit 6 set). After setting a new value, the commands AT&W and ATZ are required for the new value to become effective. Ensure Bit 6 is always set. This is mandatory for HFP1.5 – AG.
S650	0	0..1	GPIO	GPIO pin state mask: 0 – No mask – enable configuration bit fields 1 – Enable I/O pin state Mask; disable configuration bit fields.
S651	Depending on alternative GPIO usage and wiring	$0..$ffff BTM51x	GPIO1	GPIO Configuration Register S650 must be set to 0 to enable configuration access. Controls pin state, pin direction, pin inversion, notification enable, function mapping select and function mapping code / av_operation_id. See Table 2-82.
S652			GPIO2	
S653			GPIO3	
S654			GPIO4	
S655			GPIO5	
S656			GPIO6	
S657			GPIO7	
S658			GPIO8	

Register	Default	Range	Category	Description
S669	$0000	$0..$ff	GPIO	Enable strong bias (strong pull up / pull down) mask if GPIO is configured as input. 0x0001: GPIO1; 0x0002: GPIO2; 0x0004: GPIO3; 0x0008: GPIO4 0x0010: GPIO5; 0x0020: GPIO6; 0x0040: GPIO7; 0x0080: GPIO8
S670	$000A	$0..$ff	GPIO	Read/Write all GPIOs in one atomic step (Write operation only affects GPIOs configured as outputs) 0x0001: GPIO1; 0x0002: GPIO2; 0x0004: GPIO3; 0x0008: GPIO4 0x0010: GPIO5; 0x0020: GPIO6; 0x0040: GPIO7; 0x0080: GPIO8
S689	-90	-450 ..215	Audio	Set output overall gain (dBr); See Gain Table; value must be entered (and is returned) multiplied by 10; S589 is also affected.
S690	-90	-450 ..215	Audio	Set input overall gain (dBr); See Gain Table; value must be entered (and is returned) multiplied by 10; S590 is also affected.
S730	0	0..1		
S731	0	0..255		
S744	OK	1..18,0.. 10		
S745	OK	1..18,0.. 10		
S1001 to S1010	0	0..2^32	n/a	Ten General Purpose 32 bit Registers for use by host. These are stored in non-volatile memory.

3.2 ATI Commands

The following table lists all ATIn parameters supported by the modules. ATI commands provide general information about the modules and status information.

Table 3-2: BTM ATI Commands

Register	Description
ATI0	The product name/variant.
ATI1	The CSR firmware build number.
ATI2	The AT firmware build number. For internal use only.
ATI3	The AT firmware revision.
ATI4	A 12 digit hexadecimal number corresponding to the Bluetooth address of the module.
ATI5	The manufacturer of this device.
ATI6	The maximum size of trusted device database.
ATI7	The manufacturer of the Bluetooth chipset.
ATI8	The chipset format.
ATI9	SPP connection status: 0 Not connected 1 Connected in local command mode 2 Connected in remote command mode
ATI11	The reason why a NO CARRIER resulted in the most recent attempt at making an outgoing connection. The response values are as follows: 0 No prior connection 1 Connection timeout 2 Connection attempt cancelled 3 Normal disconnection 4 Peer device has refused connection 5 Service profile <uuid> requested not available on remote device 6 Connection has failed 32 ATH was entered 33 Incoming connection aborted because of too many rings. 34 Unexpected incoming connection. 35 Invalid address. 36 DSR is not asserted. 37 Call limit of 65531 connections has been reached. 38 Pairing in progress. 39 No link key. 40 Invalid link key. 41 Link loss (not working as expected for SPP). 42 No service-level connection (SLC). 43 Invalid parameters (A2DP) 44 Remote link key missing (not working as expected for SPP)

Register	Description	
	255	Unknown reason.
ATI12	The last ERROR response number.	
ATI13	The Sniff status is returned as follows:-	
	Response:	<cr,lf>a:b,c,d,e<cr,lf>OK<cr,lf>
	a	0 when not online and 1 when online and sniff has been enabled.
	b	Sniff Attempt parameter.
	c	Sniff timeout parameter.
	d	Minimum sniff interval.
	e	Maximum sniff interval.
	All parameters b, c, d, and e are given as Bluetooth slots which are 625 ms long converted from values of S Registers 561, 562, 563 and 564 respectively.	
ATI14	The current boot mode	
ATI15	The maximum length of an AT command, not including the terminating carriage return	
ATI16	Codec Output Maximum Gain Range	
ATI17	Codec Input Maximum Gain Range	
ATI18	Bluetooth version	
ATI19	Audio connection status:	
	0	Off
	1	On
ATI20	Returns the number of pending bytes in the RF buffer during a connection.	
ATI21	The index of the profile instance owning an active SCO connection:	
	0	None
	1	HS
	2	HSG
	3	HF
	4	HFG
	5	SPP_BTA
ATI22	Last SCO output gain	
ATI23	Last SCO input gain	
ATI24	Last A2DP output gain	
ATI25	Last A2DP input gain	

<table>
<tr><th>Register</th><th>Description</th></tr>
<tr><td>ATI26</td><td>Display current A2DP decoder info (if S333=1):
<CodecTypeString>,<SysIdhex>,<BuildNohex>,"<PostProcFriendlyName>","<SdkName>"
CodecTypeString = SBC/AAC/APTX
SysId, BuildNo = 4 hex digits
Example: SBC,E006,0101,Music,ADK2.0
When A2DP is connected but not streaming, the extended data is not available. Only the 'CodecTypeString' (e.g. SBC) is printed in this case.
If S333=0, only the ID of current decoder type is printed:
<table>
<tr><td>0</td><td>None</td></tr>
<tr><td>1</td><td>SBC</td></tr>
<tr><td>2</td><td>APTX</td></tr>
<tr><td>3</td><td>AAC</td></tr>
</table></td></tr>
<tr><td>ATI27</td><td>Current scan state:
<table>
<tr><td>0</td><td>Not discoverable and not discoverable (not scanning)</td></tr>
<tr><td>1</td><td>Discoverable (inquiry scanning)</td></tr>
<tr><td>2</td><td>Connectable (page scanning)</td></tr>
<tr><td>3</td><td>Discoverable and connectable (inquiry- and page-scanning)</td></tr>
</table></td></tr>
<tr><td>ATI28</td><td>Current audio sampling rate for input and output in Hz</td></tr>
<tr><td>ATI29</td><td>Maximum EIR data size in bytes</td></tr>
<tr><td>ATI30</td><td>RAM buffer length in bytes</td></tr>
<tr><td>ATI31</td><td>Length of EIR baseband buffer</td></tr>
<tr><td>ATI32</td><td>Query CVC license key</td></tr>
<tr><td>ATI35</td><td>Compact list of available CVC images.
Appended variant flags, should always be 7:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>STD</td></tr>
<tr><td>1</td><td>BEX</td></tr>
<tr><td>2</td><td>WBS</td></tr>
</table>
<b>Note:</b> For a detailed list of CVC available images, see ATI318.</td></tr>
<tr><td>ATI36</td><td>Status of boot-time CVC license check, applies if S318 &gt; 0</td></tr>
<tr><td>ATI37</td><td>Query SCO link type: 0=unknown/no SCO link, 1=SCO, 2=eSCO</td></tr>
<tr><td>ATI38</td><td>Display Type, SysID, BuildNo, FriendlyName and CSR SDK of all available A2DP decoders capable of post processing.
Syntax:
<CodecTypeString>,<SysId$_{hex}$>,<BuildNo$_{hex}$>,<PostProcFriendlyName>,<ADK-version>
<b>Note:</b> ATI38 only works when the BTM51x does not have any connection. If A2DP is in streaming mode (sink role), the current decoder can be queried by ATI26</td></tr>
</table>

<table>
<tr><th>Register</th><th>Description</th></tr>
<tr><td>ATI39</td><td>Query current DSP image plus short string descriptor.
<DspImageID>,<ImageName>,<ScoType>
Examples:
13,x1mw,eSCO#wideband speech
11,x1m, eSCO #narrowband/CVSD
00,n/a #no SCO audio link nor DSP image currently active</td></tr>
<tr><td>ATI42</td><td>State information. The response values are as follows:
<table>
<tr><td>13</td><td>NotOpen</td></tr>
<tr><td>14</td><td>OpenIdle</td></tr>
<tr><td>15</td><td>Ringing</td></tr>
<tr><td>16</td><td>OnlineCommand</td></tr>
</table>
172 to 177 indicate waiting for connectable and/or discoverable where the third digit equates to the value stored in S Register 512 or 555.</td></tr>
<tr><td>ATI44</td><td>Current power mode (active/sniff/passive) and sniff interval of SPP/HSP/HFP.
Response format: <power_mode>,<sniff_interval_in_slots_decimal>
<table>
<tr><td>0</td><td>Active (highest power consumption, lowest latency/responsiveness)</td></tr>
<tr><td>1</td><td>Sniff mode (power consumption and latency depend on sniff interval)</td></tr>
<tr><td>2</td><td>Passive (the device does not initiate a change of the power mode)</td></tr>
</table></td></tr>
<tr><td>ATI45</td><td>Current sniff sub-rating (SSR) parameters of SPP/HSP/HFP link in slots (1 slot = 0.625 ms).
Response format: <flag>,<li>,<ri>,<lto>,<rto>
<table>
<tr><td>&lt;flag&gt;</td><td>▪ 1 – SSR active
▪ 0 – SSR not active</td></tr>
<tr><td>&lt;li&gt;</td><td>Local SSR interval, decimal integer, slots</td></tr>
<tr><td>&lt;ri&gt;</td><td>Remote SSR interval, decimal integer, slots</td></tr>
<tr><td>&lt;lto&gt;</td><td>Local timeout, decimal integer, slots
Time before changing from sniff mode to SSR when no data is being transmitted.</td></tr>
<tr><td>&lt;rto&gt;</td><td>Remote timeout, decimal integer, slots
Time before changing from sniff mode to SSR when no data is being transmitted.</td></tr>
</table></td></tr>
<tr><td>ATI46</td><td>Current link policy power table.
Response for each row: [<index>]<mode>,<time>,<min_int>,<max_int>,<attempt>,<timeout>
<table>
<tr><td>&lt;index&gt;</td><td>Row index, starting with 0</td></tr>
<tr><td>&lt;mode&gt;</td><td>Power mode:
▪ 0 = Active
▪ 1 = Sniff
▪ 2 = Passive</td></tr>
<tr><td>&lt;time&gt;</td><td>Duration of current row in seconds (switching to the next row after this timeout).</td></tr>
<tr><td>&lt;min_int>
<mas_int&gt;</td><td>Sniff mode intervals in slots (S563/S564, M)</td></tr>
<tr><td>&lt;attempt&gt;</td><td>Number of active slots at the beginning of the sniff interval (S561,</td></tr>
</table></td></tr>
</table>

Register	Description	
		N).
	<timeout>	Number of additional active slots (S562, T) if data is exchanged.
ATI54	Query remote Bluetooth device addresses of all connected profiles. Response per profile/line: \$<profile_mask$_{\text{S102\_hex}}$>,<BdAddr>	
ATI55	Query RSSI value for all connected profiles. Response per profile/line: \$<profile_mask$_{\text{\_S102\_hex}}$>,<RSSI$_{\text{\_signed\_decimal\_with variable\_length}}$>	
ATI56	Query local HCI role (master/slave) for all connected profiles. Response per profile/line: \$<profile_mask$_{\text{\_S102\_hex}}$>,<'M'/'S'>	
ATI57	Query profile flags of current connections, same format as S register 102	
ATI58	Query number of current connections, total sum of ATI60..ATI67	
ATI59	Returns '1' if a pre-set PIN code (by AT+BTK="...") is available (legacy pairing). Returns '0' otherwise. The PIN code is not displayed for security reasons.	
ATI60	SPP connection status:	
	0	Not connected
	1	Connected. Identical with AT19
ATI61	A2DP connection status:	
	0	Not connected
	1	Connected and streaming
	2	Connected but not streaming (suspended)
ATI62	AVRCP connection status: 0 = not connected, 1 = connected.	
ATI63	HSP-Headset connection status: 0=not connected,1=ACL connected, 2=audio connected.	
ATI64	HSP-Gateway connection status: 0=not connected,1=ACL connected, 2=audio connected.	
ATI65	HF connection status:	
	0	Not connected
	1	SLC connected
	2	Audio connected
	3	In call, SLC
	4	In call, audio
	5	In call setup (incoming/dialling/alerting), SLC
	6	In call setup (incoming/dialling/alerting), audio
ATI66	HFG connection status:	
	0	Not connected
	1	SLC connected
	2	Audio connected
	3	In call, SLC
	4	In call, audio
	5	In call setup (incoming/dialling/alerting), SLC
	6	In call setup (incoming/dialling/alerting), audio
	8	In call but not SLC
ATI72	Table of Auto Connect Records (persistent store). Syntax is as follows:	

<table>
<tr><th>Register</th><th>Description</th></tr>
<tr><td></td><td>[<index>],<BdAddr>,$<pm>,<interval>,<state>,$<pm_cnct>,$<flags>
With
<table>
<tr><td><index></td><td>Index of the entry, starting with 1</td></tr>
<tr><td><BdAddr></td><td>Bluetooth device address of remote device</td></tr>
<tr><td><pm></td><td>Profile mask as per S102, four hexadecimal characters with leading $</td></tr>
<tr><td><interval></td><td>Interval of reconnection attempts in seconds</td></tr>
<tr><td><state></td><td>Status of the ACR:
<table>
<tr><td>0</td><td>Attempting connection (at every interval seconds)</td></tr>
<tr><td>1</td><td>Partly connected (at least one profile of pm connected, at least one profile of pm not connected, attempting to connect the remaining profiles)</td></tr>
<tr><td>2</td><td>Fully connected (all profiles of the ACR connected)</td></tr>
</table></td></tr>
<tr><td><pm_cnct></td><td>Profile mask of actually connected profiles. As with S102, this value is four hexadecimal characters with leading $</td></tr>
<tr><td><flags></td><td>Option flags, two hexadecimal characters with leading $, bitmask:
<table>
<tr><td>Bit 0</td><td>Always reconnect: If this bit is set, a normal disconnection yields reconnect attempts. If bit is cleared, the record is deleted from the ACR table on normal disconnect. Link loss leaves the entry in the ACR-table, yielding reconnect attempts.</td></tr>
<tr><td>Bit 1</td><td>Always send AVRCP play.</td></tr>
<tr><td>Bit 2</td><td>Never send AVRCP play.</td></tr>
</table></td></tr>
</table></td></tr>
<tr><td>ATI73</td><td>Auto Connect Service (ACS) status:
<table>
<tr><td>0</td><td>Service disabled / not running.</td></tr>
<tr><td>1</td><td>Service enabled, paused between connection attempts, interval timer running.</td></tr>
<tr><td>2</td><td>Attempting: initiates connections as defined by ACR table.</td></tr>
<tr><td>3</td><td>Attempt repeat: resumes connection attempts after a profile with transitional state was found and the transitional state was finished.</td></tr>
<tr><td>4</td><td>All profiles of ACR-table are connected, monitors disconnect events.</td></tr>
</table></td></tr>
<tr><td>ATI74</td><td>Auto connect attempt interval in seconds. If the interval parameter differs across multiple ACRs without all profiles connected yet, the smallest interval applies and is displayed.
If not applicable (e.g. no ACRs exist), response is ‘0’.</td></tr>
<tr><td>ATI75</td><td>Current incoming peer address, regardless of how it was set (AT+BTM<bd_addr> or AT+BTG/P<bd_addr>)
If not set, response is ‘000000000000’.</td></tr>
<tr><td>ATI80</td><td>AVRCP 1.5 diagnostics:
<extensions>,<features>,<event_caps>,<registered_evts>,<player_features>
Intended for issue tracking</td></tr>
<tr><td>ATI101</td><td>The RSSI value in dBm. If a connection does NOT exist then a value of -32786 is returned.
A value of 0 means the RSSI is within the golden range this is quite a large band, therefore RSSI is not always a useful indicator. Use ATI111 instead which returns the bit error rate.</td></tr>
</table>

Register	Description
ATI111	Returns LinkQual which in the CSR chipset is defined as BER (bit error rate). This returns a value which is the number of erroneous bits out of 1 million. Hence a value of 0 is best, and larger values are worse. A value near 1000 (BER = 0.1%) indicates bad link quality with a high number of lost Bluetooth packets.
ATI200	Manufacturing data (e.g. module serial number, manufacturing date)
ATI318	Detailed list of available CVC images. ▪ Format – See ATI35 ▪ Variant IDs: – S = Standard (8 kHz) – B = Bandwidth extended (BEX) – W = WBS (16 kHz)
ATI333	Full AT firmware version number
ATI350	detailed list of available CVC images, duplicate of ATI318
ATI411	Short press duration in ms (S411)
ATI412	Medium press duration in ms (S411+ S412)
ATI413	Long press duration in ms (S411+ S412 + S413)
ATI144	Power mode and sniff interval in milliseconds (ms). Like ATI44, but the sniff interval is in milliseconds instead of slots. (Rounding is indicated by **!**)
ATI145	Like ATI45, but all values are in milliseconds instead of slots. (Rounding is indicated by **!**)
ATI146	Like ATI46, but all values are in milliseconds instead of slots. (Rounding is indicated by **!**)

3.3 Error Responses

As of firmware v18.1.4.0, three-digit error codes are enabled by default. Refer to section 2.9.19.

Table 3-3: BTM Error Responses

Error	Description
001	Register not recognised
002	Value for register is out of range
003	Incoming call NOT pending
004	No call to connect to. This error code has meaning for ATO only
005	Syntax Error
006	Empty String
007	Device Class could not be stored
008	Invalid Device Class Code
009	Invalid Bluetooth Address
010	Could not set Service or Friendly name
011	PS Store Write
012	PS Store Read
013	Not Idle
014	Incorrect Mode
015	Already Scanning
016	Pairing is already in progress
017	Not USED
018	Not USED
019	Not USED
020	Not safe to write to Non-volatile Store - Ongoing Bluetooth Connection
021	Link Key Cache is Empty
022	Link Key Database is Full
023	Malloc returned NULL - Resource Issue
024	Remote Address same as Local Address
025	Connection Setup Fail, DSR Not asserted
026	Unauthenticated licence
027	Max Responses too high. See S Register 518. Memory allocation error
028	Pin in AT+BTK is too long
029	Invalid Ring count for S Register 0 or 100. If S0<>0 and S100<>0 then S0 must be < S100
030	ADC Error
031	Analogue Value cannot be read. It is set for output.
032	Analogue Value cannot be written. It is set for input.
033	S Register Value is invalid
034	Both L and R modifier cannot be specified in ATD command
035	Invalid Major Device Class – valid value in range 0x00 to 0x1F inclusive

Error	Description
036	Pairing in progress – Command cannot be actioned – try again later
037	Invalid Sniff parameter specified. E.g. new Attempt value greater than MinInterval. Solution is to first increase MinInterval and re-enter the Attempt value.
038	Get Remote Friendly name Failed
039	Failed to change mode to Multipoint
040	7 Bit mode requires parity to be even or odd
041	Stream Error
042	Stream Pending Error
043	Unknown Audio Gateway Command
044	Busy, try later
045	Command or operation not allowed
046	No A2DP role has been set (see S register 300)
047	No AVRCP role has been set (see S register 301)
048	No AVRCP category has been set (see S register 302)
049	No AVRCP control connection
050	No A2DP or AVRCP connection currently incoming
051	Invalid operation ID (AVRCP)
052	Wrong AVRCP role
053	Command disabled by S-Register 310
054	No manufacturing information available
055	Audio resource error
056	Invalid UUID
057	Maximum gain level reached
058	Minimum gain level reached
059	Profile or role not enabled
060	Profile under construction
061	Unknown Headset command
062	Unknown Hands-free command
063	Incorrect state
064	not used
065	UART resource error
066	Index of subscriber number record too large
067	Maximum number of subscriber number records reached
068	No SLC exists for AT+HFGB0/1
069	In-band ringing was not enabled in HFG supported features (S596, 0x08) at boot time
070	Invalid number type, must be in range 128 <= number type <= 175
071	Maximum number of calls reached
071	Call state issue

Error	Description
073	A second waiting call is not is not allowed
074	No call held
075	Invalid index <idx>
076	Memory allocation attempt was not successful
077	Incorrect A2DP state (see ATI61)
078	Request rejected due to priority settings (S355)
079	Writing to modem control line is not permitted by GPIO S-register
080	Attempting to write the pin state of a GPIO that is configured as input
081	Maximum size of EIR data exceeded (ATI29)
082	No CVC license key stored or more than 5 words (invalid length, ATI32)
083	CVC security failed (on loading CVC image for a SCO connection)
084..087	firmware internal CVC error codes, should not occur with released firmware
088	CVC SYSID request timeout. Can occur on ATI35 if UART baud rate is too low.
089	HF request to disable NREC in HFG failed, because NREC feature is not supported by both ends. Support for both ends can be checked with AT+HFF? (NREC mask for HF=1; NREC mask for HFG=2) of AT+HFN?
090	CVC boot time check failed: occurs if S318>0 at boot time and selected CVC image does not have a valid license key.
091	HF: "ERROR" was received from the HFG on a DTMF request ('AT+HFMi' on UART = 'AT+VTS=i' on SLC)
092	AT+BTAx is not permitted in current state. Occurs if one of HF/HFG/HS/HSG and SPP is connected to one peer device. A SCO audio link of HFP/HSP has priority over SPP-SCO, so the BTM51x doesn't accept this command in this scenario.
093	A2DP not in streaming state or not in A2DP sink role (decoder). The preconditions for "AT+APMx" are to be in A2DP sink role (S300=1) and in a streaming state.
094	Command not supported by current A2DP decoder variant
095	DSP plugin doesn't support 'query mode'
096	DSP busy
097	Profile is busy with a current request
099	Real error code is greater than 99 – query ATI12 to obtain the detailed error code
100	ACR table not available in RAM
101	ACR - a record for the same BdAddr already exists in persistent store and at least one flag of that record's profile mask overlaps with the profile mask of the scratch record
102	ACR – write to persistent store was not successful
103	ACR – the history index passed doesn't exist
104	ACR is invalid (e.g. BdAddr=000000000000 or profile mask=0)
105	AC-service stopped by safety timeout (in status 'attempt')
106	ACR table empty, AC-service not started
107	AC-service must be stopped for modification of ACR-table
108	Missing profile mask in S102: at least one profile flag contained in the ACR table is not set in S102.

Error	Description
109	Command not allowed when AC-service is running. Stop service to enable full parser
110	ACS precondition failed - module is discoverable and/or connectable
111	ACS precondition failed - inquiry in progress
112	ACS precondition failed - pairing in progress
113	ACS precondition failed - legacy SPP auto connect (AT+BTR) is enabled
114	ACR option flags wrong
115	Tuple length doesn't match (DREG)
116	Internal error code (DREG)
117	Non-volatile memory for dynamic register exhausted (DREG)
118	Maximum number of tuples exceeded (DREG)
119	Maximum dynamic register data length exceeded (DREG)
120	Internal error code (DREG)
121	Internal error code (DREG)
122	Operation not enabled in HF/HFG supported features. Check S581/ S596
123	Internal error code
124	Internal error code
125	Operation timeout
126	Requested operation not enabled in remote device's supported features (HF/HFG)
127	Call records are empty
128	No call is currently waiting
129	Invalid remote gain cache
130	Internal error code
131	AVRCP, reserved
132	AVRCP, reserved
133	AVRCP, requested action or feature not supported by remote device
134	AVRCP, operation failed (internal issue)
135	AVRCP, received AVRCP status code *not successful*
136	AVRCP, browsing not supported
137	AVRCP, not connected
138	AVRCP, parameter mismatch or out of range (e.g. start offset > end offset)

3.4 Status IDs (Automation)

Table 3-4 lists status IDs. These are used with D-Regs 744,745.

Table 3-4: Status IDs

Status ID	Status
1	Not connectable, Not discoverable
2	Discoverable
4	Connectable
8	Discoverable, Connectable
16	Connected (any profile)
17	AudioOn (any audio type)

3.5 LED Blink Pattern IDs (Status Indication)

Table 3-5 lists blink pattern IDs. These are used with AT+SILx=y or D-Regs 744,745.

Table 3-5: Blink Pattern IDs

Pattern ID	LED blink pattern
0	LED off
1	LED on
2	Blink fast
3	Blink medium
4	Blink slowly
5	Flash fast
6	Flash medium
7	Flash slowly
8	Pulse fast
9	Pulse medium
10	Pulse slowly

3.6 List of UUIDs

Table 3-6 provides a list of selected UUIDs. For a complete list refer to the "Assigned Numbers – Service Discovery (SDP)" document [8] by the Bluetooth SIG.

Table 3-6: Selected UUIDs

UUID	Mnemonic / Profile	Role
0x1101	Serial Port Profile (SPP)	-
0x1102	LAN access using PPP	-

UUID	Mnemonic / Profile	Role
0x1103	Dialup Networking (DUN)	-
0x1105	OBEX Object Push	-
0x1106	OBEX File Transfer	-
0x1108	Headset Profile (HSP)	Headset
0x110A	A2DP	Audio Source
0x110B	A2DP	Audio Sink
0x110C	AVRCP	Remote Target
0x110D	A2DP	-
0x110E	AVRCP	-
0x110F	AVRCP	Remote Controller
0x1112	Headset Profile	Audio Gateway
0x111E	Hands-free Profile (HFP)	Hands-free unit
0x111F	Hands-free Profile (HFP)	Audio Gateway

3.7 Profile Mask (as per S102)

Table 3-7 below lists the profile mask as defined per S102

Table 3-7: Profile mask as per S102

Profile	Mask (hexadecimal)
SPP	$0001
HS	$0002
HSG	$0008
HF	$0010
HFG	$0040
A2DP	$0080
AVRCP	$0100

3.8 References

[1] "Bluetooth Specification Version 2.1 + EDR [vol3]", 26 July 2007
https://www.bluetooth.org/Technical/Specifications/adopted.htm
(click on "Core Specification v2.1 + EDR")

[2] "Advanced Audio Distribution Profile Specification" Rev. V12, 16/04/2007
https://www.bluetooth.org/Technical/Specifications/adopted.htm
(scroll down to section 'Traditional Profiles (Qualifiable)' -> A2DP adopted version 1.2)

[3] "Audio/Video Remote Control Profile" Revision 1.0 22/05/2003
https://www.bluetooth.org/Technical/Specifications/adopted.htm
(scroll down to section 'Traditional Profiles (Qualifiable)' -> AVRCP adopted version 1.0)

[4] "Headset Profile" Revision V12r00, 18/12/2008
https://www.bluetooth.org/Technical/Specifications/adopted.htm
(scroll down to section 'Traditional Profiles (Qualifiable)' -> HSP adopted version 1.2)

[5] "Hands-free Profile 1.6" Revision V V16r00, 10/052011
https://www.bluetooth.org/Technical/Specifications/adopted.htm
(scroll down to section 'Traditional Profiles (Qualifiable)' -> HFP adopted version 1.6)

[7] "Serial Port Profile" Specification
https://www.bluetooth.org/Technical/Specifications/adopted.htm
(scroll down to section 'Traditional Profiles (Qualifiable)' -> SPP adopted version 1.1)

[8] "Bluetooth Assigned Numbers"
https://www.bluetooth.org/Technical/AssignedNumbers/home.htm
Most interesting are the links 'Baseband' and 'Service Discovery Protocol'

[9] Class of Device Generator: this link might be helpful for creating a particular CoD
http://bluetooth-pentest.narod.ru/software/bluetooth_class_of_device-service_generator.html
Caution: this tool allows selection of more than one minor device classes, so make sure that only one minor device class is select and verify the result with [8].

[10] "Bluecore 5-Multimedia External" Data Sheet, Cambridge Silicon Radio (CSR)
http://www.csrsupport.com (log in or new account required)

[11] "Bluecore 4 External" Data Sheet, Cambridge Silicon Radio (CSR)
http://www.csrsupport.com (log in or new account required)

[12] "RFCOMM with TS 07.10" specification
http://www.bluetooth.com/Specification%20Documents/rfcomm.pdf

[13] "CVC on BTM5xx" application note, provided by Laird Technologies under NDA

[14] "FW_ReleaseNote_Btm51x_v18.1.3.0", Doc No: BTM51xv18.1.3.0
Information guide for Production and Engineering releases of firmware for part ~ BTM510 / BTM511.

[15] "BTM511 Development Kit Quick Start HFPv2"

[16] "BTM511 Development Kit Quick Start A2DP and AVRCP-v2"

[17] "BTM511 Development Kit Quick Start SPP-v2"

4 Related Documents and Files

The following additional BTM510/511 technical documents are available from the Embedded Wireless Solutions Support Center: https://laird-ews-support.desk.com/?b_id=1957

- BTM510/511 Product Brief
- BTM510/511 User Manual

Firmware

- BTM510/511 Firmware Files
- BTM510/511 Firmware Release Notes - Version 18.1.4.0

Development Kit Schematics

- BTM510 Development Kit Schematics
- BTM511 Development Kit Schematics
- BTM511 Development Kit Schematics - Version 4

Development Kit Documentation

- BTM511 Development Kit Quick Start A2DP and AVRCP - Version 2
- BTM511 Development Kit Quick Start HFP - Version 2
- BTM511 Development Kit Quick Start SPP - Version 2
- BTM511 Audio Development Kit (ADK) User Manual - Version 4

The following download is also available from the Embedded Wireless Solutions Support Center: https://laird-ews-support.desk.com/?b_id=1957

Laird (EZURiO) Terminal v6.9.0.zip

Product information can also be accessed from the BTM51x product page on the Laird website: http://www.lairdtech.com/products/btm51x-series

global solutions: local support ™
USA: +1.800.492.2320
Europe: +44.1628.858.940
Asia: +852.2268.6567
wireless.support@lairdtech.com
www.lairdtech.com/wireless

Laird Technologies is the world leader in the design and manufacture of customized, performance-critical products for wireless and other advanced electronics applications. Laird Technologies partners with its customers to find solutions for applications in various industries such as:

- Network Equipment
- Telecommunications
- Data Communications
- Automotive Electronics
- Computers
- Aerospace
- Military
- Medical Equipment
- Consumer Electronics

Laird Technologies offers its customers unique product solutions, dedication to research and development, as well as a seamless network of manufacturing and customer support facilities across the globe.

LWS-UM-BTM510-511 0312

All brands and product names in this publication are registered trademarks or trademarks of their respective holders. This material is preliminary Information furnished by Laird Technologies in this specification is believed to be accurate. Devices sold by Laird Technologies are covered by the warranty and patent indemnification provisions appearing in its Terms of Sale only. Laird Technologies makes no warranty, express, statutory, and implied or by description, regarding the information set forth herein. Laird Technologies reserves the right to change specifications at any time and without notice. Laird Technologies' products are intended for use in normal commercial and industrial applications. Applications requiring unusual environmental requirements such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional testing for such application.

Limited Warranty, Disclaimer, Limitation of Liability