Installers Guide to the Bose Serial Interface for Bose AV20, AV35 and AVM Home Theater Control Consoles

Document Version 1.04

Issued June 28, 2011

Summary: An interface exists in the rear of Bose AV20, AV35 and AVM control consoles which allows some control of the console by an external computer. This document describes the current capabilities and the format of the interactions over that interface for **Version 01.05.00 or later of the console software.**

TABLE OF CONTENTS

LIST OF TABLES

1 Introduction

This document specifies how to use the serial communications interface to control the following control consoles: Bose AV20, AV35 and AVM control consoles. The AV20 is currently used in Lifestyle T10/T20 systems. The AV35 is currently used in the Lifestyle 235, V25 and V35 systems. The AVM is currently used on the Bose VideoWave System.

The interaction between a computer and the Bose control console consists of *Commands*, *Responses*, and event *Notifications*.

In previous generation Bose Lifestyle products, the interactions were sequences of printable ASCII bytes which could be sent and received using a Terminal Emulator program.

The new control consoles interact with an external computer using a binary protocol, which consists of sequences of bytes that transmit data which is not printable characters. The sequences of bytes are structured and individual bytes generally hold numeric data. Thus the interaction requires an application program on the computer. The rules and formats that describe this interaction are called a ***protocol***, which is described in this document.

1.1 Controlling the System

The system is normally controlled using key presses on the remote controls. The system can also be controlled using the serial communication interface on the rear of the console and the protocol described in this guide. The primary means of controlling the system is by simulating user key presses using the Key Press command (see section 6.4). Other commands can query and/or modify the states of the system.

The states of the system that can be set or queried are listed below. Later sections describe the physical connection, the general format of the messages (packets) sent, the rules describing the order and timing of these packets, and the specific packet format of the *Commands*, *Responses*, and *Notifications*.

1.2 States of the System

Systems based on the AV20, AV35 or AVM consist of a Console (referred to hereafter as “the Console”) and powered speakers in the Main Room, also known as Room A. The AV35 also supports up to 14 separately controllable Bose® link room speakers (Rooms B-O).

The states of the system that can by queried or modified are listed in Table 1-1 System States. The commands to query or set these states are listed in section 6 Client Commands and Console Responses. The value for each state is described with the Command which sets or modifies the value.

In addition to these states, several types of events can occur in the Console. The Console alerts the Client to these events by sending Notifications. (see section 5.3.4 Notification Packets).

Table 1-1 System States

State	Value	Command to Set	Command to Query	Available in Low Power Mode
Main Room Speakers On	True/False	KP	RoomStatus	Yes
Main Room Speakers Muted	True/False	KP	RoomStatus	Yes
Main Room Speaker Source	Source Enumerators	KP	SR	Yes
Main Room Speaker Volume	0-100	KP, VO	VO	Yes
Bose® link Room Speakers On	True/False	KP	RoomStatus	Yes
Bose® link Room Speakers Muted	True/False	KP	RoomStatus	Yes
Bose® link Room Speaker Stream	Zone2/Local	KP	RoomStatus	Yes
Bose® link Room Speaker Volume	0-100	KP	RoomStatus	Yes
Zone1 Source	Source Enumerators	KP	SR	Yes
Zone 2 Source	Source Enumerators	KP	SR	Yes
Lower Power Mode	True/False	*TimeOut*	SysRdy	Yes
Hard Key Press Notifications	True/False	Echo	Echo	Yes
Soft Key Press Notifications	True/False	Echo	Not Available	No
Version	Text	*Read Only*	VR	No
Tuner Station	0-65535	TS	QS	Yes
Tuner RDS Name	Text	*Read Only*	RDS PS	Yes
Tuner RDS Text	Text	*Read Only*	RDS RT	Yes
Tuner Current Preset	0-25	*Read-Only*	SysInfo	No
Time (msec) since Boot	0- 4294967295	*Timer*	Uptime	Yes

2 Definitions

Table 2-1 Definitions

Name	Meaning
Client	The external computer and software controlling the interaction with the Bose product.
Console	The AV20, AV35 or AVM Control Console
Packet	A series of bytes with a well defined structure. The Client and Console communicate by sending packets. The format of each packet is described in section **4** Packet Format.
LSB	Least significant byte. The byte of a numeric field which holds the ***smallest*** portion of the number. For example the number 300 decimal is represented in hexadecimal as 0x012c. The least significant byte is 0x2c.
MSB	Most significant byte. The byte of a numeric field which holds the ***largest*** portion of the number. For example the number 1000 decimal is represented in hexadecimal as 0x03e8. The most significant byte is 0x03.

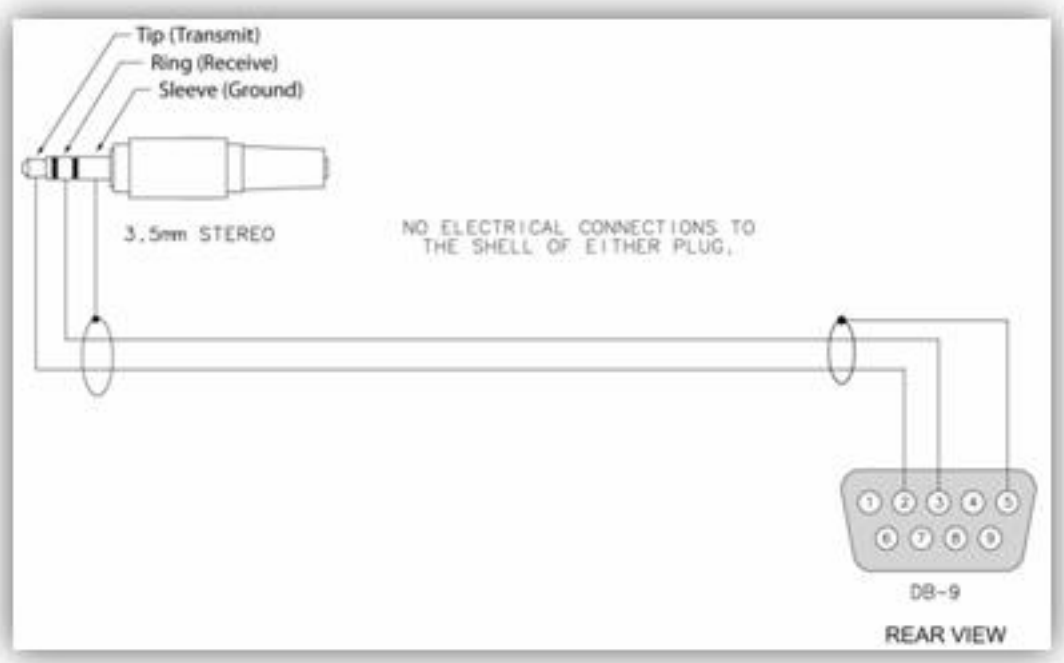
3 The Physical Connector

The interface is a 3.5mm stereo connector located on the rear of the Console at the same height as the power connector. The cable that connects the computer (for example a PC) to the Console connects the receive, transmit, and ground wires of a DB-9 connector to the tip, ring, and ground connections of the Console connector.

The DB-9 male connector (e.g. on a PC) has pins labeled 1-5 in the top row and 6-9 in the bottom row. Pin 2 (Receive) goes to tip, Pin 3 (Transmit) goes to Ring, and pin 5 (Ground) connects to sleeve. See cable figure 1 below.

The physical signaling uses the standard RS232 serial communication electrical specifications with communication parameters of: 19200 Bits per second, 8 Data bits, No parity, 1 Stop bit, and No Flow control. Note that the Console voltage is zero when the communications link is not actively transmitting a packet.

Figure 1 The Serial Connection Cable



4 Packet Format

All interactions between the Client and the Console are in the form of complete packets.

A Packet always begins with a Header field which contains the Length, Status, and Operation Code. The Payload field contains any required parameters. Packets from the Client must also include a Checksum field.

<Header> [<Payload>] <Checksum> Client to Console

<Header> [<Payload>] Console to Client -- Note: no Checksum

Note: Packets are NOT terminated with a Carriage Return or Line Feed.

4.1 Packet Fields

Table 4-1 General Packet Format

Byte Number	Name	Number of Bytes	Description
0	Length	1	Length of packet including this byte and the Checksum byte (if present)
1	Status	1	Packet Flags. (Table 4-2 Packet Status Flags.)
2-3	Opcode	2	Operation to be performed
4—	Payload	Variable	Contains OpCode parameters if needed
Last	Checksum	1	Contains a checksum. Only sent by Client.

4.1.1 Length

Count of the total number of bytes in this packet including this Length byte, the Status, OpCode, and Payload if present, and Checksum if present.

4.1.2 Status

The status byte consists of single bit flags. The bits of this byte are numbered from 7 the most significant bit, to 0 the least significant bit. This byte must be set to 0 when sent by the Client. When received by the Client, these flags are interpreted as indicated in Table 4-2 Packet Status Flags.

Only two bits in the Status byte are relevant to the Client: the Error and Notification bits. The Direction flag will always be set to 1 for all packets received by the Client and can be ignored. As noted above, the Direction bit (and all other flags in the Status Byte) should be set to 0 in packets sent by the Client.

The Error and Notification bits, as well as the OpCode field, must always be examined by the Client to determine the type of packet it has received.

Table 4-2 Packet Status Flags

Bit Number	Name	Meaning
7	Error	0 - No Error 1 - Error has occurred. Payload contains the error information. (See Section 5.3.3 Error Packets
6	Reserved	Client must ignore
5	Reserved	Client must ignore
4	Reserved	Client must ignore
3	Notification	0 - This is not a notification packet 1 - This is a notification packet (See Section 5.3.4 Notification Packets.)
2	Reserved	Client must ignore
1	Direction	0 - For packets sent by the Client 1 - For packets sent by the Console
0	Reserved	Client must ignore

4.1.2.1 The Error Bit

On packets from the Console, the Error bit and Notification bit will never both be set. If the Error bit is set, a protocol error has occurred (See section 5.3.3 Error Packets).

The only protocol error possible on a debugged Client would be due to packet data corruption, loss, or timeout. When the Console sends an error packet indicating corruption, the Client should simply retry the command.

4.1.2.2 The Notification Bit

The Console occasionally sends Notification packets to indicate internal Console states, which are normally not of interest to the Client. Thus if the Notification flag (bit 3) is set, then the packet contents should normally be ignored by the Client. One Notification packet that could be useful to a Client is notification that a key has been pressed on the Console or a remote. This notification packet is only sent if the Client has enabled Key Press notifications (See section 5.3.4.1 Key Press Notification Packets).

4.1.3 OpCode

The Operation code is always two bytes and indicates the operation requested of the Console. See the value listed for individual commands in section 6 Client Commands and Console Responses.

4.1.4 Payload

Content and length varies. See individual commands in section 6 Client Commands and Console Responses for details.

4.1.5 Checksum (if present)

The Checksum byte is used to help detect packet corruption during transit. The Checksum byte is computed by the Client by XOR-ing all bytes in the packet to be sent that precede the Checksum byte. Use the resulting value as the checksum value.

The binary operation table for the XOR operation, which is indicated by the caret ("^") character is shown in the table below.

Table 4-3 The XOR Operator

Bit A	Bit B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

Note XOR-ing a bit with 0 does not change the bit while XOR-ing a bit with 1 inverts the bit.

Thus a bit in the Checksum can be computed by summing the corresponding bit of each preceding byte. The Checksum bit is 1 if there is an odd number of 1s in the corresponding bit of each preceding byte and 0 if there is an even number of 1s.

For examples see the Checksum computed in Section 5.7 Sample Command/Response Interaction, as well as the actual Checksum values listed in the command packets sent by the Client in Section 6 Client Commands and Console Responses.

Note: The Checksum byte only appears on packets sent from the Client. There is no checksum byte in packets sent from the Console.

5 The Communication Protocol

The Client only sends **Command** packets, which are of two types:

1) A Control packet which requests the Console to do something
2) A Query packet which requests information from the Console

The Console sends four types of packets:

1) A Ready packet which indicates the Client can send another packet.
2) A Response packet which contains information requested by a Query packet
3) A Notification packet which indicates an event that occurred on the Console
4) An Error packet which indicates a problem with a packet sent by the Client.

The Command packets and the Response packet associated with them are listed in section 6 Client Commands and Console Responses.

5.1 Notational Conventions

Base 16 (hexadecimal) is used to represent the contents of packets. Base 10 is used everywhere else unless the number is preceded by 0x which indicates hexadecimal.

Numbers occurring in the Payload field which are two or more bytes in length are sent using the so called little-endian byte ordering, which sends the least significant byte (LSB) first. Thus the number 350 decimal (0x015E) when sent, for example as the key hold time for the key press command (see section 6.4KP - Simulate a Key Press), would be sent as 0x5E01.

The Exclusive Or (XOR) operation is indicated by the "^" character. This is a bitwise operation on two bytes, which results in a 1 if the two corresponding bits are different and a 0 if they are the same (see Table 4-3 The XOR Operator.)

5.2 Issuing Commands to the Console

To control or query the Console, the Client sends a Command packet and waits for packets from the Console. The Console does not send a Response packet for all commands but will always send a Ready packet. In rare circumstances an Error packet is sent instead of a response, but will always be followed by the Ready packet. The commands and their responses are described in section **6** Client Commands and Console Responses. (See also section 5.3.1 The Ready Packet and section 5.3.3 Error Packets**)**

Note: The Console sometimes sends Notification or Error packets, which can come at any time including immediately after the Client issues a command packet. To understand how to parse Console packets, see section 5.3 Interpreting Packets from the Console.

Regardless of whether the Console sends a response packet or not, the Console always sends a Ready Packet after every Client command packet. The Ready Packet indicates the current command has been accepted and the Client can send another command subject to protocol timing (see section 5.4 Protocol Timings). The Client must wait for the Console's response and must not issue another command until it receives a Ready Packet. See Section 5.3.1 The Ready Packet.

In general, the resulting Console state change to a control command (e.g. source change, Room On/Off) has not been performed by the time the Ready command is sent. A query command can be used if needed to determine when the state change has occurred.

5.3 Interpreting Packets from the Console

A Client should receive a packet by first examining the value of the next nonzero byte of the input stream. This value is the length of the incoming packet. If length-1 additional bytes are not

received within MAX_PACKET_TIME (see section 5.4 Protocol Timings) there has been a communication error. The console may also have lost power. See section 5.6 Autonomous Console State Changes.

The Client should interpret a packet using the following algorithm:

1) If the Error bit is set in the Status byte this is an Error packet. See section 5.3.3 Error Packets for the format of an Error packet.
2) If the Notification bit is set in the Status byte (the Error and Notification bits will never both be set) this is a notification packet. See section 5.3.4 Notification Packets.
3) The Ready Packet indicates the Console has accepted the previous command and the Client may send the next command subject to protocol timings (see section 5.4 Protocol Timings). See section 5.3.1 The Ready Packet.
4) Otherwise it is a response packet. The OpCode bytes of the packet will match the OpCode of the Command packet sent by the Client. The results of the command are in the Payload portion of the packet. The Client **must not** issue another command until it has received the Ready Packet.

5.3.1 Ready Packet

The Ready Packet is sent by the Console to indicate it has accepted the command and is ready to receive another command from the Client subject to protocol timings (see section 5.4 Protocol Timings).

The Client **must always** wait for this packet after issuing a command. It is always the same 4-byte packet:

Console sends: 04 <Status> 01 00

Table 5-1 Ready Packet Format

Byte Number	Value (Hex)	Name	Explanation
0	04	Length	The total number of bytes in this packet is 4.
1	<Status>	Status	Error bit = 0,Notification bit = 0
2-3	01 00	OpCode	The Operation Code for the Ready Packet

5.3.2 Response Packets

Commands that query the Console for information receive the requested information in a response packet. The format varies by command. For some control commands, a response will be returned to acknowledge the command has been accepted. The specific response for each command is listed in Section 6 Client Commands and Console Responses. All response packets have a header which includes the length, status, and opcode as well as a payload containing the requested information. The opcode of the response packet will match the opcode of the command. Note there is no checksum in a response packet or any packet from the Console.

5.3.3 Error Packets

An Error packet is sent by the Console in response to an invalid Command or in rare instances when a hardware error condition exists in the Console. The Error packet will be followed by a Ready packet. An Error packet has the Error bit set in the packet Status byte. The Payload contains the error code.

An Error packet has the form:
Console sends: 07 <Status> 01 <Cmd> <Error-Code> <01>

Table 5-2 Error Packet Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7
1	<Status>	Status	Error bit = 1, Notification = 0
2	01	OpCode1	The value 1
3	<Cmd>	OpCode2	The second byte of the OpCode of the failed command.
4-5	<Error-code>	Error Code	A two byte error code. LSB followed by MSB
6	01	Payload	The value 1

The two byte Error-code values appear below. Note since the code appears in the packet as Least significant byte followed by Most significant byte, the two bytes in the packet appear in reverse order from this list.

Table 5-3 Error Packet Error Codes

Two Byte Error Code (Hexadecimal)	Name	Explanation
0xfffe	PACKET_TIMEOUT	Time between characters exceeded
0xfffc	INVALID_STATUS_BYTE	Nonzero Status byte received by Console
0xfffb	NOT_AVAILABLE	Function not available in Low Power Mode
0xfffa	PACKET_CHECKSUM	Invalid Checksum received by Console
0xffef	INVALID_OPCODE	Unknown OpCode
0xffee	ARGUMENT	Incorrect Payload value
0xffed	HARDWARE	Internal Hardware error

With the exception of a HARDWARE error they indicate an invalid packet was received by the Console. The packet was invalid because the Console detected a format error, or a command not available in Low Power Mode was sent while the Console was in Low Power Mode.

In a debugged client an error (except HARDWARE) is generally due to a transmission error or a Console reboot. The Client can determine if the Console has rebooted using the Uptime Command (see section 6.12 UpTime - Query System Up Time). A transmission error should simply be retried.

When a command which is not available in Low Power Mode is sent when the Console is in Low Power Mode, the console delays NOT_AVAILABLE_TIME (see section 5.4 -- Protocol Timings) and then returns an error packet with the Error Code NOT_AVAILABLE. The Client can determine if the Console is in Low Power Mode by using the SysRdy Command (see section 6.11 SysRdy - Query System Ready).

5.3.4 Notification Packets

Notification packets can be sent by the Console at any time – even while the Client is waiting for a response or Ready packet. The most useful notification packets are the Key Press notifications. These notifications will only be sent by the Console when a Hard or Soft key press notification state has been enabled via the Echo command (See Section 6.3 Echo - Hard and Soft Key Press Notification). Note that the Hard Key Press Notification State is reset to False when the Console reboots and the Soft Key Press Notification state is reset when the Console enters Low Power Mode (see section 5.5 Low Power Mode).

A rare but important notification packet is the Assert packet which indicates the Console encountered a fatal internal condition and will reboot. The other notification packets indicate various internal events of the Console (debug messages) and should be ignored by the Client.

The format of a Notification Packet is as follows:

<Length> <Status> 01 <Notification-Type> <Payload>

Table 5-4 Notification Packet Format

Byte Number	Value (Hex)	Name	Explanation
0	<Length>	Length	The total number of bytes in the packet. Variable
1	<Status>	Status	Error bit = 0, Notification bit = 1
2	01	OpCode1	The Value 1
3	<Notification-Type>	Notification Type	Where <Notification-Type>indicates the notification type. See Table 5-5 Notification Packet Notification-Types
4—	<Payload>	Payload	The data associated with the notification type

Table 5-5 Notification Packet Notification-Types

<Notification-Type> (Hex)	Name	Explanation
00	Debug	See Section 5.3.4.3 Debug Notification Packets
02	Assert	See Section 5.3.4.2 Assert Notification Packets
04	Key Press Notification	See Section 5.3.4.1 Key Press Notification Packets

5.3.4.1 Key Press Notification Packets

Key press Notification packets can be enabled or disabled and the state of Hard Key press notifications can be queried using the Echo command (See 6.3 Echo - Hard and Soft Key Press Notification). Key Press Notification packets are only sent if the key Press notifications are enabled. Note that Hard Key Press Notification is set false when the Console reboots and Soft Key Press notification is set false when the console enters Low Power mode.

When Key Press Notification is enabled, the Client is notified with two separate packets: 1) when a key is pressed and 2) when the key is released. There is no separate notification if the key has been held but the Client can determine the hold time by using its own timer and subtracting the arrival times of the release and press notifications.

Since a key can be pressed/released on the remotes or the Console at any time, these packets may occur between the times a Client sends a command and receives a response packet and a ready packet.

They can be identified by the Status byte and the Key Press Notification OpCode: 0x0104.

Console sends: 08 <Status> 01 04 <Key-State> <Producer> <Key-Code> <Room>

Table 5-6 Key Press Notification Packet Format

Byte Number	Value (Hex)	Name	Explanation
0	08	Length	The total number of bytes in this packet is 8
1	<Status>	Status	Error bit = 0, Notification bit = 1
2-3	01 04	OpCode	The OpCode of the Key Press Notification command
4	<Key-State>	Key State	Pressed = 0, Released = 1
5	<Producer>	Key Producer	Console = 0, RF = 1, IR = 2
6	<Key-Code>	Key Code	The Key Code of the pressed key (See section 7.2 Key Press Codes)
7	<Room>	Room Code	Room where the Key was pressed. Decimal values 0-14 represent rooms A (Main) through O.

Note: The OpCode of a Key Press Notification packet is the same as the OpCode of a Ready packet, but the length of a Ready packet is always 4 versus 8 for a Key Press Notification packet and the Ready packet does not have the Notification bit set in the Status Byte.

5.3.4.2 Assert Notification Packets

This packet will only be sent from the Console if a fatal error in the Console has occurred, and it indicates the Console will reboot. The first four bytes of payload are the Assert code, the rest is a zero terminated ASCII string.

Console sends: <Length> <Status> 01 02 <Assert-Code> <Msg>

Table 5-7 Assert Notification Packet Format

Byte Number	Value (Hex)	Name	Explanation
0	<Length>	Length	The total number of bytes in this packet
1	<Status>	Status	Error bit = 0, Notification bit = 1
2-3	01 02	OpCode	
4-7	<Assert-Code>	Assert Number	Can be used by support to trace the location of the failure
8—	<Msg>	Message text	A zero terminated ASCII String

5.3.4.3 Debug Notification Packets

Console sends: <Length> <Status> 01 00 <Msg>

Table 5-8 Debug Packet Format

Byte Number	Value (Hex)	Name	Explanation
0	<Length>	Length	The total number of bytes in this packet
1	<Status>	Status	Error bit = 0, Notification bit = 1
2-3	01 00	OpCode	
5—	<Msg>	Message text	An ASCII String

5.4 Protocol Timings

This section contains symbolic names for all Console timings which are part of the protocol. They are referred to by name throughout the document.

5.4.1 MAX_PACKET_TIME

Once the Client sends the first byte of a packet, the rest of the packet must be sent within 300msec. The Console sends each packet at close to the maximum transmission speed. The worst case packet transmission time from the Console will not be more than 300msec. Since the first byte of a packet contains the number of bytes in the packet, the Client can determine the end of the packet so wait time is typically much less.

5.4.2 REBOOT_TIME

After a power failure, the console will reboot and be able to process all commands available from Low Power Mode within two seconds of power being restored.

5.4.3 NOT_AVAILABLE_TIME

When the console is in Low Power Mode and a command which is not available in Low Power Mode (see Table 1-1 System States) is issued, the console does not respond for approximately 10 seconds and then returns the error NOT_AVAILABLE. Use the SysRdy command to determine if the Console is in Normal Power Mode.

5.4.4 TURN_ON_TIME

When the Console is in Low Power Mode, all rooms will be off. If a command is issued to turn any room on (e.g. a power on or source key press) the system will transition to the Normal Power Mode after approximately 12 seconds.

5.4.5 ALL_OFF_TIME

The console transitions to Low Power Mode after all rooms have been off for approximately 12 minutes.

5.4.6 INACTIVITY_TIME

The Console turns off all rooms after no key presses have been received for several hours (generally 4 or 24). Note the Console will be in Low Power Mode prior to this since no key presses were received for the ALL_OFF_TIME.

5.4.7 INTER_COMMAND_TIME

Bose recommends that a client delays at least 100msec between receiving the Ready packet for the previous command and sending the next command. For the Key Press command however the delay should be at least 100msec longer than the key press hold time (see 6.4 KP - Simulate a Key Press) and longer for some commands like source change. In general do not send key presses faster than they would be sent from a remote.

5.4.8 LOW_POWER_POLL_TIME

Bose recommends not sending repetitive commands to the Console when it is in Low Power Mode more often than once per 90 seconds. A TAP command received in Low Power Mode causes the Console to move into a higher power state for several seconds and cause the LED to change from Red to Amber.

5.5 Low Power Mode

The console transitions from Normal Power to Low Power Mode when both Zone1 and Zone2 sources have been in the OFF source (see section 6.8 SR - Query Zone Source) for ALL_OFF_TIME. The system automatically turns off all rooms and enters Low Power Mode if no key press has been received for INACTIVITY_TIME.

The only way to bring the system out of Low Power Mode is by sending a key which turns on the Console such as BOSE_ON_ASSERT or BOSE_INPUT1. Some keys have no meaning when the Console is in Low Power mode (e.g. BOSE_LEFT_ARROW) and are not processed.

If Hard Key Press Notification has been enabled all (hard) key presses are echoed in both Low Power and Normal Power mode.

Since a Power failure or system crash is rare but can be detected by the Uptime command (see section 6.12 UpTime - Query System Up Time) or the Query Key Press Notification command (see section 6.3.1.3 Query Hard Key Press Notification State) since notification is reset to false after a reboot.

Use the SysRdy command (see section 6.11 SysRdy - Query System Ready) to determine if the Console is in Low or Normal Power Mode.

5.6 Autonomous Console State Changes

Several events external to the Client can change the state of the Console and alter the normal communication protocol. All of these states can be detected by issuing query commands. In addition the Client can detect most of these states by monitoring key presses via Key Press Notifications.

5.6.1 The Console Experiences a Power Interruption

If power is interrupted while the Client is communicating, the Client may receive a partial or invalid packet. A partial packet is detected when less than the number of characters in the Length (i.e. first) byte is received before MAX_PACKET_TIME. In this case the Client should discard any additional bytes until no bytes are received for MAX_PACKET_TIME and assume the Console is rebooting.

The system reboots into Low Power Mode after REBOOT_TIME and the Echo Notifications Enabled state is set to False. To detect that the Console has rebooted use the Uptime command (see section 6.12 UpTime - Query System Up Time)

All Commands which can be issued in Low Power Mode should respond within LOW_POWER_RESPONSE_TIME. All other commands respond within NOT_AVAILABLE_TIME. Thus if the Client does not receive a response to a command within the associated time, the Console may have lost power.

The Uptime command can detect that the Console has rebooted. If Key Press Notification has been enabled, the receipt of a key press notification indicates the Console has not lost power since Key Press Notification is reset on a reboot. Thus recording and monitoring the elapsed time from the last key press allows the client to test less frequently if the Console has lost power.

5.6.2 The Source is changed To the INPUT_ID_INSTALLIQ or INPUT_ID_PTS Source

The source is automatically changed to INPUT_ID_INSTALLIQ when a cable becomes unplugged and this source can also be selected from the menu by the user.

The INPUT_ID_PTS source is an information screen used by Bose for diagnostics.

Querying the current sources will detect if the system is in the INPUT_ID_INSTALLIQ or

INPUT_ID_PTS source.

When the Console is in either of these sources, some commands do not operate normally. For example the volume cannot be controlled and another source cannot be selected. When the system is in either of these sources, manual intervention may be necessary to exit the source.

5.6.3 The Console Transitions To/From Low Power Mode

Low Power Mode occurs only after all rooms are in the OFF source.

In Low Power Mode (see section 5.5 Low Power Mode) only the commands listed as available in Low Power Mode (see Table 1-1 System States) should be issued. If a command listed as not available in Low Power Mode is sent when the console is in Low Power mode, the console will not response for NOT_AVAILABLE_TIME and returns the error "NOT_AVAILABLE" (see Table 5-3 Error Packet Error Codes). The SysRdy command can be used to determine if the console has entered/left Low Power Mode.

When a key press that transitions the console from Low Power to Normal Power Mode is sent, no additional key press commands, or any commands which cannot be issued in Low Power Mode, should be sent until the Sysrdy command returns 1.

5.6.4 The Console Turns Off Due To Inactivity

When the system is in Normal Power mode, and no key presses have been detected by the console for INACTIVITY_TIME (see section 5.4 Protocol Timings) the console turns OFF. That is both Zone1 and Zone2 sources are set to the OFF source. If no additional keys are detected for ALL_OFF_TIME (see section 5.4 Protocol Timings) the Console transitions to Low Power Mode (see section 5.5 Low Power Mode).

5.7 Sample Command/Response Interaction

5.7.1 Sample 1 Query Main Room Volume

To request the current volume of the Console, issue the "VO" query command (See section 6.2.2 Query Main Room Volume).

The Client sends the following bytes (shown in hexadecimal) to the Console:

Client sends: 07 00 01 15 00 1e 0d

Table 5-9 Sample Volume Query

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7.
1	00	Status	Must be 0 for all packets sent from the Client
2-3	01 15	OpCode	The Operation Code for the VO command
4-5	00 1e	Payload	The Payload for the VO command
6	0d	Checksum	0x0d = 0x07 ^ 0x00 ^ 0x01 ^ 0x15 ^ 0x00 ^ 0x1e

The Checksum calculation is detailed in the following table where each byte is converted to binary and the number of 1 bits in corresponding bit positions is summed for all bytes preceding the Checksum byte. The Checksum byte is computed by the Client based on the packet data and sent as the last byte of the packet. The Console re-computes the checksum and compares it with the Checksum sent. This helps verify that the packet was not corrupted in transit.

Table 5-10 Sample Checksum Calculation

Byte Number	0	1	2	3	4	5	Number of 1's in corresponding bit in preceding bytes	6 (Checksum)
Value(Hex)	07	00	01	15	00	1e	N/A	0d
Bit 7	0	0	0	0	0	0	0	0
Bit 6	0	0	0	0	0	0	0	0
Bit 5	0	0	0	0	0	0	0	0
Bit 4	0	0	0	1	0	1	2	0
Bit 3	0	0	0	0	0	1	1	1
Bit 2	1	0	0	1	0	1	3	1
Bit 1	1	0	0	0	0	1	2	0
Bit 0	1	0	1	1	0	0	3	1

If the current value of the Console volume was 20 decimal, then the following response would be received:

Console sends: 05 <Status> 01 15 14

Table 5-11 Sample Response for Volume Query

Byte Number	Value (Hex)	Name	Explanation
0	05	Length	The total number of bytes in this packet is 5.
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 15	OpCode	The OpCode of the VO command
4	14	Payload	0x14 is 20 in decimal

The Ready Packet would be received after that.

Console sends: 04 07 01 00

(See Section 5.3.1 Ready Packet for the format and meaning of each byte.)

The Client can now issue another command after INTER_COMMAND_TIME (see section 5.4 Protocol Timings

> Note: The Client can receive Notification packets at any time. For example if key echoing had been enabled, it is possible that one or more key press notification packets could have been received before the Response packet and/or the Ready packet was received. The Client must wait for the Ready Packet before issuing another command.

(See section 6.2.2 Query Main Room Volume; section 6.3 Echo - Hard and Soft Key Press Notification, and section 5.3.4.1 Key Press Notification Packets).

5.7.2 Sample 2 Key Press Command

To switch Room B to the source plugged into input 1 of the Console, issue a KP command (See section 6.4 KP - Simulate a Key Press) from room B:

The Client sends the following bytes (shown in hexadecimal) to the Console:

Client sends: 0b 00 01 04 b3 01 2c 01 01 01 91

Table 5-12 Sample Key Press Command

Byte Number	Value (Hex)	Name	Explanation
0	0b	Length	The total number of bytes in this packet is 11
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 04	OpCode	The OpCode of the Key Press command
4	b3	Payload	The KeyCode for BOSE_INPUT1 (See section 7.2 Key Press Codes.)
5	01	Producer	Simulate a key press from the RF remote.
6-7	2c 01	Hold Time	Simulate a 300 msec hold of the key. 300 decimal is 0x012C The bytes for this field are sent in the byte order LSB, MSB so the bytes are reversed
8	01	Room Code	Room number is 1 for room B
9	01	Zone	The second zone (1) is always associated with rooms B-O
10	91	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which precede the checksum byte

The Checksum calculation is detailed in the following table where each byte is converted to binary and the number of 1 bits in corresponding bit positions is summed for all bytes preceding the Checksum byte. The Checksum byte is computed by the Client based on the packet data and sent as the last byte of the packet. The Console re-computes the checksum and compares it with the Checksum sent. This helps verify the packet was not corrupted in transit.

Table 5-13 Sample Checksum Calculation for Key Press

Byte Number	0	1	2	3	4	5	6	7	8	9	Number of 1's in corresponding bit in preceding bytes	10 (Checksum)
Value(Hex)	0b	00	01	04	b3	01	2c	01	01	01	N/A	91
Bit 7	0	0	0	0	1	0	0	0	0	0	1	1
Bit 6	0	0	0	0	0	0	0	0	0	0	0	0
Bit 5	0	0	0	0	1	0	1	0	0	0	2	0
Bit 4	0	0	0	0	1	0	0	0	0	0	1	1
Bit 3	1	0	0	0	0	0	1	0	0	0	2	0
Bit 2	0	0	0	0	0	0	1	0	0	0	1	0
Bit 1	1	0	1	1	1	0	0	0	0	0	4	0
Bit 0	1	0	0	1	1	1	0	1	1	1	7	1

Assuming no Notification packets were sent by the Console, the Ready packet would be received next since the Console does not send a response for the Key Press command.

Console sends: 04 07 01 00

See section 5.3.1 Ready Packet for the format and meaning of each byte.

The Client can now issue another command.

6 Client Commands and Console Responses

6.1 VR - Query Version

The version command requests the Software version of the Console. Note the Console must not be in Low Power Mode (see section 5.5 Low Power Mode) to get a valid response. Otherwise an Error packet will be sent. Use the SysRdy command to query the Low Power State of the system. (See 6.11 SysRdy - Query System Ready.)

Client sends: 05 00 04 01 00

The format and meaning of each byte is as follows:

Table 6-1 Query Version Command Format

Byte Number	Value (Hex)	Name	Explanation
0	05	Length	The total number of bytes in this packet is 7.
1	00	Status	Must be zero for all packets sent from the Client
2-3	04 01	OpCode	The Operation Code for the VR command
4	00	Checksum	0x00 = 0x05 ^ 0x00 ^ 0x04 ^ 0x01

Console sends: <Length> <Status> 04 01 <Console-Version> <Other-information>

Table 6-2 Query Version Response Format

Byte Number	Value (Hex)	Name	Explanation
0	<Length>	Length	The total number of bytes in this packet
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	04 01	OpCode	The Operation Code for the VR command
4—	<Version>	Version String	<Console-version> <Other-information>

Where:

<Console-version> is an ASCII String of the form:

Table 6-3 Version String Format

Byte Offset	Value	Name	Explanation
0-2	“DEF”	Product Acronym	The three ASCII characters: DEF
3	0x2e	ASCII Period (“.”)	Separator
4-5	<MM>	Major Version	Two ASCII Digit Major Version number
6	0x2e	ASCII Period (“.”)	Separator
7-8	<mm>	Minor Version	Two ASCII Digit Minor Version number
9	0x2e	ASCII Period (“.”)	Separator
10-11	<Seq>	Sequence Number	Two ASCII Digit Sequence Number

<Other-information> This information should be ignored

6.2 VO - Set/Query Main Room Volume

The volume commands allow the volume of the Main Room to be set and queried. Use the RmStatus command (See 6.13 RmStatus - Query Room Status) to query the volume of Bose® link rooms (Rooms B-O). Note that the volume of the Main Room is always true volume and not attenuated volume. For an explanation of true versus attenuated volume, see the note following section 6.13 RmStatus - Query Room Status.

6.2.1 Set Main Room Volume

Only the volume of the Main room can be set with this command. The volume of the main room and of the Bose® link rooms can also be varied by using the Key Press command to send the BOSE_VOLUME_UP or BOSE_VOLUME_DOWN keys.

Client sends: 07 00 01 15 01 <V> <C>

Table 6-4 Set Main Room Volume Command Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7
1	00	Status	All Status bits must be zero for packets sent from the Client
2-3	01 15	OpCode	The OpCode of the VO set command
4	01	SubCmd	A literal value of 0x01
5	<V>	Payload	<V> is a single byte with values from 0x00-0x64 (0-100 decimal)
6	<C>	Checksum	The checksum

There is no Console response for this command. The Ready Packet will be sent as usual which indicates the command was accepted and another command can be sent.

6.2.2 Query Main Room Volume

Use this command to query the volume of the Main Room. The volume of the Bose® link rooms can be queried using the RmStatus command.

Client sends: 07 00 01 15 00 1e 0d

Table 6-5 Query Main Room Volume Command Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7.
1	00	Status	All Status bits must be zero for packets sent from the Client
2-3	01 15	OpCode	The OpCode of the VO command
4-5	00 1e	Payload	Volume Query Payload
6	0d	Checksum	0x0d = 0x07 ^ 0x00 ^ 0x01 ^ 0x15 ^ 0x00 ^ 0x1e

Console responds: 05 <Status> 01 15 <V>

Table 6-6 Query Main Room Volume Response Format

Byte Number	Value (Hex)	Name	Explanation
0	05	Length	The total number of bytes in this packet is 5.
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 15	OpCode	The OpCode of the VO command
4	<V>	Payload	Current Main Room volume. Ranges 0x00-0x64 (0-100)

6.3 Echo - Hard and Soft Key Press Notifications

The Echo commands allow the state of key press notifications to be enabled/disabled or queried.

When Key Press notifications are enabled, a Key Press Notification packet (See 5.3.4.1 Key Press Notification Packets) is sent by the Console whenever a key is pressed or released on the Console, a remote, the VideoWave control frame, or a Key Press command is sent by the Client. Note that the Console responds to both IR (Infrared) as well as RF (radio frequency) remotes.

There are two forms of this command:

- The Hard Key press echo command enables/disables notification for the press/release of a physical button on a remote or the console.
- The Soft Key press echo command enables/disables notification for the press/release of the click pad for the selected control on the control frame of a VideoWave.

Enabling key press notifications allows the Client to monitor when keys are pressed on the Console or any remote. If key press notifications are enabled, key presses sent by the Client using the Key Press command (See 6.4 KP - Simulate a Key Press) also cause the Console to send a Key Press Notification packet.

Note if the Console reboots both Key Press Notification states are reset to disabled.

6.3.1 Hard Key Press Notifications

6.3.1.1 Enabling Hard Key Press Notifications

To enable Key Press Notification packets whenever hard (physical) keys are pressed, the Client sends the following packet:
Client sends: 07 00 01 1b 01 0d 11

Table 6-7 Enable Hard Key Press Notifications Command Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 1b	OpCode	The OpCode of the Echo command
4-5	01 0d	Payload	Payload for Key Press notification from Client
6	11	Checksum	0x11 = 0x07 ^ 0x00 ^ 0x01 ^ 0x1b ^ 0x01 ^ 0x0d

Note there is no response for this command. The Ready packet must be received from the Console before the Client can send another command.

6.3.1.2 Disabling Hard Key Press Notifications

To disable Hard Key Press Notification packets the client sends the following packet:

Client sends: 07 00 01 1b 01 00 1C

Table 6-8 Disable Hard Key Press Notifications Command Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 1b	OpCode	The OpCode of the Echo Command
4-5	01 00	Payload	Payload for Key Press notification from Client
6	1c	Checksum	0x11 = 0x07 ^ 0x00 ^ 0x01 ^ 0x1b ^ 0x01 ^ 0x00

Note there is no response for this command. The Ready packet must be received from the Console before the Client can send another command.

6.3.1.3 Query Hard Key Press Notification State

Use this command to query the hard key press notification state.

Client sends: 07 00 01 1b 00 00 1d

Table 6-9 Query Hard Key Press Notification State Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 1b	OpCode	The OpCode of the query key press notification state
4-5	00 00	Payload	Payload for Key press notification from Client
6	1d	Checksum	1d = 0x07 ^ 0x 00 ^ 0x01 ^ 0x1b ^ 0x00 ^ 0x00

Console responds: 05 <Status> 01 1b <M>

Table 6-10 Query Hard Key Press Notification State Response Format

Byte Number	Value (Hex)	Name	Explanation
0	05	Length	The total number of bytes in this packet is 5
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 1b	OpCode	The OpCode of the query key press notification state
4	<M>	Payload	Key press notification is disabled: <M> = 0 Key press notification is enabled : <M>≠ 0

6.3.2 Soft Key Press Notifications

6.3.2.1 Enabling Soft Key Press Notifications

To enable Key Press Notification packets whenever soft keys on the VideoWave are pressed the Client sends the following packet:

Client sends: 32 00 04 17 0c 00 00 00 00 00 00 00 00 00 00
00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 2c

Table 6-11 Enable Soft Key Press Notifications Command Format

Byte Number	Value (Hex)	Name	Explanation
0	32	Length	The total number of bytes in this packet is 50
1	00	Status	All Status bits must be zero for packets sent by the Client
2-4	04 17 0c	OpCode	The OpCode of the Soft button Echo command
5-16	00	Filler1	Reserved – must be zero
17	01	Requested Echo state	Enable (0x01) the Soft Key echo state
18-48	00	Filler 2	Reserved – must be zero
49	2c	Checksum	XOR of all preceding bytes in message.

This command is not available in Low Power Mode. The Soft Key Press Notifications state is set to false when the console enters Low Power Mode. Note if the Hard Key Press Notification state is enabled, a key press notification for the BOSE_TOUCH key will also be sent when the soft button is selected.

There are two possible responses to this command. An “accepted” response when the echo value is 0 or 1, and an “Invalid” response when the echo value is out of range

Table 6-12 Enable/Disable Soft Key Press Notifications Accepted Response Format

Byte Number	Value (Hex)	Name	Explanation
0	09	Length	The total number of bytes in this packet is 9.
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	04 17	OpCode	The OpCode of the Soft Key Press echo command
4-5	4f 4b	Response Msg	“OK”. Indicates value is in range (0 or 1).
6-8	0d 0a 00	Terminator	Message terminator <CR><LF> <NULL>

Table 6-13 Enable/Disable Soft Key Press Notifications Invalid Response Format

Byte Number	Value (Hex)	Name	Explanation
0	20	Length	The total number of bytes in this packet is 32
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	04 17	OpCode	The OpCode of the Soft Key Press echo command
4-28	49 6e 76 61 6c 69 64 20 76 61 6c 75 65 20 66 6f 72 20 4f 54 20 45 43 48 4f	Response Msg	“Invalid value for OT ECHO” indicating the Echo state value is out of range
29-31	0d 0a 00	Terminator	Message terminator <CR><LF> <NULL>

6.3.2.2 Disabling Soft Key Press Notifications

To disable Key Press Notification packets when soft keys on the VideoWave are pressed the Client sends the following packet:

Client sends: 32 00 04 17 0c 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 2d

Table 6-14 Disable Soft Key Press Notifications Command Format

Byte Number	Value (Hex)	Name	Explanation
0	32	Length	The total number of bytes in this packet is 50
1	00	Status	All Status bits must be zero for packets sent by the Client
2-4	04 17 0c	OpCode	The OpCode of the Soft button Echo command
5-16	00	Filler1	Reserved – must be zero
17	00	Requested Echo state	Disable (0x00) the Soft Key echo state
18-48	00	Filler 2	Reserved – must be zero
49	2d	Checksum	XOR of all preceding bytes in message

The Disable Soft Key Press Response Formats are identical to the Enable Soft Key Press Response format and are listed in Table 6-12 and Table 6-13.

The Soft Key Press notification state cannot be queried.

6.4 KP - Simulate a Key Press

Use this command to simulate pressing a key on the Console or a remote. This is the primary command for controlling the system.

Client sends: 0b 00 01 04 <K> <P> <H> <R> <Z> <C>

Table 6-15 Key Press Command Format

Byte Number	Value (Hex)	Name	Explanation
0	0b	Length	The total number of bytes in this packet is 11
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 04	OpCode	The OpCode of the Key Press command
4	<K>	Payload	The Key Code representing the key. See section 7.2 Key Press Codes
5	<P>	Producer	A value representing the source of the key press. 0 – Console 1 - RF remote 2 – IR remote
6-7	<H>	Hold Time	The duration in msec for which the key would have been pressed. Sent in the byte order LSB, MSB
8	<R>	Room Code	Value 0-14 (0x00-0x0e) which represents room A (Main) through O
9	<Z>	Zone	0 – Zone 1 (Room A (Main)) 1 – Zone 2 (Rooms B-O)
10	<C>	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which preceded the checksum byte

Note there is no command response from the Console for this command. The Ready packet will be sent by the Console after the command has been processed as usual. As always, the Ready Packet must be received by the Console before sending another command.

Note: If Key Press notification is enabled, the Key Press notification for that key will be sent by the Console and may arrive (as a notification packet) before the Ready packet (See section 5.3.4.1 Key Press Notification Packets.

Note that the OpCode in a Key Press Notification packet (see section 5.3.4.1 Key Press Notification Packets) identifies the key which was pressed. However only opcodes associated with Hard Key presses can be sent via the Key Press Command and only a subset of the Hard Key presses codes (see Table 7-2 Hard Key Press Codes) are appropriate when the console is in Low Power Mode or when a particular device type is the active source.

6.5 TS - Set Tuner Station

Before issuing this command, the band (AM/FM) associated with the station to be set must be the most recently set tuner band. To ensure this the Client can use the KP command to send either the AM_SOURCE or FM_SOURCE (see section 6.4 KP - Simulate a Key Press). The current source can be verified with the SR command. See section 6.8 SR - Query Zone Source.

Note: This command sets the tuner *station* not the tuner *frequency*. Use the information in section 6.6.1 Converting between frequency and station number, to convert between tuner station and tuner frequency.

Client sends: 07 00 01 4a <Station> <C>

Table 6-16 Set Tuner Station Command Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 4a	OpCode	The OpCode of the Set Tuner Station command
4-5	<Station>	Payload	The station number. Sent in the byte order LSB, MSB
6	<C>	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) that precede the checksum byte

There is no response from the Console for this command. As for all commands, the Ready packet must be received from the Console before the Client can send another command.

6.6 QS - Query Tuner Station

Before issuing this command, the band (AM/FM) associated with the station to be set must be the most recently set tuner band. To ensure this the Client can use the KP command (see section 6.4 KP - Simulate a Key Press) to send either the AM_SOURCE or FM_SOURCE (see section 7.2 Key Press Codes). The current source can be verified with the SR command (see section 6.8 SR - Query Zone Source).

Note: This command queries the tuner *station* not the tuner *frequency*. Use the information in Section6.6.1 Converting between frequency and station number, to convert between tuner station and tuner frequency.

Note: The Preset associated with the currently tuned station can be queried using the Query Current Tuner Preset command (see section 6.7 Query Current Tuner Preset).

Client sends: 05 00 01 49 4d

Table 6-17 Query Tuner Station Command Format

Byte Number	Value (Hex)	Name	Explanation
0	05	Length	The total number of bytes in this packet is 5
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 49	OpCode	The OpCode of the Query Tuner Station command
4	4d	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) that precede the checksum byte

Console Responds: 06 <Status> 01 49 <Station>

Table 6-18 Query Tuner Station Response Format

Byte Number	Value (Hex)	Name	Explanation
0	06	Length	The total number of bytes in this packet is 6
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 49	OpCode	The OpCode of the Query Tuner Station command
4-5	<Station>	Station Number	The value of the Station number. The Least significant byte is first, followed by the most significant byte. See the calculation in Section 6.6.1 Converting between frequency and station to convert a Station number to a frequency.

6.6.1 Converting between frequency and station number

The commands to control and query the tuner use station number rather than station frequency as parameters. There are also three tuner regions: USA, Europe, and Japan.

To convert between the two values use these values and formulas:

Table 6-19 Converting between frequency and station number

Tuner Region	Band	Station Start	Station End	Station Step	Frequency Start	Frequency End	Frequency Step
USA	FM	8	816	8	87700	107900	200
USA	AM	1	119	1	530	1710	10
Europe	FM	2	822	2	87500	108000	50
Europe	AM	1	122	1	522	1611	9
Japan	FM	4	564	4	76000	90000	100
Japan	AM	1	123	1	531	1629	9

In the formulas below the Frequency is expressed in kilohertz. For example for the USA FM value displayed on the radio as 87.7 the corresponding actual frequency is 87700 kHz.

Station = Integer((***Frequency*** - <Frequency Start>)/<Frequency Step>) * <Station Step> + <Station Start>

Frequency = Integer((***Station*** - <Station Start>) / <Station Step) * <Frequency Step> + <Frequency Start>

6.6.2 Sample Conversions

To convert USA FM frequency “103.3”, which is 103300 kHz, to a station number:

Station Number = Integer((103300 – 87700) / 200) * 8 + 8

= Integer(15600/200) * 8 + 8

= Integer(78) * 8 + 8

= 78 * 8 + 8

= 624 + 8

= 632

To convert the station number 632 for USA FM to a frequency:

Frequency = Integer((632 – 8) / 8) * 200 + 87700

= Integer(624/8) * 200 + 87700

= Integer(78) * 200 + 87700

= 15600 + 87700

= 103300

6.7 Query Current Tuner Preset

Use this command to obtain the preset number (1-25) associated with the currently tuned station. If there is no preset associated with the currently tuned station the value returned is zero.

Note: This command is not available in Low Power Mode.

Client sends: 12 00 04 03 05 54 55 4e 45 52 50 52 45 53 45 54 00 4d

Table 6-20 Query Current Tuner Preset Command Format

Byte Number	Value (Hex)	Name	Explanation
0	12	Length	The total number of bytes in this packet is 18
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	04 03	OpCode	The OpCode of the Query Current Tuner Preset command
4	05	Format	Indicates Ascii Payload
5-16	54 55 4e 45 52 50 52 45 53 45 54 00	Name	The zero terminated ascii string “TUNERPRESET”
17	4d	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) that precede the checksum byte

Console Responds: <Length> <Status> 04 03 <Preset-Value> 0a 00

Table 6-21 Query Current Tuner Preset Response Format

Byte Number	Value (Hex)	Name	Explanation
0	<Length>	Length	The total number of bytes in this packet including this byte
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	04 03	OpCode	The OpCode of the Query Current Tuner Preset command
4-	<Payload>	Preset Value	The value of the current station's preset as a string of one or two ASCII digits followed by Line Feed (0x0a) and Zero.er to a frequency.

6.8 SR - Query Zone Source

Use this command to query the current source of audio in Zone1 and Zone2. The source associated with Room A (the room the Console is in) is the Zone 1 source. For products that support multiple rooms the source playing in all other rooms is the Zone 2 source. Note that changing the source for any room other than A (Main) changes the source for all rooms B-O, since it changes the source for Zone 2.

To set the source, use the KP command with the Key Code corresponding to the desired source. Note that when a zone is off, the source returned will be the "OFF" source. A Zone's source will change to the OFF source when all rooms in that zone are off. However the zone source will revert to the previous zone source when the zone becomes active.

Client sends: 08 00 01 0a 00 18 18 03

Table 6-22 Query Zone Source Command Format

Byte Number	Value (Hex)	Name	Explanation
0	08	Length	The total number of bytes in this packet is 8
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 0a	OpCode	The OpCode of the Query Zone Command
4-6	00 18 18	Payload	
7	03	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which preceded the checksum byte

Console responds: 08 <Status> 01 0a <Z1> <X> <Z2> <X>

Table 6-23 Query Zone Source Response Format

Byte Number	Value (Hex)	Name	Explanation
0	08	Length	The total number of bytes in this packet is 8
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 0a	OpCode	The OpCode of the Query Zone Source command
4	<Z1>	Zone1 Source	The Source enumeration value for the Source playing in Zone 1 (Room A, the room the Console is in). See section 7.1 Source Enumerators
5	<Z2>	Zone 2 Source	The Source enumeration value for the Source playing in Zone 2 (rooms B-O). See section 7.1 Source Enumerators
6	<X>	Reserved	Reserved
7	<X>	Reserved	Reserved

6.9 RDS PS - Query RDS Program Service

This is an eight-byte value that represents the call letters or station identity name.

Client sends: 06 00 01 4f 00 48

Table 6-24 Query RDS Program Service Command Format

Byte Number	Value (Hex)	Name	Explanation
0	06	Length	The total number of bytes in this packet is 6
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 4f	OpCode	The OpCode of the Query RDS Program Service Command
4	00	Payload	Payload
5	48	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which preceded the checksum byte

Console responds: 0c <Status> 01 4f <T0> <T1> <T2> <T3> <T4> <T5> <T6> <T7>

Table 6-25 Query RDS Program Service Command Response Format

Byte Number	Value (Hex)	Name	Explanation
0	0c	Length	The total number of bytes in this packet is 12
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 4f	OpCode	The OpCode of the Query RDS Program Service command
4-11	<T0 – T7>	Program Service	The eight bytes of the Program Service

6.10 RDS RT - Query RDS Radio Text

This function allows a radio station to transmit 64-character free-form textual information that can be either static (e.g. station slogans) or in sync with the programming (such as the title and artist of the currently-playing song.)

Client sends: 06 00 01 4f 03 4b

Table 6-26 Query RDS Radio Text Command Format

Byte Number	Value (Hex)	Name	Explanation
0	06	Length	The total number of bytes in this packet is 6
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 4f	OpCode	The OpCode of the Query RDS Radio Text Command
4	03	Payload	Payload
5	4b	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which preceded the checksum byte

Console responds: 44 <Status> 01 4f <T0> … <T63>

Table 6-27 Query RDS Radio Text Response Format

Byte Number	Value (Hex)	Name	Explanation
0	44	Length	The total number of bytes in this packet is 68
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 4f	OpCode	The OpCode of the Query RDS Radio Text Command
4-67	<T0 – T63>	Program Service	The 64 bytes of Radio Text

6.11 SysRdy - Query System Ready

When the Console is first powered on and begins booting, or when all rooms including the Main room have been turned off for a while, the Console goes into a Low Power Mode. In this state, the system only responds to a limited number of commands. (See section 5.5 Low Power Mode)

If no response is received within LOW_POWER_RESPONSE_TIME, either the system is unplugged or the system is in the process of rebooting. When the system responds, the value returned indicates if the system is ready to accept commands. If it is not ready, a key press that will wake the system should be sent and the SysRdy command reissued until the Console responds it is ready.

To test if the Console is ready to accept commands which can not be issued in Low Power Mode, issue the SysRdy command.

Client sends: 07 00 01 1d 00 00 1b

Table 6-28 SysRdy Command Format

Byte Number	Value (Hex)	Name	Explanation
0	07	Length	The total number of bytes in this packet is 7
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 1d	OpCode	The OpCode of the System Ready Command
4-5	00 00	Payload	Payload
6	1b	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which preceded the checksum byte

Console responds: 05 02 01 1d <M>

Table 6-29 SysRdy Response Format

Byte Number	Value (Hex)	Name	Explanation
0	05	Length	The total number of bytes in this packet is 5
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 1d	OpCode	The OpCode of the SysRdy Command
4	<M>	Payload	<M> = 0 System is not ready <M> = 1 system is ready

6.12 UpTime - Query System Up Time

This command returns the four byte unsigned count of the number of milliseconds since the Console last booted. This command can be used to detect if the console has lost power, since

the time returned would be less than the last value returned. Note that this value wraps to zero approximately every 50 days of continuous power.

Bose recommends not issuing the UpTime command more often than LOW_POWER_POLL_TIME (see section 5.4 Protocol Timings)

Client sends: 06 00 01 24 00 23

Table 6-30 UpTime Command Format

Byte Number	Value (Hex)	Name	Explanation
0	06	Length	The total number of bytes in this packet is 6
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 24	OpCode	The OpCode of Query System Up Time
4	00	Payload	Payload
5	23	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which preceded the checksum byte

Console responds: 08 <Status> 01 24 <Time>

Table 6-31 UpTime Response Format

Byte Number	Value (Hex)	Name	Explanation
0	08	Length	The total number of bytes in this packet is 8
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 24	OpCode	The OpCode of Query System Up Time
4-7	<Time>	Payload	The number of milliseconds since the Console last rebooted. A four byte unsigned integer sent with the LSB first and MSB last.

Note this value wraps to zero after approximately 50 days of uptime.

6.13 RmStatus - Query Room Status

This command can be used to determine several attributes of a room. The information is valid for all rooms except Room A where only the Mute value is correct. The volume of the Main Room can be queried using the VO command (see 6.2.2 Query Main Room Volume). The Source of Room A can be queried using the SR command (see 6.8 SR - Query Zone Source)

The information includes:

a) If the room is connected or not
b) The source of audio playing in the room
c) The volume of the room
d) Whether the room is muted or not

Client sends: 08 00 01 11 00 02 <R> <C>

Table 6-32 Query Room Status Command Format

Byte Number	Value (Hex)	Name	Explanation
0	08	Length	The total number of bytes in this packet is 8
1	00	Status	All Status bits must be zero for packets sent by the Client
2-3	01 11	OpCode	The OpCode of the Query Room Status Command
4-5	00 02	Payload	Payload
6	<R>	Room	A value in the range 0 - 14
7	<C>	Checksum	The exclusive or (XOR) of all bytes in this packet (including length) which preceded the checksum byte

Console responds: 07 02 01 11 <Speaker-Type> <M-Loud> <RState>

Table 6-33 Query Room Status Response Format

Byte Number	**Value (Hex)**	**Name**	**Explanation**
0	07	Length	The total number of bytes in this packet is 7
1	<Status>	Status	Error bit = 0, Notification bit = 0
2-3	01 11	OpCode	The OpCode of the RoomStatus Command
4	<Speaker-Type>	Speaker Type	If the value of this byte is 0xFF, the room is not connected.
5	<M-Loud>	Mute and loudness	A bit field. The least significant bit is the mute value for the room and the most significant seven bits are the room loudness indicator. See the note below on how to interpret this value
6	<RState>	SpeakerState	A bit field. The two least significant bits have a value whose meaning is indicated in the SpeakerState Table

Note on interpreting the loudness of a room. The volume of a room can be indicated with either the true volume (0-100) or the attenuated value (100-0). The method used depends on the product in that room and will always be interpreted the same way for a given product. Only the Bose LSA2 and LSA3 amplifiers use attenuated volume.

To determine if the value is *true volume* or *attenuated volume*, the Client can query the current volume and then send a volume up or volume down key press to change the room volume and query the room information again. If a volume up key press to that room causes the value to increase then the loudness portion of the <M-Loud> field is *true volume*. If a volume up key press causes the value to decrease then the loudness portion of the <M-Loud> field is *attenuated volume*.

Table 6-34 Room Status Speaker State

Value	**Meaning**
0	Zone 1
1	Zone 2
2	Local Source
3	Speaker is OFF

The source associated with a zone can be determined using the SR - Query Zone Source command. See section 6.8 SR - Query Zone Source). The Main Room is always associated with Zone1.

7 Appendix

7.1 Source Enumerators

These are the values returned by the SR command (see Section 6.8 SR -Query Zone Source.)

Table 7-1 Source Enumerators

Source Name	Value)	Explanation
OFF	0x00	Zone is OFF
Reserved	0x01	
Reserved	0x02	
Reserved	0x03	
INPUT_ID_FM	0x04	FM band of tuner
INPUT_ID_AM	0x05	AM band of tuner
Reserved	0x06	
Reserved	0x07	
Reserved	0x08	
INPUT_ID_IN1_HDMI	0x09	Rear HDMI input connector nearest TV
INPUT_ID_IN2_HDMI	0x0A	Rear HDMI input connector
INPUT_ID_IN3_HDMI	0x0B	Rear HDMI input connector
INPUT_ID_IN4	0x0C	Rear Analog, Coax, or Optical input connector
INPUT_ID_IN5	0x0D	Rear Analog, Coax, or Optical input connector
INPUT_ID_TV	0x0E	TV optical audio input connector
INPUT_ID_USB0	0x0F	Front USB connector*
INPUT_ID_IPOD	0x10	iPod connector
INPUT_ID_INSTALLIQ	0x11	Unify menus
INPUT_ID_BOSELINK_1	0x12	Bose® link stream 1 input*
INPUT_ID_BOSELINK_2	0x13	Bose® link stream 2 input*
Reserved	0x14	
INPUT_ID_HDMI_FRONT	0x15	Front HDMI input connector*
INPUT_ID_AV_FRONT	0x16	Front Analog input connector*
INPUT_ID_PTS	0x17	System Information Screen

7.1.1 Source Enumerator Notes

* Sources INPUT_ID_USB0, INPUT_ID_BOSELINK1, INPUT_ID_BOSELINK2, INPUT_ID_HDMI_FRONT, and INPUT_ID_HDMI_AV_FRONT are only available to Room A.

7.2 Key Press Codes

7.2.1 Hard Key Press Codes

These keys can be sent using the KP command. They will also be the values listed in Key Press notification packets. These are also the codes received from an Infrared (IR) remote.

(See Sections 6.4 KP - Simulate a Key Press and 5.3.4.1 Key Press Notification Packets)

Table 7-2 Hard Key Press Codes

Key Name	Value	Explanation	Supported Rooms
BOSE_MUTE	0x01	Toggles mute/unmute	All
BOSE_VOLUME_DOWN	0x02	Volume down	All
BOSE_VOLUME_UP	0x03	Volume up	All
BOSE_TUNER_SOURCE	0x06	Selects current Tuner Source /Toggles Tuner band (second press)	B-O
BOSE_IPOD_SOURCE	0x07	Source Selection	All
BOSE_USB_SOURCE	0x08	Source Selection	A (Main)
BOSE_FRONT_HDMI_SOURCE	0x09	Source Selection	A (Main)
BOSE_FRONT_AV_SOURCE	0x0A	Source Selection	A (Main)
BOSE_PARTY	0x0D	Select Room A source	B-O
BOSE_TV_SOURCE	0x0E	Source Selection	All
BOSE_IPOD_OTHER	0x0F	Select IPOD or untyped sources	B-O
BOSE_INPUT_2	0x12	Source Selection	All
BOSE_INPUT_3	0x13	Source Selection	All
BOSE_LAST_CHANNEL	0x14	Previous channel	All
BOSE_INPUT_4	0x15	Source Selection	All
BOSE_CHANNEL_PRESET_DOWN	0x18	Channel/Preset down	All
BOSE_CHANNEL_PRESET_UP	0x19	Channel/Preset up	All
BOSE_STOP	0x1A	Depends on Source	All
BOSE_DOWN_ARROW	0x20	Menu navigation	A (Main)
BOSE_FM_SOURCE	0x25	Source Selection†	All
BOSE_AM_SOURCE	0x28	Source Selection†	All
BOSE_EXIT	0x30	Menu Navigation	A (Main)
BOSE_NUMBER_0	0x40	Numeric input	All
BOSE_NUMBER_1	0x41	Numeric input	All
BOSE_NUMBER_2	0x42	Numeric input	All
BOSE_NUMBER_3	0x43	Numeric input	All
BOSE_NUMBER_4	0x44	Numeric input	All
BOSE_NUMBER_5	0x45	Numeric input	All
BOSE_NUMBER_6	0x46	Numeric input	All
BOSE_NUMBER_7	0x47	Numeric input	All
BOSE_NUMBER_8	0x48	Numeric input	All
BOSE_NUMBER_9	0x49	Numeric input	All

Key Name	Value	Explanation	Supported Rooms
BOSE_ON_OFF	0x4C	Toggles Room On/Off	All
BOSE_MUTE_ALL	0x4D	Toggles Mute Mode in all rooms	All
BOSE_INPUT_5	0x52	Source Selection Input 5 Component Video	All
BOSE_DISC	0x53`	Select “Disc” Sources	B-O
BOSE_PLAY	0x55	Depends on Source	All
BOSE_PAUSE	0x56	Depends on Source	All
BOSE_FAST_REVERSE	0x57	Depends on Source. Tuner:next lower frequency	All
BOSE_FAST_FORWARD	0x58	Depends on Source. Tuner: next higher frequency	All
BOSE_QUICK_REPLAY	0x59	Depends on Source. Tuner: seek down.Images:previous	All
BOSE_QUICK_SKIP	0x5A	Depends on Source. Tuner seek up. Images:next	All
BOSE_DASH	0x5C	Depends on Source and Country minus or dash	A (Main)
BOSE_SHUFFLE	0x5D	Depends on Source	All
BOSE_RIGHT_ARROW	0x60	Menu navigation	A (Main)
BOSE_INFO	0x64	The INFO key on Room A remote	A (Main)
BOSE_RECORD	0x65	Supported by Control Integration	A (Main)
BOSE_PAGE_DOWN	0x68	Page down on iPod or USB List	All
BOSE_PAGE_UP	0x69	Page up on iPod or USB List	All
BOSE_SYSTEM_SETUP	0x70	May be useful for key echo notification	A (Main)
BOSE_MORE_BUTTONS	0x80	Menu Navigation	A (Main)
BOSE_ON_ASSERT	0x8C	Turns room ON. Not a toggle	A (Main)
BOSE_DVD_MENU	0x90	Depends on Source. iPod, USB backs up one menu.	All
BOSE_LEFT_ARROW	0xA0	Menu Navigation	A (Main)
BOSE_INPUT_1	0xB3	Source Selection	All
BOSE_UP_ARROW	0xC0	Menu Navigation	A (Main)
BOSE_OFF_ASSERT	0xCC	Turns room OFF. Not a toggle	A (Main)
BOSE_SOURCE_CYCLE	0xD3	Displays Source Menu	A (Main)
BOSE_ASPECT_RATIO	0xDC	Aspect Ratio	A (Main)
BOSE_MENU_ENTER	0xE0	Menu Navigation	A (Main)
BOSE_LOCAL_SOURCE	0xF4	Switches to Local Source input where possible	B-O
BOSE_NETWORK1	0xF5	Bose® link stream 1*	A (Main)
BOSE_NETWORK2	0xF6	Bose® link stream 2*	A (Main)
BOSE_ALL_OFF	0xF7	Powers off Rooms B-O‡	B-O
BOSE_TOUCH	0xF9	Click Pad for selecting Soft Keys	A (Main)

7.2.2 Soft Key Press Codes

The VideoWave Control Frame allows buttons on the video display to be selected with the click pad. When the click pad is pressed a Key Press Notification (section 5.3.4.1) will be sent if Soft Key Press echo has been enabled (See section 6.3.2 Soft Key Press Notifications).

Note that if Hard Key Press Notification (section 6.3.1 Hard Key Press Notifications) has also been enabled, a notification for the BOSE_TOUCH key is sent each time the click pad is pressed and released. Only Hard key press codes can be sent using the Key Press command. The following key press codes are associated with soft buttons and there is some overlap with the Hard Key Press codes. The set of soft buttons available to a user depends on the source.

Table 7-3 Soft Key Press Codes

Key Name	Value
BOSE_STOP	0x1A
BOSE_EXIT	0x30
BOSE_NUMBER_0	0x40
BOSE_NUMBER_1	0x41
BOSE_NUMBER_2	0x42
BOSE_NUMBER_3	0x43
BOSE_NUMBER_4	0x44
BOSE_NUMBER_5	0x45
BOSE_NUMBER_6	0x46
BOSE_NUMBER_7	0x47
BOSE_NUMBER_8	0x48
BOSE_NUMBER_9	0x49
BOSE_PLAY_PAUSE	0x54
BOSE_PLAY	0x55
BOSE_PAUSE	0x56
BOSE_FAST_REVERSE	0x57
BOSE_FAST_FORWARD	0x58
BOSE_QUICK_REPLAY	0x59
BOSE_QUICK_SKIP	0x5A
BOSE_DASH	0x5C
BOSE_SHUFFLE	0x5D
BOSE_INFO	0x64
BOSE_RECORD	0x65
BOSE_PAGE_DOWN	0x68
BOSE_PAGE_UP	0x69

Key Name	Value
BOSE_SYSTEM_SETUP	0x70
BOSE_CABLE_SAT_POWER	0x7C
BOSE_DVD_MENU	0x90
BOSE_REPEAT	0x9D
BOSE_PLAYLIST	0x9F
BOSE_INTERACTIVE	0xB2
BOSE_PIP	0xB4
BOSE_SWAP	0xB5
BOSE_LIVE_TV	0xB6
BOSE_TELETEXT	0xB7
BOSE_A	0xB8
BOSE_B	0xB9
BOSE_C	0xBA
BOSE_D	0xBB
BOSE_PREV_DISC	0xBC
BOSE_NEXT_DISC	0xBD
BOSE_PREV_DAY	0xBE
BOSE_NEXT_DAY	0xBF
BOSE_HOME	0xC1
BOSE_POPUP	0xC2
BOSE_FAVORITES	0xC3
BOSE_RED	0xC5
BOSE_GREEN	0xC6
BOSE_YELLOW	0xC7
BOSE_BLUE	0xC8
BOSE_VOD	0xC9
BOSE_EPG	0xD0
BOSE_ASPECT_RATIO	0xDC
BOSE_OPTIONS	0xDD
BOSE_MODE_KEY	0xE9
BOSE_INVALID_KEY	0xFF

7.2.3 Key Press Notes

There is only one digital audio decoder in the console so for multi-room configurations Bose recommends connecting the analog inputs for each source. Otherwise when the room A (Main Room) source is digital (e.g. HDMI) rooms B-O will hear silence if they are listening to a different digital source.

* Unlike other source keys BOSE_NETWORK1 and BOSE_NETWORK2 do not turn the Console On if it was Off.

† Since there is only one radio tuner, a band change (e.g. AM to FM) causes all rooms listening to the tuner to hear the new band

‡ In a multi-room system, the BOSE_ALL_OFF key will turn off all rooms but the room number

used should be any existing room except A.

8 Release Notes

8.1 Known Problems

TrackWeb 5765. Source Keys do not change the source when the system is in low power mode. These key presses will bring the system out of low power mode but the source is set to the last used source for that stream. Status: Resolved in 01.05.00.

TrackWeb 6151. After a reboot, (e.g. a power failure) if a Bose® link room (B-O) is turned on and then a source key is sent to turn on the main room (A), the main room will come up in the selected source but will be muted. Status: Resolved in 01.06.00.

9 Document Revision History

Table 9-1 Revision History

DOCUMENT REVISION	DESCRIPTION	ENG	DATE
0.0	Initial Draft	PF	3/12/2010
0.1	Incorporate Internal Feedback	PF	3/17/2010
0.2	Converted commands to tables	PF	3/26/2010
0.3	Added Bass Box command	PF	4/5/2010
0.4	Removed Bass Box Command and incorporated review comments	PF	4/13/2010
0.5	D.J. Wood's feedback	PF	4/27/2010
0.6	Legal, TW4566 and TW4550 See Release Notes	PF	5/18/2010
0.7	Minor Edits for Alpha	PF	5/28/2010
0.8	Updated Version String format, Key Press table. Added Key Press Notes	PF	6/17/2010
0.9	Equated Room A to Main	PF	6/22/2010
0.10	Adding Table of States, SysRdy and RmState	PF	8/03/2010
0.11	Expanded Protocol Timings and Low Power Mode. Added Autonomous Console State Changes and Query Current Tuner Preset	PF	10/25/2010
1.0	Added additional products, misc edits	PF	10/29/2010
1.01	Incorporate AM's changes to use console names rather than product names to specify applicable hardware. Removed "Draft" in watermark	AM,PF	11/9/2010
1.02	Updates for Console Software version 01.05.00. Changes to tables 6-11, 6-16, 6-19, 6-28.	PF	04/05/2011
1.03	Add VideoWave soft button echo	PF	6/7/2011
1.04	Update TW6151 fixed	PF	6/28/2011