

SIGNATURE 2.0

SIGNATURE 2.0 BUS PROTOCOL
INSTRUCTIONS & OPERATION MANUAL
MASTER DOCUMENT
VERSION 1.0

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

TABLE OF CONTENTS

| | |
|---------------------------------|----|
| Software Revision History _____ | 3 |
| How Does It Work? _____ | 4 |
| Event Driven Output _____ | 4 |
| Protocol Format _____ | 5 |
| Sample Message _____ | 7 |
| Direct Commands _____ | 8 |
| Response Commands _____ | 31 |
| Unsolicited Commands _____ | 43 |

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

SOFTWARE REVISION HISTORY

The following is a list of the software revisions from the first version of software to include RS-232 control to the most current version. Verify the version of software in your unit by viewing the VFD (Vacuum Fluorescent Display) on the front panel immediately after you apply power to the Signature 2.0 (main power, not standby).

Version 2.5: First release.

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

How Does It Work?

The Signature Communication Protocol is designed specifically for controlling audio and video equipment. It was developed in response to a need for a low-cost, simple, fault-tolerant communications protocol. It is based on the EIA RS-232 electrical specification. Data is communicated at **9600 baud, 8 data bits, 1 stop bit and no parity**.

Before beginning data transmission, check the bus for inactivity. If the bus is free, transmission may commence. If the bus is being used, wait for a period of inactivity. Check each byte of a transmission against the intended message to ensure that data corruption does not occur due to multiple devices attempting to use the bus at the same time. For each message received, the receiver will generate an acknowledgment. The device which sent the message should receive this acknowledgment before beginning its next data transmission. Under rare circumstances, the receiver may be performing a function which cannot be interrupted. The message should be retransmitted until an acknowledgment is received. Due to this fact, delays are required between incoming messages for the transmission of the required acknowledgment.

Event Driven Output:

The Signature 2.0 will generate commands when the processor is accessed from an IR or Front panel button command. This command can be used to update external displays in real time or to generate a polling request for status.

The command generated will be identical to the Signature bus command for the same function, but it will have a different Target Device Type and Command Modifier. Example: the IR "Vid 1" button on the remote will generate a "Vid 1" button command on the Signature bus. See page ?? of this document for a list of the Remote Button codes.

Volume level updates from the Front Panel rotary encoder are handled slightly differently. When a volume adjustment is made, the 2.0 will wait for a period of encoder inactivity and then transmit an unsolicited "Power/Volume Status" response. See page ??.

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

PROTOCOL FORMAT

Note: the symbols “*b*”, “*d*”, and “*h*” used in this document denote binary, decimal, and hexadecimal values, respectively. All transmissions must be in hexadecimal.

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Preamble is a byte used to alert the system that a message is about to be transmitted. This allows the system to ignore random noise, and reliably receive valid messages. The Preamble is always equal to *7Eh*.

NOTE: No other byte in any part of the message may equal *7Eh*.

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Target Device Type is a byte that distinguishes what type of unit the message is intended for (i.e. a preamp versus a CD player).

HEX Device Type

- 01 Audioaccess KPS keypad
- 04 Audioaccess Computer
- 05 Audioaccess Paging/SMM Module
- 06 Audioaccess PX-603
- 07 Audioaccess Internal Multi
- 20 Signature 2.0 Processor/Tuner
- 30 Signature 3.0 DVD player
- 50 Citation 5.0 AV Controller
- 70 Citation 7.0 AV Controller
- 99 Broadcast Message (everyone listens)

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Target Unit ID is a byte that distinguishes which unit of a given type the message is intended for (i.e. CD player #1 versus CD player #2). The Unit ID is set in the menu system of the unit you are trying to control, default = *01h*. If you are using more than one of any given unit, you will need to modify this byte to reflect the ID number of the unit you are trying to control.

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Source Device Type is a byte that distinguishes what type of unit the message is coming from (i.e. a preamp versus a CD player). *See above list.*

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

| | | | | | | | | | |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Source Unit ID is a byte that distinguishes which unit of a given type the message is coming from (i.e. CD player #1 versus CD player #2). The Unit ID is set in the menu system of the unit that the message is coming from, default = 01h. If you are using more than one of any given unit, you will need to modify this byte.

| | | | | | | | | | |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Data Length is a word representing the total number of bytes (including Command Modifier and Command bytes, but not the Data Length byte itself or the Checksum byte) in the data portion of the message. Minimum length is 02h.

| | | | | | | | | | |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Command Modifier is a byte that distinguishes what type of command is being sent.

HEX Command Type

- 01 Action request
- 02 Action response

| | | | | | | | | | |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Command byte is the action that is desired.

| | | | | | | | | | |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Data bytes consist of information that is required to complete the Command. The Data bytes are detailed throughout the rest of this document.

| | | | | | | | | | |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|

The Checksum is a byte whose value confirms/voids the validity of a received message. It is computed by adding up all of the bytes Preamble through the last Data byte, and then taking the 1s complement of the lower eight bits of that sum (i.e. add up all the bytes, truncate to eight bits, and invert each of the bits). If the result equals 7Eh, add 01h. *See the page 7 for an example.*

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

EXAMPLE OF MESSAGE CREATION

Create a Power On command from a Signature 3.0 to a Signature 2.0
in a single 2.0 / single 3.0 system.

Preamble = $7Eh$

Target Device Type = $20h$

Target Unit ID = $01h$

Source Device Type = $30h$

Source Unit ID = $01h$

Length = $03h$

Modifier = $01h$

Command = $02h$

Data = $01h$

```
0111 1110b (7Eh)
0010 0000b (20h)
0000 0001b (01h)
0011 0000b (30h)
0000 0001b (01h)
0000 0011b (03h)
0000 0001b (01h)
0000 0010b (02h)
+ 0000 0001b (01h)
-----
0 1101 0111b (sum = 0D7h)

1101 0111b (truncating = D7h)

0010 1000b (inverting bits = 28h)

0010 1000b (28h ≠ 7Eh, so don't add 01h)

Checksum = 28h
```

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| $7Eh$ | $20h$ | $01h$ | $30h$ | $01h$ | $03h$ | $01h$ | $02h$ | $01h$ | $28h$ |

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

ACKNOWLEDGMENT

COMMAND:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | Computed | 01h | Command | Data | See page 7 |

ACKNOWLEDGE:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 30h | 01h | 20h | 01h | 02h | 02h | See Below | None | See page 7 |

The acknowledge Command Byte is formed by adding 80h to the received Command Byte.

Example:

HEX COMMAND DESCRIPTION

05 Command to Mute

85 Acknowledge

The receiving unit transmits the “Acknowledge” response to let the sending unit know that the message has been successfully received.

The Data Byte(s) are not included in the Acknowledge message.

An Acknowledge is required after EVERY message sent, in either direction.

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

EXAMPLE OF MESSAGE EXCHANGE

The Signature 3.0 starts by sending a Request for Software Version command to the Signature 2.0

COMMAND:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 02h | 01h | 77h | None | 35h |

The Signature 2.0 acknowledges by sending an Acknowledge of Request for Software Version command to the Signature 3.0

ACKNOWLEDGE:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 30h | 01h | 20h | 01h | 02h | 01h | F7h | None | 35h |

The Signature 2.0 then sends a Response to Request for Software Version command to the Signature 3.0 with data of rev. 2.5

RESPONSE:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 30h | 01h | 20h | 01h | 04h | 02h | 77h | 0205h | 2Dh |

The Signature 3.0 then acknowledges by sending an Acknowledge of Response to Request for Software Version command to the Signature 2.0

ACKNOWLEDGE:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 02h | 02h | F7h | None | 34h |

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

BUTTON EMULATION COMMAND

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 03h | 01h | 01h | See Below | See page 7 |

| HEX | COMMAND DESCRIPTION |
|-----|---------------------|
| 00 | MONO BUTTON |
| 01 | BALANCE BUTTON |
| 02 | BACK BUTTON |
| 03 | BASS EQ BUTTON |
| 04 | FFWD BUTTON |
| 05 | REW BUTTON |
| 06 | STEREO BUTTON |
| 07 | TREBLE BUTTON |
| 08 | VOL DWN BUTTON |
| 09 | MENU BUTTON |
| 0A | ENTER BUTTON |
| 0B | PAN BUTTON |
| 0C | RIGHT BUTTON |
| 0D | VID1 BUTTON |
| 0E | VID2 BUTTON |
| 0F | UP BUTTON |
| 10 | LEFT BUTTON |
| 11 | VID3 BUTTON |
| 12 | DOWN BUTTON |
| 13 | TUNER BUTTON |
| 14 | CENTER BUTTON |
| 15 | MUTE BUTTON |
| 16 | CD BUTTON |
| 17 | DISPLAY BUTTON |
| 18 | TAPE BUTTON |
| 19 | TV BUTTON |
| 1A | VOL UP BUTTON |
| 1B | POWER ON BUTTON |
| 1C | POWER OFF BUTTON |

| HEX | COMMAND DESCRIPTION |
|-----|-------------------------------|
| 1D | NUM 0 BUTTON |
| 1E | NUM 1 BUTTON |
| 1F | NUM 2 BUTTON |
| 20 | NUM 3 BUTTON |
| 21 | NUM 4 BUTTON |
| 22 | NUM 5 BUTTON |
| 23 | NUM 6 BUTTON |
| 24 | NUM 7 BUTTON |
| 25 | NUM 8 BUTTON |
| 26 | NUM 9 BUTTON |
| 27 | STORE BUTTON |
| 28 | RDS BUTTON |
| 29 | TYPE BUTTON |
| 2A | PSET BUTTON |
| 2B | MUSIC BUTTON |
| 2C | MOVIES BUTTON |
| 2D | SIMUL BUTTON |
| 2E | DSK TRK BUTTON |
| 2F | AUX BUTTON |
| 30 | REC BUTTON |
| 31 | PSET UP BUTTON |
| 32 | PSET DWN BUTTON |
| 33 | FRONT PANEL ON OFF BUTTON |
| 34 | FRONT PANEL SOURCE UP BUTTON |
| 35 | FRONT PANEL SOURCE DWN BUTTON |
| 36 | FRONT PANEL MODE UP BUTTON |
| 37 | FRONT PANEL MODE DWN BUTTON |
| 38 | FRONT PANEL TUNE UP BUTTON |
| 39 | FRONT PANEL TUNE DWN BUTTON |
| 3A | FRONT PANEL FP MUTE BUTTON |
| 1A | FRONT PANEL VOLUME UP |
| 0E | FRONT PANEL VOLUME DOWN |

..... NOTE
This command should be used when one unit wants to emulate a remote control and control another unit (i.e. this command should be sent TO the unit intended to be controlled, FROM the controlling unit).

POWER COMMAND

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 03h | 01h | 02h | See Below | See page 7 |

| HEX | COMMAND DESCRIPTION |
|-----|---------------------|
| 01 | POWER ON |
| 02 | STANDBY |

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

MAIN VOLUME LEVEL

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 03h | 01h | 04h | See Below | See page 7 |

| HEX | COMMAND DESCRIPTION |
|-----|---------------------|
| 00 | +00 MIN. VOLUME |
| 01 | +01 |
| 02 | +02 |
| 03 | +03 |
| 04 | +04 |
| 05 | +05 |
| 06 | +06 |
| 07 | +07 |
| 08 | +08 |
| 09 | +09 |
| 0A | +10 |
| 0B | +11 |
| 0C | +12 |
| 0D | +13 |
| 0E | +14 |
| 0F | +15 |
| 10 | +16 |
| 11 | +17 |
| 12 | +18 |
| 13 | +19 |

| HEX | COMMAND DESCRIPTION |
|------|---------------------|
| 14 | +20 |
| 15 | +21 |
| 16 | +22 |
| 17 | +23 |
| 18 | +24 |
| 19 | +25 |
| 1A | +26 |
| 1B | +27 |
| 1C | +28 |
| 1D | +29 |
| thru | |
| 48 | +72 |
| 49 | +73 |
| 4A | +74 |
| 4B | +75 |
| 4C | +76 |
| 4D | +77 |
| 4E | +78 |
| 4F | +79 MAX. Volume. |

..... **WARNING**

If you are using this feature to ramp up and down the volume, you MUST make sure to set your limits so you do not roll over to either MAX. or MIN. volume levels. Speaker damage will occur if you do roll over to Max. volume.

MUTE

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 03h | 01h | 05h | See Below | See page 7 |

| HEX | COMMAND DESCRIPTION |
|-----|---------------------|
| 01 | Mute On |
| 02 | Mute Off |

SYSTEM RESET

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 02h | 01h | 57h | Not used | See page 7 |

WARNING: This command resets ALL data to factory defaults. ALL programming will be lost.

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

UNSOLICITED BUTTON COMMAND

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 99h | 01h | 30h | 01h | 03h | 01h | 71h | See Below | See page 7 |

| HEX | COMMAND DESCRIPTION |
|-----|---------------------|
| 00 | MONO BUTTON |
| 01 | BALANCE BUTTON |
| 02 | BACK BUTTON |
| 03 | BASS EQ BUTTON |
| 04 | FFWD BUTTON |
| 05 | REW BUTTON |
| 06 | STEREO BUTTON |
| 07 | TREBLE BUTTON |
| 08 | VOL DWN BUTTON |
| 09 | MENU BUTTON |
| 0A | ENTER BUTTON |
| 0B | PAN BUTTON |
| 0C | RIGHT BUTTON |
| 0D | VID1 BUTTON |
| 0E | VID2 BUTTON |
| 0F | UP BUTTON |
| 10 | LEFT BUTTON |
| 11 | VID3 BUTTON |
| 12 | DOWN BUTTON |
| 13 | TUNER BUTTON |
| 14 | CENTER BUTTON |
| 15 | MUTE BUTTON |
| 16 | CD BUTTON |
| 17 | DISPLAY BUTTON |
| 18 | TAPE BUTTON |
| 19 | TV BUTTON |
| 1A | VOL UP BUTTON |
| 1B | POWER ON BUTTON |
| 1C | POWER OFF BUTTON |

| HEX | COMMAND DESCRIPTION |
|-----|-------------------------------|
| 1D | NUM 0 BUTTON |
| 1E | NUM 1 BUTTON |
| 1F | NUM 2 BUTTON |
| 20 | NUM 3 BUTTON |
| 21 | NUM 4 BUTTON |
| 22 | NUM 5 BUTTON |
| 23 | NUM 6 BUTTON |
| 24 | NUM 7 BUTTON |
| 25 | NUM 8 BUTTON |
| 26 | NUM 9 BUTTON |
| 27 | STORE BUTTON |
| 28 | RDS BUTTON |
| 29 | TYPE BUTTON |
| 2A | PSET BUTTON |
| 2B | MUSIC BUTTON |
| 2C | MOVIES BUTTON |
| 2D | SIMUL BUTTON |
| 2E | DSK TRK BUTTON |
| 2F | AUX BUTTON |
| 30 | REC BUTTON |
| 31 | PSET UP BUTTON |
| 32 | PSET DWN BUTTON |
| 33 | FRONT PANEL ON OFF BUTTON |
| 34 | FRONT PANEL SOURCE UP BUTTON |
| 35 | FRONT PANEL SOURCE DWN BUTTON |
| 36 | FRONT PANEL MODE UP BUTTON |
| 37 | FRONT PANEL MODE DWN BUTTON |
| 38 | FRONT PANEL TUNE UP BUTTON |
| 39 | FRONT PANEL TUNE DWN BUTTON |
| 3A | FRONT PANEL FP MUTE BUTTON |
| 1A | FRONT PANEL VOLUME UP |
| 0E | FRONT PANEL VOLUME DOWN |

..... NOTE

This command should be used when one unit wants to broadcast to all other units that it has received and acted upon an IR command (i.e. this command should be sent TO all other units in the system, to let them know that someone has pushed a button on the front panel or remote control and changed something about the broadcasting unit.)

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

CONTENTS OF NON-VOLATILE MEMORY REQUEST

COMMAND:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 06h | 01h | 72h | See below | See page 7 |

The Contents of Non-Volatile Memory Request command causes the Target unit to transmit, in the response, the contents of its non-volatile memory. Two values are included in the data segment of the command: an “Offset” data word that determines how many bytes from the beginning of the Non-Volatile memory structure that the desired data begins, and a “Length” data word that determines the total number of bytes to be transmitted back to the Source unit.

NOTE: sending a “Length” data word value of zero will cause the Target unit to respond with the rest of the structure (from offset to the end) - the number of bytes returned in the response will reflect the actual number of bytes involved.

RESPONSE:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 30h | 01h | 20h | 01h | Computed | 02h | 72h | See below | See page 7 |

```
uint      spkr_cfg;           // current speaker configuration
uchar     spkr_mutes;        // mask bits for speaker group muting
char      Volume_preset;    // power-on volume setting
Spkr_adj_st  spkr_adj;      // speaker level adjustments
          signed char      center_level;
          signed char      front_right_level;
          signed char      front_left_level;
          signed char      back_right_level;
          signed char      back_left_level;
          signed char      subwfr_level;
          char              front_distance;
          char              cen_distance;
          char              back_distance;

char      source;           // currently active source
uint      opFlag;          // general purpose operational flags
char      svidRecSrc;      // used when tracking disabled
char      compvidRecSrc;   // used when tracking disabled
char      audRecSrc;       // used when tracking disabled
char      dsplyTimeoutVal; // seconds counter for VFD/OSD dsply timeout
char      frntPnlSts;      // Off, On, or Timeout
char      frntPnlBrit;     // Dim, Med or High
char      menuBkgrnd;
```

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

CONTENTS OF NON-VOLATILE MEMORY REQUEST (CONT.)

```
Profile_st      defaultProfile;           // the "Current" f/x profile
char            bass_eq;           // bass EQ level
char            hi_eq;            // treble cut level
char            centr_mod;        // center mode
char            surr_mod;         // surr mode
char            centr_lvl_mod;    // level modifier for current center mode
char            surr_lvl_mod;     // level modifier for current surround mode
signed char     sub_lvl_mod;      // level modifier for subwoofer
Balance_st      balance;         // balance data
                signed char lrBal; // item to be adjusted
                signed char fbBal; //
Input_st        Input[NUM_INPUTS];
char            name[INPUT_NAME_WIDTH]; // input name
char            fxMode;          // index to the fx mode struct
Profile_st      profile;        // profile for this input
char            bass_eq;        // bass EQ level
char            hi_eq;         // treble cut level
char            centr_mod;     // center mode
char            surr_mod;      // surr mode
char            centr_lvl_mod; // level modifier for current center mode
char            surr_lvl_mod;  // level modifier for current surround mode
signed char     sub_lvl_mod;   // level modifier for subwoofer
Balance_st      balance;      // balance data
                signed char lrBal; // item to be adjusted
                signed char fbBal; //
uchar           audio_src;
char            video_src;
char            video_mode;     // S-video or composite video
char            input_level_left; // left input level for this input
char            input_level_right; // left input level for this input
char            graphics;      // menu background = blue or video
uint            flags;
```

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

CONTENTS OF NON-VOLATILE MEMORY REQUEST (CONT.)

```
TunerPSet_st  TnrPSet[NUM_PRESETS_TUNER];
               int          nFreq;                // station frequency
               char         cBand;                // USA_FM, USA_AM, EUR_FM, EUR_AM
               char         cRDS;                // flag - rds auto or manual. if manual, user
                                                // can store ID & PTY
               char         szID[RDS_TYPE_SIZE]; // 8 char station id
               char         cType;                // index into RDSTypeTable string table
TunerState_st  TnrState;
               int          nCountry;
               int          nCurBand;
               int          nCurFreq;
               int          nCurPreset;
               int          nCurFMFreq;
               int          nCurAMFreq;
               int          nCurLWFreq;
char           RDSTblGapLow;                    // describe the 'hole' in the...
char           RDSTblGapHi;                    // ...RDS type table
char           unitID;                         // Signature unit ID
char           pwrUpState;                     // describe the power-up state
uint           version;                        // software revision level
char           filler[13];                     // unused at this time
```

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

WRITE SETTINGS INTO NON-VOLATILE MEMORY

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| <i>7Eh</i> | <i>20h</i> | <i>01h</i> | <i>30h</i> | <i>01h</i> | <i>03h</i> | <i>01h</i> | <i>74h</i> | None | See page 7 |

This command causes the unit to write its non-volatile settings into non-volatile memory.

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

CONTENTS OF VOLATILE MEMORY REQUEST

COMMAND:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | 06h | 01h | 75h | See below | See page 7 |

The Contents of Volatile Memory Request command causes the Target unit to transmit, in the response, the contents of its volatile memory. Two values are included in the data segment of the command: an "Offset" data word that determines how many bytes from the beginning of the Volatile memory structure that the desired data begins, and a "Length" data word that determines the total number of bytes to be transmitted back to the Source unit.

NOTE: sending a "Length" data word value of zero will cause the Target unit to respond with the rest of the structure (from offset to the end) - the number of bytes returned in the response will reflect the actual number of bytes involved.

RESPONSE:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 30h | 01h | 20h | 01h | Computed | 02h | 75h | See below | See page 7 |

```
char      bass_eq;      // bass EQ level
char      hi_eq;        // treble cut level
char      centr_mod;    // center mode
char      surr_mod;     // surr mode
char      centr_lvl_mod; // level modifier for current center mode
char      surr_lvl_mod; // level modifier for current surround mode
signed char sub_lvl_mod; // level modifier for subwoofer
Balance_st balance;    // balance data
          signed char lrBal; // item to be adjusted
          signed char fbBal; //
```

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

SOFTWARE REVISION LEVEL REQUEST

COMMAND:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 20h | 01h | 30h | 01h | Computed | 01h | 77h | None | See page 7 |

RESPONSE:

| Preamble Byte | Target Device Type | Target Unit ID | Source Device Type | Source Unit ID | Data Length | Command Modifier | Command Byte | Data Byte(s) | Checksum Byte |
|---------------|--------------------|----------------|--------------------|----------------|-------------|------------------|--------------|--------------|---------------|
| 7Eh | 30h | 01h | 20h | 01h | Computed | 02h | 77h | See below | See page 7 |

The Software Revision Level Request command causes the target unit to transmit, in the response, the revision level of its software. The response is a data word organized as follows:

MSB Major revision level

LSB Minor revision level

Example: 0205h would be software version 2.5

| DB-25 | DCE | DB-9 | | | |
|-------|-----|------|----|---|-------------------|
| 1 | | | AA | x | Protective Ground |
| 2 | TXD | 3 | BA | I | Transmitted Data |
| 3 | RXD | 2 | BB | O | Received Data |
| 4 | RTS | 7 | CA | I | Request To Send |
| 5 | CTS | 8 | CB | O | Clear To Send |
| 6 | DSR | 6 | CC | O | Data Set Ready |

SIGNATURE BUS PROTOCOL MASTER DOCUMENT

Revised 2-24-98

| | | | | | |
|----|-------|---|----|-----|--------------------------------|
| 7 | GND | 5 | AB | x | Signal Ground |
| 8 | CD | 1 | CF | O | Received Line Signal Detector |
| 9 | | | -- | x | Reserved for data set testing |
| 10 | | | -- | x | Reserved for data set testing |
| 11 | | | | x | Unassigned |
| 12 | SCF | | | O | Secndry Rcvd Line Signl Detctr |
| 13 | SCB | | | O | Secondary Clear to Send |
| 14 | SBA | | | I | Secondary Transmitted Data |
| 15 | DB | | | O | Transmisn Signl Elemnt Timng |
| 16 | SBB | | | O | Secondary Received Data |
| 17 | DD | | | O | Receiver Signal Element Timing |
| 18 | | | | x | Unassigned |
| 19 | SCA | | | I | Secondary Request to Send |
| 20 | DTR | 4 | CD | I | Data Terminal Ready |
| 21 | CG | | | O | Signal Quality Detector |
| 22 | | 9 | CE | O | Ring Indicator |
| 23 | CH/CI | | | I/O | Data Signal Rate Selector |
| 24 | DA | | | I | Transmit Signal Element Timing |
| 25 | | | | x | Unassigned |

