

# **AES Recommended practice for sound-reinforcement systems — Communications interface (PA-422)**

*Published by*

**Audio Engineering Society, Inc.**

Copyright ©1991 by the Audio Engineering Society

## **Abstract**

This Audio Engineering Society (AES) Standard specifies the electrical characteristics of a balanced-voltage circuit for the interchange of serial binary signals for the control of sound-reinforcement systems. It provides for interchange among data terminal equipment (DTE), that is, computers and microprocessors, and data circuit-terminating equipment (DCE). PA-422 is a mnemonic, signifying professional audio implementation of Electronics Industries Association EIA-422-A. Device control language is provided in an annex.

An AES standard implies a consensus of those directly and materially affected by its scope and provisions and is intended as a guide to aid the manufacturer, the consumer, and the general public. The existence of an AES standard does not in any respect preclude anyone, whether or not he or she has approved the document, from manufacturing, marketing, purchasing, or using products, processes, or procedures not in agreement with the standard. Prior to approval, all parties were provided opportunities to comment or object to any provision. Approval does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards document. This document is subject to periodic review and users are cautioned to obtain the latest edition.

Contents

Foreword.....3

1 Scope.....4

2 Applicability .....4

3 Electrical characteristics .....5

3.1 Configuration .....5

3.2 Driver.....5

3.3 Communications settings .....6

3.4 Interface connections .....6

4 Communication.....6

4.1 Intelligent devices .....6

4.2 Nonintelligent devices .....8

5 Interconnecting cable characteristics .....10

5.1 Cable type .....10

5.2 Cable length .....10

6 References.....10

ANNEX A.....11

Courtesy copy for personal information only  
 www.aes.org/standards

[This foreword is not a part of AES Recommended practice for sound-reinforcement systems — Communications interface (PA-422), AES15-1991.]

### Foreword

This standard has been prepared by the Working Group on Sound System Control, a working group of the Audio Engineering Society Standards Committee. It is designed in the public interest to eliminate misunderstandings between manufacturers, consultants, purchasers, and users, and to facilitate interchangeability, integration, and improvement of products, thus allowing the selection and utilization of the proper product for a particular need.

The standard may involve patents on articles, materials, or processes. The AES assumes no liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standard.

The following individuals have contributed to the preparation of this document: Steve Bartlett, Mitchell Bentz, Mahlon Burkhard, Larry Driskill, William Edwards, Ali Elwakhi, Dan Garrigan, Gary Hardesty, John Johnson, Hardy Martin, Richard Mitchell, Bill Murphy, Charles Richmond, Gene Rimkeit, Robert Rodgers, Tom Roseberry, Kelly Vickery, Saul Walker, and Edward Young.

TOM ROSEBERRY, *Chairman*  
AES Standards Committee Working Group on Sound System Control  
1990 May

At the time of approval of this document for publication, the AES Standards Committee had the following membership: Yoshi-Haru Abe, James S. Brawley, Richard C. Cabot, Peter D'Antonio, Donald Eger (Chair), Robert A. Finger, Irving Joel, William Hogan, Tomlinson F. Holman, Mike Klasco, David L. Klepper, Bart N. Locanthi, J.P. Nunn, Daniel Queen (Secretary), Tom Roseberry, W.T. Shelton, William D. Storm, Ted Telesky, Han Tendeloo, and Floyd E. Toole.

The American National Standards Institute version of this standard has not been reprinted and remains available as ANSI S4.49-1991.

# AES Recommended practice for sound-reinforcement systems — Communications interface (PA-422)

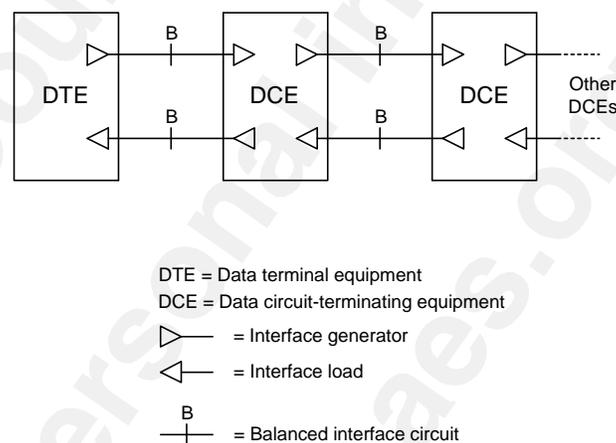
## 1 Scope

This standard specifies the electrical characteristics of the balanced-voltage circuit that may be used when specified for the interchange of serial binary signals among data terminal equipment (DTE), that is, computers and microprocessors, and data circuit-terminating equipment (DCE) of, or in, any point-to-point interconnection of serial binary signals among digitally controlled equipment. The standard contains a device control language (DCL) for controlling communication among the various products. Annex A covers this device control language. The interface includes a driver and an interconnecting cable to a receiver. The electrical characteristics of the circuitry are similar to and expected to perform in the same manner as those of the Electronics Industries Association standard EIA-422-A, hence the mnemonic, PA-422 for professional audio use.

Minimum performance requirements for the interconnecting cable are furnished. Guidance is given with respect to limitations on the data rate imposed by the parameters of cable length, balance, and termination for individual installations.

## 2 Applicability

The provisions of this standard may be applied to the interfaces used among equipment where the information being conveyed is in the form of serial binary signals at the dc baseband level. This standard shall be referenced by the specifications and specific interface standards applying these electrical characteristics. Typical points of applicability for this standard are depicted in figure 1.



**Figure 1 – Applications of communications interface**

The balanced-voltage digital interface should be utilized on data, timing, or control circuits in appropriate applications. This is to ensure the elimination of any outside electrical interference which might disrupt the operation. The data rate of the signals can range up to over  $10^6$  bits per second. The maximum rate for an individual application is dependent on the cable type and length. While a restriction on maximum cable length is not established, guidelines are given with respect to conservative operating distances as a function of the data signaling rate (see Section 5).

In general, these conservative values may be exceeded where the installation is engineered to ensure that noise and ground potential values are held to a minimum.

In this standard, the term data signaling rate, expressed in units of bits/s, is the significant parameter. Data signaling rate is defined as  $1/T$ , where  $T$  is the minimum interval between two significant instants. In a binary system for which this standard is to apply, the data signaling rate, in bits per second, and the modulation rate, in baud, are numerically equal when the unit interval used in each determination is the minimum interval.

### 3 Electrical characteristics

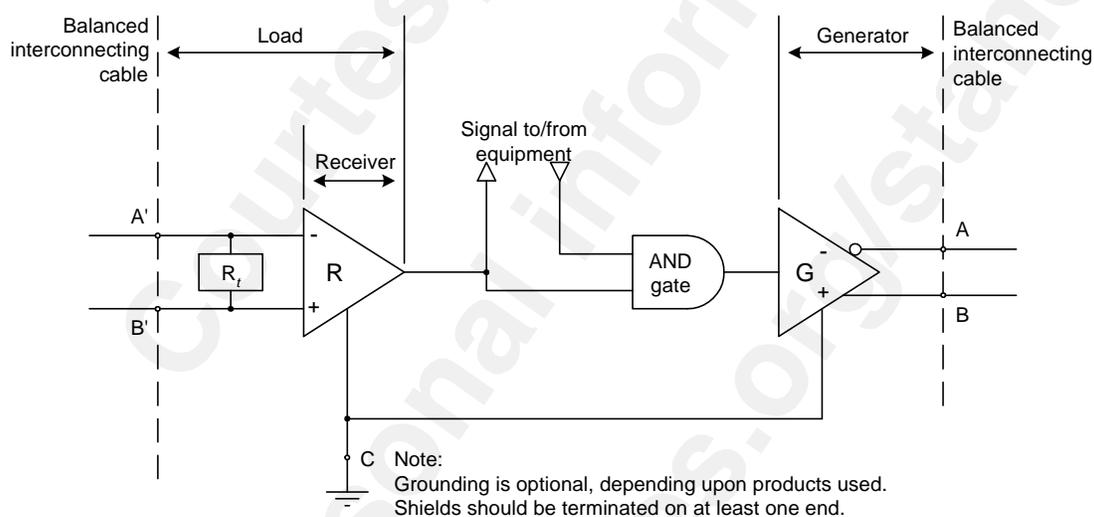
#### 3.1 Configuration

The balanced-voltage digital circuitry consists of three parts: the driver, the balanced interconnecting cable, and the load. For the electrical characteristics of these elements reference should be made to section 4.0 of EIA-422-A for balanced-voltage digital-interface circuits. Figure 2 shows the physical connections of a typical DCE for an interface with multiple DCEs. The nominal differential operating voltage between the balanced signal lines is 5 V and half of that voltage to ground. This is the operating voltage specified for the devices contained in the circuit shown in figure 2. The signaling sense of the voltages appearing across the interconnection cable is defined as follows.

**3.1.1** Terminal A of the driver shall be negative with respect to terminal B for a binary 1 (OFF) state.

**3.1.2** Terminal A of the driver shall be positive with respect to terminal B for a binary 0 (ON) state.

NOTE—Reversing of the cable pair for a particular circuit will cause inversion of the signals.



$R_t$  = Cable termination resistance

A,B = Generator interface points

A',B' = Load interface points

C = Circuit ground

**Figure 2 – Connections of multiple equipment**

#### 3.2 Driver

The driver has the capability to furnish the signal necessary to supply many parallel receivers. However, for the physical arrangement of multiple equipment throughout a facility it is best to use the method of linking depicted in figure 1.

### 3.3 Communications settings

The interface described in this standard utilizes a serial data bit transmission conforming to EIA-422-A, with one start bit, eight data bits, a parity bit, and one stop bit. The standard data rate used for this interface is 19 200 baud (or bit/s), although other data rates may be used provided that the nonstandard rate can be accommodated by all equipment on the interface.

### 3.4 Interface connections

The circuits used by this interface are listed in table 1 along with the sources (driver end) of the signal and the pin numbers for a nine-pin D-subminiature-type connector.

**Table 1 – Circuits used by interface**

Symbol	Description	Source	Terminal	
			A (-)	B (+)
GND	Protective ground	—	3	—
TxD	Transmitted data	DTE	9	8
RxD	Received data	DCE	5	4
DTR	Data terminal ready	DTE	7	6
DSR	Data set ready	DCE	2	1

## 4 Communication

### 4.1 Intelligent devices

The signal characteristics of an intelligent device (programmable element such as a microprocessor or microcontroller) applicable to this standard are shown in figure 3 and the terms defined in table 2. Figure 3 depicts the sequence of events in a typical transmission of data to and/or from a DCE and a DTE. The communication contains certain protocols that were established to be compatible with an intelligent programmable device. The initial conditions and the sequence of events as shown in figure 3 are as follows.

When there is no transmission under way, TxD, RxD, DSR, and DTR are in the OFF state on all equipment, including the DTE. The DTE initiates a transmission by sending an 8-bit equipment address code (EAC) on the TxD circuit.

The DCE unit corresponding to that EAC detects its address and responds by setting DSR to the ON state.

NOTE – If DTR is set, other devices not involved in the conversation must remain in the idle state to avoid conflict.

The DTE waits for DSR to be set to the ON state, indicating that a DCE has decoded the EAC, and then sets DTR to the ON state. The DCE detects the DTR change at its DSR input, and returns 2 bytes of data to the DTE. The first byte is its device-type (DT) code and the second is its manufacturer's identification (ID) code.

After verifying that the DT and ID codes are valid, the DTE transmits the first command to the device.

NOTE—The DCE's interpretation of the command determines the number of bytes (if any) that will follow. Some command codes have been preassigned. Their descriptions, along with examples of their usage, are given in Annex A. All other DCEs connected to the interface must ignore any data received since their address did not match the EAC.

The DTE then transmits the sequence of data bytes to the DCE on the TxD circuit.

NOTE—The type and quantity of device-specific data are defined by the manufacturing organization.

Once the data are received, the DCE returns a communications status (COMSTAT) code, resets its DTR output line, and resumes an idle state. If the returned COMSTAT code indicates a successful communication, the DTE verifies that DSR was reset. Then it resets its DTR output line and resumes an idle state. The communication with the device has concluded.

The capability of a DCE device to initiate communication with the DTE and another DCE shall be allowed. This function provides flexibility and information for appropriate applications. With reference to DT codes, unique 8-bit codes shall be assigned to all programmable devices. Refer to Annex A for a list of currently assigned DT codes.

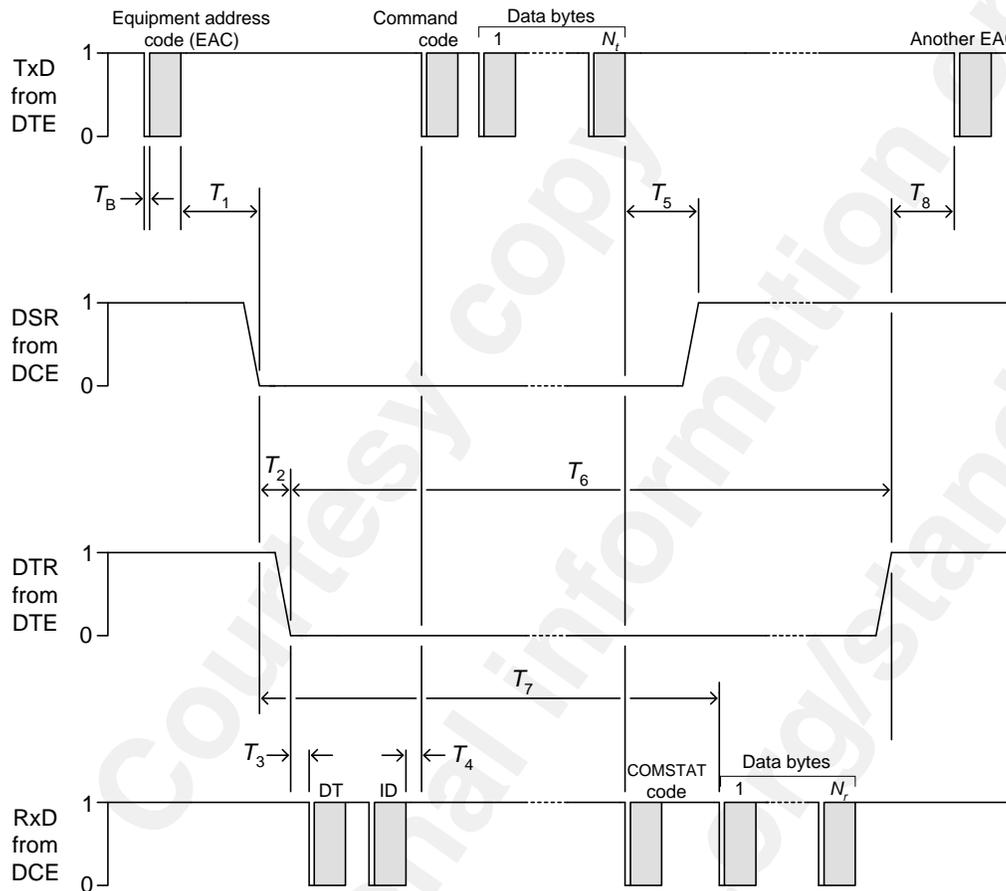


Figure 3 – Typical data transmission, programmable device

Table 2–Communications interface definitions, intelligent device

Symbol	Minimum	Maximum	Description
$T_1$	0.0	$625 \mu s^\dagger$	Address acknowledge time
$T_2$	0.0		Acknowledge to DSR response time
$T_3$	$T_B^*$	$625 \mu s^\dagger$	DTR to DT and ID data response time
$T_4$	0.0		DT and ID to command code
$T_5$	$T_B^*$		End of data to DSR response time
$T_6$	0.0		DTR active time
$T_7$	0.0		DSR to end of COMSTAT code
$T_8$	$T_B^*$		Next transmission delay

\*  $T_B$  = data signaling period = 1/baud rate; for 19 200-baud systems,  $T_B = 52 \mu s$ ;  $N_t$  = number of data bytes transmitted;  $N_r$  = number of data bytes received.

† These values may be larger for some applications, provided that the DTE can accommodate a greater delay.

A unique ID code shall be assigned to each participating manufacturing organization. The ID codes are also listed in Annex A.

The check for valid DT and ID codes helps ensure that the device receives valid instructions and data.

The DCE shall be designed to ignore an invalid command or data, then resume an idle state after returning an appropriate COMSTAT error code.

Any invalid response by a DCE shall cause the DTE to declare a fault and resume an idle state. This includes possible hardware errors, such as the failure of a device to set or reset its DTR output line. The DTE shall be designed to wait a specified period of time before declaring a device fault. If a device does not respond within this time frame, a failure is assumed and the controlling program can report the error condition.

NOTE—Refer to Annex A for further information relating to transmission errors, device control language, and associated software programming.

#### 4.2 Nonintelligent devices

For nonintelligent devices, that is, nonprogrammable or fixed-logic or both, the following applies. Figure 4 gives the sequence of events. The terms are defined in table 3.

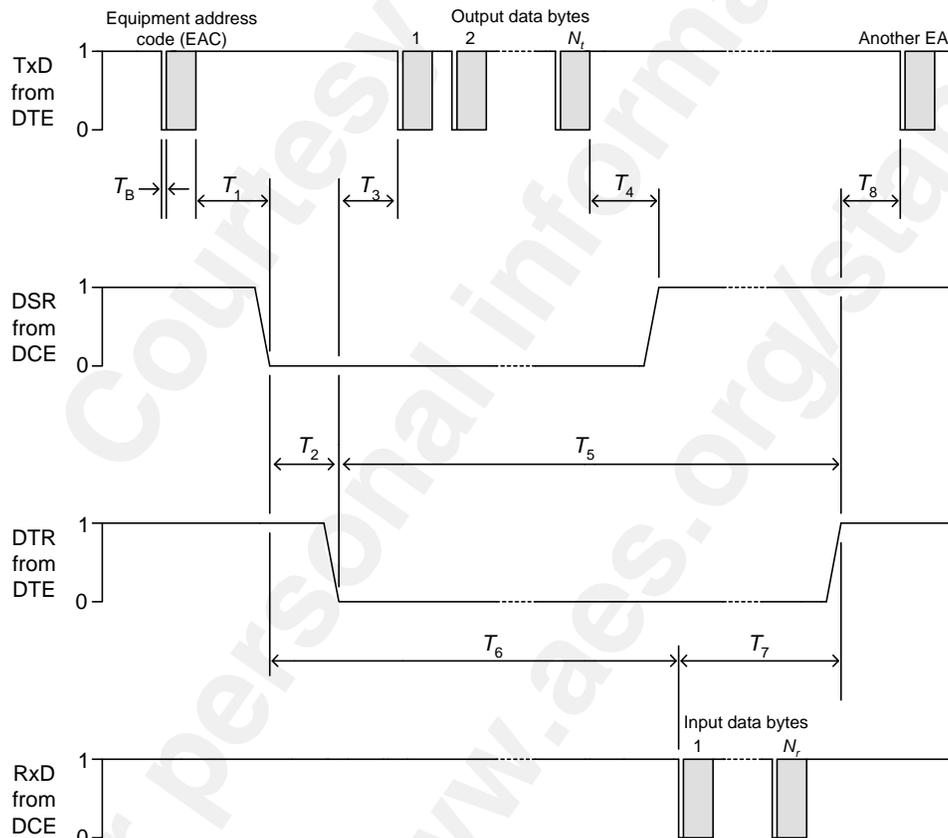


Figure 4— Typical data transmission, nonintelligent device

**Table 3—Communications interface definitions, nonintelligent device**

Symbol	Minimum	Maximum	Description
$T_1$	0.0	625 $\mu\text{s}^\dagger$	Address acknowledge time
$T_2$	0.0		Acknowledge to DSR response time
$T_3$	$T_B^*$		DTR to Tx data relay
$T_4$	0.0	625 $\mu\text{s}^\dagger$	End of data to DSR response time
$T_5$	$T_B^*$		DTR active time
$T_6$	0.0		DSR to input data relay
$T_7$	0.0		End of input data to DTR inactive time
$T_8$	$T_B^*$		Next transmission delay

\*  $T_B$  = data signaling period = 1/baud rate; for 19 200-baud systems,  $T_B = 52 \mu\text{s}$ ;  $N_t$  = number of data bytes transmitted;  $N_r$  = number of data bytes received.

$^\dagger$  These values may be larger for some applications, provided that the DTE can accommodate a greater delay.

When there is no transmission under way, TxD, RxD, DSR, and DTR are in the OFF state on all equipment, including the DTE. The DTE initiates all transmission by sending an 8-bit EAC on the TxD circuit. The DCE unit corresponding to that EAC detects its address and responds by setting DSR to the ON state. If DTR is set, other devices not involved in the conversation shall remain in the idle state to avoid conflict.

NOTE—The DTE waits for DSR to be set to the ON state, indicating that a DCE has decoded the EAC, and then sets DTR to the ON state. The DCE detects DTR in the ON state and makes itself ready to receive data from the DTE. All other DCEs connected to the interface must therefore ignore any data received, since their addresses will not match the EAC sent.

The DTE transmits a sequence of data bytes to the DCE on the TxD circuit. The number of words transmitted is dependent on the requirements of the DCE.

The DCE receives the data bytes and then, after the correct number of bytes have been sent, sets DSR to the OFF state

NOTE—The DCE can transmit data as required on the RxD data circuit. The number of bytes sent is dependent on the requirements of the DCE and may be any number, including zero. The DTE detects DSR in the OFF state and receives any data bytes sent by the DCE. The DTE sets DTR to the OFF state, indicating the end of the transmission.

The number of nonintelligent data bytes in each transmission is dependent on the requirements of the DCE. Each DCE shall have a specific number of data bytes associated with its function and shall require exactly that number of bytes to be transmitted and/or received. The DTE should check for DSR to be set to the ON state by the DCE within a specified period of time after the EAC for that DCE has been sent. If DSR is not set during this time, the DTE should declare a “time-out fault” and terminate the transfer without sending any data to the DCE.

A polling mode is included. In the polling mode, the DTE sends an EAC for a particular DCE and waits for the DSR response from the DCE. The DTE then strobos DTR (set to the ON state and then to the OFF state) without transmitting any data, and the DCE would simply reset the DSR. Also, if at any time during a normal transmission, the DTE resets DTR to the OFF state, the DCE should stop receiving or transmitting data and set DSR to the OFF state. This feature allows the DTE to poll all or some of the DCEs on a periodic basis in order to test the integrity of the interface.

The number of individual DCE units that can be connected to a single interface is limited only by the number of unique address combinations in the 8-bit EAC, which is 250. The address of an individual DCE is established in the DCE by any of various means, such as with a DIP switch, jumpers, or storing the address in nonvolatile RAM.

## 5 Interconnecting cable characteristics

### 5.1 Cable type

The characteristics of the interconnecting cable are not specified. To ensure proper operation, however, twisted-pair cable with metallic conductor should be used. The performance of any interconnecting cable used shall be such as to maintain the necessary signal quality required for the specific application. Shielded cable may be the best choice. When using interconnecting equipment with the electrical interface characteristics specified in this standard and those of EIA-422-A, consideration should be given to the problems that may be encountered due to the interconnecting cable characteristics, cable termination resistance, optional grounding arrangements, and interconnection with interfaces using different circuitry than that of figure 2.

### 5.2 Cable length

The length of the cable separating the various drivers and receivers is a function of the data signaling rate and is influenced by the tolerable signal distortion, the amount of longitudinally coupled noise, and ground potential difference introduced between the driver and receiver circuit grounds, as well as by the cable balance. Increasing the physical separation and interconnecting cable length among driver and receiver interfacing points increases exposure to common-mode noise, signal distortion, and the effects of cable unbalance and loading.

NOTE—EIA-422-A shows in its Appendix A on cable length guidelines, fig. A.1, a cable length of 1200 m (4000 ft) for a 19 200-baud rate. The information contained in the figure is based on empirical data using a 24 AWG copper-conductor, twisted-pair telephone cable with a shunt capacitance of 52 pF/m (16 pF/ft) terminated in a 100- $\Omega$  resistive load. This cable length determination was based on assumed received signal quality requirements concerning the rise and fall times of the modulation signal and the voltage loss of the cable (6 dB being the value used for the data in the EIA standard).

## 6 References

EIA-422-A, *Electrical characteristics of balanced voltage digital interface circuits*, Electronics Industries Assoc., Washington, DC (1978 Dec.).

[The following annex is not part of the AES Recommended practice for sound-reinforcement systems — Communications interface (PA-422), AES15-1991. It is included for information purposes only.]

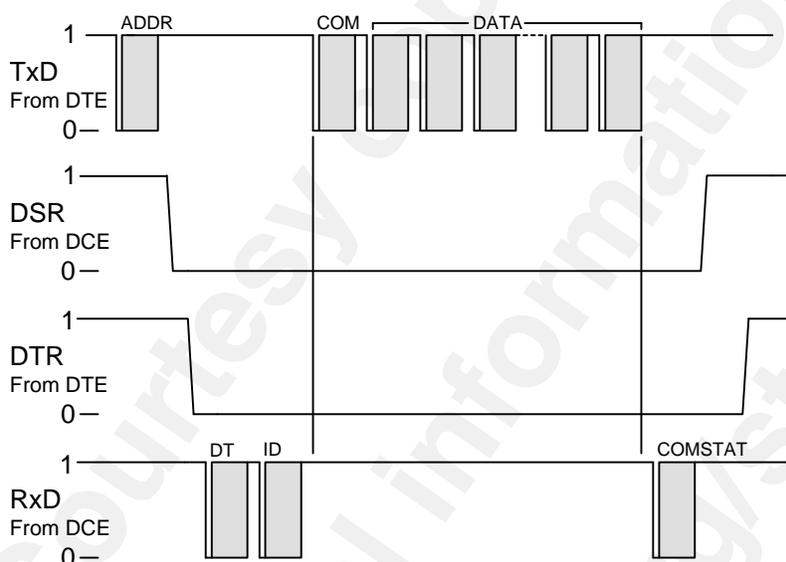
## ANNEX A

(Informative)

### Device control language for controlling communication among professional sound products

#### A.1 Overview of procedures for device communications

In order to have an effective and efficient communication between the computer system and a programmable device, the procedures listed hereafter must be followed. It may be helpful when reading them to refer to the event sequence timing diagram shown in figure A.1. Some explanations will follow the listings.



**Figure A.1—Event sequence timing diagram of communications protocol**

NOTE—In figure A.1, DTE (data terminal equipment) is the computer system and DCE (data circuit-terminating equipment), the programmable device.

- 1) Initially, TxD, RxD, DTR, and DSR are in their inactive (normally high) states.
- 2) The computer system transmits an 8-bit address code to the devices on TxD.
- 3) The device with the matching address responds by toggling DSR to its active (low) state. This also signals other devices to remain in their idle states<sup>1</sup>, thus avoiding any conflict.
- 4) The computer system detects the change in state of DSR and toggles DTR to the active (low) state.
- 5) The device detects the change in state of DTR and returns 2 bytes of data to the computer system. The first byte is the device-type (DT) code followed by the manufacturer's identification (ID) code.
- 6) After verifying that DT and ID are valid, the computer system transmits a command to the device.
- 7) Now a data transfer typically occurs. The flow can be in either direction, but the type and quantity of device-specific data must be defined by the manufacturer.

NOTE—The device's understanding of the command defines the number of bytes (if any) to receive or transmit. Therefore it is not necessary to transmit a code denoting the number of bytes to follow.

<sup>1</sup> An idle state is the condition or state that existed prior to a communication.

- 8) Once the transfer is complete, the device returns a communications status (COMSTAT) code, resets DSR, and resumes an idle state.
- 9) After the COMSTAT code is received, the computer system verifies that DSR was reset, resets DTR, and resumes an idle state. The communication with the device has concluded.

With reference to item 5), unique DT codes must be assigned to all PA-422 programmable devices. Each code definition should be generally descriptive of the particular device. Groups or blocks of codes may be reserved for device categories. For example, the range of codes from 00 hex (hexadecimal) to 0F hex is reserved for programmable equalizers, and 00 hex specifically defines a dual-channel third-octave equalizer. Other DT codes and categories are listed in A.6.

Since DT codes are 8 bits long, it would seem that there can only be 256 different types of devices. The reserved code FF hex can be used to increase the number of unique device types to 1019. When received, the controlling software program will add 255 to the next byte. For example, if a device returned FF hex followed by 02 hex, the program would calculate 257 (FF hex + 2) for the DT code. Had FF hex been returned a second or third time in succession, the program would have added 510 or 765, respectively, to the next byte. FF hex can be returned a maximum of three times (up to four total bytes returned).

NOTE—Since FF hex is reserved, a device whose DT code equals 255 would return FF hex followed by 00 hex. Otherwise the program would expect a second incoming byte. The same technique is used with DT codes equal to 510 and 765.

Although other techniques exist to expand the number of unique codes, this technique seems the simplest and does not force the program to look for a second incoming byte. Typically, only a global routine, such as a polling routine, needs to test for FF hex. Software written for a particular device already knows what code to expect.

A unique ID code has been assigned to each participating organization. Those currently assigned are listed in A.6. Any interested organization can receive an ID code by requesting one in writing. Refer to A.5 for additional information. The reserved code FF hex can be used as above to uniquely identify 1019 manufacturers.

Verifying the correctness of the DT and ID codes helps ensure that software written for one manufacturer's device will not inadvertently reprogram or disrupt the operation of another's. It also allows the controlling software program to query each device on line to determine its type and make, thus distinguishing it from another of the same type. As future applications develop, each combination of DT and ID codes could be used to expand the number of possible commands, or to define entire sets of commands specific to a particular device.

COMSTAT codes are used to report error conditions. Errors may be caused by line problems, such as an intermittent cable connection, or invalid commands or data. A device receiving any invalid information should return an appropriate COMSTAT code and resume an idle state. Some COMSTAT codes have been defined and are listed in A.6. Other codes may be defined as needed (refer to A.5).

The controlling software program should report a fault when one of the following conditions occurs.

- No response from a device. This is usually caused by an open address location, such as no device present at that address. If DSR is not set within 250 ms, a time-out error should be reported.
- Invalid response from a device. This is usually the result of a hardware or cabling problem.
- Device returned a COMSTAT error code. Depending on the specific code, this could be a hardware or a cabling problem, or the result of transmitting invalid information to the device.

## A.2 General framework for the design of PA-422 programmable devices

Before an effective device control language (DCL) can be developed, a general framework or set of guidelines relating to the design of programmable devices must be established. For example, each device must have at least one channel, but is allowed as many as 255 independent channels. The channels are accessed using channel number (CHNUM) codes which range from 01 hex, corresponding to channel 1, to the maximum number provided by the organization responsible for its design.

Each channel must have one nonvolatile memory<sup>2</sup> for the default or power-up settings, but is allowed as many as 255 additional preset or user memories. The memories are accessed using memory number (MEMNUM) codes, which range from 00 hex, corresponding to user memory 0, to the maximum number provided. User memory 0 is defined as the default memory (DEF) and always contains the active, or live, settings for the channel.

Each channel should reserve another byte in nonvolatile RAM for a memory pointer (MPTR). The memory pointer is assigned a value which specifies (points to) one of the user memories, the contents of which can be used to update the default memory. Its usage is explained in A.3.4 and A.3.5.

NOTE—There is only one memory pointer per channel, not one pointer per user memory.

Valid memory pointer values range from 00 hex to the maximum number of user memories provided. 00 hex is defined as the pointer's OFF state, 01 hex means that the memory pointer is pointing to user memory 1, 02 hex to user memory 2, and so on.

The device (not necessarily each channel) should reserve another byte in nonvolatile RAM for an operational status (OPSTAT) code. Devices able to perform self-diagnostic testing, for example, could store the result as a code in OPSTAT memory, where it could be read by the computer system. At this time, only one OPSTAT code is defined; 00 hex means that the device is functioning properly. FF hex is reserved. Other OPSTAT codes will have to be defined as more research is done.

Although few manufacturers, if any, will provide the maximum number of channels (255) and user memories (256 per channel), devices with only one channel and one memory should still adhere to the proposed syntax, that is, CHNUM should be set to 01 hex and MEMNUM to 00 hex (default memory).

### A.3 Set of basic device control language commands

The commands presently defined are fundamental and should work with any intelligent device. One major goal is to assure that all DCL commands are as generic as possible. This permits the use of identical syntaxes between devices, regardless of their types. As a result, the only difference is the actual device-specific data transmitted or received.

In general, the code numbers assigned to DCL commands indicate the direction of data flow. Codes from 80 hex to FF hex connote a transfer of information away from the controller; codes less than 80 hex connote a transfer toward the controller. All commands, assignments, codes, and so on, are summarized in A.6.

In the following command descriptions it is assumed that the controlling software program provides the means to alter the device-specific data and select the device address (ADDR), CHNUM, MEMNUM, and MPTR value.

NOTE—All devices are not required to respond to all of the commands. Each organization involved in the design of a device must determine for itself which commands should be implemented.

#### A.3.1 Transmit all data to device channel — 81 hex

This command transmits all of the device-specific data and the memory pointer value of the specified channel and user memory. If the default memory is specified, the channel will be updated with the new settings.

Figure A.2 is a flowchart for the command, figure A.3 the event sequence timing diagram. It will be helpful to examine the two figures closely when reading each step. To execute the command, the device controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).

<sup>2</sup> Each channel memory must be large enough to contain all of the programmable parameters specific to the device.  
2000-03-15 printing

- 5) Get ID (manufacturer's ID code).
- 6) Transmit 81 hex (command code).
- 7) Transmit CHNUM (user specified).
- 8) Transmit MEMNUM (user specified).
- 9) Transmit all device-specific data.
- 10) Transmit MPTR (user specified).
- 11) Get COMSTAT (should be 00 hex if successfully executed).
- 12) Check that DSR is reset.
- 13) Reset DTR.

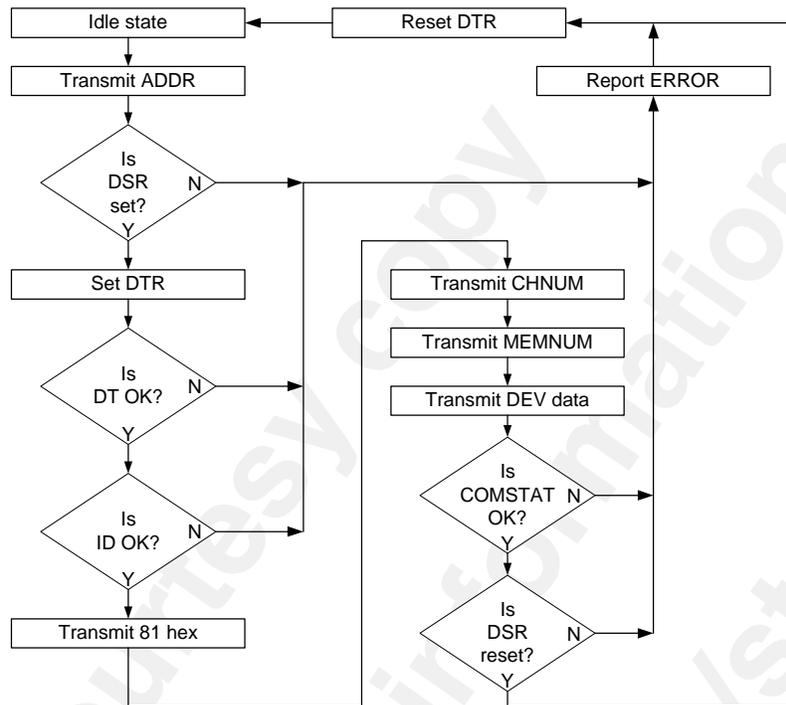


Figure A.2—Flowchart of TRANSMIT ALL DATA TO DEVICE CHANNEL command

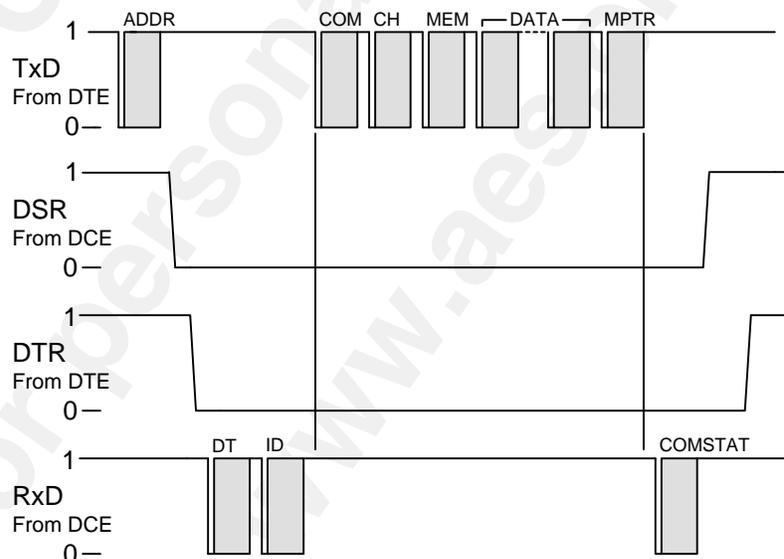


Figure A.3—Event sequence timing diagram of TRANSMIT ALL DATA TO DEVICE CHANNEL command

**A.3.2 Reprogram device channel from memory — 82 hex**

This command updates the default memory of the specified channel with the settings previously stored in the specified user memory. To update the channel from a preset memory, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 82 hex (command code).
- 7) Transmit CHNUM (user specified).
- 8) Transmit MEMNUM (user specified).
- 9) Get COMSTAT (should be 00 hex if successfully executed).
- 10) Check that DSR is reset.
- 11) Reset DTR.

NOTE—Any DCL command not involving a transfer of device-specific data should work with any device regardless of its type.

**A.3.3 Transmit pointer to device channel — 83 hex**

This command transmits a new memory pointer value to the specified channel. To transmit a new pointer value, the device controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT code (device-type code).
- 5) Get ID code (manufacturer's ID code).
- 6) Transmit 83 hex (command code).
- 7) Transmit CHNUM (user specified).
- 8) Transmit MPTR (user specified).
- 9) Get COMSTAT (should be 00 hex if successfully executed).
- 10) Check that DSR is reset.
- 11) Reset DTR.

**A.3.4 Reprogram device channel from pointer — 84 hex**

This command updates the default memory of the specified channel with the settings previously stored in the user memory location specified by the memory pointer. If the memory pointer specifies (points to) user memory 1, the channel's default memory will be updated with the contents of user memory 1. If the memory pointer's value is zero, its OFF state, nothing happens. To update the channel using its memory pointer, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 84 hex (command code).
- 7) Transmit CHNUM (user specified).
- 8) Get COMSTAT (should be 00 hex if successfully executed).
- 9) Check that DSR is reset.
- 10) Reset DTR.

### A.3.5 Transmit group execute

This command does not transmit a command code in the normal DCL format. Instead, it transmits a 00 hex device address code. This action simultaneously triggers all channels (in all devices) with active memory pointers to update their setting with the contents previously stored in the user memory specified (pointed to) by their individual memory pointers. DTR and DSR must remain in their inactive states. To execute transmit 00 hex.

This command allows groups or banks of channels to be updated simultaneously without having to transmit new data to each one. It is quite handy for quick changes.

### A.3.6 Lock device — 85 hex

This command disallows changes to the preprogrammed settings of the specified device. Attempts to write to any user memory or memory pointer will cause the device to return a 03 hex ("device locked") COMSTAT code. However, if the memory pointer is active, the channel can be updated with the GROUP EXECUTE or REPROGRAM DEVICE CHANNEL FROM POINTER commands.

If the device was locked when the power was removed, it should power up in a locked state. To lock a device, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 85 hex (command code).
- 7) Get COMSTAT (should be 00 hex if successfully executed).
- 8) Check that DSR is reset.
- 9) Reset DTR.

A 03 hex ("device locked") code could be returned for COMSTAT after an attempt to lock an already locked device.

### A.3.7 Unlock device — 86 hex

This command unlocks the specified device, thus allowing changes to its preprogrammed settings. To unlock a device, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 86 hex (command code).
- 7) Get COMSTAT (should be 00 hex if successfully executed).
- 8) Check that DSR is reset.
- 9) Reset DTR.

A 04 hex ("device not locked") code could be returned for COMSTAT after an attempt to unlock an already unlocked device.

### A.3.8 Mute output of device channel — 87 hex

This command informs the addressed device to mute the output of the specified channel. To mute a channel, the controller must:

- 1) Transmit ADDR (user specified).

- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 87 hex (command code).
- 7) Transmit CHNUM (user specified).
- 8) Get COMSTAT (should be 00 hex if successfully executed).
- 9) Check that DSR is reset.
- 10) Reset DTR.

A 05 hex ("channel(s) muted") code could be returned for COMSTAT after an attempt to mute an already muted output.

The ability of a device to mute the output of a channel is optional and may not be available on all devices. Each organization must decide for itself to include this ability.

### **A.3.9 Unmute output of device channel — 88 hex**

This command informs the addressed device to restore the output of the specified channel to its level prior to being muted. To restore the channel, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 88 hex (command code).
- 7) Transmit CHNUM (user specified).
- 8) Get COMSTAT (should be 00 hex if successfully executed).
- 9) Check that DSR is reset.
- 10) Reset DTR.

A 06 hex ("channel(s) not muted") code could be returned for COMSTAT after an attempt to unmute an already restored output.

### **A.3.10 Mute all outputs of device — 89 hex**

This command informs the addressed device to mute all of its outputs. To mute the channels, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 89 hex (command code).
- 7) Get COMSTAT (should be 00 hex if successfully executed).
- 8) Check that DSR is reset.
- 9) Reset DTR.

A 05 hex ("channel(s) muted") code could be returned for COMSTAT after an attempt to mute the already muted outputs.

### **A.3.11 Unmute all outputs of device — 8A hex**

This command informs the addressed device to restore all of its outputs to their levels prior to being muted. To restore the channels, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturers's ID code).
- 6) Transmit 8A hex (command code).
- 7) Get COMSTAT (should be 00 hex if successfully executed).
- 8) Check that DSR is reset.
- 9) Reset DTR.

A 06 hex ("channel(s) not muted") code could be returned for COMSTAT after an attempt to unmute the already restored outputs.

#### **A.3.12 Get OPSTAT code — 00 hex**

This command allows the system operator to query any device on line for its operational status, regardless of its compatibility with the controlling software program. At present only 00 hex is defined; it means that the device is functioning properly. Other codes will have to be defined by the participating organizations as more research is done. To execute the command, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 00 hex (command code).
- 7) Get OPSTAT (operational status).
- 8) Get COMSTAT (should be 00 hex if successfully executed).
- 9) Check that DSR is reset.
- 10) Reset DTR.

#### **A.3.13 Recall all data from device channel — 01 hex**

This command recalls all of the device-specific data and the memory pointer value from the specified channel and user memory. If the default memory is specified, the active, or live, settings are returned. To recall the data, the device controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 01 hex (command code).
- 7) Transmit CHNUM (user specified).
- 8) Transmit MEMNUM (user specified).
- 9) Get all device-specific data.
- 10) Get MPTR (memory pointer).
- 11) Get COMSTAT (should be 00 hex if successfully executed).
- 12) Check that DSR is reset.
- 13) Reset DTR.

#### **A.3.14 Get DT and ID codes — 02 hex**

This command tests for the presence of a device at the specific device address. If one is found, its DT and ID codes are returned. Otherwise, a time-out error occurs. To execute, the controller must:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Get DT (device-type code).
- 5) Get ID (manufacturer's ID code).
- 6) Transmit 02 hex (command code).
- 7) Get COMSTAT (should be 00 hex if successfully executed).
- 8) Check that DSR is reset.
- 9) Reset DTR.

All PA-422 devices should respond to this command. This allows the device controller to determine exactly what devices are on line. For example, a polling routine could sequentially step through each address location and store the meanings of the DT and ID codes, along with the device address, in a database file system.

#### **A.4 Control of nonintelligent devices**

Although the device control language was conceived for intelligent devices, this fact in itself places no practical limit on the ability of the interface to control nonintelligent devices. Nonintelligent devices such as switches or relays may be easily controlled using a format similar to the following:

- 1) Transmit ADDR (user specified).
- 2) Check that DSR is set.
- 3) Set DTR.
- 4) Transmit data (user specified).
- 5) Check that DSR is reset.
- 6) Reset DTR.

Other formats are possible for nonintelligent devices. Their precise definitions should be proposed by the manufacturers involved.

#### **A.5 Proposals for software extensions and enhancements**

A.6 contains the device control codes in use as of 1991 March 25. Application may be made by any affected party for a Participating Research Organization (PRO) ID code by mail to the publisher, Standards Secretariat, Audio Engineering Society, 60 East 42nd St., New York, NY 10165. Application for codes may also be made via data modem calls to the DCL database. Access phone numbers and log-in instructions are available from the AES office in your region.

A continuously updated list of device control codes will be available to any party by mail or from the DCL database. Charges are listed in the AES Standards catalog. Any PRO can propose changes and additions to the codes listed. Such proposals will be listed with their proposal dates as tentative codes. If no objection is received from any other PRO within six months, the code will become permanent.

Applications and objections will be processed according to the operating policy and the operating procedures of the AES Standards Committee.

## A.6 Summary of PA-422 system parameters and device control language

Codes followed by asterisks have been added in accordance with clause A.5.

### DT (device-type) codes

- 00 hex – 0F hex = reserved for programmable equalizers
  - 00 hex = dual-channel third-octave equalizer
  - 01 hex = dual-channel octave equalizer
  - 02 hex = four-channel octave equalizer
  - 03 hex = single-channel third-octave equalizer
  - 04 hex = single-channel sixth-octave equalizer
- 10 hex – 1F hex = reserved for programmable signal delays
  - 10 hex = single-output signal delay
  - 11 hex = dual-output signal delay
  - 12 hex = triple-output signal delay
  - 13 hex = four-output signal delay
- 20 hex – 2F hex = reserved for programmable gain/level controls
  - 20 hex = four-channel level control
  - 21 hex = eight-channel level control
- 30 hex – 3F hex = reserved for programmable signal routers
- 40 hex–4F hex = reserved for programmable crossovers\*
  - 40 hex = programmable crossover-processors\*
  - 41 hex = expansion unit for programmable crossover-processor\*
  - 42 hex = DSP expansion unit with analog output\*
- FF hex = reserved for expansion

### ID (participating research organizations' ID) codes

- 00 hex (0)Altec Lansing Corporation (USA)
- 01 hex (1)MicroAudio, Inc. (USA)
- 02 hex (2)Innovative Electronic Designs, Inc. (USA)
- 03 hex (3)VEGA (USA)
- 04 hex (4)Electro-Voice, Inc. (USA)
- 05 hex (5)University Sound Inc. (USA)
- 06 hex (6)Bouyer Division Systagemes (France)
- 07 hex (7)Audio/Digital, Inc. (USA)
- 08 hex (8)Rane Corporation (USA)
- 09 hex (9)IRP Professional Sound Products (USA)
- 0A hex (10)Orban (USA)
- 0B hex (11)FSR, Inc. (USA)
- 0C hex (12)White Instruments (USA)
- 0D hex (13)BSS Audio Ltd. (UK)
- 0E hex (14)SIEMENS Audio/Video (Austria)
- 0F hex (15)NEVE (UK)
- 10 hex (16)Dukane (USA)
- 11 hex (17)Rauland-Borg Corporation (USA)
- 12 hex (18)Intelix (USA)
- 13 hex (19)Dynacord (Germany)
- 14 hex (20)QSC Audio Products, Inc. (USA)
- 15 hex (21)Techron Industrial Products (USA)
- 16 hex (22)Protech Audio (USA)
- 17 hex (23)Zee Incorporated (USA)
- 18 hex (24)BIAMP Systems (USA)
- 19 hex (25)Peavey Architectural Acoustics (USA)

1A hex (26)JBL Professional (USA)  
 1B hex (27)Current Designs Corporation (USA)  
 1C hex (28)Eventide (USA)  
 1D hex (29)Ramko Research (USA)  
 1E hex (30)Clair Brothers Audio (USA)  
 1F hex (31)Pacific Recorders (USA)  
 20 hex (32)Optical Radiation Corporation (USA)  
 21 hex (33)TSI (USA)  
 22 hex (34)Stewart Electronics Corporation (USA)  
 23 hex (35)s.e.m.co. (Canada)  
 24 hex (36)Ashley Audio (USA)  
 25 hex (37)J. L. Cooper Electronics (USA)  
 26 hex (38)CAMCO Produktions- und Vertriebs-GmbH (Germany)  
 27 hex (39)Sennheiser electronic KG (Germany)  
 28 hex (40)DOD Electronics Corporation (USA)  
 29 hex (41)Oxmoor Corporation (USA)  
 2A hex (42)Crown International (USA)  
 2B hex (43)Quad Electroacoustics Limited (UK)  
 2C hex (44)T. C. Electronic of Denmark (Denmark)  
 2D hex (45)Aphex Systems (USA)  
 2E hex (46)Lectrosonics, Inc. (USA)  
 2F hex (47)SONOSAX S.A. (Switzerland)  
 30 hex (48) = RCF SPA\*  
 31 hex (49) = Electrosonic Ltd\*  
 32 hex (50) = NVision Inc\*  
 33 hex (51) = D&B Audiotechnik AG\*  
 34 hex (52) = Altair\*  
 35 hex (53) = Sytek Audio Systems\*

#### COMSTAT (communications status) codes

00 hex = no error  
 01 hex = bad data received  
 02 hex = bad command received  
 03 hex = device locked  
 04 hex = device not locked  
 05 hex = channel(s) muted  
 06 hex = channel(s) not muted  
 FF hex = reserved

#### OPSTAT (operational status) codes

00 hex = no error  
 FF hex = reserved

#### DCL (device control language) commands

00 hex (device address code) = transmit GROUP EXECUTE  
 00 hex = get OPSTAT code  
 01 hex = recall all data from device channel  
 02 hex = get DT and ID codes  
 7F hex = reserved  
 80 hex = reserved  
 81 hex = transmit all data to device channel  
 82 hex = reprogram device channel from memory  
 83 hex = transmit pointer to device channel  
 84 hex = reprogram device channel from pointer  
 85 hex = lock device  
 86 hex = unlock device

87 hex = mute output of device channel  
 88 hex = unmute output of device channel  
 89 hex = mute all outputs of device  
 8A hex = unmute all outputs of device  
 FF hex = reserved

**Device address range (DAR)**

00 hex = reserved for GROUP EXECUTE command  
 01 hex = FA hex (from 1 to 250 devices per port)  
 FB hex = FF hex = reserved

**MEMNUM (user memory number) codes**

(Up to 256 user memories per device channel allowed)  
 00 hex = user memory 0 = default memory  
 01 hex = user memory 1  
 02 hex = user memory 2

**MPTR (memory pointer) codes**

00 hex = memory pointer OFF (could point to default memory)  
 01 hex = memory pointer points to user memory 1  
 02 hex = memory pointer points to user memory 2

**CHNUM (channel number) codes**

(Up to 255 channels per device allowed)  
 0 = reserved  
 1 = device channel 1  
 2 = device channel 2

**Baud rate**

19 200 baud (or bit/s)

**Character frame bits**

1 start bit  
 8 data bits  
 1 parity bit (even parity)  
 1 stop bit

**Recommended cable**

Belden, part no. 9681 or equivalent

**Maximum cable length**

1.2 km (4000 ft)

**Standard connector types**

Male nine-pin D-subminiature (input)  
 Female nine-pin D-subminiature (output)

**Typical device time-out period**

250 ms (minimum)