RS-232 Protocol for NAD Products v1.03   May 5, 2003

Introduction

The purpose of this document is to define the protocol used to communicate to NAD products that have an RS-232 port. This protocol allows NAD products with an RS-232 port to be controlled by a PC or any other device which has an RS-232 port.

RS-232 Specifications

The NAD product should use a standard DB-9 female connector with the following pinout:

Pin	Function
2	Transmit Data
3	Receive Data
5	Signal Ground

This pinout allows the NAD product to be connected to a PC with a standard straight throw serial cable.

All communication should be done at a rate of 9600 bps with 8 data bits, 1 stop bit and no parity bits. No flow control should be performed.

Data Format

All commands sent to the device and responses sent back have the following basic format:

<start>	Command	Data	<checksum>	8-bit Checksum

Note: The items denoted with "<" and ">" are control characters. The format of the control characters and how to send the data is discussed later in this section.

<start>   1 byte control character which starts every packet of data.

Command   1 byte command which represents the operation to be performed or the type of response being returned.

Data   Variable length data. The length of this field depends on the command. If the data represents a multibyte number, it should be sent least significant byte first.

<checksum>   1 byte control character which denotes the end of the data stream and the beginning of the checksum.

8-bit Checksum   This checksum is the inverse sum of the Command and Data bytes of the packet. It is used to verify the validity of the data packet.

Any data received which does not follow this format should be ignored.

Since the control characters are simply sent as single byte values, there needs to be a way to differentiate between a control character and regular data. A standard way of doing this, which this protocol uses, is to use a "flag" character (which is a control character) to encode the regular data which have the same value as a control character.

Before describing how the data is encoded, the control characters will be listed so they can be used in examples. The following table shows all the control characters and their corresponding values:

Control Character	Value
<start>	1
<checksum>	2
Reserved for Future Expansion	0, 3 – 19
<flag>	94

Control characters 0 and 4 through 19 have been reserved for future expansion. For example, standard XON/XOFF flow control uses characters 17 and 19.

Any data values which are to be sent that correspond to a control character must be encoded with the following method:

1. Send the flag control character.
2. Send the data value bitwise ORed with the value 64.

All other data values which do not correspond to a control character can be sent as is.

Some examples:

Data Value to Send	Encoded byte sequence
3	94, 67
5	94, 69
20	20
94	94, 94

On the receiving side, anytime a value of 94 is received the following character must be decoded. The character is decoded using the following algorithm:

1. If the value received is 94, then the actual value is 94.
2. Else, the received value should be bitwise ANDed with the value 191.

For example, if the sequence 94, 69 is received, it should be decoded to the value 5 since 69 AND 191 equals 5.

Checksum Calculations

The checksum is simply the inverse (ones complement) of the sum of the Command and Data bytes of the packet. For example, if the command byte was 20, and the data byte was 20, then the checksum would be 215. (20 + 20 = 40, complement of 40 = 215). When the data values are encoded because they overlap with a control character, the actual value should be used in the sum and not the encoded value.

Command Set

The command set is documented in several tables in a separate document. It describes what all the commands are and which products support which commands.

Additions to the command set will be done on an ongoing basis as new functions are needed for new products.

Example 1

This example will describe exactly what is sent when the Identification of the unit is Queried. This is done by sending the Query Function command which is command 20. Since it is the Identification that is being queried, the data value 20 is sent with the command.

The above is done by sending the following byte sequence:

<start>	Command	Data	<checksum>	Checksum
1	20	20	2	215

If the identification string of the unit was "T772" then the response to the above command would be:

<start>	Command	Data	<checksum>	Checksum
1	20	20,84,55,55,50	2	227

Example 2

This example will describe how to set the volume to a level of 3 dB. This is done by sending the Set Function command which is command 21. The data sent would be the value 23 to set the volume, followed by 3 which is the value for the volume. Note that since the value 3 is a reserved value for a control character, it must be encoded.

The above is done by sending the following byte sequence:

<start>	Command	Data	<checksum>	Checksum
1	21	23, 94, 67	2	208

If the volume is successfully set to +3 dB, then the response sent back would be:

<start>	Command	Data	<checksum>	Checksum
1	20	23, 94, 67	2	209

In the above examples, note that the non-encoded value of 3 for the data is used for the calculation of the checksum.