Direct Serial Control of the XBUS

TDT Tech Note #123

Overview

It is sometimes useful to send commands directly to System II equipment without using the drivers provided by TDT. Users usually wish to do this because TDT drivers are not supported by their programming language or compiler. Whatever the reason, this process is quite straightforward.

The example below illustrates how to program a PA4, since this is one of the most popular devices to control in a standalone configuration.

Preparation

You need several pieces of information before programming XBUS equipment directly:

1. XLN. The XLN is the XBUS location number. It is a value from 4 to 127. The first rack 1 holds locations 4 to 7; the second, locations 8 to 11; and so forth. The locations are numbered from left to right. A device, if in the second position from the left of the first (or only) rack, will have XLN = 5.

2. Command Codes. The command codes for each device are shown on the device tear sheets and in the XBDRV Software Reference.

Command Forms

There are two command forms: the standard form and the short form. Each form begins with the XLN of the device being referenced.

The **short form** is used for sending single-byte commands (e.g., PA4_MUTE). The form is:

[XLN] [command code & 0x3F]

Note that there is no checksum and that command codes cannot exceed 31.

For commands that require additional data to be sent, the **standard form** must be used.

The standard form is constructed as follows:

[XLN] [0x40|n] [b1] [b2] ... [b(n-1)] [cs]

where

XLN	is the XBus location number, as described above
n	is the number of bytes to send (2 to 63)
b1...b(n-1)	are the bytes of data
and cs	is the LSB of the simple sum of b1...b(n-1).

Controlling PA4 Attenuation

The sequence of bytes to set PA4 attenuation to 99.9 would be following:

0x05	XLN: rack 1, position 2
0x44	send 4 more bytes (0x40|n)
0x20	PA4_ATT (PA4 command to set attenuation level)
0x03	high byte of atten*10
0xE7	low byte of atten*10
0x0A	data checksum (LSB of 20+03+E7=0x010A)

Other Communication Details

After a command is sent to an XBUS device, the device responds back with SLAVE_ACK (0xC3). You only have to wait for this command if you with to receive verification that your hardware is working correctly.

XBUS communications always pass a predetermined number of bytes in each packet, and don't involve large volumes of data, so they do not require any flow control.

Making Your Own Drivers

You can easily make your own low-level communications program to talk with XBUS equipment in DOS. Take the COM port initialization routine from XBCOMINI.C (in the TDT\DRIVERS\ XBDRV\SAMPLES folder) and the iosend() and iorec() routines from XBDRV.C (which just write to the COM port control registers).

Initialize your TDT equipment per the user's manual (autobaud your rack with XBCOMINI, then "ping" it with XBSYSID) so that your PA4 shows up when you run XBIDCHCK. In your program, initialize the ports to run at 38400,N,8,1 and you should be ready to communicate.

Further Reading

For additional information and background on the information presented in this document, consult the section in the "XBDRV Software Reference" titled "XBDRV Nuts & Bolts", the XBUS Tear Sheet in the back of your System II manual, and the XBUS driver source code (XBDRV.C).