System II Device Driver & Controller

System II 32-bit Device Driver & Controller User's Guide – Version 1.0
01 January 1999

Printed in U.S.A.

Contents

Preface

Software Philosophy

Chapter 1 Introduction

What is the S232 Device Driver? 1-1

Hardware Support 1-1

Who Can Use the S232 Device Driver & Controller? 1-1

Before You Begin 1-1

What You Need 1-1

Installation of System II 32-bit Device Driver & Controller 1-2

Requirements 1-2

Installation 1-2

Files 1-2

Chapter 2 System II 32-bit Device Driver & Controller

Introduction to the System II 32-Bit Architecture 2-1

What is the WinRT Generic Device Driver? 2-1

What is a System II 32-bit DLL? 2-2

What is the System II 32-bit Controller? 2-2

How Do Multiple Applications Share System II Hardware Resources? 2-2

The Lock/Unlock Protocol 2-3

Primary and Secondary Applications 2-3

Chapter 3 The S232 Controller

Starting the Controller 3-1

Restarting the Application 3-1

Using the S232 Controller From The Taskbar 3-1

Viewing and Hiding the Application 3-1

Resetting Hardware 3-1

Getting to Know the S232 Controller Dialog Window 3-2

The Menu Bar 3-2

The State Area 3-2

The Performance Area 3-2

Setting Controller Configurations 3-3

AP2 I/O Port 3-3

XBUS System ID 3-5

Testing Hardware 3-5

AP2 3-5

D/A A/D 3-6

XBUS Map 3-7

Working With Buffers 3-7

Hardware Status 3-8

Chapter 4 Advanced Topics

Working With Single Card Systems 4-1

Working With Dual Card Systems 4-1

What Are Logical Devices? 4-1

Connecting to Logical Devices 4-1

Setting the Command Line Argument 4-2

A New Look For A Second Card 4-2

Working With the OP2 Optical Controller 4-2

Connecting to the OP2 4-2

Setting the Command Line Argument 4-3

What Services are not Available? 4-3

AP2 Port Settings 4-3

Preface

Software Philosophy

TDT's philosophy on software development is simple. Design a comprehensive software platform that allows the customer to work at a level with which they are most productive. This software solution represents the continuing efforts of TDT to provide you with the latest tools for using System II.

System II includes a extensive high-level Applications Programming Interface (API). With minimal programming, TDT's API enables scientists to produce high-quality signal processing applications tailored to meet their specific needs. In complementary fashion, a suite of "turn-key" applications exist that have a minimal learning curve and facilitate signal generation, presentation, and analysis. The continuing evolution of System II productivity tools provides a broad range of opportunity for the scientist, researcher, or engineer.

With this package it is now possible to use TDT on both Windows 95/98 and Windows NT platforms. Furthermore, the new 32-bit platform allows for multiple applications to access the AP2 Array Processor or XBUS hardware modules and works with the family of SigGen Solutions.

Organization of the Guide

The *System II 32-bit Device Driver & Controller User's Guide* presents all the information necessary to become familiar with the new 32-bit Device Driver and utilize the S232 Controller. This document guides you through the process of incorporating the S232 Controller as a supplement to your current portfolio of System II software solutions. This document also serves as a general reference tool. The information provided includes:

- General Information about the System II 32-bit Device Driver
- Basic and advanced information about the 32-bit Controller.

General Information

General information includes an introduction to the 32-bit System II Device Driver. General information is presented in:

- *Chapter 1* Introduction
- *Chapter 2* System II 32-bit Device Driver

The System II S232 Controller

The aspects of using the S232 Controller are presented below, along with their associated chapter.

- *Chapter 3* The S232 Controller
- *Chapter 4* Advanced Topics

  Including the following:
  Working with Single Card Systems
  Working with Dual Card Systems
  Working with the OP2 Optical Controller

Chapter 1 Introduction

What is the S232 Device Driver?

The S232 Device Driver is a System II software solution that provides true 32-bit Windows performance. It incorporates dynamic link libraries (DLL) as a standard interface tool and works through a generic Windows device driver. In addition to offering better system compatibility and Windows NT support, the S232 Device Driver allows for multiple applications to simultaneously gain access to System II hardware.

Hardware Support

The S232 Device Driver supports all TDT System II programmable instrumentation, including any combination of converters, programmable attenuators, waveform generators, cosine switches and programmable filters.

Who Can Use the S232 Device Driver & Controller?

Anyone with a PC, Windows 95/98 or Windows NT, and TDT System II Hardware. Users include:

- Any previous System II users
- Auditory Scientists
- Physiologists
- Signal Processing Instructors
- Speech Scientist

Before You Begin

What You Need

See your Microsoft Windows documentation.

- Windows fundamentals

  You should be comfortable with Windows basics: starting Windows; using the mouse; manipulating windows; opening, closing, and saving files.

See the System II manual.

- Basic System II Concepts

  You should recognize System II concepts including: buffer & DAMA management, stack operations, module commands, etc.

Installation of System II 32-bit Device Driver & Controller

Requirements

In order to use the S232 Device Driver & Controller, you must have the following:

- Microsoft Windows 95/98 or Windows NT Operating System
- A monitor with at least VGA resolution graphics. Super VGA (1024 x 768) resolution graphics highly recommended
- TDT's AP2 Array Processor or OP2 Optical Controller

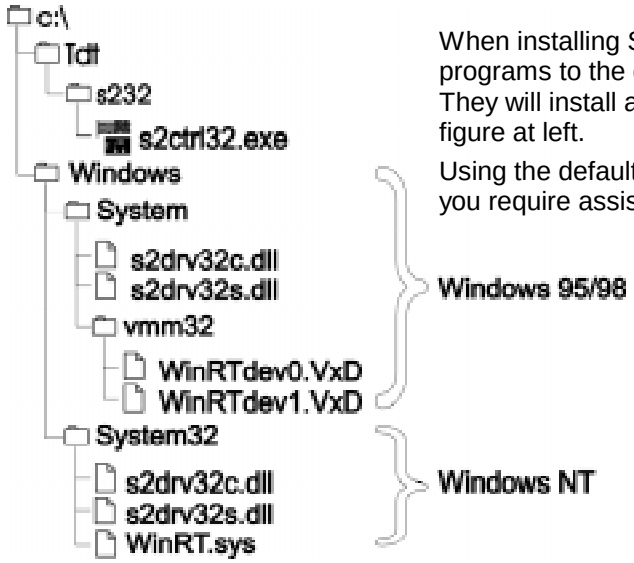
Installation

1. Make sure your TDT AP2 or OP2 card is installed properly.
2. Insert the S232 Device Driver & Controller diskette into drive A:.
3. Run setup.exe to start the installation program.

At the end of the installation you will have to reboot your computer for the installation to be completed.

NOTE: The S232 Device Driver & Controller is shipped with default settings of AP2 location, APb, and a device IRQ number, 15. It may be necessary to adjust these settings for your particular configuration. If the "Configure AP2 Device Driver" dialog window appears, proceed to the Chapter 3 section titled "Setting Controller Configuration."

Files



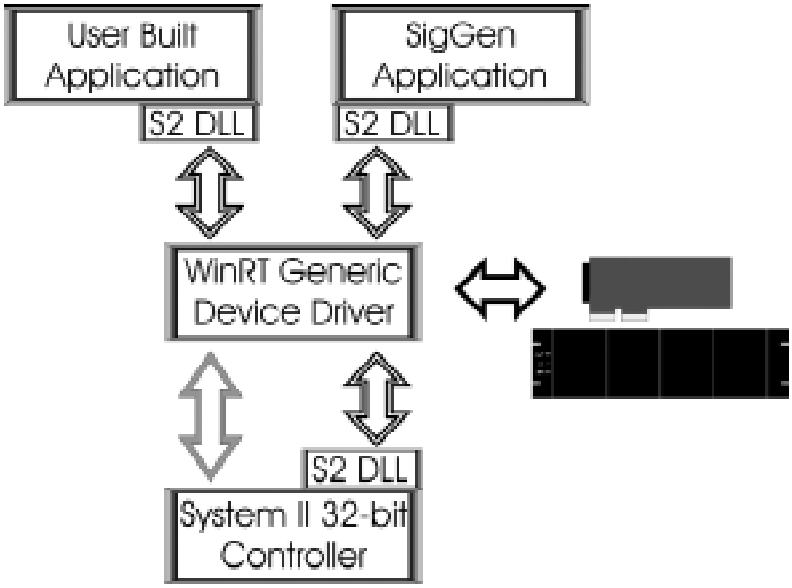
When installing System II software, it is recommend that you install the programs to the default folders suggested by the installation program. They will install all files to the standard TDT file structure as shown in the figure at left.

Using the default file structure will make it easier for TDT to support you if you require assistance.

Chapter 2 System II 32-bit Device Driver & Controller

Introduction to the System II 32-Bit Architecture

The System II 32-bit architecture is built within the Windows framework and utilizes the Windows DLL as the standard interface. This presents a number of benefits to the system, but it also implies that a new set of services will need to be provided. Specifically, the Windows operating system requires that every piece of hardware must have its own *device driver*.



What is the WinRT Generic Device Driver?

All hardware access is controlled by the operating system and performed within the kernel. The kernel is the layer of software that is closest to the hardware and provides basic operating system services. If a program attempts to communicate directly with hardware, the operating system intercedes by trapping the call and denying the access to the requesting application. All hardware is shipped with device drivers that are implemented at the kernel level and recognized by the operating system.

Device drivers can be quite involved depending on the level of services required by the hardware. Furthermore, their implementations are fundamentally different for Windows 95/98 and Windows NT. Rather than develop what already exists, a generic device driver, WinRT, was incorporated to provide the few services required by the AP2. WinRT arbitrates control of and talks directly to System II hardware for requesting applications.

What is a System II 32-bit DLL?

A DLL is a file that contains a library of routines. The functions of a DLL provide a collection of services through a well defined application programming interface (API). When applications use a DLL, they link a copy of the library at runtime and use those functions as though they were originally compiled with the executable.

There are a number of advantages to using DLLs.

- Multiple applications can share a single DLL simultaneously.
- Software upgrades become simpler. The applications that use DLLs do not need to be recompiled if the interface is standardized.
- Applications that use DLLs can be written in variety of different programming languages.

The System II 32-bit DLL, shown as S2 DLL in the preceding diagram, serves as the API for the System II hardware. It provides services to the calling application and works through the WinRT device driver for their implementation. All applications must use the S2 DLL to communicate with System II hardware.

What is the System II 32-bit Controller?

The System II 32-bit Controller is an application. As such, it uses the System II DLL as its API to the hardware, as is shown in the preceding diagram. It also provides a collection of services that are not shared in the System II DLL and are not provided to other applications. The S232 Controller monitors and provides a means to override application's control of the System II Device Driver and ultimately System II hardware. As a result it requires a higher level of access than that of normal user applications. The S232 Controller will be discussed in detail the next two chapters.

How Do Multiple Applications Share System II Hardware Resources?

Previously, only one System II application could operate on one AP2 card at a time. This meant that the user had to make sure that no other System II applications were active before starting a program.

A new hardware arbitration feature has been added to the S232 Device Driver to allow for multiple applications to simultaneously access hardware. This evolution of System II programmability is hinged on the concept of hardware access. All programs share System II resources. The Device Driver arbitrates among the applications by implementing a *Lock-Unlock Protocol*.

The Lock-Unlock Protocol

Before sending commands to the AP2 card and/or XBUS, the application must request and obtain a hardware *lock.* Hardware locks are available for the AP2 and XBUS. Possession of the resource lock gives the application control over the AP2 and/or XBUS. While an application has the resource lock no other applications can access hardware resources until the resources are *unlocked*. The *lock-unlock sessions* form the basic programming blocks of System II applications. During a lock-unlock session, applications can work without interruption.

Users of System II SigGen solutions do not have the ability to lock/unlock hardware resources since it is handled by the application itself. This information on the lock-unlock protocol is included so that the user has an appreciation for the S232 Controller hardware states. Programming with the Lock/Unlock Protocol is described in more detail in Chapter 2 of "The S232 Software Interface Suite User's Guide."

Primary and Secondary Applications

The concept of multitasking introduces the question of hardware ownership. Applications can access System II hardware resources during their lock-unlock sessions. But what happens to those hardware resources between sessions depends on an application's classification. An application may be classified as either a *primary* or *secondary.* The classification implies certain ownership privileges of AP2 resources.

There are two main features of a primary application:

- Only one primary application can be run at a time.
- Primary applications assume that their stack and DAMA buffers will be found as they left them.

The features of secondary applications are:

- Multiple secondary applications may be run at the same time.
- Secondary applications can not assume any stack or DAMA buffer will be retained between lock-unlock sessions.

BioSig is an example of a primary application, and SigGen is an example of a secondary application. Both applications can be active simultaneously with SigGen performing the stimulus generation in support of BioSig. This information is provided so that the user has an appreciation of the S232 Controller hardware states. Programming primary and secondary applications is described in more detail in Chapter 2 of "The S232 Software Interface Suite User's Guide."

Chapter 3 The S232 Controller

The S232 Controller loads the AP2 software onboard, monitors hardware, views resources, and configures the System II hardware. The S232 Controller must be active before any System II application can access hardware. The application is launched immediately after Windows has finished loading. The S232 Controller icon appears in the notification tray of the Windows taskbar as shown on the left. The notification tray is found in the lower-right corner of the computer screen.

Starting the Controller

The installation process places a shortcut to the S232 Controller application in the Start-Up folder. Windows should start the application each time the computer is turned on.

Restarting the Application

The program should only be restarted if the S232 Controller icon is absent from the notification tray of the taskbar.

1. Click the Start button.
2. Click Programs
3. Click Start Up.
4. Choose the S232 Controller by selecting S2ctrl32.

Tips

- You can also restart the application by double-clicking the S232 Controller icon from within Windows Explorer.

Using the S232 Controller From The Taskbar

Viewing and Hiding the Application

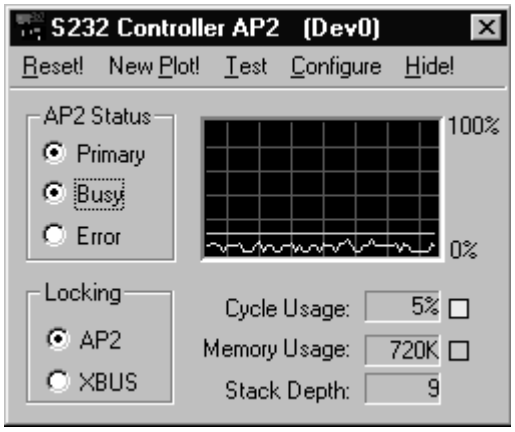
The S232 Controller is hidden after the application is launched. The S232 Controller icon should be present in the notification area of the taskbar. *To view the S232 Controller windows, move the mouse over the S232 Controller icon and double-click the left mouse button.* Subsequent double-clicking will toggle the viewing state of the application.

Resetting Hardware

It is sometimes necessary to reset the System II hardware to regain its control. The S232 Controller icon can provide this service. *To force a hardware reset, move the mouse over the S232 Controller icon and double-click the right mouse button.*

Getting to Know the S232 Controller Dialog Window

All of the S232 Controller functionality is present in the simple dialog window shown below. The compact design allows for a quick synopsis of System II resources. The S232 Controller Title bar contains the name of the logical device under control and an application kill button. If the OP2 card is used rather than an AP2 card, the reader is referred to “Chapter 4 Advanced Topics” for more details.



The application window is divided into three logical areas:

- The Menu Bar Area.
- The System II Hardware State Area.
- The Performance Area.

The Menu Bar

The menu bar area provides access to a set of System II procedures for resetting, monitoring, testing, and configuring hardware. It also allows the S232 Controller to change its viewing state to hidden.

The State Area

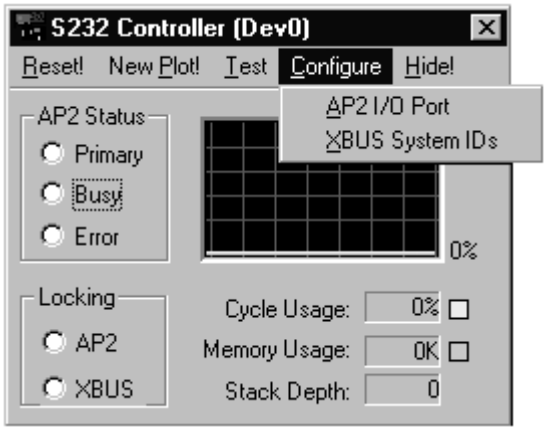
The state area provides a visual means for checking the System II state. Indicators include the hardware locks, the primary application, AP2 processing, and errors states.

The Performance Area

The performance area provides both a textual and graphical display of the AP2 hardware resources in use.

Setting Controller Configurations

The S232 Controller has a series of procedures to configure the System II hardware. The configuration menu is shown below.

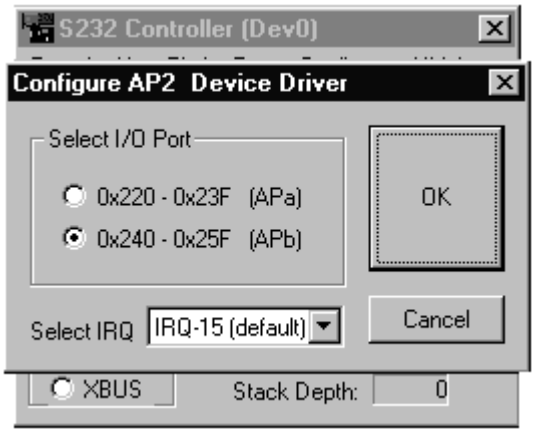


AP2 I/O Port

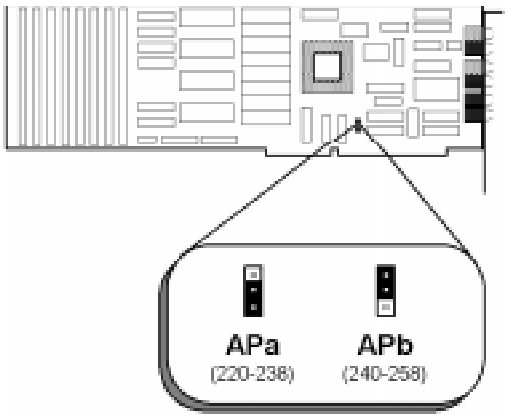
The S232 Controller application requires a set of Windows hardware settings for the device driver. The default set is I/O Port: APb and IRQ: 15. These settings were chosen because the standard I/O Port settings for soundcards is 0x220, the I/O space for APa. The reader is advised to use the default settings unless there is a known conflict.

To configure the AP2:

1. Click Configure
2. Click AP2



The majority of AP2 configuration changes will be selecting APa over APb. This software I/O Port setting must reflect the state of the APa/APb jumper on the AP2 card. The I/O jumper settings are shown in the following diagram. Changing the S232 I/O setting without matching the jumper setting will result in unpredictable behavior. See the System II installation manual for complete information regarding the changing of the APa/APb jumper.



The S232 Controller IRQ setting is required by the device driver, but not actually used by the AP2 card. However, some Windows System resources may not allow for IRQ sharing regardless of its implementation method. Before changing S2332 Controller settings, it must be determined what Windows resources are available.

To view the Windows 95/98 System resources,

1. Click the Start Menu
2. Click Settings
3. Click Control Panel
4. Double-click the System icon.
5. Choose the Device Manager tab
6. Click Computer, and then click Properties.

To view the Windows NT System resources.

1. Click the Start Menu
2. Click Programs
3. Click Administration Tools
4. Select Windows NT Diagnostic
5. Choose the Resources Tab

To view the particular Windows system resources to be changed, select either Input/output (I/O) or Interrupt request (IRQ). Once it is determined which Windows resources are available, make the specific changes and close all the windows.

The computer must be restarted before the changes can take effect.

XBUS System ID

This configuration procedure will be familiar to previous System II users. It should be used when a System II rack has been added to or removed from the system. It replaces the XBCOMINI and XBSYSID executables that Autobaud and ID System II racks.

It is not necessary to run this utility if *device modules* are repositioned, added, or removed in the XBUS. It should only be used when *device racks* are repositioned, added, or removed. The dialog window contains all the instructions needed to successfully configure System II hardware.

To configure the XBUS:

1. Click configure.
2. Click XBUS System ID

Testing Hardware

The S232 Controller comes equipped with mechanisms to test specific areas of the System II hardware from Windows. They are helpful in determining if errors are present in the hardware system or in software. The test menu is shown below.

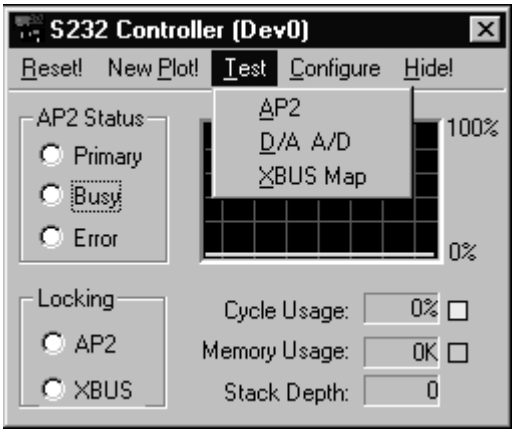
AP2

The AP2 is the centerpiece of the System II hardware. It provides a number of services including computational procedures, data management, data handling, and communications/controls of XBUS modules.

This test was designed to validate the basic DSP functionality. A window will contain a message indicating if the AP2 passed or failed the test.

To test the AP2:

1. Click the Test Menu
2. Select AP2

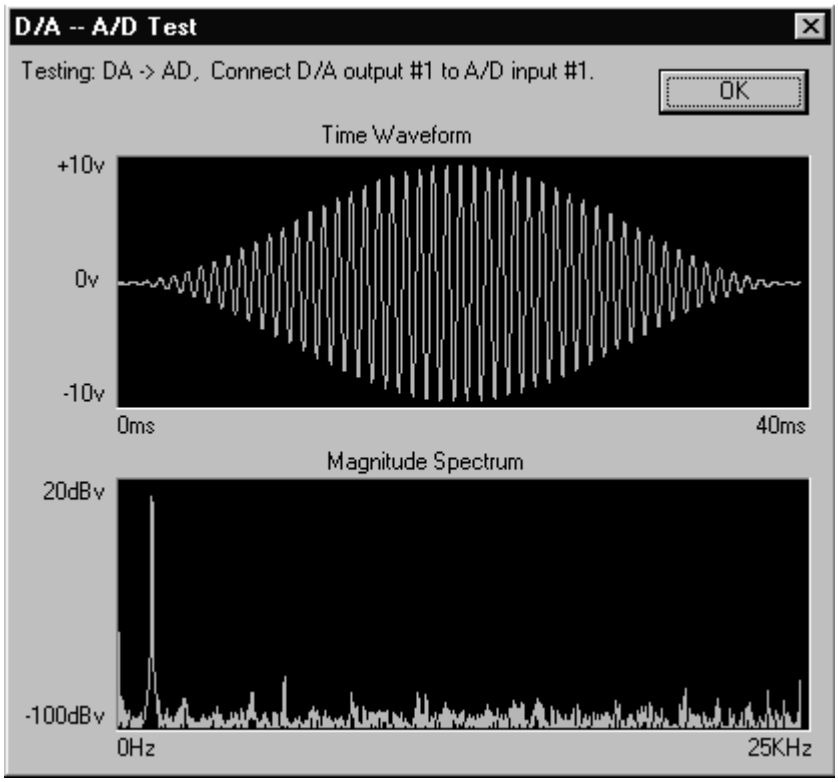


D/A A/D

This procedure determines which System II analog interface modules are present. The test is conducted with the recognized converter modules and a TG6 is present. The procedure fires all six channels of a TG6 to trigger the data converters. The D/A module will transmit the buffer shown in the window below. This window displays the data received by the A/D module in both the time and frequency domains.

To test the data converters:

1. If both modules are present, connect the D/A channel 1 to the A/D channel 1. Else connect the relevant data converter to an appropriate input or output channel on a test device, e.g. oscilloscope or function generator.
2. Click the Test Menu
3. Select D/A A/D



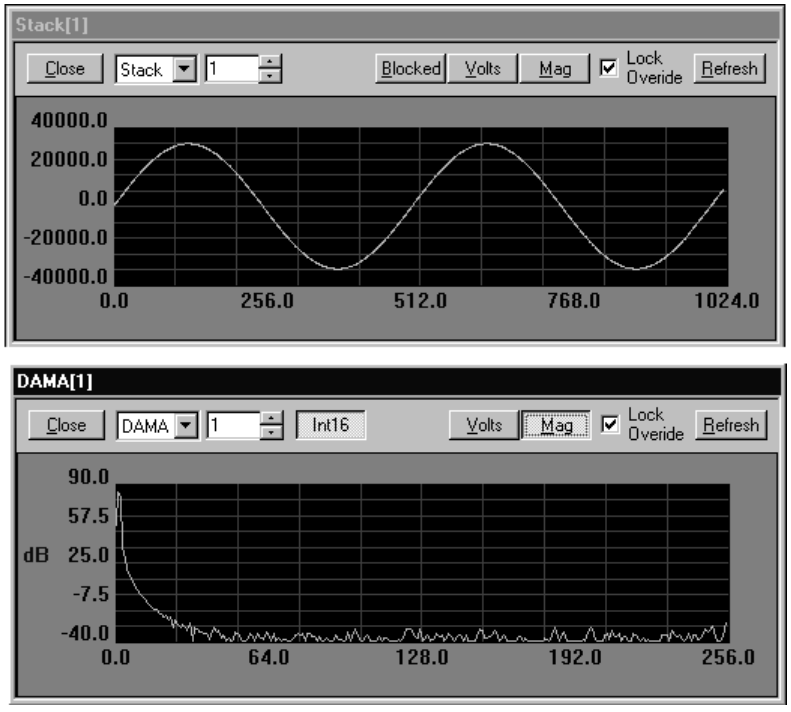
XBUS Map

The XBUS is the communications backbone of the System II hardware. It replaces the XBIDCHCK executable. This test displays all of the programmable modules that are currently recognized by the System II operating system. Modules such as FT6, HB6, SM3 etc are not shown since they are not programmable. If a programmable device does not appear in the window, select Reset! and try again.

1. Click the Test Menu
2. Select XBUS Map!

Working With Buffers

The New Plot! menu item provides a powerful feature for System II users -- the ability to view any buffer on the stack or in DAMA. A maximum of ten buffers can be displayed simultaneously. While its strength remains a debugging tool, its benefits are available to the Application User. Shown below are two buffers that contain the same data. The upper buffer resides on top of the stack and is viewed in the time domain. The lower buffer is in DAMA and is viewed in the frequency domain.



When a buffer window is opened the user must select the memory area and buffer index to be viewed. By default, the windows contain no buffers. If the specified buffer does not exist, the plotting area will be grayed out. By default, the plots are only updateable when the AP2 is not locked.

The *Lock Override* box is a method to allow for the buffer windows to be updated when the AP2 lock is engaged. Care should be taken when using this feature as the potential of AP2 failure is present. Specifically, the Lock Override should not be set while the buffer is in high use by an application, e.g. by SigGen. By leaving the Lock Override button unselected, the user is guaranteed to access the buffer while it is not in use.

A number of features are available to view the data. The format of the data includes either fixed-point or floating-point, *Int16 button.* The domain includes either time or frequency, *Mag button*. Additionally, the data can also be viewed with respect to the external analog voltage, *Volts button*.

The plot window is self scaling and initially shows the entire buffer. By using the *Blocked button*, the plot window can be made to display only the blocked portion of the buffer. Moreover, the user can "zoom-in" on a portion of the data buffer by click-and-holding the left mouse button, highlighting a section of the buffer, and releasing the button. Clicking the mouse in either half of the "zoomed" buffer will slide the time or frequency scale in that direction. By right-clicking the mouse, the entire buffer is once again displayed.

Hardware Status

The S232 Controller Status Area provides several visual hardware state indicators for the AP2 and XBUS. Most of the information displayed is basic in nature, but can be helpful to the application user. They can be critical to the application programmer.

The indicators reflect the current state of the System II 32-bit platform. They provide feedback regarding any primary application currently running, the hardware resource locks, and the status of the AP2 card.

They also can offer insight into how much of the AP2 resources are currently in use. Consider an application which reads data from a file and places that data in memory. Should the data file be unusually large, there could be little memory resources left for building stimuli or signal buffers. As a result the application could become unstable or simply fail to function correctly. The hardware status areas confers this information quickly in a graphical format that is easy to understand and interpret.

Chapter 4 Advanced Topics

Working With Single Card Systems

By default, the S232 Controller comes ready to work with an AP2 single card system.

Working With Dual Card Systems

System II can support up to two AP2 cards simultaneously. It can be advantageous to have dual cards as the computational resources for that system can be effectively doubled. Each card is a *logical device*.

What Are Logical Devices?

Logical devices can be thought of as physical hardware, i.e. AP2 cards. An AP2 card can exist as one of two varieties, APa or APb. If only one AP2 card is present in a given system, that card is by default Logical Device 0 independent of the memory area in which the card resides. If it is a dual card system, either card, APa or APb, can be defined as Logical Device 0. By default, the remaining card is Logical Device 1.

Connecting to Logical Devices

Each logical device, and hence AP2 card, requires an independent device driver. For Windows 95/98, the device drivers are WRTdev0.VxD and WRTdev1.VxD. For Windows NT, the device driver is WinRT.sys. Each have their own settings stored in the Windows Registry which are loaded at the time Windows is started. In addition to a device driver, each AP2 card must be associated with an S232 Controller which provides a core set of services. Those services include transferring the AP2 operating system to the AP2 card.

In Windows, applications can be launched more than once. Once launched, each copy of the program functions “independently” of the others. For example, the Windows provided program Notepad can be executed several times with each copy opening and editing its own document. In a similar fashion, dual copies of the S232 Controller program can be run simultaneously with each copy connected to a different device driver/AP2 card pair.

The S232 Controller requires a command line argument to connect to a specific device driver/AP2 card pair. A command line argument follows the executables name and includes the switch identifier, “/”. When the S232 Controller program is launched, the command line argument tells the application which logical device to use, e.g. /1 is logical device 1. The absence of a command line arguments implies logical device 0.

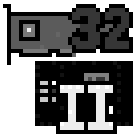
Setting the Command Line Argument

To add a command line argument to the S232 Controller:

1. Find the S232 Controller shortcut icon, *not the program itself*, by using the Windows Explorer.
2. Select the S232 Controller shortcut.
3. Click the right mouse button.
4. Select Properties.
5. Find the Target Window that contains the path to the S232 Controller application: the default is C:\Tdt\S232\S2ctrl32.exe.
6. Add the required command-line argument, e.g. /1. The entire line now appears in the default case as: C:\Tdt\S232\S2ctrl32.exe /1.
7. Click OK.

A New Look For A Second Card

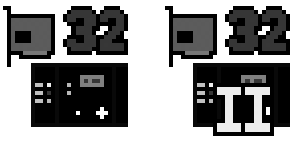
This application *is* the S232 Controller and contains all the functionality that has been previously described. It can perform diagnostic tests, communicate with the System II modules on its own XBUS, examine buffers either on the stack or in DAMA, and work with System II 32-bit applications.



The only difference is what is present on the title bar and in the notification tray of the taskbar. The title bar contains the name of the logical device that it controls, e.g. S232 Controller (Dev1). The S232 Controller that communicates with logical device 1 includes a yellow Roman numeral as shown to the left.

This S232 Controller icon functions exactly as previously described.

Working With the OP2 Optical Controller



The OP2 Optical Controller provides all the XBUS communication ability without the AP2 DSP functionality. The S232 Controller can work with the OP2 and can provide some of the resources described in Chapter 3. The OP2 icon that is present on the S232 Controller title bar and in the notification tray of the taskbar is show to the left for logical device 0 and logical device 1.

Connecting to the OP2

To connect the S232 Controller to the OP2, a command line argument must be supplied, /OP2. Without this argument, the S232 Controller would assume that all resources are present and would begin to report a series of errors with regard to the AP2 DSP functionality.

Setting the Command Line Argument

To add a command line argument to the S232 Controller:

1. Find the S232 Controller shortcut icon, *not the program itself*, by using the Windows Explorer.
2. Select the S232 Controller shortcut.
3. Click the right mouse button.
4. Select Properties.
5. Find the Target Window that contains the path to the S232 Controller application: the default is C:\Tdt\S232\S2ctrl32.exe.
6. Add the required command-line argument, /OP2. The entire line now appears as in the default case: C:\Tdt\S232\S2ctrl32.exe /OP2.
7. Click OK.

What Services are not Available?

Since the OP2 does not have all of the AP2 resources, several of the aspects of the S232 Controller become defunct. These services are grayed-out and thus not selectable. Those features are listed below.

- All plotting resources.
- AP2 test
- D/A A/D test
- AP2 Status items
- AP2 Resource Area

AP2 Port Settings

Because there may still be Windows System conflicts with the I/O address and IRQ settings, access to this menu item is still provided. The values can be changed to resolve any conflicts.