DAP 3400a Manual

Installation Guide,
Connector Reference,
and DAPL 2000 Reference

Version 1.10

Microstar Laboratories, Inc.

Microstar Laboratories, Inc.
2265 116 Avenue N.E.
Bellevue, WA 98004
Tel: (425) 453-2345
Fax: (425) 453-3199
http:// www.mstarlabs.com

Part Number MSHM006-02

Contents

1. Introduction

The Data Acquisition Processor from Microstar Laboratories is a complete data acquisition system that occupies one expansion slot in a PC. Data Acquisition Processors are suitable for a wide range of applications in laboratory and industrial data acquisition and control.

The DAP 3200a is high-performance Data Acquisition Processor specialized for fast analog sampling.

Features of the DAP 3200a:

- i486DX4 CPU
- 8 Megabytes DRAM
- 50 ns TIME resolution
- 800K samples per second per channel
- 3.2 million samples per second aggregate
- inherent simultaneous sampling of 4 input pins
- no output and no digital port
- ±2.5 volt, ±5 volt ranges

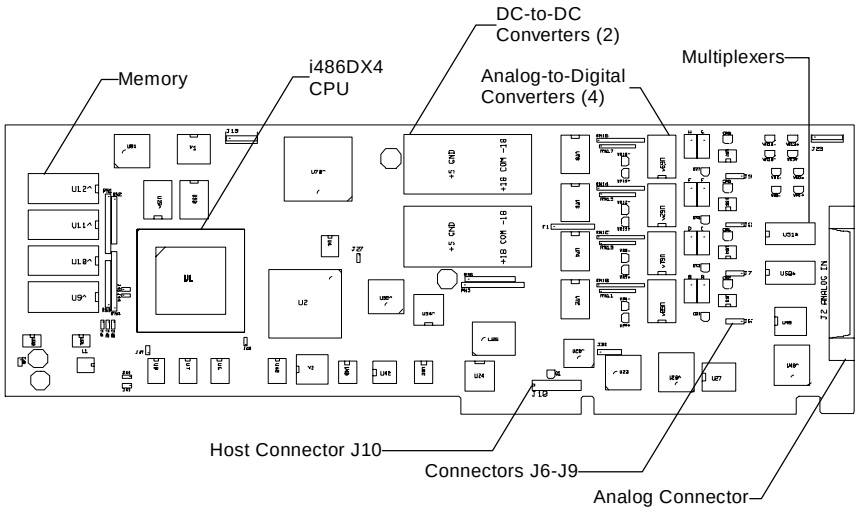


Figure 1. DAP 3400a Components

Overview of the DAP 3400a

The DAP 3400a uses four analog-to-digital converters. Each analog-to-digital converter is connected to four single-ended input pins. When the input pins are sampled, sampling occurs in groups of four, with one input on each multiplexer sampled simultaneously with one pin on each of the other analog-to-digital converters. Each analog-to-digital converter is capable of converting 800K samples per second, making a maximum aggregate sample rate of 3.2 million samples per second.

The onboard operating system for the DAP 3400a is DAPL 2000, which is optimized for 32-bit operation. DAPL 2000 has been further customized for the DAP 3400a. A few commands, like the `SET` command, have changed for the DAP 3400a because of the specialized input configuration. A summary of modified, unsupported, and available commands is provided in Chapter 8.

About This Manual

This manual includes hardware and software installation instructions, a hardware connector reference, a DAPL 2000 reference, application examples, analog expansion pin mapping, and updated information for support software. Three other manuals provide information about creating data acquisition applications:

- The DAPL Manual contains a complete DAPL 2000 reference. Please read Chapter 8 of the DAP 3400a manual for updates to the DAPL 2000 operating system.
- The Applications Manual contains many useful examples of Data Acquisition Processor applications. Please note that some commands have a new syntax for the DAP 3400a, and note that analog output and digital input and output is not available on the DAP 3400a.
- The Systems Manual describes support software that runs in the PC, including support for writing programs that run in the PC.

2. Installation, Testing, & Troubleshooting

The DAP 3400a can be inserted into any 16-bit ISA or 32-bit EISA slot in an IBM-compatible personal computer. After installing the Data Acquisition Processor, install the support software from the disk(s) provided. Hardware and software installation instructions are provided in this chapter.

Once the hardware and software are installed, test the system to make sure that software is properly configured and that the Data Acquisition Processor is responding.

If you there are any problems with installation, please read the troubleshooting guide at the end of this chapter.

Data Acquisition Processor Handling Precautions

Static control is required for handling all electronic equipment. The Data Acquisition Processor is especially sensitive to static discharge because it contains many high-speed analog and digital components. To protect the Data Acquisition Processor, observe the following precautions:

- Wear a grounding strap when handling the Data Acquisition Processor. If it is not possible to use a grounding strap, continuously touching a metal screw on a grounded PC offers protection.
- If it is necessary to transport the Data Acquisition Processor outside of the PC, be sure to shield the Data Acquisition Processor in a conductive plastic bag. If a conductive bag is not available, shield the Data Acquisition Processor by wrapping it completely in aluminum foil. Do not ship or store a Data Acquisition Processor in plastic peanuts without suitable shielding.

Static damage to analog components can cause subtle problems, including oscillation, increased settling time, and reduced slew rate. If you suspect that a Data Acquisition Processor has been affected by static discharge, return it to Microstar Laboratories for testing, repair, and quality control.

DOS Version

The Data Acquisition Processor requires version 3.0 or higher of DOS. Updates to the operating system are available from PC hardware distributors.

Getting Started

The instructions in this chapter apply to PC systems containing only hardware options in the following list:

- diskette drive adapter
- color graphics adapter (CGA)
- monochrome display adapter
- enhanced graphics adapter (EGA)
- video graphics array (VGA)
- Hercules graphics adapter
- parallel printer adapter
- asynchronous communications adapter
- hard disk controller (PC/XT or PC/AT)
- memory expansion card
- game controller card

An AT/386/486/Pentium compatible computer that contains only these options is considered a standard configuration. Any system that contains an option not on this list is considered a nonstandard configuration. Users with nonstandard configurations should read Chapter 3 before proceeding.

Installing the Data Acquisition Processor

The DAP 3400a printed circuit board is compatible with 16-bit ISA or 32-bit EISA slots in AT/386/486/Pentium computers.

Caution: Do not install the Data Acquisition Processor while the PC is on.

To Install the Data Acquisition Processor:

1. Make any necessary changes to the hardware settings. See Chapter 4 for more information about hardware connectors.
2. Turn off the PC and remove the PC's cover.
3. Insert the Data Acquisition Processor into any free slot.

Make sure the PC's power supply is sufficient for all of the expansion cards in your system. The Data Acquisition Processor requires approximately 15 Watts. If your system behaves erratically with the Data Acquisition Processor installed, you may need a larger power supply.

Note: Some sound cards use the same I/O address as the Data Acquisition Processor. See Chapter 3 for more information.

Installing DAP Software

Before installing the DAP Software, make backup copies of the diskette(s). Put the originals in a safe place, and use the backup disks for installation.

The Microstar Laboratories INSTALL program installs the Data Acquisition Processor software on your system. To use INSTALL, boot your computer using DOS, place Data Acquisition Processor diskette 1 in drive A, and type the following command:

```
A: INSTALL
```

For systems with monochrome monitors, type:

```
A: INSTALL /BW
```

INSTALL prompts for configuration information including Data Acquisition Processor type and software destination directories. INSTALL provides information about each step to guide you through the installation process.

INSTALL copies several files to your boot disk and adds information to your system configuration files `CONFIG.SYS` and `AUTOEXEC.BAT`. Backup copies of these files are created with the `.BAK` extension so that the original versions can be recovered.

INSTALL also provides options for copying DAPview software and Data Acquisition Processor support software to your PC.

Testing Installation

After running the INSTALL program, verify that software installation was successful by rebooting your PC. Before you see the DOS prompt, the following lines should appear on your screen:

```
ACCEL device driver 4.3
ACCEL driver initialization completed
DAPL initialization completed
```

The exact lines may vary slightly, depending upon configuration options. If a line is missing or if an error message appears, see the Troubleshooting section in this chapter.

When the DOS prompt is displayed, set the current directory to your DAPview directory and run DAPview by entering the following command:

```
DV
```

DAPview allows you to communicate interactively with the Data Acquisition Processor. Now everything you type at the PC keyboard is sent to the Data Acquisition Processor and all Data Acquisition Processor messages are printed on your screen. When the DAPview program begins, the following lines should be displayed on your screen:

```
*** DAPview [1.2] ***
*** DAPL Interpreter [4.XX XX/X] Serial# XXXXX ***
#
```

The appearance of the # prompt indicates that the Data Acquisition Processor is installed correctly. If the # prompt does not appear or if DAPview issues an error message and terminates, you have not established communication with the Data Acquisition Processor. Turn to the end of this chapter for troubleshooting hints.

Note: To exit from DAPview, press the Ctrl and Z keys simultaneously.

Once DAPview is communicating with the Data Acquisition Processor, you can type DAPL commands at your keyboard. The number sign (#) on the left side of your screen indicates that the Data Acquisition Processor is waiting for a command. At this time you can enter sample applications from Chapter 9 or modified applications from the Applications Manual.

Troubleshooting

The Systems Manual contains a list of the error messages which may be printed during software installation and system boot. The following errors commonly result from installation problems:

`INSTALL prints an error message.`
Find the error message in the “System Messages” chapter of the Systems Manual.

`At system startup, the “ACCEL device driver 4.3” message is not printed. OR`
`DAPview prints the error message “Host communication port is uninitialized”`
Check that the file `CONFIG.SYS` is present on your boot volume. If this file was not present before installation, it should have been placed on your boot volume by INSTALL. If the file `CONFIG.SYS` is present, check that it includes the line

```
DEVICE=x:\yyy\ACOM.SYS ...
```

The `“x”` character should be the letter of your boot disk. `“yyy”` should be the correct directory where the file `ACOM.SYS` is located. If `CONFIG.SYS` is not on your boot volume, or if the `ACOM.SYS` line of the file `CONFIG.SYS` is incorrect, use INSTALL again, being careful to install the software on the correct volume.

`The message “Bad or missing ACOM.SYS” is printed.`
The file `ACOM.SYS` probably was not copied by INSTALL from the Data Acquisition Processor diskette to your boot volume. Use INSTALL again, being careful to install the software on the correct volume.

`When your system is booted, one of the following messages is printed:`

```
DAP hardware not found or improperly configured
DAP interrupt conflict
DAP interrupt selection error
```

These messages suggest a hardware conflict with another card in the PC; one or more of the Data Acquisition Processor configuration jumpers may need to be changed to resolve the conflict. See Chapter 3.

`At system startup, the message “ACCEL driver initialization completed” is not printed.`

If an error message is printed by the ACOMINIT program, find the error message in the Systems Manual. If no error message is printed, check that the `AUTOEXEC.BAT` file on your boot volume contains a line beginning with the command ACOMINIT. If no ACOMINIT line is found, use INSTALL again, being careful to install the software on the correct volume.

`At system startup, the message “DAPL initialization completed” is not printed.`

If an error message is printed by the DAPLINIT program, find the error message in the Systems Manual. If no error message is printed, check that the `AUTOEXEC.BAT` file on your boot volume contains a line beginning with the command DAPLINIT. If no DAPLINIT line is found, use the INSTALL pro-gram again, being careful to install the software on the correct volume.

`DAPview issues the error message “Host communication port is uninitialized.`

The `CONFIG.SYS` file was incorrect or missing at boot time.

`DAPview issues the error message “Could not establish communications” OR DAPview does not display a DAPL # prompt.`

The Data Acquisition Processor is not communicating with your PC. This may indicate that an error occurred at boot time. Check that no error messages are printed when you boot your system.

If your PC has cards other than those listed at the beginning of the chapter, a card may be interfering with communications. Remove optional cards, boot the PC, and try using DAPview again.

Check that the configuration jumpers on the Data Acquisition Processor are correct. These jumpers are labeled `HOST CONFIGURE` on the Data Acquisition Processor printed circuit board. See Chapter 4 for the correct jumper selections. Check also for consistency between the jumper settings on this connector and the settings on the `ACOM.SYS` line in the file `CONFIG.SYS`.

A final possibility is that the Data Acquisition Processor may be faulty. If you suspect that this is the case, call Microstar Laboratories Customer Support. When calling for installation support, please open your PC case so that the Data Acquisition Processor jumpers are visible, and be ready to provide the following information:

1. the serial and model numbers of your Data Acquisition Processor.
2. the contents of your `AUTOEXEC.BAT` and `CONFIG.SYS` files.
3. a list of all hardware boards installed in your computer.

`Your PC keyboard locks up when DAPview is started and does not accept the Ctrl-Z key.`

Your PC may have an old style keyboard. Try starting DAPview with one of the following command line options:

```
DV /K1
DV /K2
```

`DAPview issues the error “Help file DV.HLP not found.”`

Check that the file `DV.HLP` has been copied to the directory containing the DAPview files. Check also to make sure the file `AUTOEXEC.BAT` has the line:

```
SET DV=C:\DV
```

Replace `C:\DV` with the drive and directory containing the `DV.HLP` file.

3. Advanced Installation Options

Installation for standard hardware configurations is described in Chapter 2. This chapter covers installation in more detail.

Nonstandard Configurations

The Data Acquisition Processor uses two resources from the host PC:

- an interrupt vector
- a range of I/O addresses

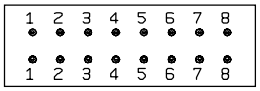
The Data Acquisition Processor allows several interrupt vector and I/O address selections. The interrupt vector may be 2, 3, 4, or 5. When selecting an interrupt vector, note the following interrupt vectors used by standard cards:

EGA/VGA	2
serial port COM2	3
serial port COM1	4
parallel port #2	5
hard disk controller on IBM XT	5
parallel port #1	7

The Data Acquisition Processor is shipped configured to use interrupt 2. EGA and VGA video adapters potentially can use interrupt 2, but most applications do not use this capability. Since interrupt 2 does not conflict with any other standard hardware, this is the default Data Acquisition Processor interrupt vector.

If any other cards are installed, determine the interrupts used and select a Data Acquisition Processor interrupt number distinct from these. Depending on the selection, the host computer may lose access to one of the serial COM ports or one of the parallel ports.

To change interrupt vectors, locate the sixteen-pin `HOST CONFIGURE` connector:



J10 HOST CONFIGURE

Connector `J10` is directly above the gold fingers on the Data Acquisition Processor printed circuit board.

The following table gives the four possible interrupt selections:

Interrupt Vector	Jumper
2	pin pair 7
3	pin pair 4
4	pin pair 5
5	pin pair 6

To change the interrupt, remove the jumper and replace it according to this table. Note that exactly one of the pin pairs 4, 5, 6, and 7 should be connected.

When the interrupt is changed, the INSTALL program must be informed of the new interrupt selection. When running INSTALL, select the `DAP 0` button and select the interrupt number that matches the interrupt of the Data Acquisition Processor.

In addition to an interrupt vector, the Data Acquisition Processor uses a range of I/O addresses. If a sound card or any nonstandard cards are used in the host PC, check that the Data Acquisition Processor I/O addresses do not conflict with those cards.

The Data Acquisition Processor is shipped configured to use I/O addresses in the range 220-22F (hexadecimal). This range may be changed by changing the jumpers on the `HOST CONFIGURE` connector. Pin pairs 1, 2, and 3 select the I/O address of the Data Acquisition Processor according to the following table:

I/O Address Range	Jumpers
220 - 22F	1, 2, 3
230 - 23F	2, 3
240 - 24F	1, 3
250 - 25F	3
320 - 32F	1, 2
330 - 33F	2
340 - 34F	1

When the I/O address is changed, the INSTALL program must be informed of the new address selection. When running INSTALL, select the `DAP 0` button and select the address that matches the address of the Data Acquisition Processor.

Note: The only effect of changing the interrupt number or address in the INSTALL program is to change the value of the `/I` or `/A` parameter of the line `ACOM.SYS` that is inserted in the file `CONFIG.SYS`.

Voltage Range Selection

When shipped, the Data Acquisition Processor is configured for analog inputs in the range from -5 volts to +5 volts. The range from -2.5 volts to +2.5 volts also is available. See Chapter 4 for voltage range selection instructions.

Installing Several Data Acquisition Processors

Up to seven Data Acquisition Processor boards can operate simultaneously in one PC. Running several boards in parallel increases the maximum sampling rate and the real-time processing power of a system. For special options to install up to 14 Data Acquisition Processor boards in one PC, contact Microstar Laboratories.

Each Data Acquisition Processor requires one PC slot. All the Data Acquisition Processors in a PC share just one interrupt line; no DMA lines are required. The Data Acquisition Processors are distinguished by their I/O addresses in the PC. Before installing Data Acquisition Processors in the PC, select a distinct I/O address for each board.

There are seven possible I/O addresses; this limits the number of Data Acquisition Processors in a PC to seven. Set the I/O addresses with the jumpers on the `HOST CONFIGURE` connector. Information about the `HOST CONFIGURE` connector is provided at the beginning of this chapter.

Note: Pin pair 8 of the `HOST CONFIGURE` connector sets the level of the PC's interrupt line. Pin pair 8 must be connected for one Data Acquisition Processor in a PC, and must not be connected for all other Data Acquisition Processors.

INSTALL can perform installation for several Data Acquisition Processors. When typing the INSTALL command, add the option `/Dx` to the end of the command, where `x` is the number of boards. For example, the following line defines installation for three Data Acquisition Processors.

```
A: INSTALL /D3
```

INSTALL provides on-screen options for configuring each Data Acquisition Processor individually. Select a DAP button to display a dialog box for choosing the Data Acquisition Processor model, address, and interrupt.

Note: The order in which the boards appear in the ACOM.SYS line in the file CONFIG.SYS determines the numbering of the Data Acquisition Processors. The first Data Acquisition Processor is DAP 0, the second is DAP 1, etc. The addresses do not matter when determining the numbering. When synchronous operation is used with DLOG, the last board in the list is the master unit.

The INSTALL Program

This section provides additional details about the INSTALL program. This information is of interest only to advanced Data Acquisition Processor users.

The INSTALL program installs Data Acquisition Processor software on a PC and configures the PC to initialize the ACCEL driver when the PC boots. INSTALL uses the following syntax:

```
A: INSTALL <options>
```

If installing from a drive other than A:, type the letter for that drive instead. INSTALL allows several command line options for special installation features. The following options are legal:

/Ix	use interrupt x
/DAPxxx:Ayyy	set Data Acquisition Processor type to xxx and set address to yyy
/COMx	Use serial port communication on com port x (DAP 801 only)
/Mxx	Default ACCEL driver mode (See the Systems Manual)
/Dx	Install more than one DAP
/?	Help

When running INSTALL, several prompts are provided for configuring Data Acquisition Processor software on a PC. Select a DAP button to choose the DAP model, address, and interrupt for the Data Acquisition Processor. Choose the “Select Software Options” button to select the specific software options to install. For help on the options press F1.

After all options have been verified, INSTALL copies Data Acquisition Processor software to the PC and modifies the system files. INSTALL copies the files ACOM.SYS, ACOMINIT.EXE, DAPLINIT, and Dx-x.STD to the PC boot drive. INSTALL creates the file ACOM.DAT and places it on the boot drive. These files are all placed on the boot drive so they are available when the PC first boots.

After copying the boot files, INSTALL copies the software that was specified in the Select Software Options dialog box to the Main directory.

INSTALL adds a line to the file CONFIG.SYS that loads the ACCEL device driver. If this line already exists from a previous installation, INSTALL replaces it. The following line is a typical line that INSTALL adds to CONFIG.SYS:

```
device=c:\dap\acom.sys i2 dap3200e:a220 m20 p19b
```

More information about the ACCEL driver configuration line is provided later in this chapter. If CONFIG.SYS does not exist, INSTALL creates it. Before modifying CONFIG.SYS, INSTALL saves an original copy in the file CONFIG.BAK.

INSTALL also modifies the file AUTOEXEC.BAT by adding several lines that configure the ACCEL driver with specific communication information. If these lines already exist from a previous installation, INSTALL replaces them. The following lines are typical lines that INSTALL adds to AUTOEXEC.BAT:

```
c:\dap\acominit c:\dap\acom.dat
@if errorlevel 1 pause
c:\dap\daplinit /reset c:\dap\d*.std
@if errorlevel 1 pause
```

The ACOMINIT program configures the ACCEL driver with specific communication pipe information provided by the file ACOM.DAT. For some Data Acquisition Processor models, the DAPLINIT program is required to initialize the DAPL operating system. More information about the ACOMINIT and DAPLINIT programs is provided later in this chapter. If AUTOEXEC.BAT does not exist, INSTALL creates it. Before modifying AUTOEXEC.BAT, INSTALL saves an original copy in the file AUTOEXEC.BAK.

Device Driver Configuration

This section explains the format of the device driver command that is placed in the `CONFIG.SYS` file by the INSTALL program. This information is of interest only to advanced programmers.

The format of the device driver command is:

```
DEVICE=ACOM.SYS [Ix] [DAP[yyy]:Azzz] [Muu] [Pwww]
```

Note: Several parameters are optional. The letters u through z in each parameter represent hexadecimal digits.

`x` specifies the interrupt vector that is used for PC communication. This number should match the configuration of jumper `J10` on the Data Acquisition Processor.

The ACCEL driver automatically detects the type of Data Acquisition Processor that is installed. `yyy` is optional and manually specifies the board type. For the DAP 3400a, the `DAPyyy` should be `DAP3400a`.

`zzz` specifies the hexadecimal starting I/O address of the Data Acquisition Processor. This number must match the configuration of jumper `J10` on the Data Acquisition Processor.

`uu` is a hexadecimal number that specifies the default mode of the ACCEL driver. See the Systems Manual for more information about ACCEL driver modes.

`www` is a hexadecimal number that specifies the number of paragraphs of memory to reserve for PC communications pipes. See "Com Pipe Configuration" later in this chapter.

The following is a typical ACCEL driver command line:

```
device=c:\dap\acom.sys i2 dap:a220 m20 p1A4
```

The ACCEL driver can be loaded into high memory with the DOS `devicehigh` statement. See your DOS manual for details.

The ACOMINIT Program

ACOMINIT configures the ACCEL driver communication pipes. ACOMINIT is placed in the file `AUTOEXEC.BAT` to configure the ACCEL driver when the PC first boots. The syntax for ACOMINIT is:

```
ACOMINIT <cfg_file>
```

`<cfg_file>` provides communication pipe configuration information. `<cfg_file>` normally is named `ACOM.DAT`. The following section describes the contents of `ACOM.DAT`.

Com Pipe Configuration

This section describes the format of communication pipe configuration in the file `ACOM.DAT`. This information is not required for most applications. During initialization, the Microstar Laboratories program ACOMINIT reads the contents of a configuration file which specifies a com pipe configuration. The configuration file determines the connection between com pipes on the Data Acquisition Processor and com pipes on the PC. Lines in the configuration file have the following syntax:

```
<source> -> <destination> [<options>]
```

`<source>` and `<destination>` are specifications of communication pipe locations. A com pipe location is either a Data Acquisition Processor com pipe, a PC com pipe, or a PC serial port:

```
(DAP[n] CPIPE v)
(PC CPIPE w)
(SERIAL x)
```

`n` is the Data Acquisition Processor number when several boards are installed in one PC. `v` is a DAPL com pipe number, `w` is a PC com pipe number, and `x` is a serial com pipe number. The space before `v, w,` and `x` can be omitted.

For example, the following lines connect the default text input and text output com pipes of the Data Acquisition Processor to the PC:

```
(dap cpipe 0) -> (pc cpipe 0)
(pc cpipe 0) -> (dap cpipe 0)
```

The first line connects DAPL output com pipe 0 to PC input com pipe 0. The second line connects PC output com pipe 0 to DAPL input com pipe 0. By default, DAPL defines two input com pipes and two output com pipes. Output com pipe 0, named `$SYSOUT`, is for text output to the PC. Output com pipe 1, named `$BINOUT`, is for binary output to the PC. Input com pipe 0, named `$SYSIN`, is for text input from the PC. Input com pipe 1, named `$BININ`, is for binary input from the PC.

Each line in the com pipe configuration file may contain one or more options, enclosed in square brackets. The following options are available:

```
BINARY|TEXT
MAXSIZE=xx
WIDTH BYTE | WORD | LONG
```

`BINARY` and `TEXT` specify the type of the data in the com pipe. The default is `TEXT`.

`MAXSIZE` specifies the size of the PC buffer of the com pipe, in bytes. The default is 1024 bytes. The INSTALL program sets the maximum size of the com pipes to be relatively small to conserve PC memory yet provide good performance. The Data Acquisition Processor automatically provides additional pipe buffering when needed. In some applications, increasing the maximum com pipe size can improve performance by allowing larger block operations. Performance can increase with com pipe sizes up to 4096 or 8192. Larger com pipe sizes typically do not provide further performance benefits.

`WIDTH` specifies the width of data items that are transferred through the com pipe. The `WIDTH` option must match the width of the corresponding Data Acquisition Processor com pipe. `BYTE` is the default for text com pipes and also is the only width allowed. `WORD` is the default for binary com pipes. Any width is allowed for binary com pipes.

The default com pipe configuration file generated by the INSTALL program is stored in the file `ACOM.DAT`. If the Data Acquisition Processor is operated inside a PC, the following configuration is placed in the file `ACOM.DAT`:

```
(dap0 cpipe0) -> (pc cpipe0) [text maxsize=1024]
(pc cpipe0) -> (dap0 cpipe0) [text maxsize=1024]
(dap0 cpipe1) -> (pc cpipe1) [binary maxsize=2048]
(pc cpipe1) -> (dap0 cpipe1) [binary maxsize=1024]
```

In some applications, additional com pipes need to be defined. More com pipes are needed when extra com pipes are defined in DAPL on a Data Acquisition Processor or when several Data Acquisition Processors are installed.

For special applications, extra com pipes can be defined in DAPL. See the `CPIPE` command in the DAPL Manual.

Note that in many cases, the commands `MERGE`, `MERGEF`, and `NMERGE` can be used instead of defining extra com pipes. It is best to use standard com pipes when possible to maintain a standard communication setup.

When more than one Data Acquisition Processor is installed in a system, additional com pipes need to be defined. For several Data Acquisition Processors, the following com pipe numbering is recommended:

```
DAP0
   system text pipe is PC com pipe #0
   system binary pipe is PC com pipe #1
DAP1
   system text pipe is PC com pipe #2
   system binary pipe is PC com pipe #3
 .
 .
DAP6
   system text pipe is PC com pipe #12
   system binary pipe is PC com pipe #13
```

P Parameter Size

In a system with additional com pipes, the memory available to the ACCEL driver must be increased. The `P` parameter in the `ACOM.SYS` line of the file `CONFIG.SYS` specifies the number of paragraphs of PC memory reserved for the ACCEL driver and com pipes. The storage requirement of the ACCEL driver and PC com pipes, in bytes, is:

```
storage = 830 + (maxsize in bytes) + (number of com
pipes)*190
```

`MaxSize in bytes` is the sum of all com pipe sizes defined in the file `ACOM.DAT`. `Number of com pipes` is the number of com pipes defined.

Note: This storage requirement applies for version 4.32 of the ACCEL driver. Subsequent driver versions may require additional storage.

The `P` parameter is a hexadecimal number, specified in paragraphs. A paragraph of PC memory is 16 bytes. To determine the `P` parameter, divide the storage requirements by 16 and convert to hexadecimal.

The following example calculates the `P` parameter for the default `ACOM.DAT` file created by INSTALL. The result, in bytes, is divided by 16 to get the `P` parameter in paragraphs.

```
storage = 830 + 5120 + 4*190 = 6710 bytes
p= 6710/16 = 420 paragraphs (decimal) = 1A4 (hex)
```

When defining additional com pipes, remember to define pipes for both the DAP-to-PC and PC-to-DAP direction. Some programs such as DAPview for Windows expect additional com pipes defined for both directions.

DAP-to-DAP Communication

Communication pipes can be configured to allow communication between two Data Acquisition Processors. DAP-to-DAP communication occurs over the PC bus in the background with no PC program intervention required. The following syntax is used in the file ACOM.DAT to configure DAP-to-DAP communication:

```
(DAPw CPIPEx) -> (DAPy CPIPEz)
(DAPy CPIPEz) -> (DAPw CPIPEx)
```

w is the number of the DAP that sends data. x is the DAP communication pipe used to send data. y is the number of the DAP that receives data. z is the DAP communication pipe to receive data. The DAPL command CPIPE is needed to define the DAP communication pipes on each DAP.

The following example shows how to configure a system for DAP-to-DAP communication. This example configures two DAPs. DAP 0 samples one channel of data and sends the data to DAP 1 for analog output.

In the file ACOM.DAT, add:

```
(dap0 cpipe15) -> (dap1 cpipe15)   [binary maxsize=1024]
(dap1 cpipe15) -> (dap0 cpipe15)   [binary maxsize=1024]
```

DAP-to-DAP com pipes require twice the storage space as DAP-to-PC or PC-to-DAP com pipes. For the above DAP-to-DAP com pipe definition, the P parameter in CONFIG.SYS must be increased by 130 (hex). The following example calculates the P-parameter increase:

```
storage increase= 2 * (2048 + 2 * 190) = 4856 (decimal)
p increase=4856/16 = 304 paragraphs (decimal) = 130 (hex)
```

The following DAPL commands provide an example of how to implement DAP-to-DAP communication.

```
;  DAPL commands for DAP 0:
CPIPE TODAP1 PC NUM=15 OUTPUT BINARY WORD
RESET
IDEF A 1
   SET(0...3) SPG0
   TIME 10000
   END
PDEF B
   COPY(IPIPE0, TODAP1)
   END
START A,B

;  DAPL commands for DAP 1:
CPIPE FROMDAP0 PC NUM=15 INPUT BINARY WORD
RESET
PDEF A
   COPY(FROMDAP0, $BINOUT)
   END
START A
```

The DAPLINIT Program

DAPLINIT initializes the DAPL operating system on the Data Acquisition Processor by downloading a binary image of DAPL to the Data Acquisition Processor. ACCEL driver communication pipes must be configured using ACOMINIT before DAPLINIT is run. The syntax for DAPLINIT is as follows:

```
DAPLINIT [/RESET] [<dapl_file>] [<dapl_file>]*
```

`<dapl_file>` specifies a binary file containing DAPL. DAPLINIT allows several DAPL binary files to initialize several Data Acquisition Processors in a PC.

The optional parameter `/RESET` requests a hardware reset of the Data Acquisition Processor before downloading DAPL. If `/RESET` is not specified, all Data Acquisition Processors retain their state during a warm PC boot.

DAPLINIT detects the Data Acquisition Processor model types on the `ACOM.SYS` line of the file `CONFIG.SYS`. DAPLINIT downloads the DAPL files, in order, to the Data Acquisition Processors that require DAPL initialization.

DAPLINIT can accept a wildcard file specification to allow flexibility for when DAP installations change. With a wildcard file name, DAPLINIT searches the current directory and the DOS PATH to find a DAPL binary file that matches the DAP type that is installed. The following example shows how DAPLINIT can be configured to search for the correct `DAPL/STANDARD` binary file.

```
DAPLINIT /RESET D*.STD
```

The INSTALL program automatically configures new installations to use a wildcard file name for DAPLINIT.

Installation on a Network

Data Acquisition Processor software can be installed on a network consisting of PC workstations connected to one or more servers. The INSTALL program can copy Data Acquisition Processor software to a PC workstation from a network that has a copy of the DAP Software disk image.

Note: When using Data Acquisition Processor software on a network, each simultaneous user must have a licensed copy of the software.

When installing Data Acquisition Processor software from a network, INSTALL copies several files to the PC workstation boot drive. The files are `ACOM.SYS`, `ACOMINIT.EXE`, `ACOM.DAT`, and, if necessary, `DAPLINIT.EXE` and `Dx-x.STD`. These files must be on the PC boot drive so that they are available immediately at boot time. INSTALL copies the remaining Data Acquisition Processor software to a network drive for use once the network is connected. INSTALL modifies `AUTOEXEC.BAT` and `CONFIG.SYS` on the workstation boot drive as in regular installations.

DAPL Licensing

When a Data Acquisition Processor is shipped from the factory, a copy of DAPL is provided that is licensed to run on the Data Acquisition Processor. When a DAP Software Upgrade is shipped, the DAPL file on the upgrade is licensed to be used with the Data Acquisition Processor that was specified when the upgrade was ordered.

Removing DAP Software

When a Data Acquisition Processor is removed from a PC, the software can be removed as well. This section describes how to remove a Data Acquisition Processor software installation.

1. Delete the directory where Data Acquisition Processor software was in-stalled. Usually this directory is `c:\dap`. Delete all the subdirectories under the DAP directory. Make sure that there are no important data files in these directories before deleting them.

2. Edit the file `AUTOEXEC.BAT`. Delete the four lines that match the following four lines:

```
c:\dap\acominit c:\dap\acom.dat
@if errorlevel 1 pause
c:\dap\daplinit /reset c:\dap\d*.std
@if errorlevel 1 pause
```

3. Edit the file `CONFIG.SYS`. Delete the line that matches the following line:

```
device=c:\dap\acom.sys i2 dap:a220 m20 p1A4
```

4. DAP 3400a Connectors

This chapter discusses the interface connectors on the DAP 3200a. Figure 2 shows component placement outlines of the DAP 3400a. The only components shown are connectors, whose labels begin with the letter J, some integrated circuits, whose labels begin with the letter U, and trim potentiometers, whose labels are single letters.

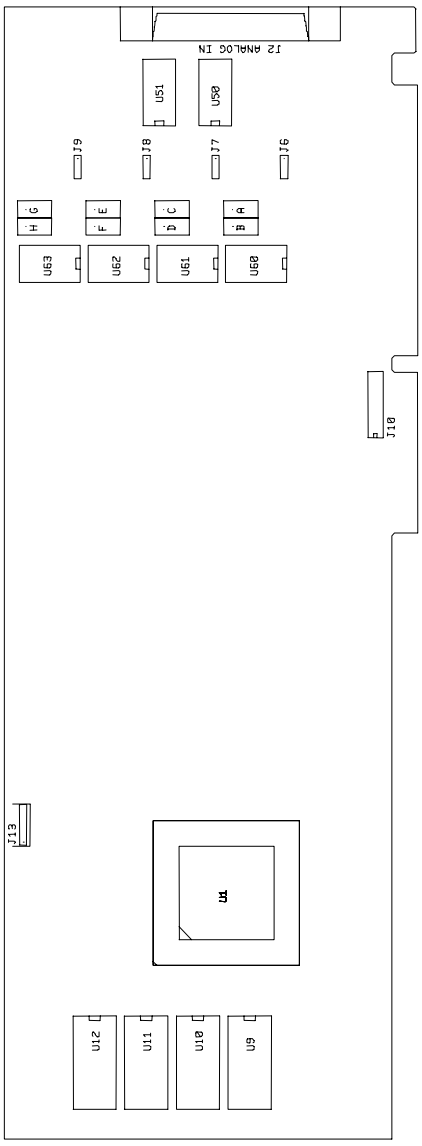


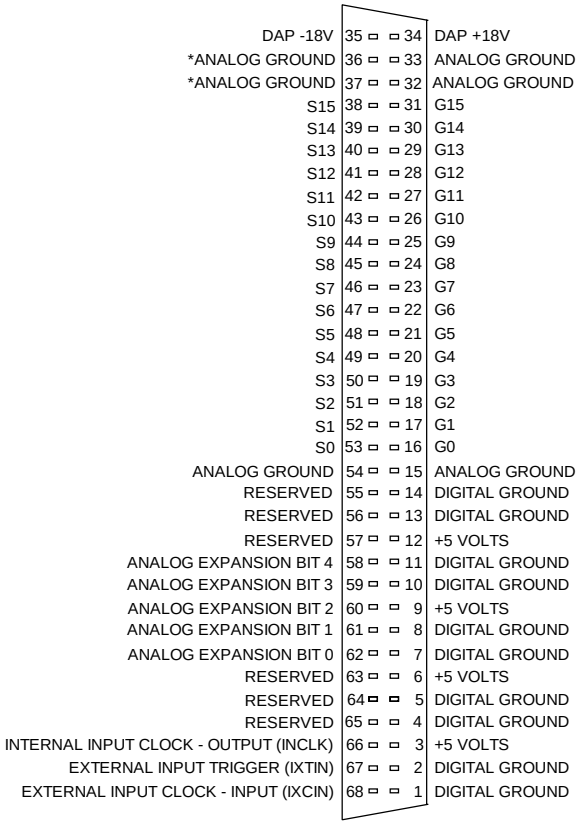
Figure 2.

DAP 3400a

Analog Input Connector

Analog voltages are connected to the Data Acquisition Processor through a 68-pin analog connector on the back panel of the PC. This connector is located on the right side of the Data Acquisition Processor and is labeled `ANALOG IN`. It has a double row of pins on 0.050 inch centers. The connector is 3M part number 10268-52E2VC or AMP part number 2-178238-8. It mates with discrete wire connector 3M part number 10168-6000EC or AMP part number 2-175677-8. Both connectors are shielded and are compatible with round cable. The analog input connector also mates with insulation displacement ribbon cable connector 3M part number 10168-8100EE.

Looking at the analog connector from the back of a PC, the pin numbering is:



* For compatibility with other DAPs, pins 36 and 37 should not be used. These pins are DAC outputs on other DAPs.

Note: Use the pin numbering on this chart, rather than numbers which may be found on your connector. Connectors from different manufacturers are not numbered consistently.

Inputs are indicated by S0 through S15; their corresponding ground inputs are G0 through G15.

A single-ended analog signal should be connected to an analog input pin with the signal ground connected to the corresponding ground pin, for example to pins 16 and 53.

An analog termination board, part number MSTB 009, is available from Microstar Laboratories. The termination board provides terminal blocks for connecting all lines of the analog connector to discrete wires.

The DAP 3400a features fault-protected input multiplexers. Fault protected input multiplexers allow signals to be connected to the Data Acquisition Processor with power off and allow a higher input voltage without damaging the inputs. Spare fault-protected input multiplexers are available from Microstar Laboratories.

Analog input signals should be within the range from -25 volts to +25 volts, relative to the ground of the Data Acquisition Processor. Input signals may be applied to the Data Acquisition Processor when the PC's power is off. See Chapter 5 for electrical characteristics of the analog input pins.

The analog connector of the Data Acquisition Processor includes pins for analog power supply voltages. Pin 15 is analog power ground. Pins 34 and 35 are connected to +18 volt and -18 volt analog supplies, respectively. The maximum allowable current drain from these supplies is 20 milliamps per side. These supplies can be used for low current, low noise devices such as external multiplexers. To supply power to other devices, either use an external supply or use the 5-volt digital power supply found on the analog control connector with an external DC-to-DC converter.

Pins 58 - 62 provide TTL-compatible analog input expansion control signals. These control signals are the five highest order address bits of the input pin group number (bits 2-6). The analog input expansion control signals are valid from immediately after one sample to immediately after the next sample. The analog input expansion control bits designate the address of the next sample to be taken.

External analog input expansion boards are available from Microstar Laboratories. Each input expansion board allows up to 64 single-ended inputs. See Chapter 7 for more information.

The analog connector has an input pin for an active high external input trigger. The external input trigger can be used to control when input sampling occurs. To use the external trigger, an `HTRIGGER` command is needed in the active input procedure. The external trigger is ignored if there is no `HTRIGGER` command in the active input procedure. See Chapter 6 for more information.

The external input trigger signal must be within the standard TTL logic range of 0 to +5 volts. The Data Acquisition Processor provides a 10k Ohm pull-up resistor on the external trigger input. This forces the trigger input active if it is left unconnected. Clock and trigger inputs may have signals applied when the Data Acquisition Processor is off.

The analog connector has an input pin for an external input clock. The input pin should be used to connect an external input clock. The analog connector also has an output pin for the input clock. The output pin is the buffered output of the internal clock circuit. See Chapter 6 for more information on external clocks and triggers.

Shunts

Several of the Data Acquisition Processor options are set by shunts. These are jumper wires enclosed in plastic, designed for connecting pins on 0.100" centers.

Each shunt has a top and a bottom. When a shunt is placed correctly, a probe point is visible in the shunt. Shunts must not be placed upside down on the pins, as incorrectly placed shunts do not provide reliable contacts.

Analog Signal Path Selection

In addition to the DAPL configuration options, the Data Acquisition Processor has one hardware configuration option which determines the analog input voltage range.

The analog signal path between the input pins and the analog-to-digital converter consists of the following functional units:

1. input multiplexers
2. buffer amplifier
3. range amplifier
4. analog-to-digital converter with sample-and-hold amplifier

Analog signals pass through all of these functional units. Jumpers determine the analog input voltage range for each of the four analog input sections. Each section may be configured independently. The Data Acquisition Processor should be recalibrated after changing the input voltage range.

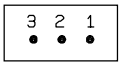
A signal range is called bipolar if it includes both positive and negative voltages; a signal range is called unipolar if it includes voltages of only one sign. The DAP 3400a has only bipolar input ranges.

A signal is single ended if it is measured relative to ground. A signal is differential if it is measured relative to another signal The DAP 3400a accepts only single-ended inputs; there are no provisions for differential inputs.

Analog Input Voltage Range Configuration

The following connectors control the analog input voltage range of the DAP 3400a. Note that changing voltage ranges generally requires recalibration.

Connectors J6 through J9 select the input signal range of the range amplifier. J6 selects the range for channels 0 through 3, J7 selects the range for channels 4 through 7, J8 selects the range for channels 8 through 11, and J9 selects the range for channels 12 through 15.



J6, J7, J8, J9

One jumper should be placed on each of J6 through J9 as follows:

Jumper	Input signal range
1-2	± 2.5 volts
2-3	± 5 volts

Note: Regardless of the input voltage range, positive and negative signals may range from -25 volts to +25 volts without damaging the Data Acquisition Processor.

The following table summarizes the DAP 3400a jumper connections:

ADC Range	J6, inputs 0-3	J7, inputs 4-7	J8, inputs 8-11	J9, inputs 12-15
± 2.5v	1 - 2	1 - 2	1 - 2	1 - 2
± 5v*	2 - 3 *	2 - 3 *	2 - 3 *	2 - 3 *

* Factory Configuration

Host Configuration Connector

Information on the Host Configuration Connector (`J10`) is provided in Chapter 3 in the "Nonstandard Configurations" section.

Synchronization Connector

The synchronization connector J13 has a single row of five pins on 0.100 inch centers. J13 is located at the upper left of the Data Acquisition Processor printed circuit board. The synchronization connector allows several Data Acquisition Processors to share the same sampling clock. See the Systems Manual for more information about using synchronous Data Acquisition Processors.

For synchronization, the shared sampling clock requires special cabling between Data Acquisition Processors. The analog sampling clock of the master unit is supplied to the slave units by means of the cable MSCBL 015-01. Contact Microstar Laboratories for more information about cabling for synchronous Data Acquisition Processors.

5. Analog Input Circuits

The analog input hardware of the Data Acquisition Processor is discussed in some detail in this chapter. The following summary gives sufficient information for most Data Acquisition Processor applications:

- The DC input impedance is very high.
- At high sampling rates, the signal source impedance should be low.

Analog Input Circuits

Data Acquisition Processor analog input signals pass through an analog multiplexer and then to an op amp with an FET input. The DC input impedance is very high, typically far in excess of 10M Ohms. The AC input impedance is dominated by the capacitance of the multiplexer.

Figure 3 shows an equivalent circuit for each Data Acquisition Processor input. As the Data Acquisition Processor scans through the input pins defined by an input procedure, the switches in the multiplexers open and close, connecting the specified inputs to multiplexer outputs. When an input signal is connected to the FET amplifier, the signal source must supply sufficient current to charge the equivalent capacitance of the multiplexers before the analog-to-digital conversion can start.

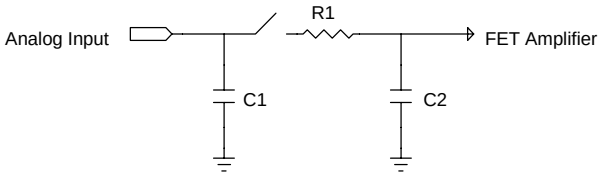


Figure 3.

The DAP 3400a has fault-protected multiplexers. The following table shows typical resistance and capacitance values, in Ohms and picofarads, for fault-protected multiplexers.

Component	Value
R1	300Ω
C1	5 pF
C2	30 pF

6. Clocks and Triggers

The Data Acquisition Processor is designed to operate either using internal clocks or external clocks. The Data Acquisition Processor has onboard crystal controlled timers to provide an internal input sampling rate and also has provisions for external clocks.

The DAP 3400a has hardware control lines for an input clock and an input trigger. These lines are TTL compatible. The input clock is positive edge triggered.

The input clock has two modes. In the first mode, called Channel List Clocking, the Data Acquisition Processor starts conversion of an entire channel list on the positive edge of the clock. In the second mode, the Data Acquisition Processor converts a single pin group on the positive edge of the clock.

The input trigger also has two modes, a one-shot mode and a level triggered gate mode.

Software Triggers vs. Hardware Triggers for Input

DAPL provides a powerful software triggering mechanism suitable for most applications. For those applications that require precise synchronization to external hardware or that are too fast to take advantage of software triggering, hardware triggering is provided. Software triggering is more versatile than hardware triggering. In applications with low enough sampling rates, software triggering almost always provides a better solution than hardware triggering.

Software triggers rely on DAPL tasks to scan input data to detect events within the data. When an event is detected, a task asserts a software trigger. After the trigger is asserted, another task may act, based on the assertion. The most common action is to pass a number of values around the trigger event either to another task or to the PC. The trigger mechanism is much like the trigger on an oscilloscope. Since all of the processing functions of DAPL may be used to define events, however, much more complex events may be detected.

Hardware input triggers are implemented using a digital control line which is separate from the sampling stream. This control line starts and stops input sampling. Since the trigger line is not dependent on the input data, external hardware must be provided to detect events of interest.

Software triggers have several advantages over hardware triggers. First, a software trigger may be changed by changing a few lines in a DAPL command list. In contrast, a hardware trigger event must be detected by external hardware which may be inflexible and costly to modify. Second, software triggers scan input data to detect events, so pretrigger data are available. Because a hardware trigger starts the Data Acquisition Processor input section, no samples are taken before a trigger event. Finally, with software triggers DAPL provides precise timing information. With hardware triggers DAPL is not able to provide accurate timing information because hardware triggers start and stop input sampling at undefined times.

Hardware triggering does provide precise synchronization of acquisition to external events. Hardware triggering also allows detection of events which are too fast to process with software triggers.

Software and hardware triggering are implemented separately and may be used together.

External Input Clock

The external input clock is a positive-edge triggered TTL signal. The external input clock is activated by the command `CLOCK EXTERNAL` in an input procedure. The `TIME` command of an input procedure with input clocking enabled must be at least tSYNCH less than the period of the external clock. tSYNCH is defined at the end of this chapter.

External input clocking has two modes. The first mode, called Channel List Clocking, starts conversion of an entire channel group list on the positive edge of the external clock. The second mode converts a single pin group on the positive edge of the external clock. The selection between the modes is made by a parameter to the `CLCLOCKING` command in an input procedure. The two options for `CLCLOCKING` are `ON` and `OFF`. The default is `ON`.

Example:

```
IDEF A 4
   CLOCK EXTERNAL
   CLCLOCKING ON
   SET(0...3) SPG0
   SET(4...7) SPG1
   SET(8...11) SPG2
   SET(12...15) SPG3
   TIME 1000
   . . . .
   END
```

In this application, external input clocking is enabled for the input procedure `A`. With Channel List Clocking selected, each positive edge of the external clock causes conversion of the entire channel list consisting of pin group 0 (`SPG0`) to pin group 3 (`SPG3`). The pin groups are converted in sequence with pin group `SPG0` synchronized to the positive edge of the external clock and each of the subsequent pin groups converted according to the `TIME` command. Pin group 1 (`SPG1`) is converted 1000 µs following the edge of the external clock, pin group 2 (`SPG2`) is converted 2000 µs following the edge of the external clock, and pin group 3 (`SPG3`) is converted 3000 µs following the edge of the external clock. When using Channel List Clocking, the period of the external clock (rising edge to rising edge) must be greater than the `TIME` command times the number of pin groups plus tSYNCH. The external clock may be as slow as required—there is no maximum period.

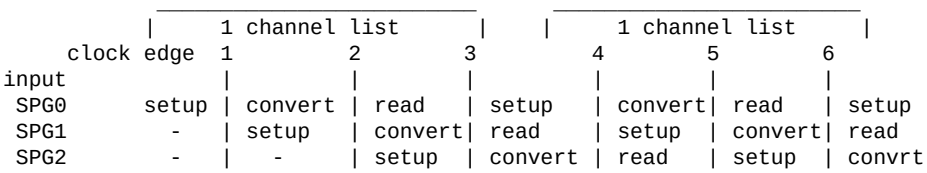
If single group clocking is selected rather than Channel List Clocking, each positive edge of the external clock causes conversion of only one pin group. The pin groups are converted in sequence. Each pin group is synchronized to a positive edge of the external clock. In the previous application, pin group 0 (`SPG0`) is converted on the first edge of the external clock, pin group 1 (`SPG1`) is converted on the second edge of the external clock, and so on up to pin group 3 (`SPG3`), which is converted on the fourth edge of the external clock. The pin group list then is repeated with pin group 0 converted again on the fifth positive edge of the external clock. When using single group clocking, the period of the external clock (rising edge to rising edge) must be greater than the `TIME` command plus tSYNCH. The external clock may be as slow as required; there is no maximum period.

Input Pipeline Timing

This section is of interest only when using external clocking or low latency with internal clocking.

The DAP 3400a has two pipeline stages for analog inputs. An acquired value is read by the CPU after the following clock edge. This causes one sample period of time to be added to Data Acquisition Processor response latency. When channel list clocking is used with an external clock, the values for the last channel group in the channel list are not read until the following external clock or until the input procedure is stopped. There are several cases described later that increase the number of extra clocks that are necessary before the expected number of values are read.

The following timing diagram illustrates the input timing for the DAP 3400a.



Note that `setup` indicates the setup of analog input circuits before the value is sampled, `convert` indicates when the value is held and converted to digital, and `read` indicates the value is read by the CPU. Each action occurs at or soon after the previous clock edge in the diagram.

Hardware Input Trigger

There are two modes for the input trigger. The first is a one-shot mode and the second is a level-controlled gated mode. The mode of the hardware trigger is selected by a parameter to the `HTRIGGER` command in an input procedure. The three options for `HTRIGGER` are `ONESHOT`, `GATED`, and `OFF`. The default is `OFF`.

In the one-shot mode, the trigger line is held in a low state when an input procedure is started. Input sampling does not start until the trigger line is high. Sampling continues until a `STOP` command is issued, or the number of samples specified by the `COUNT` command of the input procedure is reached, or all onboard memory is consumed. The first sampled value is precisely synchronized to the trigger edge and all subsequent values are within ± tSYNCH of the `TIME` command of the input procedure. tSYNCH is defined at the end of this chapter. The active period of the external input trigger must be greater than 60ns to guarantee proper operation.

In the level-triggered gated mode, input sampling may start and stop repeatedly, depending on the level of the trigger signal. The input is sampled continuously when the trigger signal is high. Input sampling stops when the trigger signal is low.

When input clocking is configured in Channel List Clocking mode, the input is stopped only at channel list boundaries. When input clocking is configured to clock single pin groups, the input is stopped on pin group boundaries. The effect of this is that the start of sampling is precisely synchronized to the positive edge of the trigger signal, assuming that sampling has stopped. Sampling stops when the Data Acquisition Processor has completed sampling of either a channel list or a pin group. When input sampling has been stopped with the gated trigger, synchronization of sampling to the positive edge of the trigger signal is the same as for the one-shot mode.

Timing Considerations

When an external clock is used, the time of an event with respect to the start of sampling may be determined only if the period of the external clock is known. DAPL establishes event times as sample times. If the external clock period is variable or the period is unknown, the time of an event cannot be determined. The event's sample number may still be useful in other contexts. Note that the results of all frequency domain processing such as `FREQUENCY`, `FFT32`, and `FIRFILTER` depend on the period of the external clock and may not be defined if the external clock period varies.

When hardware triggering is used, DAPL provides timing information relative to the start of each external trigger. In a case of a one-shot trigger, sampling starts on a single event, so all timing information is relative to the trigger event. In the case of a gated trigger, sampling may start or stop at arbitrary times. Timing information may still be obtained if means are provided to distinguish one external trigger event from the next.

Using the Input Trigger with External Input Clocking

Input triggering may be used with external input clocking. When these functions are used together, however, precise synchronization of acquisition to a trigger edge is not available. The reason for the loss of synchronization is that the Data Acquisition Processor has no control over the external clock.

The Data Acquisition Processor acts upon the first external clock cycle after the trigger has been asserted, assuming input sampling has stopped. To guarantee recognition of an external trigger, the external trigger must occur at least 50 ns before the positive edge of the external clock.

Timing tables

tSYNCH	200 ns	Time needed to synchronize an internal clock to an external clock
tTRIG_MIN	60 ns	Minimum high period for the input and output trigger
tEXTCLK_PW	25 ns	Minimum high or low period of an external clock
tTCSETUP	50 ns	External trigger to external clock setup time
tITRIG_MAX	250 ns	Maximum high period of the input trigger to guarantee a single conversion
tINSKEW	200 ns	Time from input clock or trigger to conversion value held

7. Termination, Expansion, & Cabling

DAP 3400a Termination Boards

Part Number	Description	Cables
MSTB 009	Analog Termination Board	MSCBL 040-01 or MSCBL 041

DAP 3400a Expansion Boards

Part Number	Description	Cables
MSXB 018	Analog Input Expansion Board	MSCBL 040-01 or MSCBL 041

8. DAPL 2000 for the DAP 3400a

The DAP 3400a uses DAPL 2000 as the onboard operating system. Because the DAP 3400a has no analog output or digital section, some DAPL 2000 commands have changed, and some commands are not supported.

This chapter summarizes changes to DAPL 2000 for the DAP 3400a and provides a summary of all available commands. Following the summary of commands, all commands that have been modified for the DAP 3400a are fully documented. Any commands not documented in this manual remain syntactically unchanged and are documented in the DAPL Manual.

Modified DAPL 2000 commands

```
CLCLOCKING
COUNT
IDEFINE
SET
TIME
```

Unsupported DAPL 2000 commands

```
ALARM
AUTORANGE
CTCOUNT
CTRATE
CYCLE
DACOUT
DEXPAND
DIGITALOUT
DISPLAY UNDERLOWQ/OCOUNT/OUTPORT
ODEFINE
OPTION BPOUTPUT/DACEXPAND/DIGITALEXPAND/UNDERFLOWQ
OUTPORT
OUTPUTWAIT
```

Summary of Available Commands

DAPL commands are divided into four groups:

- system commands
- structure definition commands
- input configuration commands
- task definition commands

More information on each type of command is available in the DAPL Manual. Commands with an asterisk (*****) are documented in this manual, as the syntax and/or operation of that command is unique for the DAP 3400a.

System commands:

System commands start and stop sampling, set system options, and request status information:

`DIAGNOSTIC` .....tests Data Acquisition Processor hardware
`DISPLAY` .....displays symbol and system status information
`EDIT` .....modifies input procedures and com pipes
`EMPTY` .....empties all data from a pipe
`ERASE` .....removes a symbol
`FILL` .....adds data values to a pipe
`HELLO` .....returns a line including the DAPL version number
`HELP` .....returns a list of currently valid DAPL commands
`LET` .....changes the value of a variable or constant
`OPTIONS` .....changes a system option
`PAUSE` .....pauses DAPL
`RESET` .....resets DAPL
`RESTART` .....performs a power-up restart of DAPL
`SAMPLEHOLD` .....waits for input processing to stop
`SDISPLAY` .....displays information about symbols
`START` .....starts input and processing procedures
`STATUS` .....displays system status
`STOP` .....stops input and processing procedures
`TASKSTAT` .....displays task statistics

Structure Definition Commands:

Structure definition commands define symbols, allocate memory, and assign values to symbols:

`BDOWNLOAD` ..... defines a custom command
`CONSTANTS` ..... defines constants
`CPIPE` ..... defines a communication pipe
`IDEFINE *`..... defines an input procedure
`PDEFINE` ..... defines a processing procedure
`PIPES` ..... defines pipes
`STRING` ..... defines a string
`TRIGGERS` ..... defines triggers
`VARIABLES` ..... defines variables
`VECTOR` ..... defines a vector.

Input Configuration Commands:

Input configuration commands define the sampling configuration of the Data Acquisition Processor:

`BUFFERS`..... selects the input buffer mode
`CLCLOCKING*` ..... selects the channel list clocking mode
`CLOCK` ..... selects internal or external clocking
`COUNT *`..... selects the number of samples to acquire
`HTRIGGER` ..... selects the hardware triggering mode
`SET *`..... associates an input channel group with an analog input pin group
`TIME*` ..... selects the sampling interval
`UPDATE` ..... selects continuous input operation or burst input operation

Task Definition Commands:

Task definition commands cause the Data Acquisition Processor to set up the structures required for executing tasks:

ABS ..... computes absolute values

ALARM ..... generates digital alarm signals
AUTORANGE ..... performs data autoranging
AVERAGE ..... averages pipe data
BAVERAGE ..... performs block averaging
BINTEGRATE ..... integrates blocks of data
BMERGE ..... merges blocked data
BMERGEF ..... merges blocked data with identifying flags
BPRINT ..... transfers binary input channel pipe data to the PC
CABS ..... computes the sum of squares of the values in two pipes
CHANGE ..... scans for changes in data
COMPRESS ..... compresses data flow for inputs that change infrequently
COPY ..... copies the data in a pipe into several other pipes
COPYVEC ..... copies data from a vector to a pipe
CORRELATE ..... computes cross correlations
COSINEWAVE ..... generates cosine waveforms
CROSSPOWER ..... computes cross power spectrum
CTCOUNT ..... accumulates long word counts from counter timer board data
CTRATE ..... computes frequencies from counter timer board data
DECIBEL ..... converts positive values to decibels
DELTA ..... computes the derivatives of the values in a pipe
DLIMIT ..... scans data for slopes which are out of range
EXTRACT ..... extracts single bits from word data
FFT ..... computes fast Fourier transforms, emphasizing speed
FFT32 ..... computes fast Fourier transforms, emphasizing precision
FINDMAX ..... finds the locations of maxima in blocks of data
FIRFILTER ..... applies digital filtering and reduces data volume
FIRLOWPASS ..... applies predefined lowpass digital filtering
FORMAT ..... format data as text and transfer to the PC
FREQUENCY ..... determines frequencies of trigger assertions
HIGH ..... computes maxima of blocks of data
INTEGRATE ..... computes the running integral of pipe data

`INTERP` ..... interpolates with a lookup table
`LCOPY` ..... copies the data in a pipe into several other pipes with minimal latency
`LIMIT` ..... scans data for values which are out of range
`LOGIC` ..... scans binary data for bit transitions
`LOW` ..... computes minima of blocks of data
`MAXTIME` ..... enforces a maximum time between trigger assertions
`MERGE` ..... merges data from several pipes into one pipe
`MERGEF` ..... merges binary data from several pipes, appending identifying flags
`MINTIME` ..... enforces a minimum time between trigger assertions
`NMERGE` ..... merges different quantity of data from several pipes into one pipe
`NTH` ..... removes excess trigger events
`OFFSET` ..... performs an offset adjustment
`PCASSERT` ..... generates triggers based on PC control
`PCOUNT` ..... counts the number of values placed in a pipe
`PEAK` ..... searches for maxima and minima of pipe data
`PID` ..... computes data for closed-loop process control
`PID1` ..... computes data for closed-loop process control, reducing the control error to 0
`POLAR` ..... converts from rectangular to polar coordinates
`PRINT` ..... prints all input channel pipe data to the PC
`PULSECOUNT` ..... counts the number of digital input pulses
`PVALUE` ..... determines the most recent value in a pipe
`PWM` ..... performs pulse width modulation
`RANDOM` ..... generates pseudorandom numbers
`RANGE` ..... removes data values which are out of range
`RAVERAGE` ..... computes running averages of pipe data
`REPLICATE` ..... copies data, repeating each value a specified number of times
`RETRIGGER` ..... adjusts trigger assertions to force a minimum time between triggers
`RMS` ..... computes root mean square values
`SAWTOOTH` ..... generates sawtooth waveforms
`SCALE` ..... scales pipe data and adds an offset
`SCAN` ..... waits for a group of input pins to be sampled before transferring data
`SEPARATE` ..... separates merged data

SEPARATEF .....separates flagged merged data
SINEWAVE .....generates sine waveforms
SKIP .....deletes selected blocks of data
SQRT .....computes square roots
SQUAREWAVE .....generates square waveforms
TAND .....combines triggers with logical ‘and’
TCOLLATE.....combines triggers producing a combined event stream
TFUNCTION1 .....calculates transfer functions from frequency domain data
TFUNCTION2 .....calculates transfer functions from cross-power spectra and auto-power spectra
TGEN .....generates periodic triggers
THERMO .....performs thermocouple linearization on data
TOGGLE .....tests trigger events for alternating ON and OFF events
TOGGWT .....collects data for alternating ON and OFF trigger events
TOR .....combines triggers with logical ‘or’
TRIANGLE .....generates triangle waveforms
TRIGARM.....allows a task to arm or disarm software triggers by setting trigger properties
TRIGRECV.....receives encoded software trigger information
TRIGSCALE.....modifies a stream of trigger events
TRIGSEND.....encodes trigger information to be transferred to another DAP
TSTAMP .....converts trigger assertions to time stamps
VARIANCE .....computes the statistical variance of pipe data
WAIT .....waits for a trigger event and transfers trigger data to a pipe
WAVEFORM .....generates analog waveforms

Command Reference

The following commands have been modified in DAPL 2000 for the DAP 3400a. For these commands, please use the syntax described here and ignore the syntax shown in the DAPL Manual.

CLCLOCKING

Set the channel list clocking mode of an input configuration.

CLCLOCKING *<switch>*

Parameters

<switch>
A keyword, either `ON` or `OFF`.

Description

`CLCLOCKING` sets the channel list clocking mode of an input configuration. *<switch>* is `ON` or `OFF`. The default value is `ON`.

The syntax of `CLCLOCKING` is the same for the DAP 3400a as with other Data Acquisition Processors. With the DAP 3400a however, all four samples in an input channel group are sampled simultaneously.

Example

```
IDEFINE A1
   CLCLOCKING OFF
   ...
   END
```

Turn channel list clocking off.

COUNT

Establish a fixed data block length for an input configuration.

COUNT *<sample_count>*

Parameters

<sample_count>

An integer which specifies the number of sampling or updating operations.
`WORD CONSTANT | LONG CONSTANT`

Description

A `COUNT` command in an input configuration definition sets the number of input samples the input configuration acquires. *<sample_count>* specifies the number of samples in a block. *<sample_count>* must be an integral multiple of the number of input pins defined in the channel list. For the DAP 3400a, input pins are always sampled in groups of four, so *<sample_count>* must also be a multiple of four.

If the number of channel groups defined with `IDEFINE` is `N`, `COUNT` must be a multiple of (`N*4`). For example, to sample each input pin five times, *<sample_count>* is specified by the result of 5(`N*4`).

Example

```
IDEFINE A 2
   SET IPIPES(0..3) SPG0
   SET IPIPES(4..7) SPG1
   COUNT 16
   END
```

When the input configuration is started, the Data Acquisition Processor acquires 16 samples and then stops. Since eight pins are sampled (0-7), each pin is sampled twice in this procedure.

IDEFINE

Create and initialize an input configuration.

IDEFINE *<name> <n>*

Parameters

<name>

The unique name of the input configuration.
Alphanumeric string limited to 11 characters.

<n>

The number of input channel pipes.
`WORD CONSTANT`

Description

`IDEFINE` creates and initializes an input configuration. An input procedure name must be specified, as well as the number of input channel pipes. When `IDEFINE` is issued, input procedure definition mode is entered. This is indicated by a change of the Data Acquisition Processor prefix character from # to >. In input procedure definition mode, input configuration commands are entered to configure input channel pipes, set gains, define sampling time, etc. To leave input procedure definition mode, an `END` command is issued.

<name> is a unique name given to the input configuration. *<name>* must be an alphanumeric string with no spaces and is limited to 11 characters.

<n> is the number of input channel pipes that will be defined in the input configuration, and must be a positive integer between 1 and 128. There should be *<n>* `SET` commands defined in the input procedure.

See the `SET` command for information about defining input channel groups.

Example

```
IDEFINE BLUE 2
   SET IPIPES(0..3) SPG0
   SET IPIPES(4..7) SPG1

END
```

Begin the definition of an input configuration named `BLUE` with 2 input channel pipes.

SET

Associate an input channel group with an input pin group.

SET IPIPES(*<channel_group>*) *<pin_group>*

Parameters

<channel_group>
A special input channel pipe list notation for a group of four input channels.

<pin_group>
A keyword representing the single-ended pin group.

Description

SET associates an input channel group with an input pin group. One channel group is associated with any one of the input pin groups. Once this association is made, the connections between channel and inputs are fixed.

The term IPIPES may be abbreviated to IP.

The parameter *<channel_group>* is a range of four input channel numbers. The range must start with an integer that is a multiple of 4 and must use four consecutive integers. For example, the following ranges are acceptable:

```
IP(0..3)
IP(4..7)
IP(8..11)
IP(12..15)
...
IP(508..511)
```

The range must be specified as shown above. Other syntax is not accepted for DAP 3400a boards. Input channel pipe numbers must appear in ascending order and must not be repeated. Ranges above IP(12..15) are available only with analog input expansion.

The parameter *<pin_group>* is specified by SPGx, where x is an integer from 0 to 127. A pin group consists of four single-ended analog input pins. Each pin in a group is connected to a separate analog-to-digital converter. All four pins in a group are sampled simultaneously.

DAP 3400a boards have 16 analog inputs, which can be expanded to 512 inputs. Each pin group represents a fixed set of input pins. The mapping of input pins to pin groups is as follows:

```
SPG0   S0, S4, S8, S12
SPG1   S1, S5, S9, S13
SPG2   S2, S6, S10, S14
SPG3   S3, S7, S11, S15
```

For expansion, the numbering of pin groups continues in the same manner:

```
SPG4   S16, S20, S24, S28
SPG5   S17, S21, S25, S29
SPG6   S18, S22, S26, S30
SPG7   S19, S23, S27, S31
   ...
SPG127,S499, S503, S507, S511
```

Pin mapping for analog expansion is provided in Appendix A.

Any of the input channel pipes defined with the `SET` command can be used in a DAPL task. A DAPL task can read sampled data by reading from an input channel pipe named by `IPIPE`*<n>*, where *<n>* specifies the input channel pipe number. `IPIPE` can be abbreviated to `IP`.

For example, the command

```
SET IPIPES(0..3) SPG2
```

configures the following connections:

```
IPIPE0 to S2
IPIPE1 to S6
IPIPE2 to S10
IPIPE3 to S14
```

A valid DAPL `AVERAGE` command may look like this:

```
AVERAGE (IP2, 100, P1)
```

Example

```
IDEF A 1
  SET IPIPES(0..3)  SPG3
  SET IPIPES(4..7)  SPG0
  SET IPIPES(8..11) SPG1
  ...
  END
```

Associate the channel pipes `IP0-IP3` to pins `S3`, `S7`, `S11`, and `S15`; channel pipes `IP4-IP7` to pins `S0`, `S4`, `S8`, and `S12`; and channel pipes `IP8-IP11` to pins `S1`, `S5`, `S9`, and `S13`.

TIME

Set the rate at which successive inputs are sampled.

TIME *<value>*

Parameters

<value>

The time interval in microseconds between consecutive pin group sampling.

Description

`TIME` specifies the time interval at which two consecutive pin groups are sampled. Each pin group consists of four input pins that are sampled simultaneously. The next input group is sampled *<value>* microseconds after the previous group.

With a `TIME` value of `N` with `M` pin groups, an input configuration will sample the entire channel list every `N*M` microseconds.

The minimum *<value>* for the `TIME` command for the DAP 3400a/445 is 1.25 µs, with the smallest increment of 0.05 µs. The largest `TIME` allowed is 52,428 us.

The Data Acquisition Processor allows two decimal places of precision in the time specification.

For compatibility among all current Data Acquisition Processor models, the time parameter should not exceed 10,000 µs. Note that even though `TIME` restricts the maximum sampling time, the effective sampling interval can be made as slow as desired by using `SKIP` or `AVERAGE` commands.

Examples

```
TIME 10000
```

Set sampling speed to 10 milliseconds.

```
TIME 7.5
```

Set sampling speed to 7.5 microseconds

9. Sample Applications

The Applications Manual contains many sample applications for use with Data Acquisition Processors. However, the DAP 3400a uses a different input configuration, which requires a new syntax for a few DAPL commands. Documentation for the modified commands can be found in Chapter 8.

The most significant change to the DAPL syntax for the DAP 3400a is how inputs are configured. Also, be aware that since the DAP 3400a samples in groups of four, data may need to be sent to the host PC differently. For example, if you are concerned with the inputs of only two channels, you do not need to send all four sampled channels to the PC.

In this chapter, two applications from the Applications Manual have been rewritten for the DAP 3400a. Following the syntax changes that were made in these examples, many examples in the Applications Manual can be modified for use with the DAP 3400a. Please note that analog output applications and digital input and output applications will not work with the DAP 3400a.

Example 1 -- Sampling Inputs Sequentially

This application configures the DAP 3400a to sample three input signals sequentially and send the digitized values to the PC. Since the DAP 3400a always samples input pins in groups, this application is different than for other Data Acquisition Processors in that certain input pins must be sampled (one from each group).

This application is similar to Application 1 in the Applications Manual. The `IDEFINE`, `SET`, and `COPY` commands have been changed for the DAP 3400a. The same single-ended input pins are used, but the differential input has been changed to single-ended input `S8`.

The following DAPL commands configure the Data Acquisition Processor for this application. Indentation is optional since DAPL ignores extra spaces.

```
RESET
 IDEFINE A 3
 SET IPIPES(0..3)  SPG2 ;SPG2 includes S2 connected to IP0
 SET IPIPES(4..7)  SPG1 ;SPG1 includes S5 connected to IP5
 SET IPIPES(8..11) SPG0 ;SPG0 includes S8 connected to IP10
 TIME 100               ;100 us between each group
 END
PDEF B
 COPY(IPIPES(0,5,10),$BINOUT)
 END
```

The `RESET` command on the first line clears all definitions and errors. It is a good idea to start each application with a `RESET`.

The next line begins an input procedure definition. An input procedure definition starts with the word `IDEFINE` and ends with the word `END`. `IDEFINE` usually is abbreviated to `IDEF`. The `IDEFINE` command requires the name of the input procedure and the number of input channel groups read by the input procedure. "`A`" is the name chosen for the input procedure in this application.

The line `IDEFINE A 3` configures the Data Acquisition Processor to sample three input channel groups, or 12 single-ended input pins. The `SET` commands associate input channel groups with pin groups. Since inputs are connected to `S2`, `S5`, and `S8`, pin groups `SPG2`, `SPG1`, and `SPG0` must be used. The three `SET` commands used connect input channel 0 (`IP0`) to single-ended input `S2`, `IP5` to single-ended input `S5`, and `IP10` to single-ended input `S8`.

The TIME command sets the sampling time to 100 microseconds. Since the input configuration samples three pin groups, each group is sampled every 300 microseconds.

END marks the end of the input procedure definition.

The word PDEFINE begins a processing procedure definition. PDEFINE is usually abbreviated to PDEF. The PDEFINE command is followed by the name of the processing procedure, which is B in this application. You are free to choose other names for procedures in your applications.

The COPY command transfers binary data from input channels 0, 5, and 10 to the binary communications pipe $BINOUT. The Data Acquisition Processor transfers binary data in $BINOUT directly to the PC. The COPY task continues until sampling is stopped.

END marks the end of the processing procedure definition.

Data collection begins when a START command is issued:

```
START A, B
```

The COPY task transfers data values from each input channel pipe in order. The host program running on the PC must know the number of data channels sent from the Data Acquisition Processor in order to correctly display the data. Some programs automatically determine the number of data channels by examining the DAPL command file. DAPview, must be user-configured.

To stop sampling, issue a STOP command. This command stops the input procedure and the processing procedure. Analog sampling is stopped and the COPY task is halted. The application can be restarted by reissuing the START command.

This application can be simplified if the inputs are sampled simultaneously. To do this, all inputs must be from one pin group. In the following example pin group SPG0 is used, which contains pins S0, S4, and, S8, and S12. Samples from S0, S4, and S8 are sent to the PC with the COPY command:

```
RESET
  IDEF A 1
  SET(0..3)  SPG0
  TIME 300
  END
PDEF B
  COPY(IPIPES(0..2),$BINOUT)
  END
```

Note that `TIME` was adjusted to make the data rate the same as with the first example.

Example 2 -- True Simultaneous Sampling

Some applications require sampling two or more analog inputs simultaneously. The DAP 3400a can sample up to four channels simultaneously. In this application, the DAP 3400a samples single-ended inputs `S1`, `S5`, `S9`, and `S13` simultaneously, with all four channels being sampled every 1000 microseconds.

This application is similar to Application 33 from the Applications Manual.

```
RESET
IDEF A 1
  SET IPIPES(0..3),SPG1
  TIME 1000
END
```

The term `IPIPES` is optional for the `SET` command.

All four channels defined in the `SET` command will be sampled. Extra channels can be ignored if you need the data from only two or three channels.

Example 3 -- Simulated Simultaneous Sampling

This application uses `FIRFILTER` commands to perform a time-shifting interpolation on 16 channels of raw data.

The DAP 3400a samples data in groups of four channels. If all 16 analog inputs are sampled, the normal sampling sequence is to sample four channels, then, one interval later, sample the next four channels, and so on.

Sometimes it is necessary to get a "snapshot" of data, or to look at what is happening to all input data channels at one instant. This is known as simultaneous sampling. The DAP 3400a hardware is designed for simultaneous sampling of up to four input channels. If more than four channels are involved, a software correction to simulate simultaneous sampling is possible using the DSP features of the DAP 3400a hardware and software. In effect, the correction algorithms construct local approximating curves for each data sequence, and use the approximating curves to estimate the sample values that would have been recorded if all of the signals had been sampled at the same time. (Technically speaking, the approximating curves used in this application are the least-squares best fit of a cubic polynomial model.)

The approximation process has some drawbacks for the case of very high frequency signals because the approximation tends to suppress very rapid local changes in the data. However, if the rapid local changes are due to high frequency noise, the noise-suppression side effect is beneficial.

To create a software simulation of simultaneous sampling for sixteen channels, samples of all four channels groups need to be taken in a repeating sequence.

The first four channels do not need to be corrected, but this application applies filtering so that all 16 channels experience the same high-frequency noise reduction. The interpolation approximates the data in last twelve channels for the time at which the first channel group was sampled by “time-shifting” the data in the second, third and fourth channel groups by 1, 2 and 3 sampling intervals, respectively.

The following DAPL commands configure the DAP 3400a for this application:

```
RESET
VECTOR SHIFT000 = (-3121,4681,9362,10923,9362,4681,-3121)
VECTOR SHIFT025 = (-3700,6845,11160,10825,7418,2517,-2298)
VECTOR SHIFT050 = (-3950,8923,12727,10533,5413,439,-1317)
VECTOR SHIFT075 = (-3785,10831,13977,10045,3431,-1469,-262)
PIPES  P0, P1, P2, P3, P4, P5, P6, P7, P8, P9,
PIPES  P10, P11, P12, P13, P14, P15
IDEF  SAMP 4
  SET IPIPES(0..3)   SPG0
  SET IPIPES(4..7)   SPG1
  SET IPIPES(8..11)  SPG2
  SET IPIPES(12..15) SPG3
  TIME  20.00
  END
PDEF  FILT
  FIRFILTER( IP0,  SHIFT000, 7, 1, 0, 0, P0 )
  FIRFILTER( IP1,  SHIFT000, 7, 1, 0, 0, P1 )
  FIRFILTER( IP2,  SHIFT000, 7, 1, 0, 0, P2 )
  FIRFILTER( IP3,  SHIFT000, 7, 1, 0, 0, P3 )
  FIRFILTER( IP4,  SHIFT025, 7, 1, 0, 0, P4 )
  FIRFILTER( IP5,  SHIFT025, 7, 1, 0, 0, P5 )
  FIRFILTER( IP6,  SHIFT025, 7, 1, 0, 0, P6 )
  FIRFILTER( IP7,  SHIFT025, 7, 1, 0, 0, P7 )
  FIRFILTER( IP8,  SHIFT050, 7, 1, 0, 0, P8 )
  FIRFILTER( IP9,  SHIFT050, 7, 1, 0, 0, P9 )
  FIRFILTER( IP10, SHIFT050, 7, 1, 0, 0, P10 )
  FIRFILTER( IP11, SHIFT050, 7, 1, 0, 0, P11 )
  FIRFILTER( IP12, SHIFT075, 7, 1, 0, 0, P12 )
  FIRFILTER( IP13, SHIFT075, 7, 1, 0, 0, P13 )
  FIRFILTER( IP14, SHIFT075, 7, 1, 0, 0, P14 )
  FIRFILTER( IP15, SHIFT075, 7, 1, 0, 0, P15 )
  MERGE( P0,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,\
         P11,P12,P13,P14,P15,$BINOUT )
  END
START SAMP,FILT
```

The `RESET` command on the first line clears all definitions and errors.

The four `VECTOR` commands define data for four filters. `SHIFT000` is the filter data used for the first channel group, `SHIFT025` is the filter data used for the second channel group, `SHIFT050` is the filter data used for the third channel group, and `SHIFT075` is the filter data used for the fourth channel group.

The PIPES commands define 16 data pipes for use by the FIRFILTER commands.

The input procedure SAMP configures input channel pipes 0-15 with the four input pin groups SPG0-SPG3. This associates each analog input pin with one input channel pipe.

The TIME command sets the sampling interval to 20 microseconds. Since data is being time-corrected for four channel groups, the results will be an approximation for simultaneous sampling every 80 microseconds.

The processing procedure FILT is where the interpolation of data is configured. Each FIRFILTER command receives data from one input channel. Depending on which group the channel was sampled in, one of the four filter vectors is applied to the data. After the data is filtered, each is sent to a data pipe.

The MERGE command sends all filtered data to the PC through the binary communications pipe $BINOUT.

END marks the end of the processing procedure.

START begins processing of the input procedure SAMP and the processing procedure FILT. Sampling can be stopped by issuing a STOP command.

10. Appendix A: Analog Expansion Pin Mapping

Expansion board MSXB 018 has four connectors for expansion. Each connector has 16 inputs numbered `S0` to `S15`, for a total of 64 analog inputs per board. The DAP 3400a can use up to eight analog input expansion boards, for a total of 512 analog inputs, or 128 pin groups. This Appendix documents how the expansion board pin numbers map to pin group numbers used in the DAPL `SET` commands.

For the DAP 3400a, all channel groups in an input procedure must be explicitly configured.

Microstar Laboratories strongly suggests that if you are using analog expansion, use `OPTION AINEXPAND=ON`. This option maps the pins of the Analog Input Expansion Board to match the order on the Data Acquisition Processor analog connector.

With `AINEXPAND ON`, the mapping for one expansion board will be:

Pin Group	Analog Expansion Pin Numbers
SPG0	S0 on each termination board
SPG1	S1 on each termination board
SPG2	S2 on each termination board
SPG3	S3 on each termination board
SPG4	S4 on each termination board
SPG5	S5 on each termination board
SPG6	S6 on each termination board
SPG7	S7 on each termination board
SPG8	S8 on each termination board
SPG9	S9 on each termination board
SPG10	S10 on each termination board
SPG11	S11 on each termination board
SPG12	S12 on each termination board
SPG13	S13 on each termination board
SPG14	S14 on each termination board
SPG15	S15 on each termination board

For a second expansion board, `SPG16` maps to `S0` on each termination board `SPG17` maps to `S1`, etc. Because of the sequential numbering of pin groups, it is important to keep track of expansion board and termination board sequencing.

Within a channel group, the four channels are connected to the four corresponding pins on connectors `J5`, `J4`, `J3` and `J2` in that order. For example, if the channel group `IPIPES(0..3)` is connected to the expanded `SPG0`, then `IP0` is connected to `S0` on connector `J5` of the expansion board, `IP1` to `S0` on connector `J4`, `IP2` to `S0` on connector `J3`, and `IP3` to `S0` on connector `J2`. The following table shows how DAPL pin numbers map to analog expansion pins and termination board inputs.

DAP 3400a Expansion Pin Mapping

(OPTION AINEXPAND=ON)

DAPL Pin	Connector Number	Connector Pin	Termination Board Label
S0	J5	53	S0
S1	J5	52	S1
S2	J5	51	S2
S3	J5	50	S3
S4	J4	53	S0
S5	J4	52	S1
S6	J4	51	S2
S7	J4	50	S3
S8	J3	53	S0
S9	J3	52	S1
S10	J3	51	S2
S11	J3	50	S3
S12	J2	53	S0
S13	J2	52	S1
S14	J2	51	S2
S15	J2	50	S3
S16	J5	49	S4
S17	J5	48	S5
S18	J5	47	S6
S19	J5	46	S7
S20	J4	49	S4
S21	J4	48	S5
S22	J4	47	S6
S23	J4	46	S7
S24	J3	49	S4
S25	J3	48	S5
S26	J3	47	S6
S27	J3	46	S7
S28	J2	49	S4
S29	J2	48	S5
S30	J2	47	S6
S31	J2	46	S7

DAP 3400a Expansion Pin Mapping, continued

(OPTION AINEXPAND=ON)

DAPL Pin	Connector Number	Connector Pin	Termination Board Label
S32	J5	45	S8
S33	J5	44	S9
S34	J5	43	S10
S35	J5	42	S11
S36	J4	45	S8
S37	J4	44	S9
S38	J4	43	S10
S39	J4	42	S11
S40	J3	45	S8
S41	J3	44	S9
S42	J3	43	S10
S43	J3	42	S11
S44	J2	45	S8
S45	J2	44	S9
S46	J2	43	S10
S47	J2	42	S11
S48	J5	41	S12
S49	J5	40	S13
S50	J5	39	S14
S51	J5	38	S15
S52	J4	41	S12
S53	J4	40	S13
S54	J4	39	S14
S55	J4	38	S15
S56	J3	41	S12
S57	J3	40	S13
S58	J3	39	S14
S59	J3	38	S15
S60	J2	41	S12
S61	J2	40	S13
S62	J2	39	S14
S63	J2	38	S15

With `OPTION AINEXPAND=OFF`, the input pin mappings are as shown below:

Pin Group	Analog Expansion Pin Numbers
SPG0	S0 on each termination board
SPG1	S1 on each termination board
SPG2	S8 on each termination board
SPG3	S9 on each termination board
SPG4	S2 on each termination board
SPG5	S3 on each termination board
SPG6	S10 on each termination board
SPG7	S11 on each termination board
SPG8	S4 on each termination board
SPG9	S5 on each termination board
SPG10	S12 on each termination board
SPG11	S13 on each termination board
SPG12	S6 on each termination board
SPG13	S7 on each termination board
SPG14	S14 on each termination board
SPG15	S15 on each termination board

11. Appendix B: Support Software Modification

Earlier in this manual, changes to the DAPL 2000 operating system were documented. Modifications made to other support software are documented in this appendix.

Developer's Toolkit for DAPL

A new option for the Developer's Toolkit for DAPL `sys_get_info()` function is provided in DAPL to return the size of an input channel or pin group in a particular Data Acquisition Processor model. The option ID is `GI_INPUT_GROUP_SIZE` with a value of 33. With this option and the option `GI_ICHAN_CNT`, a custom command application is able to determine the number of input groups in an input procedure.

See the Developer's Toolkit for DAPL Manual for more information about writing custom commands.

12. Glossary

The terms defined in this Glossary are specifically for the DAP 3400a. More terms that apply to DAPL in general are defined in the DAPL Manual.

`a-Series`
: "a-series" refers to Data Acquisition Processor models that use the letter 'a' in the model name. A Data Acquisition Processor that is an "a series" board uses a low-noise 50-pin (DAP 800) or 68-pin analog connector.

`Analog Input Pin`
: An analog input is a single hardware input pin. For a DAP 3400a, all analog inputs are single-ended, and all pins must be referred to in groups (`SPGx`).

`Channel Group`
: A channel group is a group of four input channel pipes associated with an input pin group. All four input pins in an input pin group are sampled simultaneously and are transferred to the four input channel pipes in the channel group in a predefined order.

  For a DAP 3400a, a channel group must be specified by a range of four consecutive integers beginning with a multiple of 4. See Chapter 8 for more information.

`Input Channel Pipe`
: An input channel pipe is a special pipe into which the DAP 3400a hardware places analog conversion values.

  On the DAP 3400a, there must be a one-to-one relationship between analog inputs and input channel pipes since associations between input channel pipes and input pins are set by hardware; however, not all input channel pipes have to be used by DAPL tasks. Also, any number of tasks can read data from the same input channel pipe.

`Input Pin Group`
: An input pin group is a group of four analog input pins that are sampled simultaneously. Each pin in a group is connected to a separate analog-to-digital converter. The grouping of input pins is set by hardware.

A single-ended pin group is represented in the DAPL `SET` command by `SPGx`, where `x` is the number of the pin group. See Chapter 8 for more information about pin group numbers.

`Pipe`

A pipe is a first-in-first-out buffer for temporary storage of data. Data are added to one end of a pipe and removed from the other end. If data are added to a pipe faster than they are removed, the size of the pipe increases. Storage space is allocated and released automatically, allowing a pipe to grow or shrink as required. A pipe also has a user-defined maximum size and does not expand indefinitely. Data always leave the pipe in the order in which they enter it. A pipe can be defined to hold 8-bit, 16-bit, or 32-bit data.

Index