xDAP 7410 Manual

Connector and Hardware Operation Reference

Version 1.00

Microstar Laboratories, Inc.

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

Part Number xDAP7410M100

Contents

1. Introduction

The xDAP 7410 from Microstar Laboratories is a high-performance data acquisition system for intensive high-speed data capture on multiple channels. It features multiple converter channels for parallel, synchronized data capture. Typical applications are high-speed data logging and recording of fast transient events. The xDAP 7410 is independently packaged, but uses a PC host to configure and operate it via a standard USB 2.0 cable.

The xDAP 7410 provides 8 hardware converter devices that can provide data to 8 data selector channels simultaneously. When multiple selector channels are used in a parallel fashion, they are clocked simultaneously, making the xDAP 7410 suitable for precision measurement of very fast signals, without time or phase skew between signal channels. Each selector channel can provide one million samples per second. With all 8 of these channels operated in parallel, this means a composite rate of 8 million samples per second. This maximum rate can be sustained continuously across the USB connection to the host – provided that your host system is up to the challenge.

A brief summary of the xDAP 7410:

- 2.0 GHz Intel Celeron embedded processor
- 1 Gbytes of active DDR3 memory
- USB 2.0 interconnection to the host
- Compatible with portable laptop systems
- Eight 16-bit A/D converters
- Up to 16 configurable signal sources
- 20 ns TIME resolution
- 7 configurable bipolar input voltage ranges
- All channels true differential
- 1 million samples per second per converter channel
- optional 1 or 2 panels with 8 BNC differential input channels each
- 8M samples per second aggregate
- Sustained host transfers at maximum data capture rates
- High repeatability with excellent SNR performance
- 16 bits digital inputs and 16 bits digital outputs for control I/O

The xDAP 7410 is operated by the embedded DAPL 3000 system The DAPL system provides powerful and automatic control of memory resources, configurable data pre-processing, and data transfer activity. Its multi-level prioritized task scheduling makes it possible to have both intensive data streaming and responsive control at the same time.

About This Manual

This manual discusses hardware connections, device configuration, and device operation.

- Connectors, both for signals and the host interface.
- The electrical characteristics of the input sampling circuit.
- Instructions for configuring the sampling channels.
- Using software-controlled triggering for semi-autonomous operation requiring no host interaction.

Some other manuals that provide further information about installation and processing software configurations:

- The DAP USB Installation Manual describes host system requirements and the process of hardware and software installation.
- The DAPL 3000 Manual provides complete details about all DAPL system configuration commands.
- The DAPstudio Manual explains how to configure applications.

2. xDAP 7410 Panel Connections

The xDAP 7410 has four external connections and no accessible internal connections.

Front Panel

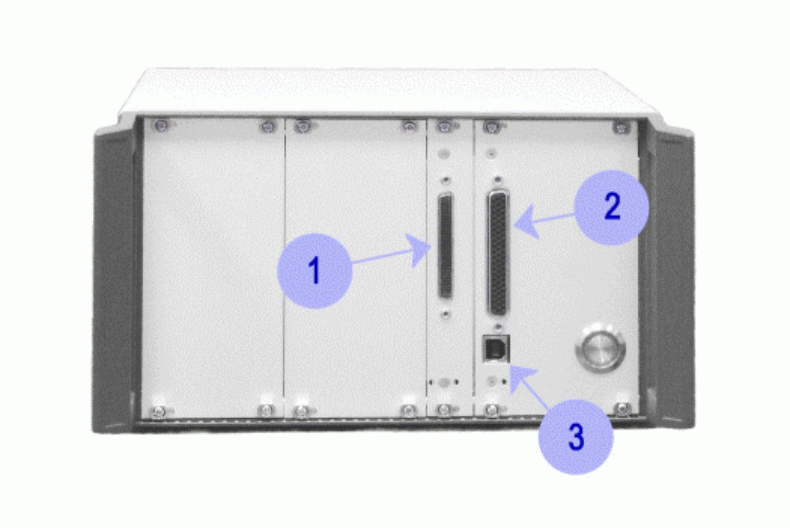

Figure 1. Front panel connections.

1. Analog differential input connector for 16 parallel differential input signals
2. Digital input connector for 16 asynchronous digital inputs and 16 asynchronous digital outputs
3. USB 2.0 connection to the host

Rear Panel

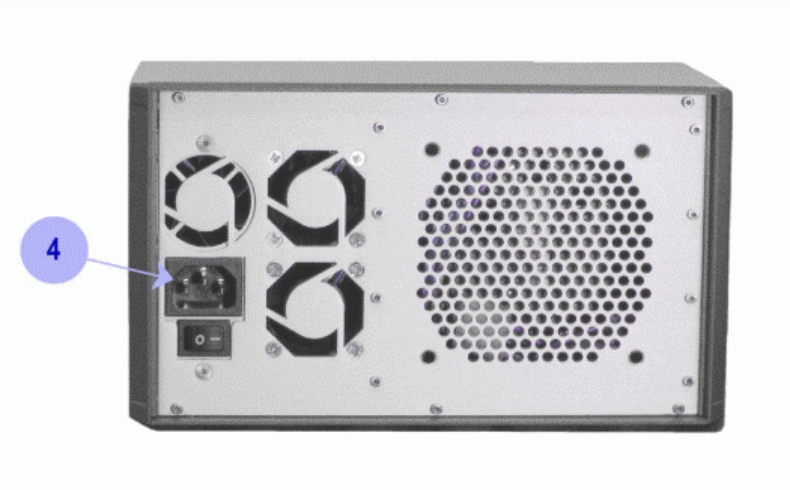

Figure 2. Rear panel connections.

4. Type IEC 60320 C14 AC Power connector

Analog Input Connector

Analog voltages are connected to the Data Acquisition Processor through a 68-pin connector on the front panel. This connector has a double row of pins on 0.050-inch centers. This connector is Tyco Electronics part number 5787170-7 or equivalent. For round cable, it mates with shielded discrete wire connector Tyco Electronics 1-5750913-7 or equivalent. For 0.025" pitch ribbon cable, the connector also mates with insulation displacement ribbon cable connector Tyco Electronics part number 5786090-7 or equivalent. Any standard cable compatible with SCSI 3 can connect to an xDAP, but signal quality may be degraded for some cables. MSCBL125-01-L70 from Microstar Laboratories is a standard 68-line SCSI 3 cable qualified by Microstar Laboratories for high analog signal quality.

Looking at the connector socket on the front panel, the pin numbering is as illustrated in the following diagram.

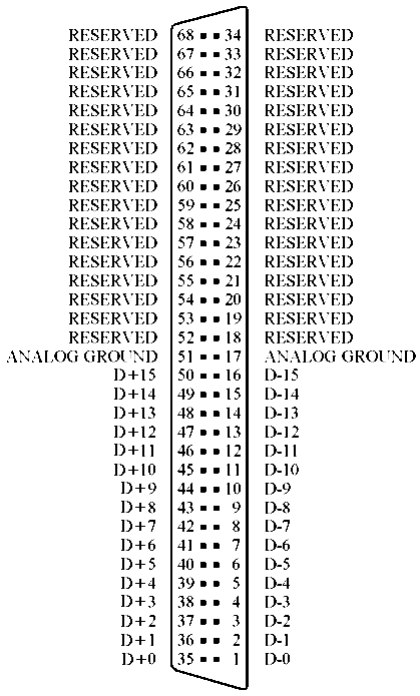

Figure 3. Analog Input Connector

The Analog Ground pins share a common fuse. The fuse is for purposes of fault protection only. Anything connected to analog ground pins could directly influence measurement quality on all signal channels. Applications requiring a reference ground level can, with care, use one of the analog ground pins for this purpose. Avoid direct connections between the analog signal grounds and other system grounds such as AC power grounds, as these connections provide ground loop paths through which high levels of noise can propagate.

Note: The pin numbering shown on this diagram is consistent with conventions for SCSI devices, which are often used with this style of connector. The pin numbering should be consistent from various manufacturers.

The positive differential input pins are labeled `D+0` through `D+15`; the corresponding negative inputs of the differential pairs are labeled `D-0` through `D-15`. Do not connect to the pins marked as *reserved* on this diagram.

Termination boards that connect all lines of the analog connector to discrete wires are available from Microstar Laboratories. These can rest on a tabletop, or they can be mounted on a standard DIN rail.

To avoid damage, analog input signals should never exceed the range from -35 volts to +35 volts, relative to the ground of the Data Acquisition Processor. Measurable input signals must be within the range –13 volts to +13 volts. Within the safe operating range, only signal pairs within the configured full differential range will yield useful measurements. 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.

Full details about input channel properties are provided in the xDAP 7410 Specifications. You can find a copy on the http://www.mstarlabs.com/ web site.

USB Communications Connector

The USB connector on the xDAP front panel requires a high-quality, generic USB 2.0 cable with a *type-B* (beveled square) plug. For compatibility with the host PC, the other end of the cable must have a standard *type-A* rectangular plug. This is a common cable format used with many high-speed PC peripheral devices such as printers and external disk drives.

What constitutes a "high-quality" cable is tricky to specify and extremely difficult to verify. The USB cable that Microstar Laboratories provides with each xDAP 7410 is qualified to work well with xDAP devices. If you use other cables, be aware that seemingly identical cables from different manufacturers can give different results, with no clear correspondence to price and brand name. If you are not successful at getting the expected transfer rates, try a cable from a different manufacturer.

Digital Input/Output Connector

The xDAP 7410 provides one HD-62 connector, a receptacle with 16 digital input lines and 16 digital output lines. The connector is Tyco Electronics part number `5748394-5` or equivalent. It mates with L-com part number `SDH62P` or equivalent. Connectors can be built from Tyco Electronics connector shell `1658673-1` or equivalent. The connector also mates with cables `MSCBL126-01-L56` or cable kit `MSCBL109-01K` from Microstar Laboratories. Looking at the front panel, the following is the pin diagram.

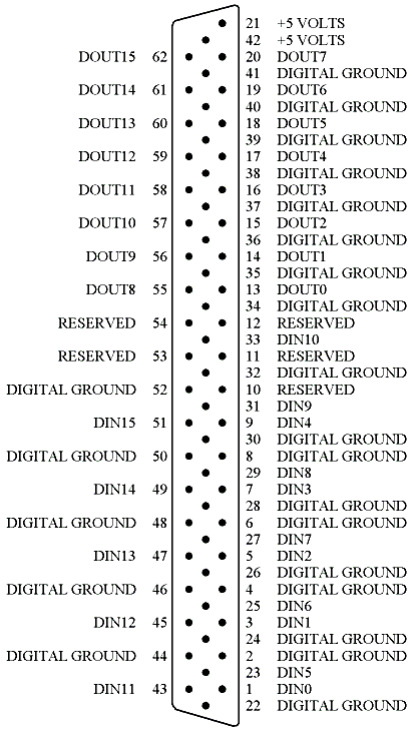

Figure 4. Digital Input/Output Connector

The digital input and digital output lines are TTL compatible and terminated with a 10K pull-down resistor to ground. Don't make connections to the lines that are labeled *reserved* in the pin diagram. The two +5V supply pins share a common fuse. The *Digital Ground* pins all share a common fuse. Fuses are for purposes of fault protection only. Digital ground connections are intended as return paths for digital logic only, and should never be used for other purposes. For details about input channel electrical properties, see the xDAP 7410 Specifications. You can find a copy on the http://www.mstarlabs.com/ web site.

These digital input and output ports are for purposes of control signaling, not data logging, so they are not defined within a sampling configuration. Special processing commands operate the digital ports. On the xDAP 7410, both ports are located at port address 0.

3. Analog Input Circuits

All signals received on the analog input connector are differential. That is, each signal is actually a pair of signals, and the measured voltage is the difference between the positive input terminal and the negative input terminal. Differential inputs work best with true differential signal sources, but in practice many signals consist of a single-ended voltage with respect to a local reference ground. It is completely acceptable to connect the positive differential input to the measured voltage and the negative differential input to the signal's reference ground.

In sampling configurations defined for the DAPL system, a differential signal pair is referenced using a single pin-notation, for example the following:

```
SET  IPIPE0  D14
```

In this example, the differential signal selected for measurements is the voltage difference between the `D+14` and the `D-14` signal pins.

Voltage Ranges

While the analog range is fixed, the usable differential voltage range (within that analog range) is configurable. The input voltage ranges are sometimes described by a "gain" – if you take the selected full range voltage times the gain setting, this would produce the same results as applying the gain to the original differential signal and then measuring on the maximum differential input range of –10V to +10V.

Differential Input Range	**Equivalent Gain**
-10 V to +10 V	1
-5 V to +5 V	2
-2 V to +2 V	5
-1 V to +1 V	10
-0.5 V to +0.5 V	20
-0.2 V to +-0.2 V	50
-0.1 V to + -0.1 V	100

Selecting a more limited voltage range does not limit the precision of measurements. Since the xDAP 7410 has 16-bit analog-to-digital conversion devices, the selected full input range is divided into 65536 discrete levels. At the *gain 1* setting, this means that the increment of voltage resolution is 305.2 microvolts. At the *gain 100* setting, the increment of voltage resolution is 3.052 microvolts. To select a voltage range other than the `-10 V` to `+10 V` default, specify the equivalent gain on the `SET` command that configures the input channel. For example, the following specification selects the –0.5 V to +0.5 input range.

```
SET  IPIPE0  D14  20
```

The effects of reducing the input range approximately balance the loss of amplifier bandwidth when operating at a higher gain setting. As a result, there is little degradation in response speed for operating at higher gains. At all gain levels, due to slewing limitations of amplifiers, the input electronics will not track abrupt signal changes perfectly. In most cases, distributed capacitance and impedance of connected cables will limit tracking performance more than the slewing limitation.

Equivalent Circuit

Each analog signal path, from the input pins to the analog-to-digital converter, consists of the following stages:

1. Input multiplexer

2. Instrumentation amplifier
3. Range amplifier
4. Analog-to-digital converter with integrated sample-and-hold amplifier

Figure 5 shows an equivalent circuit for each Data Acquisition Processor input.

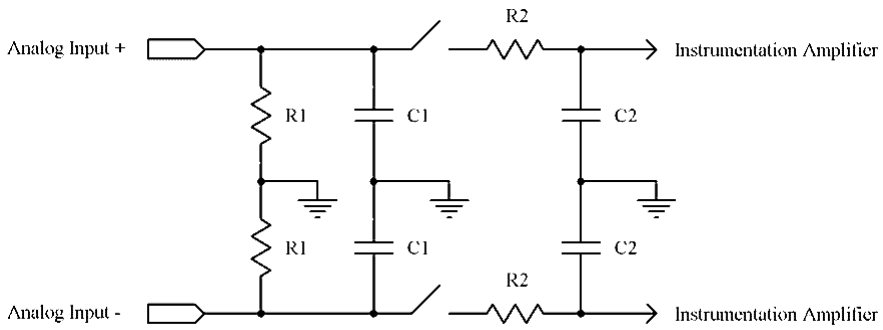

Figure 5. Equivalent input circuit

Component values for this circuit are as follows.

Component	Value
R1	10 M Ohm
C1	48 pF
R2	180 Ohm
C2	6 pF

The DC input impedance at DC is high, typically 20M Ohms. The AC input impedance at high frequencies is dominated by the capacitance of the input multiplexer, 48 pF typical. To avoid attenuated signals, the signal source impedance should be low. That means cables should be short, and driving sources should have low impedance. Long cables increase the input capacitance while at the same time increasing the signal source impedance, increasing the settling time of the input section.

As the Data Acquisition Processor collects measurements from the input pins defined by an input procedure, the multiplexer switches open and close, connecting the selected inputs to the conversion amplifiers. Unless the signal source impedances are very low, the capacitance of the input multiplexers does not get fully charged during very short sampling intervals, and this reduces measurement accuracy.

Most of the charging delays are eliminated in a *simultaneous sampling* configuration, in which each channel selector connects to a single input signal line and remains connected to that line for every scan cycle. This yields the best settling performance, but limits the number of input signal lines that can be used.

4. Channel Architecture

The xDAP 7410 features a new channel selector architecture. Users of the older DAP board families should review Appendix A for important information about how xDAP devices are different from previous DAP models.

An xDAP uses multiple channel selectors to fetch input data for data channels. All configured channel selectors will capture data simultaneously, for as many channel selectors as you have configured. Channel selectors are an architectural feature, implemented cooperatively by a number of actual hardware devices – they do not correspond to a single device.

The xDAP 7410 supports 8 channel selectors for its input sampling. However, it supports 16 total differential input channels. Clearly, with 16 signal sources and 8 selectors, you can't capture all of the channels at once. The DAPL 3000 system will schedule the sampling for each channel that you specify.

For example, the following channel configuration defines four channels that connect to separate channel selectors, so they all can be sampled simultaneously. There is one sampling cycle during the scan interval of 10 microseconds.

```
CHANNELS  4
SET  IP0   D0
SET  IP1   D2
SET  IP2   D4
SET  IP3   D6
SCAN  10.0                // cycle completed in 10 microseconds
```

Previous DAP models had the equivalent of only one selector channel, so a similar channel list configured using the TIME command would require 40 microseconds to capture all four channels in the channel list.

```
CHANNELS 4
SET  IP0   D0
SET  IP1   D1
SET  IP2   D2
SET  IP3   D3
TIME  10.0         // cycle completed in 40 microseconds
```

The eight channel selectors for the xDAP 7410 connect to the following input signal pins.

Input selector channel	Connected pins
0	D0, D1
1	D2, D3
2	D4, D5
3	D6, D7

4	D8, D9
5	D10, D11
6	D12, D13
7	D14, D15

5. Digital Ports

The xDAP 7410 provides one port with 16 digital input lines and one port with 16 digital output lines. These ports are for purposes of control signaling, not data logging, so they are not defined in the sampling configuration. Special processing commands give you the ability to operate these digital ports. The port address is `0` for both the digital input and the digital output port on the xDAP 7410

The hardware clock that controls sampling does not drive the digital ports. Among the advantages, the digital ports do not occupy any channel selector positions and do not interfere with the timing on high-speed data channels. Among the disadvantages, the timing is dependent on the task scheduler, so very high priority system activity can produce a varying latency for detecting and responding to events.

The status of the digital input port is observed using the `DIGITALIN` processing command. The timing interval between observations is specified as the last parameter of this command. To test the digital port 20 times per millisecond, you would use the following command form, specifying a timing delay of 50 microseconds between observations.

```
DIGITALIN( MyDigPipe, 0, 50 )
```

In a similar manner, processing can send signals to the output digital port lines using the `DIGITALOUT` processing command.

```
DIGITALOUT( MyDigOutput, 0 )
```

For outputs, the time interval is usually not needed and is omitted, as above. When new data arrive to go to the digital output port, the transfer is done as soon as possible.

There will be more examples later in this manual.

6. Software Triggering

Software triggering is more versatile than traditional hardware triggering, though a little less precise. It allows an application to analyze a data stream and select data on the basis of what is observed there.

With the software triggering strategy, data are captured and discarded continuously, until something is detected by the software triggering. Then, the samples of relevance are extracted from the data stream for whatever further processing or logging you intend. The xDAP 7410 can perform triggering analysis while capturing data through all channel selectors at maximum rates.

When a processing task identifies an event in a sampled data stream, or on the digital control port, it asserts a software trigger. This allows other tasks to act, retaining and processing data. This processing strategy is illustrated in application example 2 in *Chapter 8*.

7. Latency

Latency is the real-time delay between the occurrence of an event and the issue of a response to that event. Included within this latency interval is all of the time required to detect, capture, process, and respond to the event. Latency is not a matter of concern for most applications. For example, most so-called “real-time” data displays are not real-time — they run at the same data rates as real-time, but the displayed values appear only after a considerable delay. For applications such as high-speed control or safety monitoring, delays can be critical.

There are three factors that determine xDAP 7410 latency.

- The observation rate. The xDAP cannot respond to an event until the event is captured. This causes a variable latency that can be as large as one observation interval. If a sampled data channel provides the data, the latency can be as long as one sampling scan interval.
- Internal hardware and data management processing must move the data to a task for processing. This adds approximately 8 microseconds fixed delay.
- Once the data become available, scheduling of the task to generate the output response is unpredictable. Higher-priority processing can occasionally interfere with the scheduling of the output task and result in longer delays. In a well-designed application, this randomly distributed latency can be up to about 12 microseconds.

As a general rule, for a sampling time interval of 20 microseconds or longer, a well-configured application will be able to complete delivery of every computed result before the next input sample arrives, and a worst case bound on the response delay will be about 40 microseconds.

8. Example Applications

This chapter will present two typical applications of the xDAP 7410, one for high-channel-count logging applications, and one for high-speed triggered data capture.

Example 1 — Logging maximum channels

The xDAP 7410 supports 16 analog input lines, and all 16 of these are defined in the input sampling configuration for this application. The signals are somewhat noisy, so the data channels are captured at higher speed, and the data filtered by averaging in blocks of 40 samples.

Differential channels `D0` through `D14`, the even-numbered channels, are connected to 8 separate selector channels, so these 8 will be sampled simultaneously. Channels `D1` through `D15`, the odd-numbered channels, will be similarly sampled simultaneously. The complete scan of all 16 channels, requiring 2 sample operations for each selection channel, is configured to take 25 microseconds. The following input sampling configuration named `capture` defines the clocked data capture processing. (For purposes of illustration, channels are grouped according to the sampling order.)

```
RESET

IDEFINE capture
    CHANNELS  16
    SET IPIPE0   D0     // First sampling operation
    SET IPIPE1   D2
    SET IPIPE2   D4
    SET IPIPE3   D6
    SET IPIPE4   D8
    SET IPIPE5   D10
    SET IPIPE6   D12
    SET IPIPE7   D14
    SET IPIPE8   D1     // Second sampling operation
    SET IPIPE9   D3
    SET IPIPE10  D5
    SET IPIPE11  D7
    SET IPIPE12  D9
    SET IPIPE13  D11
    SET IPIPE14  D13
    SET IPIPE15  D15
    SCAN  25.0
END
```

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

The `CHANNELS` command tells the configuration how many samples must be captured to produce a value for each configured channel. In this case, every signal is listed once and provides one value to the group of 16.

The `SCAN` command tells the configuration how much time is allowed to capture all of the samples for one pass. Because each channel selector is connected to two of the channels in the list, there will be two sampling operations within the sampling scan interval. Every logical channel pipe receives one value per scan.

The processing configuration receives the sampled data in the form of a stream, with one value from each channel, in groups of 16 channels. It could process these 16 channels independently, but it can also take advantage of the parallel nature of the processing to perform the averaging for all 16 channels as a group. Averaging 40 samples to produce one average value reduces the sample rate from 25 microseconds per sample to 1 millisecond per sample in each channel.

```
PDEFINE reduce
    BAVERAGE (IPIPES(0..15), 16, 40, $BinOut)
END
```

The BAVERAGE command is configured to take the data from all 16 input sampling channels, process the data in groups of 16, average 40 values from each of the input channels, and deliver the results as groups of 16 samples. The results are streamed directly to the host PC via the $BinOut communication pipe, which connects through the USB cable.

The whole configuration begins execution when given the START command.

```
START
```

To stop sampling, issue a STOP command. This command stops the input procedure and the processing procedure. Analog sampling is stopped and no new data transfers are initiated. The application configuration is preserved and can be run again by reissuing the START command.

Grouping the channels according to the sampling sequence of channel selectors was only for purposes of illustration. Usually, it will be easier for you (and also for the DAPL system) to specify the channels in a natural sequence:

```
IDEFINE capture
    CHANNELS  16
    SET IPIPE0   D0     // Let DAPL take care of sampling order
    SET IPIPE1   D1
    SET IPIPE2   D2
    . . .
    SET IPIPE14  D14
    SET IPIPE15  D15
    SCAN  25.0
END
```

Example 2 — Simultaneous Sampling

This application performs a destructive impact test on a part placed into a fixture. High-speed strain sensors are mounted on the part, two of them placed near the two ends of the part, one at the center of the part near the point of impact, and two at intermediate locations. This gives a total of 5 measurement channels. The entire event spans a few milliseconds. The data are captured at near maximum rates to allow some redundancy for noise reduction during the post-processing analysis. Even with high sampling rates, the simultaneous sampling is helpful to reduce the channel-to-channel time skew.

There is no point in recording data while nothing is happening. To determine whether anything is happening, one of the digital input port lines is connected to the release control button for the experiment.

To get five channels all sampled simultaneously, attach the sensors to analog signal pins connected through five different channel selectors. The sampling rate is set to 2 microsecond intervals for each channel. This will allow capturing 500 samples per millisecond on each channel.

```
RESET

IDEFINE  fastevent
  CHANNELS   5
  SET  IPIPE0     D0
  SET  IPIPE1     D2
  SET  IPIPE2     D4
  SET  IPIPE3     D6
  SET  IPIPE4     D8
  TIME  2.0
END
```

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

The input procedure definition named `fastevent`, as shown above, defines the connections to the five input signals. Since each configured channel receives data from a different input channel selector, all five channels operate in parallel and receive new data simultaneously. The remaining three channel selectors are not used in this configuration, and receive no data.

For triggering, connect digital input port 0 bit 0 to the user-provided test-initiation switch. When this bit goes active-high, that indicates the start of an event to be observed. Other digital port bits are pulled down to zero by the xDAP. In effect, if there is any nonzero bit on the digital input port, this indicates the start of an event.

First define a software trigger. There will be a single event, so we do not need multiple responses if there is a *switch contact bounce*. Since the sampling interval is 2 microseconds, 5000 sample intervals cover 10 milliseconds, which should be plenty of time to bypass the switch bounce. This interval is specified as a `HOLDOFF` property.

```
TRIGGER  tevent  MODE=NORMAL  HOLDOFF=5000
```

We will also reserve two pipes for capturing the bit values from the digital port.

```
PIPES    pbits WORD,   pnonzero WORD
```

Now define the processing, which is assigned the name `detect_retain`. The first part will detect events on the digital port, and the other part will respond by extracting the desired data from the five sensor channels.

```
PDEFINE  detect_retain
    …
```

The `DIGITALIN` command provides access to the digital input port values. Because this is not under control of the high-speed sampling hardware, it must be tested at a longer time interval to allow for task timing. A 40 microsecond testing interval allows 25 digital port tests per millisecond. The bit patterns are placed into pipe `pbits`. The `RANGE` command discards any values that are equal to zero. When the `PCASSERT` command sees the arrival of a nonzero value, it generates a trigger event at the current input stream location, which is determined by watching the input channel pipe `IPIPE0`.

```
    DIGITALIN (pbits, 0, 40)
    RANGE ( pbits,  OUTSIDE, 0, 0,  pnonzero)
    PCASSERT( pnonzero, tevent,  IPIPE0 )
    …
```

To select the appropriate input samples, use the `WAIT` command. The impact event is completed in 50 milliseconds, which is covered by 25000 samples in each channel, or 125000 samples total. To avoid missing any activity that might occur during the 40 microsecond intervals while the `DIGITALIN` command timer is waiting, and to allow for latency delays of up to 40 microseconds in the response of `DIGITALIN`, retain data from a cautious 100 microseconds prior to event detection. This time interval covers 50 sampling periods, or a total of 250 samples in all channels combined. The WAIT command collects the 250 pre-trigger samples plus the 125000 post-trigger samples and sends them all directly to the host for recording and later analysis via the `$BinOut` communication pipe and the USB 2 connection. This completes the processing definition.

```
    …
    WAIT ( IPIPE(0..4), tevent, 250, 125000, $BinOut )
END
```

The `START` command will activate the processing, but of course no data will be retained, and it will appear as if nothing were happening, until the test-initiation switch is pressed.

The configuration will be ready to repeat the measurements each time the test-initiation switch is pressed. When there will be no more measurements, discontinue operation by issuing a `STOP` command.

This application has the advantage that only one bit changes value. If other bits are in service and they also can change in value, this would produce false triggering. A DAPL expression can be added to the configuration to isolate a single desired bit as in the following example that selects bit position 4.

```
    DIGITALIN (pbits, 0, 40)
    pshifted = (pbits & $00000010) >> 4
    RANGE ( pshifted,  OUTSIDE, 0, 0,  pnonzero)
    PCASSERT( pnonzero, tevent,  IPIPE0 )
    …
```

Applications that test the value of one of the analog data streams, rather than using the digital input port, are already synchronized to the input data sequence and can generate a triggering event directly.

```
    LIMIT (IPIPE0,  INSIDE, 22000, 32767,  tevent)
```

...

9. Calibration

Every Data Acquisition Processor is fully calibrated by Microstar Laboratories as part of the manufacturing process. After long periods of time, incremental changes can occur in amplifier gains and offsets. Failing to recalibrate does not harm the Data Acquisition Processor, but it can result in persistent small measurement errors.

Few application sites will have the kind of testing standards necessary to perform an accurate calibration. If you think your device needs calibration, please contact Microstar Laboratories for information. A complete calibration is also available from Microstar Laboratories for a fee plus shipping costs.

10. BNC Panels

For applications that require reconfigurable "quick connect" signals, the xDAP 7410 is available with optional connector panels with 16 BNC coaxial input connectors. Behind these connectors are "analog expansion boards" similar to analog expansion boards previously used exclusively with the PCI-bus DAP models. If your system includes the BNC panel options, you will need to use special addressing notations to access the signals connected to the BNC connectors..

The internal expansion boards have all of the normal features of Microstar Laboratories expansion boards. They connect to the xDAP main module through backplane connections compatible with *a-Series* PCI analog backplane boards. (The boards most typically used are `MSXB037-05-E2E08-B` analog expansion, which support the BNC connectors.) Labeling the front panel slots of the xDAP 7410 enclosure from left to right as `J1`, `J2`, `...`, `J7`, then the two slots at positions `J4` and `J5` are the ones with the compatible analog backplane connections, and the two BNC panels cover slots `J3` through `J6`. Signal wires from the connectors are internally routed to terminals on terminal strip connectors. The termination boards provide convenient "breadboarding" areas suitable for mounting various signal line loading and bypass components at the terminals.

To access signals connected at panels in these two slot positions, you must tell the DAPL system how the signals are routed. DAPL 3000 version 1.00 introduces some new routing identifiers for this purpose. For an xDAP 7410, the routing codes are:

- `c0` - this accesses signals via the standard D-connector and it is the default connection if no connector name is specified.
- `c1` - this accesses signals via the first analog backplane slot at location `J4` behind the left panel.
- `c2` - this accesses signals via the second analog backplane slot at panel location `J5` behind the right panel.

The new routing identifiers can then be added as a prefix to the usual *pin designation* notation in the `set` commands that you use to define your input sampling configurations. Since the xDAP 7410 supports differential analog inputs, the individual channels are addressed for each board using the familiar differential input channel notations `d0` through `d7` for each of the connector panels, with the signals distinguished by their routing identifiers. For example, to access one differential channel `d2` through left-side connector panel (with connectors over slot locations `J3` and `J4`, connecting to the backplane at slot location `J4`), your input configuration would specify routing code `c1`:

```
set   ipipe0  c1:d2
```

The analog backplane slots at connector locations `J4` and `J5` are are wired independently of one another, allowing for 16 distinct differential analog input pairs on the backplane. Thus, if you use all of the available differential input channels, plus the high density analog connector, this gives the xDAP 7410 a maximum capacity of 32 separate input signals.

Suppose that you are using only the BNC connectors. Since the xDAP 7410 provides 8 independent channel selectors and 8 parallel digitizers, when using both slots you will typically configure the system to sample 8 inputs from each slot on alternating sampling cycles, as in the following example.

```
set ipipe0  c1:d0    // Analog connector at slot J4
set ipipe1  c1:d1    // Access all 8 channels, d0 to d7
set ipipe2  c1:d2
set ipipe3  c1:d3
set ipipe4  c1:d4
set ipipe5  c1:d5
set ipipe6  c1:d6
set ipipe7  c1:d7

set ipipe8  c2:d0    // Analog connectors at slot J5
set ipipe9  c2:d1    // Access all 8 channels, d0 to d7
set ipipe10 c2:d2
set ipipe11 c2:d3
set ipipe12 c2:d4
set ipipe13 c2:d5
set ipipe14 c2:d6
set ipipe15 c2:d7
```

For sampling 8 dedicated signals, using only one panel, omit the declarations for `ipipe8` through `ipipe15`.

In general, the `J4` and `J5` analog input slots on xDAP 7410 are compatible with a-Series accessory boards that do not require input expansion addressing, and that have the property that they can drive the analog backplane differentially (so that signals reach the DAP through a pair of signal lines). Not all of the a-Series boards have these properties. Using any accessory boards other than ones normally installed with the xDAP 7410 requires factory customization. Some alternative a-Series accessory boards that are compatible with xDAP 7410, and possibly useful, include the following:

- MSXB037-05-E2E08-B – provides BNC-type connectors
- Other MSXB037 board models – for user-supplied input signal conditioning, quick connect wiring, or DB37 connectors instead of BNC.
- MSXB065 – provides anti-alias filtering with optional amplification.
- MSXB067 – provides strain gauge input with analog input filtering.

The routing notations are not expected to change significantly for future versions of the DAPL 3000 system, or for future members of the xDAP family, but the codes and related channel numbering could be different for different xDAP models. For example, the routing code `c1` could refer to a different slot location or a different kind of signal channel on a different xDAP model. Any changes will be noted in the release notes for DAPL 3000.

11. Appendix A - xDAP and Other DAPs

The xDAP 7410 features a new and more configurable "channel selector" architecture. Though this is not a drastic departure from the way that Data Acquisition Processors worked in the past, it is an important departure. Users of other DAP board families will need to be aware of some important differences.

DAPL 3000

The xDAP is run by the embedded DAPL 3000 operating system. Many application configurations used with other DAP models will run unchanged, but this does not mean that they will run exactly the same.

The DAPL 2000 operating system allowed a number of obsolete configuration notations. In general, the DAPL 3000 system can no longer accept these. Among the changes that you must make to upgrade your configurations:

- Command abbreviations not specifically documented in the manual will not be recognized.
- Processing commands previously indicated as obsolete will not be recognized. (There are better alternatives.)
- The `CHANNELS` command is mandatory for sampling and updating configurations.
- Back-compatibility software triggering notations are not supported.

Input Sampling Configurations

xDAP devices have multiple channel selectors rather than just one. Where previously channels needed to be captured serially, the xDAP can measure multiple channels simultaneously.

The time interval for sampling activity on an xDAP is the time for collecting one sample for each data channel. The *scan time* to cover all of the channels in this manner is specified by the `SCAN` command, appearing after the channel assignments defined by `SET` commands. This is different from the older `TIME` command, which specified the time interval between measurements for individual data channels. If you use an old input configuration with a `TIME` command rather than a `SCAN` command, the DAPL 3000 system will accumulate the scan time required to cover all of the channels in the list serially, and set up the equivalent of a `SCAN` command by default. Your net data rates will be the same, but with channels sampled simultaneously, the sampling instants for each channel are probably not exactly as before.

For example, an input sampling configuration for an older DAP model could look something like the following.

```
// Example 1 for DAPL 2000
CHANNELS  4
SET  IP0   D0
SET  IP1   D2
SET  IP2   D4
SET  IP3   D6
TIME    5.0        // Total scan time 20.0
```

In effect, previous DAP models had only one channel selector, and all channels were routed through it. Consequently, 20 microseconds were required to complete the capture of all four channels in the channel list above.

With the xDAP 7410, there are multiple channel selectors. The four selected input signals of *example 1* connect to four separate channel selectors, so under DAPL 3000 they are captured in parallel. To match the timing of the older DAP models, you would use the following sampling configuration.

```
// Example 2 for DAPL 3000
CHANNELS  4
SET  IP0   D0
SET  IP1   D2
SET  IP2   D4
SET  IP3   D6
SCAN    20.0              // Scan time 20.0 i
```

If you give DAPL 3000 an old DAPL 2000 configuration, such as the one shown in *example 1* above, DAPL 3000 will reinterpret it as equivalent to *example 2*. Chapter 4 Channel Architecture provides more information about timing and parallel sampling.

For DAP models such as the DAP 5400a, there was a limited kind of parallel sampling. Channel groups, in certain restricted channel combinations with identical configurations, could be sampled simultaneously. With an xDAP, you can configure simultaneous sampling activity on the number of channels you want, with separate channel configurations for each channel. You can see exactly which pins are configured. A slight disadvantage, however, is that configuring each individual pin takes a little more work.

The following channel definition example is for a DAP 5400a, with one channel group of 8 channels. The range is restricted to gain 1 on all 8 channels in the group.

```
SET  IPIPE(0..7)    SPG0
```

The following is an equivalent definition for an xDAP 7410, except with two different range settings used.

```
SET  IPIPE0    D0      1
SET  IPIPE1    D2      1
SET  IPIPE2    D4      1
SET  IPIPE3    D6      1
SET  IPIPE4    D8      5
SET  IPIPE5    D10     5
SET  IPIPE6    D12     5
SET  IPIPE7    D14     5
```

Index