

# Appendix F: Configuration and Implementation Notes

---

## Keithley MetraByte DAS-16/1600

### Introduction

Configuration of DriverLINX requires installing the data acquisition hardware inside the computer and then creating a configuration file with the installation parameters for the hardware. To install the hardware, read and follow the manufacturer's instructions for the data acquisition board. DriverLINX does not require a specific hardware setup as it will adopt to any configuration allowed by the manufacturer. To create the configuration file, read either the *Interactive Configuration* or *Editing a Configuration File* sections that follow.

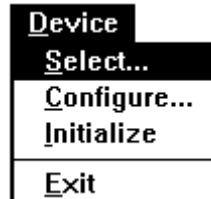
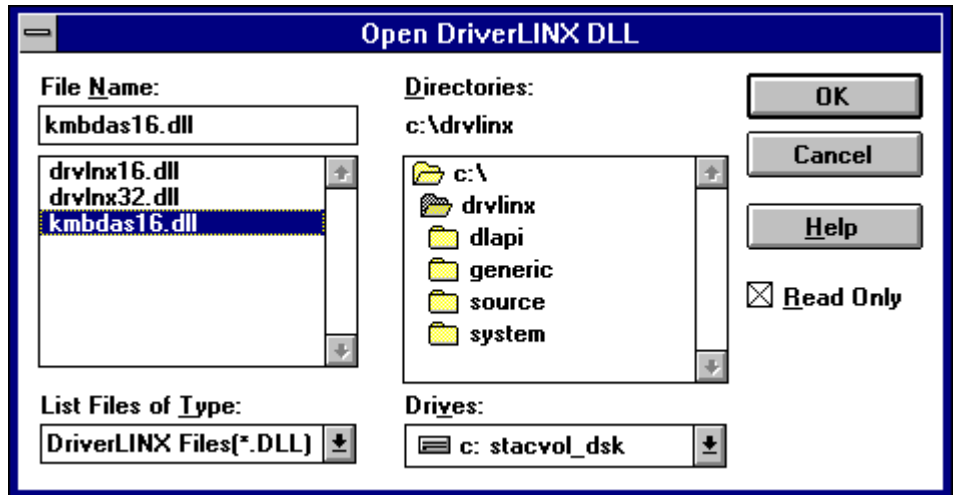
### Interactive Configuration

The simplest method to configure DriverLINX is with the interactive Device Configuration Dialog Box feature of DriverLINX itself. This dialog box presents a pull-down list for each of the allowed hardware setup options. After all the entries in the Configuration Dialog Box are completed, closing the dialog box writes an ASCII configuration file, **KMBDAS16.INI**, into the disk directory containing the Windows system files. Whenever DriverLINX is started, if this configuration file exists, DriverLINX will automatically read the file for the hardware configuration parameters.

To display the DriverLINX Configuration Dialog Box, perform the following steps:

1. Gather information about all the hardware switch settings and options available for the installed data acquisition board(s). Also locate any information or notes about the interrupt and DMA channels used by other hardware devices in your computer system. If you need help, see the appendix, *I/O Port, Interrupt, and DMA Channel Usage*.
2. Start Windows as you normally would and select the Program Manager window. Install DriverLINX now if you have not previously done so. The installation procedure for DriverLINX is described in the chapters *Quick Start* and *Installing DriverLINX(VB)*.

3. Either select the "Learn DriverLINX" icon created when you installed DriverLINX or enter "<drive>:\DRVLINX\LEARNDL"<sup>1</sup> in the *Command Line* edit box activated by selecting from the *File* menu the *Run...* option. <drive> is the letter of the hard disk drive where DriverLINX is installed.
4. Immediately after loading LearnDL, the Open DriverLINX DLL dialog box appears. Select the name of the hardware-specific DLL from the list for your A/D board. The name is an abbreviation of the board's model number. Typically only one hardware-specific DLL is listed along with other DriverLINX interfacing layer DLL's used by the application program internally.

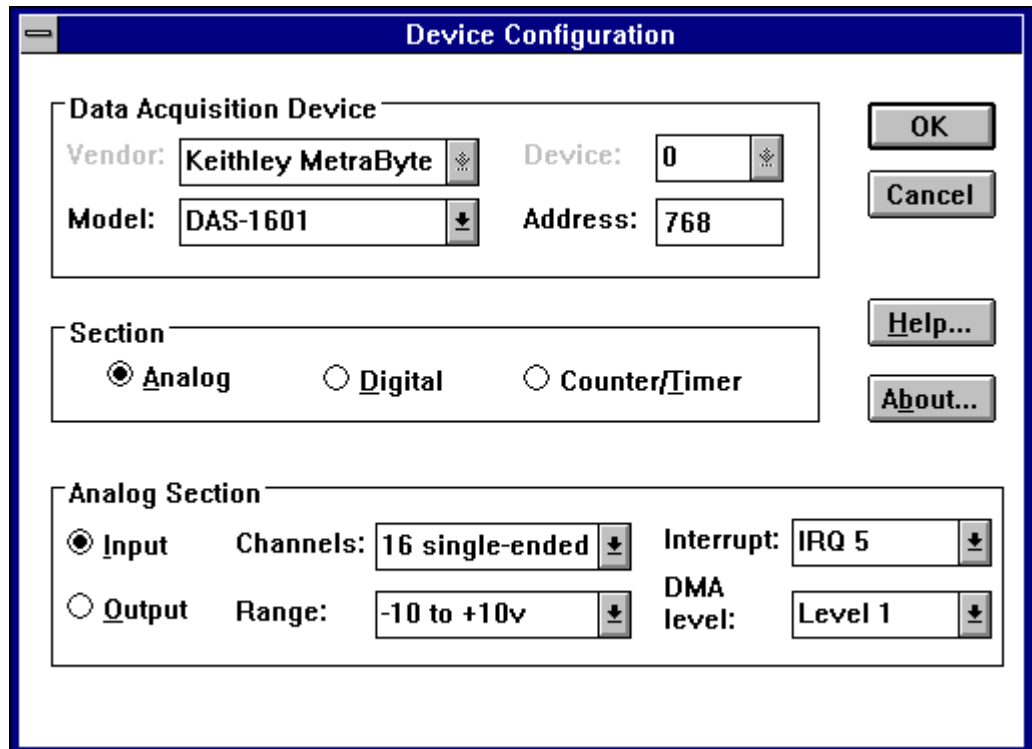


5. From the main menu bar of **LearnDL**, select the *Device* menu and choose *Select...*
6. Select the Logical Device you wish to configure and then click on the *OK* button (return).

<sup>1</sup> For DriverLINX/VB, the file specification is "<drive>:\DRVLNXVB\LEARNDL".



7. Again select the *Device* menu and then chose the *Configure...* option to display the Device Configuration Dialog Box.



8. From the *Model* list, select the model name for the data acquisition hardware you are configuring.
9. If the value displayed in the *Address* edit box is not correct, type the correct value into the box. You may enter the address in decimal or hexadecimal using the C-notation for hex, i.e., 768 decimal = 0x300 hexadecimal.
10. Chose the correct options for the *Analog*, *Digital*, and *Counter/Timer Sections* by first clicking on the appropriate radio button in the middle of the dialog box and then completing the group of dialog fields in the lower third of the dialog box. Be sure to click on both the *Input* and *Output* radio buttons for the *Analog* and *Digital* groups to see all the dialog fields.

11. After all the selections have been made, save the configuration parameters by clicking on the *OK* button. This will create or update the configuration file, **KMBDAS16.INI**, in the Windows directory.
12. Repeat the preceding steps, starting at step 4, for each Logical Device you wish to configure.

## Editing a Configuration File

The configuration file, **KMBDAS16.INI**, can also be created or edited using a plain ASCII text editor, such as the **Notepad** editor that comes with Windows. The layout and format of the configuration file is similar to the initialization files created by Windows and other Windows applications.

All entries in the configuration file have the following form:

```
[logical device number]
keyname=string
```

where “keyname” is the name of a field recognized by DriverLINX and “string” is the value for that field. The case of the letters in keyname and string is not significant.

A typical entry for Logical Device 0 in the configuration initialization file follows:

```
[Device 0]
Vendor=Keithley MetraByte
Model=DAS-16G1
Address=768
A/D channels=16
Min A/D volts=-10.0
Max A/D volts=10.0
Min D/A 0 volts=0
Max D/A 0 volts=5.0
Min D/A 1 volts=0
Max D/A 1 volts=5.0
IRQ=7
DMA=1
Clock=1 MHz
```

The string values recognized for the Vendor and Model keynames are defined in the DriverLINX API files, **OEMCODES.H** or **OEMCODES.BAS**. Case is not significant for matching a string value. The numeric values must all be decimal or floating point. The range of allowed values is defined by the hardware manufacturer.

### ***Vendor***

The only allowed string value is “Keithley MetraByte”.

## **Model**

DriverLINX for Keithley MetraByte DAS-16/1600 Models<sup>2</sup> supports the following hardware:

DAS-16
DAS-16F
DAS-16G1
DAS-16G2
DAS-1601
DAS-1602
DAS-1401
DAS-1402
DAS-1201
DAS-1202

The difference among these boards is maximum A/D throughput and the way the A/D gain is selected. The DAS-16G1, DAS-16G2, DAS-1601, DAS-1602, DAS-1401, and DAS-1402 have software selectable gain while the DAS-16, DAS-16F, DAS-1201, and DAS-1202 have A/D gain selected by DIP switch settings on the board. In addition, the DAS-1601, DAS-1602, DAS-1401, and DAS-1402 support 100 kHz A/D sampling rates, burst mode sampling. The DAS-1601, DAS-1602, DAS-1201, and DAS-1202 have 24 additional digital I/O lines.

## **Address**

The default address used by all DAS-16/1600 boards is 768 decimal or 0x300 hex. If you have more than one Keithley MetraByte board or another peripheral card at the same address, you will have to change the DIP switch settings for the device address on the board. Consult the hardware reference manual for the location and settings for these switches.

## **A/D Channels**

The DAS-16/1600 A/D channels can be configured as 8 differential or 16 single-ended analog inputs using a slide switch on the hardware board.

## **A/D Volts**

The minimum and maximum analog full-scale voltages at unity gain must be specified. For the DAS-16G1, DAS-16G2, DAS-1601, DAS-1602, DAS-1401, and DAS-1402, the minimum voltage is -10 volts for bipolar operation and 0 volts for unipolar operation. The maximum voltage is +10 volts. For the DAS-16 and DAS-1200, the minimum and maximum voltages are set by the unipolar/bipolar switch and the gain switches on the board. The DAS-1200 models only have bipolar analog input ranges with a minimum input voltage of -5 volts and a maximum of +5 volts. See the manufacturer's literature for details.

## **D/A Volts**

To use the D/A channels on the DAS-16 boards, a DC reference voltage must be connected to the **D/A REF IN** terminals for each channel. The D-type male connector that projects through the rear panel of the computer supplies a -5 v reference that can be jumpered to the **D/A REF IN** terminals to provide a 0 to +5 volt D/A range. Consult the hardware reference manual for details. Remember, the output has the opposite polarity of the reference input.

---

<sup>2</sup> "DAS-16/1600" when used alone, refers to the entire group of A/D board models listed.

The output range for the D/A channels on the DAS-1600 boards can be selected by DIP and slide switches located along the top edge of the board. Each channel can be independently configured as unipolar or bipolar with a 5, 10, or user-selected output voltage.

### ***IRQ***

An interrupt request level must be selected in the range 2-7. Interrupt lines can be shared so long as two devices sharing an interrupt will not be used concurrently. A list of common hardware interrupt assignments is presented in the appendix, *I/O Port, Interrupt, and DMA Channel Usage*.

### ***DMA***

A slide switch on the hardware board selects between levels 1 or 3 for DMA operations. The factory configuration is for level 1. If you want to use DMA and you have more than one Keithley MetraByte board, or another card which uses channel 1, then you will have to slide the switch to select channel 3. If you are using an XT, the hard disk may be using level 3. DMA operations are available for A/D conversions only.

### ***Clock***

The input to the counter/timers used for A/D timing is selectable for a 1 or 10 MHz input frequency.

### ***Other Entries***

The DAS16/1600 initialization file (KMBDAS16.INI) supports several other keynames that must be entered using a text editor. These entries are used for less frequently required options or to support hardware engineering changes.

DriverLINX supports several optional keynames for configuring the DAS-16/1600 during initialization. These options are useful for overriding the default initial states of hardware outputs either programmed by DriverLINX or established by the hardware during device initialization.

You must add these options to the <device>.INI file in the \WINDOWS directory manually using an ASCII text-editor, such as NOTEPAD. Add the options under each Logical Device for which they apply. The option strings are case-insensitive, but they must otherwise be typed exactly as shown below. Unspecified keywords default to false.

Option Keyname	Option Value	Default Value
D/A %u init	= <Boolean>	(TRUE)
Init D/A %u volts	= <value>	(0.0)
DIO init config	= <Boolean>	(TRUE)
Config DIO %u	= <value 0 1>	(1)
DO %u init	= <Boolean>	(TRUE)
Init DO %u value	= <value>	(0)

Replace "%u" with a Logical Channel number. <Boolean> may be "TRUE", "FALSE", "T", "F", "YES", "NO", "Y", "N", "0", or "1". D/A <value> is a floating point D/A output voltage. DO <value> is an unsigned integer that will be written to the specified output port. DO <value 0|1> is zero for output and one for input. Examples of initialization configuration option settings are shown in the following table.

Keyname	Description
"D/A %u init=TRUE"	Enables initialization of the analog output values, if supported by the hardware.
"Init D/A %u volts=0.0"	Sets the initial value if initialization has been enabled.
"DIO init config=TRUE"	Enables configuration of any configurable ports supported by the hardware.
"Config DIO %u=1"	Configures the specified port as input, if configuration is enabled.
"DO %u init=TRUE"	Enables initialization of the output values on the specified digital port.
"Init DO %u value=7"	Sets the digital output value to seven, if initialization is enabled.

## Implementation Notes

The following sections describe miscellaneous information about how the Keithley MetraByte DAS-16/1600 A/D boards are implemented using the DriverLINX API.

The following table summarizes allowed data acquisition modes supported by each subsystem of DriverLINX using the Keithley MetraByte DAS-16/1600.

Modes Supported by DAS-16/1600 Models

<b>Subsystem</b>	<b>Polled</b>	<b>Interrupt</b>	<b>DMA</b>	<b>Other</b>
Analog Input	√	√	√	√
Analog Output	√	√		√
Digital Input	√	√		√
Digital Output	√	√		√
Counter/timer	√			√
Device				√

The following table summarizes the operations implemented for each of the supported modes of DriverLINX's subsystems using the Keithley MetraByte DAS-16/1600 boards.



Allowed Operations and Events for Supported Subsystem Modes

Subsystem	Operation <sup>3</sup>	Events		
		Timing	Start	Stop
Mode				
<b>Analog Input</b>				
Polled	Start	rate, dig	cmd, ana, dig	TC, ana, dig
Interrupt	Start, Stop, Status	rate, dig	cmd, ana, dig	cmd, TC, ana, dig
DMA	Start, Stop, Status	rate, dig	cmd, dig	cmd, TC
Other	Initialize			
<b>Analog Output</b>				
Polled	Start	rate, dig	cmd, dig	TC
Interrupt	Start, Stop, Status	rate, dig	cmd, dig	cmd, TC, dig
Other	Initialize			
<b>Digital Input</b>				
Polled	Start, Configure <sup>4</sup>	rate, dig	cmd, dig	TC
Interrupt	Start, Stop, Status	rate, dig	cmd, dig	cmd, TC
Other	Initialize			
<b>Digital Output</b>				
Polled	Start, Configure	rate, dig	cmd, dig	TC
Interrupt	Start, Stop, Status	rate, dig	cmd, dig	cmd, TC
Other	Initialize			
<b>Counter/timer</b>				
Polled	Start	rate		
Other	Initialize			
<b>Device</b>				
Other	Initialize, Capabilities, Request, Release			

<sup>3</sup> All subsystems allow the *MESSAGE* operation which is not shown in the table.

<sup>4</sup> Supported only by the DAS-1600 series on Logical Channels 2-7.

## Analog Input Subsystem

The DAS-16/1600 models contain one A/D converter and a 16-channel multiplexer. The A/D channels can be used in differential (channels 0-7) or single-ended (channels 0-15) configuration. The inputs are labeled **CH0 HI IN** [37], **CH0 LO IN/CH8 HI IN** [18], etc., and are located on the D-type male connector that projects through the rear panel of the computer. The DAS-16 and DAS-1200 models provide switch selectable input ranges while the other models provide software selectable programmable gain .

### **Analog Input Initialization**

Initialization of the analog input subsystem aborts any active DMA or interrupt data acquisition tasks, sets the channel multiplexer to scan all channels, and resets the clock to the slowest possible rate. After a Logical Device is configured, the next time initialization is called, a check for the presence of hardware and agreement with the configuration parameters is performed.

### **External Triggering**

External triggers are implemented using a Digital Start Event reading from the 4-bit digital input port. DriverLINX will use hardware triggering if the Digital Start Event corresponds to an active high level at input port line, **IP0/Trig 0** [25]. However, all external triggers require software polling so the data hold time for a trigger must be at least 3  $\mu$ s on a 10 MHz AT, to at least 20  $\mu$ s on a 4.77 MHz XT.

External triggers using the additional digital I/O lines on the DAS-1600 and DAS-1200 model boards are not yet supported.

### **External Clocking**

External clocking is supported using either a Rate Timing Event on channel 1 with external clock input or a Digital Timing Event that tests input port line, **IP0/Trig 0** [25]. Both techniques implement the hardware external clocking feature of the DAS-16/1600. Note that these models use **IP0/Trig 0** for both external clocking and external triggering.

### **Burst Mode Sampling**

Burst mode sampling is supported by the DAS-1600/1400/1200 model boards. To setup burst mode sampling, specify the *mode* field of a *Rate Event* as "BURSTGEN". The *period* field is the time in clock tics between bursts. The *onCount* field is the time between successive samples of a burst in clock tics, and *pulses* specifies the number of samples in a burst. Usually, the number of pulses in a burst will equal the number of channels to be scanned, but this is not required by the hardware or DriverLINX. The number of pulses must be between 1 and 16. Burst mode sampling is supported in *POLLED*, *INTERRUPT*, and *DMA* modes, but the maximum sustainable acquisition rate in each mode cannot be exceeded without loss of data. Also, the duration specified in the *onCount* field must be between 10 and 254  $\mu$ sec.

## Data Coding

The A/D uses offset binary coding for bipolar inputs and straight binary for unipolar inputs. For all ranges and configurations the minimum value reported is 0 and the maximum value is 4095.

A/D data can be returned in native or integer formats. The native format of the DAS-16/1600 boards returns the 12 A/D bits left shifted by four bits and the low-order four bits encode the A/D channel of the sample. The integer format returns the data as a two's-complement number with a range of -2048 to 2047 in bipolar mode and 0 to 4095 in unipolar mode. Integer format is only supported for *POLLED* and *INTERRUPT* modes, while native format is supported by all modes.

Example: For a Gain of 1 in the  $\pm 10$  v range (bipolar configuration) the voltage

resolution would be  $\frac{20\text{v}}{4096} = 4.88$  mv.

Input Voltage	A/D Binary Value
-10 v	0
0 v	2048
+4.88 mv	2049
+10 v	4095

For a Gain of 1 in the 0 to +10 v range (unipolar configuration) the voltage resolution

would be  $\frac{10\text{v}}{4096} = 2.44$  mv.

Input Voltage	A/D Binary Value
0 v	0
+ 5 v	2048
+5.002 v	2049
+10 v	4095

The values returned by the A/D converter always range from 0 to 4095 independent of the gain or channel configuration. For example for a Gain of 100 (bipolar configuration) the voltage resolution would be  $\frac{20\text{v}}{4096 \times 100} = 48.8 \mu\text{v}$ .

Input Voltage	A/D Binary Value
-100 mv	0
0 v	2048
48.8 $\mu\text{v}$	2049
+100 mv	4095

### ***A/D Conversion Delay***

The maximum A/D conversion delay is 429,496,729  $\mu\text{s}$  or 7 minutes, 9 seconds with a 10 MHz input frequency and 4,294,967,296  $\mu\text{s}$  or 1 hour, 11 minutes, 35 seconds with a 1 MHz clock.

The minimum conversion delay for the counter/timers is 4 clock ticks. However, minimum usable conversion delay depends on the data acquisition mode, the speed of the computer, and the gain and A/D conversion speed of the data acquisition board. The maximum sustainable data acquisition rate decreases with gain on boards with programmable gain. Consult the hardware manual for additional details and perform empirical tests with your specific hardware setup to determine the minimum practical conversion delay.

### ***A/D Data Lost***

DAS-16/1600 models do not provide hardware monitoring of A/D data overruns. You must determine empirically that the required conversion delay is supported by your system. The easiest way to check is by inserting a known periodic signal into an A/D channel and picking the smallest conversion delay required. Use the time measurement cursors on the Digital Storage Oscilloscope in **LearnDL** to measure the period of the recorded waveform. If the selected conversion delay is too small, the measured period of the input signal will decrease. If this happens, increase the conversion delay until the measured period matches the input period.

## **Analog Output Subsystem**

The DAS-16/1600 boards contain two channels of multiplying 12-bit D/A converters. The outputs are labeled **D/A 0 OUT** [9] and **D/A 1 OUT** [27] and are located on the D-type male connector that projects through the rear panel of the computer. The DAS-1400 and DAS-1200 models do not support analog output.

### ***Analog Output Initialization***

Initialization of the Analog Output subsystem loads zero into both D/A channels forcing the output voltage to zero in unipolar mode and negative full scale in bipolar mode by default.

## External Triggering

External triggers are implemented using a Digital Start Event reading from the 4-bit digital input port. For analog output all external triggers require software polling so the data hold time for a trigger must be at least 3  $\mu$ s on a 10 MHz AT, to at least 20  $\mu$ s on a 4.77 MHz XT.

## External Clocking

External clocking is supported using either a Rate Timing Event on channel 1 with external clock input or a Digital Timing Event that tests input port line, **IP0/Trig 0** [25]. Both techniques implement the hardware external clocking feature of the DAS-16/1600 models. Note that these boards use **IP0** for both external clocking and external triggering.

## Data Coding

The D/A channels use straight binary coding. The minimum value that can be used is 0 and the maximum value is 4095. Analog output values are determined by the external voltage reference.

To utilize any D/A functions you will need to connect a reference voltage into the **D/A 0 REF IN** [10] terminal when using D/A channel 0, or **D/A 1 REF IN** [26] terminal when using D/A channel 1. You can use the on-board fixed -5 vdc reference (**Vref (-5v)** [8]) for analog output in the 0 to +5v range. The maximum reference voltage which can be used is  $\pm 10$ v. For DAS16's, the output is unipolar in a range from 0 to -Vref. For DAS1600's, the output can also be bipolar and would then range from +Vref to -Vref. For example, if you use the -5 v reference voltage with a DAS-16, the D/A output range will be from 0 to +5 v as follows:

$$\text{Voltage Resolution} = \frac{-V_{\text{ref}} = 5\text{v}}{4096} = 1.22 \text{ mv}$$

Binary value	D/A Output Voltage
0	0
2048	+2.5
4095	+5.0

If you select the 5 v range and select bipolar mode with a DAS-1600, the D/A output range will be from -5 to +5 v as follows:

$$\text{Voltage Resolution} = \frac{(-V_{\text{ref}} = 5\text{v}) \times 2}{4096} = 2.44 \text{ mv}$$

Binary value	D/A Output Voltage
0	-5.0
2048	0
4095	+5.0

If you output larger or smaller values, the high-order bits will be ignored. So 4096 would give an output voltage of 0v, and -1 would give an output voltage of 5.0 v. See the hardware reference manual for more information on using the DAC's.

### ***D/A Conversion Delay***

D/A conversions are timed using either software polling of the counter/timer or interrupts from A/D end-of-conversion signal. Either technique requires software to update the hardware registers with new values as rapidly as possible after a clock event is recognized. The DAS-16/1600 models cannot support direct hardware timing of D/A output.

The minimum D/A conversion delay is 4 clock tics. However, the practical minimum conversion delay is much higher and depends on the type and the speed of your machine. Variations in interrupt latencies and bus timing will add noticeable jitter to D/A signals output at high rates. This is also strongly influenced by the CPU's operating mode—real versus protected mode. Consult the appendix, *Interrupt Latencies in Multitasking Environments* for more information and run empirical tests to determine suitability of D/A output for the intended purpose.

The D/A conversion delay to output a periodic waveform with a given number of samples per period can be determined simply. For instance, the conversion delay for a 200 Hz wave with 100 samples/cycle is

$$T = \frac{1}{\text{frequency}} = \frac{1}{200 \text{ Hz}} = 5 \text{ ms}$$

$$\text{D/A conversion delay} = \frac{5 \text{ ms}}{100 \text{ samples}} = 50 \text{ ms}$$

### ***D/A Data Lost***

There is no way for the hardware to monitor D/A overruns. You must empirically insure that the required conversion delay can be supported. One way to check is using the D/A waveform generator to output a periodic waveform. Select the desired delay and then measure the frequency output either with an oscilloscope or a frequency counter. If the D/A conversion delay you chose was too small, the output frequency will vary from the selected frequency. Try larger delays until the frequencies match.

# Digital Input and Output Subsystems

## Logical Channels

The DAS-16/1600 boards have a custom 4-bit digital input port and a 4-bit digital output port (Logical Channel 0). The DAS-1600 and DAS-1200 models also support an additional 24-bits of digital I/O using an Intel 8255 Programmable Peripheral Interface (PPI) chip (Logical Channels 2-7). The custom input channel is configured as **IP0**, **IP1**, **IP2**, **IP3** and the output channel as **OP0**, **OP1**, **OP2**, **OP3**. **IP0** also serves as the DAS-16/1600 external clock and triggering input (Logical Channel 1) and **IP2** is also the gate for counter 0. The 8255 channels are mapped to DriverLINUX's Logical Channels as follows:

Logical Channel	Physical Channel
0	Custom Digital I/O port
1	IP0/Trig 0 [25]
2	Port A
3	Port B
4	Port C, upper
5	Port C, lower
6	Port C, upper and lower
7	Port A and Port B

DriverLINUX implements all digital I/O on the DAS-16/1600 boards as immediate, single-value transfers on execution of the Service Request. All digital I/O is right-justified. Unused bits are ignored on output and are returned as zero on input. Logical Channels 0, 4, and 5 return 4-bit values, Channel 7 returns 16-bit values (Port A is MSB and Port B is LSB), and other channels return 8-bit values.

Logical Channels 2-7 corresponding to the 8255 PPI ports may be dynamically reconfigured using a Digital Setup Event for the Timing Event and *CONFIGURE* as the operation code in the Service Request. The Digital Setup Event specifies the channel to configure and the configuration mode. For the DAS-1600 and DAS-1200 only basic mode, *DIO\_BASIC* (unlatched inputs, latched outputs), is supported. To configure a channel for output, specify the Digital Output subsystem, and, for input, specify the Digital Input subsystem.

The output values of channels corresponding to the 8255 PPI ports are latched and may be read at any time by requesting Digital Input on the desired channel. The output value of Logical Channel 0 on the DAS-16/1600 cannot be read back. If required, the application must maintain a shadow copy of the current output value on this channel.

### ***Digital Input Initialization***

Initialization of the Digital Input subsystem reprograms all ports of the 8255 PPI (Logical Channels 2-7) for input in 8255 Mode 0 (unlatched inputs). Logical Channel 0 is unchanged by initialization.

### ***Digital Output Initialization***

Initialization of the Digital Output subsystem resets the value of all output ports to zero. Ports currently configured as inputs are not affected. Note that, unlike Digital Input initialization, the 8255 PPI ports are not reconfigured.

### ***Digital I/O Conversion Delay***

Digital I/O on Logical Channel 0 may be clocked using the same technique as discussed for the analog output subsystem. Digital I/O on Logical Channels 2-7 is not clocked. Values on these channels can only be accessed using *POLLED* mode with single value I/O transfers.

### ***Digital I/O Data Lost***

Digital I/O data overruns are not detected by the hardware.

## **Counter/Timer Subsystem**

The DAS-16/1600 uses an Intel 8254 Programmable Interval Timer which consists of 3 internal 16-bit counters, Counter 0, Counter 1, and Counter 2. The DAS-16/1600 boards operate Counters 1 and 2 in a fixed divider configuration with the input of Counter 1 connected to a 1/10 MHz crystal oscillator and the output of Counter 1 connected to the input of Counter 2. Counter 0 is independent and its input, output, and gate control are available to the user. Counter 0 uses either an external clock signal at **Counter 0 Clock In** [21] or a 100 kHz internal clock source.

Although the DAS-16/1600 has 3 counters, two are internally tied together (Counter 1 and 2). Therefore, DriverLINX implements only two logical counters. Logical counter 0 corresponds to Intel 8254 Counter 0, while logical counter 1 corresponds to Intel 8254 Counters 1 and 2. These two logical counters are not equivalent in capabilities due to the internal design of the DAS-16/1600. Logical counter 0 can only generate square waves and pulses that are low for one clock cycle while logical counter 1 is more flexible.

### ***Counter/Timer Initialization***

Power-up state of the counters is undefined. Initialization establishes the state of each counter as a binary rate generator with a down count of 65535.

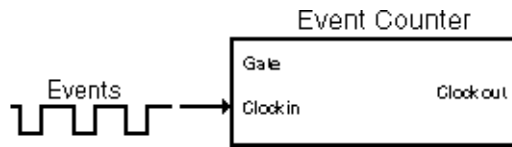
### ***Counter/Timer Interrupt***

Counter/timer interrupts are not supported.



## Event Counting

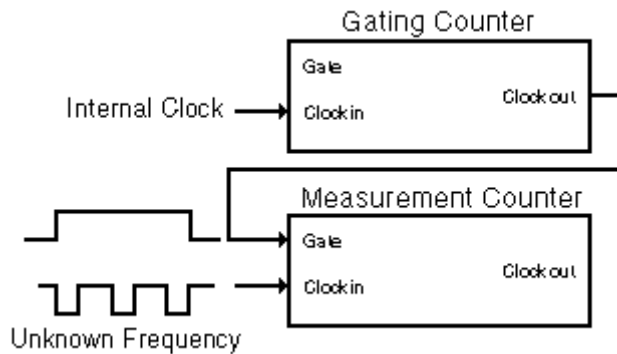
Event counting is the simplest counter/timer function. Only one counter is required and the external events to be counted are used as an external clock input signal to the counter. The application program reads the accumulated count as it deems appropriate.



As the DAS-16/1600 allows an external clock input only for Counter 0, DriverLINX supports event counting only for logical counter 0. The events to be counted are connected to **Ctrl 0 Clock In** [21]. Events are active low TTL-clock pulses, where a clock pulse is defined as a falling edge followed by a rising edge. **Gate 0/IP2** [24] must be left open or held high. The maximum number of events that may be counted before the counter rolls over is 65535. Counter rollover is detected and reported.

## Frequency Measurement

Without external hardware, support of frequency measurement using the DAS-16/1600 hardware configuration is limited. Frequency measurement requires an external connection between two counters.

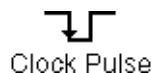


Frequency measurement requires two counter/timers used as measurement and gating counters. The unknown frequency is input as the clock source to the measurement counter. The gate input of the measurement counter is then activated for a known interval as timed by the gating counter. The gating counter is clocked from a known internal crystal reference clock. The unknown input frequency is then calculated as

$$\text{frequency} = \frac{\text{measured count}}{\text{gate time}}$$

The accuracy of the measurement is a function of the unknown input frequency and the gate time. As the input frequency decreases, the gate time must increase to preserve accuracy. It is the responsibility of the application program to choose the gate time. To measure a 0.1 Hz signal, the gate time should be approximately 3 minutes.

In DriverLINX for the DAS-16/1600, logical counter 0 is the measurement counter and logical counter 1 is the gating counter. Therefore, the operation is only defined for logical counter 1 which uses the 1/10 MHz internal clock. The output of logical counter 1 (**Counter 2 Out** [20]) must be externally connected to the gate input of logical counter 0 (**Gate 0/IP2** [24]). The frequency to be measured is fed to the clock input of logical counter 0 (**Counter 0 In** [21], active low). Remember the signal must be TTL (0 to 5 volts). Do not exceed this voltage range as damage to the counter could result.



Clock pulses are defined as a falling edge followed by a rising edge. This is a function of the Intel 8254 hardware and cannot be changed without external hardware.

The duration of the gating interval has a small jitter as the gate activates before the software can load the time duration into the gating counter. Therefore, the relative accuracy of the measurement will be less with shorter measurement intervals and higher input frequencies than would be anticipated from hardware considerations alone.

The application must poll for the results using the counter/timer *STATUS* Service Request. Overflow of the measurement counter is detected, but recycling of the gating counter cannot be prevented. As the gating counter is operated in a square wave mode, the application program has a time equal to the gate count period after the measurement is completed to check the results before measurements are restarted. Also note that the gating counter has a resolution of 1/.1  $\mu$ sec (1/10 MHz clock).

If the *STATUS* function returns with the measurement completed, the task is automatically canceled. The *STOP* function is only necessary to terminate a measurement that is still in progress or that has not returned from a status check as completed.

Also see the Burr-Brown Data Acquisition Handbook for simple external circuits using an Intel 8254 for performing frequency measurements.

### ***Pulse Output***

The DAS-16/1600 counter/timers can be used to generate TTL pulse trains. Due to the design of the DAS-16/1600, the two logical counters have different capabilities.

Square waves must specify the output *period* and variable duty cycle waves must specify the output *period* and the *onCount*.

Logical counter 0 can only generate square waves from 50 kHz to 1.5 Hz using the 100 kHz internal clock. The output frequency is determined by the *period* field of the Rate Event and the *onCount* field is returned as  $\frac{\text{period} + 1}{2}$ .

Logical counter 1 can generate square and variable duty cycle waves using the 1 or 10 MHz internal clock. However, arbitrary duty cycles are not possible. The *onCount* must be  $\geq$  off count ( $= \text{period} - \text{onCount}$ ) and the ratio  $\frac{\text{onCount}}{\text{off count}}$  must be an integer. If the above conditions are not met, the *onCount* will be adjusted to comply. Output pulses cannot be entirely stopped, only slowed to period of 1.2(0.12)/hr.

### ***Oneshot Pulse Output***

A single TTL pulse can be output by setting the *pulses* field in the Rate Event to 1. However, hardware restrictions of the DAS-16/1600 timers do not allow exact compliance with the DriverLINX API. The pulse output is inverted (active low); the Intel 8254 cannot generate a single active high pulse. A warning message about inverted pulse levels is always returned by DriverLINX when using the DAS-16/1600.

To implement oneshot pulse outputs, DriverLINX uses Mode 0 of the Intel 8254 (pulse on terminal count): Output goes low when counter is loaded and remains low until end of count. The gate must be high for counting. Actual delay is *onCount* + mode setup time (as loading control register forces output low).

As with other counter/timer functions, the two logical counters are implemented differently.

Logical counter 0:

**Gate 0/IP2** [24] is always enabled by DAS-16/1600 hardware. Therefore, a low gate will inhibit counting and increase the duration of the pulse.

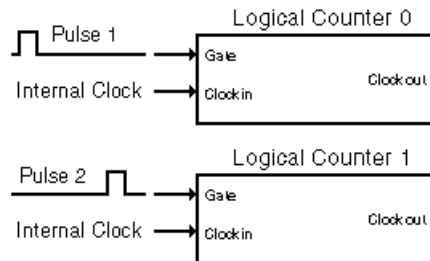
Logical counter 1:

**IP0/Trig 0** [25] gate is disabled by software.

N.B. Only the *onCount* determines duration; the *period* count is ignored.

## Time Interval Measurements

Counter/timers can be used to measure the time interval between the receipt of two consecutive pulses. Either a separate counter/timer can be used to detect the edge of each pulse, or a single counter/timer can measure the interval between two pulse edges.



Support for time interval measurement using the Intel 8254 is limited to using two counter channels. Both the Intel 8254 and the DAS-16/1600 implementation limit the ability of the DriverLINX API to support time interval measurements.

DriverLINX can only measure time interval using both logical timers. Pulse inputs (rising edges) are sensed at **Gate 0/IP2** [24] on logical counter 0 and at **IP0/Trig0** [25] on logical counter 1.

Time interval measurements are implemented using logical counters 0 and 1 with two different internal clock rates. The count from the slower counter is scaled to the clock rate of the faster counter. Therefore, the overall accuracy is governed by the 100 kHz clock on logical counter 0. DriverLINX, however, reports the time interval with the resolution of the 1/10 MHz clock.

The Intel 8254 has only two edge-triggered modes (mode 1, hardware retriggerable one-shot, and mode 5, hardware triggered strobe) which can be used for time interval measurements using two channels. However, the DAS-16/1600 uses a 1/10 MHz clock for counters 1 and 2, and a 100 kHz clock for counter 0. As the gates of Counters 1 and 2 are tied together, Counter 1 cannot be used as a prescaler for Counter 2 to equalize the clock inputs without stopping Counter 1 when the gate is low.