Appendix F: Configuration and Implementation Notes

Keithley MetraByte DAS-8

Introduction

Configuration of Driver*LINX* 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. Driver*LINX* does not require a specific hardware setup as it will adopt to almost any configuration allowed by the manufacturer. To create the configuration file, read either the *Interactive Configuration* or *Editing a Configuration File* sections that follow.

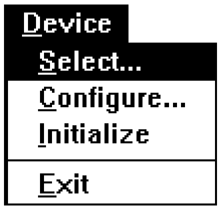
Interactive Configuration

The simplest method to configure Driver*LINX* is with the interactive Device Configuration Dialog Box feature of Driver*LINX* 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, KMBDAS8.INI, into the disk directory containing the Windows system files. Whenever Driver*LINX* is started, if this configuration file exists, Driver*LINX* will automatically read the file for the hardware configuration parameters.

To display the Driver*LINX* Configuration Dialog Box, perform the following steps:

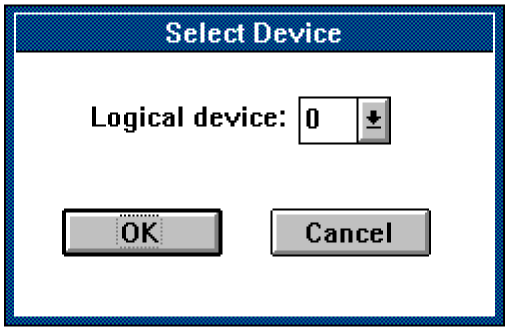
1. Gather information about all the hardware switch settings and options available for the installed data acquisition board(s). Also locate any information or notes about the interrupt and DMA channels used by other hardware devices in your computer system. If you need help, see the appendix, *I/O Port,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

3. Either select the Driver*LINX* icon created when you installed Driver*LINX* or enter “`<drive>:\DRVLINX\LEARNDL`”[1] 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 Driver*LINX* is installed.

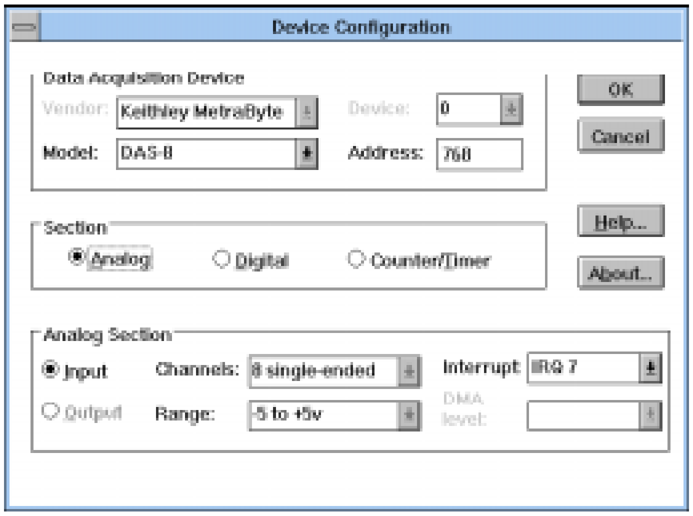


4. From the main menu bar of **LearnDL**, select the *Device* menu and chose *Select...*

5. Select the Logical Device you wish to configure and then click on the *OK* push button (return).



6. Again select the *Device* menu and then chose the *Configure...* option to display the Device Configuration Dialog Box. Note that extensive online help is available by pressing the *Help* push button or by selecting a field and pressing the F1 key.


Device Configuration

Data Acquisition Device
Vendor: Keithley MetraByte
Device: 0
Model: DAS-8
Address: 768

Section
[✓ Analog] Digital Counter/Timer

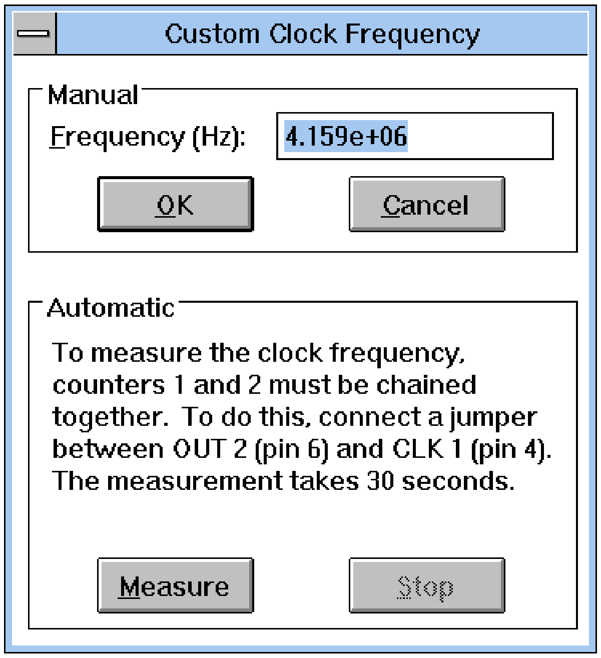
Analog Section
[✓ Input] Output
Channels: 8 single-ended
Range: -5 to +5v
Interrupt: IRQ 7
DMA level:

[1] For Driver*LINX*/VB, the file specification is "`<drive>:\DRVLNXVB\LEARNDL`".

7. From the *Model* list, select the model name for the data acquisition hardware you are configuring.
8. 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.
9. 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. For the *Counter/Timer Section*, see the instructions below for setting a custom clock frequency.
10. After all the selections have been made, save the configuration parameters by clicking on the *OK* push button. This will create or update the configuration file, **KMBDAS8.INI**, in the Windows directory.
11. Repeat the preceding steps, starting at step 4, for each logical device you wish to configure.

Custom Clock Frequencies

For timing, the DAS-8 series uses the clock signal on the PC I/O bus or an on-board crystal oscillator. While the original IBM PC XT, and compatibles, always had a 2.384 MHz bus clock, ISA and EISA bus machines have no accepted standard for the PC bus clock. On many machines, the clock frequency is ½ the CPU clock frequency while, on others, it is adjustable using motherboard jumpers or advanced BIOS setup options. Few manufacturers provide this information in their documentation. This frequency, however, can be measured by Driver*LINX* using the 8254 timer/counters on the DAS-8.



Frequency (Hz): 4.159e+06

To measure the clock frequency, counters 1 and 2 must be chained together. To do this, connect a jumper between OUT 2 (pin 6) and CLK 1 (pin 4). The measurement takes 30 seconds.

To measure the PC bus clock frequency, Driver*LINX* requires that Counters 1 and 2 be chained together using an external jumper. To chain the counters, connect **Counter 2 Out** [6] to **Counter 1 In** [4] on the DAS-8 external connector. Then, select the *Counter/Timer Section* in the Configuration Dialog Box. Choosing the item "Custom clock" in the drop-down list of the Resolution combo-box displays the Custom Clock Frequency Dialog Box. Selecting *Measure* will start a 30 second measuring interval for determination of the PC bus clock frequency. The frequency box will update every second with the currently measured frequency. The measuring interval may be stopped at any time using the *Stop* push button. The displayed frequency is added to the Resolution list box by selecting *OK*, or the measurement can be abandoned by selecting *Cancel*. The displayed frequency may be edited manually by selecting the numbers in the edit box and typing a new value for the selected digits.

Editing a Configuration File

The configuration file, **KMBDAS8.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 Driver*LINX* 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-8
Address=768
A/D channels=8
Min A/D volts=-5.0
Max A/D volts=5.0
IRQ=7
DMA=-1
Clock=4.15 MHz
```

The string values recognized for the Vendor and Model keynames are defined in the Driver*LINX* API file, **OEMCODES.H**. 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

Driver*LINX* for Keithley MetraByte DAS-8 supports the following hardware:

```
DAS-8
DAS-8PGA
DAS-8/AO
```

Address

The default address used by the DAS-8 is 768 decimal or 0x300 hex. If you have more the one Keithley MetraByte device 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-8 A/D channels are configured as 8 single-ended or differential analog inputs. Driver*LINX* does not distinguish between single-ended and differential inputs.

A/D Volts

The minimum and maximum analog full-scale voltages at unity gain must be specified. For the DAS-8, the minimum and maximum voltages are fixed at ±5 volts.

IRQ

If hardware interrupts are not supported, select -1 as the value, otherwise 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 an appendix.

DMA

DMA is not used by the DAS-8 series. This item must either have the value -1, or the option may be deleted from the file with a text editor.

Clock

The input to the counter/timers used for A/D timing varies. The DAS-8 uses the PC bus clock while the DAS-8PGA and DAS-8/AO have an on-board 1 MHz crystal oscillator. The frequency of the PC bus clock may be determined automatically as described above.

Implementation Notes

The following sections describe miscellaneous information about how the hardware features of the Keithley MetraByte DAS-8 series correspond to the Driver*LINX* API.

The following table summarizes allowed data-acquisition modes supported by each subsystem of Driver*LINX* using the Keithley MetraByte DAS-8 series.

Modes Supported by DAS-8 Series

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

The following table summarizes the operations and events implemented for each of the supported modes of Driver*LINX*'s subsystems using the Keithley MetraByte DAS-8.

Allowed Operations and Events for Supported Subsystem Modes

Subsystem	**Operation**[2]		**Events**	
Mode		Timing	Start	Stop
Analog Input				
Polled	Start	rate, dig	cmd, ana, dig	TC, ana, dig
Interrupt	Start, Stop, Status	rate, dig	cmd, ana, dig	cmd, TC, ana, dig
DMA				
Other	Initialize			
Analog Output[3]				
Polled	Start	rate, dig	cmd, dig	TC, dig
Interrupt	Start, Stop, Status	rate, dig	cmd, dig	cmd, TC, dig
Other				
Digital Input				
Polled	Start	null	null, cmd	null, TC
Interrupt				
Other	Initialize			
Digital Output				
Polled	Start	null	null, cmd	null, TC
Interrupt				
Other	Initialize			
Counter/timer				
Polled	Start	rate		
Other	Initialize			
Device				
Other	Initialize, Capabilities			

Logical Channels

Driver*LINX* uses Logical Channel numbers to specify physical analog, digital, and counter/timer channels on the DAS-8. Generally, the Logical Channel number is the same as physical channel number described in the manufacturer's documentation, but, in some cases, additional Logical Channels are defined by Driver*LINX* to specify combinations of physical channels. For instance, Counter/Timer Logical Channel 4 refers to physical counter/timers 1 and 2 which have been chained together using an external jumper on the DAS-8's analog connector. Refer to the appropriate subsystem sections below to see how Logical Channels are mapped onto the hardware.

[2] All subsystems allow the *MESSAGE* operation which is not shown in the table.

[3] Analog output is only supported by the DAS-8/AO.

Analog Input Subsystem

The DAS-8 contains one A/D converter and an 8-channel multiplexer. The A/D channels can be used in differential (channels 0-7, DAS-8PGA and DAS-8/AO) or single-ended (channels 0-7, all models) configuration. The inputs are labeled **CH0 HI IN** [37], **CH0 LO IN** [19], etc., and are located on the D-type male connector that projects through the rear panel of the computer.

Logical Channels

Analog Input Logical Channels 0-7 correspond to physical analog input channels 0-7.

Analog Input Initialization

Initialization of the analog input subsystem aborts any active interrupt data acquisition tasks and tests the counter/timer connections. The counter/timer connections test is used by Driver*LINX* to check the validity of later Service Requests and to select the internal commands for programming the counter/timers. For instance, if a Service Request specifies Logical Counter 3, but Driver*LINX* did not detect the required external connections during subsystem initialization, the Service Request will be rejected with an error. Similarly, if *INTERRUPT* mode is requested, but Driver*LINX* did not detect previously a clock signal at the **INTERRUPT INPUT** [24] line, the request will fail.

Three tests are performed every time the analog input subsystem is initialized. First, Driver*LINX* determines if Logical Counter 3 is available by testing if physical counter 2 can clock physical counter 1. Second, the presence of Logical Counter 4 is detected by testing if physical counter 2 clocks physical counter 0. Third, Driver*LINX* determines if Logical Counters 2, 3, or 4 are connected to the **INTERRUPT INPUT** [24] line. The first counter that clocks this input stops the test. All these tests may fail to detect a valid connection if the gate inputs of any counter are held low during testing.

External Triggering

Applications may require that the start of a data-acquisition task be synchronous with the occurrence of an external signal. Data-acquisition devices commonly support this requirement by providing an input line for a TTL-level signal that will start, or trigger, the data-acquisition process on a specified state change of the signal line. Once data collection starts, further transitions on the external trigger are usually ignored.

The DAS-8 hardware does not support external triggering as defined in the last paragraph, but you can obtain almost the same results using one of two techniques. A hardware trigger can be implemented by using the gate input of the appropriate clock counter. Holding the gate low will inhibit the clock from counting. An active high level will start the clock. Note that pulling the line low during the counting will suspend the counting operation again unlike the preceding definition of an external trigger. When using this technique, you do not need to modify the Service Request as Driver*LINX* cannot control or detect the gate inputs to the DAS-8 counters.

The second technique to implement an external trigger uses a Digital Start Event. The Service Request can specify that data-acquisition starts when one or more digital input lines change value. Driver*LINX* supports this technique only on the 4-bit digital input port (digital Logical Channel 0) . External triggering using this technique requires software polling so the data hold time for a trigger must be at least 3 ms on a 10 MHz AT, to at least 20 ms on a 4.77 MHz XT.

External Clocking

Some applications require that the timing, or pacing, of data-acquisition sampling be synchronized to an external clock signal. To do this with the DAS-8, you must connect the external clock to the **INTERRUPT INPUT** [24] input on the external analog D-connector on the board's mounting bracket. To inform Driver*LINX* that external clocking is requested, use either a Rate Timing Event on Logical Channel 6 with external clock input or a Digital Timing Event that specifies "Ext Clock" as the digital mask, zero as the pattern, and "not equals" as the test. External clocking can be used in either *POLLED* or *INTERRUPT* modes.

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-8 returns the 12 A/D bits left shifted by four bits and the low-order four bits are zero. 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 ±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 $\frac{20\text{ v}}{4096 ´ 100}$ = 48.8 mv.

Input Voltage	A/D Binary Value
-100 mv	0
0 v	2048
48.8 µv	2049
+100 mv	4095

A/D Conversion Delay

The maximum A/D conversion delay depends on which Logical Counter is specified and the frequency of the timing clock. For Logical Counter 2, the longest delay between samples is 65,535 times the clock frequency. For Logical Counters 3 and 4, the longest delay is 4,294,967,295 times the clock frequency. With a 1 MHz clock, Logical Counter 2 can delay 65.5 ms while Logical Counters 3 and 4 can delay 1 hour, 11 minutes, and 35 seconds.

The minimum conversion delay for Logical Counters 0, 1, and 2 is two clock tics and for Logical Counters 3 and 4 is four clock tics. However, the 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

The DAS-8 does 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-8/AO boards contain two channels of multiplying 12-bit D/A converters. The outputs are labeled **DAC 0 OUT** [20] and **DAC 1 OUT** [1] and are located on the D-type male connector that projects through the rear panel of the computer.

Analog Output Initialization

Initialization of the Analog Output subsystem loads a value into both D/A channels to force the output voltage to zero.

External Triggering

See the External Triggering section under the Analog Input Subsystem.

External Clocking

See the External Clocking section under the Analog Input Subsystem.

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 Analog Output Range switches (S3 and S4). For example, if you use the 0-5 v range, the D/A output range will be as follows:

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

Binary value	D/A Output Voltage
0	0
2048	2.5
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.

D/A Conversion Delay

D/A conversions are timed using either software polling of the counter/timer or interrupts from **INT IN** [24] 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-8 models cannot support direct hardware timing of D/A output.

The minimum D/A conversion delay is 2 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 on interrupt latencies 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-8 has a custom 3-bit digital input port and a 4-bit digital output port (Logical Channel 0). The custom input channel is configured as **IP1**, **IP2**, IP**3** and the output channel as **OP0**, **OP1**, **OP2**, **OP3**.

Driver*LINX* implements all digital I/O on the DAS-8 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.

Digital Input Initialization

Initialization of the Digital Input subsystem does not change Logical Channel 0.

Digital Output Initialization

Initialization of the Digital Output subsystem resets the value of all output ports to zero.

Digital I/O Conversion Delay

Digital I/O is not clocked.

Digital I/O Data Lost

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

Counter/Timer Subsystem

The DAS-8 uses an Intel 8254 Programmable Interval Timer which consists of 3 internal 16-bit counters, Counter 0, Counter 1, and Counter 2. The clock, gate, and output connections of Counters 0 and 1 are undedicated and available on the DAS-8 external connector. Counter 2's gate and output connections are externally available also, but the clock input is internally connected to either the PC bus clock or a 1 MHz crystal oscillator.

Logical Channels

The Logical Channels for the counter/timer subsystem correspond to the following physical channels:

Logical Channel	Physical Channel
0	Counter/timer 0
1	Counter/timer 1
2	Counter/timer 2
3	Counter/timers 2 and 1 (connect **Ctr 2 Out** [6] to **Ctr 1 In** [4])
4	Counter/timers 2 and 0 (Connect **Ctr 2 Out** [6] to **Ctr 0 In** [2])
5	Counter/timers 0 and 1 (Connect **Ctr 0 Out** [3] to **Ctr 1 In** [4])
6	Pseudo-channel for external clock input at **INTERRUPT IN** [24]

Driver*LINX* uses physical Counter 2 whenever the application requests an internal clock for a data-acquisition task. The timing interval generated by Counter 2 can vary from 2 to 65,535 clock tics. If a longer timing interval is needed for an application, Logical Channels 3 or 4 can be specified. Be sure to externally connect the appropriate lines as indicated in the table. The availability of Logical Channels 3 and 4 is tested whenever the analog input subsystem is initialized.

Counter/Timer Initialization

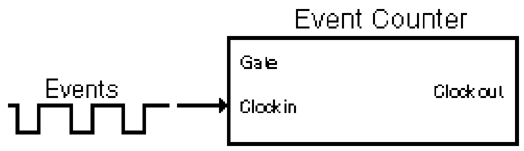
Power-up state of the counters is undefined. Initialization establishes the state of each counter as a binary rate generator with a down count of 65535.

Counter/Timer Interrupt

Counter/timer interrupts are not supported.

Event Counting

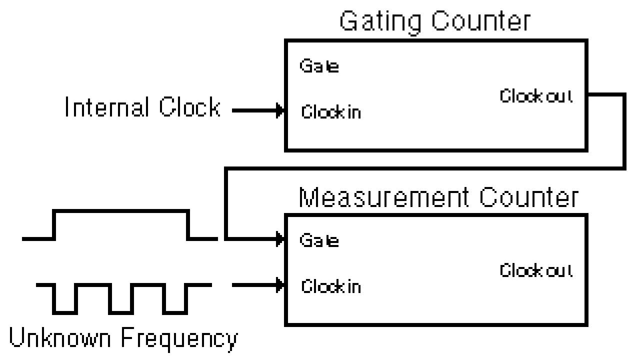
Event counting is the simplest counter/timer function. Only one counter is required and the external events to be counted are used as an external clock input signal to the counter. The application program reads the accumulated count as it deems appropriate.



As the DAS-8 allows an external clock input only for Counters 0 and 1, Driver*LINX* supports event counting only for Logical Counters 0 and 1. The events to be counted are connected to **Ctr 0 In** [2] or **Ctr 1 In** [4]. Events are active low TTL-clock pulses, where a clock pulse is defined as a falling edge followed by a rising edge. **Gate 0** [21] or **Gate 1** [22] 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-8 hardware configuration is limited. Frequency measurement requires an external connection between two counters.

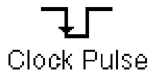


Frequency measurement requires two counter/timers used as measurement and gating counters. The unknown frequency is input as the clock source to the measurement counter. The gate input of the measurement counter is then activated for a known interval as timed by the gating counter. The gating counter 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 chose the gate time. To measure a 0.1 Hz signal, the gate time should be approximately 3 minutes.

In Driver*LINX* for the DAS-8 models, Logical Counter 0 is the measurement counter and Logical Counter 3 is the gating counter. The output of Logical Counter 3 (**Counter 1 Out** [5]) must be externally connected to the gate input of logical counter 0 (**Gate 0** [21]). The unknown frequency to be measured is fed to the clock input of Logical Counter 0 (**Counter 0 In** [2], active low) and **Dig Common** [11]. Remember the signal must be TTL (0 to 5 volts). Do not exceed this voltage range as damage to the counter could result. Also, Logical Counter 3 must be configured with an external jumper as indicated in the preceding table. Note that, unlike the technique used by Computer Board's DOS driver, connecting the gating signal generated by Logical Counter 3 to a digital input line is not required.



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 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.

In *POLLED* mode, Driver*LINX* does not return control to the calling application until the frequency measurement is completed. The measured frequency in clock tics counted during the gating period is returned the Result field of the Service Request. Overflow of the measurement counter is detected if the gating interval is too long. Frequency measurement in *INTERRUPT* mode is not supported by the DAS-8.

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

Pulse Output

The DAS-8 counter/timers can be used to generate TTL pulse trains. Due to the design of the DAS-8, the 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 2 can only generate square waves using the 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 Counters 3 and 4 can generate square and variable duty cycle waves using the internal clock. However, arbitrary duty cycles are not possible. The *onCount* must be ³ off count (= *period* - *onCount*) and the ratio $\frac{\textit{onCount}}{\text{off count}}$ 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-8 timers do not allow exact compliance with the Driver*LINX* 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 Driver*LINX* when using the DAS-8.

To implement oneshot pulse outputs, Driver*LINX* 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).

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

Time Interval Measurements

DriverLINX does not support Time Interval Measurements using the DAS-8.