

Chapter P: The Output Module

	<u>Page</u>
P-1. Introduction and Organization of the Output Module	P-1
P-2. Organization of the Output Module	P-1
Elements of the Output Protocol	P-2
Synchronizing Output Across Channels	P-4
Coordinating Output and Acquisition	P-4
P-3. The Output Control Window	P-4
Configuring the D-A Output Hardware	P-5
Adjusting the Output "Throughput", or Update Rate	P-5
Turning Channels On and Off, Naming Them, and Editing Their Contents	P-6
Operating the Output Module With and Without Data Acquisition	P-6
Accuracy of Output/Acquisition Synchrony	P-7
The Output Always Feature	P-7
P-4. The Edit Output Window	P-7
The Delay Interval	P-8
The Train Interval	P-9
Copying Parameters from One Segment to Another	P-10
Copying Parameters from One Channel to Another	P-10
Synchronizing Trials Across Channels	P-10
P-5. The Output Display Window	P-11

Datapac 2K2 User's Manual, Ver 3

Chapter P: The Output Module



Document Release Date: March 21, 2002

P-1. Introduction and Organization of the Output Module

The Output module is designed to produce up to four channels of output in the form of pulse trains or voltage levels to control external equipment and present stimuli of various sorts. With the high speed 16-bit D-A hardware the module is designed to drive, signal amplitudes are adjustable with an accuracy of 0.3 millivolts over a ± 10 volt range at aggregate throughputs up to 100,000 hertz. The module is therefore capable of a high degree of precision in both the time and amplitude domains.

P-2. Organization of the Output Module

The basic organization of the Output module is shown in Figure P-1. As illustrated, the Output module is largely composed of three main windows: The Output Control window, the Edit Output window, and the Output Display window. As the name implies **Output Control window** is the primary window for controlling output. It is used to turn on and turn off individual output channels, to adjust the "throughput", or rate at which the output is updated, and to determine the inter-trial interval, the total duration of the output, and whether the output is synchronized with data acquisition, among a few other things. Details can be found in Section P-3. The **Edit Output window**, which is accessed from the Output Control

window by clicking on the  Edit button, is used to construct and edit the details of the output signal in each channel -- i.e., the delay, duration, frequency, pulse width, and the base and pulse amplitudes of each pulse train. Details can be found in Section P-4. The **Output Display window**, which can be accessed from either the Output Control or Edit Output windows by clicking the  Display button, provides the opportunity to preview the output signals that you design in a graphical format so that you may review them before you produce them. Details can be found in Section P-5.

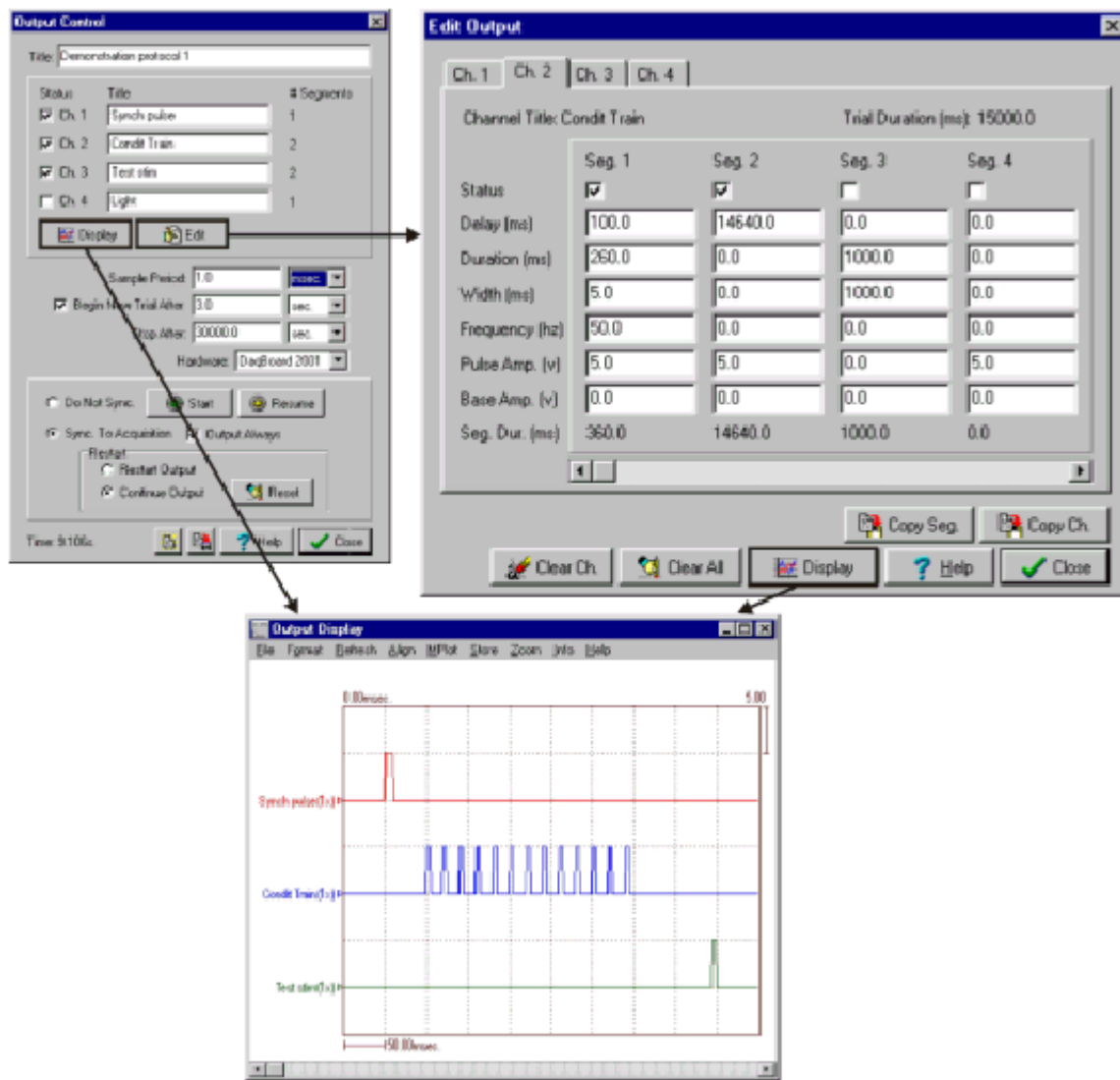


Figure P-1. Basic organization of the Output Module.

Elements of the Output Protocol

The Output module is designed in a way that assumes that the "*output protocol*" -- i.e., the output signal that is actually produced over time -- is composed of a series of **trials** that are presented one after another in a sequential fashion. A rather simple example is illustrated in Figure P-2. The characteristics of the output on any one trial are identical to the characteristics on any other trial. In other words, the trial is the primary repeating unit in the output protocol.



Figure P-2. A simple example of an output protocol showing three successive trials.

A trial can be composed of up to 20 individual **segments** which are themselves presented sequentially. Two separate examples are shown below in Figures P-3 and P-4. When designing voltage level outputs, each segment can be used to present a different voltage level for a different interval. When designing pulse train output, each segment can be used to present a pulse train of a different delay, frequency, duration, amplitude, and/or pulse width.

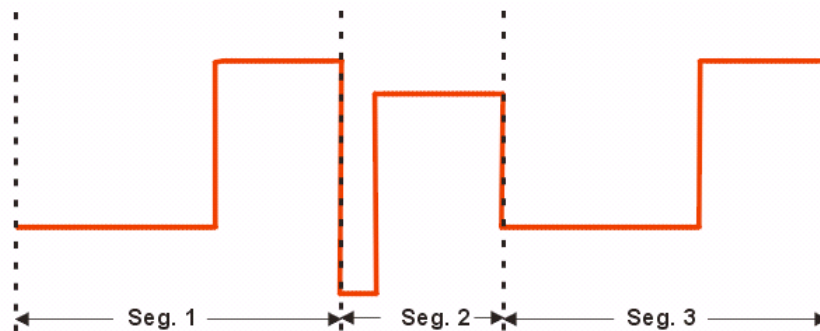


Figure P-3. An example of trial segments when designing voltage level output.

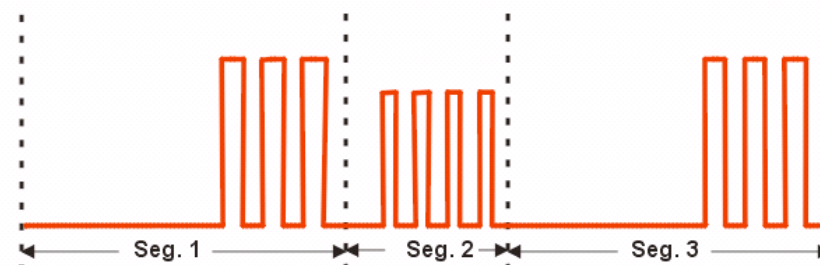


Figure P-4. An example of trial segments when designing pulse train output.

Each segment can be considered to be composed of two separate parts. One part is the **delay interval**, the other is the **train interval**. When designing pulse train output, the train interval is used to define the frequency, duration, and amplitude of the pulse train as well as the width of each pulse. The delay interval is used to separate individual trains in time. An example is shown in Figure P-5. The delay interval is completely defined on the basis of only two parameters -- a duration parameter and an amplitude parameter. Consequently, it is ideal for use in defining voltage levels. The train interval can also be used to define a constant voltage level as well (i.e., by defining the "train" as a single pulse whose width is the same as the duration of the train interval itself). Either interval can be assigned a duration of 0.00, by the way, and thus effectively eliminated from the segment. However, when they both exist the delay interval always precedes the train interval.

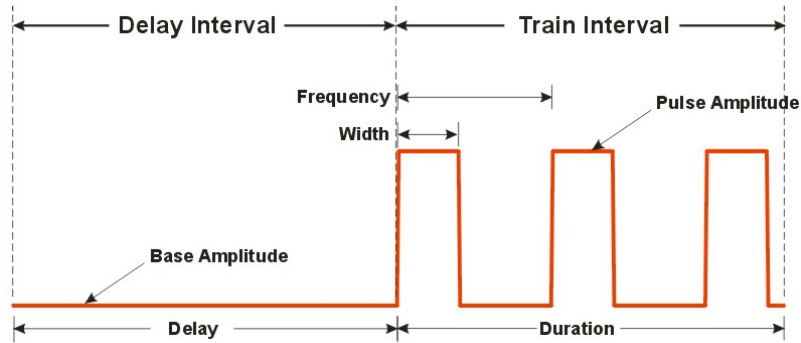


Figure P-5. Components of a trial segment.

Synchronizing Output Across Channels

As described above, a trial consists of up to 20 individual segments. Since the segments are presented sequentially, the duration of a trial is the sum of the durations of the segments that have been defined for it. It was also mentioned that trials are presented sequentially. Thus, under normal circumstances, a new trial begins the instant the previous trial ends. These facts are true for each individual channel as well. Consequently, if the duration of a trial differs from one channel to the next, subsequent trials will start at different times on each channel. Sometimes that is desirable, but many times it is not. There are several ways to prevent channel desynchronization. For details, see Section P-4.

Coordinating Output and Acquisition

The Output module can be operated in isolation to produce only outputs, or in combination with the Data Acquisition module to produce output in synchrony with acquisition. The Output Control Window is the interface for making the decisions as to whether output will be generated alone or in synchrony with acquisition. For details, see Section P-3.

P-3. The Output Control Window

The **Output Control window** is the primary interface for controlling output. An example is shown in Figure P-6. If you are using the Output module to produce output signals independently of data acquisition, this is the place where you turn the signals on and off (via the **Start** and **Resume** buttons). Alternatively, if you intend to produce output in conjunction with acquisition, this is the place where you set that up as well (by selecting the **Synch to Acquisition** radio button). The Output Control window is also the interface that controls the more fundamental characteristics of the output signals. Specifically, you use it to turn on and off individual channels (by checking or clearing the **Status checkboxes** associated with each channel), to adjust the "throughput", or rate at which the output is updated (by adjusting the sample period parameter), to determine the inter-trial interval (via the **Begin New Trial After** option), and the total duration of the output (by adjusting the **Stop After** value). Details are provided in the remaining paragraphs.

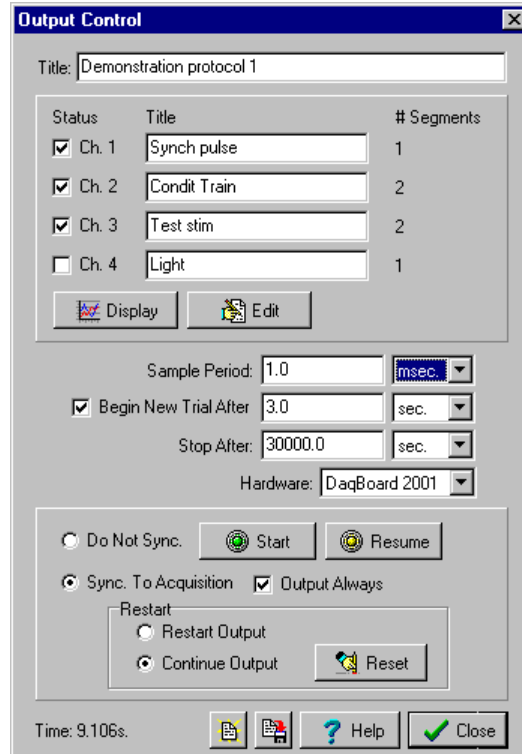


Figure P-6. An example of the Output Control window.

Configuring the D-A Output Hardware

The Output module is designed to drive the Iotech DaqBoard 2000 family of D-A and A-D/D-A hardware devices. Select the device you have installed using the **Hardware** list box. The DaqBoard hardware driver software must also be installed and the board must be configured correctly within the driver software. Details are provided in Appendix A of your Datapac 2K2 manual.



Adjusting the Output "Throughput", or Update Rate

The "output throughput" refers to how frequently the output signal is updated on each channel. The throughput is determined by the **Sample Period** value. Specifically, the sample period determines the interval between successive updates in each individual channel. For example, a sample period of 1 msec. means that the output signal in each channel is updated every 1 millisecond, or 1000 times per second.


There are a couple of issues to keep in mind when selecting a sample period value. First, the value that you select for the sample period will limit the accuracy with which the time values associated with the output signals themselves can be generated. The time values in question are the **Delay**, **Duration**, **Width**, and **Frequency** values described in Section P-4. For example, if you set a pulse width equal to 5 msec, it would be inappropriate to set your sample period to 2 msec, 0.8 msec, or any other value for which 5 msec. is not an integer multiple. Said in another way, all time values that you select should be integer multiples of the sample period value. If they are not, then the program cannot represent the value correctly, and will use instead the sample period multiple that is closest to but less than the desired time value. For example, if the sample period value is set to 2 and a width parameter is set to 5, the width that is actually presented on output will be 4 because 4 is the multiple of 2 that is closest to but less than 5.

The second issue concerns the computer processing time required to generate output. As the sample period gets smaller the processing time required to produce the output increases. And as the time required to produce the output increases, there is less time available for other tasks -- such as performing acquisition. On a faster computer, however, the impact on processing time is negligible. This concern becomes more of an issue on slower computers.

Turning Channels On and Off, Naming Them, and Editing Their Contents

Individual output channels are turned on or off by checking or clearing the **Status** checkbox associated with each. The Status checkboxes are located in the "channel summary" section of the Output Control window. The channel summary section also reports the **Title** associated with each channel, and the number of trial segments (**# Segments**) currently defined for them. The channel titles are merely verbal descriptions to help you identify the nature of the intended output. To add, delete, or edit trial segments, click on the  button. Doing so opens the Edit Output window. To view the outputs defined for each channel in a graphical display, click on the  button to open the Output Display window.

Operating the Output Module With and Without Data Acquisition

The Output module can be operated in isolation, or in combination with the Data Acquisition module. To produce outputs alone (that is, not in conjunction with data acquisition) first click the **Do Not Sync** radio button. Then to begin output, click the  button. The Output Active window, shown in Figure P-7, then appears to indicate that the output is active and reports the elapsed time. To terminate output, click the **Stop** button in the window, or press the **Esc(ape)** key on your keyboard.

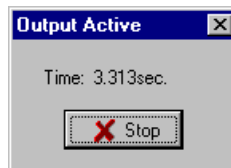





Figure P-7. The Output Active window appears when output is started from the Output Control window.

When you are ready to resume output, you have two choices. Click the  button again to begin the output protocol from the beginning. Alternatively, click the  button to continue the output protocol from the point where it was last stopped. Note that when the Output Active window appears, time is measured from the point where the output was resumed. However, the time that has elapsed up to the point when the protocol was last interrupted is also reported in the lower left hand corner of the Output Control window.

To operate the Output module in combination with the Data Acquisition module, first click the **Sync to Acquisition** radio button. Operation of the Output module is then under the control of the Data Acquisition module. Output then begins when an acquisition session begins, and terminates when the session ends. The options in the **Restart** section below the Sync to Acquisition button then determine whether the output protocol restarts or resumes from its point of last termination when additional acquisition sessions are performed. Specifically, click the **Restart Output** radio button if you want to begin the output protocol from the beginning each time an acquisition session is run. Likewise, click the **Continue Output** radio button if you wish to resume the output protocol from the point of its last termination each time an acquisition session is run. Note that the elapsed time of the output protocol is reported in the lower left hand corner of the Output Control window under these conditions as well. You have to return the focus to the Output Control window (i.e., click on it) before the value is updated,

though. Finally, while the Continue Output option is in effect, you can manually reset the output to its starting point by clicking the  button.

Accuracy of Output/Acquisition Synchrony

When the Synch to Acquisition option is in effect both the output and acquisition hardware circuitry are signaled simultaneously at the start of an acquisition session. However, both circuits may not react simultaneously. Consequently, there may be a noticeable lag in the relative onset times of the two operations. The lag time is sensitive to the output sample period, the acquisition sample period, and the number of active output and input channels. Consequently, whenever you change one of those parameters the lag time may also change. That's the bad news. The good news is that as long as the sample period and channel parameters stay the same, the lag time remains constant from one acquisition session to the next.


The Output Always Feature

The Output Always feature allows the Output module to remain active even when the Output Control window (and therefore the Output module itself) is closed. Of course, this feature is only relevant when the **Sync to Acquisition** feature is selected, because that is the only condition under which outputs can be controlled from outside the Output Control window.

The primary purpose of the Output Always feature is to allow synchronized operation of output and acquisition within a macro (macros allow only one module to be open at a time). But it can be used outside of a macro as well, and has the same effect in either case. Specifically, when the **Output Always** checkbox is checked, outputs are generated during acquisition sessions even if the Output module is closed. Again, this assumes that the **Sync to Acquisition** feature is selected. When the Output Always check box is cleared, outputs are produced only when the Output Control window is open.

P-4. The Edit Output Window

The Edit Output window is the interface for constructing and editing the detailed characteristics of the output signals. An example is shown in Figure P-8. Note that the window shown in the example contains four tabbed sections, each one of them containing the parameters associated with an individual channel. Presently, only the parameters associated with Channel 2 are shown.

 There are four tabs in this example because the currently installed hardware device supports four output channels. Other hardware devices support only two output channels. If a two channel hardware device was installed the window would contain only two tabs.

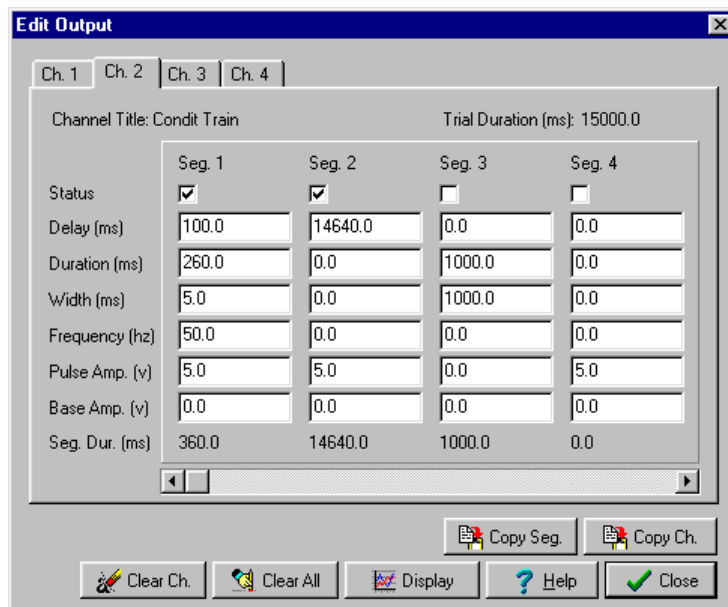


Figure P-8. An example of the Edit Output window.

The information contained in each tab is arranged in a series of columns. Each column reports the parameters associated with an individual segment. Recall that the output on any given channel is composed of one or more **trials** where each trial is composed of up to 20 individual **segments**. Because of space limitations the parameters associated with only four segments are shown at a time. The scroll bar, located along the bottom of the tabbed section, allows you to adjust the display to show the information associated with the desired segments.

One of the parameters associated with each segment is its **Status**. Specifically, the status checkbox determines whether the associated segment is included in the trial or not. When the checkbox is checked segment is included. When the checkbox is cleared the segment is omitted.

It is useful to think of each segment as being composed of two parts: a **delay interval** and a **train interval**. Details are provided in the following paragraphs. The duration of the delay interval can be set to zero, effectively eliminating it from the segment. The same is true of the train interval. Thus, when desired, a segment can contain only a delay interval or a train interval. But when both occur, the delay interval always precedes the train interval.

The Delay Interval

The delay interval is so named because its primary purpose is to delay the beginning of the pulse train from the beginning of the segment. But it also has another purpose: it can be used to present a voltage level. In fact, if you intend to present a series of voltage levels rather than pulse trains, it is often easiest to use only delay intervals. As illustrated in Figure P-9, the delay interval is completely defined on the basis of two parameters, **Delay** and **Base Amp**. The Delay parameter specifies the duration of the delay interval. The Base Amp parameter specifies the level of the output during the delay interval. If you want to eliminate the delay interval from the segment, set the Delay parameter to 0.00.

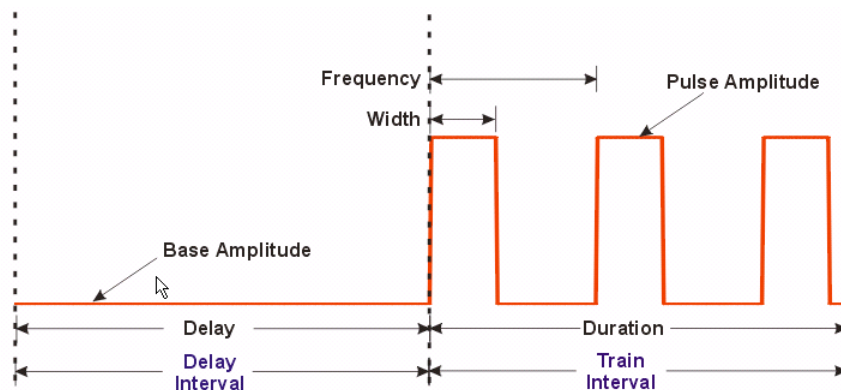



Figure P-9. Components of a trial segment.

The Train Interval

The train interval is so named because its primary purpose is to present a series of pulses. However, it too can be used to present a constant voltage level. As indicated in Figure P-9 presented above, the output during the train interval is determined by the interaction of the values defined for the **Duration**, **Width**, **Frequency**, **Pulse Amp**, and **Base Amp** parameters. The **Frequency** parameter determines the pulse frequency, or the inter-pulse interval, measured from the onset of one pulse to the onset of the next. The entered value is interpreted in units of hertz, or pulses per second. Thus, a value of 10 means that pulses are presented at a rate of 10 per second. The interpulse interval is therefore 100 ms. Note that a frequency value of 0 (zero) has a special meaning: it means that only one pulse is presented throughout the duration of the train interval. Regardless of the selected Frequency value, the first pulse is always presented immediately after the termination of the delay interval.

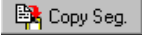
The **Pulse Amp** parameter determines the amplitude of each pulse, measured in units of volts. Thus, a value of 5 means that the amplitude of the output during a pulse is 5 volts. Similarly, the **Base Amp** parameter determines the amplitude of the output *between* pulses. Thus, a value of 1 means that the amplitude of the output between pulses is 1 volt.

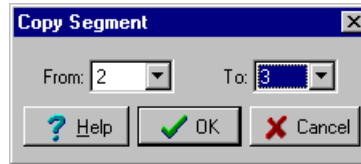
 Note that the Base Amp value determines both the amplitude of the output during the delay interval and the interval(s) between pulses in the same segment.

The **Width** parameter determines the duration of each pulse. The entered value is interpreted in units of milliseconds. Thus, a value of 10 means that each pulse in the train is 10 msec in duration. Note that if the entered width value exceeds the inter-pulse interval (defined by the Frequency parameter), the level of the output will remain at the amplitude defined by the Pulse Amp parameter. This, by the way, is the easiest way to use the train interval to produce a voltage level output. An alternative method is to set the value of the Width parameter equal to or greater than the Duration parameter.


Finally, the **Duration** parameter determines the total duration of the train interval. The entered value is interpreted in units of milliseconds. Thus, a value of 500 means that the train interval is 500 milliseconds in duration. When the train interval terminates the output protocol proceeds to the next segment in the trial if there is one, or to the next trial if there are no additional segments. It is important to understand that the train interval ends when the interval specified by the Duration parameter elapses, regardless of the output level at the time. What this means is that if the output is in the middle of a pulse when the duration interval elapses, the pulse is truncated. For example, say you define the Frequency as 10, the Width as 20, and the Duration as 110. Under such conditions the first pulse begins at 0 msec into the duration interval, then terminates 20 msec. later. The second pulse begins at 100 msec into the duration interval but terminates 10 msec later -- even though the Width parameter indicates that the pulse duration should be 20 msec. Why does this happen? Because the specified Duration interval has elapsed.

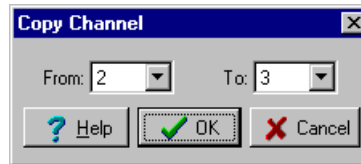
Copying Parameters from One Segment to Another

To copy the parameters you have established for one segment to another segment, select the  button. Doing so opens the window shown below. In the **From:** box indicate the number of the segment that you wish to copy the parameters from. Likewise, in the **To:** box, indicate the number of the segment that you wish to copy the parameters to. Then select the OK button to return to the **Edit Output** window.



Copying Parameters from One Channel to Another

To copy the parameters you have established for one channel to another channel, select the  button. Doing so opens the window shown below. In the **From:** box indicate the number of the channel that you wish to copy the parameters from. Likewise, in the **To:** box, indicate the number of the channel that you wish to copy the parameters to. Then select the OK button to return to the **Edit Output** window.



Synchronizing Trials Across Channels

Output is conducted on a trial-by-trial basis. Unless the output is terminated or restarted beforehand, a new trial begins the instant the previous one ends. The duration of a trial is the sum of the durations of the segments that have been defined for it. Because different channels can contain different numbers of segments with different durations, the duration of a trial may be different from one channel to another. Consequently, it is possible for trials to become desynchronized -- i.e., for one trial to terminate and a new one begin at different times in different channels. Sometimes that is desirable, but many times it is not.

There are several ways to prevent channel desynchronization. The most direct way is to make sure that the trial durations of all channels are exactly the same. To help you keep track, the **Trial Duration** is reported in the top right corner of the tabbed section associated with each channel. Likewise, the duration of each active segment in the trial -- i.e., the **Seg. Dur.** -- is reported at the bottom of each column of values.

Although equalizing trial durations across channels is the most direct way to prevent desynchronization, it is also the one that is the most difficult to achieve and maintain. After all, as soon as you change the duration of any one interval within any one segment you change the duration of the trial. So if you shorten one interval you have to lengthen another by the same amount in order to keep the trial duration the same. That gets a little tedious if you change things around a lot. And it's even more tedious if you simply want to change the inter-trial interval.

Fortunately there are two alternative methods for maintaining trial synchronization across channels that are much easier to use. The first is to use the **Begin New Trial After** option in the Output Control

window. When this option is checked a new trial begins in each channel after the specified period of time. For example, if the entered value is 15 sec, a new trail is begun in all channels every 15 sec. regardless of whether the trial duration established for each channel has been attained. Of course, the idea is to ensure that the trial duration of each channel always exceeds the Begin New Trial After value.

The second method for maintaining trial synchronization is to use the **Sync to Acquisition** feature with the **Restart Output** option, then employ the Delay feature in the Data Acquisition module to generate trials.

P-5. The Output Display Window

The Output Display window provides the opportunity to view the characteristics of the defined output protocol in a graphical time series display. An example is shown in Figure P-10. The procedures involved in initializing and manipulating an output display are essentially identical to those of a standard time series display and will not be reviewed here.

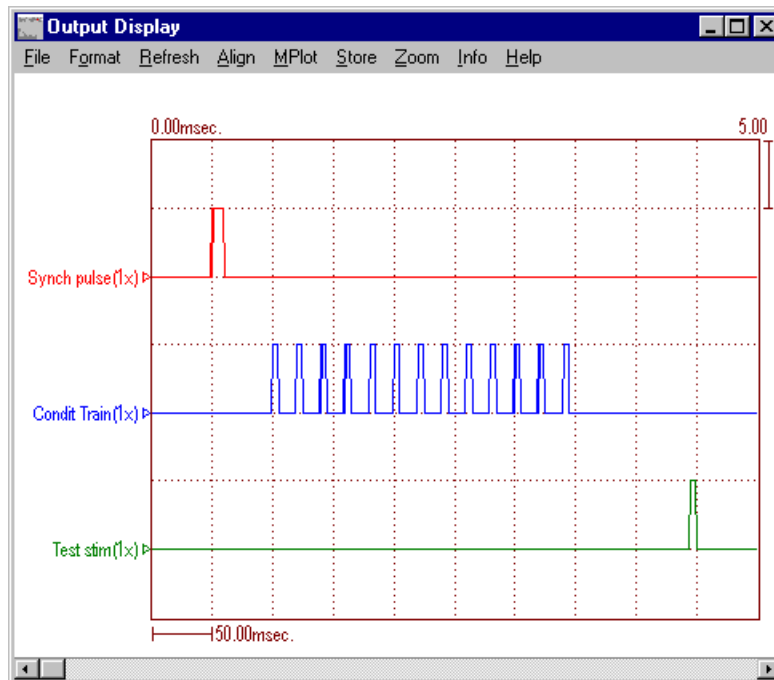


Figure P-10. An example of the Output Display window.

With a few notable exceptions, the output display is a very accurate representation of the true amplitude and time course of the output protocol. It is important to keep in mind, however, that the output display plots the values that are sent to the hardware device. It does not monitor the integrity of the hardware device or the ultimate fate of the signals in any way. Consequently, if your cables aren't connected correctly (or aren't connected at all), or if your hardware is malfunctioning, the fate of your signals may be very different than how they are represented in the display.

Additionally, and perhaps most notably, it is important to keep in mind that only the transition points in the signal -- i.e., the points in time when the output signal is updated -- are accurately represented in the display. In the actual output signal the amplitude remains at the indicated level until the next time it is updated. Then, assuming the amplitude changes at the next transition point, it changes to the new value very quickly (i.e., in less than a microsecond). These abrupt changes are not represented in the display. Rather, the transition points are merely connected with lines. Consider the example shown in Figure P-

11. The example shows a situation where an output signal is updated once every 10 milliseconds (i.e., at every division of the horizontal grid). The blue broken line shows the signal that is actually generated -- a square-wave pulse with a latency of 50 milliseconds and a width of 10 milliseconds. The signal arises from baseline to its peak value within a microsecond. Likewise it falls from peak to baseline within a microsecond. The amplitude remains stable in between. The red line shows the very same signal as it is plotted in the output display. The reason why the output signals are drawn this way is because the drawing algorithm is shared by many different display contexts. In all contexts except this one it makes perfect sense to draw the signals this way. Only in this context is it a little confusing. Anyway, now that you are aware of the situation, you shouldn't be fooled.

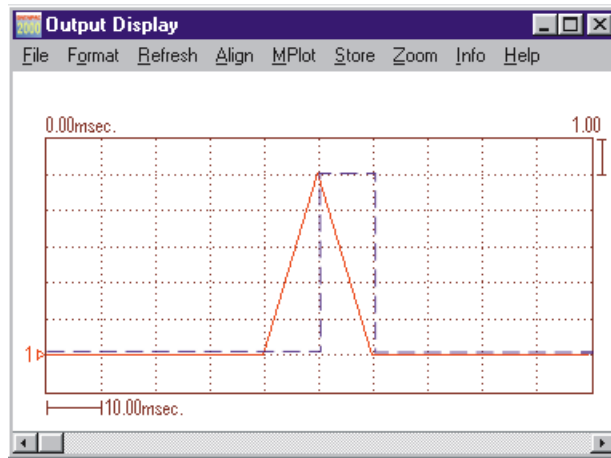


Figure P-11. Comparison between the way a signal is represented in the Output Display window (solid line) and the way it is actually generated (broken line).