DAQtimer

Manual

DAQtimer is an alternative to Labview designed to operate a National Instruments PCI6221 card and microEye CMOS camera using a simple scripting language. It is designed almost exclusively for control of the digital and analog output lines on the NI card, for the purpose of automation of experiments. It allows output of arbitrary waveforms.

This manual covers setting up and programming DAQtimer scripts for control of behavioral experiments on larval zebrafish.

DAQtimer is written by Harold Burgess (NICHD, haroldburgess@mail.nih.gov). Initial code development was in the laboratory of Michael Granato (U. Penn). Jeremy Magland (U. Penn) and Neelima Panth (NICHD) have contributed to code development.

9/15/2010 version 1.03

Table of Contents

1. Installation

1. 1. Hardware requirements

Computer
PC running Windows XP, preferably quadcore. The major constraint is that it should have two PCI slots. One is for the PCI-6221 card (below). The other is needed to accomodate the PCI card for the high speed camera. While DAQtimer is not tied to a particular high speed camera (anything that accepts a TTL trigger is fine), the two cameras that we use (DRS Lightning RDT/1 and Fastcam-x 1024 PCI) both require a full-length PCI slot. Full length slots are very rare in computers and the only computer that we are sure accomodates these PCI cards is the DELL PWS T3400.

Turn off as many services as possible. Uninstall all unnecessary software. PCs are not designed for real time operation!

High speed camera
We use:
- Fastcam-x 1024 PCI from TechImaging
- DRS Lightning RDT/1 from DEL Imaging

For optics we use a 50 mm macro lens (Cat #SI5028MDGNAF) with a C-mount adapter (Cat #gbcmn) both from B&H Photo Video.

For mounting the camera we prefer the Bogen/Manfrotto Mini Repro Copy Stand (Cat #BO1700) from B&H but this is very hard to get now. So we also use a Beseler CS-14 Copy Stand (Cat #BECS14) from Adorama.

We build a cheap enclosure using rods (Cat #94435A344) and vinyl fabric (Cat #8810K145) from McMaster Carr, or you can purchase a more sturdy enclose from ThorLabs or Newport.

National Instruments
DAQ card: PCI-6221 (Cat # 779066-01) - NOT the 37 pin or USB model
Cable: SHC68-68-EPM Cable (2m) (Cat # 192061-02) or equivalent
Connector block: BNC-2110 (Cat #777643-01) or BC-2120 or equivalent.

Bruel & Kjaer
Vibration shaker (Cat #4810)
Power amplifier (Cat #2718) - you can also use a cheaper one if the linearity of the response is satisfactory

You'll also need:

A. To stabilize the shaker somehow. We had the workshop attach it to a 3" square piece of brass: McMaster Carr (Cat #8951K661). Best to mount slightly off center so the shaker can be positioned vertically or horizontally.
B. A rod and holder for the Petri dish. The Penn machine shop has custom made these in the past (contact Michael Granato) or any machine shop should be able to do this.

MicroEye Camera (optional)
For imaging and motion trigger capability, you need a microEye camera (model UI-154xLE-M). Available from 1stVision (Cat #IDS-1545LE-M).

Lighting
For lighting we use Luxeon K2 Stars (Cat #5027-PW14) with a BuckPuck 1000mA DC LED Driver (PCB Mount) (Cat #3021-D-E-1000) and a Fraen Medium Beam Lens With Holder (Cat #FHS-HMB1-LLK2-H). These are nice because you can digitally control the light activation from the PCI card and add a potentiometer to manually adjust intensity. However assembly is beyond the scope of this manual but not too hard. Alternately, you can buy mounted LEDs with a current controller from ThorLabs (we haven't done this because of cost, but their MCWHL1 looks about right) for the same effect.

For infra-red lighting we use Cat #YY_IR100 from YY Trade Inc, removing the array from the housing and disconnecting the light sensor.

1. 2. Computer preparation

To prepare the computer for DAQtimer it is worth trying to optimize its performance, as DAQtimer will attempt to output waveforms in real time. This manual assumes the user has reasonable familiarity with Windows XP and will not cover this in detail but the some key items.

1. As computers from the manufacturer come preloaded with software that contributes to overhead, it is a good idea to reinstall the operating system if you are comfortable with that. This is also a good time to partition the hard disk (or install a new hard disk) so that you have a drive that will be exclusively used for Data storage. It will save you a lot of time if you can simply reformat the Data drive when it is full, rather than try to individually delete folders. Do not install superfluous items like MSN explorer, outlook, media player, messenger, games.

2. Open Control panel -> Display
- Desktop Tab: turn off background image.
- Screensaver Tab: set to (None)
- Appearance: Effects: turn all off

3. Right click My Computer, select properties, then Advanced tab
- Performance settings: Visual effects tab: select adjust for best performance
- Performance settings: Advanced tab: adjust for best performance of programs

4. Right click My Computer, select properties, then System Restore tab
- Turn off system restore on all drives

5. Right click My Computer, select properties, then Automatic Updates tab
- Turn off automatic updates. Note: doing this and other steps will leave the computer unsecured, so it is better if it is not connected to the internet.

6. Right click My Computer, select properties, then Remote tab
- Turn off Remote assistance and Remote desktop

7. Open Control panel -> Security center
- firewall off, autoupdates off, alerts off. Again, this is not a good idea for computer connected to the internet.

8. Open Control panel -> Power options
- Never turn off monitor or hard disks
- Never enter system standby.

9. On the drive you are using to save data, right click, select properties
- General tab: turn indexing OFF

1. 3. Install High speed camera

As per manufacturer's directions. For the DRS lightning be sure that they send you the correct drivers for multicore computers.

1. 4. Install National Instruments PCI-6221

1. Install the software from the disks provided. Use all the default options.

2. Shut down computer and install the card in a free PCI slot.

3. Reboot computer

4. Use NI cable to connect PCI card to the connector block.

Use BNC cables to connect the connector block to your outputs:
- A0.0 to amplifier/pump or whatever requires analog control.
- P0.0 camera trigger
- Other P0.x go to light sources or anything else digitally controlled.

5. Check that the NI board is properly installed using the NI test panel software (see manufacturer's manual for instructions).

1. 5. Install uEye camera (optional)

1. Download and install the microEye software drivers (select complete installation)

2. Plug in camera and follow the directions on the screen to install the drivers.

3. Check the camera is operating using the uEye Demo program provided with the camera.

1. 6. Install DAQtimer

1. Download and install IDL. Download the full installation as it includes the IDL virtual machine which will run the et.sav file. You will be prompted for a license, but ignore this as you do not need a license to run the et.sav file.

2. Create a directory C:\DAQtimer.

3A. If you **did** install a uEye camera, place these files in the DAQtimer folder: et.sav, EventTimer.exe, uEyeConfig.eye

3B. If you **did not** install a uEye camera, place these files in the DAQtimer folder: et.sav, EventTimer.exe_nocam. Then, change the name of the second file to EventTimer.exe

4. Test by opening et.sav. In the text window type:

```
begin input
text hello world
delay 3000
end
```

Click the **Run** button. A window should open, with the worlds "hello world" and close after 3 seconds.

5. If you have a uEye camera, check that you can operate the camera from DAQtimer using File->Setup camera.

2. Scripting overview

The et.sav program allows you to write simple scripts in order to achieve millisecond coordination of high speed camera activation with analog and digital stimuli. Scripts are written in the large text window. Multiple tabs allow you to keep several programs ready to run.

Scripts are completely case insensitive - before compiling, all case information is stripped. Also tabs and spaces are stripped before compiling.

To comment the text, use two forward slashes //. After // all remaining text on the line is ignored.

Before compiling the script the compiler does a brief syntax check. This is not comprehensive so you should carefully look over the script to be sure it will perform as expected. Note that the compiler only checks text inside function blocks that are actually called. Function blocks that are never called are ignored.

3. Script Control

3. 1. Routines: BEGIN and END

Scripts are composed of one main routine called 'input' and subroutines which can be called anything. All routines are bracketed by the commands **BEGIN** [routine name] and **END**.

So the main routine, from which input begins, must look like this:

```
BEGIN input
        line1
        line2
        line3
        etc.
END
```

To repeatedly execute a particular subroutine, use the **RUN** [subroutine name] [number of repeats] command. For example, to execute a subroutine called loopA 51 times you would have a command
RUN loopA 51

The subroutine, as for the main routine would look like this:

```
BEGIN loopA
        line1
        line2
        line3
        etc.
END
```

3. 2. Quick routines

Occasionally you may want to quickly write and execute a simple script (for example to turn a pin or read a signal). If your program does not have a 'BEGIN input' and a 'END' line, then DAQtimer will automatically add these as the first and last line of code.

For example a program consisting of the single line
ON 1
would turn on digital line 1 then terminate.

3. 3. Timing

3. 2. 1. DELAY

Fixed timing of commands relative to each other is provided by the **DELAY** [ms] command. For example, to insert a 384 ms delay before initiation of the next instruction, use the command
DELAY 384

Delays are completely responsible for creating the timing of output. Output commands occur at the time created by the series of delays preceding them and do not contribute any time themselve. For example in a script:

```
BEGIN input
        DELAY       100
        TRIGGER
        WAVE0       pulse
        VOLT0       1
        DELAY       100
        TRIGGER
        WAVE0       pulse
        VOLT0       1
```

the trigger, waveform and volt commands will all occur at time point 100 ms, and 200 ms into the script. In fact, this would probably generate an error as the event timer can not handle such closer spaced operations. You would be better off writing

```
BEGIN input
        DELAY       100
        TRIGGER
        DELAY       10
        WAVE0       pulse
        DELAY       10
        VOLT0       1
```

A 50 ms delay is automatically added at the start of the script as the event timer does not do well if the first output has to happen at 0 ms after you press run. In addition, there is a 50 ms delay at the end of the script. However, if you last command is a waveform that takes longer than 50 ms to execute, you should add an additional delay to ensure that it is executed before event timer quits.

3. 2. 2. Digital wait: WAITD

The **WAITD** command pauses program execution until the specified digital line goes high or low. The syntax is **WAITD** [line #] [0 *or* 1]

Example:
WAITD 2 0
Will pause program execution until digital line P1.2 is at 0 volts.

Note that the line number is always on port 1.

3. 2. 3. Analog read and wait: READA, WAITA

The **WAITA** command pauses program execution until the specified analog line is greater or less than a value. The syntax is **WAITA** [line] [< *or* >] [value].

Example:
WAITA 1 < 0.1
Will pause program execution until analog line 1 is set to less than 0.1 volts.

For calibration purposes, you can write a small script using the **READA** command to reveal the current value of an analog line. The syntax is **READA** [line]

Example
READA 1
Will show on screen *Line 1: 10.0000* if line 1 is connected to a 10 volt source.

3. 2. 4. Motion Trigger: MOTION

To use the motion trigger, first the uEye camera needs to be set to watch part of the field (see section on Camera configuration). The configuration program also gives useful information about the region of interested selected: the minimum and maximum pixel values and the average value. The coordinates of the lowest pixel are also displayed (x,y), however note that as the image is rotated in memory, they are not simple x,y positions on the screen.

The command **MOTION** [pixel] [distance] establishes a delay until a fish enters or moves inside the watch area. Fish are recognized only by being darker in color than the rest of the watch area. Use the Camera configurator to work out the pixel value of a fish in your lighting conditions. Thus the *pixel* parameter specifies the darkness of a fish and the *distance* parameter is the threshold for movement that will exit the delay.

If no pixel in the watch window is darker (smaller value) than the *pixel* value, the **MOTION** command waits until a dark pixel appears (ie a fish swims into view).

If there is a sufficiently dark pixel in the watch area, **MOTION** waits until it moves the distance in pixels specified by *distance* to exit the delay.

3. 2. 5. Clock time: UNTIL

The **UNTIL** command allows you to delay execution of a script until a specific time. The time must be in twenty-four hour time format XX:XX:XX (hours:mins:seconds). For example

```
BEGIN INPUT
        UNTIL       21:30:00
        ON          1
        UNTIL       21:30:10
        OFF         1
END
```

will turn digital line on from 9:30 pm until 10 seconds after 9:30 pm.

3. 4. TAB

The **TAB** command will cause your program to run the script in the designated tab at the completion of the currently running program

For example, if tab 1 contains this script

```
BEGIN INPUT
```

```
        TEXT        Starting tab 1
        TAB         2
        DELAY       2000
END
```

When it runs, it will print 'starting tab 1' then pause for two seconds, quit, then immediately run the script in tab 2. Of course if tab 2 contains the instruction TAB 1, then it would then run tab 1 at the end. This can lead to infinite loops, so use with caution!

4. Digital output

4. 1. ON and OFF

The **ON** and **OFF** commands set the digital output lines high and low respectively. For example:

```
BEGIN input
        ON          1
        DELAY       100
        OFF         1
END
```

This will set digital line 1 to 5 volts for 100 msec, then return it to 0 volts.

4. 2. TRIGGER

The **TRIGGER** command sets digital line 0 high for 1 ms then low again. It is therefore just a shortcut for writing

```
ON 0
DELAY 1
OFF 0
```

You can also send a trigger to other digital lines by specifying the line after the trigger. For example to send a 1 ms long high-low pulse to line 2:

TRIGGER 2

You can also change the duration of the pulse - this is sometimes required to trigger devices running at lower speed (for example, DRS cameras operating at 25 fps are not successfully triggered by a 1 ms high/low pulse). For a 33 ms long trigger on line 0, you would write

TRIGGER 0 33

4. 3. Pin1/2 buttons

Radio boxes at the bottom of the screen provide quick control of digital lines 1 and 2.

5. Analog output

DAQtimer controls the analog output lines using two buttons and four analog out commands: WAVE0, WAVE1, VOLT0 and VOLT1.

5. 1. VOLT0/1

VOLT0 and VOLT1 set the analog voltage on analog output lines 0 and 1 respectively. For example to set analog out line 1 to +10 volts use the command:
VOLT1 10

5. 2. WAVE0/1

The WAVE0 and WAVE1 commands allow you to send an arbitrary waveform to analog output lines 0 and 1 respectively. You need to specify a file containing the waveform. For example, if you have a waveform in a file called testpulse.txt and wish to send it out on analog out line 1 you would write:
WAVE1 testpulse

Waveform files are plain text files where the first line specifies the sample rate and each subsequent line contains a voltage level. The wave command reads the waveform file, remembers the sample rate, then sends out the voltages on the analog line at the required rate until the end of the file is reached.

You can manually create a waveform file to sculpt an arbitrary waveform. It must be placed in the C:\DAQtimer\outs directory. If the file has a **.txt** extension, then this must be specified as part of the WAVE0 or WAVE1 command, for example
WAVE0 mywaveform.txt

For backward compatibility, this version of DAQtimer also accepts the command **WAVEFORM** as a synonym for WAVE0.

5. 3. SINE

The script can create a waveform for output using the SINE command. The SINE command specifies the amplitude, frequency, ramp up duration in ms, the hold duration in ms, the rampdown duration in ms and the sample rate, in that order:
SINE [amplitude][freq][rampup][ms][rampdown][sample rate]
The total duration of the waveform is the ramp up duration plus the hold duration plus the ramp down duration. You must always specify the ramps, although you can set them to zero to eliminate ramping.

So to create a waveform file called myfirstpulse containing a 10 volt waveform at 200 Hz, lasting for a total of 100 ms, with 9 ms ramps in both directions, to be output at 30 kHz, you would write:
SINE myfirstpulse 10 200 9 82 9 30000

When you execute the script, or if you press the **Test** or **Graphic** buttons, a file is created in the C:\DAQtimer\outs directory containing the waveform. You can open the file in a text editor.

To check that the waveform really looks like you intended, place the cursor anywhere on the line containing the SINE command and press the **Wave** button. The window will show what the waveform created looks like. Use File → Print Graphic to print this window.

5. 4. Analog buttons

The An0=0 and An1=0 buttons can be used to quickly reset the A0 and A1 lines to 0 volts respectively. These provide fast access to the lines when voltage output needs to be urgently cancelled.

6. Video input

Daqtimer is primarily designed to send trigger signals to high speed cameras for synchronizing video imaging with stimulus presentation. It also contains a limited capability for digital imaging by controlling a microEye CMOS camera. This is intended to add two capabilities:

1. Motion trigger: insert a delay in the script until a change is detected within a region of interest within the field of view of the uEye camera.

2. 3D imaging: taking a snapshot of the imaged area from an alternate perspective to allow calculation of Z-position.

6. 1. Camera configuation: WATCH

Before the camera can be used for snapshots or as a motion trigger, it must be set to watch part of the camera field. The **WATCH** command sets the camera to start recording, with the format:

WATCH [x position] [width] [y position] [height] [exposure] [gain] [camera]

For example:
WATCH 0 1280 0 1024 10.0 50 b
will set camera B to watch the entire recording field, with an exposure time of 10 milliseconds, and a gain of 50. A window will open showing a still image of the area observed. Use the **SNAP** <ref> command to update the window.

Only a single watch window can be specified during a program - you can't watch one area of a program and then change to a new area of interest on the fly.

Use File → Setup camera to open a utility that will help you define the region of interest and generate an appropriate WATCH command. If the current script contains a WATCH command, the area of interested defined in that command, and exposure and gain settings will be applied in the camera configuration utility.

In the camera configurator, follow these steps:

1. Enter live mode (using the **Live** button). Set the exposure (in msec) and percentage gain to adjust the image. As per the microEye capabilities, exposure can be set from 0.037 ms to 983 ms. Gain can be set from 0 to 100. Exit live mode (press **Stop**).

2. Define a region of interest within the uEye imaging area. The uEye gives a 1280x1024 image, however you will rarely want to watch the entire field. The region of interest can be set by typing the x and y coordinates and the width and the height in the text boxes. Note that as the uEye only accepts certain step values for x, y, width and height, values you type in will be adjusted to match the uEye specifications. Alternately, left click and drag the mouse inside the image window to select an area of interest. The coordinates are automatically entered into the text boxes below the window.

The **Snap** button refreshes the image in the window with a snapshot of the current coordinates (those in the text boxes).

The **Live** button sets the configurator to constantly refresh the image - this is useful to adjust the position of the camera or the testing arena. During live imaging the time is updated in red on the bottom left corner. Click the **Stop** button to return to a still image - you may need to hold it down for a couple of seconds.

The **Max** button resets the image coordinates to fill the entire viewing area of the camera.

When the viewing area, exposure and gain are correctly set, you can copy/paste the command in the **Set** box back into your script (highlight all the text in the box, press Ctrl-C to copy, quit the configurator and press Ctrl-V to paste into your script).

6. 2. Taking snapshots

6. 2. 1. Setting filename: SETDIR, SETFILE

The **SNAP** command saves a copy of the watch region to the hard disk. Images are sequentially numbered from 0 upto 99999 and the number is appended to the base filename. You can specify the base filename in two ways:

1. As a full pathname to the save directory and file. For example:
SNAP c:\images\mutant
This will save a file in the c:\images directory, called mutant_00000.jpg. **SNAP** will always add a number to the end of the filename, incremented with successive **SNAP** commands. It is ok to use directory and file combinations with successive **SNAP** commands.

2. Use the **SETDIR** [directory] and **SETFILE** [name] parameters to specify the filenames. **SETDIR** specifies a directory in which to place the images, and **SETFILE** specifies the base file name. These two commands can appear anywhere in the script - however all instances after the first will be ignored - you can only specify a single directory and base file name per script.
For example, if you place these two lines at the top of the script:
SETDIR d:\images
SETFILE mutant
Then the SNAP command will save the file as d:\images\mutant_00000.jpg.

Specifying a filename with **SNAP** over-rides any **SETDIR/SETFILE** commands issued. In addition, specifying a filename with **SNAP** that lacks directory information will cause DAQtimer to use the **SETDIR** directory. Thus it is possible to have multiple output filenames. For example, if you have a program:
SETDIR c:\newimages
SETFILE pencil
SNAP c:\zebra
SNAP d:\zebra
SNAP lettuce
SNAP
Then you would create these four files (in order):
c:\zebra_00000.jpg
d:\zebra_00001.jpg
c:\newimages\lettuce_00002.jpg
c:\newimages\pencil_00003.jpg

6. 2. 2. SNAP

Once the watch region is set and a filepath specified, use the **SNAP** command to save a copy of the watch region to the hard drive.

If you are using SNAP for 3d imaging, it is important to consider how much time the snapshot takes. This is going to be more than just the exposure time, as there is also the time taken to transfer the image to the computer memory then to disc. Use the Configuration → Camera check command. The line 'Snapshot duration:' calculates the total time to take the image and transfer to memory. For a 512x512 image it typically takes about 13 ms to transfer to RAM and another 5 ms to save to disk, so around 20 ms for a 1 ms exposure. For the sake of speed when the **SNAP** command is used to save images to disk, the contents of the watch window are not refreshed.

To refresh the watch window use the command **SNAP <ref>** which will update the watch window on the screen without saving its contents to disk. For example, to open a live watch window for 10 seconds you could use this program:

```
BEGIN loop
        SNAP        <ref>
        DELAY       50
END

BEGIN input
        WATCH       0 512 0 512 1.00 1 a
        RUN         loop 200
END
```

6. 2. 3. Background subtraction

It is often useful to take a preliminary shot as a background image. If the **SNAP** command is followed by BG, then a background shot will be taken but not saved, rather kept in memory. All subsequent images will be adjusted by this original background shot, such that any pixel equal to or greater than the background image will be set to 255 (white).

SNAP <bg>

You can also alter the sensitivity of background by using a value

SNAP <bg> 10

This means that all pixels greater more than 10 units below the background image will be set to white.

6. 3. TRACK objects in video window

The **TRACK** command tracks objects inside the watch window. The syntax is

TRACK [grid][pixel intensity threshold][object size][video rate][duration of tracking]

Grid refers to the n x n size of the grid.

Pixel intensity threshold specifies the maximum intensity of pixels that are part of objects to be tracked.

Object size sets an approximate size in pixels of objects to be tracked

Video rate species the number of milliseconds between frames to be tracked and displayed. Note that this is limited by the download speed of the microEye to the camera, so tracking a 512 by 512 pixel image faster than about 15 ms intervals (ie 60 frames/second) is not possible and will generate error messages. For very small windows (eg 100 by 100 pixels) you should be able to track up to 250 frames per second.

Duration of tracking is the number of frames that will be tracked. If this is set to 0, then tracking will continue as long as the script is open.

The output of the tracking is to the files specified by the **SETDIR** and **SETFILE** commands. If no **SETDIR** or **SETFILE** commands are specified, output is to the c:\DAQtimer\tracking directory.

This file contains a tab-separated list containing:
Frame #, Fish #, x-pos, y-pos

7. Screen output

7. 1. TEXT

The TEXT command allows you to print a block of text on the screen. For example
TEXT hello world
will print *hello world*.

To print a blank line, just use TEXT command by itself.

To suppress the line break at the end of the text output, place a colon **:** in the text
TEXT hello:
TEXT world
will print hello: world

To bring the cursor to the start of the line, use '<<'
TEXT hello:
DELAY 1000
TEXT <<
TEXT world
Will print 'hello:', wait for a second, then replace 'hello' with 'world'

To clear the console window, use the command "<clr>"
TEXT <clr> hello
Will print hello at the top of the console window.

7. 2. COUNT, CLEAR

The **COUNT** command will increment the specified counter and show the current value. For backwards compatibility, the command COUNTER is synonymous. For example
COUNT 3
will show the value in the console window, then increment the counter. So the first time you call count, the result will be:
c3:0

By default, each call to the count command places the output on a newline. To suppress this and repeatedly print the counter on the same line use the '<' character. For example:
COUNT 3 <
COUNT 3 <
Will output c3:0 and c3:1 at the same position on the screen

The CLEAR command resets the specified counter to zero.

7. 3. ELAPSE, TIME

The **ELAPSE** command shows the time since the begining of execution of the script.

For example

```
BEGIN input
        DELAY 1995
        ELAPSE
END
```

Will show Elapsed: 1.995 s in the display window.

Similarly the **TIME** command shows the clock time on the computer.

7. 4. WARN

The **WARN** command will change the screen color to attract attention.
WARN 1 turns the console red.
WARN 0 turns it back to black.

8. Video output

8. 1. Psychophysics utility

File → Setup Psychophysics (or use the button at the bottom right of the window) to open a psychophysics utility for creating simple visual stimuli for playback by DAQtimer. The utility lets you test, then save up to 10 video sequences.

To use the utility, first select a stimulus type from the Stimulus dropdown menu. Then use the Parameters menu and the text boxes to enter information to modify the stimulus presentation. To check the video, press the lower left button (marked with the name of the currently selected stimulus). This will play it in the video output window (note there is an artefactual 'black flash' during playback using the psychophysics utility that does not occur during playback with DAQtimer).

Once the video looks suitable, save it to a folder using the Save dropdown menu. At any time you can playback the contents of a video folder using the buttons 0 to 9 above the playback button.

8. 2. Playback window

You can resize the playback window:
1. By selecting an option from Parameters → Window Size
2. By resizing its frame
3. When working with user defined image stacks, selecting Parameters → Window Size → Fit user image so that the window automatically adjusts to the size of the user image stack.

At any time select Parameters → Window Size → Show current to see the size of the playback window.

Move the window to a suitable location using the Up/Down/Light/Right buttons in the psychophysics utility (located in a panel to the right of the message panel). Each click moves the window by one pixel, but after clicking on a button you can press and hold the spacebar to move it quickly.

Once the window is suitably positioned and sized, use File → Save window setup to save the configuration. You can reload the configuration using File → Load window setup. Note that if you save a configuration with the name 'Default', this will be loaded when the psychophysics utility is opened and the windows placed accordingly.

8. 3. PLAY

After making a suitable video, you can play it back during script execution using the command **PLAY**.

For example

```
BEGIN input
        PLAY 5
END
```

will play the video set as number 5.

The video is played in the video output window, which is opened alongside the psychophysics utility window. You should resize and position this window using the psychophyics utility before playback with DAQtimer.

To keep a hardcopy of the stimulus settings and window positions, use File → Print settings.

9. SERIAL (Com1) Port

The command **SERIAL** will output the text following to the com1 serial port. If the text is followed by a full stop, then a carriage return (ascii 13) will be sent immediately following the text.

For example

```
SERIAL c111
```

Will send the string c111 to the serial port, and print the response on the screen

```
SERIAL c111 .
```

Will send the string c111<cr> to the serial port. Note this is 'c111' followed by a space, then a fullstop.

DAQtimer will then print any response from the serial port to the command window.

20. Testing scripts

To check that a script contains no errors, press the **Test** button. This performs simple syntax checking. It also checks that any waveform files that are part of a **WAVE0** or **WAVE1** command actually exist, or are created in the C:\DAQtimer\outs directory.

As a second method to check the script will operate as expected, press the **Graphic** button. A window will appear graphically displaying the flow of the script. Use File → Print Graphic to print this window. This feature is also useful because it calculates the total number of triggers that will be generated, and the total duration of the script.

The sequence of commands that DAQtimer will actually execute is stored in the input.txt file in the C:\DAQtimer\outs directory. This is a text file where each line shows the time a command will be executed, then the command itself. You can open and examine the file directly.

Another way to debug scripts is to execute them after checking the Configuration → Debug mode option. In this mode detailed messages about the execution of each element are printed on screen including the planned start time of an event, deviation from the actual start time and duration of the event. Briefly, a line like this:

```
68596 DIGITAL 1 1 0 0 0 0 0 0 [start] err=-0.05 ms
.    [Done] Dur=1.08 ms
```

Means that at 68,696 msec after the beginning of script execution, digital lines 0 and 1 were set high, all the others low. Note in this case line 1 was already high and line 0 was being briefly set high, so both are shown high. The actual start time deviated from the planned start by -0.05 ms, and the command took 1.08 ms to execute.

21. Executing scripts

Press the **Run** button to compile the script in the currently selected tab and initiate its execution. While the script is running, a window will open on the screen where the script output is printed. You can stop execution of the script at any time by closing the window. Doing so will leave analog and digital output lines at whatever setting they were at when script execution was terminated. When the script reaches the end, the output window will close.

There are two switches uner the Configuration menu that alter screen output during execution:

Debug mode - outputs information for debugging script execution. In this mode, the DOS window is left open at the end of the execution of the script. Debug mode is normally off (ie unchecked).

Verbose mode - when on, screen output commands (COUNTER) are displayed on the screen. When unchecked, this is suppressed. Verbose mode is typically kept on.

The third option IDL compile should always be checked, and provides legacy support for older versions of the program where the script was compiled by a separate program RunEventScript.exe.

DAQtimer requires that the computer output commands in real time. This can be problematic on machines where operating system functions interrupt programs. DAQtimer therefore works best on multi core machines. You should not try to set the sample rate too high. Generally a sample rate of 100,000 will fail. A sample rate of upto 30,000 samples per second is more realistic. However for very high frequency output this could cause trouble with Nyquist theorum, setting a limit on the realistic output frequency attainable with DAQtimer.

22. File maintenance

File → Open: open a previously saved script in the currently visible tab. These are also accessible from the **Recent** menu, which contains the 10 last files opened or saved.

File → Save: save script in currently visible tab.

File → Format script: cleans up the script, removing excess spaces, capitalizing commands, inserting tabs etc. This is purely cosmetic but may make your script more readable.

File → Print script: sends a copy of the script to the printer.

File → Cleanup: deletes all contents of the C:\DAQtimer\outs directory. This is not really ever necessary, but is useful if you want to debug a script and get rid of all the extraneous files.

23. Command reference

begin [name] - start of subroutine
count [num] [<] - Display/increment counter number [num]
clear [num] - Reset counter [num] to zero
delay [ms] - Pause [ms] milliseconds
elapse - Display time since start of script execution
end - End of subroutine
motion [pixel] [distance] - Wait until pixel moved distance in watch
on [pin] - Set [pin] to 10 v
off [pin] - Set [pin] to 0 v
play [video] - Playback video [number]
read [line] - Show current analog value on line
run [routine] [num] - Repeat [routine], [num]
serial [text][.] - Output to com1 serial port
setdir [directory] - Save directory for images acquired
setfile [file] - Base file name for images acquired
sine [amplitude][freq][rampup][ms][rampdown][sample rate]
snap [file] - Save current image in camera to disk
text [text] [:] - Display text on screen
track [thresh][size][isi][duration][filename] - Track fish
trigger [pin] [delay]- Low-High-Low on [pin]
volt0 [volt] - Set analog 0 to [volt]
volt1 [volt] - Set analog 10 to [volt]
waita [line] [< / >] [value] - Wait for line to be < or > than value
waitd [pin] [dir] - Wait for [pin] to be [0 or 1]
warn [state] - Set red background for command window
watch [x0] [width] [y0] [height] [exposure] [gain] - Start camera
wave0 [file] - Output [file] on analog 0
wave1 [file] - Output [file] on analog 1
// - comment, ignore rest of line

Keyboard shortcuts are available when the cursor is in the script window:
Control-d : display window
Control-o : open new file
Control-p : print script
Control-r : run script
Control-s : save script