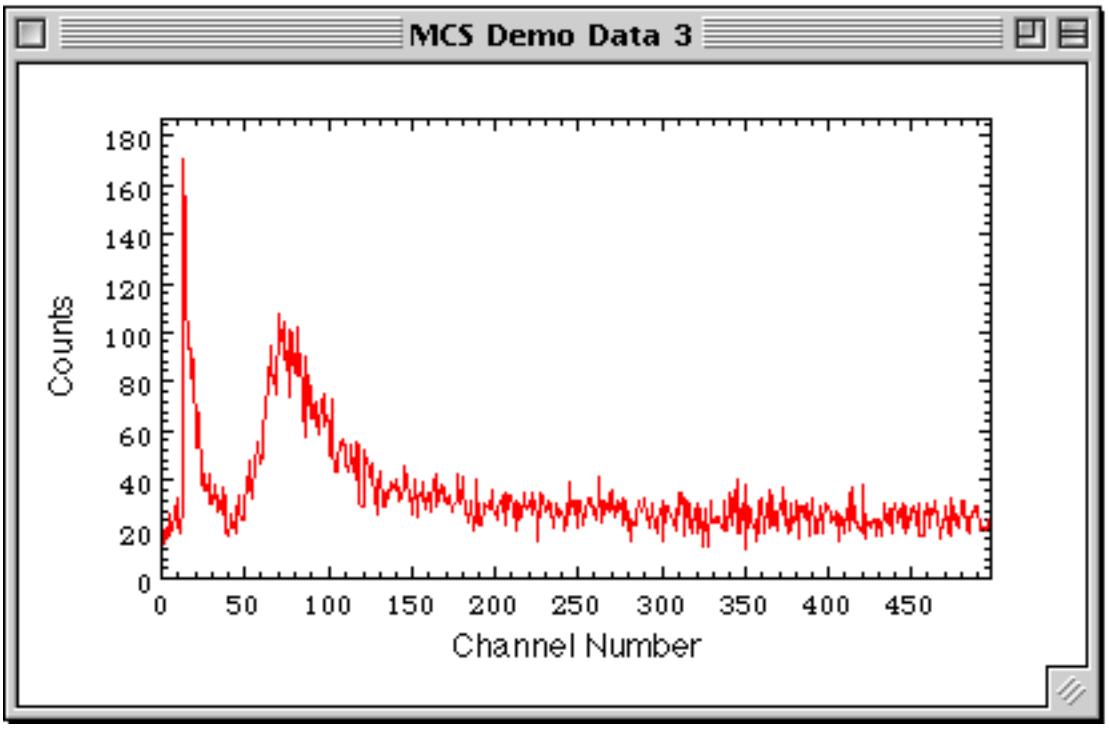
User Guide to the TurboMCS Controller



A multichannel scaler scan from a time-of-flight experiment measuring methyl fragments (CH_3) photodissociated from methyl bromide (CH_3Br) adsorbed on LiF(100).

Copyrights

Acknowledgements

First I would like to thank my wife Suzanne for her patience and support in letting me play with the computer at odd hours of the day/night to work on various pieces of this program and to get them just right. Also I want to thank my daughter Johanne for sleeping quietly on my lap as I programmed and debugged. Many thanks to various folks on the `comp.sys.mac.programmer.*` newsgroups for much assistance in figuring out the Macintosh toolbox. Also many thanks to the folks at Waters Edge Software (`http://www.interlog.com/~wateredg/`) for making the excellent ToolsPlus software and for their great software support.

Portions of the *TurboMCS Controller* application are thanks to:
Kevin Boyce at NASA for his 'Bluchert' libraries that simplify Igor Pro PPC communications
Jim Luther at Apple DTS for his MoreFiles library
Peter Lewis at Stairways Software for his PNL libraries and Mac Pascal sample code

Contact Information

```
Erik Jensen
Physics Program
University of Northern BC
Prince George BC Canada
V2N 4Z9
email: ejensen@unbc.ca
```

TurboMCS Hardware

The *TurboMCS Controller* is a MacOS™ application that controls and communicates with the EG&G Ortec TurboMCS™ multichannel scaler. All communication with the TurboMCS is done via the serial communications hardware, and all of the features of the TurboMCS hardware are available to the user.

Proper use of the *TurboMCS Controller* software implicitly requires an understanding of how the TurboMCS hardware operates and getting the hardware in your experiment set up correctly. Please refer to the TurboMCS User Manual [1] for an overview of how to set up the hardware.

Running TurboMCS Controller

1. TurboMCS General Introduction

The *TurboMCS Controller* application is a standard Macintosh application. It will run on any Macintosh that has at least a 68020 processor (or better, including PowerPC) and is running System 7 or later. A colour monitor is strongly recommended, with a size of at least 640x480 pixels (13 inch or larger). The standard memory allocation is 1 MB of RAM, though you may choose to increase this if you keep many data windows open.

1.1 Computer Requirements

A MacOS™ computer running System 7 or greater is required. It has been tested on System versions from 7.0.1 to 8.1. Both 680x0 and PowerPC versions of the controller software are supplied. The application can run in 500kB of memory and normally has an allocation of 1MB of RAM. To use the AppleScript™ facilities, the computer will need to have AppleScript installed. A colour monitor is recommended but not required, and a monitor size of at least 640x480 is recommended. To communicate with the TurboMCS hardware, the computer will require one available serial port (usually the modem port). Third party serial port cards (NuBus and PCI bus) are supported as long as they are communications toolbox compliant (most are).

1.2 Additional Features

The *TurboMCS Controller* is a scriptable application using AppleScript. To use the scripting facility you will need a script editor such as the one provided free by Apple (‘Script Editor’) or one of the commercial scripting systems (eg. Late Night Software’s ‘Script Debugger’, see their web site at: <`http://www.latenightsw.com/`>). More information on AppleScript is available from Apple Computer <`http://www.applescript.apple.com`>.

The TurboMCS Controller can communicate directly with “Igor Pro”, a data analysis and presentation package that is widely used in MacOS science and technical computing. For more information about Igor Pro, contact WaveMetrics Inc. at :

WaveMetrics, Inc.
P.O.Box 2088
Lake Oswego, OR 97035
USA
Tel.: (503) 620-3001
Fax: (503) 620-6754
Web: `http://www.wavemetrics.com`

1.3 Installation

The EG&G TurboMCS hardware should be installed as per the directions in the hardware user manual [1].

To communicate between the Macintosh and the TurboMCS hardware, a serial communications cable is needed. The type of cable required is commonly available at computer stores that deal with Macintosh computers and is the same type that works with most modems. The one you need has an 8-pin DIN connector on the Macintosh end and a 25-pin D connector (male) on the TurboMCS end. If you cannot find a suitable cable (in the lab you may need a fairly long cable), it is also possible to make your own cable.

The *TurboMCS Controller* software is installed by creating a folder on your hard drive in the desired location and copying the desired files from the floppy disk. This normally would include:

For PowerPC MacOS computers:

TurboMCS (ppc).sea

For 680x0 series Macintosh computers lacking a floating point coprocessor:

TurboMCS (68k).sea

For 680x0 series Macintosh computers that have a floating point coprocessor:

TurboMCS (68k fpu).sea

For all computers copy the following folders:

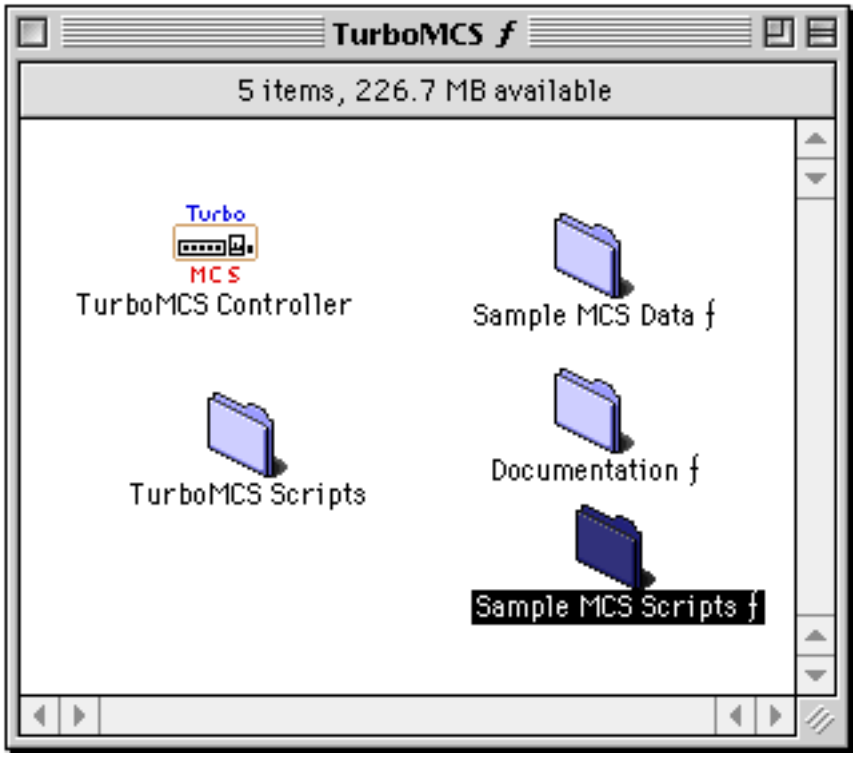
‘TurboMCS Scripts’

‘Sample MCS Data *f*’

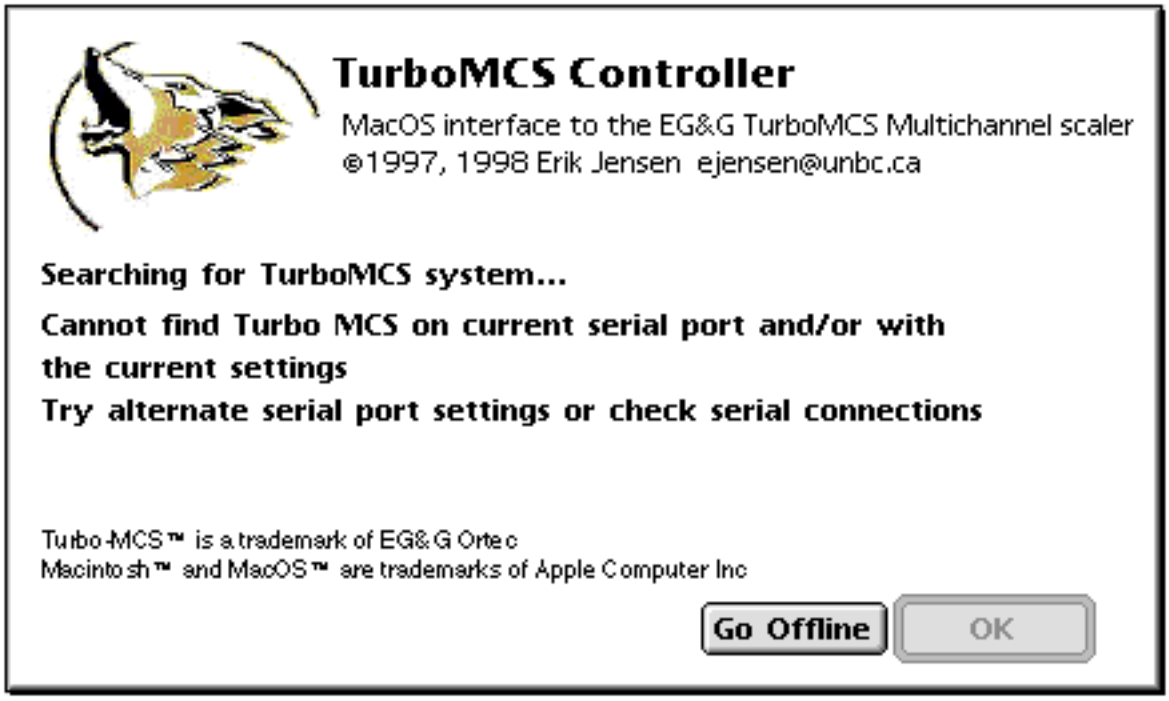
‘Sample MCS Scripts *f*’

‘Documentation *f*’

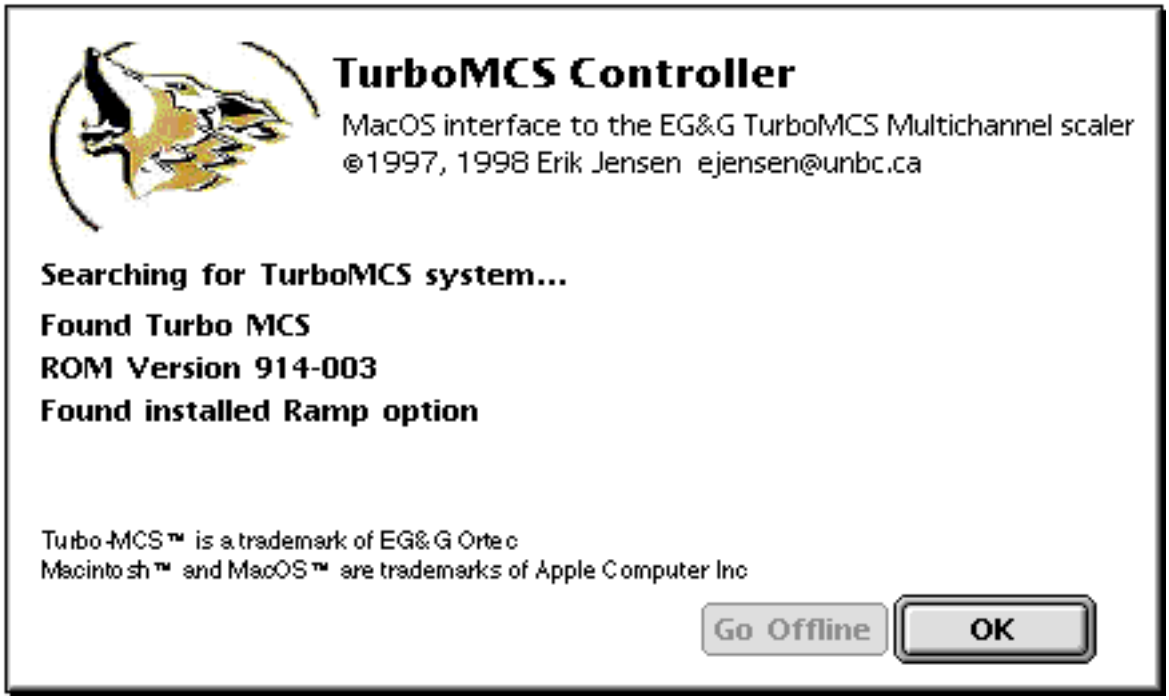
After copying the desired .sea file to your hard drive, double click on its icon to expand the file into the *TurboMCS Controller* application. You may want to rename the application “TurboMCS Controller”. When this is finished, you may double click on the icon of the *TurboMCS Controller* application to start. The folder should look something like this:



When the *TurboMCS Controller* application starts, it begins by searching for TurboMCS hardware on the default serial port. If this is the first time TurboMCS runs, it will search on the modem port. If the TurboMCS is found, the splash screen will display some information; if the TurboMCS is not found then it will tell you this and wait for you to click “Go Offline” (no commands will be sent on the serial port). To go ‘online’, you will need to have a proper serial cable connection between the Macintosh and the TurboMCS hardware. Turn on the TurboMCS power, and set up the Macintosh serial port settings in the “Serial Control” dialog (in the Acquire menu). The factory settings for the TurboMCS hardware are: 9600 baud; parity off; 1 stop bit. After selecting the desired serial port settings and choosing the appropriate serial port, select the “OK” button and the application will search for TurboMCS hardware on the selected port using the new settings. When you have found the correct settings, they will be stored by the *TurboMCS Controller* application in its preferences file. These will be used every time you run the TurboMCS application until you change them in the serial control dialog.

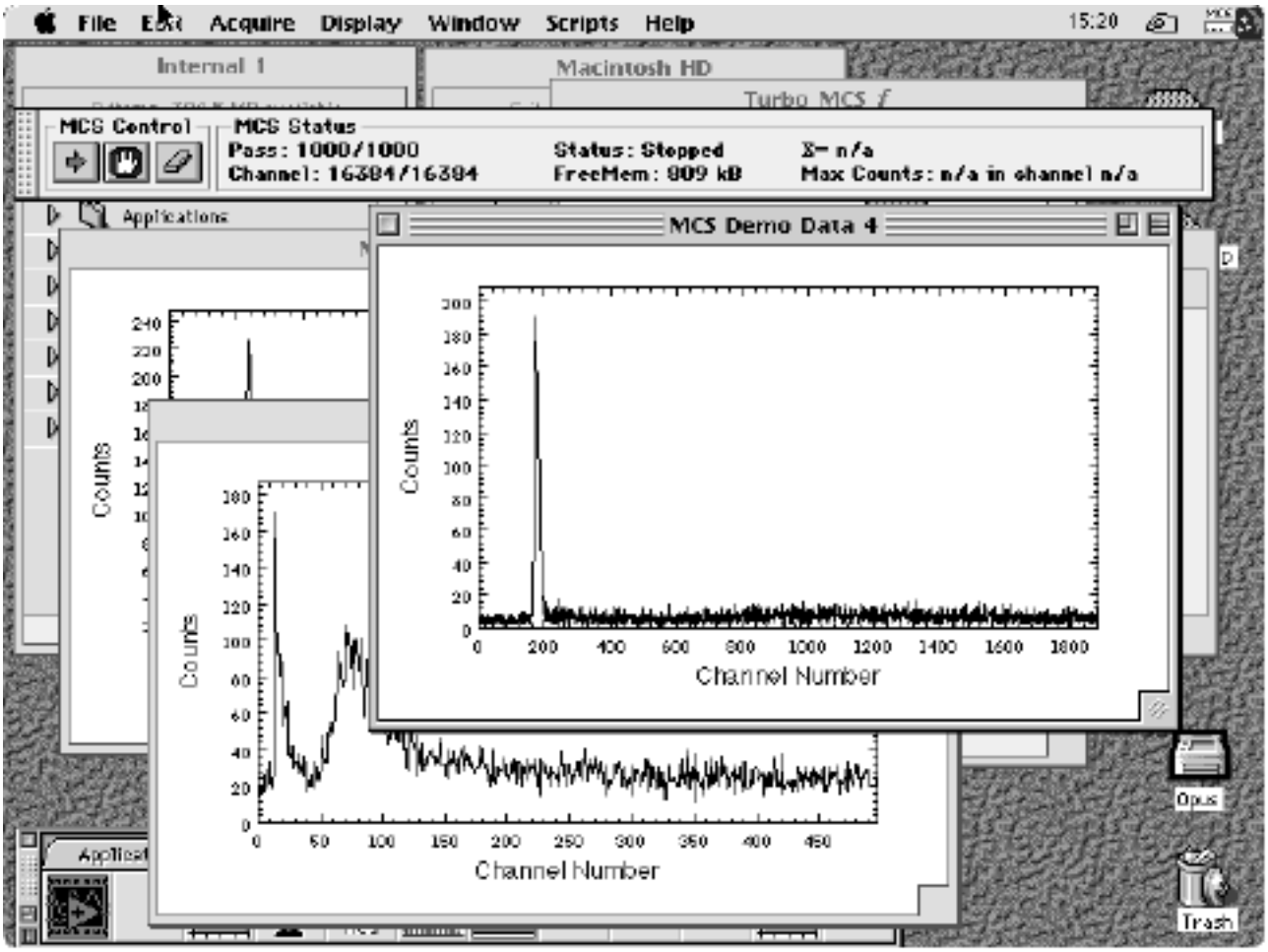


Startup splash screen if a TurboMCS is not found on default the serial port



Splash screen after startup and finding a TurboMCS on the selected serial port

When the *TurboMCS Controller* application is run, it looks for its preference file (located in '`System Folder:Preferences`') and if found, loads these settings. If no preference file is found, default settings are used. Then the *TurboMCS Controller* attempts to locate the TurboMCS hardware unit on the default serial port and if it is found, reads the current settings as reported by the hardware. If the hardware unit is not found, you may try to change the serial port settings (section 2.4.7) to establish communications. When the *TurboMCS Controller* application quits, it saves the current preference settings in the preference file in anticipation of the next time it is run.



Macintosh desktop showing the TurboMCS Controller with several open windows.

1.4 Bug Reports

If you think that you have found a bug that you can attribute to TurboMCS Controller, please let me know. Please tell me what steps reproducibly lead to the bug, what hardware you are using, and what MacOS System version you are using.

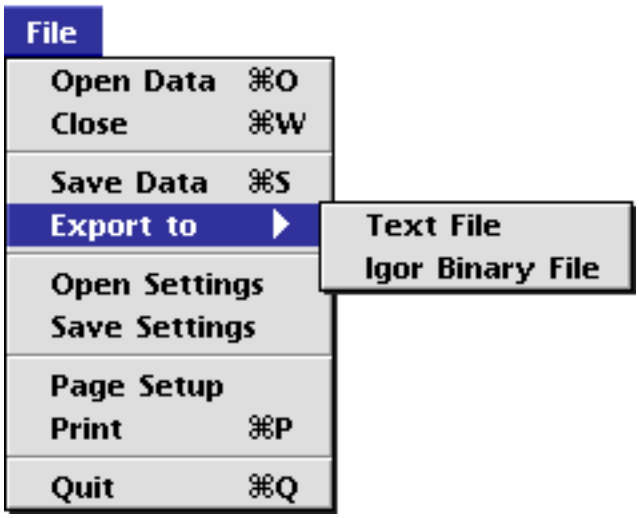
2 TurboMCS Menus

2.1 Apple (□) Menu

2.1.1 About TurboMCS Controller…

Displays a very interesting and informative dialog about the *TurboMCS Controller* application

2.2 File menu



2.2.1 Open Data (⌘-O)

Presents the user with a standard Macintosh 'Open File' dialog that allows opening of *TurboMCS Controller* data files that have been previously saved to disk. If you have not saved any of your own data yet, you can open the demo data files that accompany the software on the installation disks.

2.2.2 Close (⌘-W)

The frontmost data window will be closed. If the data in the window has not been saved to disk, you will be prompted with a dialog that allows you to either

(i) Save the data to disk

(ii) Close the window without saving the data (the data will be lost)

(iii) Cancel the close operation (data not lost)

2.2.3 Save Data (⌘-S)

Saves the data to a disk file using the standard Macintosh 'Save File' dialog. You may choose any file name that you want, though you should probably use some standard format for file names.

The data is written in a binary file format that is described in Appendix 1. The data file contains the Turbo-MCS settings used for data acquisition as well as the data points. These files can be read back only by the *TurboMCS Controller* application (or your own if you write one using the file format

information in Appendix 1). For most purposes you should store your data in this format until you need to export it (see ‘Save As Text’ (2.2.4) and ‘Send Data to Igor’ (2.5.4) commands for export options). Note that if you choose a file name that is the same as an existing file, you will be prompted if you want to replace the named file with the new one.

2.2.4 Export to Text File

Saves the data to disk, writing out the data in an ASCII format that is easily read by a number of programs. There are two parts to this text file- the header part and the data part.
The header part of the text file consists of lines beginning with two backslashes (//) followed by text describing various TurboMCS settings.
The data part of the text file consists of lines with the time value for each dwell channel, followed by a tab (ascii character 9), followed by the counts in that dwell channel, followed by a carriage return (ascii character 13).

This text data can be read into many different computer applications (eg. data analysis, presentation) such as WaveMetrics’ Igor Pro or Microsoft Excel. Note that if you choose a file name that is the same as an existing file, you will be prompted if you want to replace the named file with the new one.

2.2.5 Export to Igor Binary File

Saves the data to disk, writing in a binary file format that can be read by Igor Pro (a product of Wavemetric Inc.).

2.2.6 Open Settings

Using the standard Macintosh ‘Open File’ dialog, a previously saved *TurboMCS Controller* settings file may be opened and loaded into the attached TurboMCS hardware (if it is available). This is useful if one runs many different types of standard experiments– you can save a settings file for each and instantly recall it to the TurboMCS. The affected settings include:

- dwell time
- internal/external dwell advance
- internal/external trigger
- discriminator or SCA input
- discriminator level
- rising or falling edge discriminator
- upper and lower SCA levels
- number of preset passes
- number of channels per pass
- ramp style (begin/end or begin/mid/end)
- ramp level settings
- ROI start channel
- ROI length

Please note that the new settings will be loaded into the attached TurboMCS hardware, so the current

settings will be gone unless you have saved them. If the TurboMCS hardware is currently scanning, not all of the new settings may be loaded as not all parameters can be changed during a scan. You must then stop scanning and reload the settings.

2.2.7 Save Settings

Prompts the user with a standard Macintosh ‘File Save’ dialog to save the current TurboMCS settings (no data) into a disk file. See Open Settings (2.2.5) for a description of which settings are saved. Note that if you choose a file name that is the same as an existing file, you will be prompted if you want to replace the named file with the new one.

2.2.8 Page Setup

Standard Macintosh ‘Page Setup’ dialog used before printing a data window.

2.2.9 Print (⌘-P)

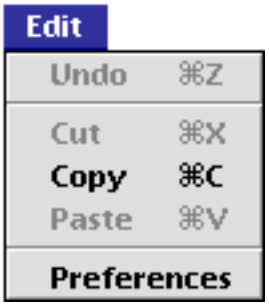
Prints the frontmost data window graph using a standard Macintosh ‘Print’ dialog.

2.2.10 Quit (⌘-Q)

Quits the *TurboMCS Controller* application. The application preferences are saved to disk (in the ‘`System Folder:Preferences`’ folder). You can then turn off the TurboMCS hardware if you wish. Note that the TurboMCS hardware can also continue to run quite happily after the controller application quits. If you run the *TurboMCS Controller* application later it will read the current settings without interrupting any ongoing scan. As long as the Turbo-MCS hardware is not turned off, the data will be retained and can be downloaded later.

Note also that it is not a good idea to turn off the Turbo-MCS hardware *before* quitting the application. The application continuously probes the hardware via the serial port, so if the hardware is suddenly turned off, the application will wait a long time for data to arrive on the serial port.

2.3 Edit Menu



2.3.1 Undo (⌘-U)

Supported only for cutting/pasting in the dialog data fields (eg. the pass control dialog)

2.3.2 Cut (⌘-X)

Supported only for dialog data fields (eg. the pass control dialog)

2.3.3 Copy (⌘-C)

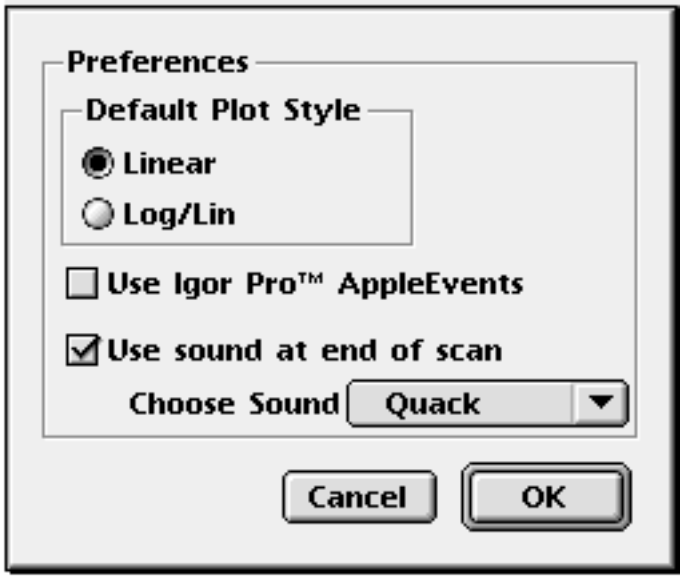
Copies the graph in the frontmost data window into the Macintosh 'Clipboard' (the graph is in a standard PICT format). The clipboard data can then be pasted into any other Macintosh application that can interpret PICT graphics (eg. most word processors, graphics editors). Can also be used in text fields in the dialog boxes.

2.3.4 Paste (⌘-V)

Supported only for dialog data fields (eg. the pass control dialog)

2.3.5 Preferences

Open the *TurboMCS Controller* 'Preferences' dialog



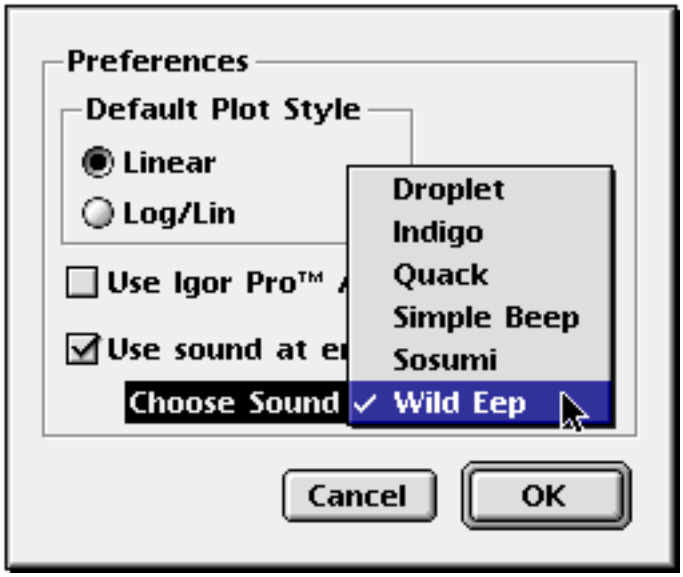
The default plot style affects what sort of graph is plotted when data is first downloaded from the Turbo-MCS hardware or read from a disk file. If you subsequently want to change the plot style for a graph you can select Log/Lin (⌘-L) in the 'Display' menu.

The 'Use Igor Pro™ AppleEvents' checkbox allows the *TurboMCS Controller* to exchange data with Igor Pro (product of WaveMetrics). If you have Igor Pro (version 2.x or 3.x) you may want to use it for data analysis and presentation purposes. The first time this button is checked (or if Igor Pro cannot be found), you will be prompted with a dialog to locate a copy of Igor Pro. The *TurboMCS Controller* remembers this location, so as long as you don't move Igor on your disk, the *TurboMCS Controller* can find it again without assistance.

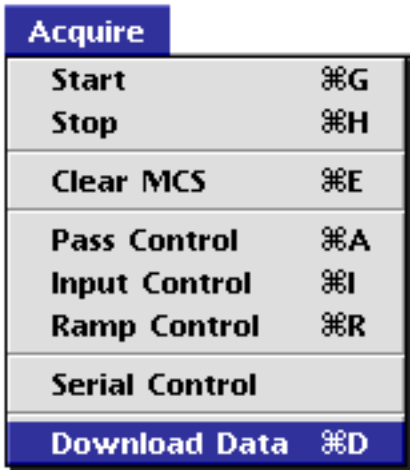
When this option is selected, the Igor Pro application will be launched into the background (ie. behind *TurboMCS Controller*) and a 'Program-To-Program Communications' (PPC) link established. The link will be kept active until one of the applications (Igor or TurboMCS) quits or the checkbox is deselected.

This option is stored in the application preferences. If it is selected, then the next time you run the TurboMCS application, it will automatically launch Igor Pro too (if it is not already running). To stop this behaviour, deselect this preferences option.

The 'Use sound at the end of a scan' option allows an alert sound to be played by the Macintosh speaker at the finish of a TurboMCS hardware scan. You can choose any of the installed 'Alert Sounds' in your Macintosh system file. Refer to your Macintosh system documentation for information on adding new alert sounds.


Preferences
Default Plot Style
Linear
Log/Lin
Use Igor Pro™
Use sound at
Choose Sound
Droplet
Indigo
Quack
Simple Beep
Sosumi
Wild Eep
Cancel
OK

2.4 Acquire Menu



2.4.1 Start (⌘-G)

Starts a new scan on the Turbo-MCS hardware using the currently loaded settings. Note that the hardware settings cannot be changed while a scan is in progress, though you can view the current settings in the relevant dialog boxes. A ‘Start’ can also be initiated on the floating toolbar by pressing the right-pointing arrow.

2.4.2 Stop (⌘-H)

Selecting ‘Stop’ once will stop a Turbo-MCS scan at the end of the current pass. Selecting ‘Stop’ twice will stop the Turbo-MCS immediately (ie. in mid-scan). Has no effect if the Turbo-MCS is not scanning. A ‘Stop’ can also be initiated on the floating toolbar by pressing the stop sign icon.

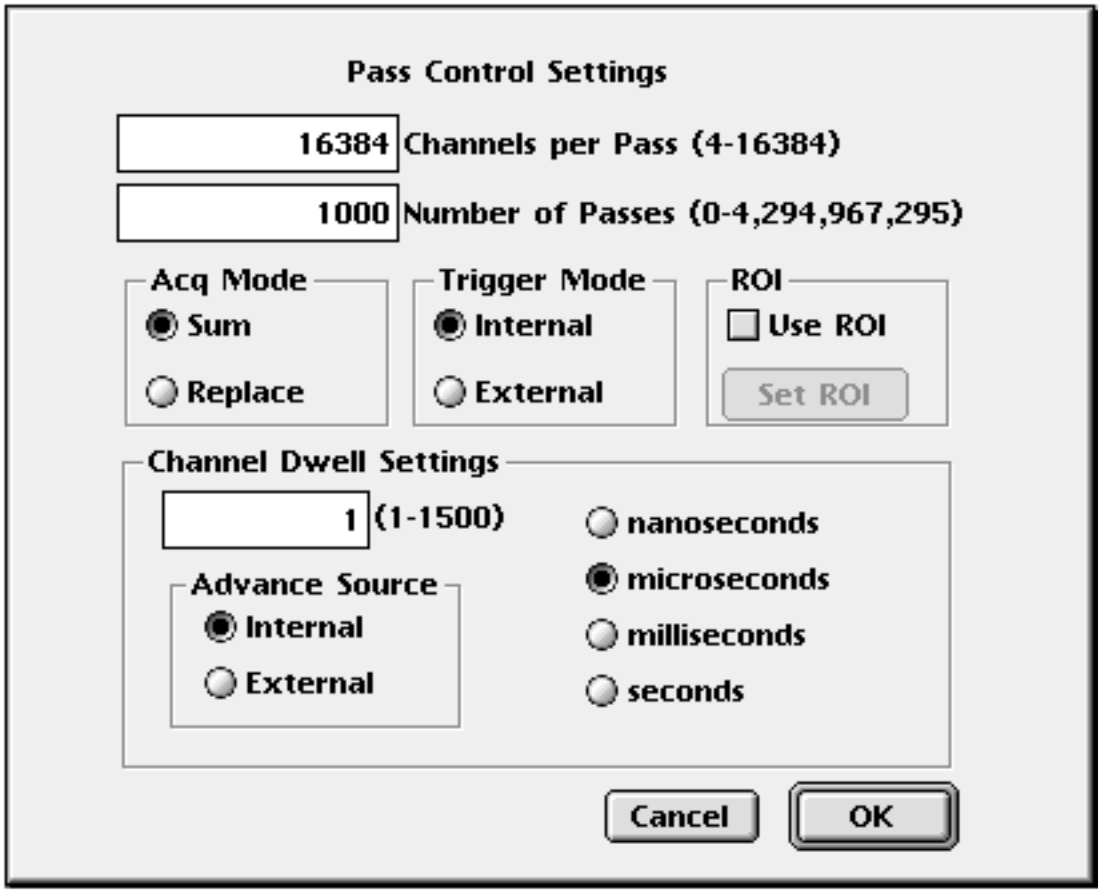
2.4.3 Clear MCS (⌘-E)

If the Turbo-MCS is not currently scanning, all of its data memory channels are set to zero. This is normally done before a new scan, after the previous data has been downloaded. Has no effect if the Turbo-MCS is currently scanning.

A ‘Clear’ can also be initiated on the floating toolbar by pressing the ‘eraser’ icon.

2.4.4 Pass Control (⌘-A)

Opens the Pass Control dialog (shown below).



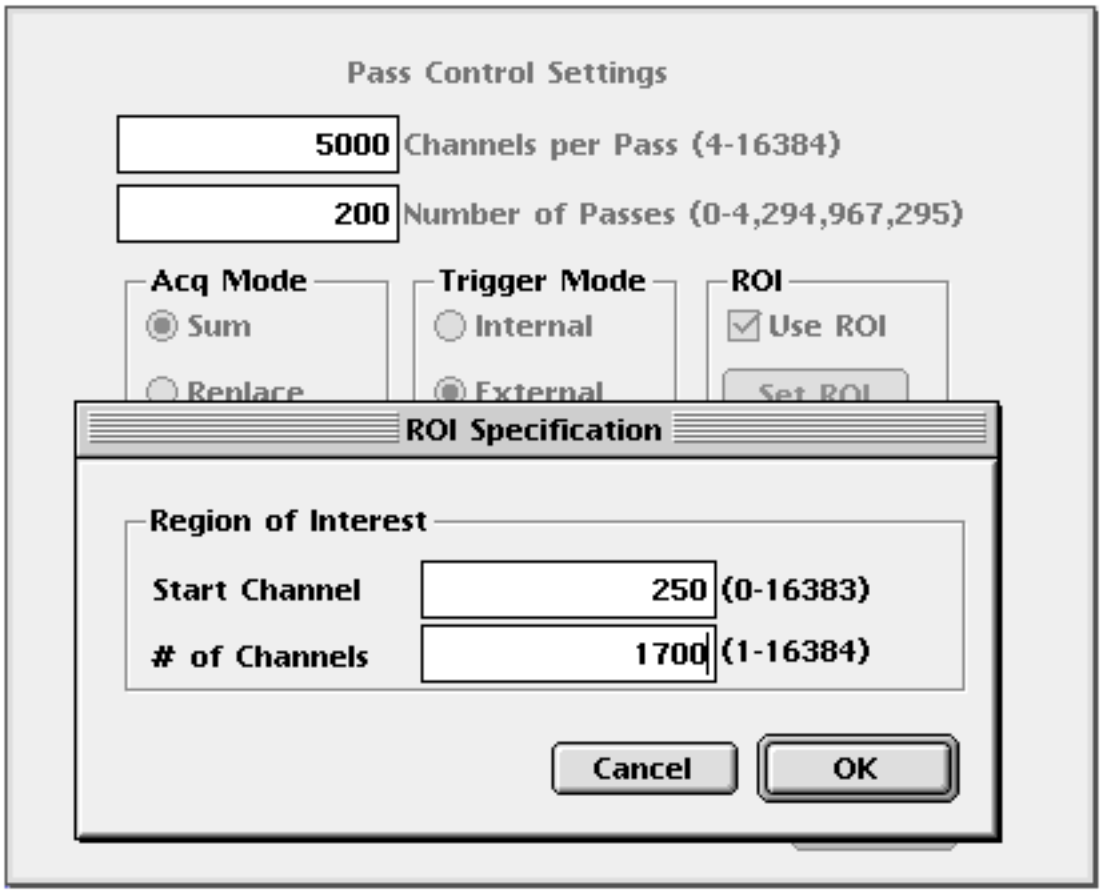
Allows settings of the TurboMCS hardware to be changed. The ‘Channels per Pass’ (ie. number of scaler bins) must be in the range of 4 to 16384 due to the way the TurboMCS hardware is configured. The ‘Number of Passes’ can be in the range 1 to 4,294,967,295 or can be essentially infinite (manually control of stopping) if the value is set to 0.

The MCS hardware can be set to continuously add scans in memory (‘Acq Mode: Sum’) or replace the contents of memory with each successive pass (‘Acq Mode: Replace’). The most usual setting is ‘Sum’ mode.

Pass start triggering can be set to be internal (‘Trigger: Internal’) or use an external trigger source connected to the ‘Start’ input on the TurboMCS hardware unit (see [1] for details).

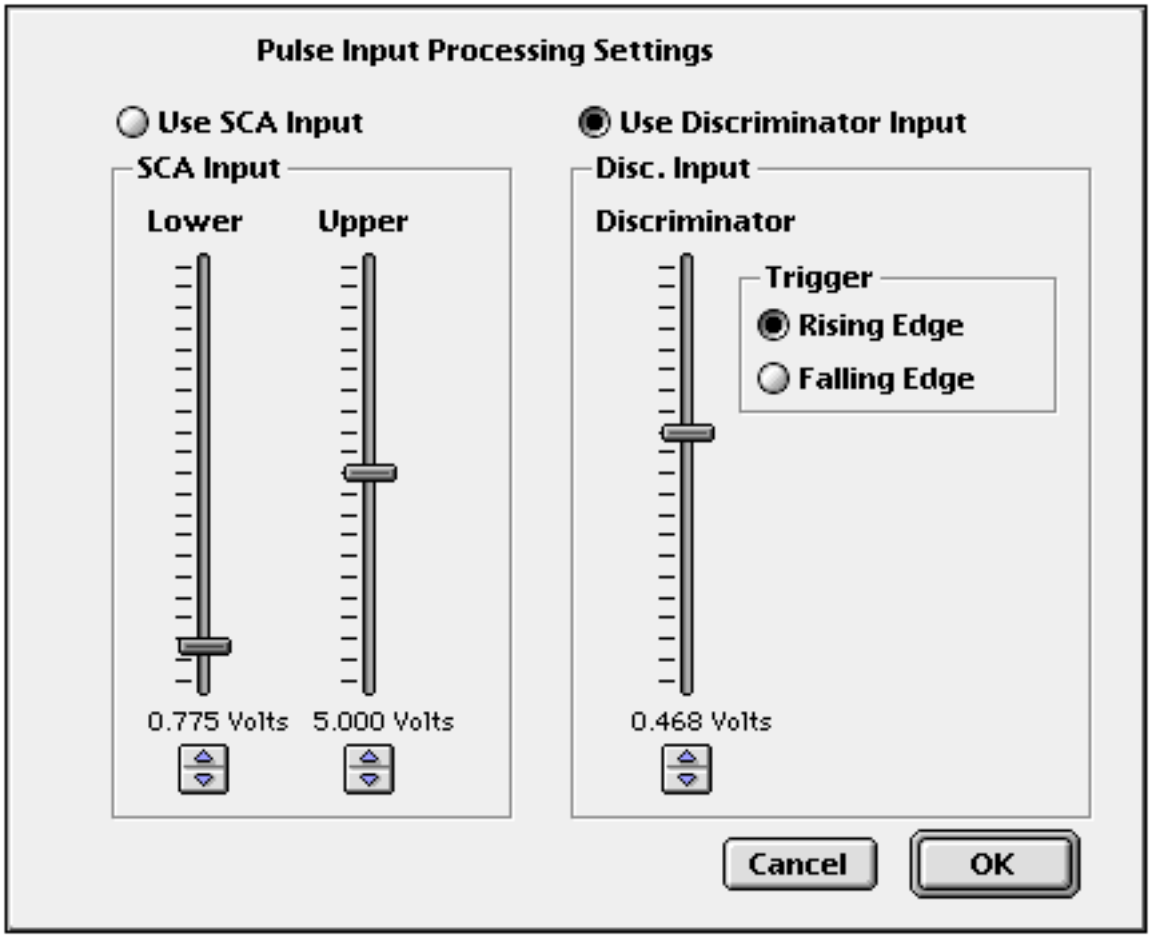
The MCS dwell time (bin length) can be specified and use either internal timing (‘Advance Source: Internal’) or an external bin timing (Advance Source: External). Note that not all dwell times within the range of the TurboMCS hardware (5ns to 65535s) are allowed. Improper dwell values are automatically rounded to the nearest legal setting using the ‘VERIFY_DWELL’ command described the the TurboMCS hardware manual [1].

A ‘region of interest’ (ROI) can be specified so that only a subset of the entire data set will be downloaded at the end of a scan. This may be useful for someone though I can’t think of why.



2.4.5 Input Control (⌘-I)

Opens the Input Control Dialog (shown below).



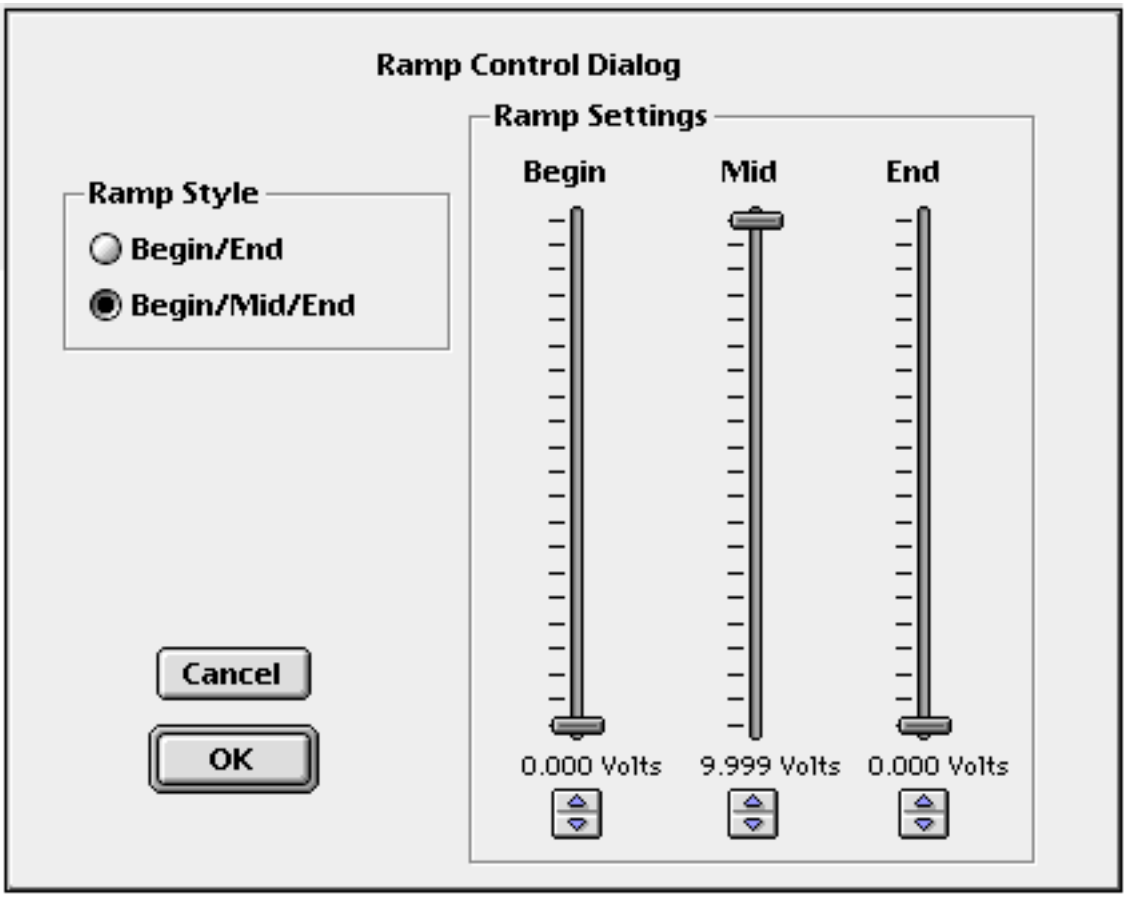
This allows the selection of either 'Single Channel Analyzer' (SCA) pulse input or 'Discriminator' pulse input (see the Turbo-MCS hardware manual [1] for more information). The SCA input is selected by choosing the left-hand radio button. SCA levels are selected using the slider controls are the up/down arrows at the bottom of the slider control. The current voltage settings are also displayed at the bottom of the slider. The arrows are useful for fine control of the voltage levels- pressing one once increments or decrements the setting by 1mV. If the command key (⌘) is held down, the setting will change by 10mV.

The discriminator input can be set to generate a count on either a rising or a falling voltage, at the level set by the discriminator slider. Again fine control is possible using the up/down arrows, with or without pressing the ⌘-key.

If the ‘OK’ button is selected, the Turbo-MCS hardware will use these new values[1]; if ‘Cancel’ is selected, the hardware settings are not changed.

2.4.6 Ramp Control (⌘-R)

Opens the Ramp Control Dialog (shown below).



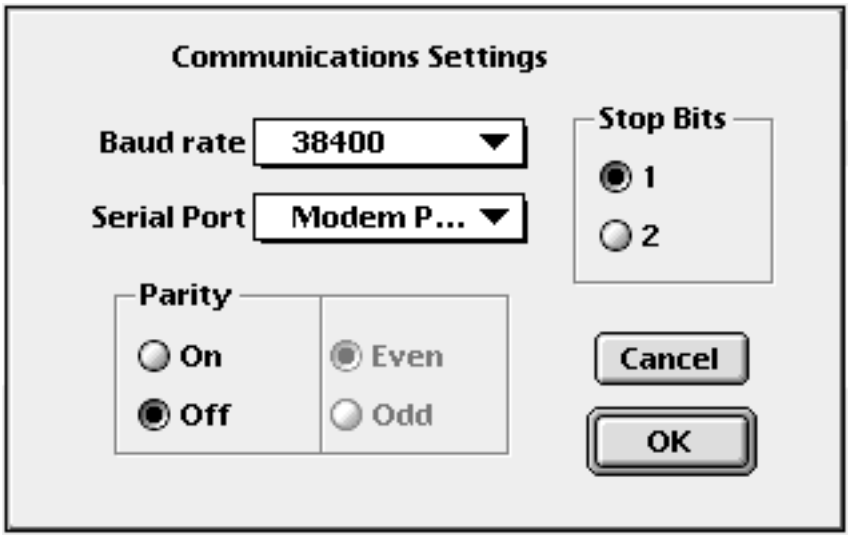
If the Turbo-MCS ramp hardware option is installed and is detected by the software, then it can be accessed by this dialog. The ‘Ramp Style’ radio buttons allow the selection of either the simple ramp (Begin/End) or the two stage ramp (Begin/Mid/End). See the Turbo-MCS hardware manual [1] for a description of the ramp hardware. If a Begin/End ramp is selected, then the ramp start voltage is set by the left-hand slider and the ramp end voltage is set by the right-hand slider. If a Begin/Mid/End ramp is selected then the ramp midpoint voltage is set by the centre slider. The sliders may be ‘fine controlled’ using the arrow controls at the bottom (incrementing in 1mV steps or 10mV steps if the ⌘ key is depressed).

[1] Note that not all voltage settings are allowed on the Turbo-MCS hardware (see Ref [1]). The Turbo-MCS hardware will use the closest legal value to the selected voltage setting. This new setting will be retained by the software and displayed the next time the dialog is opened.

If the ‘OK’ button is selected, the Turbo-MCS hardware will use these new values[1]; if ‘Cancel’ is selected, the hardware settings are not changed.

2.4.7 Serial Control

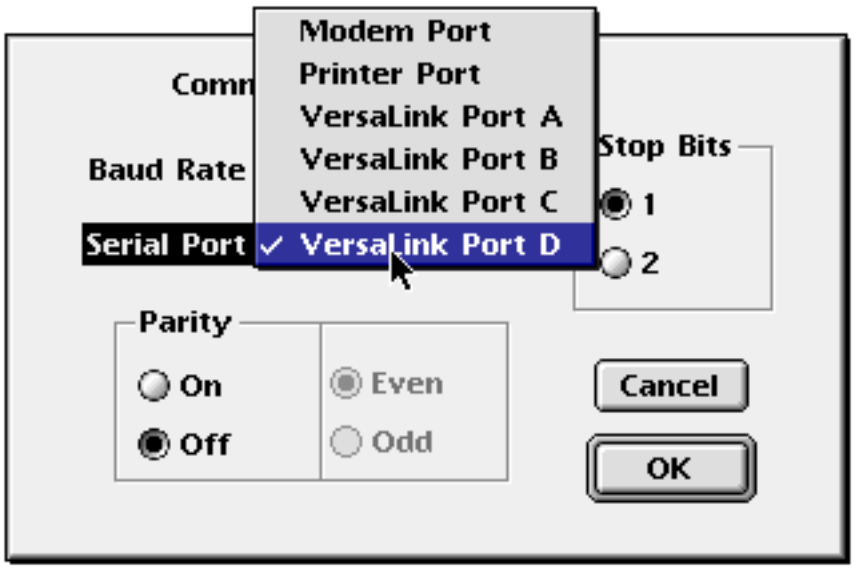
This dialog allows control of the serial communications settings with the Turbo-MCS hardware (shown below).



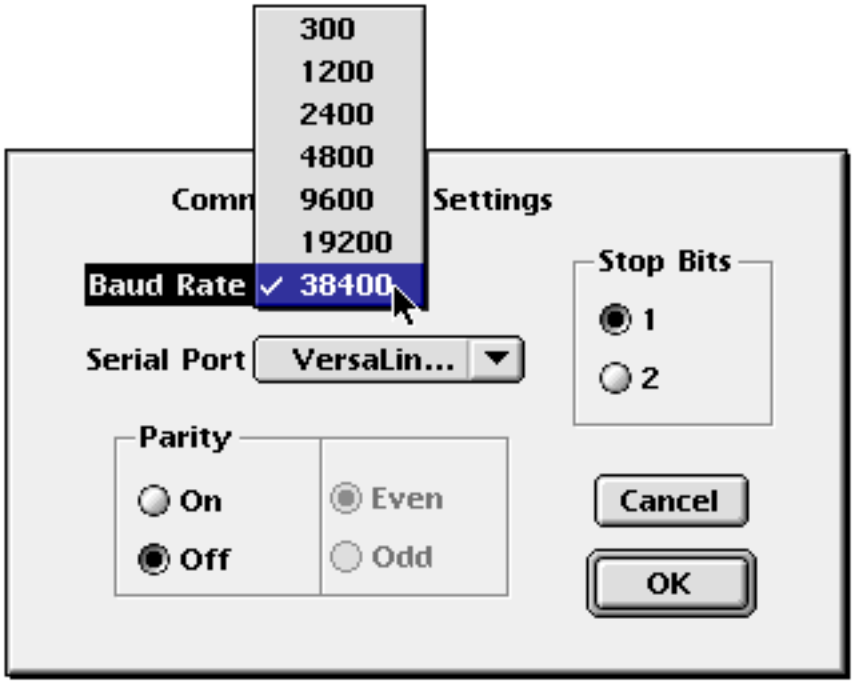
The communications baud rate is set by a pop up menu, and the port that the hardware is connected to should be selected in the serial port pop up menu. For the other settings it is normal to leave parity checking off and 1 stop bit. After making the selections and hitting the OK button, the *TurboMCS Controller* software searches for the TurboMCS hardware using the supplied settings. If the TurboMCS is found, you can proceed with other settings and communications with the unit. If the TurboMCS is not found, you may have to check either the software settings and/or the cable connection between the Macintosh and the TurboMCS.

On a standard Macintosh there are 2 serial ports- the printer port and the modem port. Generally one should choose to use the modem port as it has a higher internal priority for data. If there is a third party serial port card installed, these added ports should show up in the serial port pop up menu list[2]. If they do not, please refer to the documentation that accompanies your serial card for troubleshooting instructions.

[2] Any serial port that is registered under the Communications Toolbox should be detected by the TurboMCS software.



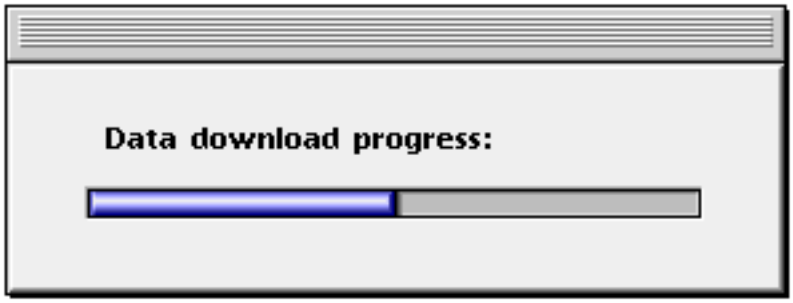
The default settings for the serial port on the TurboMCS hardware (factory settings) are 9600 baud, no parity and 1 stop bit. You should initially set up the *TurboMCS Controller* to work with these settings until you are sure about how the connections work. If you are routinely downloading large data files (>500 channels per scan) then you will probably want to run at a higher baud rate setting. Changing the relevant DIP switches on the TurboMCs hardware is quite difficult– refer to the hardware manual [1] for instructions.



The limiting factors on the serial port speed can come from the serial cable length and shielding

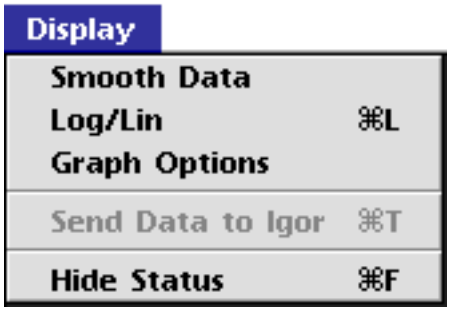
(choose as short a cable as possible, and a good quality one) and the Macintosh processor speed (I use a Macintosh IIfx, which communicates without problems at 38,400 baud).

2.4.8 Download Data (⌘-D)



Retrieves the current data set from the TurboMCS hardware. If a ROI is selected in the Pass Control Dialog (see section 2.4.4), then only the ROI channels will be retrieved. When the data has been downloaded, it is displayed in a new graph window. The data is also still present in the TurboMCS hardware memory until you send a 'clear' command to the unit (see section 2.4.3). It may be convenient to download data periodically for inspection until you have accumulated enough of the signal you are looking for before saving the data to disk. Note also that it is possible to download data *during* an ongoing scan, although the number of passes finished may be different in different channels.

2.5 Display Menu



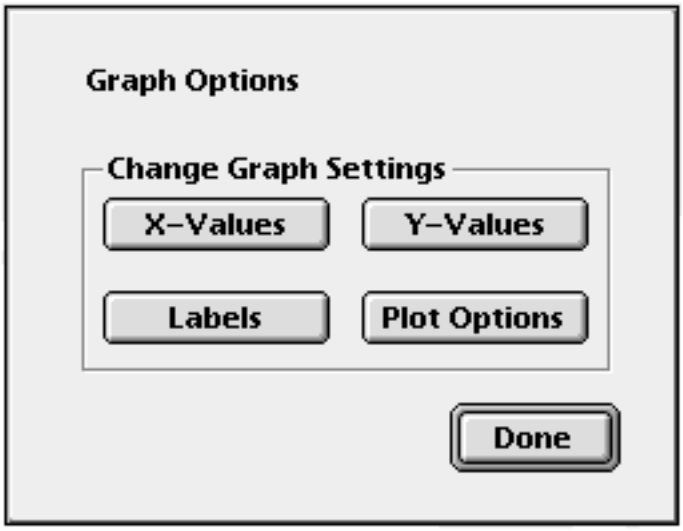
2.5.1 Smooth Data

Applies a 5 point smoothing to the data in the frontmost graph window. The smoothing algorithm applied to data point `Y[i]` is `Y[i]=(Y[i-2]+4*Y[i-1]+6*Y[i]+4*Y[i+1]+Y[i+2])/16`, and the result is rounded to the nearest integer value. Note that this process is not reversible– be sure to save the data beforehand if you are not sure about the result.

2.5.2 Log/Lin (⌘-L)

This switches the display in the frontmost graph window between a linear and a logarithmic Y-axis.

2.5.3 Graph Options



This dialog allows various changes to be made to how the frontmost graph window is displayed. Any changes that are made also apply to any subsequent graph window that is opened (either from disk or by downloading). The same changes can be made to the frontmost graph by double-clicking in the relevant area of the graph.

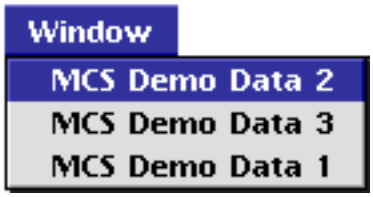
2.5.4 Send Data to Igor (⌘-T)

If a PPC connection to *Igor Pro* has been established, then this selection will send the data in the frontmost window directly to Igor Pro and will be displayed in a graph.

2.5.5 Hide/Show Status (⌘-F)

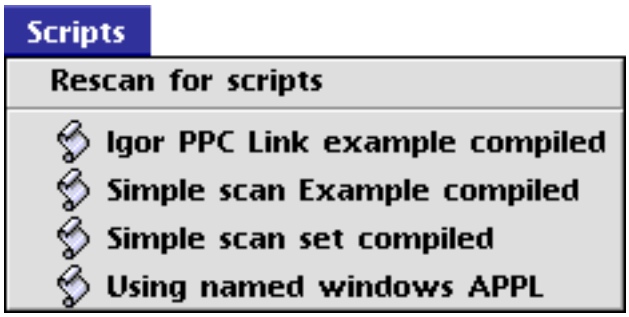
This selection toggles the display of the floating window that shows some information about the TurboMCS multichannel scaler.

2.6 Window



The names of data windows currently open are displayed in a list and any of the windows may be brought to the front by selecting its name in this menu.

2.7 Scripts



When the *TurboMCS Controller* application is run, it scans the folder named '`TurboMCS Scripts`', located in the same folder as the application (note that if no such folder exists, it is created automatically). If a valid compiled AppleScript file is found, the name of that script is recorded and displayed in this menu. When a script in this menu is selected, the script is run, allowing multiple commands of the *TurboMCS Controller* or any other scriptable application to be made. This allows for automated execution of data acquisition runs and synchronization with other application programs (eg. controlling other hardware). If you decide to add, remove or change scripts to the folder during while the *TurboMCS Controller* is running, you can rescan the scripts folder to update the script list.

In order to have scripts loaded into this menu they must not only be placed inside the '`TurboMCS Scripts`' folder, but the scripts must be saved as 'compiled' scripts or as 'application' scripts. If you are using the Apple 'Script Editor' application that comes with AppleScript, saving a script as a compiled or as an application script is available as an option in the 'Save' or 'Save As…' dialog boxes.

3. TurboMCS Floating Status Window

3.1 MCS Control

These pushbuttons allow starting, stopping (press once for end of pass, twice for immediate) and clearing of the TurboMCS hardware. They perform the same function as the menu selections in the 'Acquire' menu.

3.2 MCS Status

Information about the current MCS hardware status is displayed and updated continuously during normal operation.

'Pass': the current and target number of passes

'Channel': the current acquisition bin (only valid if the dwell time is ≥200ns)

'Status': whether the TurboMCS hardware is currently scanning or not

'FreeMem': the number of kilobytes that are free for the *TurboMCS Controller* software to utilize (mainly for data display windows)

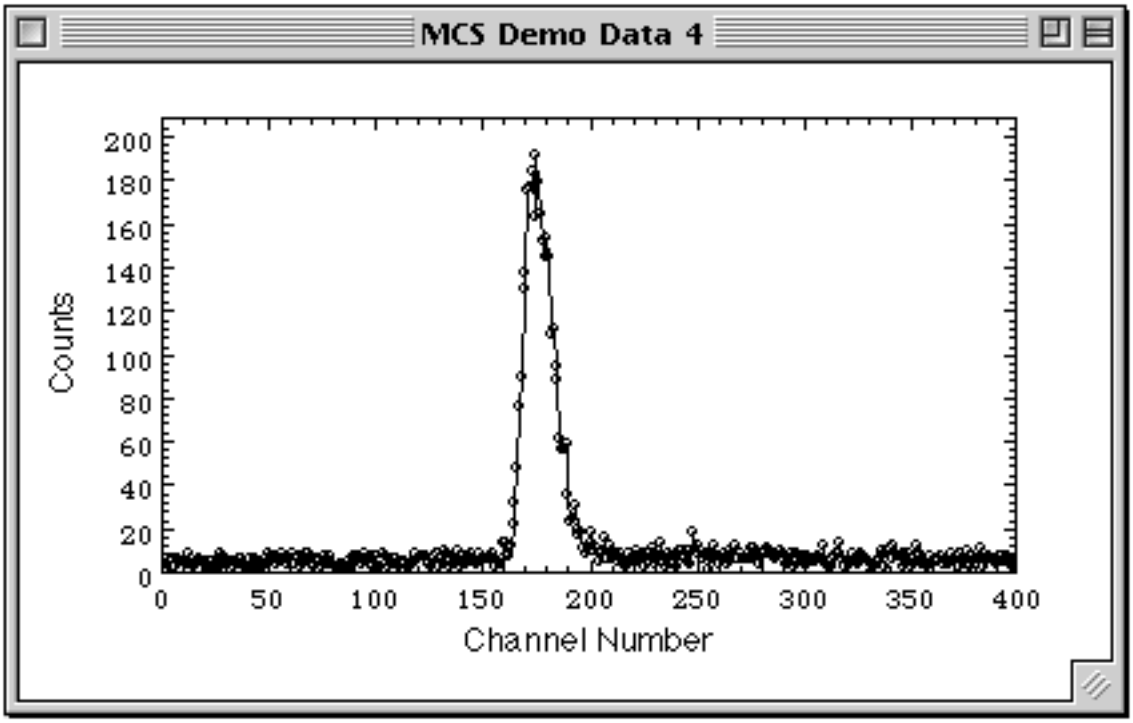
'∑': the total integrated number of counts in the current scan

'Max Counts': the highest count and which bin this high count is located in

These status indicators are updated periodically (roughly once per second in normal use) by serial port communications with the TurboMCS hardware unit. There is one caveat however– if a very short dwell time is used (<100ns) in combination with a long pass length (>500 channels or so, the exact number depends on the dwell time)– then serial communication with the hardware unit are very slow during a scan. This inhibits the ability of the *TurboMCS Controller* application to determine the various status values so they may not always be available under these conditions.

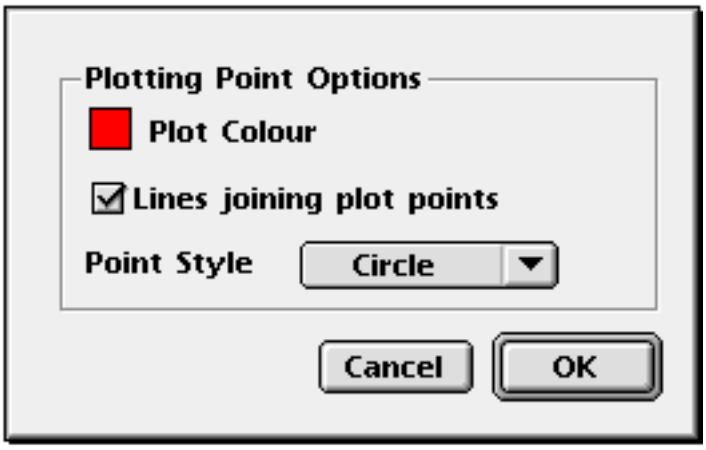
4. Graph Windows

Any TurboMCS data that has been downloaded from the hardware unit or read from disk is displayed in a graph window (see example below). The format of this display can be controlled in various ways via menu selections (see section 2.5) or by double clicking on the graph itself.



4.1 Graph Colour, Point Style, Graph Style

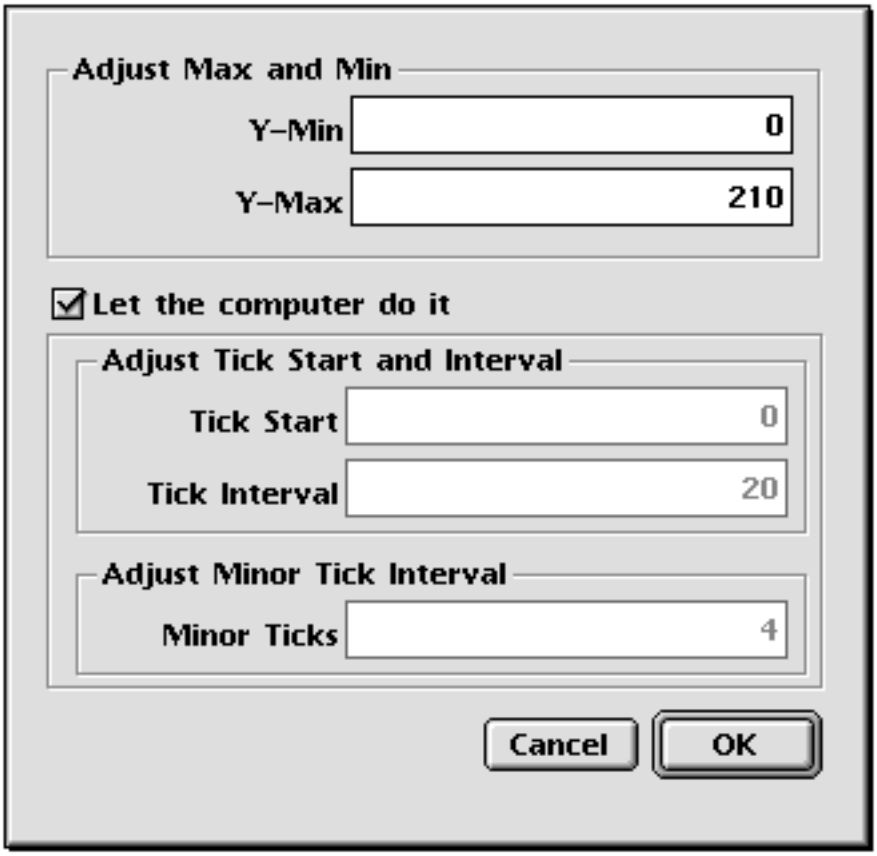
If the mouse is double-clicked in the interior region of the graph, a dialog box controlling the plot colour, display point type and graph style is presented. The type of point plotted for the data points is displayed in a pop-up dialog.



A mouse click in the ‘Plot Colour’ colour box brings up the standard Macintosh Colour Picker dialog (MacOS System version dependent). You can choose any available colour for the graph display.

4.2 X– and Y–Axis Scales

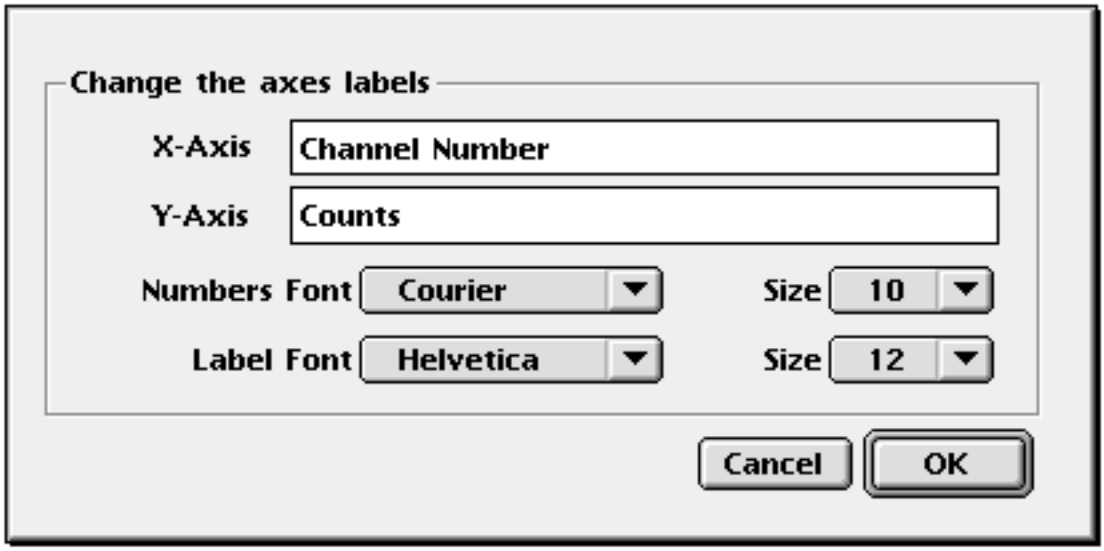
A double-click in the region of the x-axis or the y-axis of a graph will bring up a dialog that controls the display of the range and tick-mark intervals of a graph.



The range of x– or y–axis display can be controlled. The major and minor tick intervals and starting values can usually be left for the TurboMCS Controller application to determine, but the automatic values can be overridden by deselecting the ‘Let the computer do it’ check box and entering the desired values.

4.3 Labels, Fonts and Font Sizes

By double-clicking in the region of either the x– or y–axis text labels, the label and font selection dialog is opened. You may choose any label that is less than 255 characters long. The font and font size choices are presented in popup dialogs



References

[1] Turbo-MCS User's Manual, EG&G Ortec Part No. 762840 (included with Turbo-MCS hardware).

Appendix 1: TurboMCS Controller Data File Format

The *TurboMCS Controller* application saves its data files in a binary format that is described below. All of the data is written in the 'data fork' of the Macintosh file. The data files are compatible with both PowerPC and 680x0 versions of the application. The data alignments are the natural ones for 680x0 processors (hence the presence of unused bytes in the MCSSettings record).

The sizes of the different data types are:

```
   boolean: 1 byte
   integer: 2 bytes
   long: 4 bytes
   double: 8 bytes (IEEE 8-byte real number)
```

File Start

```
version: longint  { always 1 so far }
nPoints: longint  { number of data points }
MCSSettings: (76 bytes)
   MCSSettings= record
      DwellTime: double; { time in seconds }
      ExternalDwell: boolean; { true if external dwell signal used }
      SummationMode: boolean; { true if using sum mode }
      ExternalTrigger: boolean; { true if an external trigger is used }
      DiscInput: boolean;  { true if using discriminator, false if using SCA }
      DiscLevel: double; { discriminator setting in volts }
      DiscLeading: boolean; {true if disc uses rising edge, false if falling }
      <unused byte>
      SCALower,  { lower SCA setting in volts }
      SCAUpper: double; { upper SCA setting in volts }
      PassLength: integer; { number of dwells per scan }
      PassCount: longint; { number of preset scans, 0 if free running
      BeginMidEndRamp: boolean; { true if using begin/mid/end ramp style }
      <unused byte>
      RampStart, { ramp start setting in volts }
      RampMid,  { ramp middle setting in volts }
      RampEnd: double; { ramp end setting }
      WindowStart, { first channel in ROI }
      WindowLength: integer; { length of ROI }
      EnableAlarm: boolean; { not used }
      <unused byte>
   end;
data (2*nPoints*8 bytes)
End of file.
```

Each data point is written as a pair (x,y) or doubles (real numbers) using:

```
   RealPoint = record
      x, y: double;
   end;
```

Appendix 2: Scripting TurboMCS Controller

The *TurboMCS Controller* application is scriptable using AppleScript™. This allows the TurboMCS to be run remotely or for extended data acquisitions without user intervention. For a full discussion of the possibilities using AppleScript, the interested reader is referred elsewhere [1].

Many of the settings that can be controlled via the graphical user interface (GUI) can also be controlled via AppleScript. For our purposes there are two types of keywords– *properties* and *commands* – to consider.

Properties– variables that can be read or set, although some are read only

```
dwelltime real -- the MCS dwelltime in seconds
summode boolean -- summation mode (true) or substitution mode (false)
externaltrigger boolean -- start on external or internal trigger?
discriminatorinput boolean -- use discriminator (true) or single
    channel analyzer (false) for pulse input?
discriminatorlevel real -- discriminator setting in Volts
discriminatorleading boolean -- discriminator trigger on leading
    (true) or trailing (false) edge?
scalower real -- lower SCA setting (Volts)
scaupper real -- upper SCA setting (Volts)
passlength small integer -- number of channels in each pass
passcounts integer -- number of passes to scan (0=infinite)
beginmidendramp boolean -- ramp style- begin/mid/end (true) or
    begin/end (false)?
rampstart real -- ramp starting voltage (Volts)
rampmid real -- ramp middle voltage (Volts)
rampend real -- ramp end voltage (Volts)
usewindow boolean -- if true download defined ROI window only
windowstart small integer -- first channel of ROI to download
windowlength small integer -- number of channels after windowstart to
    download in the ROI
IgorPPCLink boolean -- Get/set state of the TurboMCS->Igor Pro PPC Link
datacounter integer [read only] -- count of data files since startup
isscanning boolean [read only] -- are we scanning now?
currentpass integer [read only] -- what is the current pass?
currentchannel small integer [read only] -- what is the current
    channel?
mcsonline boolean [read only] -- Is there an MCS connected?
mcsversion string [read only] -- TurboMCS version string
maxcounts integer [read only] -- peak channel counts
maxcountschannel small integer [read only] -- location of the peak
    channel counts
integralcounts integer [read only] -- sum of channel counts
FreeMemory integer [read only] -- Number of bytes of memory available
```

Commands– tell the application to do something

Required Suite

open: Open the specified file(s)
    open *alias* -- list of files to open, can be data or settings files

print: Print the specified file(s)
    print *alias* -- list of files to print (data files only)

quit: Quit application
    quit

run: Starts the application
    run

TurboMCS Suite: The core events supported by TurboMCS Controller.

get: Get the data for a property (see above)
    get *reference* -- the object whose data is to be returned
    Result: anything -- The data from the object

set: Set a property's data (see above)
    set *reference* -- the object to change
        to *anything* -- the new value

StartMCS: Start the MCS scanning
    StartMCS

StopMCS: Stop the MCS scanning at end of current pass
    StopMCS

HaltMCS: Stop the MCS immediately (do not wait for current pass to finish)
    HaltMCS

ClearMCS: Clear the MCS memory (all channels to zero)
    ClearMCS

DownloadMCSData: Transfer data from the MCS to the Macintosh
    DownloadMCSData

SendDataToIgor: Send current data in front window to Igor Pro™ if PPC link is established beforehand
    SendDataToIgor

SmoothData: Apply smoothing to the current data in frontmost window
    SmoothData

save: save the data in the targetted window
    save *reference* -- the window to save
        in *alias* -- the file to save it in

```
close: close the targetted window
      close  reference  -- the window to close
```

Example Scripts

The best way to get the hang of scripting is to try things out. Once you TurboMCS in connected to your computer and works via the GUI you can try running it via scripts. AppleScript can be a deceptively tricky programming language because it appears to be English-like. Like all programming, expect to spend some time debugging your scripts.

Example 1.

This example shows how to set a few properties and then set the TurboMCS scanning. One cute aspect of this example is the ability to scan only long enough until a preset target peak count is reached or the preset number of passes are finished.

```
tell application "TurboMCS Controller"
    -- set up our scan parameters
    set passcounts to 50000
    set passlength to 1024
    set dwelltime to 5.0E-6

    ClearMCS -- zero the memory
    StartMCS -- start the scan
    repeat while isscanning and (maxcounts < 5000)
        -- loop until we reach the preset value or our target peak count is reached
        -- note that you can't do this via the GUI except by direct user intervention
    end repeat
    -- now stop scanning on the current pass and get the data
    StopMCS
    DownloadMCSData -- data put in frontmost window, ie. window 1
    save window 1 in "test data 1" -- !! must be a unique file name/path or data won't be saved
    close window 1
end tell
```

When automating scans by scripting, it is important to save the data to disk. This can be a bit tricky since one must make sure to use unique filenames. The *TurboMCS Controller* will not overwrite preexisting data files from a script, so new data with the same filename may be lost.

There are a few simple strategies to reduce the risks. First, use *paths* (*absolute or relative*) in the file names. Second, make sure that you are generating a unique file name in the particular path (folder) where you are saving the data.

An *absolute path* is one that specifies the file hierarchy from the hard drive down to the desired folder or file

eg. `"Internal:System Folder:System"`

a *relative path* specifies folders from a given (default) starting point

Example 2.

```
tell application "TurboMCS Controller"

    set passlength to 100
    set passcounts to 1000
    set dwelltime to 1.0E-6

    copy 4 to maxCount -- number of passes in the loop

    repeat with counter from 1 to maxCount
        set dwelltime to (counter * dwelltime) -- silly example- increase dwelltime in each scan

        ClearMCS -- remember to zero the MCS memory
        StartMCS
        repeat while isscanning
            -- wait until done
        end repeat
        DownloadMCSData -- put the data into the frontmost window, ie. window 1

        -- create a __unique__ filename for each file
        -- here use the current time in seconds from midnight
        -- note should include an explicit path too
        set currentTime to (time of (current date))
        set newFilename to "MCSData" & currentTime

        save window 1 in newFilename
        close window 1 -- close each window so we know that we won't fill memory
    end repeat
end tell
```

Note how in the previous example that the data to be saved was referred to by the window it occupies after being downloaded. In this case we referred to ‘window 1’, which is the frontmost window. This numbering convention (fron to back) is typical of AppleScript. A second way to refer to a window containing data is by its name. This is also convenient from the TurboMCS Controller since the window naming scheme is fairly simple. *Note the slight wrinkle that after saving, the name of the window changes to that of the file.*

Example 3.

```
tell application "TurboMCS Controller"
        --"MCS Data-" is the default window name
        copy "MCS Data-" to wNameDefault
        -- chose a file name
        copy "MCSDataFile" to fNameDefault

        ClearMCS -- remember to zero the MCS memory
        StartMCS
```

```
        repeat while isscanning
                -- wait until done
        end repeat

        DownloadMCSData -- get some data and the counter is incremented

        -- use the counter to determine the window name
        set wNameString to (wNameDefault & datacounter)

        -- use the counter to create a file name
        -- note- the counter starts at 1 each time TurboMCS is run, so the name may not be unique
        set fNameString to (fNameDefault & datacounter & ".data")

        -- save the data in the named window to the named file, then close the window
        -- note: should use some explicit file path
        save window wNameString in fNameString
        close window fNameString  -- note the changed window name after saving
end tell
```

Possible Scripting Pitfalls

It pays to be careful designing scripts as it is possible to hang up your data acquisition. Firstly, be aware that if your commands generate a *TurboMCS Controller* error (bringing up an error message dialog box in the GUI), the application will wait in the dialog until someone presses a button. Obviously at this point an unattended application could wait a very long time while doing nothing.

There is another pitfall related to a TurboMCS hardware problem– namely the TurboMCS's difficulty in handling serial communications when both i) a very short dwell time is used (<100ns per dwell) and ii) long pass length (>5000 channels roughly, the exact value depends on the dwell time). In this circumstance, serial port interaction with the TurboMCS hardware is unreliable during the scan[3], so some or all of the properties that one may want to read (eg. isscanning, integralcounts) may return misleading values. If you must run the TurboMCS from scripts under these conditions, you may well need to employ other strategies in your scripts (eg. waiting for preset periods of time for a scan to finish).

References

[1] The Tao of AppleScript (2nd Edition), D. Schneider, Hayden Books (1994)

[3]This is a 'feature' of the TurboMCS hardware unit and how it handles serial communications