Chell

Chell Instruments Ltd
Folgate House
Folgate Road
North Walsham
Norfolk NR28 0AJ
ENGLAND

Tel: 01692 500555
Fax: 01692 500088

microDAQX

User Guide

SOFTWARE MANUAL

e-mail:- info@chell.co.uk

Visit the Chell website at:
http://www.chell.co.uk

900170-1.1

Software Licence Agreement

This licence agreement is provided to you before you install the software. On installing the software, you are demonstrating your agreement to the Terms & Conditions of Chell Instruments Ltd. If you do not agree to these Terms & Conditions then you may not use the software.

Licence: Chell Instruments Ltd. hereby grants to the customer who has purchased a microDAQ with this software a limited, non-exclusive licence to use the software on a single computer. The software may also be installed on a second computer where it is necessary to support a Chell product from a second location, such as a service engineers Notebook PC. The customer may transfer the software to another computer provided that the software may not be used on more than one computer at any given time, except in the aforementioned situations.

This licence is granted solely for the purpose of interfacing, display and configuration of a Chell product. Use with other equipment is prohibited unless an additional fee is paid and a licence variation issued.

Subject to the foregoing limitation, the customer is entitled to copy the software into backup in support of the use of the software on a single computer. The customer may likewise merge the software or incorporate the same into another programme provided that such programme will, for so long as the software is included therein, be subject to all of the Terms & Conditions of this licence. Any copy of this software permitted by the preceding provision shall include a copyright notice of Chell Instruments Ltd.

Terms of Licence: The licence will continue until the customer physically destroys all copies of the software and merged portions thereof and returns the original programme disk to Chell Instruments Ltd.

Copyright: The software is Copyrighted and all rights therein are reserved by Chell Instruments Ltd.

Acknowledgements: The customer acknowledges that he/she has read this agreement, understands it and agrees to be bound by its Terms & conditions, which agreement shall be acknowledged in writing upon the request of Chell Instruments Ltd.

General Provisions: The customer shall have no rights to rent, lease, sub-licence time-share, lend or transfer the software or software copies, or the customer’s rights under this licence. In the event of a breach of this agreement by the customer, the customer shall be liable for all damages to Chell Instruments Ltd. and this agreement shall be terminated.

Chell Document No. : 900170 Issue 1.1
ECO : 1273 Date: 1st Decemeber 2014

Chell’s policy of continuously updating and improving products means that this manual may contain minor differences in specification and design from the actual software supplied.

CONTENTS

1 Quick Start (Ethernet)

This section outlines the steps necessary to start collecting data with a single microDAQ using Ethernet. This section does not deal with all the microDAQ options, which are described further on in this manual or in the microDAQ operating manual.

[1] Connecting up.

The microDAQ is supplied configured from the factory. It will have been given an i/p address that it written on the back of the microDAQ. Connect the microDAQ to the PC by an Ethernet lead.

NOTE: If the microDAQ is connected directly to a PC then a crossed RJ45 lead will need to be used. If it is connected to a hub/switch then a normal RJ45 lead should be used.

[2] Installing the software.

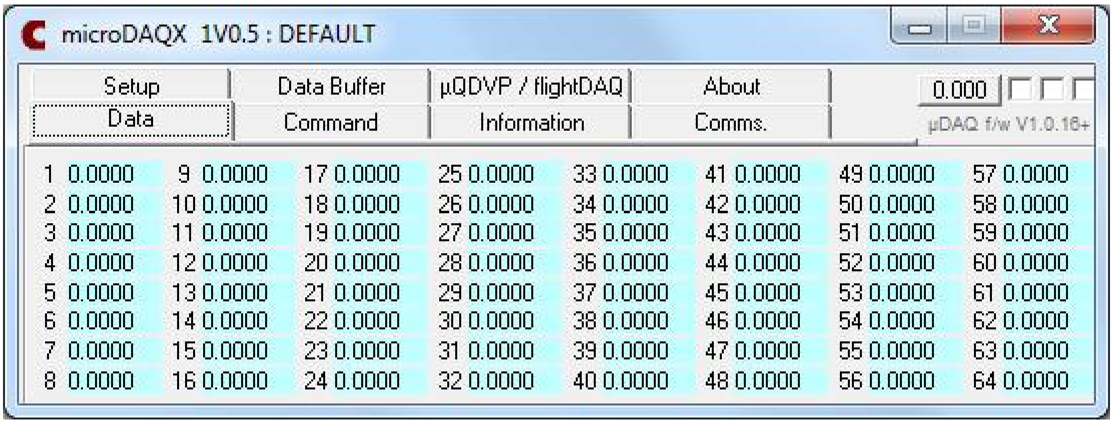
The software is installed by running the setup.exe in the microDAQX directory on the CD supplied. Follow the instructions until the installation is complete.

[3] Powering up the microDAQ.

Turn on the power to the microDAQ. The blue LED will light for between 10 and 40 seconds depending on the calibration type and number of scanner channels. The blue LED will start to flash.

[4] Running the software.

Run microDAQX.exe from the menu bar and the following will appear:

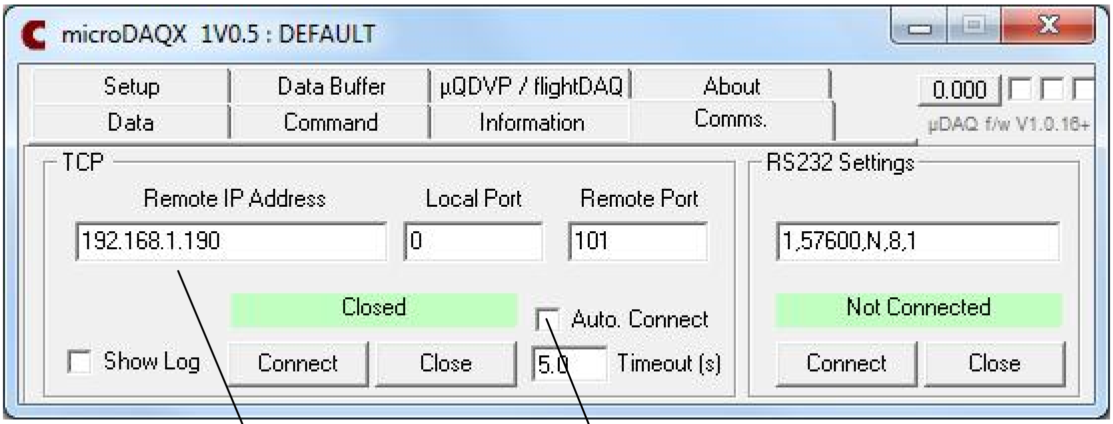


1	0.0000	9	0.0000	17	0.0000	25	0.0000	33	0.0000	41	0.0000	49	0.0000	57	0.0000
2	0.0000	10	0.0000	18	0.0000	26	0.0000	34	0.0000	42	0.0000	50	0.0000	58	0.0000
3	0.0000	11	0.0000	19	0.0000	27	0.0000	35	0.0000	43	0.0000	51	0.0000	59	0.0000
4	0.0000	12	0.0000	20	0.0000	28	0.0000	36	0.0000	44	0.0000	52	0.0000	60	0.0000
5	0.0000	13	0.0000	21	0.0000	29	0.0000	37	0.0000	45	0.0000	53	0.0000	61	0.0000
6	0.0000	14	0.0000	22	0.0000	30	0.0000	38	0.0000	46	0.0000	54	0.0000	62	0.0000
7	0.0000	15	0.0000	23	0.0000	31	0.0000	39	0.0000	47	0.0000	55	0.0000	63	0.0000
8	0.0000	16	0.0000	24	0.0000	32	0.0000	40	0.0000	48	0.0000	56	0.0000	64	0.0000

If the i/p addresses are configured correctly then data will appear. If not the i/p address will need to be entered.

[5] Entering the i/p address

The i/p address is entered in the comms tab as follows:



TCP

Remote IP Address: 192.168.1.190

Local Port: 0

Remote Port: 101

Closed

□ Auto. Connect

□ Show Log

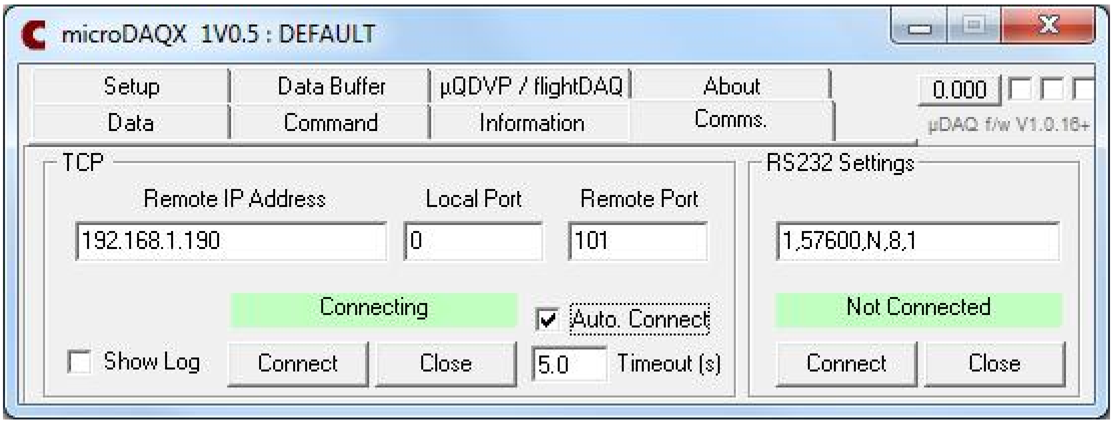
Timeout (s): 5.0

RS232 Settings: 1,57600,N,8,1

Not Connected

Enter the correct i/p address in the box.

Then select the Auto connect function.



TCP

Remote IP Address: 192.168.1.190

Local Port: 0

Remote Port: 101

Connecting

☑ Auto. Connect

□ Show Log

Timeout (s): 5.0

RS232 Settings: 1,57600,N,8,1

Not Connected

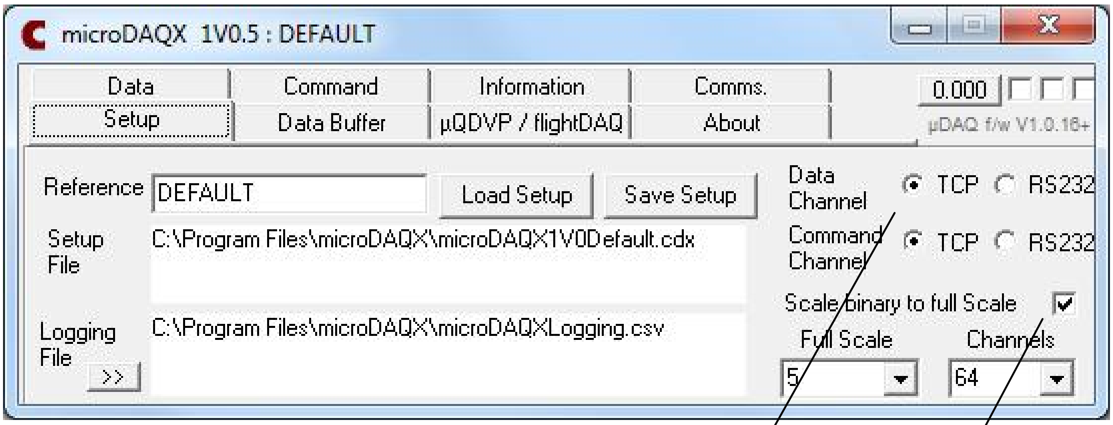
This window shows that the microDAQ is connecting. The lights on the upper right hand corner will light periodically to show that data is being received.

NOTE: If the system does not connect, look at the following:

Is the i/p address correct?
Is the link LED on the hub/switch? – If not, the wrong cable type may have been used.
Is the PC i/p address in the same domain as the microDAQ? In this case, the PC that is being used to connect to the microDAQ must have an i/p address starting with 192.168. The machine i/p must also be different to the microDAQ.
Has the microDAQSetup software been run. After the microDAQSetup software has been used, the microDAQ <u>must</u> be reset by a power cycle (or a soft reset from within the Setup program) before the microDAQX software will connect to it.

[6] Saving the settings.

It is important that the i/p address is saved. Do this by selecting the Setup tab as follows:



Reference DEFAULT

Setup File C:\Program Files\microDAQX\microDAQX1V0Default.cdx

Logging File C:\Program Files\microDAQX\microDAQXLogging.csv

Data Channel ● TCP ○ RS232

Command Channel ● TCP ○ RS232

Scale binary to full Scale ☑

Full Scale 5

Channels 64

First ensure that the data and command channel are set to TCP

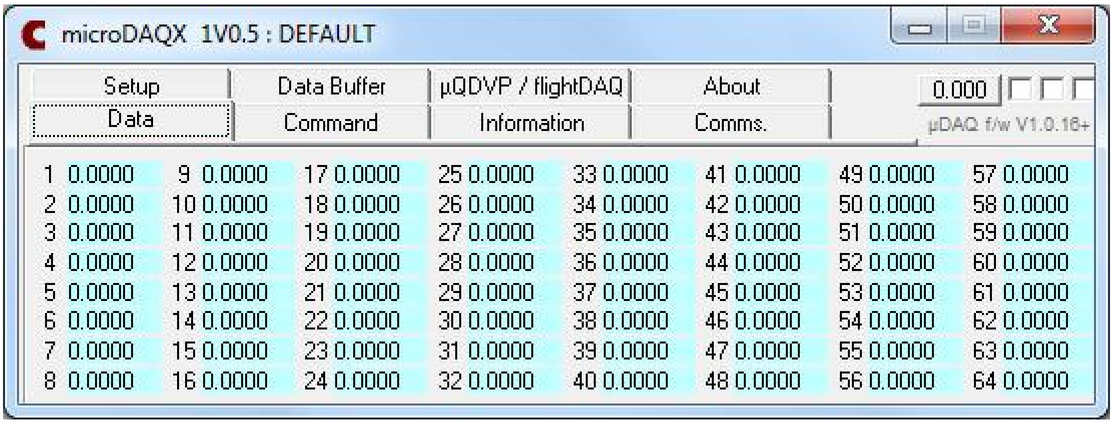
Then ensure that the scale binary to full scale is selected. This will scale the 16 bit number transmitted from the microDAQ into the appropriate units. The number of channels and full scale will also need to be correct.

NOTE: The full scale setting can be used to change the units. In this case, the scanner has a 5 psi full scale. If you wanted the units to be in, say, Pascals then add the appropriate full scale in Pascals into this text box (5 x 6894.757).

Then save the setting to the default file by overwriting the existing setup file.

[7] Viewing the data

The data can now be viewed with the data tab as follows:

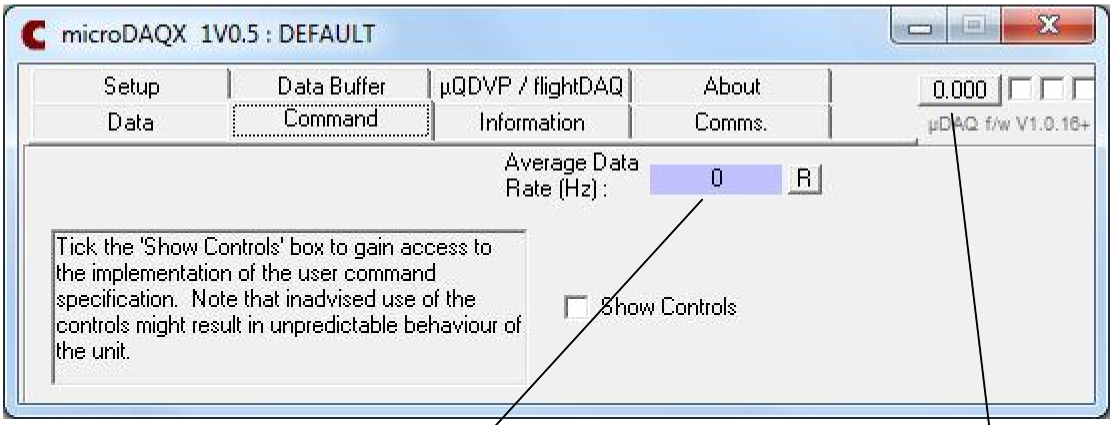


1	0.0000	9	0.0000	17	0.0000	25	0.0000	33	0.0000	41	0.0000	49	0.0000	57	0.0000
2	0.0000	10	0.0000	18	0.0000	26	0.0000	34	0.0000	42	0.0000	50	0.0000	58	0.0000
3	0.0000	11	0.0000	19	0.0000	27	0.0000	35	0.0000	43	0.0000	51	0.0000	59	0.0000
4	0.0000	12	0.0000	20	0.0000	28	0.0000	36	0.0000	44	0.0000	52	0.0000	60	0.0000
5	0.0000	13	0.0000	21	0.0000	29	0.0000	37	0.0000	45	0.0000	53	0.0000	61	0.0000
6	0.0000	14	0.0000	22	0.0000	30	0.0000	38	0.0000	46	0.0000	54	0.0000	62	0.0000
7	0.0000	15	0.0000	23	0.0000	31	0.0000	39	0.0000	47	0.0000	55	0.0000	63	0.0000
8	0.0000	16	0.0000	24	0.0000	32	0.0000	40	0.0000	48	0.0000	56	0.0000	64	0.0000

In this case, a 64 channel scanner is displayed. If a 32 or 16 channel scanner is used, the unused channels will be blank.

[8] Re-zeroing the data

The data can be re-zeroed at any screen using the 0.000 button next to the main tab bar. Here it is shown with the Command tab:



Average Data Rate (Hz) : 0

Tick the 'Show Controls' box to gain access to the implementation of the user command specification. Note that inadvised use of the controls might result in unpredictable behaviour of the unit.

☐ Show Controls

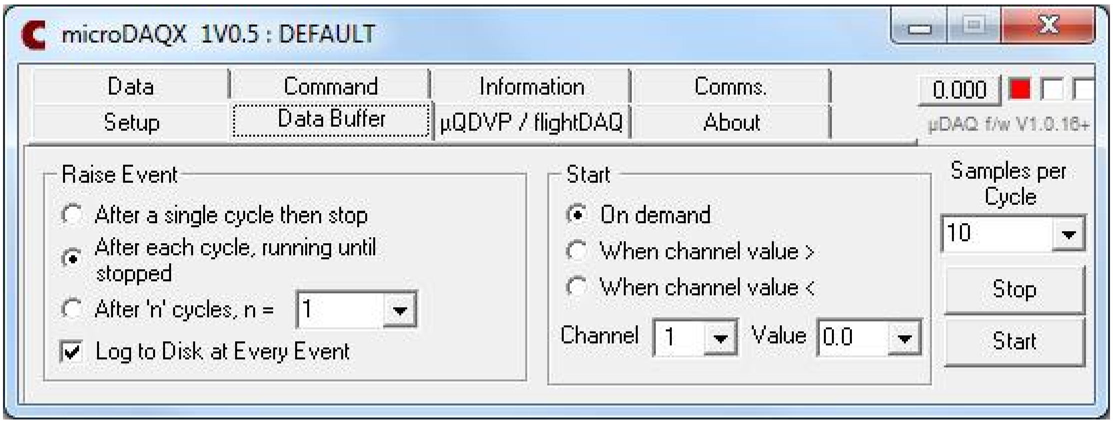
Click on the Rezero button and a re-zero will take place. (The number of samples used in a re-zero can be altered by running the microDAQSetup software.) An acknowledge will appear on the header bar when the re-zero command is sent.

The average data rate shows the speed at which data is being received by the software. (The ‘R’ resets this displayed rate)

[9] Logging the Data

The microDAQX software can log data to disk in CSV format. The file name is that listed under logging in the setup tab.

Open the Data Buffer tab as follows:



Raise Event

○ After a single cycle then stop

● After each cycle, running until stopped

○ After 'n' cycles, n = 1

☑ Log to Disk at Every Event

Start

● On demand

○ When channel value >

○ When channel value <

Channel 1 Value 0.0

Samples per Cycle 10

If this window is configured as above, when the start button is pressed, the software will write 10 samples at a time to the file and will keep running until the stop button is pressed (the left hand light on the top right hand corner will be lit when data is being logged. For more information, look at the Data Buffer tab section of this manual (section 4.7).

2 Quick Start (RS232)

This section outlines the steps necessary to start collecting data with a single microDAQ using RS232. This section does not deal with all the microDAQ options which are described further on in this manual or in the microDAQ operating manual.

This section presupposes that the microDAQ has been setup to transmit data onto the RS232 port. This is done by running the microDAQSetup wizard (see the microDAQ operating manual). This is NOT the default factory setting.

[1] Connecting up.

The microDAQ should be connected to the PC with a suitable RS232 lead. Connect the power lead to the microDAQ.

[2] Installing the software.

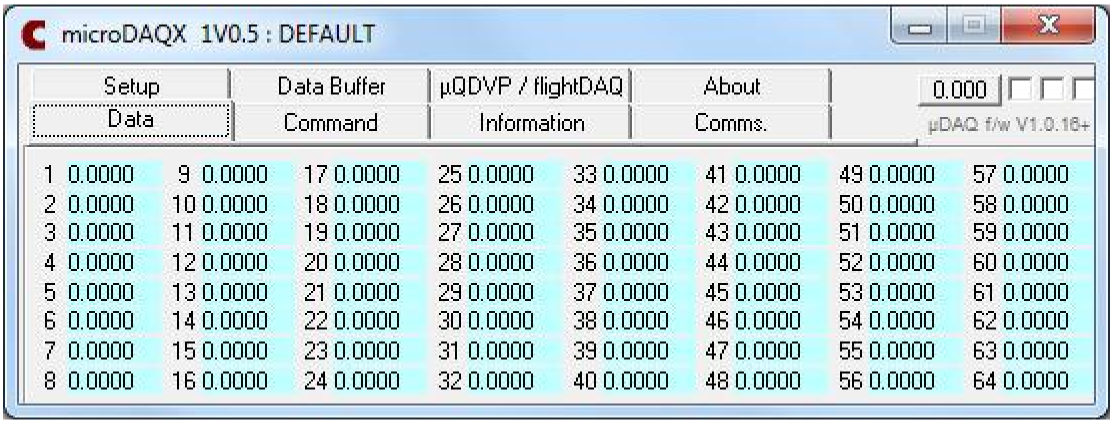
The software is installed by running the setup.exe in the microDAQX directory on the CD supplied. Follow the instructions until the installation is complete.

[3] Powering up the microDAQ.

Turn on the power to the microDAQ. The blue LED will light for between 10 and 40 seconds depending on the calibration type and number of scanner channels. The blue LED will start to flash.

[4] Running the software.

Run microDAQX.exe from the menu bar and the following will appear:

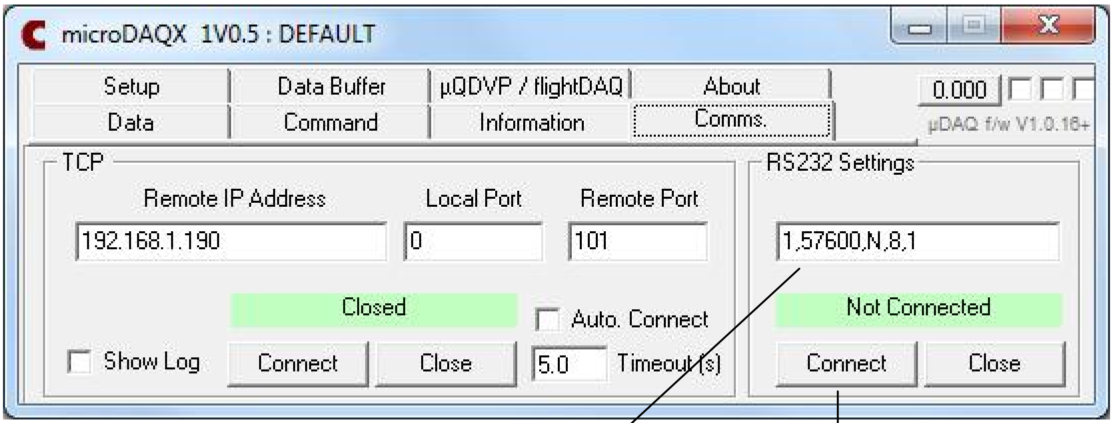


1	0.0000	9	0.0000	17	0.0000	25	0.0000	33	0.0000	41	0.0000	49	0.0000	57	0.0000
2	0.0000	10	0.0000	18	0.0000	26	0.0000	34	0.0000	42	0.0000	50	0.0000	58	0.0000
3	0.0000	11	0.0000	19	0.0000	27	0.0000	35	0.0000	43	0.0000	51	0.0000	59	0.0000
4	0.0000	12	0.0000	20	0.0000	28	0.0000	36	0.0000	44	0.0000	52	0.0000	60	0.0000
5	0.0000	13	0.0000	21	0.0000	29	0.0000	37	0.0000	45	0.0000	53	0.0000	61	0.0000
6	0.0000	14	0.0000	22	0.0000	30	0.0000	38	0.0000	46	0.0000	54	0.0000	62	0.0000
7	0.0000	15	0.0000	23	0.0000	31	0.0000	39	0.0000	47	0.0000	55	0.0000	63	0.0000
8	0.0000	16	0.0000	24	0.0000	32	0.0000	40	0.0000	48	0.0000	56	0.0000	64	0.0000

The RS232 communications will need to be initialised so no data will be present at this time.

[5] Initialising the RS232 port

The RS2323 port is initialised and connected in the comms tab as follows:



0.000

Remote IP Address: 192.168.1.190

Local Port: 0

Remote Port: 101

Closed

□ Auto. Connect

□ Show Log

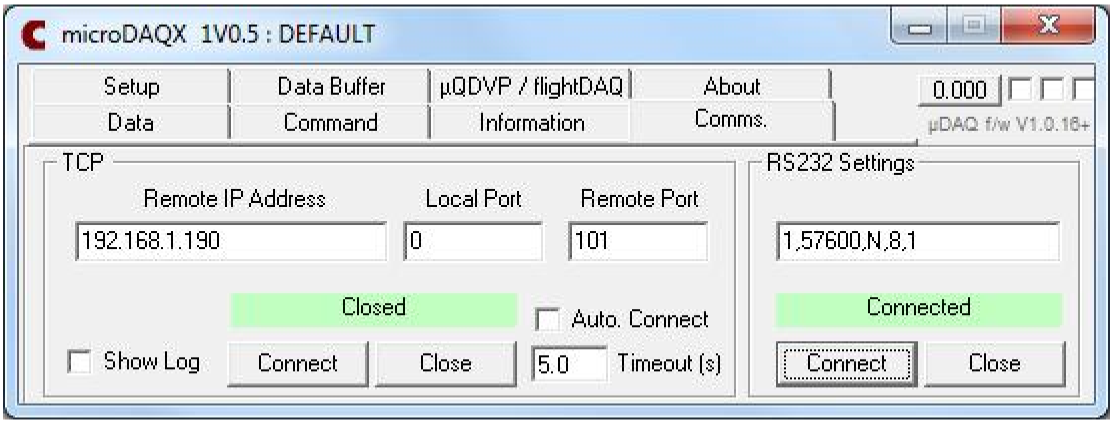
Timeout (s): 5.0

RS232 Settings: 1,57600,N,8,1

Not Connected

Ensure the correct RS232 setting are present.

Then select Connect.



0.000

Remote IP Address: 192.168.1.190

Local Port: 0

Remote Port: 101

Closed

□ Auto. Connect

□ Show Log

Timeout (s): 5.0

RS232 Settings: 1,57600,N,8,1

Connected

This window shows that the microDAQ has connected via RS232. The lights on the upper right hand corner will light periodically to show that data is being received.

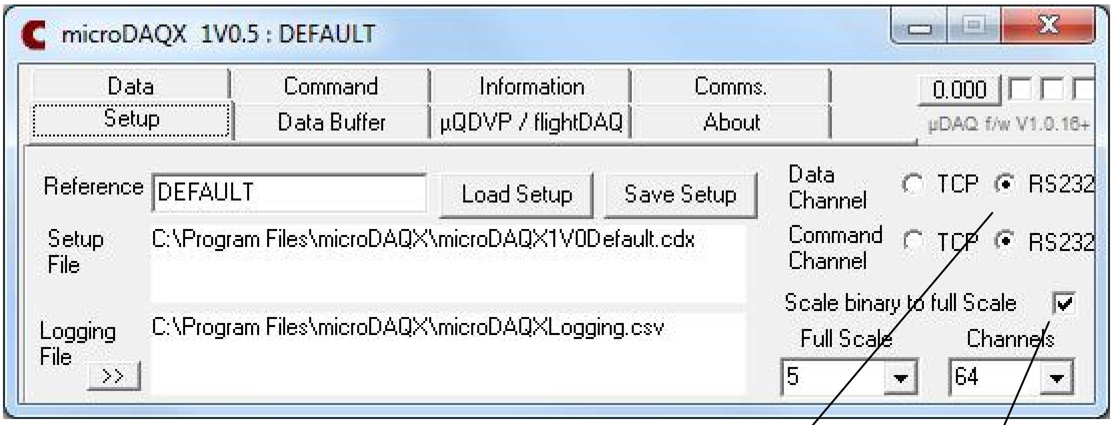
NOTE: If the system does not connect, look at the following:

Are the RS232 settings correct? These are :
comm port, baud rate, parity, number of data bits, number of stop bits

Is the right comm port selected?
Is the right baud rate selected. This can be changed with the microDAQSetup software.

[6] Saving the settings.

It is important that the settings are saved. Do this by selecting the Setup tab as follows:



Reference DEFAULT

Setup File C:\Program Files\microDAQX\microDAQX1V0Default.cdx

Logging File C:\Program Files\microDAQX\microDAQXLogging.csv

Data Channel ○ TCP ● RS232

Command Channel ○ TCP ● RS232

Scale binary to full Scale ☑

Full Scale 5

Channels 64

First ensure that the data and command channel are set to RS232

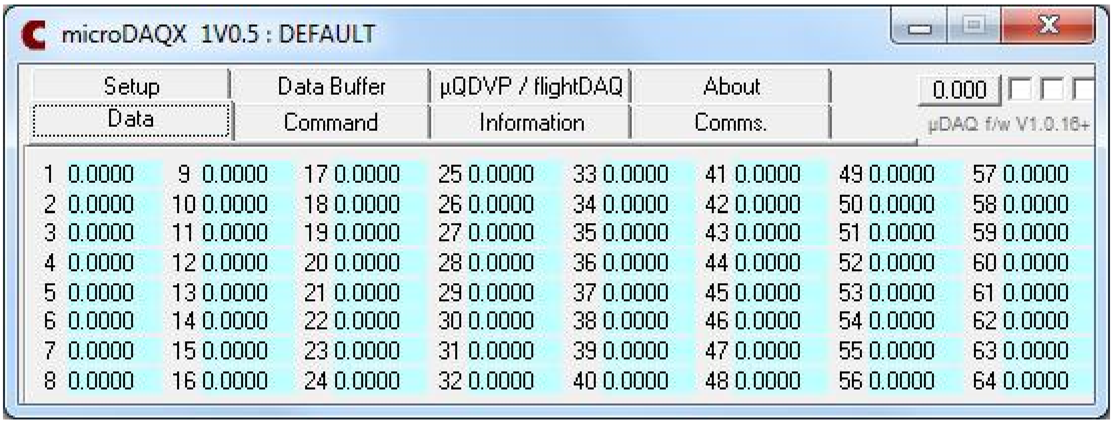
Then ensure that the scale binary to full scale is selected. This will scale the 16 bit number transmitted from the microDAQ into the appropriate units. The number of channels and full scale will also need to be correct.

NOTE: The full scale setting can be used to change the units. In this case, the scanner has a 5 psi full scale. If you wanted the units to be in, say, Pascals then add the appropriate full scale in Pascals into this text box (5 x 6894.757).

Then save the setting to the default file by overwriting the existing setup file (microDAQX1V0Default.cdx).

[7] Viewing the data

The data can now be viewed with the data tab as follows:

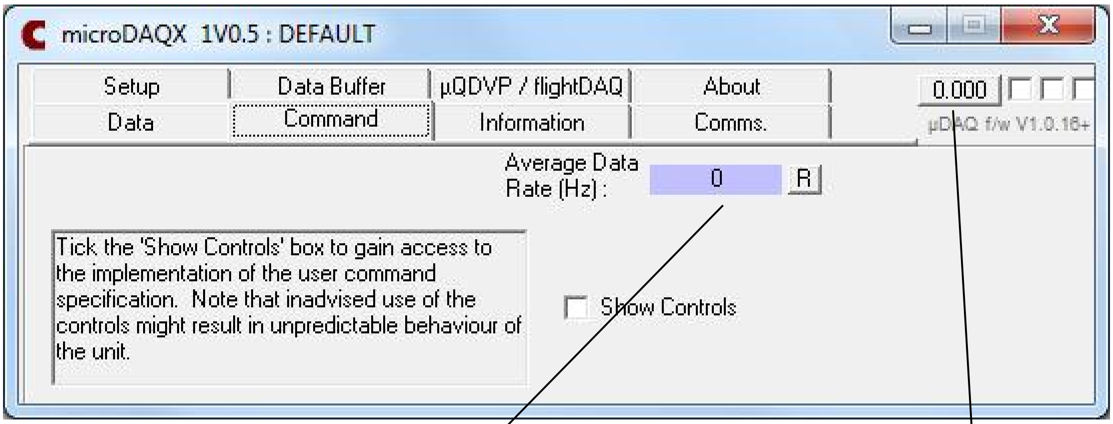


1	0.0000	9	0.0000	17	0.0000	25	0.0000	33	0.0000	41	0.0000	49	0.0000	57	0.0000
2	0.0000	10	0.0000	18	0.0000	26	0.0000	34	0.0000	42	0.0000	50	0.0000	58	0.0000
3	0.0000	11	0.0000	19	0.0000	27	0.0000	35	0.0000	43	0.0000	51	0.0000	59	0.0000
4	0.0000	12	0.0000	20	0.0000	28	0.0000	36	0.0000	44	0.0000	52	0.0000	60	0.0000
5	0.0000	13	0.0000	21	0.0000	29	0.0000	37	0.0000	45	0.0000	53	0.0000	61	0.0000
6	0.0000	14	0.0000	22	0.0000	30	0.0000	38	0.0000	46	0.0000	54	0.0000	62	0.0000
7	0.0000	15	0.0000	23	0.0000	31	0.0000	39	0.0000	47	0.0000	55	0.0000	63	0.0000
8	0.0000	16	0.0000	24	0.0000	32	0.0000	40	0.0000	48	0.0000	56	0.0000	64	0.0000

In this case, a 64 channel scanner is displayed. If a 32 or 16 channel scanner is used, the unused channels will be blank.

[8] Re-zeroing the data

The data can be re-zeroed at any screen using the 0.000 button next to the main tab bar. Here it is shown with the Command tab



Average Data Rate (Hz) : 0

Tick the 'Show Controls' box to gain access to the implementation of the user command specification. Note that inadvised use of the controls might result in unpredictable behaviour of the unit.

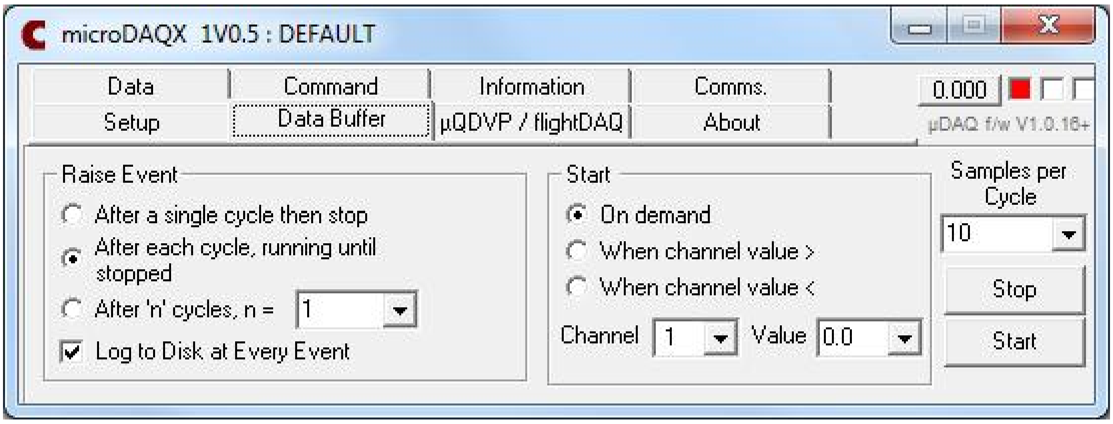
Click on the Rezero button and a re-zero will take place. (The number of samples used in a re-zero can be altered by running the microDAQSetup software.) An acknowledge will appear on the header bar when the re-zero command is sent.

The average data rate shows the speed at which data is being received by the software. (The 'R' resets this displayed rate)

[9] Logging the Data

The microDAQX software can log data to disk in CSV format. The file name is that listed under logging in the setup tab.

Open the Data Buffer tab as follows:



Raise Event
○ After a single cycle then stop
● After each cycle, running until stopped
○ After 'n' cycles, n = 1
☑ Log to Disk at Every Event

Start
● On demand
○ When channel value >
○ When channel value <
Channel 1 Value 0.0

Samples per Cycle 10

If this window is configured as above, when the start button is pressed, the software will write 10 samples at a time to the file and will keep running until the stop button is pressed (the left hand light on the top right hand corner will be lit when data is being logged. For more information, look at the Data Buffer tab section of this manual (section 4.7).

3 microDAQX Introduction and Overview

microDAQX has been compiled as a versatile interface for microDAQ units running in default user mode and connected to a PC by either RS232 or TCP over Ethernet. The software may be run standalone by a user with minimal technical expertise, or used as an ActiveX component within a higher level application such as a bespoke program or Microsoft (MS) EXCEL. Use of the program as a component requires an intermediate degree of programming knowledge.

Running standalone, the user controls on the software implement microDAQ's user interface commands (see document "microDAQ User Programming Guide"). A user may read scanner data directly from microDAQ for basic diagnostic purposes and testing, or log received data to disk. Additionally, slow rate data may be read directly into EXCEL via a DDE link with the data display.

The main benefit of the microDAQX application however, is its ability to be instanced from another application, whether a user's own or from EXCEL or similar. Users may create their own data logging and display applications within a spreadsheet without the real time concerns of interfacing at a hardware level. Once instanced from the application, microDAQX provides a public interface that implements the command set, some status information and the logging functions available via controls on the form. microDAQX provides data buffering for the application, signalling via events that the application should retrieve the last buffer full of data.

Settings for microDAQX are stored in a user named file, allowing multiple setup files for multiple instances of the program. For user applications employing multiple microDAQ's, microDAQX offers an economical means of connecting them simultaneously into a high level application.

Note that although the user command set allows the alteration of a certain subset of microDAQ settings at run time, these changes are not stored to microDAQ's non volatile memory. Operating settings required at start-up should be selected as usual via the microDAQSetup front end software.

microDAQX also includes a built-in Control Centre to allow multiple microDAQs to be controlled from a central point, with the possibility of all data acquisitions being combined into one log file. All information on the control centre is detailed later in this document.

microDAQ firmware V1.0.16 (or flightDAQ firmware V1.0.18) or later is required to use the latest version of the microDAQX front end software.

4 User Interface from the Standalone Application.

4.1 Overview.

As a standalone application, microDAQX can be run to receive data from microDAQ running in its standard user mode. Data may be logged to a comma separated value (CSV) text file and all of the user commands may be activated from the controls on the application's form. User controls and displays are organised into a number of different frames by category and are selected by clicking on the relevant tab.

Additionally the microDAQX Control Centre can be run from the appropriate shortcut in Windows. This brings up the main control centre form for controlling multiple microDAQs, instead of the normal microDAQX form. See section 4.9 for more details on this.

The following sub-sections detail the user controls of the microDAQX form, and their functions by category.

4.2 Setup Tab.

The controls for the Setup tab are shown in figure 4.1 and their function detailed in table 4.1. microDAQX stores and reads settings from a proprietary format file with extension '.cdx'. Alterations to the current settings may be saved to a user chosen filename with the 'Save Setup' button on the form, similarly alternative setups may be loaded and applied with the 'Load Setup' button.

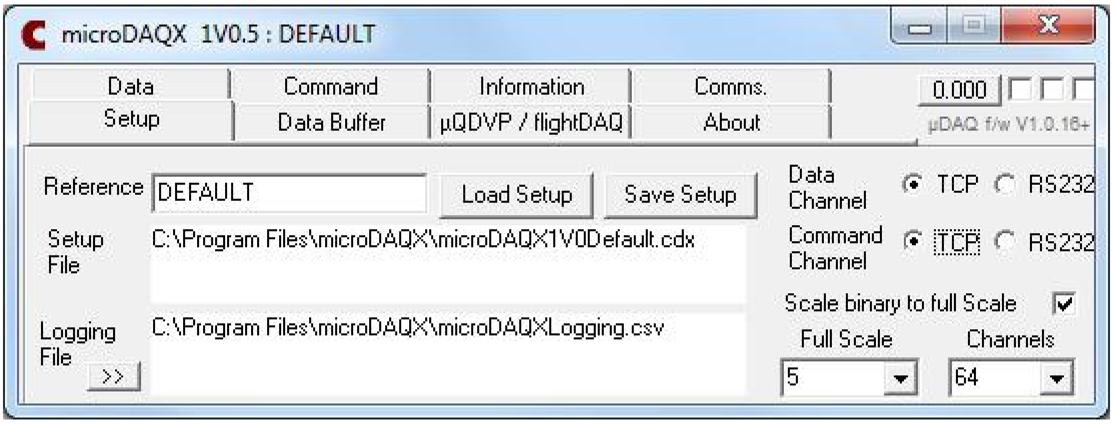
On startup, the application looks for a file 'microDAQX1V0Default.cdx' in the application's exe directory - preferred default settings should be written to this file. Starting for the first time, or if the file is unavailable, a file with typical default settings will be created and the settings applied.

Every setup file contains a 'reference' which is a text string, word or phrase to identify that particular setup. The purpose of the reference is to be able to identify a particular instance of the application when running multiple copies for multiple microDAQ's attached to a network for example. The reference appears appended to the caption in the application's main form and so will also appear on the Windows task bar.

microDAQX offers the user a basic logging to file facility controlled through the Data Buffer tab. The name of the CSV logging file may be selected by clicking on the button labeled '>>' next to the logging filename display box. If the logging option is active, data will be written to the desired file at the completion of each buffer cycle.

microDAQX allows access to the microDAQ over both RS232 serial and Ethernet (TCP) connections. Provision is made for splitting the sent commands and received data over either channel, so for example, microDAQ commands may be sent over RS232 while data is received over TCP. The channel for commands and data is selected from the option buttons on the setup tab.

microDAQ’s full scale may be set on the drop down box on the setup tab for the purpose of scaling received binary data to engineering units. The ‘Scale binary to full scale’ check box selects the option of displaying received binary data as engineering units scaled to the selected value. It is more efficient to receive binary data from the microDAQ and apply the conversion in microDAQX, especially at high data rates.



microDAQX 1V0.5 : DEFAULT

Reference: DEFAULT

Setup File: C:\Program Files\microDAQX\microDAQX1V0Default.cdx

Logging File: C:\Program Files\microDAQX\microDAQXLogging.csv

Data Channel: ● TCP ○ RS232

Command Channel: ● TCP ○ RS232

Scale binary to full Scale ☑

Full Scale: 5

Channels: 64

Figure 4.1, The Setup Tab.

User Control	Function
Reference	Type the name a particular setup file is to be identified by - this appears in the main form’s caption.
Setup file	The name of the currently loaded setup file.
Logging file	The name of the file used for logging received data.
‘>>’ button	Click to select an alternative logging file.
‘Load Setup’ button	Click to choose a setup file, and apply its settings.
‘Save Setup’ button	Click to save the currently selected settings to a user chosen filename.
‘Data’ option buttons	Select to receive data from microDAQ over either RS232 or TCP.
‘Command’ option buttons	Select to send user commands to microDAQ over either RS232 or TCP.
Full scale dropdown.	Select the pressure scanner’s full scale.
Channels dropdown.	Select the number of active channels set in microDAQ’s options.
Scale binary to full scale checkbox	Enable conversion of binary data to engineering units before display or logging.

Table 4.1, User Control Functions for the Setup Tab.

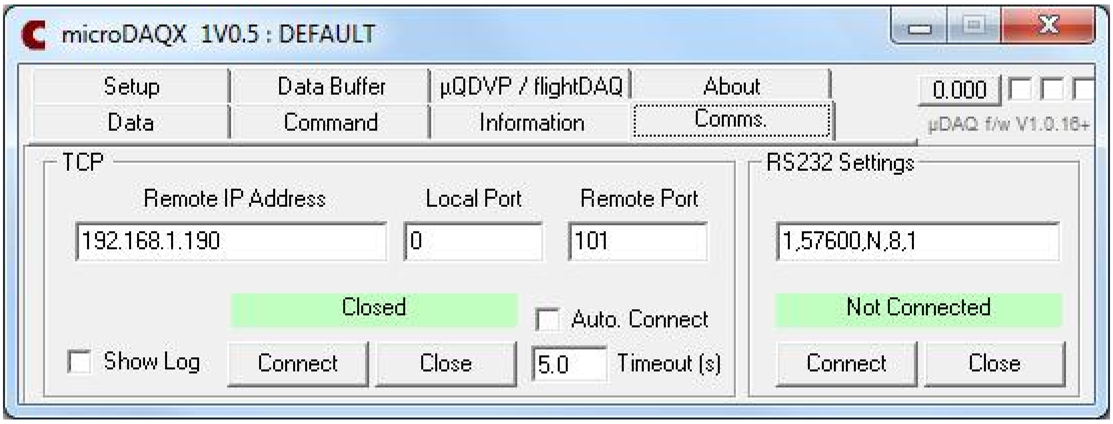
4.3 Communications Tab.

The communications tab gives access to the settings for both RS232 and TCP channels used to communicate with an attached microDAQ. The TCP IP settings refer to the attached microDAQ, i.e. its designated IP address. The microDAQ should be assigned an IP address compatible with the network it is to run on (contact the network administrator), and a local port for the PC should be chosen. The absolute value is unimportant as long as it is not used by any other service, however leaving the value set at 0 allows the system to automatically allocate a free port number. The remote port value is related to the chosen port within microDAQ, and is fixed at 101; the value should therefore not be altered.

The ‘Auto. Connect’ option makes the software attempt to maintain a connection with a running microDAQ – no received data within the chosen timeout period forces the software to disconnect and reconnect, making the assumption that there has been a communications problem. The local port should be set to 0 to use this option, and it should not be used if microDAQ is to sit in standby. A quiet microDAQ causes the connection to cycle off and on, which may be disruptive to a user’s application.

The settings for the RS232 communications are shown and set in the text box, and are in the usual format of number of com. port, baud rate, parity, data bits, number of stop bits. microDAQ's serial parameters are fixed other than the baudrate, which may be selected and programmed from the microDAQSetup front end software, and should match the value set in microDAQX.

Both channels may be manually connected and disconnected with their respective 'Connect' and 'Close' buttons, and their current status is shown in the green display boxes. The communications tab and its control functions are shown and detailed in figure and table 4.2.


microDAQX 1V0.5 : DEFAULT

TCP

Remote IP Address: 192.168.1.190

Local Port: 0

Remote Port: 101

Closed

□ Auto. Connect

□ Show Log

5.0 Timeout (s)

RS232 Settings: 1,57600,N,8,1

Not Connected

0.000

µDAQ f/w V1.0.16+

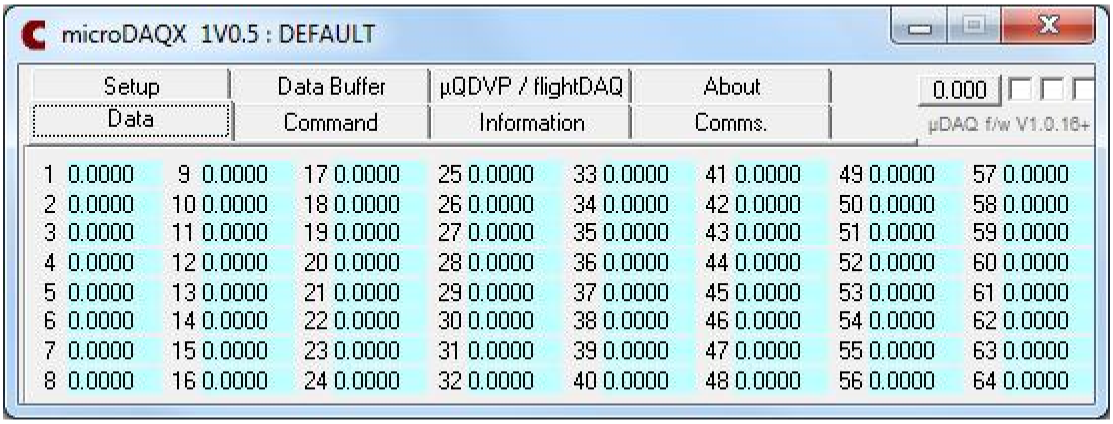
Figure 4.2, The Communications Tab.

User Control	Function
Local Port	Set the local port on the PC for TCP communication with microDAQ.
Remote Port	The port on microDAQ used for TCP communication - refer to current microDAQ documentation.
IP	microDAQ's IP address, as set from the front end software - must be compatible with the network - refer to microDAQ documentation and network administrator.
Socket state	Current state of the TCP communications socket, and display of other information concerning TCP communication.
Connect button (TCP)	Attempt to make a connection to microDAQ using the parameters shown.
Close button (TCP)	Close the connection to microDAQ.
RS232 settings text box	Settings for the RS232 serial communication - com. port, baudrate, parity, bits, stop bits.
RS232 status display	Shows connected or not for the shown settings.
Connect button (RS232)	Try to connect to the desired port with the settings shown.
Close button (RS232)	Close the connection with the RS232 port.
Auto connect	Option to close and remake the TCP connection should no data be received within the timeout period.
Show log checkbox	Shows system information (re data buffering).

Table 4.2, User Control Functions for the Communications Tab.

4.4 Data Tab.

The data tab (shown in figure 4.3), displays the last received data from microDAQ, and is compatible with both little ended binary data, and engineering unit data protocols. Received data packets of pressure readings are verified and split into their channels for display. The displays have the dual function of providing a user with a visual confirmation of the measured pressures, and in addition are DDE aware. This allows the setting of links within for example EXCEL, such that live values from a microDAQ unit may be piped via microDAQX into a spreadsheet for further analysis, capture and display in charts. A link within an EXCEL spreadsheet cell should be of the form **"=microDAQX|Data!CHn"**, where n is the channel number (1-64). The value shown on the data tab of microDAQX should then be linked into the cell in the spreadsheet. Similarly, other DDE aware applications might be linked into microDAQX using the application name " microDAQX", and the link topic "Data!CHn".



1 0.0000	9 0.0000	17 0.0000	25 0.0000	33 0.0000	41 0.0000	49 0.0000	57 0.0000
2 0.0000	10 0.0000	18 0.0000	26 0.0000	34 0.0000	42 0.0000	50 0.0000	58 0.0000
3 0.0000	11 0.0000	19 0.0000	27 0.0000	35 0.0000	43 0.0000	51 0.0000	59 0.0000
4 0.0000	12 0.0000	20 0.0000	28 0.0000	36 0.0000	44 0.0000	52 0.0000	60 0.0000
5 0.0000	13 0.0000	21 0.0000	29 0.0000	37 0.0000	45 0.0000	53 0.0000	61 0.0000
6 0.0000	14 0.0000	22 0.0000	30 0.0000	38 0.0000	46 0.0000	54 0.0000	62 0.0000
7 0.0000	15 0.0000	23 0.0000	31 0.0000	39 0.0000	47 0.0000	55 0.0000	63 0.0000
8 0.0000	16 0.0000	24 0.0000	32 0.0000	40 0.0000	48 0.0000	56 0.0000	64 0.0000

Figure 4.3, The Data Tab.

Also visible in the figure, and visible from all tabs of the application, are the rezero button labelled ‘0.000’, and the live status indicators at the top right of the form. The indicator furthest to the right blinks for each new data packet received into microDAQX, the central blinks for a full output buffer (green if buffering is active, red otherwise) and the leftmost shows solid red for the buffering being active.

4.5 Command Tab.

Figures 4.4a & 4.4b show microDAQX's command tab, which allows the user a degree of manual control over the microDAQ unit while it is running. The actual command set available is detailed separately in the *'microDAQ user programming guide* document, and within microDAQX is implemented as user controls. The 'Status' command is implemented separately on the Information tab. Note that Figures 4.4a & 4.4b show the form after the ‘Show Controls’ check box has been ticked.

It should be emphasised that the controls that are associated with a command parameter (for example the data delivery rate of a channel) do not necessarily show the current state of the parameter within the microDAQ. The microDAQ powers up with the settings programmed into it from the microDAQSetup front end software and the command buttons on microDAQX allow the modification of certain parameters (to the values selected in the dropdown boxes), but only until the next reset. It is therefore unlikely that the parameter

value within the microDAQ unit matches that of the microDAQX display, unless the microDAQX controls have been used to change that value since a reset.
Clicking a command button formats and sends the appropriate command packet to the microDAQ. The microDAQ will respond with an acknowledge, however if the communication settings are set so both command and data use the same channel, unless data streaming is turned off beforehand, it is possible that this acknowledge will be lost in a data packet. Under these cirumstances, a message on the caption bar of the main form informs the user of no positive acknowledgement within the timeout period, however it is likely that the command has been implemented successfully. A successfully acknowledged command will also be shown in the caption bar.

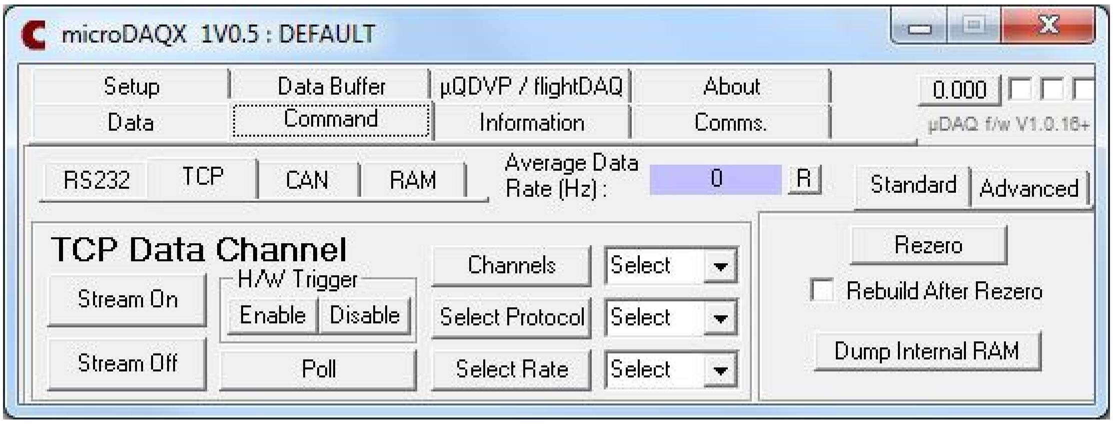


Figure 4.4a, The Command Tab – Standard controls.

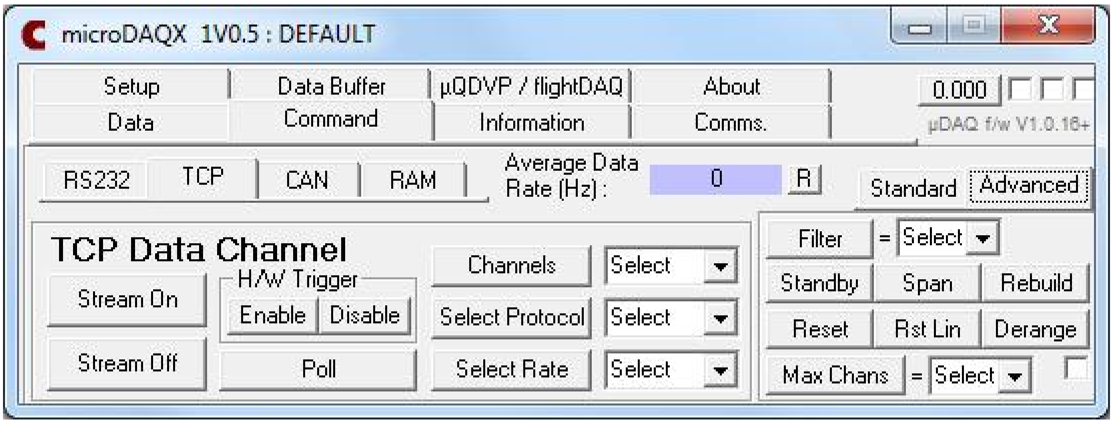


Figure 4.4b, The Command Tab – Advanced controls.

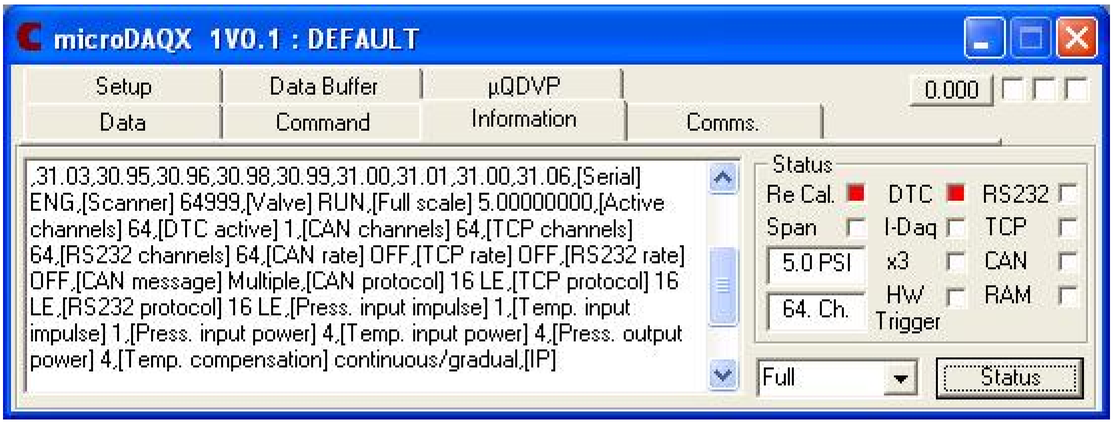
The three main data delivery channels (RS232, TCP and CAN) have symmetric controls, and are accessed through three separate sub tabs on the command tab. (Internal) RAM is used to initiate and stop logging to the internal RAM in the microDAQ – the standard and advanced control in the right hand frame are not applicable when using this ‘data delivery’ method. An overview of the control functions is provided in table 4.3.

User Control	Function
Stream On button (all channels)	Switches data streaming on, at the current delivery rate in the current protocol, for the channel.
Stream Off button (all channels)	Switches data streaming off for the channel.
Poll button (all channels)	Requests a single data packet (not available for Internal RAM)
H/W Trigger Enabble/Disable buttons	Enables or disables the hardware trigger feature of the microDAQ. Note enabling the trigger will automatically disable streaming
Channels	Sets the active channels delivered by the data channel.
Select Protocol (all channels)	Sets the current data protocol for a channel to that selected in the dropdown box.
Select Rate (all channels)	Sets the current data delivery rate for a channel to that selected in the dropdown box.
Rezero button and Rebuild after Rezero checkbox	Starts a rezeroing operation. If the associated checkbox is ticked then a rebuild of the calibration table is forced and the zero offset is applied to the calibration data on a per channel basis – data delivery is suspended.
0.000 button	Additional rezero button visible throughout application – identical in function to that above except that no rebuild is forced after the rezero even if the checkbox is ticked.
Filter button (advanced tab)	Sets the pressure output filter values to the selected value in the dropdown box. Settings for 'Off' and averaging between data packets are available.
Reset button (advanced tab)	Forces a software reset on the microDAQ, similar to a power cycle.
Rebuild Cal. button (advanced tab)	Forces a total immediate rebuild of the calibration data - data delivery is suspended.
Standby button (advanced tab)	Data streaming (all channels) off.
Max. Chans button (advanced tab)	Select the number of active channels on the microDAQ.
Derange button and checkbox (advanced tab)	Sets deranging on (checked), or off (unchecked) when the button is clicked - causes a total rebuild of calibration data.
Span button (advanced tab)	Perform a span operation – calculate the span correction to the applied pressure. Perform a rezero before using this function. Saves new values of zero offset and span correction to the microDAQ's non volatile memory, and rebuilds the calibration tables.
Rst Lin button (advanced tab)	Reset the unit's linear calibration to 0, 1.0 for all channels (zero/span respectively) and rebuild the calibration tables.
Dump Internal RAM	Initiates a dump of the current contents of the Internal RAM out of the chosen data delivery channel (obviously not applicable if data delivery is set to RAM !!). If logging has been enabled form the Data Buffer tab, then this RAM dump will be logged to the chosen CSV file (in the same format as if it had been captured directly from RS232, TCP or CAN)

Table 4.3, User Control Functions for the Command Tab.

4.6 Information Tab.

The information tab comprises a single button to request status data from the microDAQ, in addition to a dropdown to select from the three levels of status available (ie short, with temp. or full - see user commands). Once the button is clicked, the microDAQ should respond with the requested level of information, and this should be visible in the large text box on the tab, shown in figure 4.5.



```
,31.03,30.95,30.96,30.98,30.99,31.00,31.01,31.00,31.06,[Serial]
ENG,[Scanner] 64999,[Valve] RUN,[Full scale] 5.00000000,[Active
channels] 64,[DTC active] 1,[CAN channels] 64,[TCP channels]
64,[RS232 channels] 64,[CAN rate] OFF,[TCP rate] OFF,[RS232 rate]
OFF,[CAN message] Multiple,[CAN protocol] 16 LE,[TCP protocol] 16
LE,[RS232 protocol] 16 LE,[Press. input impulse] 1,[Temp. input
impulse] 1,[Press. input power] 4,[Temp. input power] 4,[Press. output
power] 4,[Temp. compensation] continuous/gradual,[IP]
```

0.000 □ □ □

5.0 PSI

64. Ch.

Full

Figure 4.5, The Information Tab.

The received information is parsed into its component parts for display. The status word received from the microDAQ at all levels of status request, is split into its component bits and displayed by the on/off indicators within the Status frame (on red, off white - pink until status requested). Note an additional flashing state can occur for the hardware trigger if it has inadvertently dropped out (e.g. due to loss of trigger pulse). If the next level of status (ie. including temperatures) is requested, then the temperature data is split out and displayed on the data tab. This will remain visible until it is replaced by the next incoming pressure data packet.

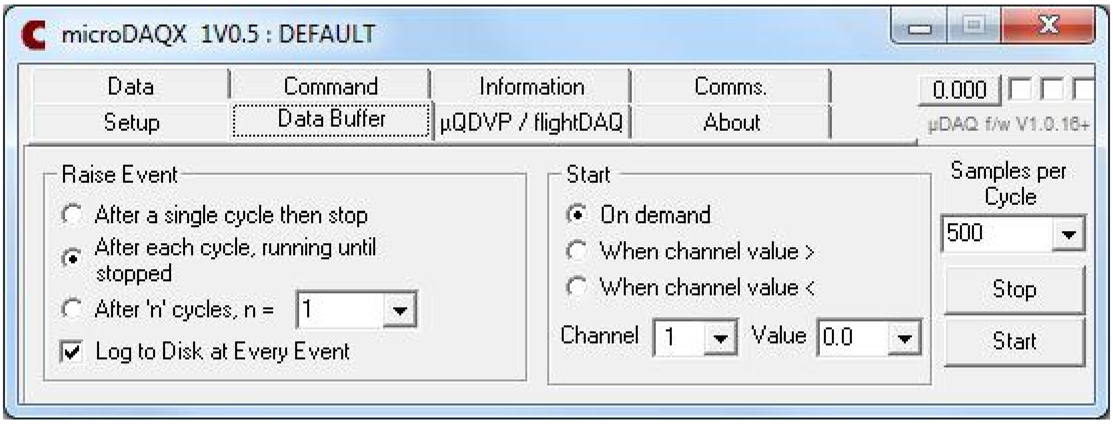
The highest level of status includes microDAQ's operating options, which are returned comma delimited with a parenthesised explicit description of the field. Values returned for microDAQ's full scale and the number of active scanner channels are split out and displayed in the two text boxes within the status frame.

Note that if writing software to look for particular options, these field descriptors should be used, as the number and order of fields may change for future versions of the microDAQ.

4.7 Data Buffer Tab.

When running as an ActiveX object instanced from a parent application, the data buffering functions give control over the frequency of signaling to the parent that there is data available for upload from microDAQX. As well as signaling the parent, it is possible to perform rudimentary data logging from microDAQX, writing values to a comma delimited text file.

This rudimentary data logging function will work from microDAQX running as a standalone application. The controls are shown in figure 4.6, and detailed in table 4.4



Raise Event

○ After a single cycle then stop
● After each cycle, running until stopped
○ After 'n' cycles, n = 1
☑ Log to Disk at Every Event

Start

● On demand
○ When channel value >
○ When channel value <

Channel 1 Value 0.0

Samples per Cycle 500

Figure 4.6, The Data Buffer Tab.

A buffering cycle comprises a fixed number of data points per channel received from the microDAQ, this number of samples being selected from the 'Samples per Cycle' dropdown. The 'Raise Event' frame controls when microDAQX signals that it has data ready to be read by a parent application in terms of completed buffer cycles. The total data points collected per channel is the product of the samples per cycle and the number of buffer cycles selected. The data passed up to the parent is therefore identical for the two settings examples of 20 cycles of 100 samples or 1 cycle of 2000 samples, though in the case of the latter the parent application will receive only 1 event rather than 20.

Buffering may be started and stopped by the two buttons on the form, whose function is replicated as a property over the public interface for parent applications. Alternatively, the buffering may be started and then wait for a trigger of a certain data value on a particular channel before starting the desired buffering regimen.

The furthest left indicator in the upper right of the main form shows red for buffering 'on', and the central indicator blinks when a buffer cycle is complete. On the completion of this buffer cycle the signaling event is raised by microDAQX.

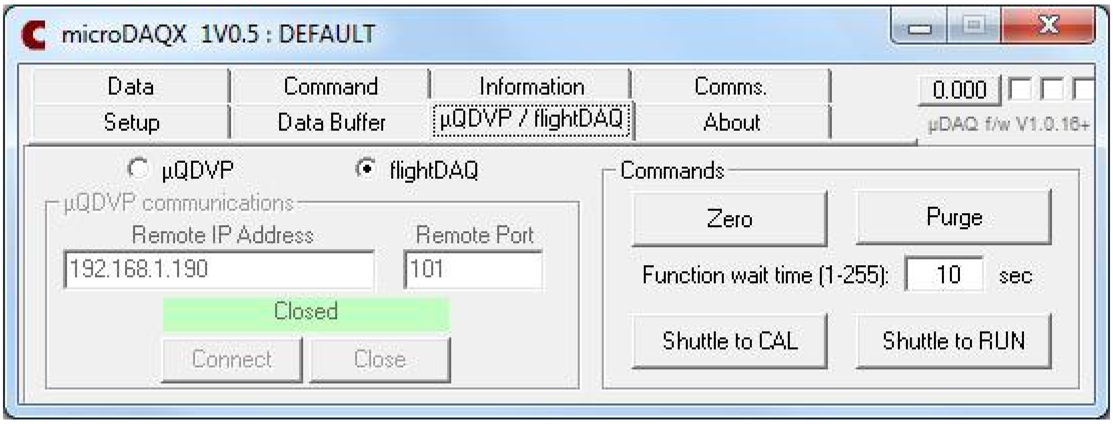
The 'Log to Disk…' checkbox forces the appending of the last buffered data to the selected logging file whenever the Raise Event condition is met.

User Control	Function
Raise Event options	Select how often an event is raised with a parent application, or the last data is written to disk, in terms of buffer cycles.
Log to Disk… checkbox	Check to append values to the selected CSV file when the raise event condition is met.
Start options	Choose between simple on/off, or wait for a suitable data value on a channel before starting.
Samples per Cycle dropdown	The number of data points (per channel) comprising a buffer cycle for the purposes of calculating the Raise Event timing.
Start button	Start the buffering/acquisition process
Stop button	Stop the buffering/acquisition process

Table 4.4, User Control Functions for the Logging Tab.

4.8 µQDVP / flightDAQ tab

This tab provides access to user commands for the valve control of a flightDAQ unit or an attached microQDVP unit – selected using the option buttons to the left side of the window. If the unit is a flightDAQ then the commands are sent using the same comms channel as all other commands within microDAQX and hence the µQDVP comms frame is disabled (as shown in figure. 4.7 below). However if the unit is a microDAQ connected to a microQDVP, then separate comms must be opened to do this. As of V1.05 of microDAQX, the only comunciation channel available is TCP, although the microQDVP commands can be sent via RS232 or CAN if written into a users own proprietry application. These commands are detailed in a separate section in the 'microDAQ User Programming Guide'.


microDAQX 1V0.5 : DEFAULT

○ µQDVP ● flightDAQ

µQDVP communications

Remote IP Address: 192.168.1.190

Remote Port: 101

Closed

Function wait time (1-255): 10 sec

Figure 4.7 The µQDVP / flightDAQ Tab.

The µQDVP TCP IP settings refer to its designated IP address. The microQDVP comes with its own cut down version of the setup program (microQDVPSetup) from within which its IP address can be assigned. The microQDVP should be assigned an IP address compatible with the network it is to run on (contact the network administrator), that does not clash with the microDAQ IP address (or any other for that matter). The remote port value is related to the chosen port within microQDVP, which is the same as the microDAQ, fixed at 101. The value should therefore not be altered.

Manually connect to and disconnect from the microQDVP using the buttons provided and they current connection status will be shown in the green box.

The commands available for the flightDAQ & microQDVP control the valves within (and connected to) the unit, with appropriate configuration available for wait time, which should be set before the function is actuated. The functions performed by these commands are detailed in table 4.5, as follows:

Command	Description
Zero	Performs a Zero function: 1. Shuttle scanner to CAL mode 2. Wait for configured time (from text box) 3. Shuttle scanner to RUN mode
Purge	Performs a Purge function: 1. Shuttle scanner to CAL mode 2. Switch on external purge valve 3. Wait for configured time (from text box) 4. Switch off external purge valve 5. Shuttle scanner to RUN mode

Shuttle to CAL	Moves the scanner shuttle valve to CAL mode.
Shuttle to RUN	Moves the scanner shuttle valve to RUN mode.

Table 4.5, microQDVP commands for the µQDVP Tab.

4.9 microDAQ Control Centre

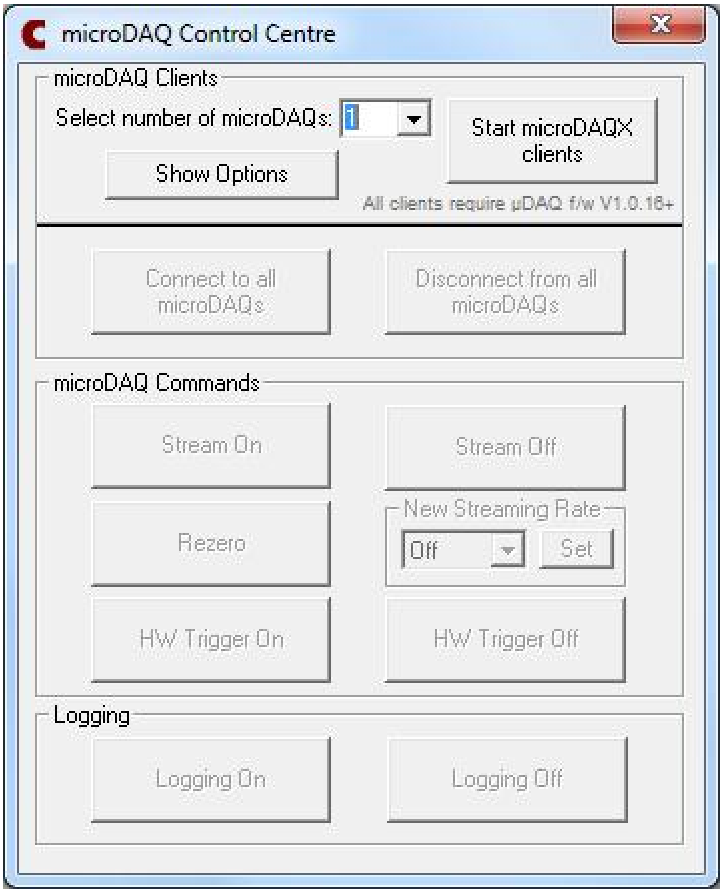
The microDAQ Control Centre runs from a separate shortcut and is used to setup and control several instances of microDAQX. This is particularly useful when the need arises to log data from several scanners at the same time. Each scanner can be connected to a microDAQ and the control centre would be used to connect instances of microDAQX to the microDAQs and acquire all the data into one log file for review in Excel (or similar).

Note there are a few pre-requisites when using the microDAQ Control Centre.

- The only communications method used is Ethernet (for both Data and Commands). RS232 is not allowed, simply due to its slow speed and the nature of its single peer-to-peer bus.
- The streaming data rate is the same for all microDAQs being controlled via the Control Centre. It is not possible (nor is it feasible) to acquire and log at different rates between the set of microDAQs being controlled.
- Likewise any hardware trigger pulse is sent to all microDAQs at the same time. Again it is not possible to trigger different microDAQs at different rates.
- The chosen samples per cycle setting is applied to all microDAQs being controlled by the Control Centre, for logging synchronicity
- If using the combined logging facility (see later) the logfile will be written as a binary file which must be run through the conversion function before it can be viewed in Excel, etc. This because combined logfiles can get very large, very quickly – the quickest and most efficient way of storing this data is in a binary file on the PC. Depending on the circumstances, it may even be necessary to run a RAMdisk on the PC and log the combined data to that because writing to RAM is even quicker than writing to a hard disk. This all depends on the PC system being used and is beyond the scope of this document.
- The more microDAQs being connected together, coupled with higher data acuqitision rate, the more likely it is that TCP packets could be missed due to collisions on the network. Chell therefore advise that if high data rates are to be used then the microDAQs should be connected to the PC on a dedicated network via a high speed managed network switch that supports the store and forward feature – the larger the packet buffer the better!!. This should mean all data acquired will get through to the PC without packet collisions and loss of data.
- At very high data rates with lots of microDAQX instances, it is possible that there may be too many Windows GDI (drawing) updates going on for the PC to keep up with TCP reception and data writing. It has been found in extreme circumstances that this interrupts the reception of data packets and therefore packets get lost leading to loss of synchronicity between microDAQs. To protect against this, it is recommended that the majority, or all, microDAQX data windows should be minimised whilst logging data.

The following details the various controls and features of the microDAQ Control Centre.

a) Main screen



All clients require µDAQ f/w V1.0.16+

Figure 4.7, microDAQ Control Centre main screen

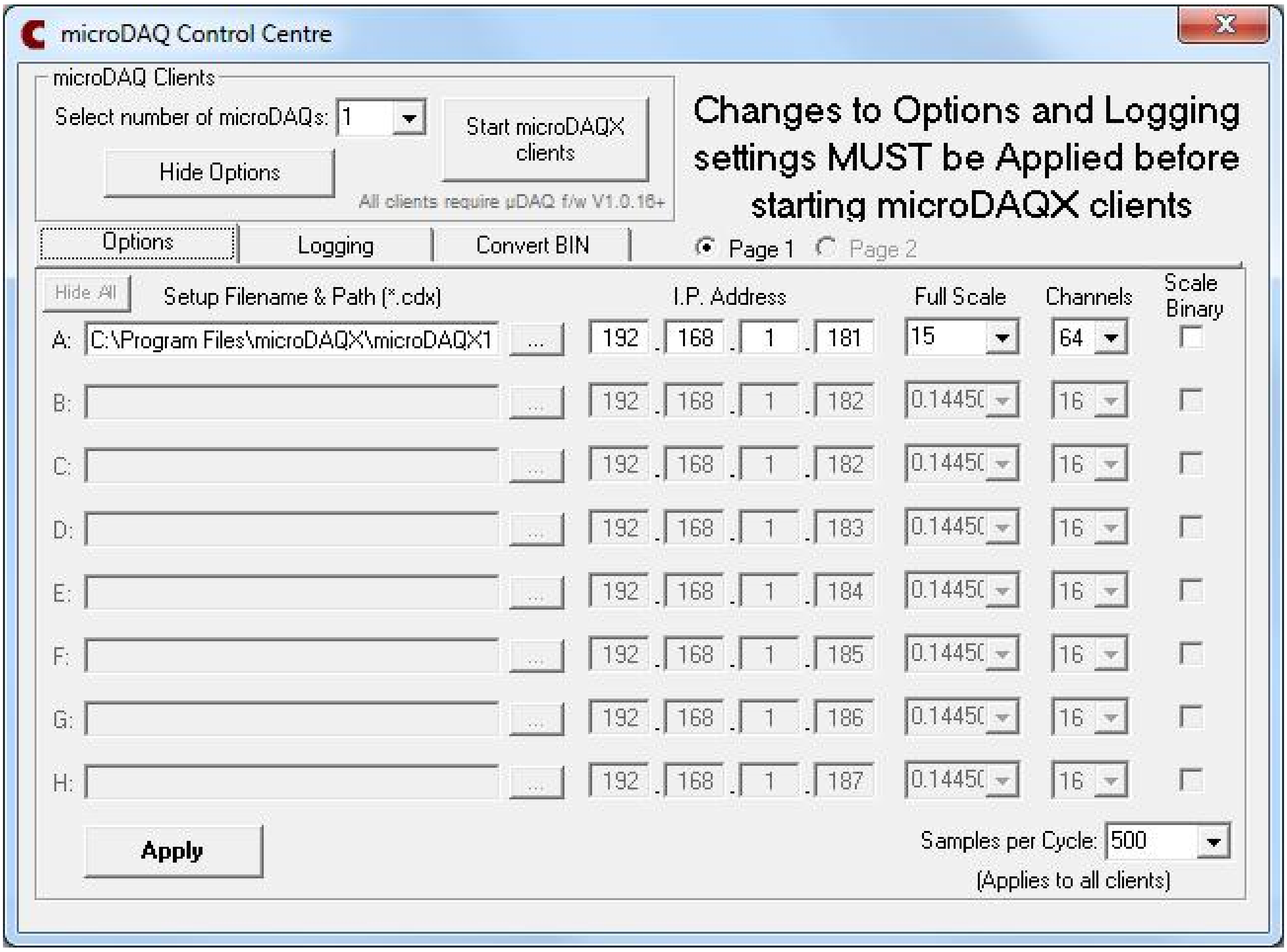
On startup, the Control Centre has very few options available. The general idea is that the user would setup the microDAQ settings first (via the Options screens – see later) and then start the appropriate number of microDAQX clients. Then the connect and disconnect buttons become active and allow the user to get each microDAQX client to connect to their microDAQ hardware. Once connection is achieved for all clients, the various commands and logging buttons become available to control and log data accordingly. These buttons simply emulate the user pressing on the various command buttons in a microDAQX window but sychronise for all clients currently running at the same time. The following table details each controls function shown in figure 4.7.

User Control	Function
Number of microDAQs dropdown	Select the number of microDAQs (and hence number of microDAQX instances) that are to be controlled at the same time. A maximum of 8 microDAQs can be controlled at any one time.
Show Options button	Opens the options and logging windows for pre-configuration of the microDAQX instances. The combined binary logfile converter is found here as well.
Start microDAQX clients button	Starts (and stops) the microDAQX instances. Once the instances have been started, the Connect & Disconnect buttons become active.
Connect to all microDAQs button	Instructs each microDAQX instance to connect to its microDAQ via TCP. Note if there is a connection error with one or more microDAQs (e.g. due to an incorrect IP address) then the rest of the command and logging buttons will remain disabled. In this case correct the connection error and try to connect again.
Disconnect from all microDAQs	Instructs each microDAQX instance to close the TCP connection with

button	its microDAQ.
Stream On button	Starts data streaming at the current rate for all microDAQs.
Stream Off button	Stops data streaming for all microDAQs.
Rezero button	Instructs each microDAQ to start a rezero operation. Ideally this should be done with streaming switched off.
New Streaming Rate frame	Selects a new streaming rate for all microDAQs. Press the Set button to perform the operation. Valid rates are the same as is available in the TCP command controls in a single microDAQX instance.
HW Trigger On button	Enables the Hardware Trigger feature of all microDAQs.
HW Trigger Off button	Disables the Hardware Trigger feature of all microDAQs.
Logging On button	Turns on the Logging feature. This opens the file stream(s) ready for logging the next set of data acquired.
Logging Off button	Turns off the Logging feature. This closes the file stream(s) and MUST be clicked before viewing any resultant CSV logfile in Excel, or converting the combined binary logfile.

Table 4.5, User Control Functions for the Control Centre main screen.

b) Options setup window



microDAQ Control Centre

microDAQ Clients

Select number of microDAQs: 1

All clients require µDAQ f/w V1.0.16+

Changes to Options and Logging settings MUST be Applied before starting microDAQX clients

● Page 1 ○ Page 2

	Setup Filename & Path (*.cdx)	I.P. Address	Full Scale	Channels	Scale Binary
A:	C:\Program Files\microDAQX\microDAQX1	192 . 168 . 1 . 181	15	64	□
B:		192 . 168 . 1 . 182	0.1445(	16	□
C:		192 . 168 . 1 . 182	0.1445(	16	□
D:		192 . 168 . 1 . 183	0.1445(	16	□
E:		192 . 168 . 1 . 184	0.1445(	16	□
F:		192 . 168 . 1 . 185	0.1445(	16	□
G:		192 . 168 . 1 . 186	0.1445(	16	□
H:		192 . 168 . 1 . 187	0.1445(	16	□

Samples per Cycle: 500

(Applies to all clients)

Figure 4.8, microDAQ Control Centre options setup window.

From this window, the user can set some of the required setup parameters for each microDAQX instance. The number of fields accessible depends on the number of microDAQs selected in the microDAQ clients frame. The Control Centre can cope with up to 16 microDAQs at once, so if more than 8 are selected, the Page 2 option button becomes available and can be clicked to access the second page of microDAQs for their setup parameters.

The most common parameters are available to be changed here, like I.P. address, scanner full scale and number of channels, etc. Other less common parameters should be changed (if necessary) from within the microDAQX instances once they have been started. Any changes made in this window MUST be applied before the microDAQX clients are started and if clients have already been started then no further changes will be allowed until they are stopped. As mentioned before, the Samples per Cycle parameter applies to all microDAQs. The following table lists all functions for the controls available in Figure 4.8 above (excluding the microDAQ Clients frame which has been detailed previously).

User Control	Function
Setup Filename & Path text boxes and '...' buttons	Sets the filename of the setup file (*.cdx) to use for each active microDAQX client. The textboxes are editable, but typically the user should click on the '...' button to open a standard Windows dialog and select the .cdx file to use. If the user requires a new setup file then it will be created with the default setup parameters.
I.P. addresses text boxes	Sets the I.P. address of the microDAQ being connected to by each microDAQX client.
Full Scale dropdown	Selects the full scale engineering units (in PSI) to use when displaying and logging data with the binary scaling option set. Valid full scales range from 4"WC (0.144509 PSI) to 100 PSI.
Channels dropdown	Selects the number of scanner channels to be displayed and logged for each microDAQ. Valid settings are 16, 32 or 64 channels.
Scale Binary checkbox	Sets whether the displayed and logged data will be represented in 16bit binary format or engineering units (PSI)
Samples per Cycle dropdown	Sets the number of samples collected before a logging event occurs. This setting applies to all microDAQs and effectively sets the Samples per Cycle parameter in microDAQX (see section 4.7 above)
Apply button	This causes any changes made to the setup parameters to be saved to the Setup file. **This button MUST be pressed if any changes are made, BEFORE starting the microDAQX clients.**
Page 1 & Page 2 option buttons	Switch between pages 1 & 2 of microDAQX setups, if more than 8 microDAQs selected
Hide All button	Minimises all microDAQX client windows. Click again to restore all windows. Only accessible once microDAQX clients have been started.
A: - P: labels	Click to minimise/ restore the associated microDAQX client window. Only accessible once microDAQX clients have been started.

Table 4.6, User Control Functions for the Control Centre options setup window.

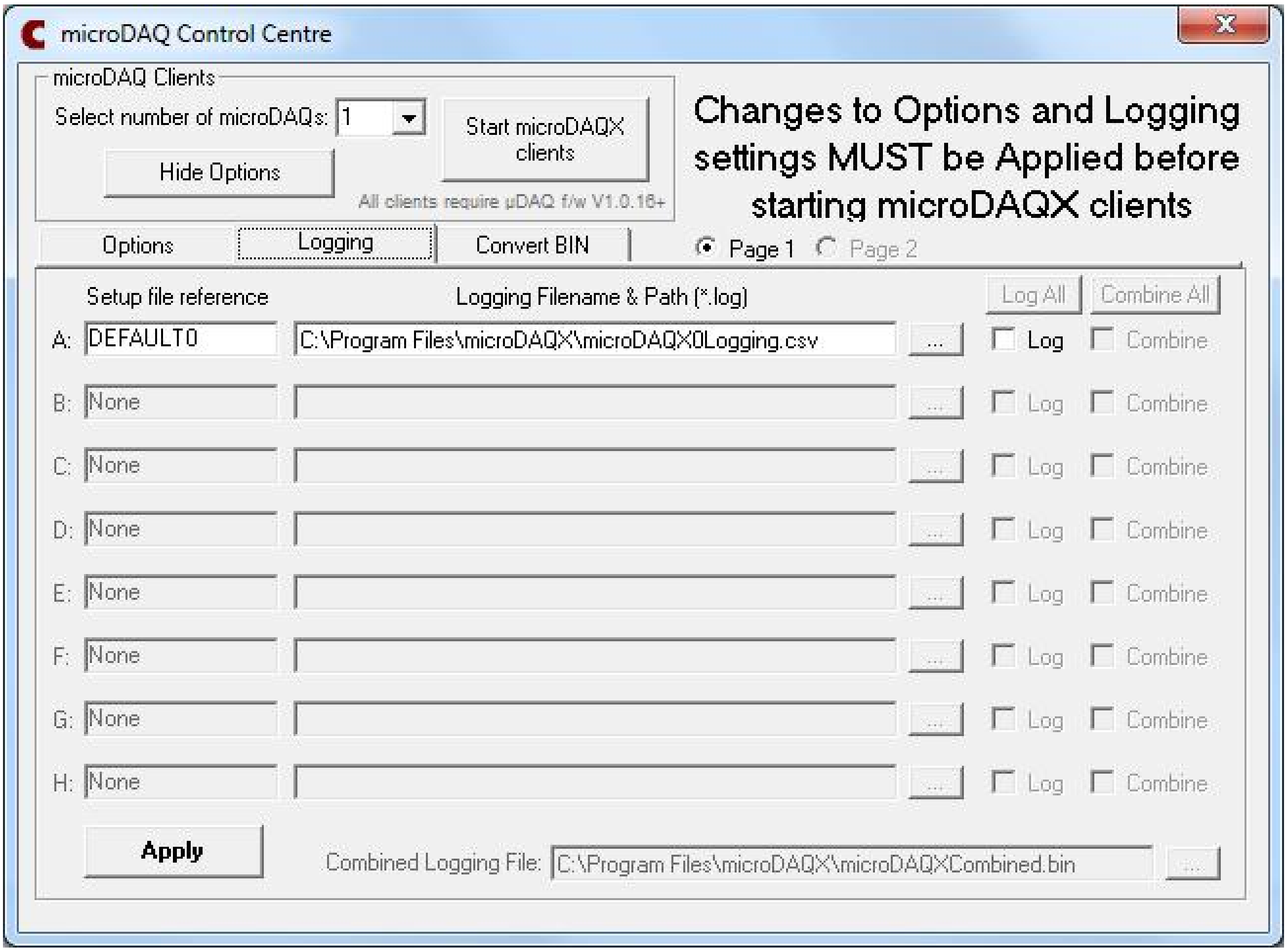
Note if a default setup file is created then the following default settings are applied (settings that are not applicable, i.e. RS232 settings, are not mentioned here):

- I.P. address = 192.168.1.18x ... where x is 0 to 7 depending on microDAQ 'number'.
- Local port = 0 (*)
- Remote port = 101 (*)
- Auto Connect = Off (*)
- Data & Command Channels = TCP (*)
- Scale Binary = Off
- Full Scale = 15 PSI
- Channels = 64
- Setup Reference = DEFAULTx ... where x = 0 to 7 depending on microDAQ 'number'.

- Logging Event = Raised after every cycle until logging stopped
- Logging Trigger = On demand
- Log To Disk = Off (*)
- Samples per Cycle = 20 (*)

NOTE: All settings marked with an asterisk (*) should not be changed from microDAQX when being used with the microDAQ Control Centre. Any changes to those settings outside of the Control Centre could result in unpredictable/undesirable operation and logging results from within the Control Centre.
In addition, be careful when changing the Logging Event and Trigger options – if you don't know what you are doing, this could equally result in unpredictable results in the logfile(s).

c) Logging setup window



microDAQ Control Centre

microDAQ Clients

Select number of microDAQs: 1

All clients require µDAQ f/w V1.0.16+

Changes to Options and Logging settings MUST be Applied before starting microDAQX clients

● Page 1 ○ Page 2

	Setup file reference	Logging Filename & Path (*.log)			
A:	DEFAULT0	C:\Program Files\microDAQX\microDAQX0Logging.csv	...	□ Log	□ Combine
B:	None		...	□ Log	□ Combine
C:	None		...	□ Log	□ Combine
D:	None		...	□ Log	□ Combine
E:	None		...	□ Log	□ Combine
F:	None		...	□ Log	□ Combine
G:	None		...	□ Log	□ Combine
H:	None		...	□ Log	□ Combine

Combined Logging File: C:\Program Files\microDAQX\microDAQXCombined.bin ...

Figure 4.9, microDAQ Control Centre logging setup window.

From this window the user can set the setup parameters relating to logging. The logging filename & path is for the individual CSV files created when logging is started, if you are NOT combining logs. The Log check box must be ticked for each active microDAQ that is required to be logged to disk. Once two or more microDAQs are made active, the Combine checkbox is available and can be ticked to add that microDAQs acquisitions to the combined binary logfile. When microDAQ acquisitions are marked for combined logging, the individual CSV files WILL NOT be generated by the microDAQX instances as well.

The combined logfile stores channel data in order from the ‘lowest’ selected microDAQ ID to the highest, assigning channel numbers from 1 through to the total number of combined channels. E.g if microDAQ ‘A’ and ‘C’ are selected for combined logging (both with 64 channel scanners attached) then the logged data in the combined logfile is stored for channels 1 to 128, where channels 1 to 64 are microDAQ ‘A’ data and channels 65 to 128 are microDAQ ‘C’ data.

Again, any changes made in this window MUST be applied before the microDAQX clients are started and if clients have already been started then no further changes will be allowed until they are stopped. The following table lists all functions for the controls available in Figure 4.9 above (excluding the microDAQ Clients frame which has been detailed previously).

User Control	Function
Setup file reference label	This serves as an indicator as to what setup file is being used to store the logging setup parameters.
Logging Filename & Path text boxes and ‘...’ buttons	Sets the filename and path of the CSV logfile to use for uncombined logging. Use the ‘...’ button to open a Windows dialog to select the CSV file and path to use.
Log & Combine checkboxes	Determines whether the associated microDAQ data is logged to a file. If only the Log check box is ticked then the logging is written to the CSV file set in the associated text box. If Combine is ticked as well then the acquired data is added to the Combined Logging file.
Log All & Combine All buttons	Provide quick ways to tick all Log & Combine checkboxes.
Combined Logging File text box and ‘...’ button	Sets the filename and path of the combinaed binary logfile (.BIN) to use for combined logging. Use the ‘...’ button to open a Windows dialog to select the BIN file and path tio use. (Note any existing binary logfile will be cleared and overwritten when logging starts)
Apply button	This causes any changes made to the logging setup parameters to be saved to the Setup file. **This button MUST be pressed if any changes are made, BEFORE starting the microDAQX clients.**
Page 1 & Page 2 option buttons	Switch between pages 1 & 2 of microDAQX logging setups, if more than 8 microDAQs selected.

Table 4.7, User Control Functions for the Control Centre logging setup window.

d) Convert BIN window

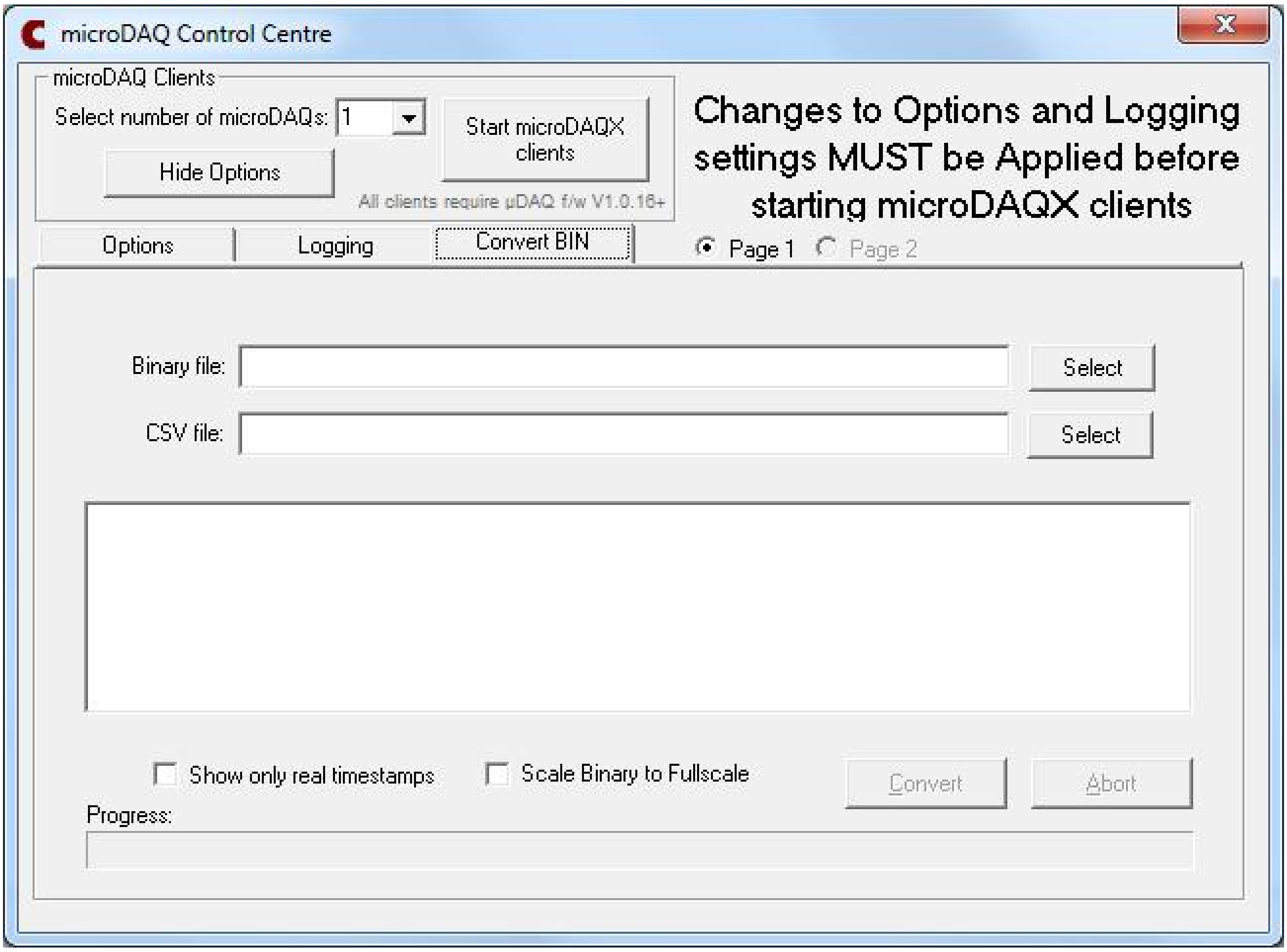

microDAQ Control Centre
microDAQ Clients
Select number of microDAQs: 1
Start microDAQX clients
Hide Options
All clients require µDAQ f/w V1.0.16+
Changes to Options and Logging settings MUST be Applied before starting microDAQX clients
Options
Logging
Convert BIN
● Page 1 ○ Page 2
Binary file:
Select
CSV file:
Select
□ Show only real timestamps
□ Scale Binary to Fullscale
Convert
Abort
Progress:

Figure 4.10, microDAQ Control Centre binary logfile converions window.

This window is used to convert a previously acquired binary logfile into a more user friendly CSV file that can be viewed in Excel or similar. After selecting the source binary file and destination CSV file the user simply clicks Convert and the information window shows the progress of the conversion. The user can choose to only show timestamps in the csv file that are 'real' (i.e. Don't duplicate the timestamps for all samples in one received sample block) and can also choose to have the data logged to the CSV file in engineering units (the full scale of each logged microDAQ is stored in the binary file at the start of logging). The following table lists all functions for the controls available in Figure 4.10 above (excluding the microDAQ Clients frame which has been detailed previously).

User Control	Function
Binary file text box and Select button	Sets the filename of the Binary logfile to be converted. Use the Select button to open a Windows dialog to select the BIN file to use.
CSV file text box and Select button	Sets the filename of the CSV logfile to stored theconverted data in. The Select button opens a dialog to select the file to save to.
Show only real timestamps checkbox	A timestamp is associated with a block of samples but the only real timestamp is the one for the last sample in the block. To avoid any confusion in the CSV file the other duplicate timestamps for a block can be removed if this checkbox is ticked.
Scale Binary to Fullscale checkbox	This converts the binary data into engineering units as retrieved for each microDAQ, from the header part of the combined binary logfile.
Convert and Abort buttons	Starts and Aborts the conversion routine.

Table 4.7, User Control Functions for the Control Centre logging setup window.

5. Running microDAQX as a Component.

5.1 Introduction.

microDAQX has been compiled as an ActiveX executable, meaning it can be instanced as an object by another program conforming to the appropriate specifications, whether from a user's bespoke application or another proprietary application such as MS EXCEL. The following discussion centers on the use of microDAQX with EXCEL, how an instance of microDAQX is created and how data might be retrieved from an attached microDAQ unit.

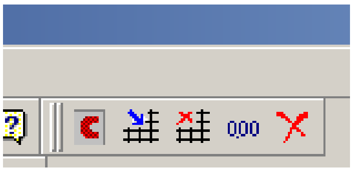
The use of microDAQX as a component requires a degree of programming experience, and familiarity with the use of the Visual Basic (VB) environment of EXCEL or similar. Note the microDAQ Control Centre cannot be instanced from another program.

5.2 Running microDAQ from Excel using the supplied template

5.2.1 Installation

On the CD supplied with the microDAQ is an Excel template (microDAQXtoolbar.xlt). This has been configured to run the microDAQX software from within Excel. Please note that currently this template only works for version of MS Excel from 97 – 2003.

First close any copy of microDAQX that is running and open the spread sheet (enabling the macros). The microDAQ tools will appear on the menu bar.



These have the following meaning (from left to right)

Run microDAQX
Start Logging
Stop Logging
Re-zero
Exit microDAQX

If the microDAQX software has been run and will connect to the microDAQ (see sections 1 or 2) then click on the microDAQX button and the software will run up and data will appear in the data window.

If the reference to microDAQX cannot be found see section 5.4 (this will be evident, the macro not being found)
If the toolbar has no associated macro, see section 5.3 (this will be evident by a message indicating that microDAQXtoolbar1.xls cannot be found)

If you now click on the start logging tool, data will be transmitted into Excel according to the settings in the logging tab.

5.2.2 Re-associating the microDAQ tools.

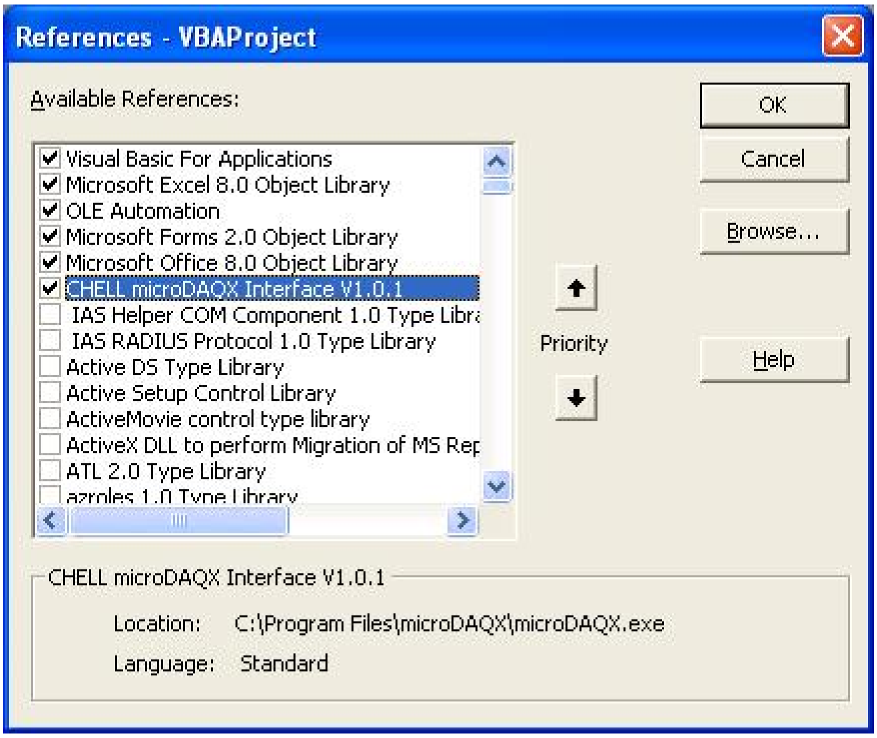
On occasion, Excel can lose the reference between the toolbar and the macro that is being used. To re-enter this data, right click on the tool (starting with the Open microDAQX tool) and select customize. Then right click on the tool again and select assign macro. Then select the appropriate macro and click on OK. Then close customize.
The macro for the first tool is CreateMicroDAQX.

This is repeated for all the tools assigning them to the following macros:

Tool	Macro
Run microDAQX	CreateMicroDAQX
Start Logging	Startlogging
Stop Logging	Stoplogging
Re-zero	Rezero
Exit microDAQX	KillMicroDAQX

5.2.3 Creating a Reference to microDAQX.

With microDAQX correctly installed on a computer, it is still necessary to create a reference to the application from the Visual Basic environment before correct references to it can be made from program code. This is achieved from the Tools/References menu, which should bring up a form similar to that shown in figure 5.1. The reference to 'CHELL microDAQX Interface' should be checked - if not, browse the list box to find the reference, check it and close the dialogue with 'OK'. References to microDAQX may now be made in the code.



References - VBAProject

Available References:

☑ Visual Basic For Applications
☑ Microsoft Excel 8.0 Object Library
☑ OLE Automation
☑ Microsoft Forms 2.0 Object Library
☑ Microsoft Office 8.0 Object Library
☑ CHELL microDAQX Interface V1.0.1
☐ IAS Helper COM Component 1.0 Type Libra
☐ IAS RADIUS Protocol 1.0 Type Library
☐ Active DS Type Library
☐ Active Setup Control Library
☐ ActiveMovie control type library
☐ ActiveX DLL to perform Migration of MS Rep
☐ ATL 2.0 Type Library
☐ azroles 1.0 Type Library

CHELL microDAQX Interface V1.0.1

Location: C:\Program Files\microDAQX\microDAQX.exe
Language: Standard

Figure 5.1, The Tools/References Dialogue.

5.2.4 Procedure for adding the reference to Excel :

[1] With the microDAQXtoolbar.xlt open go to ‘tools’, ‘macros’, ‘Visual Basic editor’.

[2] Then select ‘tools’ references.

[3] If there is a reference listed as missing (or not there at all) then locate the correct reference (in this case microDAQX Interface V1.0.1) and select OK. Then return to the spread sheet and save the spread sheet.

5.3 Using an Instance of microDAQX.

With microDAQX referenced, it is a straightforward matter to create an instance of it from an EXCEL subprogram, for example the following code declares an instance of microDAQX as the object variable 'microDAQ1'.

```
Public WithEvents microDAQ1 As microDAQX
```

The WithEvents keyword is necessary in VB to enable the signaling of events from microDAQX to EXCEL. Once declared, the object should be created with the following statement,

```
Set microDAQ1 = New microDAQX
```

Thereafter, the properties and methods of that instance of microDAQX are available to code written within the EXCEL VB environment. Other instances may be created to other object variable names, allowing communication with multiple microDAQs.

Once declared, the microDAQX events of NewData(), and InfoEvent() are available to a form or class module, declared as

```
Private Sub microDAQ1_NewData()
End Sub

Private Sub microDAQ1_InfoEvent(InfoString As String)
End Sub
```

The event NewData() is raised after the completion of logging (depending on settings within microDAQX), and the data is retrieved by passing a dynamic array of singles to microDAQX's ReadData() subprogram.

```
microDAQ1.ReadData(LogData())
```

The InfoEvent() event is used to pass messages up from microDAQX.

When finished with an instance of microDAQX, it is very important that it is closed down properly with the KillForm method, and its object variable set to Nothing as follows,

```
microDAQ1.KillForm
Set microDAQ1 = Nothing
```

Stray references to objects within EXCEL can lead to undesired effects, such as applications hanging or even system lockup.

5.4 The microDAQX Public Programming Interface.

Table 5.1 lists the available properties and methods of the public interface offered by an instance of microDAQX.

Function	Declaration	Parameter	Return
Close an instance of microDAQX	Public Sub KillForm()	None	None
Load a setup for the instance	Public Function LoadedSetupOK(x$) As Boolean	String of the setup file pathname	True if file found and loaded, false otherwise
Connect both data and command channels	Public Function ConnectAll() As Boolean	None	True if both channels connected to respective hardware
Disconnect both data and command channels	Public Sub DisconnectAll()	None	None
Get the last logged data	Public Sub ReadData(x!())	Dynamic array of singles	Returns the data in the parameter array
Get the last read temperature(s)	Public Sub ReadTemperature(x!())	Dynamic array of singles	Returns the data in the parameter array
Logging function on/off	Public Property Let LoggingOn(x As Boolean)	True for logging on, false for off	None
Implementation of the microDAQ User Command Set			
Data streaming on	Public Sub StreamOn(s%)	Data stream channel	Acknowledge returned via InfoEvent()
Data streaming off	Public Sub StreamOff(s%)	Data stream channel	Acknowledge returned via InfoEvent()
Hardware trigger on	Public Sub HWTriggerOn(s%)	Data stream channel	Acknowledge returned via InfoEvent()
Hardware trigger off	Public Sub HWTriggerOff(s%)	Data stream channel	Acknowledge returned via InfoEvent()
All data streaming off	Public Sub Standby()	None	Acknowledge returned via InfoEvent()
Change the pressure output filter characteristic	Public Sub Filter(x%)	Filter value	Acknowledge returned via InfoEvent()
Change the data streaming rate	Public Sub Rate(s%, x%)	Data stream channel, data rate value	Acknowledge returned via InfoEvent()
Force a soft reset	Public Sub Reset()	None	Acknowledge returned via InfoEvent()
Force a new calibration	Public Sub RebuildCal()	None	Acknowledge returned via InfoEvent()
Switch deranging function	Public Sub Derange(x%)	Deranging on/off	Acknowledge returned via InfoEvent()
Request a rezero	Public Sub ReZero()	None	Acknowledge returned via InfoEvent()
Request status information	Public Sub GetStatus(Level%)	Level of status required	Acknowledge returned via InfoEvent()
Set stream channels	Public Sub Channels(s%, x%)	Set number of active ports for a data channel	Acknowledge returned via InfoEvent()
Set maximum channels	Public Sub MaxChannels(x%)	Set number of active ports for microDAQ	Acknowledge returned via InfoEvent()
Request a span operation	Public Sub Span()	None	Acknowledge returned via InfoEvent()
Request reset of linear calibration	Public Sub RstLinear()	None	Acknowledge returned via InfoEvent()

Table 5.1, The microDAQX Public Programming Interface.