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

Tel: +44 1692 500555

DAQ Systems

Covering:
microDAQ3,
flightDAQ2
flightDAQ3,
CANdaq5,
flightDAQ-TL,
2432T,
2432, 2416,
microDAQ-PCU

USER PROGRAMMING GUIDE

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

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

900204-3.7

Please read this manual carefully before using the instrument.

Use of this equipment in a manner not specified in this manual may impair the user's protection.

Chell Document No. 900204 : Issue 3.7
ECO: 6532 Date: 4th June 2025

Chell's policy of continuously updating and improving products means that this manual describes the current software release. Whilst every effort is made to document current and legacy products and maintain backward compatibility there may be operational differences or deficiencies between the documentation and the supplied product.

1. Introduction. 1
2. User Command Protocol. 2
2.1. Command Packet. 2
2.2. Data Packet. 2
2.3. Acknowledgement. 2
2.4. User Commands. 2
3. Status Data Format. 3
3.1. CANdaq5 (microDAQ2) 3
3.2. microDAQ3 and flightDAQ2 4
3.3. microDAQ-PCU 5
3.4. flightDAQ3 & 2432 scanner 7
3.5. flightDAQ-TL, 8
4. Communication Channels. 9
4.1. TCP 9
4.1.1 Overview. 9
4.1.2 Connection. 9
4.1.3 TCP Protocol. 9
4.1.4 TCP Data Rate. 13
4.1.5 Control Via TCP. 14
4.2. UDP 15
4.2.1 Overview 15
4.2.2 Connection 15
4.2.3 Chell UDP Protocol 15
4.2.4 UDP Data Rate. 18
4.2.5 Control Via UDP. 18
4.3. Timestamping 19
4.4. IENA specification 19
4.5. CAN 21
4.5.1 Overview 21
4.5.2 CAN Baudrate 21
4.5.3 CAN Protocol 22
4.5.4 CAN Data Rate 24
4.5.5 Control Via CAN 24
4.6. Internal RAM – CANdaq5 only 25
4.6.1 Overview 25
4.6.2 Internal RAM Protocol 25
4.6.3 Internal RAM Data Rate 25
4.6.4 Internal RAM Dump 25
5. Valve Control user commands (not flightDAQ-TL) 27
6. microDAQ-PCU Specific user commands 28
6.1. Do Function { / 123 28
6.2. Function Length x / 120 29
6.3. CRG y / 121 29
6.4. Data Return z / 122 29
6.5. Leak Test 32
6.6. Operation Sequence 32
7. Annex A – Span Hi Values 33
8. Annex B – User Command Table 33

1. Introduction.

This manual covers the native protocol of the scanner products listed in the table below. The communication protocol is common across all ranges although not all options are available for all products.

Where the manual specifically refers to the CANdaq5 product this applies also to all the microDAQ2 product.

From power up, the products read its non volatile setup information and calibration, and after applying these settings is then 'up and running', typically flashing a blue LED indicating it is reading the transducers and delivering calibrated data. Although for many applications this is enough, more demanding applications may need to control the data delivery, request data rezero operations and more.

Commands may be sent over any of the unit's communication channels, whether the current active data channel or not. A simple block parity check adds security to the command protocol, and a correctly received command may be acknowledged if required.

The following sets out the essentials of the command protocol, the data packet format for all channels in addition to the message identifier arrangement for the CAN channel.

This document version supports the following firmware versions of each supported product:

Supported device	Firmware version (from)
Current Product	
microDAQ3	V1.0.0
microDAQ-PCU	V1.0.0
2432	V2.0.0
2432T	V1.0.0
flightDAQ3	V2.0.0
flightDAQ2	V3.0.0
flightDAQ-TL	V1.1.0
Legacy Support Product	
CANdaq5/microDAQ2	V2.0.0
microQDVP2	V1.0.0

Older or unlisted Product
Chell looks to standardise the protocol where possible so this document may offer guidance however where a device is not listed or is of a previous firmware version some user commands and/or parameters may be invalid or function differently or adversely.

2. User Command Protocol.

2.1. Command Packet.

The command protocol is based around a simple delimited control packet, allowing easy identification of command start and end. The packet includes a block parity byte increasing the robustness of transmission; the packet format is shown in Figure 1.

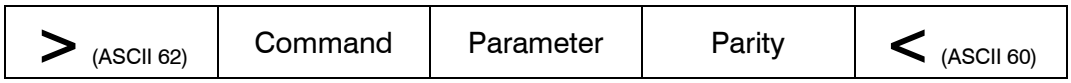

Figure 1 Command Packet Format

The command byte values are defined in the following section, and may or may not require a parameter byte, for example data rate. The parity byte is an even block parity of all bytes (other than itself), including the delimiters. Calculation for the parity byte is therefore modulo 2 arithmetic (exclusive-OR) of all other bytes.

For a command not requiring a parameter, an arbitrary dummy byte should be used. This can be any value (but '0' is recommended) as the number is not processed other than in determining the parity of the command packet.

2.2. Data Packet.

Some commands require passing of a data packet. This data packet follows the command packet only when the command packet supports such function and only where the command instructs that a data packet is to follow. The data packet contains a 4 byte IEEE754 float (note the data shares the endianness of the protocol, LE or BE) followed by parity byte.

Figure 2 Data Packet Format

e.g. a value of 2.5 will calculate as a data packet of '00 00 32 64 96'

2.3. Acknowledgement.

A command packet is positively acknowledged if it is correctly formatted and the parity byte is correct. A positive acknowledgement is denoted by the transmission of ASCII 42 ('*'), a negative acknowledge indicating an invalid command by ASCII 33 ('!'). As of this document issue 3.2 an acknowledgement may be returned of ASCII 63 ("?") for a recognised command that cannot be actioned. Recognised and actionable command values are then acted upon, though unrecognised values are discarded. Note that UDP and TCP use two acknowledgements, such as **,?? or !! CAN using only a single acknowledgement of *,? or !

2.4. User Commands.

The available user command set is summarised in the following table. Where an orange tick is given ensure the text is consulted to understand any limitations.

Note: for CANdaq5/microDAQ3/flightDAQ2; All settings changes are stored in volatile memory unless otherwise stated. This means the 'Burn to EEPROM' command will need to be sent to retain settings after a power cycle or reboot of the device.

3. Status Data Format.

3.1. CANdaq5 (microDAQ2)

Status data is not currently supported over the CAN channel, however for TCP connections, status data may be requested from the CANdaq5 as three forms – 'short', 'full' and 'with temp'. The short form returns 4 bytes, the 16 bit status word delimited by the ASCII characters ">" and "<", the less significant byte is returned before the more significant. The bit assignment is shown in Figure 3.

15						I-daq conn.	Hardware Trigger Active	Derange acvtive	DTC conn.	CAN Active	TCP Active	*Reserved*	Cal. table	Span	Rezero

Figure 3 Status word bit assignment.

Requesting the status 'with temp.' returns the short status 4 bytes, followed by the temperature data. For a non DTC scanner, a single value (ASCII unsigned 14 bit) representing the read temperature voltage is appended to the status bytes. For a DTC application, all active channels are returned in ascending order as comma delimited ASCII engineering units (ie degrees C).

The 'full' status data contains the above, followed by fields for the setup options of the CANdaq5. Each field is comma delimited, and its function is indicated in plain text between square parentheses. On/off is indicated by 1/0. An example 'full' status string is shown in Figure 4 for a CANdaq5 with a non DTC scanner. Note that the figure for full scale includes the derange constant (ie scanner full scale x derange constant) if the derange option has been selected in setup.

*>Mó<8198,[Full scale] 15.00000000,[Active channels] 32,[DTC active] 0,[CAN channels] 32,[TCP channels] 32,[CAN rate] OFF,[TCP rate] OFF,[CAN protocol] 16 LE,[TCP protocol] 16 LE,[Press. input impulse] 1,[Temp. input impulse] 0,[Press. input power] 3,[Temp. input power] 0,[Press. output power] 0,[Reset on delivery] 0,[Temp. compensation] 0,[Period] 10m,[IP] 0.0.0.0,[Mask] 0.0.0.0,[Gateway] 0.0.0.0,[CAN timing] (BRP) 5 (TSEG1) 2 (TSEG2) 0 (SJW) 1,[CAN message] 00n,[Rezero order] 4,

Figure 4 Full Status Information

3.2. microDAQ3 and flightDAQ2

Status data is not currently supported over the CAN channel, however for TCP and UDP connections, status data may be requested from the microDAQ3 as three main forms – ‘short’, ‘full’ and ‘with temp’. The short form returns 4 bytes, the 16 bit status word delimited by the ASCII characters ">" and "<", the less significant byte is returned before the more significant. The bit assignment is shown in Figure 5.

bit 15 8

Valve CAL	Valve RUN	Purge ON*	PTP sync. active		HW Trigger Error detected		Hardware Trigger active
		CAN stream active	TCP stream active				Rezero in progress

Figure 5 Status word bit assignment

7 bit 0

Where both bit 14 & 15 are set then the valve position is indeterminate, meaning the valve is not in a valid position.

Bit 13 “Purge ON” applies to flightDAQ2 only.

Requesting the status 'with temp.' returns the short status 4 bytes, followed by the temperature data. Temperatures for all active channels are returned in ascending order as comma delimited ASCII engineering units (ie degrees C).

The 'full' status data contains the above, followed by fields for the setup options of the microDAQ3. Each field is comma delimited, and its function is indicated in plain text between square parentheses. On/off is indicated by 1/0. An example 'full' status string from a microDAQ3 is shown in Figure 6.

```
>@.<,19.88,20.01,20.07,20.23,20.25,20.35,20.37,20.28,20.19,20.26,20.33,20.37,20
.33,20.32,20.18,20.16,[Serial] 1810801,[Full scale] 2.50000000,[Active channels]
64,[CAN channels] 64,[TCP channels] 64,[CAN rate] OFF,[TCP rate] OFF,[CAN
message] Multiple,[CAN protocol] 16 LE,[TCP protocol] 16 LE,[Press. input
impulse] 0,[Press. input power] 4,[IP] 192.168.3.190,[Mask] 255.255.0.0,[Gateway]
0.0.0.0,[CAN timing] (BRP) 4 (TSEG1) 11 (TSEG2) 4 (SJW) 3,[CAN message]
100,[IENA key] 0x3101,[IENA end word] 0xDEAD,[Ethernet power] Auto,[CAN
power] Auto,[Press. units] psi,[Press. type] Differential,[PTP sync] Off,[Stream
timestamp] None,[Time format] UTC,
```

Figure 6 Example full status information

3.3. microDAQ-PCU

Status data is available over TCP and UDP connections, status may be requested in two main forms; ‘short’ and ‘full’.
The short form returns 5 bytes, the 24 bit status word delimited by the ASCII characters ">" and "<", the less significant byte is returned before the more significant. The bit assignment is shown below in Figure 7.

bit 23 16

TimeLeft(bit7)	TimeLeft(bit6)	TimeLeft(bit5)	TimeLeft(bit4)	TimeLeft(bit3)	TimeLeft(bit2)	TimeLeft(bit1)	TimeLeft(bit0)
Test Devices Connected	Std Connected		Cal Pt (bit4)	Cal Pt (bit3)	Cal Pt (bit2)	Cal Pt (bit1)	Cal Pt (bit0)
Leak	Misc Busy	Fault	Exit Code	5Pt Cal	Span	Zero	Idle

Figure 7 Status word bit assignment

7 bit 0

Bit explanation:
Bit0, Idle. Set true if the microDAQ-PCU is idle waiting for instruction. Whilst performing a task this is set to false.
Bit1, Zero. Set true if a zero routine in in progress. Check or Adjust reports the same.
Bit2, Span. Set true if a span routine in in progress. Check or Adjust reports the same.
Bit3, 5Pt Cal. Set true if the 5 point calibration routine is in progress.
Bit4, Exit Code. Set true for a clean exit indicating the last request completed without fault.
Bit5, Fault. Set true where a command has failed or completed out of tolerance. **Note that when this fault bit is set the other bits indicate the fault, not their declared function. i.e. Fault and Std Connected together indicates a problem with the Pressure Standard.**
Bit6, Busy. Set true where the device is busy or temporarily unable to accept a request.
Bit 8-12, Reported Status of the current command (see below).
Bit14, Std Connected. Set true where the pressure calibration standard is connected and responding with pressure readings.
Bit15, Test Devices Connected. Set true when all Test devices are connected. Note if a CRG contains multiple devices and one device is failing connection it will keep this bit cleared.
Bit 16-23, Time remaining for an active command or segment thereof. 0-255 seconds, a value of ‘0’ corresponds to a completed command or segment. Any command or segment requiring longer duration or of no fixed end will report 255. Note that some commands, such as 5 point Cal, are multi-segmented. The end time is not reported, only the time remaining on the segment.

Bit 8-12 Status of the current command
00: idle
01: Opening comms to devices
02: Shuttle to Cal operation
03: Zero check
04: Zero adjust
05: Span adjust
06: Negative Span
07: Negative 50%
08: Zero check
09: Positive 50%
0A: Positive Span (check)
0B: Saving data
0C: Shuttle to Run operation
0D: Closing comms to devices
0E: Leak Test

Full status response

The full status data contains the above followed by details of the CRG group and the devices contained within it, including ‘model’ ‘ip address’ & ‘range’

[CRG] 2, [Calibrator] Pace5000, [Full scale] 2.50000000, [Press. units] kPa, [Press. type] Differential, [Module slot] 1, [IP] 192.168.3.100,[Mask] 255.255.0.0,[Gateway] 0.0.0.0, [Connected] Yes, [NumDevices] 3, [Device1] microDAQ3 [Serial1] 2412301,[Full scale1] 1.00000000, [Press. units1] kPa,[Press. type1] Differential, [Active channels1] 64, [IP1] 192.168.3.190,[Mask1] 255.255.0.0,[Gateway1] 0.0.0.0, [Connected1] Yes, [Device2] microDAQ3 [Serial2] 2412302,[Full scale2] 1.00000000, [Press. units2] kPa,[Press. type2] Differential, [Active channels2] 64, [IP2] 192.168.3.191,[Mask2] 255.255.0.0,[Gateway2] 0.0.0.0, [Connected2] Yes, [Device3] microDAQ3 [Serial3] 2412303,[Full scale3] 1.00000000, [Press. units3] kPa,[Press. type3] Differential, [Active channels3] 64, [IP3] 192.168.3.193,[Mask3] 255.255.0.0,[Gateway3] 0.0.0.0, [Connected3] Yes,

Figure 8 Example full status information

3.4. flightDAQ3 & 2432 scanner

Status data may be requested from the flightDAQ3 & 2432 as three main forms – ‘short’, ‘full’ and ‘with temp’. The short form returns 4 bytes, the 16 bit status word delimited by the ASCII characters ">" and "<", the less significant byte is returned before the more significant. The bit assignment is shown in Figure 9 below:

bit 15 8

Valve CAL	Valve RUN	Purge ON	PTP sync. active		HW Trigger Error detected		Hardware Trigger active
			TCP stream active			Span in progress	Rezero in progress

Figure 9 Status word bit assignment

7 bit 0

Where both bit 14 & 15 are set then the valve position is indeterminate, meaning the valve is not in a valid position.

Requesting the status 'with temp.' returns the short status 4 bytes, followed by the temperature data. Temperatures for all active channels are returned in ascending order as comma delimited ASCII engineering units (ie degrees C).

The 'full' status data contains the above, followed by fields for the setup options of the microDAQ3. Each field is comma delimited, and its function is indicated in plain text between square parentheses. On/off is indicated by 1/0. An example 'full' status string from a microDAQ3 is shown in Figure 6.

```
>@.<,19.88,20.01,20.07,20.23,20.25,20.35,20.37,20.28,20.19,20.26,20.33,20.37,20
.33,20.32,20.18,20.16,[Serial] 1810801,[Full scale] 2.50000000,[Active channels]
64,[TCP rate] OFF,[TCP protocol] 16 LE,[Press. input impulse] 0,[Press. input
power] 4,[IP] 192.168.3.190,[Mask] 255.255.0.0,[ Press. type] Differential,[PTP
sync] Off,[Stream timestamp] None,[Time format] UTC,
```

Figure 10 Example full status information

3.5. flightDAQ-TL,

Status data is currently not defined . .

4. Communication Channels.

4.1. TCP

4.1.1 Overview.

The devices' TCP channel affords it the ability to stream real time data at high speed over standard Ethernet connections.

4.1.2 Connection.

When using the TCP/IP channel, DAQ devices are programmed to listen on local port number 101. The device will respond to a connection request, connect and immediately start streaming data if it has been setup to use the TCP channel. Note that the microDAQ3, CANdaq & FlightDAQ2 devices only support one TCP connection at a time.

Setup requires the device to be given an IP address, and the network's subnet mask. It is possible either to run the device over a local Ethernet connection, or directly from a PC's network card where a crossover cable is typically used. A dual network card (or 2 cards) may be used, however care should be taken about network routing, as similar subnets for two network connections on a single Windows ® machine can cause the device not to be seen on the second card.
Note the CANdaq5 will not respond to a 'ping', however later devices do support this. It is also possible to 'hail' the devices using UDP broadcast address with the command 'cistat'.

4.1.3 TCP Protocol.

With TCP channel selected, calibrated data is streamed continuously over the TCP interface. Depending on the device variant, TCP supports a range of different data formats as shown in Figure 13 The binary data formats scale the data to 0 to 65535 bits which represent either -/+ full scale respectively (IEEE 754 representation for 32 bit). Binary protocols require less communications bandwidth as well as less processor overhead within the device; it is recommended that engineering unit conversions be applied at the client. Note that the microDAQ3 and flightDAQ pressure variants (not -TL) can be configured to stream absolute or differential pressure channel data. When configured for absolute pressure, the binary representation is configurable, but by default, follows the following for each supported scanner range:

Table 1 - Typical absolute scaling

Range (psid)	0 bits	65535 bits	Conversion 'M' val	Conversion 'C' val
FS <= 2.5 psi	15000 Pa (2.17557psi)	115000 Pa (16.68psi)	4518.539969	2.1755661
5 & 8 psi (FS > 2.5-8 psi)	13000Pa (1.885psia)	160000Pa (23.206psia)	3073.842279	1.8854906
8 psi (FS > 8-15 psi)	15000Pa (2.17557psi)	207000Pa (30.0228psia)	2353.374583	2.1755661
30 & 45 psi (FS >15 psi)	0Pa (0 psia)	400000Pa (58.0151psia)	1129.619798	0
100 & 150 psi	0 Pa (2.17psi)	1140000 Pa (165.343psi)	396.3578239	0

When converting from binary to engineering units, the following equation can be used, substituting the 'M' & 'C' values from the above table, as appropriate for the range in question:

Engineering units (psi) = (Binary value / M) + C

In the case of the TCP channel, the engineering unit protocol should be avoided if possible, due to the significant processing overhead. The 16 bit little ended protocol is most efficient from the device viewpoint, the data being calibrated and spanned to 0 to 65535 (zero at 32767). Scaling to floating point full scale is better achieved within a PC client application where there is more processing power available.

16bit binary (16LE and 16BE)

microDAQ3, flightDAQ2, flightDAQ3, 2432 & CANdaq5

This is a binary output representing a scaling of the full-scale ((0-65535/65535)x100%). It is possible to configure for big endian or little endian. The content of the stream has selectable content, see the following diagram:

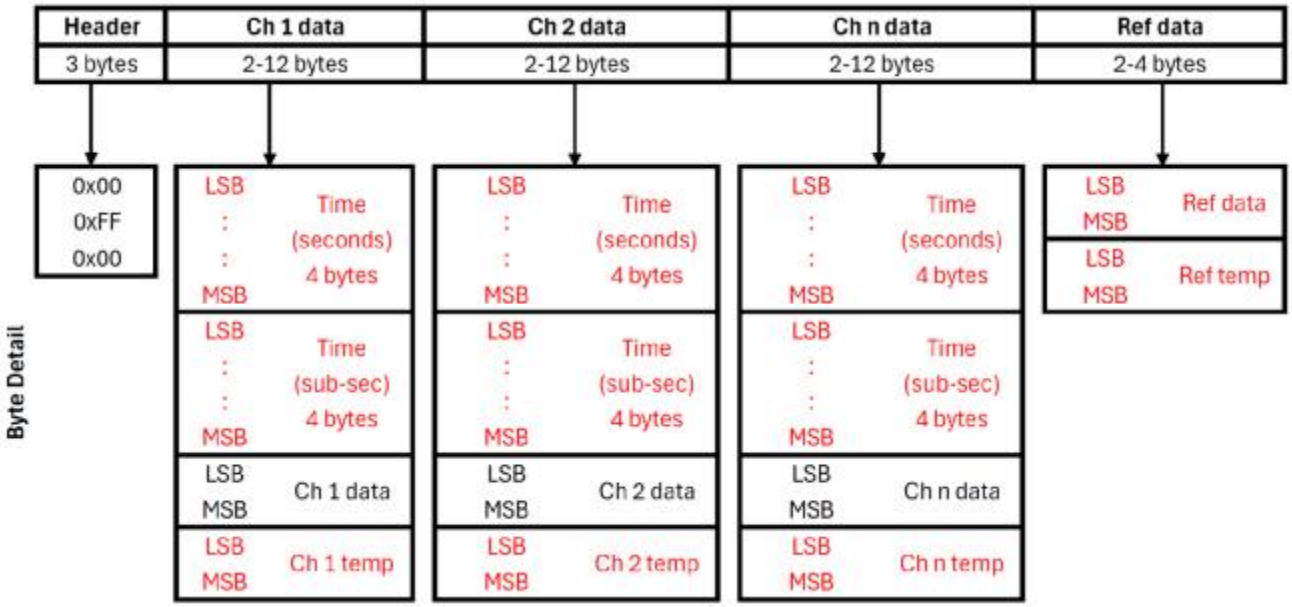

Figure 11 Data Packet Protocol, 16bit TCP

Key:
Configurable Options: shown in RED, when unconfigured the field is omitted (reducing length)
Endianess: The example is shown as 16bit LE (little endian). 16bit BE (big endian) would transpose the LSB and MSB byte order.
Channel Count: This is determined by the device capability, configuration or the 'Channels' (H/32) command.
Timestamp: Where a timestamp of 'Start of Cycle' is selected only Ch1 is timestamped.
Time: This is comprised of two unsigned 32bit numbers, one number for whole seconds and another number for sub-seconds. Typically, (for 16bit data) this is in microseconds, some devices may offer configuration to nanoseconds.
Ref data: This is the Absolute value of the pressure reference, this is typically used to clarify the absolute value of differential data. Where configured this data is a 16bit representation of absolute pressure between limits of 15000Pa to 115000Pa (0-65535 respectively). Some devices may offer this range as a configuration.

32bit IEEE754 float (32LE and 32BE)

flightDAQ-TL

Data is transmitted as a 32 bit IEEE754 float, 4 bytes per reading. It is possible to configure for big endian or little endian. The content of the stream has selectable content, see the following diagram:

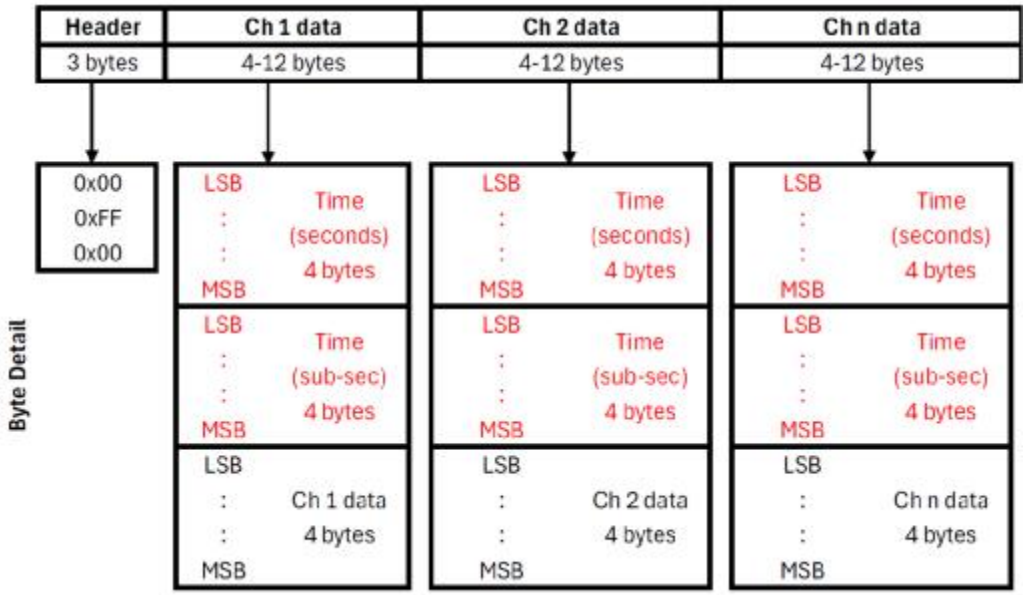

Figure 12 Data Packet Protocol, 32bit TCP

Key:
Configurable Options: shown in RED, when unconfigured the field is omitted (reducing length)
Endianess: The example is shown as 32bit LE (little endian). 32bit BE (big endian) would transpose the LSB and MSB byte order.
Timestamp: Where a timestamp of 'Start of Cycle' is selected only Ch1 is timestamped.
Time: This is comprised of two unsigned 32bit numbers (not float), one number for whole seconds and another number for sub-seconds. Typically, (for 32bit data) this is in nanoseconds, some devices may offer configuration to microseconds.

ASCII comma separated (legacy support; microDAQ2, CANdaq5)

Single character header followed by all channels (1 to ‘N’ where ‘N’ = 16 or 32 depending on setting), 5 decimal places, comma delimited. (flightDAQ-TL returns 6 decimal places). Note that flightDAQ2 includes the Abs sensor data in the same format, comma separated, located as the final entry of the string.

Hdr		Channel 1 Data		Channel 2 Data		….		Channel ‘N’ Data
*	,	#.######	,	#.######	,		,	#.######

Figure 13 Data Packet Protocol, TCP Channel.

4.1.4 TCP Data Rate.

The data delivery rate is selected from the webserver, and as an example, the following values are available for the microDAQ3 (Hz) – 1, 5, 10, 20, 25, 50, 100, 150, 200, 400. Although the system endeavours to deliver the rate with maximum accuracy, ultimate responsibility for data timing lies with the user's host system. The actual data rates can vary between products, for example the 2416 offers up to 1000Hz Response setting.

Where the product uses a Sensor Response setting this does limit the published data rate. Applying a lower sensor response setting gives more oversampling to the sensors producing a more accurate reading at the cost of output speed, for this reason it is recommended that the sensor response is set to the lowest frequency that exceeds your desired maximum output frequency. For a microDAQ3 example, to achieve the maximum data rate of 400Hz the sensor response should be set to '400Hz', however if only 150Hz output is required, the sensor response should be set to '200Hz'. See Figure 14 Sensor response vs maximum data rate. The only notable exception being where timestamp accuracy takes precedence over measurement accuracy, in which case selecting the highest available response will yield the best accuracy timestamp. shows the available sensor responses and the subsequent maximum data rate allowed.

Sensor response	Maximum available output data rate
1000 Hz	1000 Hz
400/500 Hz*	400/500 Hz*
200 Hz	200 Hz
100 Hz	100 Hz
50 Hz	50 Hz

Figure 14 Sensor response vs maximum data rate

*microDAQ3 will list a maximum of 400Hz Response.

When using CANdaq5, note that since the scanners are addressed at a fixed channel rate (20kHz if configured for Gen1 scanners, 50kHz if configured for Gen2 scanners), requesting a data delivery of more than 20k/No. Channels (Gen1) or 50k/No. Channels (Gen2) - i.e. 312Hz for a 64 channel Gen 1 scanner) - wastes resources and in some cases can actually cause the device to hang and require a power cycle.

The data rate over TCP is vulnerable to low level operating system considerations, also note that the TCP channel of the device is subject to buffering both within the unit itself (the size of the buffer depending on number of channels and data rate), and within Windows ® itself. At low data rates, it is possible to receive single complete data packets, however as the data rate increases Windows ® is more likely to deliver chunks of data 2k to 4k bytes in length, with arbitrary boundaries with respect to the device's data packets. User software should take this into account to avoid losing data.

Please note that if streaming at high data rates it is essential that a good network infrastructure is used. It is recommended that any streaming is performed over a 'private' network (i.e. disconnected from any corporate network structure) to reduce the number of packets flying around the network. It is also highly recommended that a high speed managed network switch with a large store and forward buffer is used between the device and client PC, particularly if several devices are being used together to acquire data.

4.1.5 Control Via TCP.

A positive acknowledgement is returned as '**' and a negative acknowledgement as '!!'. With newer firmware releases the acknowledgement of '??' is being phased in, indicating that a correctly formatted command was received but it cannot be acted upon. Older firmware will issue a positive acknowledgement if the command is understood, regardless of if it is possible to action.

The argument regarding loss of acknowledgements in the data stream holds good for the TCP channel, and it is recommended that a 'Stream Off' or 'Standby' is sent before any other command.

4.2. UDP

4.2.1 Overview

As with TCP the device's UDP channel affords it the ability to stream real time data at high speed over standard Ethernet connections.
All devices, apart from the flightDAQ-TL, support two UDP data stream formats, the Chell data stream format and an IENA specification output, which can be setup from the web server.

4.2.2 Connection

When using the UDP/IP channel, the device is programmed to listen on its local port. It will respond to a connection request, connect and immediately start streaming data if it has been setup to use the UDP channel with remote IP address and port configured in the settings.

Setup requires the device to be given an IP address, and the network's subnet mask, these will be the same as for the TCP, however in addition to those UDP also needs a remote IP address and remote port number. It is possible either to run the device over a local Ethernet connection, or directly from a PC's network card (a crossover cable may be required). A dual network card (or 2 cards) may be used, however care should be taken about network routing, as similar subnets for two network connections on a single Windows ® machine can cause the device not to be seen on the second card.

4.2.3 Chell UDP Protocol

With UDP channel selected calibrated pressure data are streamed continuously over the UDP interface. Depending on the device variant, UDP supports a range of different data formats as shown in Figure 17 (same as with the TCP protocol above). The binary data formats scale the data to 0 to 65535 representing -/+ full scale respectively. Binary protocols require less communications bandwidth as well as less processor overhead within the device; it is recommended that engineering unit conversions be applied at the client.

Note that as with TCP protocol variants (not -TL) can be configured to stream absolute or differential pressure channel data. The conversion table & equation from the TCP section are shown here again for completeness:

Range (psid)	0 bits	65535 bits	Conversion 'M' val	Conversion 'C' val
FS <= 2.5 psi	15000 Pa (2.17557psi)	115000 Pa (16.68psi)	4518.539969	2.1755661
5 & 8 psi (FS > 2.5-8 psi)	13000Pa (1.885psia)	160000Pa (23.206psia)	3073.842279	1.8854906
8 psi (FS > 8-15 psi)	15000Pa (2.17557psi)	207000Pa (30.0228psia)	2353.374583	2.1755661
30 & 45 psi (FS >15 psi)	0Pa (0 psia)	400000Pa (58.0151psia)	1129.619798	0
100 & 150 psi	0 Pa (2.17psi)	1140000 Pa (165.343psi)	396.3578239	0

When converting from binary to engineering units, the following equation can be used, substituting the 'M' & 'C' values from the above table, as appropriate for the range in question:

Engineering units (psi) = (Binary value / M) + C

16bit binary (16LE and 16BE)

microDAQ3, flightDAQ2, flightDAQ3, 2432 & CANdaq5

This is a binary output representing a scaling of the full-scale ((0-65535/65535)x100%). It is possible to configure for big endian or little endian. The content of the stream has selectable content, see the following diagram:

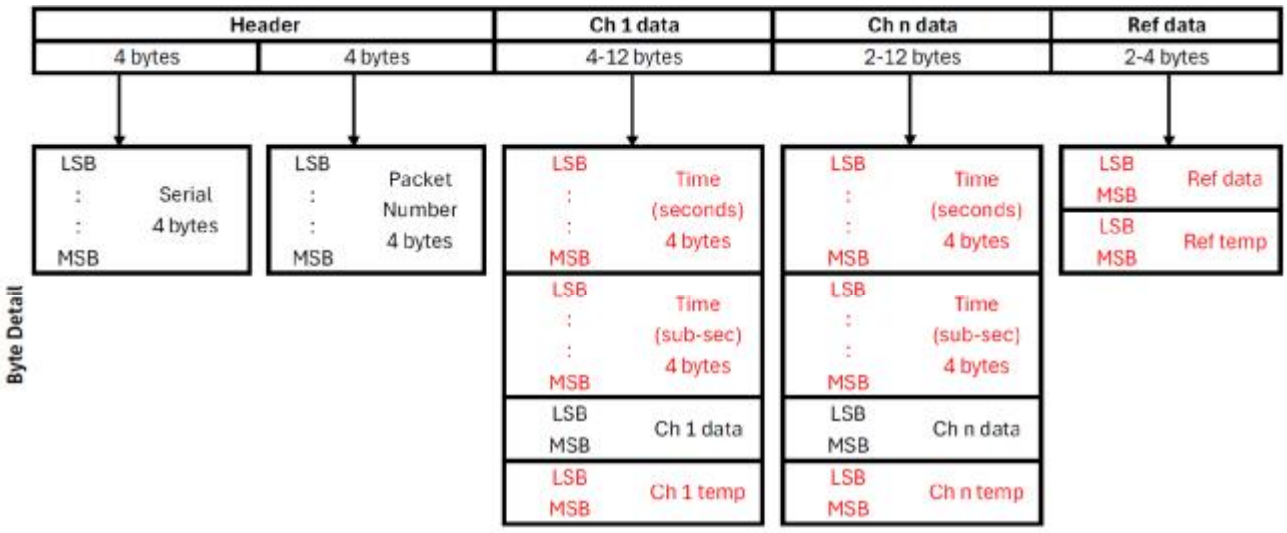

Figure 15 Data Packet Protocol, 16bit UDP

Key:
Configurable Options: shown in RED, when unconfigured the field is omitted (reducing length)
Endianess: The example is shown as 16bit LE (little endian). 16bit BE (big endian) would transpose the LSB and MSB byte order.
Header Serial Number: An unsigned 32bit number representing the device serial number (e.g. 2412301)
Header Packet Number: An unsigned 32bit number incrementing from 1 in steps of 1. This number can wrap when the maximum value is reached.
Channel Count: This is determined by the device capability, configuration or the 'Channels' (H/32) command.
Timestamp: Where a timestamp of 'Start of Cycle' is selected only Ch1 is timestamped.
Time: This is comprised of two unsigned 32bit numbers, one number for whole seconds and another number for sub-seconds. Typically, (for 16bit data) this is in microseconds, some devices may offer configuration to nanoseconds.
Ref data: This is the Absolute value of the pressure reference, this is typically used to clarify the absolute value of differential data. Where configured this data is a 16bit representation of absolute pressure between limits of 15000Pa to 115000Pa (0-65535 respectively). Some devices may offer this range as a configuration.

32bit IEEE754 float (32LE and 32BE)

flightDAQ-TL

Data is transmitted as a 32 bit IEEE754 float, 4 bytes per reading. It is possible to configure for big endian or little endian. The content of the stream has selectable content, see the following diagram:

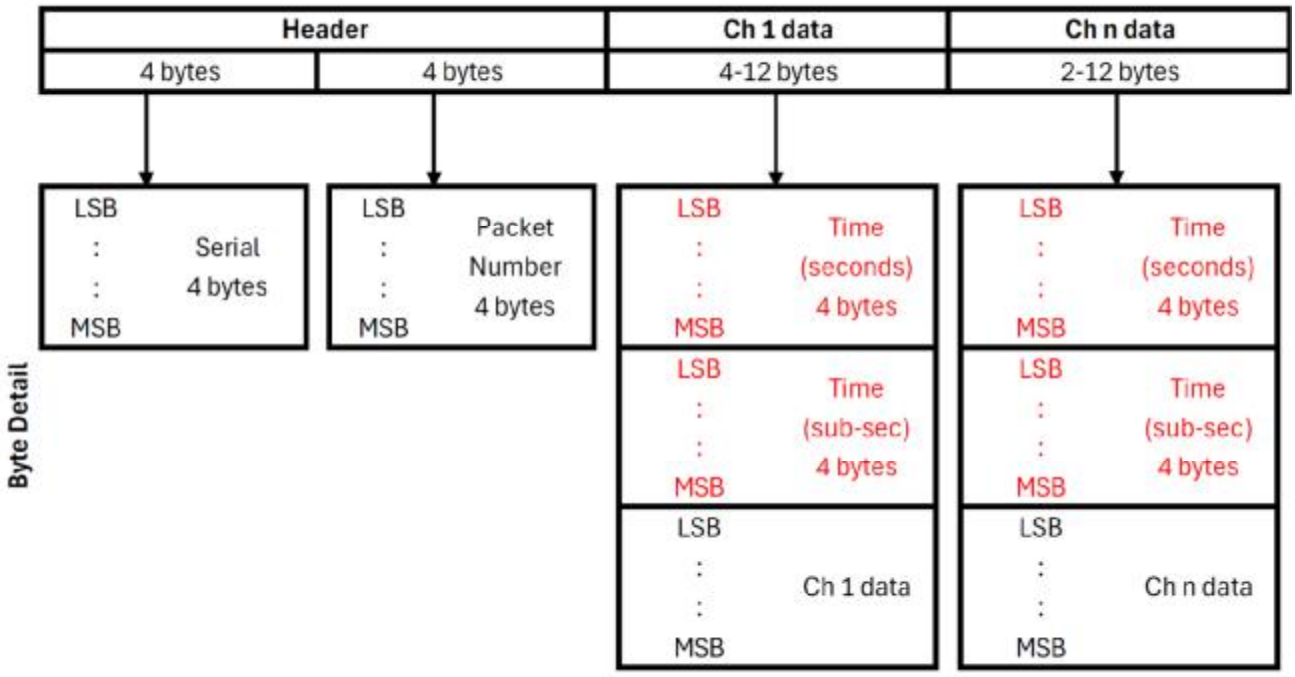

Figure 16 Data Packet Protocol, 32bit UDP

Key:
Configurable Options: shown in RED, when unconfigured the field is omitted (reducing length)
Endianess: The example is shown as 32bit LE (little endian). 32bit BE (big endian) would transpose the LSB and MSB byte order.
Header Serial Number: An unsigned 32bit number representing the device serial number (e.g. 2412301)
Header Packet Number: An unsigned 32bit number incrementing from 1 in steps of 1. This number can wrap when the maximum value is reached.
Channel Count: This is determined by the device capability, configuration or the ‘Channels’ (H/32) command
Timestamp: Where a timestamp of 'Start of Cycle' is selected only Ch1 is timestamped.
Time: This is comprised of two unsigned 32bit numbers (not float), one number for whole seconds and another number for sub-seconds. Typically, (for 32bit data) this is in nanoseconds, some devices may offer configuration to microseconds.

ASCII comma separated (legacy support; microDAQ2 & CANdaq5 only)

Hdr		Channel 1 Data		Channel 2 Data		....		Channel 'N' Data
*	,	#.######	,	#.######	,		,	#.######

Figure 17 Data Packet Protocol, UDP Channel.

Single character header followed by all channels (1 to 'N' where 'N' = 16 or 32 depending on setting), 5 decimal places, comma delimited. (flightDAQ-TL returns 6 decimal places). Note that flightDAQ2 includes the Abs sensor data in the same format, comma separated, at the end of the string

4.2.4 UDP Data Rate.

The UDP data rate is much the same as the TCP data rate in its configuration and use, so all the TCP data rate information is applicable for UDP.

4.2.5 Control Via UDP.

A positive acknowledgement is returned as '**' and a negative acknowledgement as '!!'. With newer firmware releases the acknowledgement of '??' is being phased in, indicating that a correctly formatted command was received but it cannot be acted upon. Older firmware will issue a positive acknowledgement if the command is understood, regardless of if it is possible to action.

The argument regarding loss of acknowledgements in the data stream holds good for the UDP channel, and it is recommended that a 'Stream Off' or 'Standby' is sent before any other command.

4.3. Timestamping

Standard data streaming over both TCP & UDP can include time stamping information to aid with synchronisation of data packets. The device keeps track of time to microsecond (or nanosecond for OS based devices) resolution since it was powered on but this can be turned into a real time by either synchronising the time with a PC (via the webserver) or for a more accurate time sync, by using an IEEE1588 V2 PTP grandmaster clock to provide a constant PTP time sync message.

In both cases the time stamp is represented as two 32 bit values, the first being the Unix Epoch time (i.e. the number of seconds since 1st January 1970) and the second being the number of microseconds in that second. The OS based products will use nanoseconds rather than microseconds, this includes; flightDAQ3, flightDAQ-TL, 24xx Series, TP-9000 & microDAQ-PCU.

In the device's data streaming, the timestamps can be placed at the start of a channel cycle (ie. before channel 1's data) or in front of every channel. This is configurable from the Timestamping page of the embedded webserver. Note that the byte order of the timestamp is the same as that of the channel data, as configured by the protocol selection (16 bit LE or BE) and timestamps are not added if the Engineering units protocol is selected instead. Adding timestamps (particularly on a per channel basis) can significantly affect the speed of the data streaming – some of the high data rates may not be achievable due to the extra data being transmitted per packet to facilitate the time stamps in the stream.

4.4. IENA specification

All devices (except the flightDAQ-TL) supports the option to output data in the IENA specification data packet format. This specialised format arranges the data in a specific format containing various different information. The data format is only available when using the UDP comms protocol.

Data packets formatted in this way start with a specific header format before any data in the packets.

Key	Size	Time			Status	Seq	Data 0	-	Data 63	Temperature	Scanner Status	End
16 Bits	16 bits	16 bit	16 bit	16 bit	16 bits	16 bits	32 bits	-	32 bits	32 bits	16 bits	16 bits
	Frame size	Time in microseconds (this year)				Rolling 16 bit counter	Byte order 3-2-1-0		Byte order 3-2-1-0	Byte order 3-2-1-0		

Note: The network byte ordering was historically always big endian but now will action any data protocol format setting chosen on the webserver or set via the protocol command, each field in the IENA packet will be sent with the most significant byte first.

Starting with the 16 bit key field, this is a user defined key that will appear at the start of every IENA packet. The default value is 0x3101.

The next 16 bits of the data packet is the size of the packet, this is the total number of bytes in the IENA header and the data blocks.

The next section of data in the data packet is the time, this is 48 bits long and contains the time that has elapsed in microseconds from the 1st of January of the current year. For this to work accurately IEEE 1588 has to have been enabled and the device has to be synchronised with a grandmaster. It is also possible for the user to get the time from a PC on the embedded webserver, this is however less accurate than using a IEEE 1588 synchronised time. This will be in UTC or TAI depending on the users choice.

The next 16 bits is a STATUS word. This word is used to show the status of various parameters in the device. This varies between device variants, as follows:

Bit	**microDAQ**	**flightDAQ2**
1(LSB)	Synced to external time	Synced to external time
2	Synced to PTP master clock	Synced to PTP master clock
3		Device within user defined temperature range
4		Heaters are on
5		Device is over 90 degC

Following this is a rolling 16 bit sequence number that counts how many packets have been sent in this data stream, this is a rolling counter, so if the packet sent is greater than the maximum value this field can hold then it will reset to 0 and start counting again.

After the header is finished the next part of the packet is the main data in channel order, this is 32 bit floating point data, this data can be in big endian or little endian format depending on the users selection. There is one 32 bit value for each channel of data.

The data is followed by a single 32 scanner temperature value.

The scanner status is a 16 bit field of which is used to show various details about the scanner and device. The least significant bit will show if the scanner is in purge mode or not, the second bit will show if the scanner valve is in run or cal mode (cal mode = bit is set, run mode = bit is clear). Further bits are reserved for future use and are set as 0.

The final 16 bits are the end field, This value is a user settable value that signifies the end of an IENA packet. The default value of this field is 0xDEAD.

4.5. CAN

4.5.1 Overview

The CAN channel is somewhat different to the above channels, in that it is not a simple serial communications channel, the data being sent in discrete chunks on specified message identifiers. CAN is not supported on the flightDAQ-TL, flightDAQ3 & 2432 devices.

4.5.2 CAN Baudrate

The devices offers a single 'standard' CAN bus connection running at a selectable baudrate, the user having access to the values used for the microcontroller CAN timing registers to enable customising of sample point and jump width. For the 64 channel variants the values should be calculated based on the microcontroller CAN peripheral clock of 90MHz and for the 16 channel variant 150MHz should be used, users should bear these frequencies in mind when calculating suitable values for BRP, TSEG1, TSEG2 and SJW for their own physical network implementation. Factory defaults for all versions of the microDAQ3 set the CAN baud rate to 1MHz with a sample point of 72%.

Below are tables of example timing configurations for the 64 and 16 channel variants on buses with a 72% sample point.

CAN Bus Rate (kHz)	BRP	TSEG1	TSEG2	SJW
1000	4	11	4	3
500	9	11	4	3
250	19	11	4	3
125	39	11	4	3
100	49	11	4	3
50	99	11	4	3

Table 2, 64 channel (microDAQ3 firmware 1.X.X & CANdaq5) CAN bus timings with a 72% sample point

CAN Bus Rate (kHz)	BRP	TSEG1	TSEG2	SJW
1000	14	5	2	2
500	19	8	4	4
250	23	16	6	6
125	29	26	11	11
100	49	19	8	8
50	59	34	13	13

Table 3, 64 channel microDAQ3 firmware 2.X.X and flightDAQ2 and 16 channel CAN bus timings with a 72% sample point

4.5.3 CAN Protocol

Two separate protocols are available from the setup – either multiple or single message. For the former, the device is allocated a fixed message for each group of 4 pressure channels, ie for a 64 channel scanner, 16 discrete CAN message identifiers are required. Alternatively for a more economical use of identifiers within a system, a single message id may be used, with all channels sent over this message sequentially. Channel numbers are positively identified with a channel identifier byte within the message.

For the multiple message option, data are sent on 8 byte messages, 4 channels per message – the pressure channels associated with a particular message number being fixed. The 2 most significant digits of the message identifier are user selectable via the setup program – so for example the identifier for channels 1-4 for user selected 0x22n would be 0x220, and the identifiers for channels 5-8, 9-12 etc. follow on incrementally from this first identifier as shown in Figure 18. It is the user's responsibility to avoid overlap of message identifiers in a multiple device installation.

Data are two bytes unsigned, scaled to +/- full scale, ie 0x0000 might represent –5psi, 0xffff +5psi. The data are user selectable as 16 bit big or little ended. An example of the packing is illustrated in Figure 18. Use of the 16 bit little ended option is recommended as it requires least processor overhead within the device.

The flightDAQ2 also supports absolute pressure encoding over CAN. The encoding is the same as for the TCP & UDP protocols discussed previously. The conversion table & equation are shown here again for completeness:

Range (psid)	0 bits	65535 bits	Conversion 'M' val	Conversion 'C' val
FS <= 2.5 psi	15000 Pa (2.17557psi)	115000 Pa (16.68psi)	4518.539969	2.1755661
5 & 8 psi (FS > 2.5-8 psi)	13000Pa (1.885psia)	160000Pa (23.206psia)	3073.842279	1.8854906
8 psi (FS > 8-15 psi)	15000Pa (2.17557psi)	207000Pa (30.0228psia)	2353.374583	2.1755661
30 & 45 psi (FS >15 psi)	0Pa (0 psia)	400000Pa (58.0151psia)	1129.619798	0
100 & 150 psi	0 Pa (2.17psi)	1140000 Pa (165.343psi)	396.3578239	0

When converting from binary to engineering units, the following equation can be used, substituting the 'M' & 'C' values from the above table, as appropriate for the range in question:

Engineering units (psi) = (Binary value / M) + C

The single message identifier option is included to reduce the number of message ID's required within a system. Data are packed in 7 byte messages as 3 channels per message with one channel identifier byte. This identifier byte is incremented for each message in the sequence, starting at 0x00 for channels 1,2 3, then 0x01 for channels 4,5,6 etc. Odd leftover channels (eg 16 channels/3 = 5 full messages plus a message with a single channel and two 'leftovers') are set to 0x0000 and should be discarded. Figure 19 shows the structure of a message for the single message identifier protocol. A selectable inter message delay of between 1ms and 200ms is available to match message generation timing to the user's system. Note that it is the user's responsibility to select appropriate delays with respect to channel rate and CAN bus baudrate.

	Message ID								
Data Byte	**0x220**	**0x221**	**0x222**	**0x223**	**0x224**	**0x225 – 0x22E**	**0x22F**		***0x230***
7	CH4 MSB	CH8 MSB	CH12 MSB	CH16 MSB	Ch20 MSB	Ch x+3 MSB	Ch64 MSB		
6	CH4 LSB	CH8 LSB	CH12 LSB	CH16 LSB	Ch20 LSB	Ch x+3 LSB	Ch64 LSB		*F/W Minor Revision*
5	CH3 MSB	CH7 MSB	CH11 MSB	CH15 MSB	Ch19 MSB	Ch x+2 MSB	Ch63 MSB		*F/W Minor Version*
4	CH3 LSB	CH7 LSB	CH11 LSB	CH15 LSB	Ch19 LSB	Ch x+2 LSB	Ch63 LSB		*F/W Major Version*
3	CH2 MSB	CH6 MSB	CH10 MSB	CH14 MSB	Ch18 MSB	Ch x+1 MSB	Ch62 MSB		*Temp MSB*
2	CH2 LSB	CH6 LSB	CH10 LSB	CH14 LSB	Ch18 LSB	Ch x+1 LSB	Ch62 LSB		*Temp LSB*
1	CH1 MSB	CH5 MSB	CH9 MSB	CH13 MSB	Ch17 MSB	Ch x MSB	Ch61 MSB		*Ref MSB*
0	Ch1 LSB	Ch5 LSB	Ch9 LSB	Ch13 LSB	Ch17 LSB	Ch x LSB	Ch61 LSB		*Ref LSB*

Figure 18 Example of 64 channel variant CAN multiple message packing using the 16 bit little ended data protocol – base identifier 0x220, with reference pressure setting enabled, to include in data stream as extra message (ID 0x224)

Data Byte	Content
6	CH3 - MSB
5	CH3 - LSB
4	CH2 - MSB
3	CH2 - LSB
2	CH1 - MSB
1	CH1 - LSB
0	0x00

Figure 19 Example of the CAN single message protocol using 16 bit little ended data– first message (ie channels 1,2,3)

Data Byte	Content
6	0x00
5	
4	
3	
2	CH64 - MSB
1	CH64 - LSB
0	0x16

Figure 20 Structure of the last CAN message using single message protocol (16 bit little ended data)

4.5.4 CAN Data Rate

The data delivery rate is selected from the setup program, and the following values are available (Hz) – 1, 2, 5, 10, 25, 50, 100, 312, 500, 625, 750, 1000. Although the system endeavours to deliver the rate with maximum accuracy, ultimate responsibility for data timing lies with the user's host system.

For microDAQ3 and flightDAQ2, as with TCP & UDP, the available data rates are also dependant on the sensor response setting of the microDAQ3. The maximum data rate available is equal to the sensor response setting thus to achieve the maximum data rate of 400Hz sensor response must be set to '400Hz', this is previously explained by Figure 14 Sensor response vs maximum data rate in the TCP section.

Care should be taken in selection of delivery rate, as it is possible to select unfeasibly fast data delivery for a particular bus baudrate, resulting in data loss. Similarly, for CANdaq5, since the scanners are addressed at a fixed channel rate of 20kHz (Gen1) or 50kHz (Gen2), requesting a data delivery of more than 20k/No. Channels or 50k/No. channels - ie 312Hz for a 64 channel Gen1 scanner - wastes resources and in some cases can cause the CANdaq5 to hang, requiring a power cycle.

4.5.5 Control Via CAN

The user command set is supported over the device CAN channel. The specification and function of the commands are detailed in section 2 above, however the CAN implementation is as follows.

The incoming message number is selected by the user from the front end software, though is constrained to be relative to the outgoing data base message identifier. The offset from the base identifier may be selected as being +0x10, +0x20, +0x30, +0x40 or +0x50. For example the base data message identifier of 0x220 might be set up to receive commands over CAN on message 0x230 (0x220 + 0x10).

In addition to the incoming message offset, the user may select whether the incoming command is acknowledged or not. The user command is a delimited 5 byte message that includes a block parity check, as shown in Figure 21. If the command is received without detected error, and the acknowledge option has been selected, the device will respond to a user command with a positive acknowledge byte ('*' or ASCII 42). Alternatively, if the command is received incorrectly it will respond with the negative acknowledge byte ('!' or ASCII 33). The acknowledgement is sent as a single byte message with identifier one greater than the receiving message. For the above example, the acknowledge will be sent on message 0x231.

Data Byte	Content
4	< (ASCII 60)
3	Parity
2	Parameter
1	Command
0	> (ASCII 62)

Figure 21 Data Structure of the CAN Incoming Control Message

4.6. Internal RAM – CANdaq5 only

4.6.1 Overview

Data acquisition to internal RAM allows for fast acquisition without the potentially limiting factor of external comms hardware (e.g. network hubs, PC CAN interfaces, etc.). Data can be acquired at whatever speed is necessary and then dumped to the host PC as a post-acquisition task.
Logging to RAM is only supported on the CANdaq5.

4.6.2 Internal RAM Protocol

The data protocols available for internal RAM storage are 16bit little endian or big endian only. Engineering unit conversion should be performed on the host PC once acquired data has been dumped to the PC.

4.6.3 Internal RAM Data Rate

The rate at which the data is acquired into Internal RAM is determined by the rate user command, and the following values are available (Hz) – 1, 2, 5, 10, 25, 50, 100, 312, 500, 625, 750, 1000.

As with other communications channels, care should be taken not to request a data delivery of more than 20k/No. channels (ie 312Hz for a 64 channel scanner).

4.6.4 Internal RAM Dump

Data can be dumped from internal RAM on any of the other three available communications channels. The format of the data dump is designed to be (almost) the same as if data was being streamed out of that communications channel. So for RS232 & TCP the format of data is the same as the 16bit LE or BE protocol formats detailed in sections 4.1.3 & 4.2.3 above (ie. Channel data preceded by a 3 byte header (0x00, 0xFF, 0x00). For CAN the format of the data is dependant on the selected CAN message type (single message or multiple message IDs) as detailed in section 4.4.3 above.

In addition to the above, an internal RAM dump sends a 9 (or 6) byte header, which is sent as soon as the 'Start Internal RAM Dump' user command has been received by the device. This header has the following format (note the first three bytes are <u>not</u> sent if the comms channel used for the dump is CAN).

Data Byte	Content
8	Total size of data dump (in bytes)
7	
6	
5	
4	Blocks per op
3	Number of channels
2*	0x00
1*	0xFF
0*	0x00

Figure 22 Data structure of Internal RAM dump header packet
(bytes marked * are not sent when dumping via CAN comms)

‘Blocks per op’ details the number of cycles of ‘n’ channels that are sent in one transmission packet – this is dependent on the number of channels being acquired (byte 3 in header above) and is irrelevant for CAN (always set to 1). After each packet is sent, the device waits for a handshake to confirm that the packet of data has been received at the host PC end. The dump handshake user command should be sent by the host PC to facilitate this (Note to avoid deadlock, the device will wait for a maximum 10 seconds for the handshake before auto sending the next packet). A handshake is also expected after the header packet has been sent by the device.

5. Valve Control user commands (not flightDAQ-TL)

The flightDAQ2, flightDAQ3 and 2432 contain a shuttle valve and internal purge valve
The microQDVP module is an accessory to the CANdaq5 (and older microDAQ2) unit that also contains the valving and drive signal.

The control of these valves is done via the same command protocol as with the other commands (see previous sections above). For the microQDVP a separate comms connection has to be made to the module (via Ethernet or CAN), whereas for the flightDAQ devices, the commands are just an extension of the standard command set and can be transmitted all through the same comms connection. These commands change the valves accordingly to perform a zero function or a purge function and also to simply move the shuttle valve between CAL & RUN

The microQD-VP Supports the following command subsection:

Command	Byte Char	ASCII	Parameter	Description
Reset	R	82	0: Hard Reset 1: Soft Reset	Request a device reset – A soft reset is used to re-initialise after some commands the QD-VP does not process the parameter, a soft reset is always performed.
Purge	U	85	1-255 – number of seconds to wait in Purge.	Performs a Purge function: 1. Shuttle scanner to Purge/CAL position 2. Switch on external purge valve 3. Wait for configured time (from Parameter byte) 4. Switch off external purge valve 5. Shuttle scanner to RUN mode Note that most valved scanners have a purge capability when placed in CAL position, the Purge U/85 command actuates the separate purge valve.
Shuttle	Y	89	0 – shuttle to CAL 1 – shuttle to RUN	Moves the scanner shuttle valve between CAL and RUN modes.
Shuttle time	u	117	1-255	Only supported in QD-VP f/w 2.0.1 on. Configures the duration the shuttle signal is asserted. Set 1-255 seconds.

The above table is an extract of microQD-VP applicable commands taken from section 2.4.

6. microDAQ-PCU Specific user commands

This section contains an excerpt of commands particular to this device.

Command	Byte Char	ASCII	Parameter	Description
Do Function	{	123	byte = 0xab a = 0 Perform Check a = 1 Perform Adjust b = 0 Abort Test b = 1 Zero (shuttle) b = 2 Span b = 3 5-Point* b = 4 FPN* b = 5 Leak Test	This command is used to make the device perform certain actions that may be specific to the device. e.g. have a microDAQ-PCU perform a CRG level rezero. A 5-point report is valid for check only and performs points of -90%, -50%, 0, 50% and 90% FPN permits a number to be passed specifying the exact pressure point to be checked. The Value is specified by following the command by a Data Packet as specified in Section 2.2. The pressure value is in kPa. See the dedicated section below. *These commands cannot include adjustment
Function Length	x	120	0 or 10-255	This command is used to set a function length for a 'Do' command. e.g. the function length for a calibration segment. The value is in integer seconds (i.e. 10=10 seconds) A parameter passed of '0' asks the microDAQ-PCU to calculate the required time. (future availability only) Command supported on firmware V3.0.0 on.
CRG	y	121	<CRG number>	Set the active CRG (Calibration Range Group). Valid numbers are 1-255. A value of 0 caused the microDAQ-PCU to hang-up connections with all devices and enter an idle state.
Data Return	z	122	byte = 0xab a = 0 Last Op Data a = 1 Coeff Data b = 0 Last Op Data b = 1 Zero b = 2 Span	Returns the data from the last operation. This could be zero offsets, span data or a full 5-point calibration report. 00 = Return the last measurement data 11 = Outputs the Zero linear calibration coefficients 12 = Outputs the Span linear calibration coefficients

6.1. Do Function { / 123

This command instructs the microDAQ-PCU to perform a number of separate actions contained within the function. Given the functions can be of longer duration, any Do Function command will offer an immediate acknowledgement, the progression and ultimate success of the request must be monitored by the Get Status command.

The command has a single parameter byte split into two nibbles, where byte=0xab.

Where nibble a is '0' the request is a check only. No adjustments re performed. Where a is '1' adjustments are performed for Zero and Span coefficients, these adjustments are non-volatile.

Nibble b performs the function type. Note a bad command return will be given if a 5-point adjustment is requested.

A typical usage example:

62 123 17 104 60 – Zero with shuttle and adjustment

A typical usage example:

62 123 4 125 60 <wait for ack> – Perform check at a pressure specified as:

00 00 32 64 96 – Value of 2.5 kPa

6.2. Function Length x / 120

This command sets the length of a segment to allow for settling time of the pressure measurement. To determine the duration required consider the Scanner filter settings and any settlement or lag time that may exist in the pipework. Setting this value too short may result in acquiring data that has not yet settled. The function permits delegation of this decision to the microDAQ-PCU, in this instance the system will make calculation based on the filter settings of the devices within the CRG.
A typical usage example:
62 120 0 122 60 – Set the function length to automatic.

6.3. CRG y / 121

This command selects (or deselects) the appropriate Calibration Range Group.
A CRG contains a list of devices and a pressure standard that have similar properties. The CRG must be configured using the microDAQ-PCU webserver.
Note that setting a CRG will attempt to connect to all devices within that CRG. User streaming MUST be stopped and the host disconnected or the microDAQ-PCU will fail to take control of the device.
The CRG command will offer an immediate acknowledgement regardless of success. Use the Get Status (short) to verify the success of the request. To change CRG ensure the CRG is set to '0', perform a Soft Reset (6 seconds) before reconnecting to select a new CRG.
A typical usage example:
62 121 1 122 60 – Connect CRG 1

6.4. Data Return z / 122

This command returns data from the Do Function or parameters from the Scanners. There is a requirement to have a CRG set to use this command. The response from this command is shown below, some of the responses are of significant length.
The channel data is returned in increasing channel number order so the ZCoef data below would be [ZCoef1] <Ch1>, <Ch2>, etc The number of fields will be determined by the scanner channel count.

The following tags are supported where *n* is the Scanner number within the CRG:
[CRG] The Calibration Range Group in use (1-255)
[Scanner*n*] The serial number of the Scanner in CRG slot *n*
[ZCoef*n*] The Zero Linear Calibration Coefficient for the Scanner in in CRG slot *n*
[SCoef*n*] The Span Linear Calibration Coefficient for the Scanner in in CRG slot *n*
[Press. units] The pressure units used within the Data Return response (usually kPa)
[Press. type] The pressure calibrator mode. Absolute or Differential.
[PressureNS] The pressure Calibrator declared Pressure for the Negative Span point
[PressureNH] The pressure Calibrator declared Pressure for the Negative 50% point
[PressureZ] The pressure Calibrator declared Pressure for the Zero point
[PressurePH] The pressure Calibrator declared Pressure for the Positive 50% point
[PressurePS] The pressure Calibrator declared Pressure for the Positive Span point
[Data*n*NS] Data for Scanner in CRG slot *n* at the NS calibration point
[Data*n*NH] Data for Scanner in CRG slot *n* at the NH calibration point
[Data*n*Z] Data for Scanner in CRG slot *n* at the Z calibration point
[Data*n*PH] Data for Scanner in CRG slot *n* at the PH calibration point
[Data*n*PS] Data for Scanner in CRG slot *n* at the PS calibration point

It should be noted that a Span adjust will adjust both Zero and Span and return data for both using the [Data*n*Z] and [Data*n*PS] tags. It is not possible to adjust the Span only with this function.
Note that absolute products such as the 2432 and microDAQ3 will return 2 additional coefficients at the end of the list for Reference1 and Reference2.

Usage examples:
62 122 17 105 60 - Read Zero coefficient data.

[CRG] 2, [Scanner1] 2412301, [Scanner2] 2412302, [ZCoef1] 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, [ZCoef2] 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
0.00000000, 0.00000000,

Figure 23 Zero Coef Data Return example

62 122 18 106 60 - Read Span coefficient data

[CRG] 2, [Scanner1] 2412301, [Scanner2] 2412302, [SCoef1] 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, [SCoef2]…. etc

Figure 24 Span Coef Data Return example

62 122 0 120 60 – Get Last Op Data.
The data will always be returned in this order for a Zero, Span, specified value or 5 point check. Fields that are not relevant will be omitted.

[CRG] 2, [Press. units] kPa, [Press. type] Differential, [Scanner1] 2412301, [Scanner2] 2412302, [PressureNS] -0.90000000, [Data1NS] -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, -0.90000000, [Data2NS]…. Etc….[PressureNH] -0.50000000, [Data1NH] -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, -0.50000000, [Data2NH]…. Etc….[PressureZ] 0.00000000, [Data1Z] 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, [Data2Z]…. Etc….[PressurePH] 0.50000000, [Data1PH] 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, 0.50000000, [Data2PH]…. Etc…. [PressurePS] 0.90000000, [Data1PS] 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, 0.90000000, [Data2PS]…. Etc….

Figure 25 Get Last Op Data Return (5 point check example)

62 122 0 120 60 – Get Last Op Data (where last Op was Zero adjust or check)

[CRG] 2, [Press. units] kPa, [Press. type] Differential, [Scanner1] 2412301, [Scanner2] 2412302, [PressureZ] 0.00000000, [Data1Z] 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,

6.5. Leak Test

The Do Function Leak test will pressurise the calibration line then monitor this for a pressure decay for a duration set by the function length. The command will issue a Pass / Fail in response to a Get Data command based on a criterion set on the webserver PCU config tab. The Response is as follows:

[Press. Type] Differential, [Scanner1] 2411702, [Scanner2] 2411701, [PressureStart] 2.5000, [PressureEnd] 2.49500, [Leak] Pass

6.6. Operation Sequence

A typical usage would be:

Disconnect any host operation currently running to permit the microDAQ-PCU to obtain control, then:

62 121 1 122 60 – Select a CRG, in this example CRG 1
62 120 0 122 60 – Set the function length to automatic
62 123 17 104 60 – Zero with shuttle and adjustment
62 63 0 61 60 – Get a short status to verify the Do Function is complete with a clean exit
62 123 3 122 60 – Perform a 5 point check
62 63 0 61 60 – Get a short status to verify the Do Function is complete with a clean exit
62 122 0 120 60 – Data Return (to permit performance validation)
Where adjustments are necessary, the Span can be adjusted..
62 123 18 107 60 – Adjust Span.
62 63 0 61 60 – Get a short status to verify the Do Function is complete with a clean exit
62 123 3 122 60 – Perform a 5 point check
62 63 0 61 60 – Get a short status to verify the Do Function is complete with a clean exit
62 122 0 120 60 – Data Return (to permit performance validation)
62 121 0 123 60 – Drop the selected CRG
62 63 0 61 60 – Get a short status to verify disconnection is complete with a clean exit

7. Annex A – Span Hi Values

The following table lists selectable 'Span Hi' values:

Span Hi Index	Span Hi Value (kPa)	Span Hi Value (psi unless stated)
0	0.747266757	0.10838188 (3" WC)
1	0.996355654	0.14450917 (4" WC)
2	1.245444550	0.18063646 (5" WC)
3	1.494533377	0.21676374 (6" WC)
4	1.743622273	0.25289103 (7" WC)
5	1.992711170	0.28901832 (8" WC)
6	2.241800066	0.32514561 (9" WC)
7	2.989066754	0.43352748 (12" WC)
8	3.487244547	0.50578206 (14" WC)
9	3.985422339	0.57803664 (16" WC)
10	4.483600132	0.65029122 (18" WC)
11	4.13685	0.6
12	4.82633	0.7
13	5.51581	0.8
14	6.20528	0.9
15	10.3421	1.5
16	12.06583	1.75
17	13.7895	2
18	15.5132	2.25
19	20.6843	3.0
20	24.1317	3.5

Span Hi Index	Span Hi Value (kPa)	Span Hi Value (psi unless stated)
21	27.579	4.0
22	31.0264	4.5
23	41.3685	6.0
24	48.2633	7.0
25	55.1581	8.0
26	62.0528	9.0
27	72.39495	10.5
28	82.7371	12.0
29	93.07922	13.5
30	124.106	18.0
31	144.79	21.0
32	165.474	24.0
33	186.158	27.0
34	217.1849	31.5
35	248.211	36.0
36	279.2377	40.5
37	413.685	60.0
38	482.633	70.0
39	551.581	80.0
40	620.528	90.0

Table 4 Valid Span Hi values for Set Span Hi command

8. Annex B – User Command Table

8. Annex B - User Command Table

✓ - Fully supported ✓ - Partially supported (see description for more info) × - Not supported

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Stream OFF	0	48	1 - TCP / UDP 2 - CAN 3 - Internal RAM	Disable the data delivery for the chosen channel. CAN: microDAQ3, CANdaq5, flightDAQ2 Only RAM: CANdaq5 Only	×	✓	✓	✓	✓	✓	✓	✓
Stream ON	1	49	1 – TCP / UDP 2 - CAN 3 - Internal RAM 4 - Internal RAM (stop on full)	Enable the data delivery for the chosen channel. The delivery rate is as selected in the webserver setup, unless it is modified via the Rate command. CAN: microDAQ3, CANdaq5, flightDAQ2 Only RAM: CANdaq5 Only Internal RAM data streaming will either log continuously, wrapping when the end of available RAM is reached, or will stop when the RAM is full, depending on parameter.	×	✓	✓	✓	✓	✓	✓	✓
Get Status	?	63	0 : Short 1 : With temp. 2 : Full 3 : Pressure reading 4 : Temp. readings 5 : Fault / Excitation* 6 : Valve position 7 : Firmware ID 8 : Unit serial number 9 : Internal Scanner serial no.	Return a status packet from the microDAQ. Three main versions are available, '*short*' , '*with temp* .' and '*full*' . Short returns status byte information showing current operating state only, whereas '*full*' returns microDAQ setup options and temperature data in addition. '*With temp* .' returns the status and temperature information intended for continuous temperature monitoring applications. The data format is documented in a separate section. *Parameter 5 reports fault information for microDAQ-PCU and Excitation for CANdaq5. *Parameter 9 only applicable for CANdaq5. Additional readings poll raw values for temperature and pressure as read from the scanner, as well as other fields as indicated in the table	✓	✓	✓	✓	×	✓	✓	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Span	A	65	0 – absolute 1 - differential	Request a 'span' operation. The unit should have been recently rezeroed, then the appropriate span pressure (as selected from the webserver or with the Set Span Hi command) applied to the scanner. Requesting a span operation will then calculate the span correction coefficient for each channel. All rezero values and span coefficients are then written to the device. *microDAQ2 and flightDAQ2 (pre V3.0.0) do not support a passed parameter. Devices are only differential. Deprication Advice: Consider using 'Do Function' ({/123 para 4) to support a passed span value.	✗	✓	✓	✓	✓	✓	✓	✓
Rebuild Calibration	C	67	None	Legacy support only - The calibration is rebuilt for the current temperature, using the active settings.	✗	✗	✓	✓	✗	✗	✗	✗
Derange	D	68	None	Legacy support only - Calibration is rebuilt with the appropriate deranging factor applied, the DTC scanner deranging gain is switched into circuit.	✗	✗	✓	✓	✗	✗	✗	✗
Reset Linear Calibration	E	69	0 – **Clear** and store rezero and span values to eeprom 1 – **Store current** rezero and span values to eeprom	Either clears zero offsets, resets absolute and differential span values, then stores the values in EEPROM or stores the current zero, absolute span and differential span values to EEPROM. *CANdaq5 does not support parameter '1'	✗	✓	✓	✓	✓	✓	✓	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Data Filters	F	70	Byte = 0xab a - Impulse filter a = 0 - Off a = 8 - On b - Averaging filter b = 0 - Off b = 1 - 2 b = 2 - 4 b = 3 - 8 b = 4 - 16 b = 5 - 32 b = 6 - 64 b = 7 - 128 b = 8 - 256 b = 9 - 512 b = 10 - 1024 b = 11 - 2048 b = 12 - 4096 b = 13 - 8192 b = 14 - 16384 b = 15 - 32768 b = 16 – 65536	Sets the input filtering options for the primary readings, applied before the calibration LUT is applied. Options available are: - Impulse filter: applies a median filter to remove noise spikes – parameter is On or Off - Averaging filter: applies a rolling average buffer – parameter is size of averaging buffer Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset. Note the impulse filter is likely to remove data samples that appear anomalous, repeating the last data. As such when using timestamping this can appear as missing / duplicate data. For best performance with timestamped data ensure this setting is off. 2416, 2432T do not support the impulse filter so bit 'a' of the parameter is irrelevant and ignored.	✗	✓	✓	✓	✓	✓	✓	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Rezero and Rebuild	G	71	None	Legacy support only - Requests a rezero followed by a calibration table rebuild, with the calculated zero offsets being applied to the calibration data.. Note these rezero offsets are separate from any normal rezero that may be performed afterwards.	✗	✗	✓	✓	✗	✗	✗	✗
Sensor Response	G	71	0 – 400Hz/500Hz 1 – 200Hz 2 – 100Hz 3 – 50Hz 5 – 1000Hz (2416 only!)	Sets the response mode of the internal sensors. The sensor response setting determines the maximum available data rate. Note: This setting will not be applied until the device is power cycled/reset. Best timestamp accuracy is achieved with this setting at the highest permissible value. Note: For 2416, 500Hz replaces 400Hz and in addition it supports a 1KHz sensor response.	✗	✓	✗	✓	✗	✗	✓	✓
Channels	H	72	byte = 0xab a = 1 TCP / UDP a = 2 CAN a = 3 Internal RAM b = 0 - 16 b = 1 - 32 b = 2 - 48 b = 3 - 64	Set the number of active channels returned to the user for a data channel. Setting a value greater than the current system maximum channels results in the maximum channels value being used. For flightDAQ-TL channels 1-16 are primary channels, 17-32 are secondary channels. flightDAQ3 & 2432 only support a maximum of 32 channels. CAN: microDAQ3, CANdaq5, flightDAQ2 Only RAM: CANdaq5 Only microDAQ3 does not process nibble 'a'. All protocols are set the same. Bug Notice: microDAQ3 V1.0.13 to V1.0.16 use transposed index values: 0:64, 1:16, 2:32, 3:48	✗	✓	✓	✓	✓	✓	✗	✗
Start Internal RAM Dump	I	73	1 - TCP / UDP 2 - CAN	Starts the internal RAM dump, sending data using the comms channel chosen by the parameter. The first packet is sent immediately and consists of a 9 byte header (only 6 bytes for CAN - see later).	✗	✗	✓	✗	✗	✗	✗	✗

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Internal RAM Dump Handshake	J	74	None	Handshake command to indicate previous packet was received. Must be sent after header packet & all subsequent data packets.	✗	✗	✓	✗	✗	✗	✗	✗
Maximum Channels	M	77	0 - 16 Jan-32 Feb-64	Legacy support only - Set the system maximum channels, ie the number of channels read from the scanner. Allows the dynamic alteration of the effective per channel sampling rate.	✗	✗	✓	✗	✗	✗	✗	✗
Temperature Filters	N	78	Byte = 0xab a - Impulse filter a = 0 - Off a = 8 - On b - Averaging filter b = 0 - Off b = 1 - 2 b = 2 - 4 b = 3 - 8 b = 4 - 16 b = 5 - 32 b = 6 - 64 b = 7 - 128 b = 8 - 256 b = 9 - 512 b = 10 - 1024 b = 11 - 2048 b = 12 - 4096 b = 13 - 8192 b = 14 - 16384 b = 15 - 32768 b = 16 - 65536	Legacy support only - Sets the input filtering options for the CANdaq5 temperature readings. Options available are: - Impulse filter: applies a median filter to remove noise spikes – parameter is On or Off - Avergaing filter: applies a rolling average buffer – parameter is size of averaging buffer Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset.	✗	✗	✓	✗	✗	✗	✗	✗

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Poll	O	79	1 - TCP / UDP 2 – CAN	Request a single data packet (in the current active format) for the channel selected. Data streaming should be set off before using this command. Note that there is no positive acknowledge for this command. This command is not valid for logging to Internal RAM. CAN supported on CANdaq5, microDAQ3, flightDAQ2 only.	✗	✓	✓	✓	✓	✓	✓	✓
Protocol	P	80	byte = 0xab a = 1: TCP / UDP* a = 2 :CAN[1] b = 0 : Binary LE b = 1 : Binary BE b = 2 : Engineering Units[2]	Permits the changing of the data protocol. The abbreviations LE and BE in the table refer to little and big endian respectively, denoting whether the less significant byte is sent first or last. Acquired data is available in engineering units over UDP and TCP connections, only binary data is available with CAN and Internal RAM. CAN supported on CANdaq5, microDAQ3, flightDAQ2 only. Binary data provides pressure scaled as unsigned integers to the operating full scale, eg for 16 bit resolution, 0 to 65535 represents -FSD to +FSD for differential. Absolute scaling follows configurable scaling typically as per *Table 1 - Typical absolute scaling* flightDAQ-TL uses only 32bit binary IEEE754 float data, rather than 16 bit. But the endianess may be set using b=0 or 1. [1]microDAQ3 and flightDAQ2 do not process nibble 'a' to select TCP(UDP) or CAN. The command sets all protocols. [2]Engineering Units on microDAQ2 & CANdaq5 result in an ASCII data stream output. Engineering Units on 2416, the data stream is 32bit IEEE754 encoded float data. On all other variants, the EU protocol is NOT supported. Note: IENA endianness may be set on later firmware versions.	✗	✓	✓	✓	✓	✓	✓	✓
Reset	R	82	0: Hard Reset 1: Soft Reset	Request a device reset – A soft reset is used to re-initialise after some commands, a hard reset performs a full reset similar to a power cycle. . Note there is no positive acknowledge for this command. *microDAQ3, CANdaq5 & flightDAQ2 do not process the parameter, a soft reset is always performed.	✓	✓	✓	✓	✓	✓	✓	✓
Standby	S	83	None	Set all data streaming off.	✓	✓	✓	✓	✓	✓	✓	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Hardware Trigger	T	84	byte = 0xab a = 0 Disable a = 1 Enable b = 1 TCP / UDP b = 2 CAN b = 3 Internal RAM* b = 4 Internal RAM (stop on full)*	Enables or disables the hardware trigger on the selected channel. Note that there is no positive acknowledge for this command. Assuming a valid hardware trigger is detected for the duration, Internal RAM data streaming will either log continuously, wrapping (and therefore overwriting) when the end of available RAM is reached, or will stop when the RAM is full (and automatically disable the hardware trigger), depending on the parameter. **Always ensure sensor response (G/71) is set to maximum available rate to minimise 'jitter'.** *Only CANdaq5/microDAQ2 offers stream to RAM. flightDAQ-TL & microDAQ-PCU do not support Hardware Trigger	✗	✓	✓	✓	✗	✓	✓	✓
Purge	U	85	0-255 – number of seconds to Purge. 0 – indefinite, cancel by sending command with 1 second wait.	Performs a Purge function: 1. Shuttle scanner to Purge/CAL position 2. Switch on external purge valve 3. Wait for configured time (from Parameter byte) 4. Switch off external purge valve 5. Shuttle scanner to RUN mode Note that most valved scanners have a purge capability when placed in CAL position, the Purge U/85 command relates only to those products with an internal, separate purge valve	✗	✗	✗	✓	✗	✓	✗	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Rate (Part1)	V	86	byte = 0xab a = 1 TCP / UDP a = 2 CAN a = 3 Internal RAM For TCP / UDP: b = 0 : Off b = 1 : 1000Hz * b = 2 : 625Hz * b = 3 : 500 Hz * b = 4 : 400 Hz b = 5 : 312 Hz * b = 6 : 225 Hz * b = 7 : 200 Hz b = 8 : 150 Hz b = 9 : 100 Hz b = 10 : 50 Hz b = 11 : 25 Hz b = 12 : 20 Hz b = 13 : 10 Hz b = 14 : 5 Hz b = 15 : 1 Hz	The rate command can be different between products; therefore 'Rate' has duplicated entries here. Be sure to check the compatibility matrix to the right to select the correct information. Adjust the data delivery rate for each communication channel. The upper nibble of the parameter byte selects which communication channel, while the lower selects its data rate. CAN & Internal RAM 'comms' have the same available data rates. (*) TCP Data rates are the same for all devices up to 200Hz. microDAQ3 does not permit selections of 1,2,3,5 or 6. Note that the achieved rate achieved can be limited by other device configuration, such as Response.	✗	✓	✓	✓	✗	✗	✗	✗

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
			For CAN & Internal RAM: b = 0 : Off b = 1 : 1000 Hz b = 2 : 750 Hz b = 3 : 625 Hz b = 4 : 500 Hz b = 5 : 312 Hz b = 6 : 100 Hz b = 7 : 50 Hz b = 8 : 25 Hz b = 9 : 10 Hz b = 10 : 5 Hz b = 11 : 2 Hz b = 12 : 1 Hz								

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Rate (Part2)	V	86	byte = 0xab a = 1 TCP / UDP For TCP / UDP: b = 0 : Off b = 1 : reserved b = 2 : reserved b = 3 : reserved b = 4 : 1000Hz (2416 only) b = 5 : 500Hz (2416 only) b = 6 : 250 Hz b = 7 : 200 Hz b = 8 : 150 Hz b = 9: 100 Hz b = 10 : 50 Hz b = n/a* : 33 Hz b = 11 : 25 Hz b = 12 : 20 Hz b = 13 : 10 Hz b = 14 : 5 Hz b = 15 : 1 Hz	The rate command can be different between products; therefore 'Rate' has duplicated entries here. Be sure to check the compatibility matrix to the right to select the correct information. The upper nibble of the parameter byte selects which communication channel, while the lower selects its data rate. CAN & RAM commands are not supported on flightDAQ-TL, flightDAQ3, 2432, 2432T & 2416 devices. *33 Hz is only available via webserver or iDDS selection. Note: rates above 250Hz are only supported on 2416 Note that the achieved rate achieved can be limited by other device configuration, such as Response.	✗	✗	✗	✗	✓	✓	✓	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Zero (with shuttle action)	W	87	1-255 – number of seconds to wait in CAL before switching back to RUN	Performs a Zero function: 1. Shuttle scanner to PURGE/CAL mode 2. Wait for configured time (from Parameter byte) 3. Perform Rezero 4. Shuttle scanner to RUN mode Command only supported where a CAL valve feature exists	✗	✗	✗	✓	✗	✓	✗	✓
Shuttle	Y	89	0 – shuttle to CAL 1 – shuttle to RUN 2 – shuttle to PURGE	Moves the scanner shuttle valve between CAL and RUN modes. *supported on microDAQ where a valve is fitted. PURGE position only exists on 2416	✗	✓	✓	✓	✗	✓	✗	✓
Rezero	Z	90	0: All channels 1 to 32: rezero specified channel	Request a rezero. Channel parameter is only supported on the flightDAQ-TL, flightDAQ3, 2432, 2432T & 2416 devices – where no parameter is specified a '0' is implemented.	✗	✓	✓	✓	✓	✓	✓	✓
Temperature Compensation	[	91	0 – Off 1 – Continuous, inline 2 – With zero only	Legacy support only - Enables/Disables the temperature compensation feature. Parameter indicates whether the compensation is run inline continuously or on demand, with zero. Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset.	✗	✗	✓	✗	✗	✗	✗	✗
Set Span Hi	\	92	0 to 40 = Span Hi Value (in psi) 41 = Save Zero only	This command sets the Span value to be used for **differential** Span ONLY (See Span A/65). Absolute Span Hi (where applicable) is fixed at 90% Sets the span hi pressure expected when performing a Span calibration. See Figure 4 for Span pressures for each parameter value. 'Save Zero only' is used to save only the current zero offsets when a Span function is performed. Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset.	✗	✓	✓	✓	✗	✗	✗	✗

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Scanner Acquisition Frequency	]	93	0 – 20KHz 1 – 50KHz	Legacy support only - Sets the scanner acquisition frequency. Note Gen1 ESP scanners do not support 50KHz acquisition mode. Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset.	✗	✗	✓	✗	✗	✗	✗	✗
Pressure Type	a	97	Byte = 0xab b = 0 : Absolute pressure output b = 1 : Differential pressure output a (upper **5** bits) = 1 to 16 : channel number * 0 : all channels *	Sets the pressure type of the output stream to either Absolute pressure or a differential from the reference pressure. (*) 2416 only – specify the channel number that the pressure type setting refers to. Parameter of 0 indicates all channels.	✗	✓	✗	✓	✗	✓	✗	✓
Base CAN ID (LSB)	c	99	Byte = 0xab a = 0 to F b = 0, 4, 8 or C	Sets the least significant Byte of the Base CAN ID. .	✗	✓	✗	✓	✗	✗	✗	✗
Base CAN ID (MSB)	d	100	Byte = 0x0a a = 0 to 7	Sets the most significant Byte of the Base CAN ID.	✗	✓	✗	✓	✗	✗	✗	✗
Burn To EEPROM	e	101	Legacy: no parameter 2416: Byte = 0xab a (upper **5** bits) = 1 to 16 : channel number * b (lower 3 bytes) = 0 : all cases	Tell the device to Burn all configured settings to EEPROM for non-volatile storage. This includes, data rate, protocol, filters, time stamping, etc.). (*) 2416 only – this command is used to save all PSM settings to the non-volatile storage of the given channel PSM e.g write Ch1: 62 101 08 111 60	✗	✓	✓	✓	✗	✗	✓	✓
PTP Sync	p	112	0 – Off 1 – On	Sets whether the device attempts to synchronise to an external PTP Grandmaster clock. Clock time domain number is '0' as default. Domain number setting (where supported) is on the webserver. Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset. *flightDAQ-TL, flightDAQ3, 2432, 2432T & 2416 have PTP Sync enabled by default.	✓	✓	✓	✓	✓	✓	✓	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Time Format	q	113	0 – UTC 1 – TAI	Sets whether the internally stored time is represented in TAI or UTC format. TAI is atomic time, whereas UTC incorporates additional leap seconds (currently -37 seconds at time of publication) Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset. *microDAQ-PCU, CANdaq5, flightDAQ-TL, flightDAQ3, 2432, 2432T and 2416 support this feature on webserver ONLY.	✓	✓	✓	✓	✓	✓	✓	✓
Ref CAN ID (LSB)	r	114	Byte = 0x00 to 0xFF	Sets the least significant Byte of the units reference output CAN ID.	✗	✓	✗	✓	✗	✗	✗	✗
Ref CAN ID (MSB)	s	115	Byte = 0x0a a = 0 to 7	Sets the most significant Byte of the units reference output CAN ID.	✗	✓	✗	✓	✗	✗	✗	✗
Timestamp in Data stream	t	116	0 – None 1 – Start of Cycle 2 – Every Channel	Sets how the time stamp data is included in the data output, available options: None – No timestamps are added to the data packets. Start of Cycle – A single 8-byte timestamp is added before a set complete set of channel data. Every cycle – An 8-byte timestamp is added before the data for each channel. (can limit maximum data rate) This command is only relevant for the TCP/UDP data stream. Note this setting is volatile unless the Burn To EEPROM command is issued prior to a power cycle/reset.	✗	✓	✓	✓	✗	✓	✓	✓
Shuttle time	u	117	1-255	For pneumatic shuttle (QD-VP) Only. Configures the duration the shuttle signal is asserted. Set 1-255 seconds.	✗	✗	✗	✗	✗	✗	✗	✗
Function Length	x	120	0 or 10-255	This command is used to set a function length for a 'Do' command. e.g. the function length for a calibration segment. The value is in integer seconds (i.e. 10=10 seconds) A parameter passed of '0' asks the microDAQ-PCU to calculate the required time.	✓	✗	✗	✓	✗	✓	✗	✗
CRG	y	121	<CRG number>	Set the active CRG (Calibration Range Group). Valid numbers are 1-255. A value of 0 caused the microDAQ-PCU to hang-up connections with all devices and enter an idle state.	✓	✗	✗	✗	✗	✗	✗	✗

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Data Return	z	122	byte = 0xab a = 0 Last Op Data a = 1 Coeff Data b = 0 Last Op Data b = 1 Zero b = 2 Span	Returns the data from the last operation. This could be zero offsets, span data or a full 5-point calibration report. 00 = Return the last measurement data 11 = Outputs the Zero linear calibration coefficients 12 = Outputs the Span linear calibration coefficients *flightDAQ2,flightDAQ3, 2432, 2432T & 2416 support return of zero and span coefficients only.	✓	✗	✗	✓	✗	✓	✓	✓
Do Function	{	123	byte = 0xab a = 0 Perform Check a = 1 Perform Adjust b = 0 Abort Test b = 1 Zero (shuttle) b = 2 Span b = 3 5-Point* b = 4 FPN b = 5 Leak Test b = 6 FPN with mask	This command is used to make the device perform certain actions that may be specific to the device. e.g. have a microDAQ-PCU perform a CRG level rezero. A 5-point report is valid for check only and performs points of -90%, -50%, 0, 50% and 90% FPN permits a number to be passed specifying the exact pressure point to be adjusted or checked. The Value is specified by following the command by a Data Packet as specified in Data Packet.Section 2.2. The pressure value is in kPa. See the dedicated section below. *These commands cannot include adjustment microDAQ3-64, FD2, FD3, 2432, 2432T & 2416 can process adjust FPN (a=1,b=4) only. For shuttle Zero (if applicable) (a=1,b=1) use command W/87. flightDAQ-TL supports only adjust FPN with mask (a=1,b=6) this command expects two following Data Packets as specified in Data Packet.2.2. As follows: <command> <Span value> <Mask Value> The -TL uses a mask per bit principal LSB = Ch1	✓	✓	✓	✗	✓	✓	✓	✓

Command	Byte Char	ASCII	Parameter	Description	Supported Device							
					microDAQ-PCU	microDAQ3-64	CANdaq5 / microDAQ2	flightDAQ2	flightDAQ-TL	2432 / flightDAQ3	2432T	2416
Pressure Units	|	124	Byte: 0xab b = 0 : Pa b = 1 : PSI b = 2 : bar b = 3 : mbar b = 4 : kPa a (upper **5** bits) = 1 to 18 : channel number (inc Ref & Purge) 0 : all channels	Sets the units configuration for EU data for the given channel. Parameter of 0 indicates all channels.	✗	✗	✗	✗	✗	✗	✗	✓
Temperature Units	|	124	Byte: 0xab b = 0 : Celcius b = 1 : Farenheit b = 2 : Kelvin	Sets the units configuration for EU data.	✗	✗	✗	✗	✗	✗	✓	✗