Fabry Perot Temperature/ Vacuum Controller

Simon Tulloch

QUCAM

Manual V1.1

Jerez, 26 Jan 2021

1 Introduction

The QUCAM milliKelvin controller has been designed for low-noise high-stability temperature control. Low drift has been the principle design driver. The basic temperature controller design has been modified to add control of the vacuum pump, the electro-valve and the vacuum gauge in the high stability Fabry-Perot Cavity system. The electro-valve and pressure sensor connect directly to this controller which provides all necessary power. The pump is also powered from this controller via a 220V Solid State Relay. Three automatic pumping modes are available to either pump on a schedule or alternatively whenever the pressure raises above a threshold. These modes provide automatic sequencing of valve and pump. Valve and pump states can always be over-ridden using the front panel controls.

2 System Overview

The main system elements are shown schematically in Figure 1. There are 4 temperature sensor inputs with 4-wire geometry for increased accuracy. Three of these are designed for diode sensors, the fourth for Pt100 sensors. Preamplifiers , high resolution ADC and voltage reference are contained in a temperature controlled “oven” section of the mother-board PCB. Three linear servo power amplifiers are present, one of which is used to control this oven, the other two are available to the user.

The Real Time Clock (RTC) uses a CR2325 Lithium battery that will maintain power for several years when the controller is powered off. The clock accuracy is +/-2ppm from 0 to 40C.

Calibration curves for up to three types of temperature sensors are contained within an internal EEPROM device. The standard curves present are Pt100, Lakeshore DT470 and the low-cost IN4148 diode. Other curves can be supplied on request and programmed using a Python application.

Front panel LED indicators and an 11-page LCD screen give “at a glance” confirmation of controller operation and stability. All temperature data are internally logged so that the performance history is always available to the user.

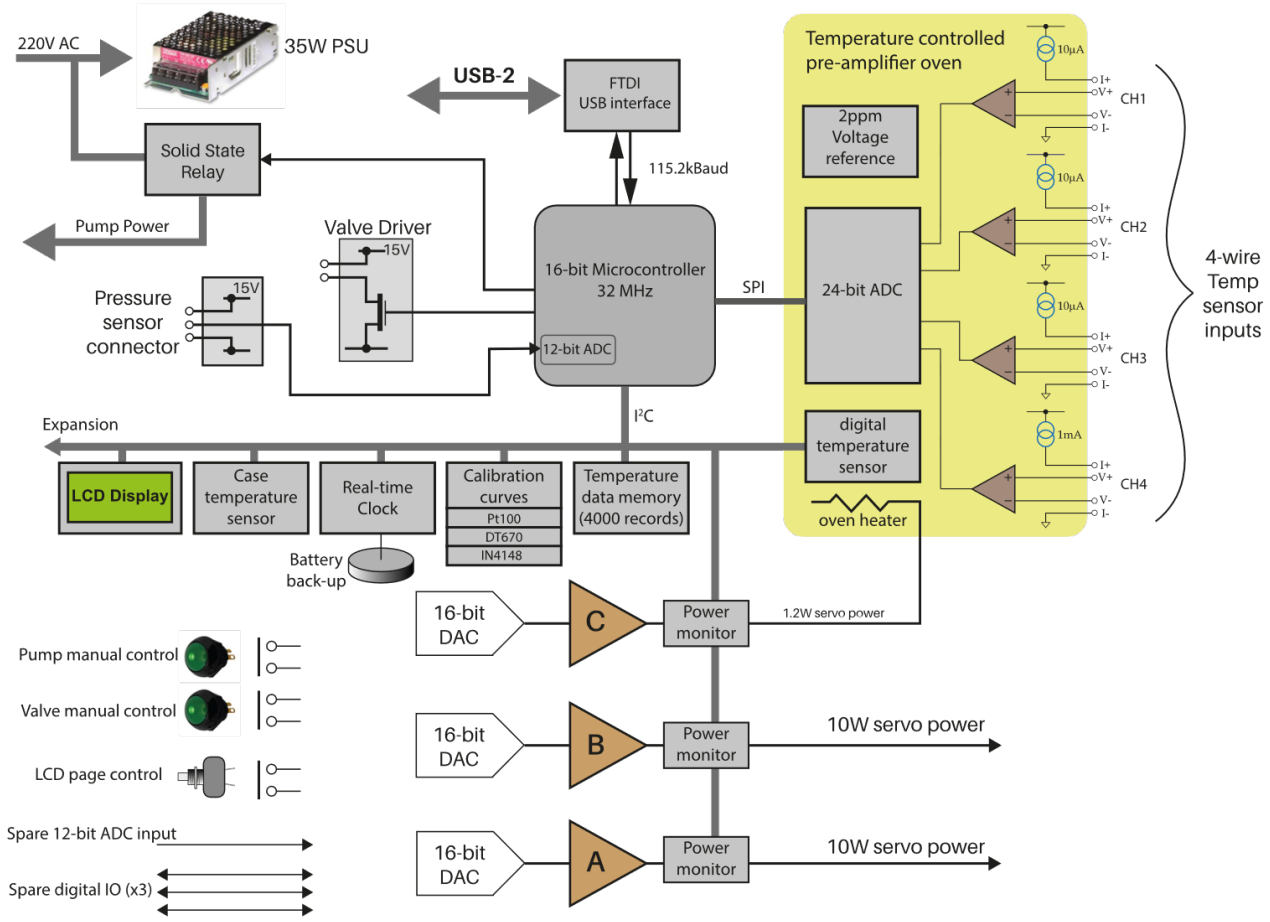


Figure 1. System schematic

The controller is “stand alone” although for initial configuration and data transfer it will need to be connected to a host device via USB2.

Data conversion is done using 24-bit ADCs and 16-bit DACs for maximum performance. Additionally the ADC contains a 50Hz/60Hz notch filter to reject mains interference.

The preamplifier oven, with insulation removed, is shown in Figure 2.

3 Front and rear panel description.

Front and rear panels are shown in Figure 3. The green illuminated buttons on the right hand side of the front panel provide manual control of the valve and pump. These buttons will always over-ride any automatic pumping modes that may be currently running. There is also a 4-line LCD

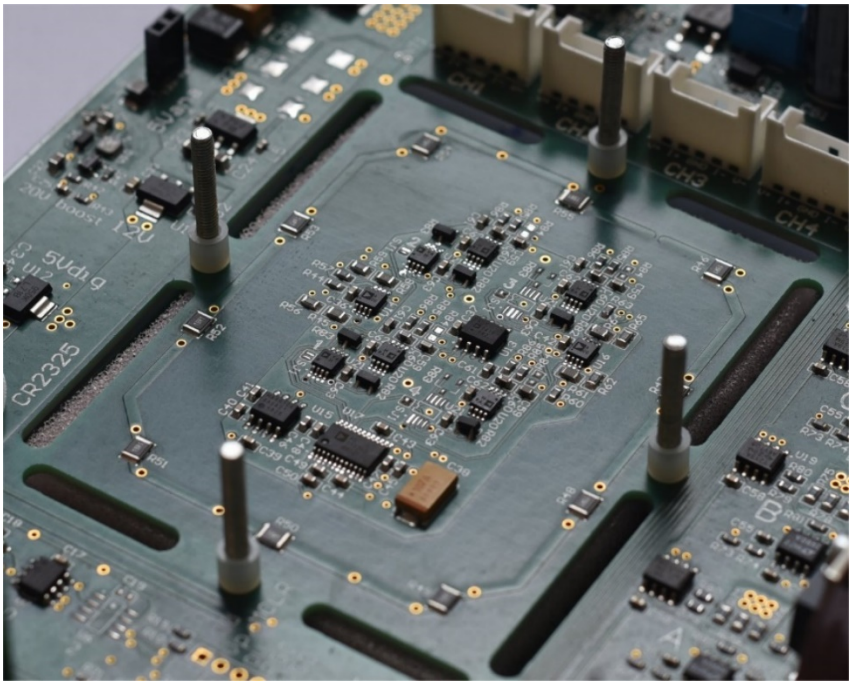


Figure 2. Preamplifier oven with insulation removed.

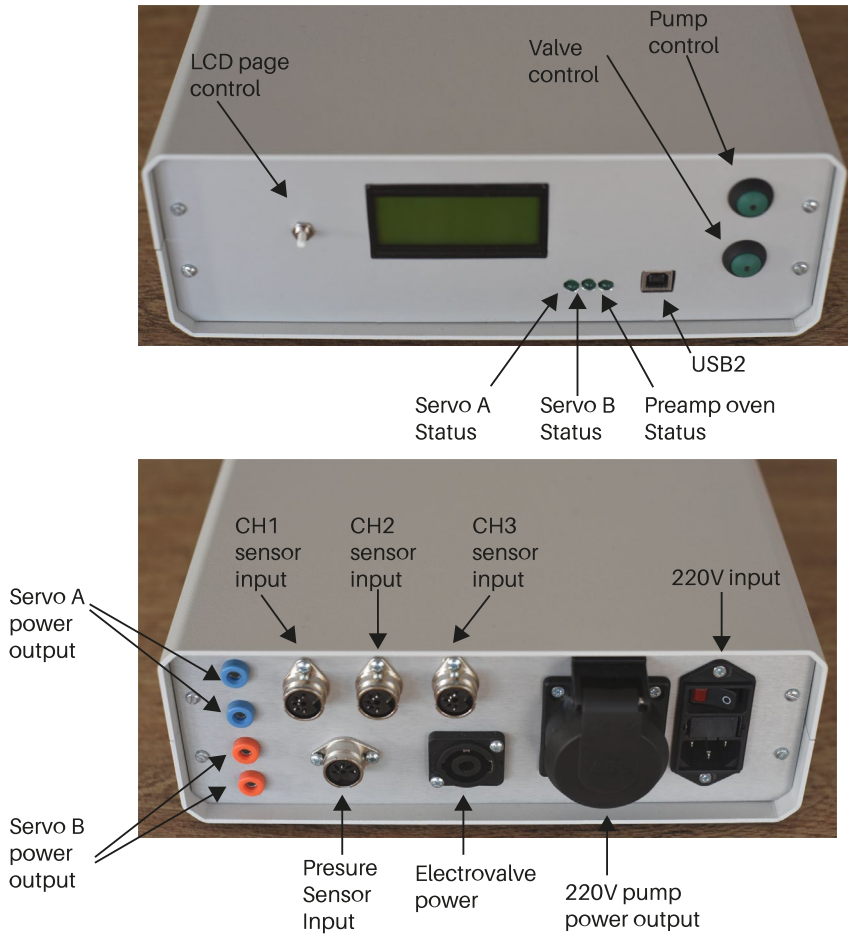


Figure 3. Front and Rear Panels.

screen with 11 information pages. Pages can be selected using the push button to the left of the LCD. The front panel also has three green LED indicators to show the state of the 3 temperature servo loops and a USB2 socket for attachment to the host PC. All other electric connections are made on the rear panel. Temperature channel 3 is not used in this application so has no external connector. Channels 1,2,4 are available on 3 DIN sockets whose pinout is shown in Figure 4. The sheath pin must not be electrically connected to anything other than the sheath of the 4 sensor wires. If it is allowed to short to any other structure it could result in an increase in noise in all channels.

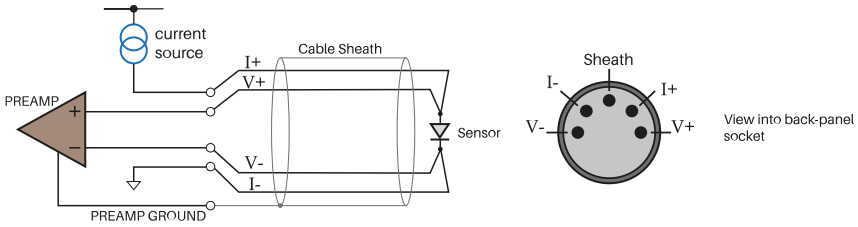


Figure 4. Temperature connector pinout.

The pinouts for the electro-valve and pressure sensor connectors are shown in Figure 5.

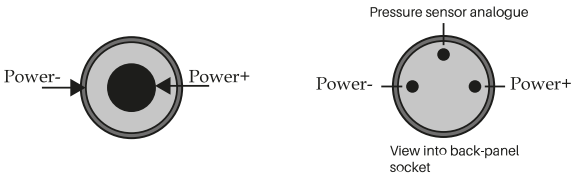


Figure 5. Electro-valve and pressure sensor connections.

The large hinge-lid power output socket is to power the vacuum pump. Next to it is the power entry socket which with integrated fuse holder. This is fitted with a 4A fuse and contains a spare. Finally there are two groups of servo power output connections. Servo A is color-coded blue and Servo B orange. These supply up to 15V @ 750mA to external servo heaters.

4 Servo Operation

The three servo loops are identical. Servo C is dedicated to internal regulation of the controller preamplifiers whereas Servos A,B are available to the user. A PI algorithm is used with clipping of the integral term to prevent wind-up effects. Input temperature signals can be digitally filtered to reduce read noise. It is also possible to filter the output power demand but this should not normally be necessary. The servos are shown schematically in Figure 6 together with the relevant commands required for configuration. The servos operates on a 1Hz update rate. Max servo power is 10W per channel.

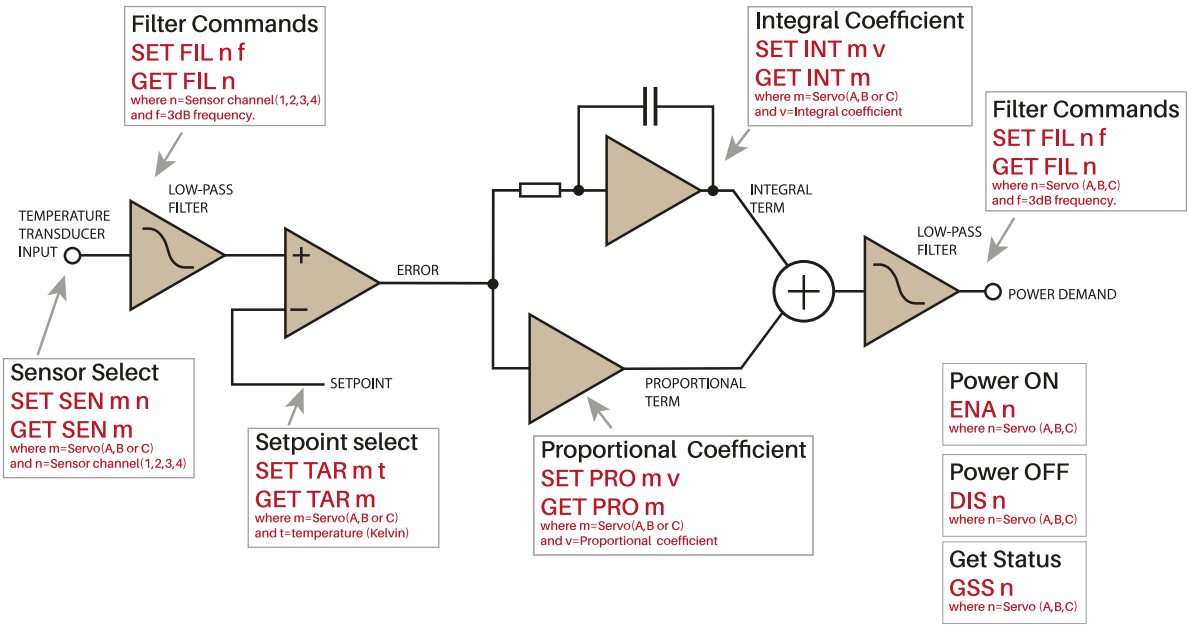


Figure 6. temperature servo operation and key configuration commands.

5 Valve and Pump Control

This version of the controller contains extra code to control a specific application in high-stability radial velocity (RV) spectroscopy. The hardware being controlled is described in this section.

5.1 External hardware.

A high stability optical reference source was mounted within a vacuum chamber. The walls of the chamber were maintained a few degrees above

ambient temperature using one of the controller servos. A second heater and temperature sensor were placed on the reference source. The two nested control loops then maintain a very stable reference source temperature (requirement is +/-15mK). The pressure within the vacuum chamber is kept $<5x10^{-3}$mBar using a vacuum pump and Pfeiffer EVC110M Electro-valve. Pressure is measured using a Thyracont VSP63MV gauge. The arrangement is shown schematically in Figure 7.

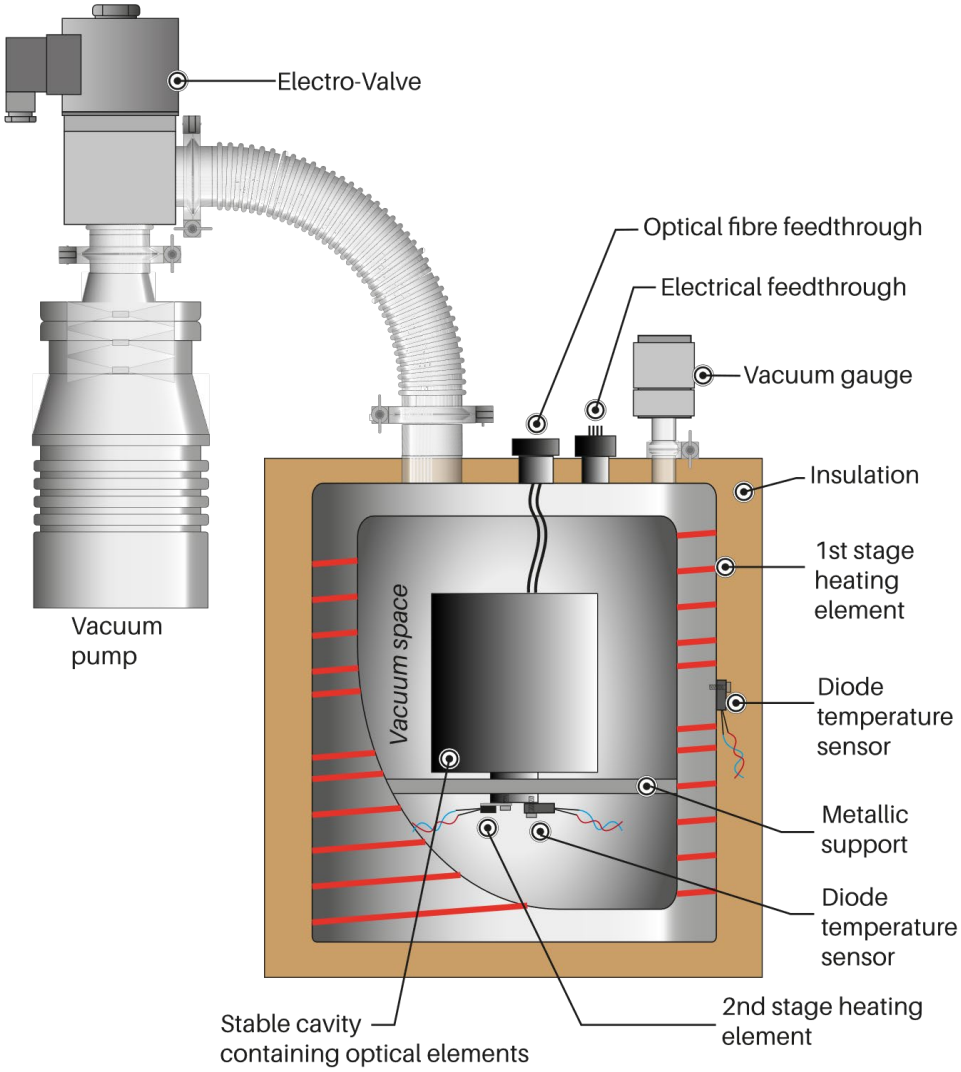


Figure 7. Reference source vacuum vessel.

5.2 Pump Modes

The pump and valve can always be manually controlled using the front panel buttons. These will toggle the state of either the pump or valve if depressed for more than 100ms.

There are three automatic pumping modes available. They are selected by the "SET PMO n" command . Manual mode corresponds to n=0. The other automatic modes are described below.

Threshold Pump Mode (n=1):

The pump will come on whenever the pressure exceeds a programmed threshold. The threshold is set by the "SET PTG". After a programmed delay("SET VDL" command) the valve will then open and the pump will run for programmed time ("SET PDU n" command, where n is time in seconds). If at the end of that time the pressure is still above the threshold then the pump will run for an additional n seconds. Otherwise the valve will shut and after a 1s delay the pump will stop.

Scheduled Pump Mode (n=2):

The pump will be activated at up to six programmed times during each 24hour cycle. Times are selected using the "SET PHR" command. Only hours can be specified, not minutes or seconds. At these times the pump will come on for the time programmed using the "SET PDU" command. The valve is controlled in the same way as the threshold mode with a programmed delay between pump coming on and valve opening and another short delay between valve closing and pump switching off.

Hourly Pump Mode (n=3):

The pump will come on at the top of each and every hour . Once again "SET VDL" and SET PDU" dictate valve delays and pump duration.

The controller can sense if the valve is not electrically connected. In this case the controller will revert to manual mode. Trying to select an automatic mode without the valve present will return an error.

Depressing the front panel pump or valve buttons will also revert the controller to manual mode. If this is happens when an automatic run is in progress then the valve will be snapped shut and the pump powered off 1s later.

The relative sequencing of pump and valve is shown in Figure 8.

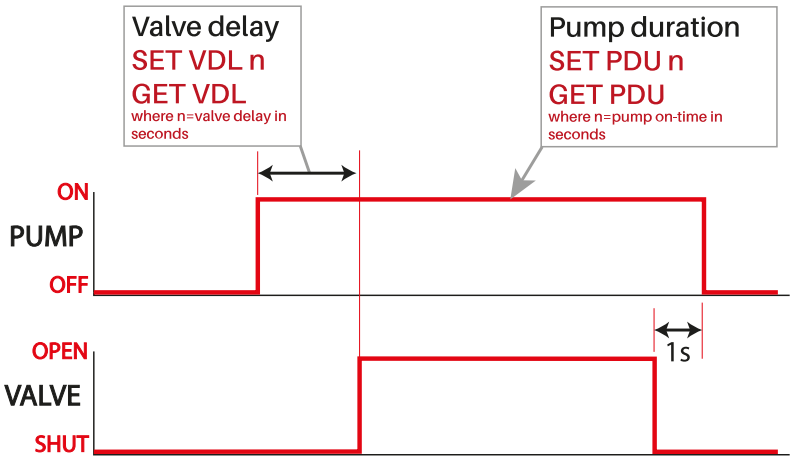


Figure 8. Sequencing of valve and pump when in an automatic pumping mode.

6 GUI Control

Commands can be sent to the controller from a serial terminal (such as "RealTerm") set to 115200 baud, 8 bits no parity. When attached to a host PC via a USB cable the controller will appear as a virtual coms port (no drivers need to be installed). Alternatively there is a Python-based GUI available on the product web site. The GUI is shown in Figure 9 with a more detailed description of its elements in Figure 10. The GUI uses standard packages that come preinstalled with the Anaconda Python environment. If the user wishes to write their own GUI then the product website also contain the "MKCon" Python class that identifies and then communicates with the controller via the USB interface.

7 Typical setup recipe

All commands are explained in detail in the appendix. The ones needed for the servo setup are also described in this section. First it is necessary to let the controller know which sensor channel is being used for the chosen servo channel. This is done using the "SET SEN n m" command where n= servo (A or B) and m= temperature sensor channel (1,2,3 or 4).

It is then necessary to program the desired target temperature using the "SET TAR n m" command, where n= servo channel we are using and m is the desired temperature in Kelvin.

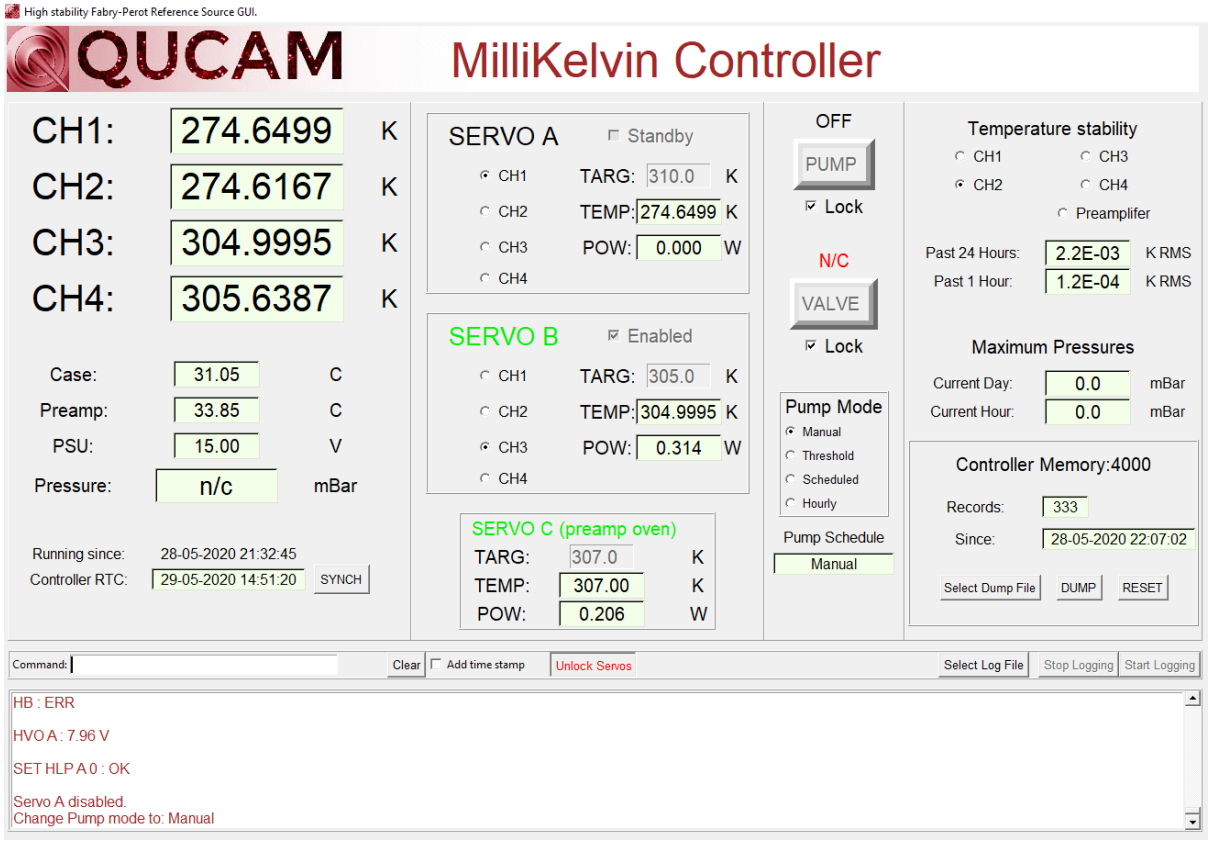


Figure 9. Controller GUI

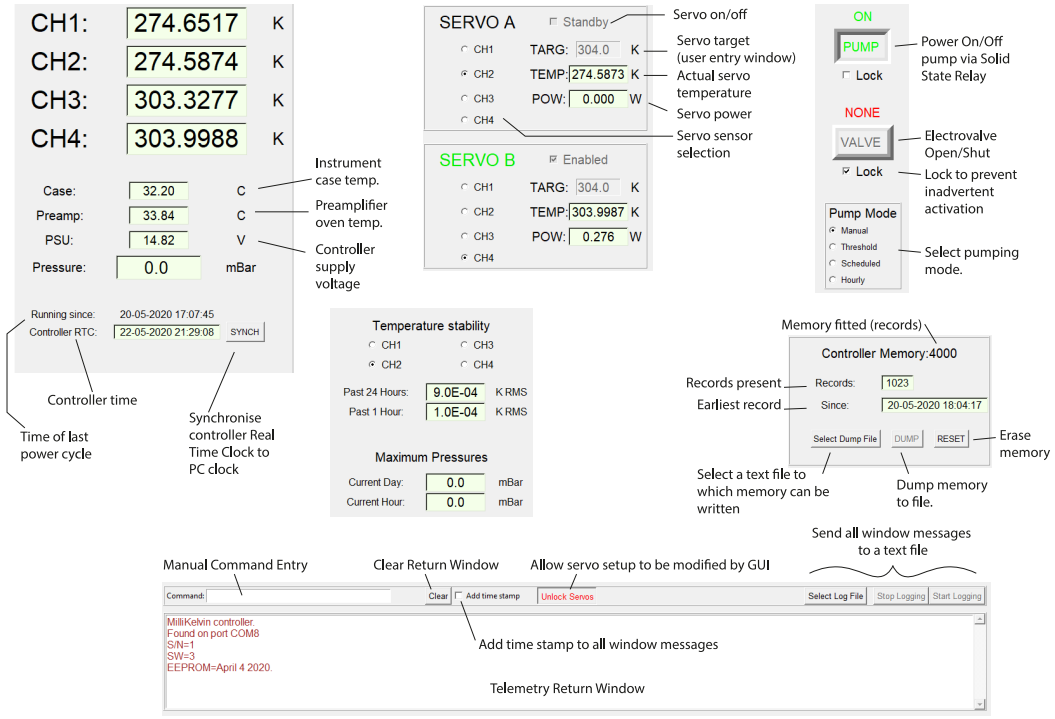


Figure 10. Controller GUI with its panels described.

The servo parameters may also need to be adjusted for the exact application although the default factory settings will probably give a good result. These parameters are the Integral and Proportional coefficients, the maximum temperature slope, the limit temperature (above which the heater is killed) and the heater power range high/low. This requires use of the "SET INT n m", "SET PRO n m", "SET SLO n m", "SET LIM n m" and "SET HLP n m" commands respectively.

Finally the servo can be activated using the "ENA n" command. If it is the first time the system has been used in the new configuration it should be confirmed that the heater is actually drawing current using the "HCU n" command (This of course assumes that our starting temperature is below the target temperature). Also confirm that the chosen temperature channel is actually registering a rise in temperature.

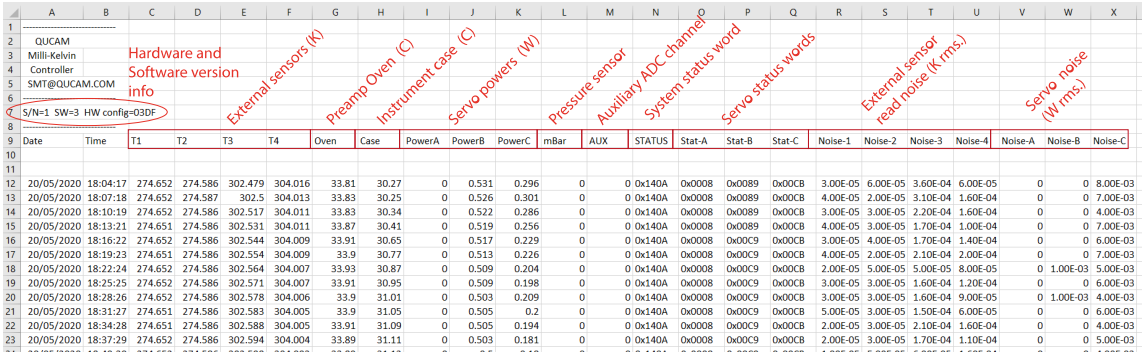
The configuration can be saved into the controllers non-volatile memory using the "SAV" command. The saved configuration is loaded after a power cycle or reset. This configuration includes whether servos A and B are enabled or not. This makes the system more robust to power outages. It is highly recommended that the SAV command be issued once the servo loops are all configured and running. Servo C which controls the preamp oven starts automatically a few seconds after power-on.

The Proportional and Integral coefficients need some further explanation. The Proportional coefficient effectively determines the range of error temperature over which the servo heater will go from zero to full power. P=1 means that the heater will start to decrease in power once the error falls below 1K. P=0.25 will means that the heater will start to decrease once the error falls below 4K. The integral term should initially be set to the reciprocal of the thermal time constant of the object being controlled. Any overshoot and ringing will then indicate that the Integral coefficient should be adjusted down.

8 Internal Data Storage

The controller has internal memory for 4000 telemetry records (optionally expandable to 6000) and automatically saves temperature, servo power and status information on a user-specified schedule. The record save interval can be programmed using the “SET RSI” command. Using a

record save interval of 1000s means the controller will store over 6 weeks of temperature and status history. The memory is circular and once full, previous data will be overwritten. Data can be read using the "DMP" command or the relevant GUI buttons. Transferred data is then saved in a comma-separated text file where it can easily be imported into Excel or viewed using the "MKplotter.py" tool (see product web site). Figure 11 shows the first few lines of data file with the column headers explained in more detail. Figure 12 shows a small example of the output of the plotter tool.



QUCAM
Milli-Kelvin
Controller
SMT@QUCAM.COM
S/N=1 SW=3 HW config=03DF

Hardware and Software version info

External sensors (K)
Preamp Oven (C)
Instrument case (C)
Servo powers (W)
Pressure sensor
Auxiliary ADC channel
System status word
Servo status words
External sensor read noise (K rms.)
Servo noise (W rms.)

Date	Time	T1	T2	T3	T4	Oven	Case	PowerA	PowerB	PowerC	mBar	AUX	STATUS	Stat-A	Stat-B	Stat-C	Noise-1	Noise-2	Noise-3	Noise-4	Noise-A	Noise-B	Noise-C
20/05/2020	18:04:17	274.652	274.586	302.479	304.016	33.81	30.27	0	0.531	0.296	0	0	0x140A	0x0008	0x0089	0x00CB	3.00E-05	6.00E-05	3.60E-04	6.00E-05	0	0	8.00E-03
20/05/2020	18:07:18	274.652	274.587	302.5	304.013	33.83	30.25	0	0.526	0.301	0	0	0x140A	0x0008	0x0089	0x00CB	4.00E-05	2.00E-05	3.10E-04	1.60E-04	0	0	7.00E-03
20/05/2020	18:10:19	274.652	274.586	302.517	304.011	33.83	30.34	0	0.522	0.286	0	0	0x140A	0x0008	0x0089	0x00CB	3.00E-05	3.00E-05	2.20E-04	1.60E-04	0	0	4.00E-03
20/05/2020	18:13:21	274.651	274.586	302.531	304.011	33.87	30.41	0	0.519	0.256	0	0	0x140A	0x0008	0x0089	0x00CB	4.00E-05	3.00E-05	1.70E-04	1.00E-04	0	0	7.00E-03
20/05/2020	18:16:22	274.652	274.586	302.544	304.009	33.91	30.65	0	0.517	0.229	0	0	0x140A	0x0008	0x00C9	0x00CB	3.00E-05	4.00E-05	1.70E-04	1.40E-04	0	0	6.00E-03
20/05/2020	18:19:23	274.651	274.586	302.554	304.009	33.9	30.77	0	0.513	0.226	0	0	0x140A	0x0008	0x00C9	0x00CB	4.00E-05	2.00E-05	2.10E-04	2.00E-04	0	0	7.00E-03
20/05/2020	18:22:24	274.652	274.586	302.564	304.007	33.93	30.87	0	0.509	0.204	0	0	0x140A	0x0008	0x00C9	0x00CB	2.00E-05	5.00E-05	5.00E-05	8.00E-05	0	1.00E-03	5.00E-03
20/05/2020	18:25:25	274.652	274.586	302.571	304.007	33.91	30.95	0	0.509	0.198	0	0	0x140A	0x0008	0x00C9	0x00CB	3.00E-05	3.00E-05	1.60E-04	1.20E-04	0	0	6.00E-03
20/05/2020	18:28:26	274.652	274.586	302.578	304.006	33.9	31.01	0	0.503	0.209	0	0	0x140A	0x0008	0x00C9	0x00CB	3.00E-05	3.00E-05	1.60E-04	9.00E-05	0	1.00E-03	4.00E-03
20/05/2020	18:31:27	274.651	274.586	302.583	304.005	33.9	31.05	0	0.505	0.2	0	0	0x140A	0x0008	0x00C9	0x00CB	5.00E-05	3.00E-05	1.50E-04	6.00E-05	0	0	6.00E-03
20/05/2020	18:34:28	274.651	274.586	302.588	304.005	33.91	31.09	0	0.505	0.194	0	0	0x140A	0x0008	0x00C9	0x00CB	2.00E-05	3.00E-05	2.10E-04	1.60E-04	0	0	4.00E-03
20/05/2020	18:37:29	274.652	274.586	302.594	304.004	33.89	31.11	0	0.503	0.181	0	0	0x140A	0x0008	0x00C9	0x00CB	2.00E-05	3.00E-05	1.70E-04	1.10E-04	0	0	5.00E-03

Figure 11. First few lines of a data record file.

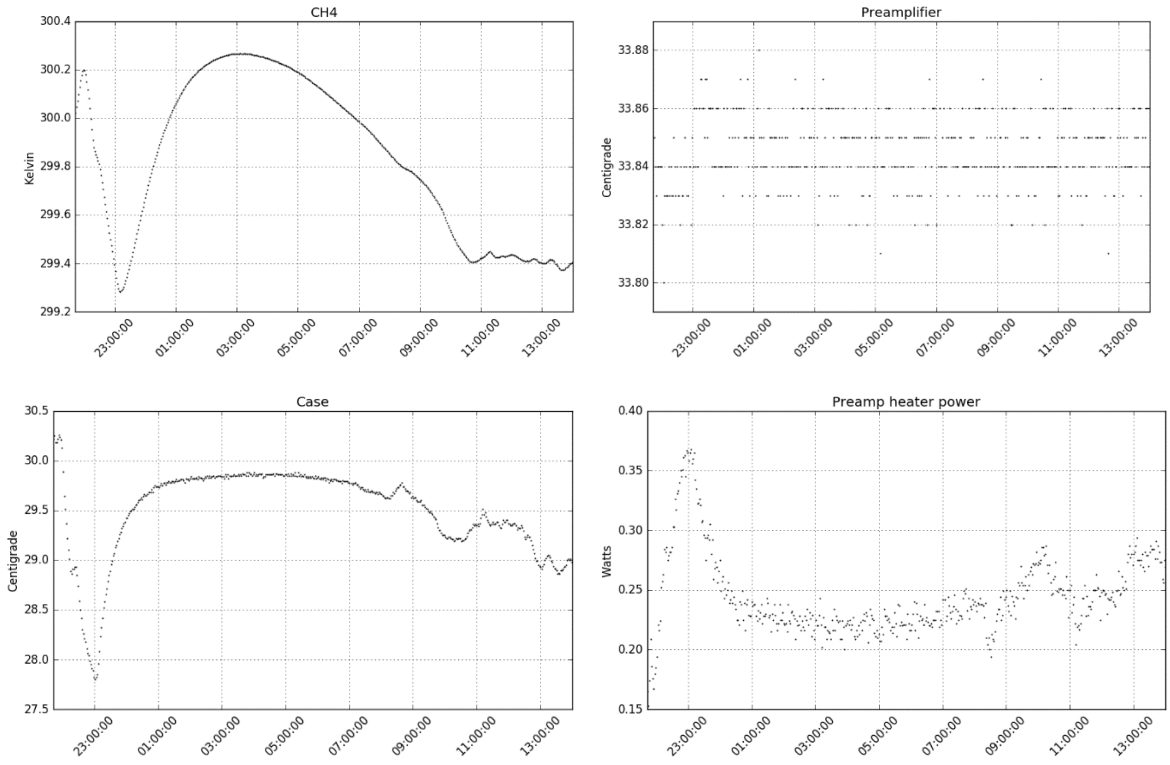


Figure 12. Example of output of MKplotter.py.

9 APPENDIX.

A. Range and Accuracy.

Three temperature input channels are configured for diode sensors. These are energised with 10μA precision current sources. The fourth channel is configured for use with a Pt100 RTD and is energised with a 1mA precision current source. Any combination of diode and RTD sensors can be factory configured. The characteristics of the default settings are shown below.

Table1. Temperature channel performance

Sensor channel	Sensor type	Minimum temperature	Maximum temperature	Read noise	24-hour stability
CH1	Diode	34K	321K	50μK	+/-1mK
CH2	Diode	34K	321K	50μK	+/-1mK
CH3	Diode	34K	321K	50μK	+/-1mK
CH4	Pt100	74K	321K	500μK	+/-3mK

B. Status Bits.

Each of the servos A,B and C has a status word (read by "GSS n") with the following encoding:

Table 2. Servo Status bits

Bit	Name	Function	Notes
0	ENABLE	Servo is enabled	
1	SENSORCHANbit0	These three bits select the sensor channel used.	000=ch1, 001=ch2 010=ch3, 011=ch4 101=preamp oven
2	SENSORCHANbit1		
3	SENSORCHANbit2		
4	OVERHEAT	Above limit temperature	Limit temp set by LIM command
5	Not used		
6	ATTEMPERATURE	Servos A,B within 0.01K of target. Servo C within 0.1K of target	Servo C used for preamp oven, lower precision acceptable.
7	Not used		
8	OVERCURRENT	Heater resistor has drawn >0.75A	Only cleared by re-enabling servo
9	LOWPOWER	1=low power mode 0=high power mode	

There is also a system status word (read by "SYS") mainly used for debugging. The word has the following encoding:

Table 3. System status bits

Bit	Name	Function	Notes
0	WDTERROR	System error. Watch-dog timer has timed out.	Requires service
1	LEDENABLE	LEDs are enabled.	
2	PALARM	Above trigger pressure	Trigger pressure set by " SET PTG"
3	SEEPROM	Is programmed with Calibration data	
4	stateSSI1	1 means solid state relay 1 is activated	Used to power pump
5	stateSSI2	1 means solid state relay 2 is activated	Unused, future expansion
6	stateEV0	These two bits indicate electrovalve state.	00=not attached 01=shut 10=open
7	stateEV1		
8	PumpModebit0	Pump mode selected	00=Manual, 01=Threshold 10=Scheduled 11=Hourly
9	PumpModebit1		
10	HardwareOK	All expected hardware on I2C bus is present.	
11	OVEN	Preamp oven within 0.1K of target	
12	USB	USB cable attached	
13	BATFAIL	RTC lost time during last power cycle	Battery needs replacing

C. LCD pages.

The 11 LCD pages are shown in Figure 13. Page numbers can be incremented using the button next to the screen.

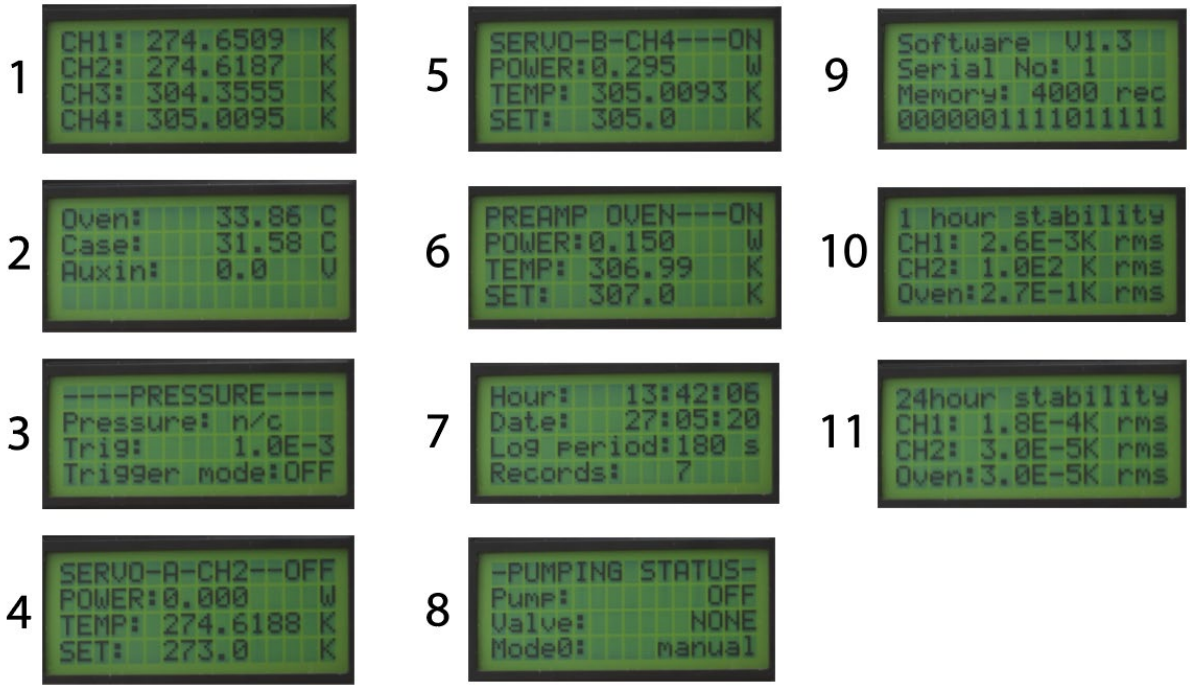


Figure 13. LCD screens

D. Command Dictionary.

Commands must be terminated by a carriage return and can be upper or lower case. The controller responds to any command by echoing each character of the command and returning the requested data or sending an “OK” or an “ERR”. This will then be followed by a Carriage Return, a Line Feed and the prompt (“>”). Preceding a command with a “#” causes the controller not to echo the command and not to return a prompt. This will be more convenient if the controller is operated through a program on the host PC since it will reduce the communications load.

Some commands are only available if the controller is in Development Mode. This mode can be set using the “DEV” command. At midnight the controller returns to non-dev mode. Dev mode adds some protection against inadvertent changes to the preamplifier oven and other parameters that only need to be set up during commissioning.

Float parameters can be entered in normal or scientific format.

The command vocabulary is described below:

General system information/status----------

RID	Read controller ID. Useful for locating the controller's COMS port. If preceded by "#" the string "Q_MKC" is returned.
SAV	Save controller setup to non-volatile memory.
SYS	Returns a binary string with system status bits.
GSS n	Returns a binary string with status bits for servo n.
LED n	Enable or disable front panel LEDS and LCD backlight. n=1 switches on light sources, n=0 switches them off.
LCD n	Select LCD info page n (these are listed in the appendix. Screen blanked by n=0.
DEV	Select DEV mode.
SET TIM d m y h m s	Set the controller real time clock.
(GET) TIM	Read the controller RTC. "GET" is optional.
PSU	Read Power supply voltage.
SUT	Read the time and date when the controller was last powered on.
AUX	Read auxiliary analogue input (not implemented).

Controller data memory----------

SET RSI n	Set the record save interval to n seconds. Setting n=0 stops any data recording.
GET RSI	Read back the record save interval.
MEM	Read amount of record memory installed.
RWF	Read back the "Record Wrapped Flag". If this flag is 1 then it means that the circular record memory is full and that the oldest record is being overwritten by the newest record.
RST	Reset the record memory.
RECS	Read the number of records in memory.
DMP	Dump all of the records via the serial interface in a single block. A header is automatically included. This can take up to 50s if the memory is full. During this time the controller will not respond to any commands. Any received character will abort the dump.
DM20 n	Dump a block of 20 records starting at record n. This is the maximum number of records that can be transmitted at a time without blocking additional commands.
DLR	Read just the last record that was written to memory.
HED	Read the record data header (includes column headings and controller ID).
FRT	Read time stamp of earliest record in memory.

Temperature sensors----------

KEL n	Read the temperature in Kelvin of channel n. n=1,2,3,4 refer to the external temperature channels, n=5 refers to the preamplifier temperature and n=6 refers to the instrument case temperature. Returns "n/c" if sensor not connected.
NOI n	Read RMS temperature noise in sensor channel n over the last 10s. Alternatively if n=A,B or C the command returns the servo noise in Watts RMS.
SET MAP n m	Map the external sensor on channel n to the calibration curve m. Calibration curves present are: m=1: Pt100 RTD m=2: IN4148 diode m=3: Lakeshore DT470 diode This command is only available in Dev mode and when issued, any active servos will be disabled. Other curves can be supplied. This version of the controller is configured to allow diode sensors to be used on CH1,2,3 and Pt100 sensors on CH4. Other combinations are possible upon request.
GET MAP n	Read back calibration curve number used by external sensor on channel n.
STH n	Read RMS temperature noise in sensor n over the last hour (not available for case sensor).
STD	Read RMS temperature noise in sensor n over the last day (not available for case sensor).

Pumping Control----------

PRE	Read Thyracont VSP63MV pressure sensor. Returns pressure in mBar.
VLV n	Activate electro-valve, where n="open" or "shut".
PMP n	Activate vacuum pump, where n="on" or "off".
SET PTG n	Set pressure trigger to n mBar. This is the pressure at which the pump activates when in threshold mode.
GET PTG	Get pressure trigger value in mBar.
SET PDU n	Set pump duration to n seconds. This is the time for which the pump remains on after being triggered automatically.
GET PDU	Get pump duration in seconds.
SET PHR m,n,o,p,q,r	Set hours (24hour system) at which pump will be triggered in Scheduled mode. Up to six hours can be entered (separated by spaces or commas).
GET PHR	Get hours at which pump is scheduled to come on.
SET PMO n	Set pump mode to n. n=0 is Manual mode (operated by front panel buttons, GUI buttons or using the "PMP" command). n=1 is Threshold mode. n=2 is Scheduled mode. n=3 is Hourly mode. Changing pump mode will immediately close the valve and then stop the pump 1s later. Pump mode cannot be changed from manual if there is no valve connected.

GET PMO	Get pump mode.
SET VDL n	Set Valve delay to n seconds. This is the time between the pump starting and the valve opening in one of the automatic pumping modes. Gives pump enough time to evacuate vacuum hose and reach operating speed.
GET VDL	Get Valve delay.
PTR	Read pump time remaining in seconds when pumping in one of the automatic modes.
MPH	Read maximum pressure in current hour.
MPD	Read maximum pressure in current day.

Servo loops----------

ENA n	Enable servo n (A,B or C). Since servo C controls the preamp oven it can only be enabled/disabled in Dev mode.
DIS n	Disable servo n (A,B or C). Since servo C controls the preamp oven it can only be enabled/disabled in Dev mode.
SET TAR n m	Set target temperature of servo n (1,2,3,4) to value m. Units are Kelvin.
GET TAR n	Get target temperature in Kelvin of servo n.
SET SEN n m	Select sensor m (1,2,3,4) for use with servo n (A or B). Servo should be disabled prior to changing sensor (except in Dev mode).
GET SEN n	Get sensor used for servo n.
SET PRO n m	Set proportional coefficient of servo n (A,B,C) to value m. Servo C is used by the preamplifier oven

	and can only be modified in Dev mode. m must be between 0 and 15.
GET PRO n	Get proportional coefficient for servo n .
SET INT n m	Set integral coefficient of servo n (A,B,C) to value m. Servo C is used by the preamplifier oven and can only be modified in Dev mode. m must be between 1e-5 and 0.05.
GET INT n	Get integral coefficient for servo n.
SET SLO n m	Set maximum slope of servo n to value m. Units are Kelvin per minute. Range is 0 to 100. Servo C is used by the preamplifier oven and can only be modified in Dev mode.
GET SLO n	Get maximum slope for servo n.
SET HLP n m	Set heater low-power range for servo n (A,B,C). m=1 to select low power, 0 for high power. Selecting low power restricts the servo heater voltage to approx. 8.0V. In the default high-power setting, the maximum heater voltage is 14.9V.
GET HLP n	Get power-range setting for servo n.
SET LIM n m	Set temperature limit of servo n (A or B) to value m Kelvin. Both servos A and B are disabled if the chosen sensor for servo n exceeds this temperature.
GET LIM n	Get temperature limit for servo n.
SET FLW n m	Set flash window of servo n (A,B or C) to value m Kelvin. The front panel LEDs will flash if the servo temperature error exceeds m.
GET FLW n	Get flash window for servo n.

SET FIL n m	Apply single pole low-pass IIR digital filter with 3dB point = m Hertz. If n=1,2,3 or 4 it applies to the temperature input channels. If n=5 it applies to the preamplifier temperature sensor. If n=6 it applies to the instrument case temperature. If n=A,B or C it applies to the power demand output of the servos. Must be below 0.5Hz.
GET FIL n	Get digital filter frequency currently used by channel n.
HVO n	Read the voltage on servo output n.
HCU n	Read the current output by servo n (Amps).
HPO n	Read the power output by servo n (Watts).
GST n	Get current temperature of servo . Returns "n/c" if sensor has been disconnected.

E. Product website

http://www.qucam.com/products.html

F. Notes on stability of temperature input channels

Characterising the temperature stability of a temperature controller over long periods really requires the use of an absolute temperature reference. One example would be a stirred ice-slush bath. We have been unable to get good results from this method. Instead we have used highly stable reference resistors in place of the temperature sensors and recorded the stability of the temperature telemetry over periods of 100 hours to obtain the data in Table 1. Further details of this method can be found here: http://www.qucam.com/assets/notesonstability.pdf.

----------End of Document----------