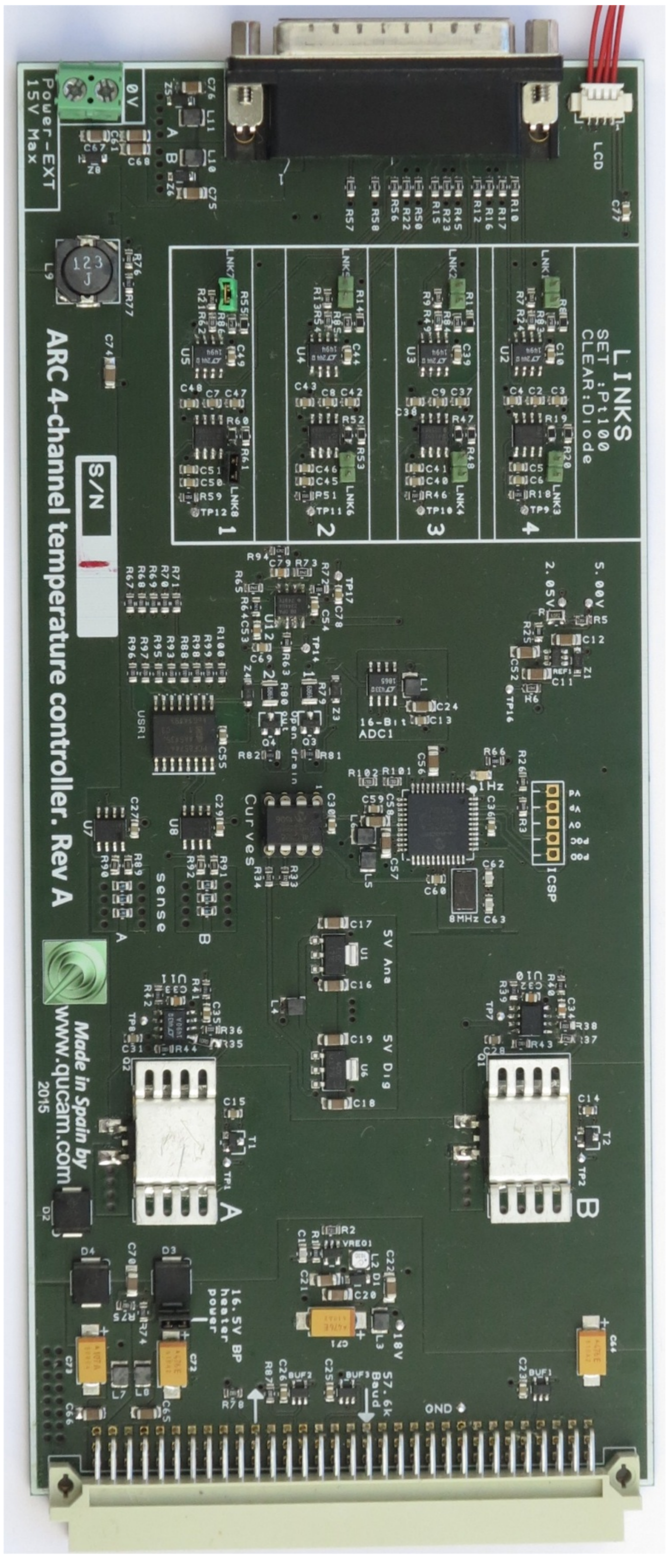
ARC temperature controller board

Manual



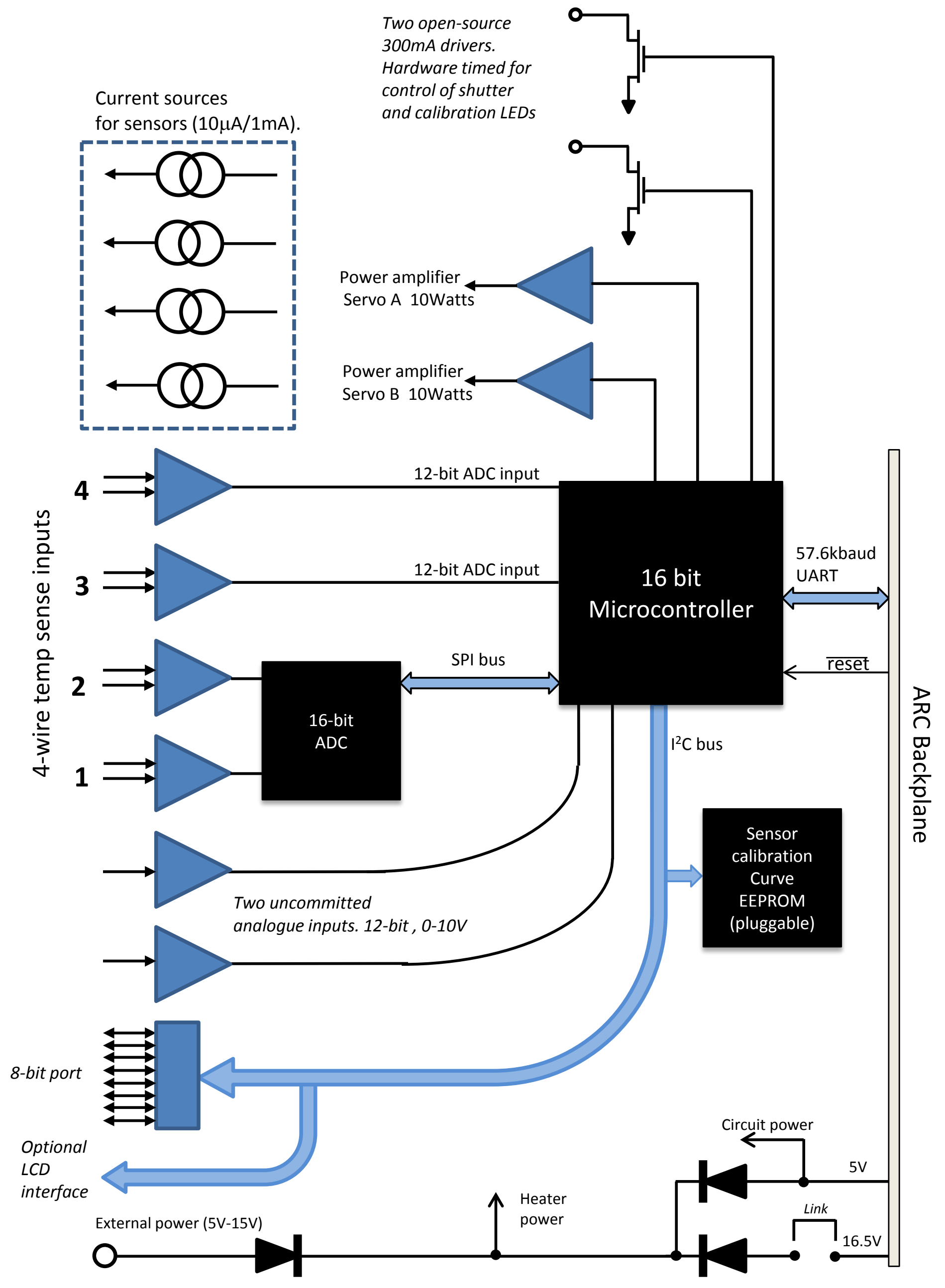
Simon Tulloch May 2015

This board is intended for use in the Astronomical Research Cameras (ARC) Gen III controller. Almost any scientific imaging system will require a temperature controller and also a vacuum gauge. The board combines these functions in a low cost solution. System design is also simplified since the Data Acquisition System has direct access to temperature and pressure telemetry. Also rack-space is reduced and cabling simplified.

Simon Tulloch

smt@qucam.com

1. Overview of main features.



Two open-source 300mA drivers. Hardware timed for control of shutter and calibration LEDs
Current sources for sensors (10µA/1mA).
Power amplifier Servo A 10Watts
Power amplifier Servo B 10Watts
4-wire temp sense inputs
4
3
2
1
12-bit ADC input
12-bit ADC input
16 bit Microcontroller
57.6kbaud UART
reset
SPI bus
16-bit ADC
I²C bus
ARC Backplane
Sensor calibration Curve EEPROM (pluggable)
Two uncommitted analogue inputs. 12-bit , 0-10V
8-bit port
Optional LCD interface
Circuit power
5V
Link
16.5V
Heater power
External power (5V-15V)

Figure 1. Overview

1.1.Temperature Inputs

The board has four temperature input channels. Each can be configured for either Pt100 (RTD) or Diode sensors by means of on-board links. These links are necessary both to change the sensor excitation current (10μA for diodes and 1mA for Pt100) but also to change the preamplifier gain (Pt100s have a much lower sensitivity and therefore need more gain). If links are changed then it is also necessary to reconfigure the software on the board since the micro has no way of reading the link settings. This is done using the select sensor command "SET SEN m n" (see Section 7). Instructions for setting/clearing links can be seen on the PCB silk screen. Two of the temperature channels (1 and 2) are intended as main servo inputs. They are read using a precision 16-bit ADC (Nat Semi LT1765) and have better noise performance than channels 3 and 4 which are read using the 12-bit converter in the board's micro-controller. These latter two channels are intended as auxiliary inputs for where lower precision is acceptable.

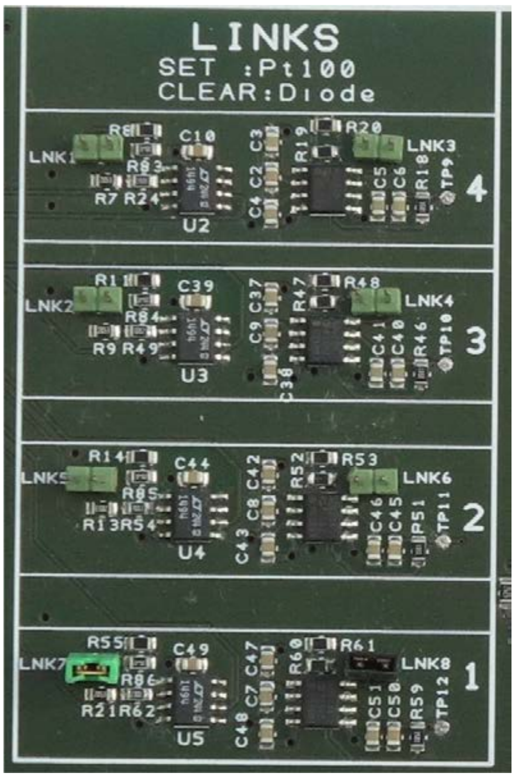


Figure 2. Link settings

There are also two engineering sensors on the board used to monitor the temperature of the servo power amplifiers. They are used to automatically protect against damage from overheating if airflow over the heat-sinks should be interrupted. If the servos are disabled then these sensors will measure the internal temperature of the ARC controller which could be a useful diagnostic, for example to measure the effectiveness of glycol loops attached to the ARC chassis. Note that when operated at full power the servo amplifiers actually dissipate very little power on the board. The worst case is when both servos are outputting 50% of their maximum voltage in which case the on-board dissipation can rise to 2.4W per channel.

1.2. Analogue Inputs.

There are two general purpose analogue inputs on the board which are digitised to 12-bit precision. The input range is 0-12V, the input resistance is 100kΩ and the bandwidth is 2Hz. The inputs can be read directly in units of volts. Additionally, one channel can be used as the input from either a Pfeiffer PKR251 or MKS970 vacuum gauge. The board automatically calculates the pressure, depending on the sensor that has been selected, and the value can be read directly in units of milli-Bar.

1.3.Open-drain digital outputs

The board has two open-drain digital outputs. They can sink up to 300mA of current at a maximum voltage of 20V. They are diode protected to allow switching of inductive loads such as relays. Their states can manually be set hi or lo or alternatively they can be fired as mono-stables with programmable hardware timers defining the on-time. These could be useful for operating calibration LEDs and camera shutters. One of the open-drain outputs has a timer with a resolution of 100μs, the other has a resolution of 100ms. The two timers are independent.

1.4. 8-bit parallel port

This uses the PCF8574A port expander IC. The port can be read or written to although it has no data direction register. It is described in the data sheet as "quasi-bidirectional". It is possible to write to the output port or read from the input port via a single register. If a bit is required as an input then it must have '1' written to it. Writing a '1' to a bit will only pull the output pin hi through a low current source allowing it to be used as an input. See Section 6.4. for more details. The port has 1k series resistors mainly for protection purposes although it also allows direct driving of indicator LEDs. Since the port can sink much more current than it can source, LEDs are best driven with their Anodes connected to 5V and their Cathodes connected to the relevant port pin. Writing '0' to the port pin then turns on the LED with at least 2mA of drive.

1.5. Indicator/Alarm outputs

There are a further 5 digital outputs under control of the micro and not directly available to the user. These also have series resistors and can directly drive indicator LEDs on the ARC front panel. Two of these are used for signalling the status of the two Servos. If flashing then they indicate that a servo is active but not yet at temperature. If fully on they indicate that the servo error is less than 1K. Each servo has a user selectable alarm temperature. Two outputs indicate if these temperatures are exceeded. The final output is a Pressure alarm that indicates if the pressure as measured by one of the analogue inputs exceeds a user-programmed limit. The board has an internal LED that gives a 1Hz "heartbeat". This was useful during development. Any LED may be undesirable in an observatory environment so all can be disabled with a single command.

1.6. LCD Interface.

The board can drive an optional LCD display on the front panel of the ARC. Either a small 8x2 character or larger 16x4 character display can be used. Eight separate info pages can be displayed. The pages can be selected under software or stepped through by pressing a small button on the display itself.

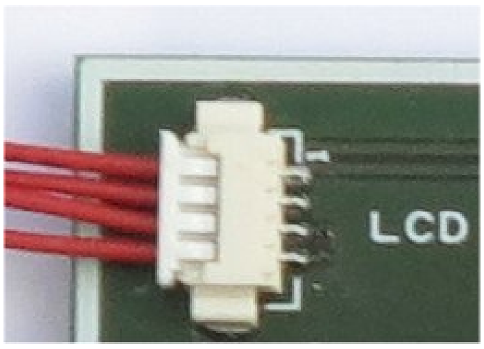


Figure 3. LCD connector

1.7. Use of an external power source.

The board has an external power input connector that must be used if high heater powers are required. The ARC backplane powers the board's analogue and digital circuitry from its 5V bus. Some power can be drawn from this bus to supply the servos but it will be quite limited. For example if 20Ω servo heaters are used (the minimum recommended) then the maximum power available will be around 1W per servo. This might be enough for certain applications and has the advantage that the 5V bus is always available. For higher power there are two options. The first is to draw power from the ARC backplane 16.5V bus. This requires setting a link on the ARCTEMP board. In tests up to 250mA of current has been drawn from this bus without disturbing the ARC. This means that a single servo could have up to 3.6W of power which is sufficient for most CCD camera requirements. In this case a higher value servo heater resistor, approximately 50Ω, should be used to avoid overloading the backplane. Using the 16.5V backplane bus, however, has a serious drawback. If the timing card is reset or the timing card does not have its software loaded then the 16.5V bus drops to 0V. This will obviously disturb the detector temperature. This loss of 16.5V power at reset can, however, be remedied by a change of ARC Timing card boot code. If higher heater powers are needed it is recommended that an external power supply be used. This can have any value between 5V and 15V. Use of higher voltages can cause overheating of the board and servos will automatically be disabled if the input voltage exceeds 15.5V. Note that if the external power input is used, the 16.5V backplane power link must be removed or the servos will continue drawing current from the backplane. With an external power source the board will be able to simultaneously give 9.6W of heater power on both servos. To get this power requires the use of 20Ω servo heaters and an external PSU able to give at least 1.5Amps (double this would be better).

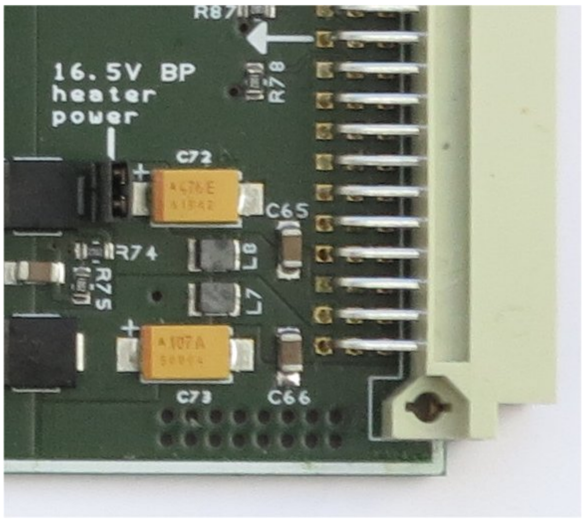


Figure 4. Backplane power ink

1.8. Protection features.

Shorts in the servo power outputs or use of excessively low heater resistances could cause damage. To reduce this risk the servo power amplifiers have their temperatures measured every second and if either exceeds 325K both servos are powered off. Additionally the servo current is directly measured using small (100mΩ) on-board sense resistors. If either servo draws more than 700mA it is automatically powered off within 250ms. Another potential risk is servoing on a broken temperature sensor, especially one that when disconnected or shorted, erroneously indicates a lower temperature. This could cause the servo to then fry the camera head. Test have been done with both diode and RTD sensors where wires were removed one at a time to see the effect of indicated temperature. In all cases the temperature input channels all "failed hot" i.e. indicated temperatures drift to their maximum values within a few seconds. In addition the user can program limit temperatures above which servos will be disabled but this of course does not deal with sensor hardware faults.

1.9. Pluggable sensor-curve EEPROM.

This has four standard calibration look-up tables (LUT) that translate sensor voltage into temperature. There is room for three more LUTs should the user have another type of sensor. In this case a new EEPROM can be supplied and plugged into the board. The EEPROM comes with the following response curves programmed.

Curve number	Type	Range minimum	Range maximum
1	Pt100 RTD	73K	383K
2	DT670 Diode	45K	400K
3	S900 Diode	45K	380K
4	1N4148 diode	70K	360K

Table 1. Sensor calibration curves

1.10. Interface to ARC Controller.

The board mounts in a standard ARC slot (any slot will do). As well as drawing 5V power and optional 16.5V power (depending on the link setting described in Section 1.7.) the board also uses three 5V digital lines to communicate with the ARC timing card. One of these is the #ARC_RESET line (backplane pin B4) and the Serial Communications interface (SCI) RX and TX lines (backplane pins B18 and B19). The #ARC_RESET line allows the ARCTEMP card to be hard-reset by pressing the button on the ARC PSU or under command from the Timing card. The default ARC-supplied timing code asserts the #ARC_RESET line low each time the timing code is downloaded and executed. This could be undesirable if extreme detector temperature stability is required and the user may opt to edit the timing card code to prevent this happening. The default timing card will anyway need to be edited to change the SCI interface speed from 391kBaud to 57.6kBaud. More details on how to do this in Section 4.1.

The ARCTEMP board imitates the ARC50 Utility board. It has the same bus address of 3 and uses the same communications protocol whereby all commands and data packets are preceded by a 24-bit word indicating the length of the packet, the source of the command/data and the destination for the command/data. The board responds equally to commands originating in the host PC or ARC timing card. It cannot be used in conjunction with the ARC50 since they both use the same serial communications pins on the backplane.

2. Temperature sensor selection.

2.1. Diode temperature sensors.

The forward voltage of any p-n junction is strongly dependent on temperature and weakly dependent on current. Passing a constant current through a diode and then measuring the forward voltage can then be used in conjunction with a look-up table or polynomial fit to read the temperature. A signal of about -2.5mV/K is obtained with a 10μA current. The forward voltage decreases with temperature. One of the most widely used diode sensors is the Lakeshore DT670. It is good from 500K down to around 25K. Below 25K it can still be used but is very non-linear. The low currents used mean that cable resistance produces a negligible error and that sensor self heating is minimised. Calibrated diode sensors are very expensive. Lakeshore charge 150€ just for delivery. The sensor itself costs at least 150€ more depending on calibration accuracy.

2.2. Pt100 RTD sensors.

Resistive Temperature Detectors (RTDs) vary their resistance with temperature. The most popular type is the Pt100 which has a resistance of 100Ω at 0C. It increases its resistance with temperature. It is composed of a fine platinum wire generally encased in a ceramic package. The change in resistance is approximately linear and equal to 0.385 Ω/K. To measure the change in resistance a constant current of 1mA is passed and the voltage drop measured. This gives a tiny voltage change of around 385μV/K. Use of higher currents would cause errors from self heating. Cable resistance can also strongly affect the accuracy unless 4-wire connections are used (see next section). Pt100 sensors are good from 1000K down to 20K at which point they start to lose sensitivity. They are easily available at low cost from electronic suppliers such as Farnell although the packaging is often inconvenient for use in scientific cameras where some sort of screw-down mount is desirable.

2.3. Four-wire connection.

In order to remove the effects of cable resistance on Pt100 sensors it is important to make a four-wire connection to the sensor. The energising current is carried by one twisted pair cable and the sense voltage carried by second pair. Loss of accuracy due to cable resistance adding to the sensor resistance is thus removed entirely. The use of twisted pairs also reduces the effect of external noise pickup. The arrangement is shown below in Figure 5. Four wire connection is less useful for a diode sensor where cable resistance is of little importance. The use of twisted pairs is nevertheless important if the sensor is not to function as an antenna.

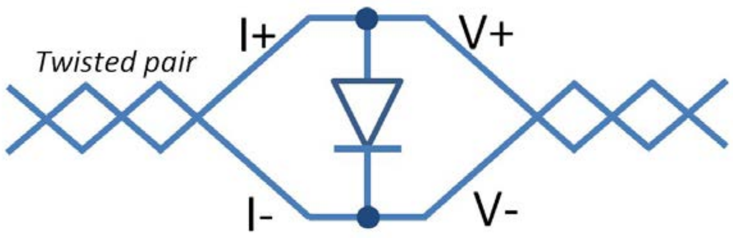


Figure 5. Four-wire sensor connection

The ARCTEMP board can be used with either 2 or 4-wire sensor connections. If two wire connection is chosen it is important to connect I+ to V+ and I- to V- at the 44-way D connector i.e. I+ and V+ as well as I- and V- *must* be connected if not at the sensor then at the controller-end of the cabling.

2.4. Optional low-cost sensors from QUCAM.

With scientific CCD cameras the absolute temperature of the imaging detector needs only to be measured to a few degrees of accuracy. Often the temperature sensor is not even mounted on the CCD but on some other structure within the focal plane. What is important, however, is temperature stability especially in the case of EMCCDs. Lakeshore sensors are probably over-specified for most CCD applications and experiments have been done with cheaper alternatives. The first alternative tested was a BYV25F-600 power diode mounted in a TO-220 package. It was chosen fairly randomly from the Farnell catalogue, the big attraction being its ease of mounting and low cost. Unfortunately the diode was found to be very non-linear and it was impossible to fit a good polynomial given the limited number of calibration points taken.

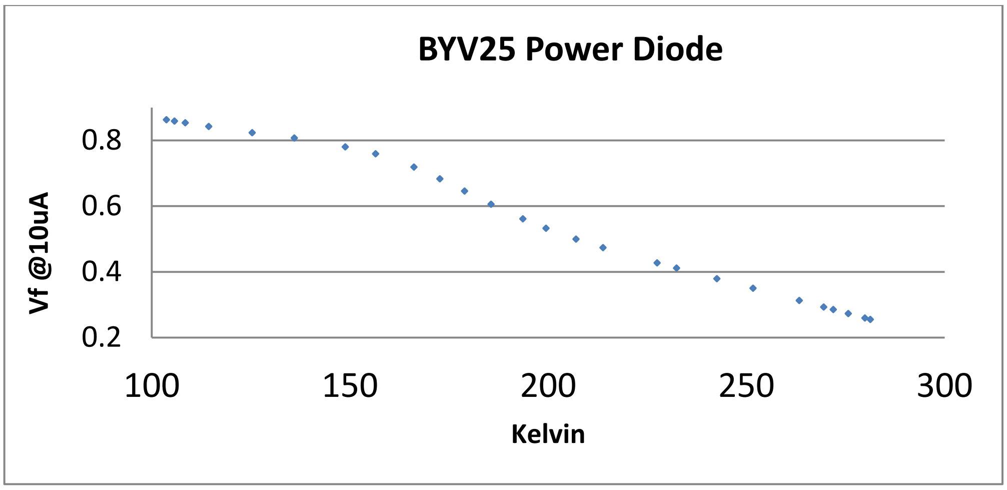


Figure 6. Investigation of use of power diode as thermal sensor.

The next experiment was with a standard 1N4148 small signal diode epoxied into a metal tag with an M3 clearance screw hole. This gave a surprisingly linear response and encouraged further work. Figure 7 shows the result.

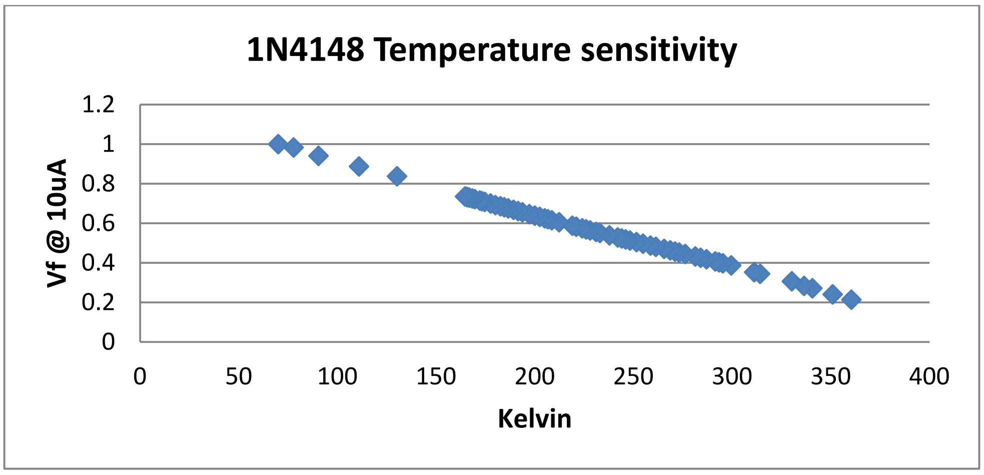


Figure 7 calibration curve for 1N4148 diode.

Ease of mounting is of key importance when using sensors in a CCD camera. The glass package diode is not really convenient. It was therefore decided to make a PCB-based sensor using a surface mount (SMT) diode. The use of PCB technology makes it very easy to produce a screw-mountable sensor with convenient solder attachment points and a low-emissivity gold-plated finish. A panel of 49 sensors was manufactured and the diodes attached in a reflow oven. The result is shown in Figure 8. A second panel was made for use with SMT Pt100 sensors. Each sensor measures 14 x 9 x 2.1mm. The mounting hole is for an M3 screw. The sensor connections are electrically isolated from the mechanical mount whilst the PCB layout ensures good thermal coupling from sensor to mounting surface and bolt.

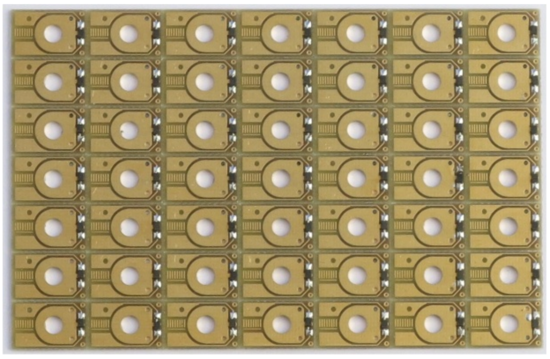


Figure 8. Manufactured panel of diode sensors.

The sensors were probe tested whilst the panel was sitting on a sturdy aluminium plate that minimised thermal gradients. A histogram was made of the indicated temperature for 45 of the sensors. This is shown in Figure 9 and indicates a spread of about +/-1K.

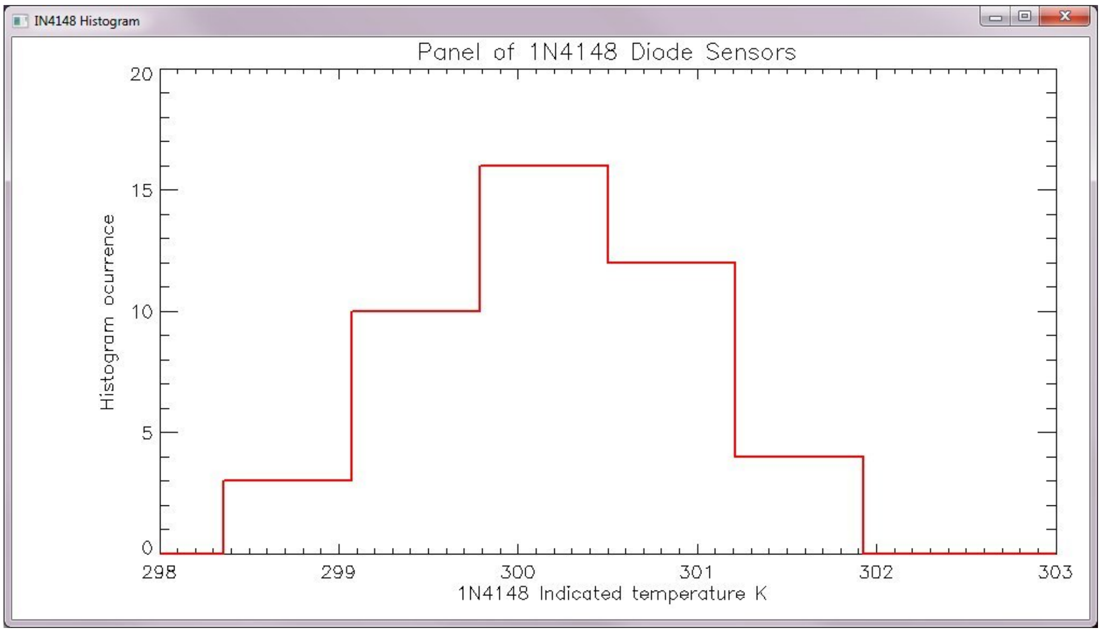


Figure 9. Spread in sensor response at room temperature.

Seven randomly chosen sensors were then dipped in liquid nitrogen to see if the scatter in signal was worse at lower temperatures. The scatter about the mean was found to be very similar between room temperature and LN2 temperatures (77K).

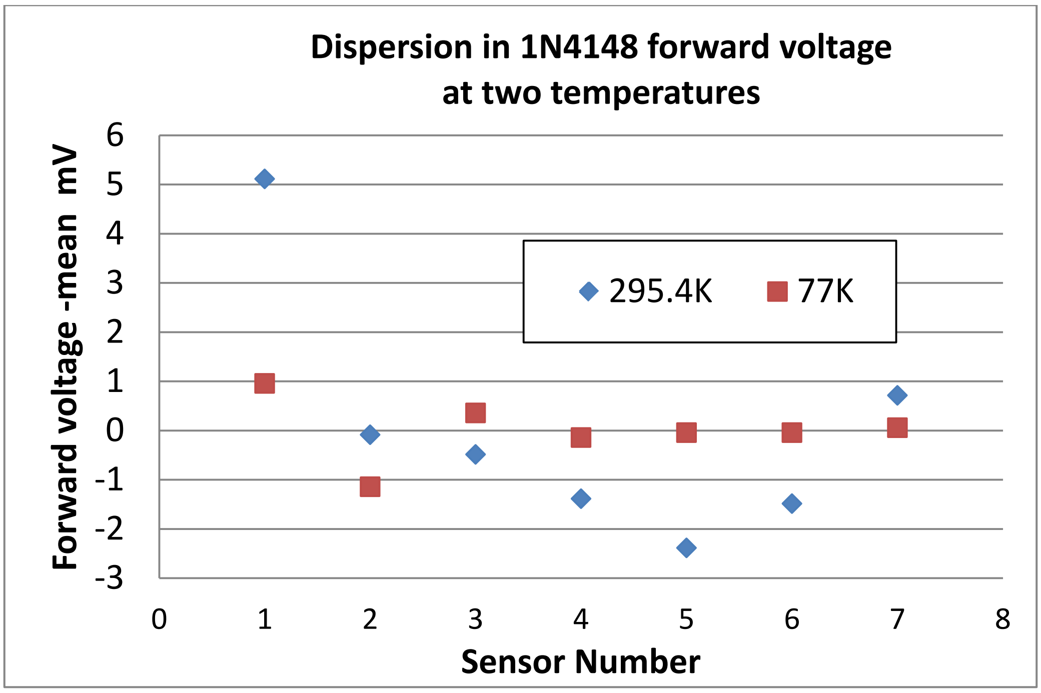


Figure 10. Spread in sensor response at room temperature and 77K.

One of the panel sensors was snapped off (the panel was V-scored in the factory) and wires were attached. It was then calibrated by LN2 immersion followed by a slow warm up. Its calibration curve is included pre-programmed in the on-board EEPROM. One of these sensors is included free with the board.

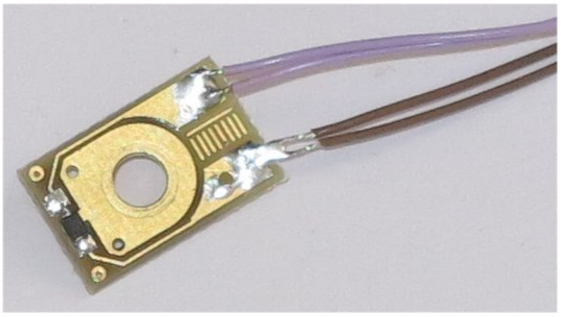


Figure 11. 1N4148 diode sensor

3. Details of the temperature control servo loops.

3.1. PI control.

Proportional-Integral (PI) control is a standard way of servoing temperature. It works as follows. The error between the actual temperature and the set point temperature is calculated. This error is multiplied by a proportional coefficient to yield the proportional term. The error is then multiplied by the integral coefficient and added to an integrator. The integrator and proportional term are then added together to yield a power demand level. A proportional term is not enough to reach our desired set point since at this point the heater power will be zero. Instead the final temperature will end up slightly below our set point. The inclusion of an integral term takes out this small error and allows us to servo precisely at the set point. The proportional and integral terms may need tuning for each specific application. If either is too large, oscillation can occur. If too small then the system can take too long to reach the set point.

3.2. Avoiding integral term wind-up.

If our CCD camera takes a long time to cool down then it can cause the Integral term in our PI controller to accumulate a large value. This can extend the time taken for the servo to reach equilibrium. The same can happen if we start out well below the set point and the camera takes a long time to warm up. This effect is called Integral term wind-up. It can be avoided by only switching on the integral term once we are already close to the set point. the controller algorithm uses a window that extends *below* the set point and only when the servo temperature enters this window does the Integral term switch on. If the temperature then leaves the window it makes no difference, the Integral term remains on. The width of the window is factory set to 10K. It can be changed using the "SET IWI n" command.

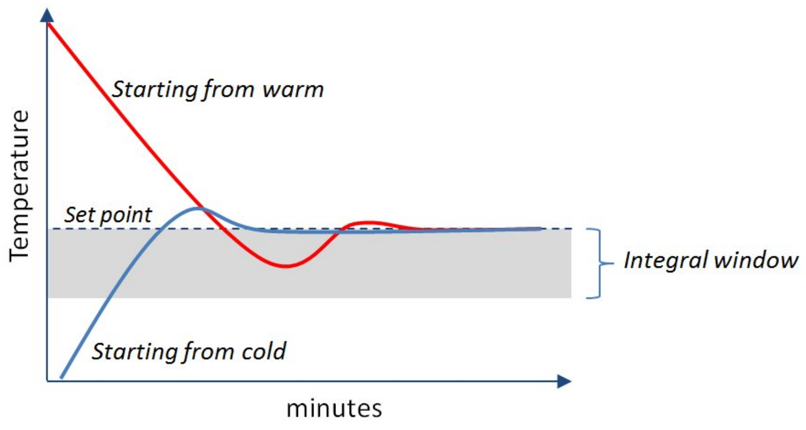


Figure 12. Integral window

3.3. Temperature slope control.

The servos can be made ramp slowly to their set points using the " SET SLO" command. This could be useful if the thermal mass of the CCD camera focal plane is very low. In such cases immediate application of high servo heater powers could damage the head.

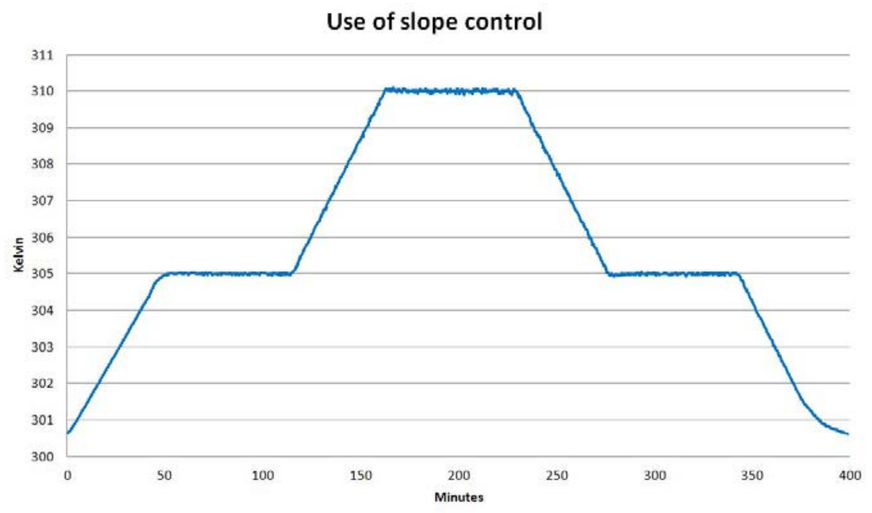


Figure 13. Use of slope control with rate set to 0.1K/minute.

3.4. Suggested PI control parameters.

The proportional and integral coefficients of each loop can be independently adjusted using the "SET PRO" and "SET INT" commands. The proportional coefficient has a range of 0-2000, the integral range is 0-1000. As part of the board testing a heatsink was fitted with a 50Ω servo heater and a Pt100 sensor. The heatsink consisted of 80g of aluminium with a thermal resistance to ambient of 7.5Kelvin/Watt. An external 15V power source was used. A series of data logging runs were done in which the heatsink was alternatively

3.5. Reducing sensor noise using digital filters.

The four temperature input channels have their voltages sampled at 1Hz. The voltage is turned into a temperature using the calibration curves and made available to the servo algorithm and user. It is also possible to pass the calculated temperatures through a single pole digital filter. The filter has switchable bandwidths of 0.3Hz,0.1Hz and 0.03Hz using the "SET FIL n m" command. In this command n=temperature channel of interest and m=filter used.

Filter number (m)	**Filter 3dB point**
0	Filter off
1	0.3Hz
2	0.1Hz
3	0.03Hz

Table 2. Digital filter parameters.

It can be used to reduce sensor noise but should be used with care if the thermal time constant of the focal plane being controlled is similar to the time constant of the filter. In this case instability could result.

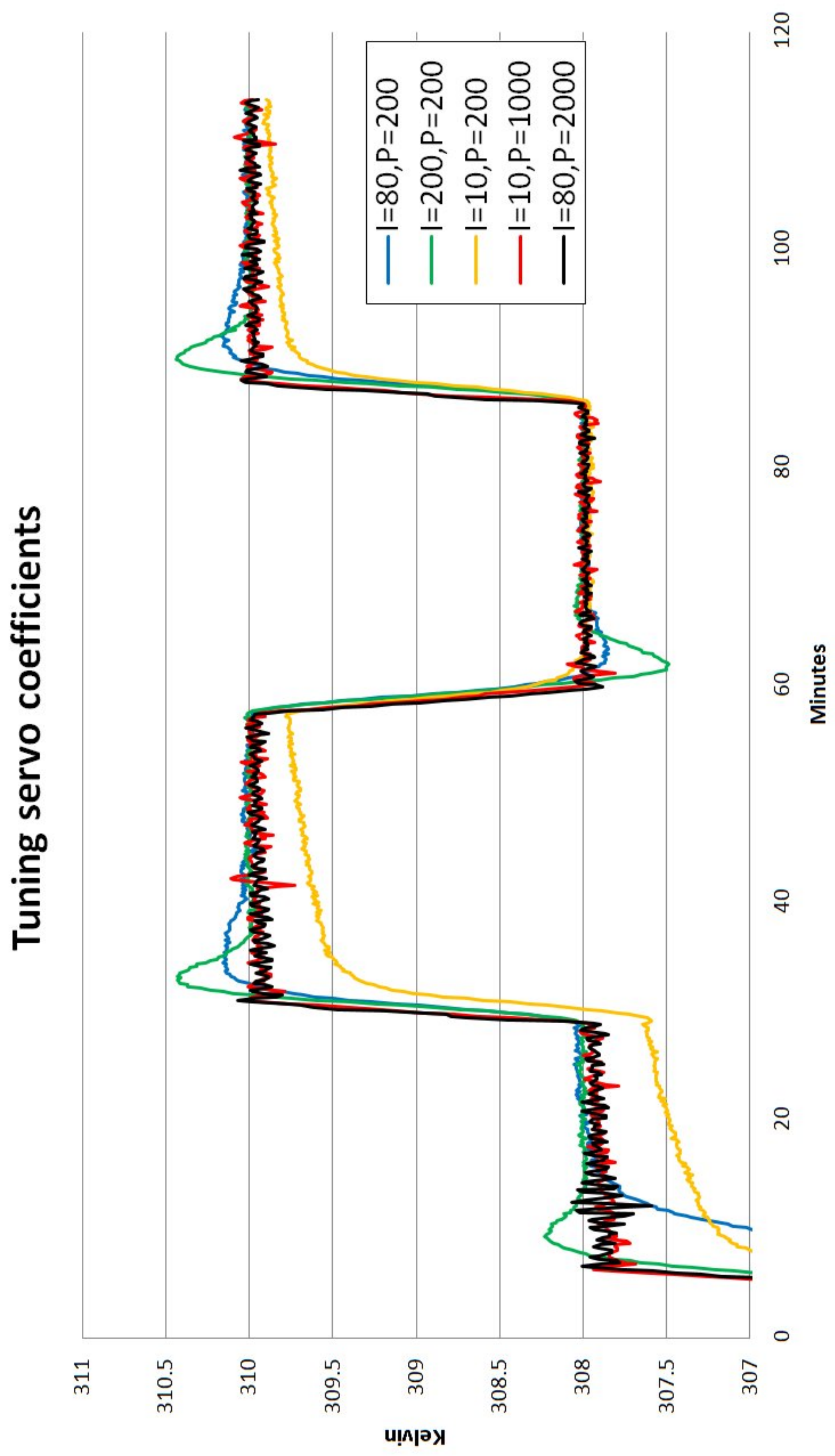


Figure 14. Experiments with various values of P and I servo coefficients.

servoed at 308K and then 310K. In each run the P and I coefficients were slightly changed. The result is shown in Figure 14. The optimum values for this particular thermal mass/heater combination were I=80,P=200. It can be seen from the yellow trace that a lower value for I meant that the set point was not reached within the available time. Setting I to 200 caused excessive overshoot. Use of higher values for P caused oscillation about the set point. Although not shown in this graph, high values for P also cause the heater voltage to swing widely as the heater demand value becomes increasingly affected by sensor noise.

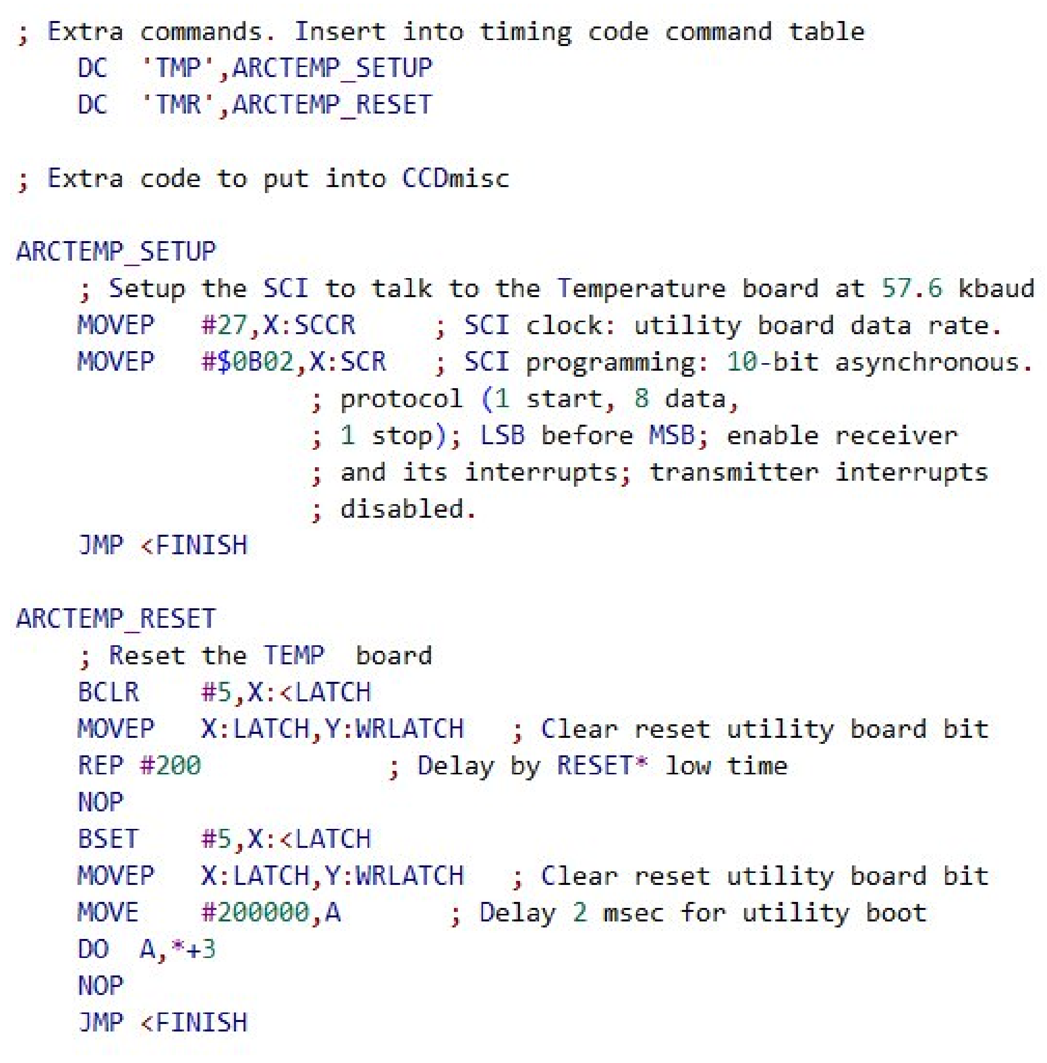
3.6. Suggested servo heater resistors.

This very much depends on which power source is being used. In the case of an external 15V PSU the heater resistors should not be below 20Ω. If they are then excess current will be detected and the servos will be disabled. If the 16.5V backplane power link is set (see Section 1.7) then a higher value resistor must be used to avoid overloading the ARC PSU. In tests the maximum current that was drawn from the 16.5V rail was 220mA. Higher currents are untested. If 16.5V rail power is used then a 68Ω resistor is recommended with only one servo channel active. This will give up to 3.5W of heater power. If the 16.5V link is clear and there is no external power source then the servos will by default draw 5V bus power (the various power sources are "diode or-ed" on the board). Use of 5V power limits severely the maximum servo output although a well designed CCD head shouldn't require too much heater power anyway. If a 20Ω resistor is used in combination with 5V power we can still obtain about 900mW of power

4. Software interface to ARC controller.

4.1. Modifications required to TIM card code.

The default timing card code supplied by ARC must be modified so as to slow down the communication speed on the SCI interface. This requires two extra commands to be inserted into the timing card command table. "TMP" sets up the SCI interface and "TMR" is an optional command that can be used to hard reset the temperature control card. The most convenient place to add the command definition code is in the 'CCDmisc.asm' file. The extra code is shown below:



```
; Extra commands. Insert into timing code command table
    DC   'TMP',ARCTEMP_SETUP
    DC   'TMR',ARCTEMP_RESET

; Extra code to put into CCDmisc

ARCTEMP_SETUP
    ; Setup the SCI to talk to the Temperature board at 57.6 kbaud
    MOVEP   #27,X:SCCR     ; SCI clock: utility board data rate.
    MOVEP   #$0B02,X:SCR   ; SCI programming: 10-bit asynchronous.
                   ; protocol (1 start, 8 data,
                   ; 1 stop); LSB before MSB; enable receiver
                   ; and its interrupts; transmitter interrupts
                   ; disabled.
    JMP <FINISH

ARCTEMP_RESET
    ; Reset the TEMP  board
    BCLR    #5,X:<LATCH
    MOVEP   X:LATCH,Y:WRLATCH   ; Clear reset utility board bit
    REP #200              ; Delay by RESET* low time
    NOP
    BSET    #5,X:<LATCH
    MOVEP   X:LATCH,Y:WRLATCH   ; Clear reset utility board bit
    MOVE    #200000,A       ; Delay 2 msec for utility boot
    DO  A,*+3
    NOP
    JMP <FINISH
```

4.2. Communicating via the ARC API.

The ARC controller comes with an API consisting of library functions that cover all aspects of controller operations. As a third party board the API contains no functions intended directly for the temperature controller. Instead the general "Command" library function is used to send commands and parameters and to receive responses. All parameters are 24-bit integers and the return value is also a 24-bit integer. The "board_id" must be set to 3. The API function call format is as follows:

CameraAPI.Command(board_id, command, arg1, arg2, arg3, arg4);

Further details can be found on the ARC website www.astro-cam.com.

4.3. Encoding real number parameters as integers.

Since the ARC interface deals in 24-bit integers any real numbers must be encoded. Real numbers are required for the following parameters:

-Volts

-Amps

-Temperature in Kelvin

-Slopes in Kelvin per minute

-Pressure in mBar

The first four of these can simply be converted into integers with units of mill-Volts, milli-Amps, milli-Kelvin per minute and milli-Kelvin respectively prior to transmission. A 24-bit integer easily gives the required range. The pressure, however, has too large a range of values to be encoded linearly in 24 bits so a custom fixed point format is used. This is described below.

The pressure is encoded in the lower two bytes of the 24-bit ARC parameter. The higher byte is the mantissa and the lower byte is the exponent. It is sufficient to give a 1% precision across the full range of pressure values.

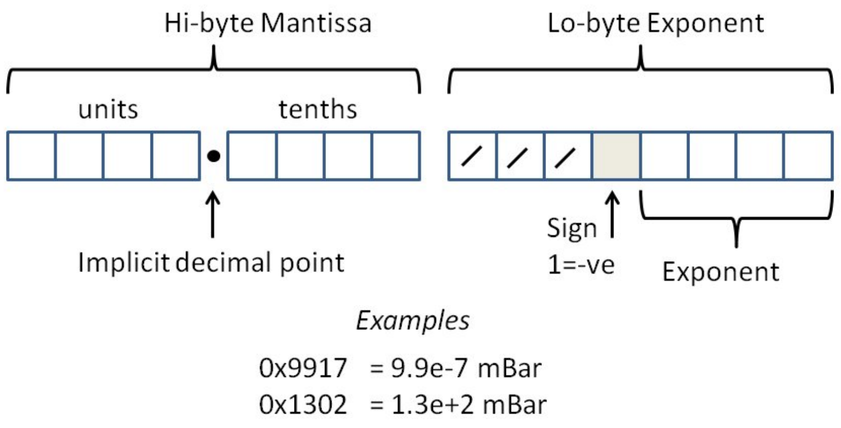


Figure 15. Encoding pressure data

This encoding is only used by the "PRE", "SET PAL" and "GET PAL" commands.

4.4. Java script GUI.

The ARC OWL data acquisition program allows the user to write their own Java scripts which access the API functions. Scripts are loaded and run using a control at the bottom right of the OWL window (see Figure 16).

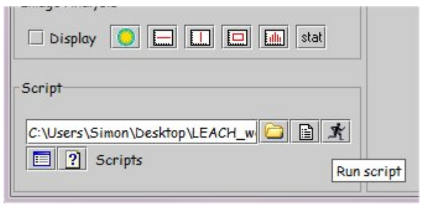


Figure 16. Loading and running scripts from within OWL..

A GUI has been written to allow the user to display temperature and servo data as well as configure and activate the servos. The GUI has a total of 5 pages, 3 of which are shown in Figures 17, 18 and 19. The GUI updates at 1Hz but pauses operation if an image readout is in underway. It gives access to most but not all commands. The script is available on the Qucam website (www.qucam.com).

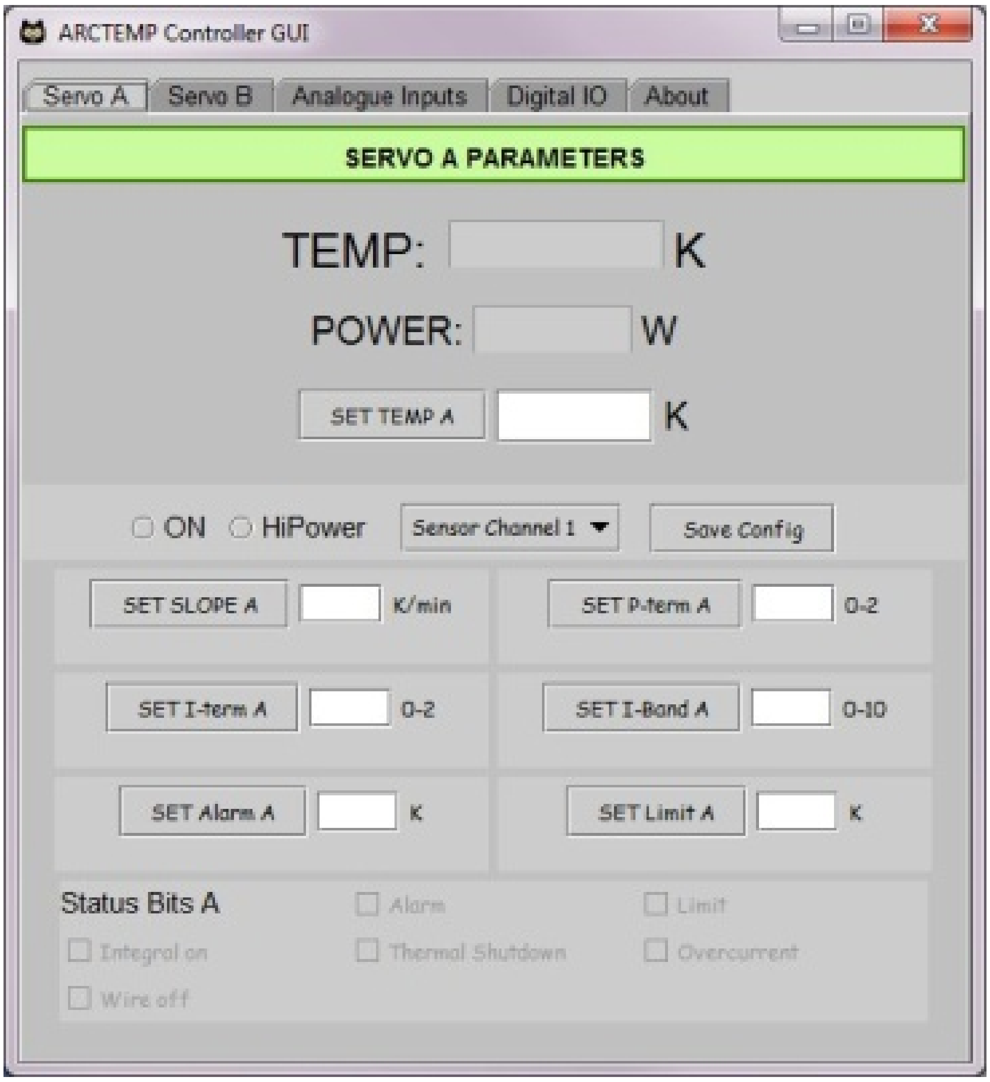


Figure 17. The ATCTEMP controller GUI Servo A control page.

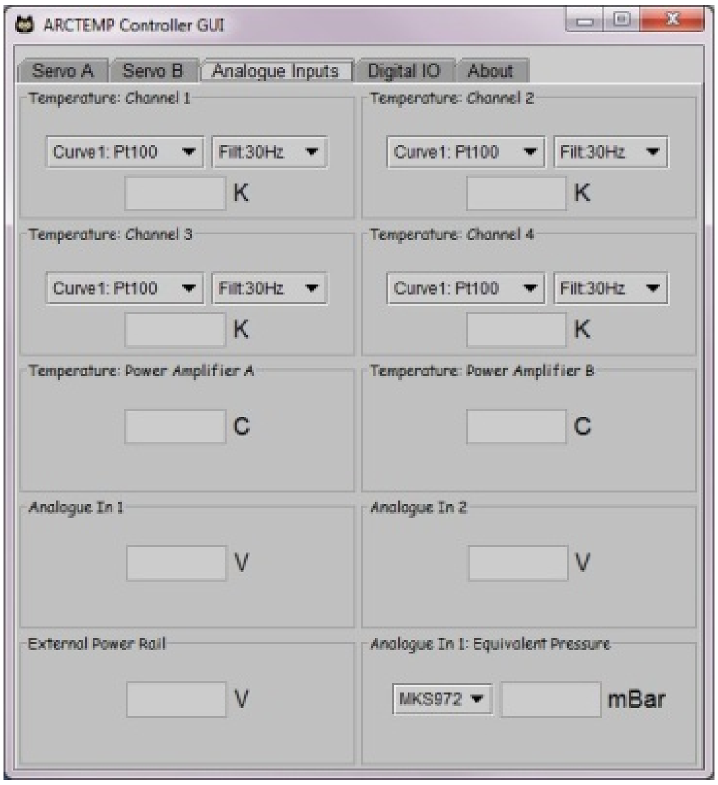


Figure 18. The ATCTEMP controller GUI Analogue input page.

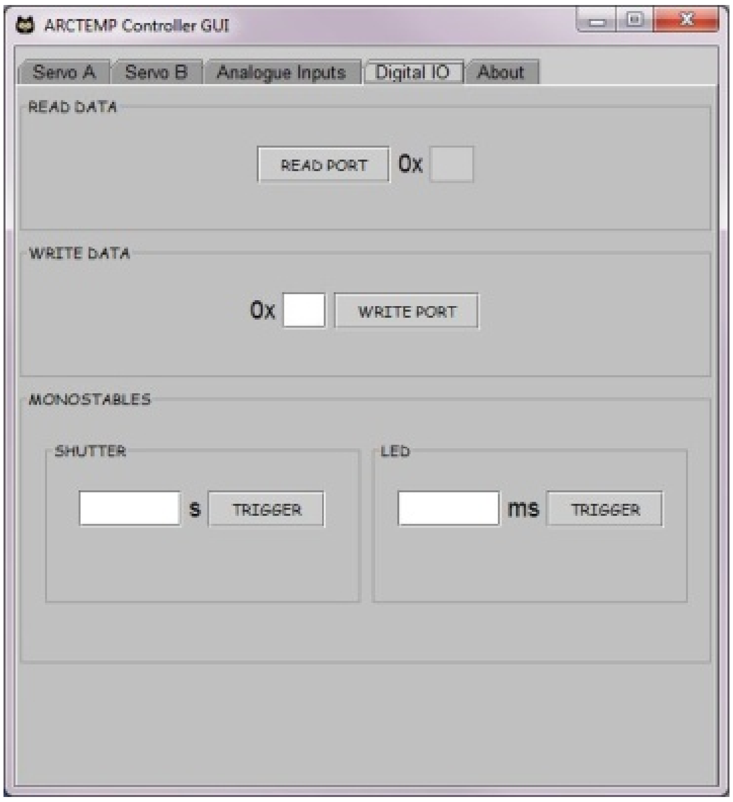


Figure 19. The ATCTEMP controller GUI Digital IO page

5. LCD display.

This is an optional extra. Two sizes are available, shown in Figure 20. The larger is really only suitable for a 12-slot controller chassis since it occupies a large fraction of the front panel of a standard 6-slot chassis. Both displays have backlights activated by a push button. A second button is held down to cycle through up to 8 display pages. The displayed page can also be selected by software. Page 0 corresponds to a blank display. The LCD connects directly to the on-board micro via an I^2C interface. This interface has very limited protection and so should not be used with displays mounted external to the ARC chassis.

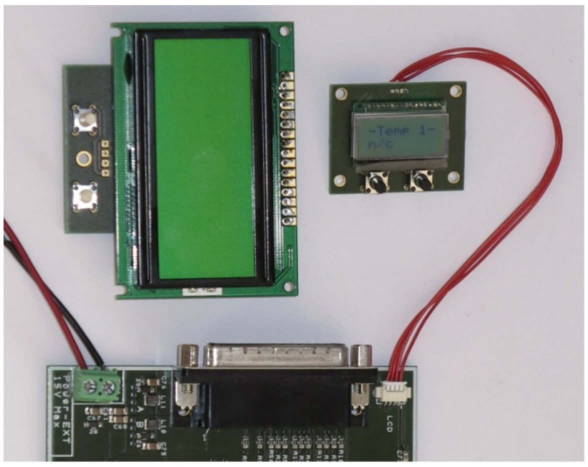


Figure 20 The two optional LCD units

5.1. Small 8x2 character display.

This can only display limited information. It measures 37 x30mm. The information displayed is as follows:

Page	Information
0	blank
1	Temp channel 1
2	Temp channel 2
3	Temp channel 3
4	Temp channel 4
5	Pressure
6	Power servo A
7	Power servo B

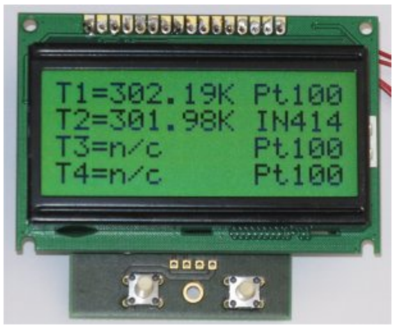
Table 3. Small LCD info page content.

5.2. Large 16x4 character display.

This displays more in depth information. It measures 72 x70mm. The information displayed is as follows:

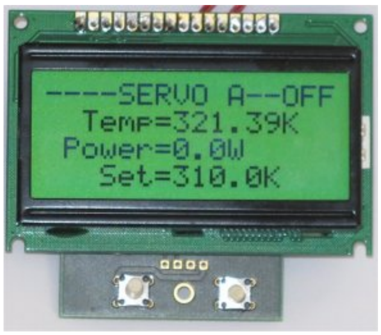
Page	Information
0	blank
1	Temp channels 1,2,3,4. Kelvin and sensor type
2	Servo A information: Temperature, set point, power
3	Servo A information: Temperature, set point, power
4	Analogue input voltages, external power, pressure.
5	System, ServoA and ,ServoB status bits (in that order)
6	About
7	Servo power amplifier temperatures

Table 4. Large LCD info page content.



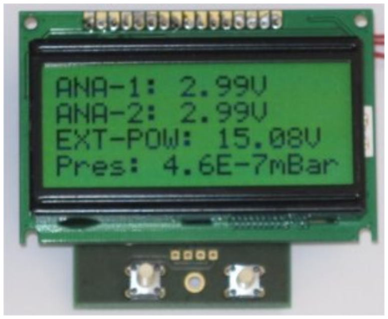
```
T1=302.19K Pt100
T2=301.98K IN414
T3=n/c     Pt100
T4=n/c     Pt100
```

Page 1



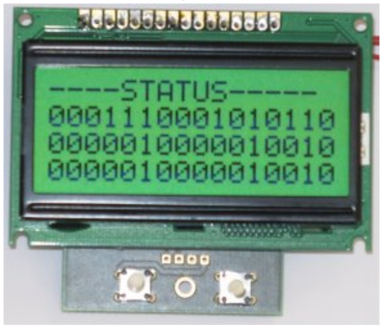
```
----SERVO A--OFF
 Temp=321.39K
Power=0.0W
  Set=310.0K
```

Page 2



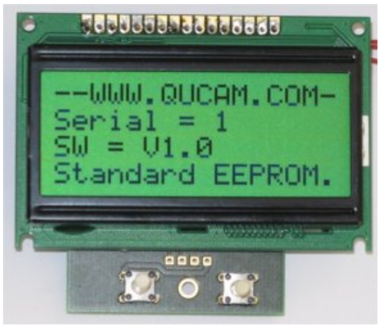
```
ANA-1: 2.99V
ANA-2: 2.99V
EXT-POW: 15.08V
Pres: 4.6E-7mBar
```

Page 3



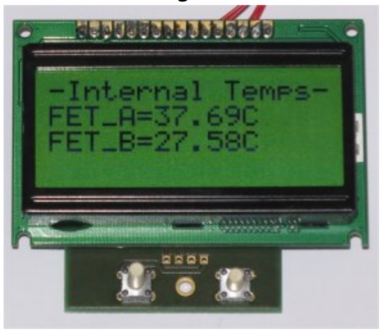
```
----STATUS------
0001110001010110
0000010000010010
0000010000010010
```

Page 5



```
--WWW.QUCAM.COM-
Serial = 1
SW = V1.0
Standard EEPROM.
```

Page 6



```
-Internal Temps-
FET_A=37.69C
FET_B=27.58C
```

Page 7

Figure 21 Large LCD info pages

6. Detailed characteristics.

6.1. Electrical characteristics.

The board draws 90mA from the 5V ARC backplane when no servo heaters are connected. The micro runs from an 8MHz quartz oscillator boosted to 32MHz by an on-chip PLL . The servo power demand is derived from a PWM signal with a frequency of 240Hz filtered by a second order 1.2Hz Sallen-Key. The board has a boost converter running at a commutation frequency of 1.6MHz. On-board coms is via an I^2C interface running at 100kHz and an SPI interface running at 2MHz.

6.2. Temperature inputs.

In diode mode the maximum differential input voltage is 1.08V. This limits the minimum temperature measurable with diode sensors. In RTD mode the corresponding maximum voltage is 142mV which limits the *maximum* temperature measurable to about 383K with a Pt100 sensor. Precision components are used in the preamps. A 5ppm voltage reference, precision instrumentation amplifiers and op-amps were used and 10ppm resistors were used in the critical areas. Initial tolerances for the critical resistors was 0.1% which still requires individual calibration of each board. The end-to-end transfer function of each temperature input channel was measured in both RTD and diode modes (link selected) at between 5 and 10 differential input voltages. These calibration voltages were provided by a custom calibration board based around a 15ppm AD581KH reference and checked with a freshly calibrated Rigol DM3068 6-1/2 digit DVM. The ADU signal was plotted as a function of input voltage and a single order fit made. The offset and gradient of each fit was then programmed into the micro's EEPROM to effectively calibrate-out gain and offset tolerances in the preamps. Higher order errors, for example non-linearity in the ADCs were not removed by this process. Plots of the residual errors after calibration (Figures 23,24) showed that these higher order errors were, however, very tiny: approximately 10µV for RTD mode and 100µV for diode mode (both referenced to the sensor inputs). The superiority of the discrete 16-bit converter used on channels 1 and 2 was clearly revealed by these plots.

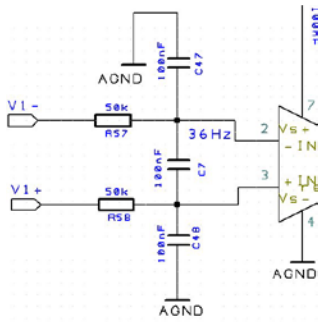


Figure 22 Temperature sensor differential inputs

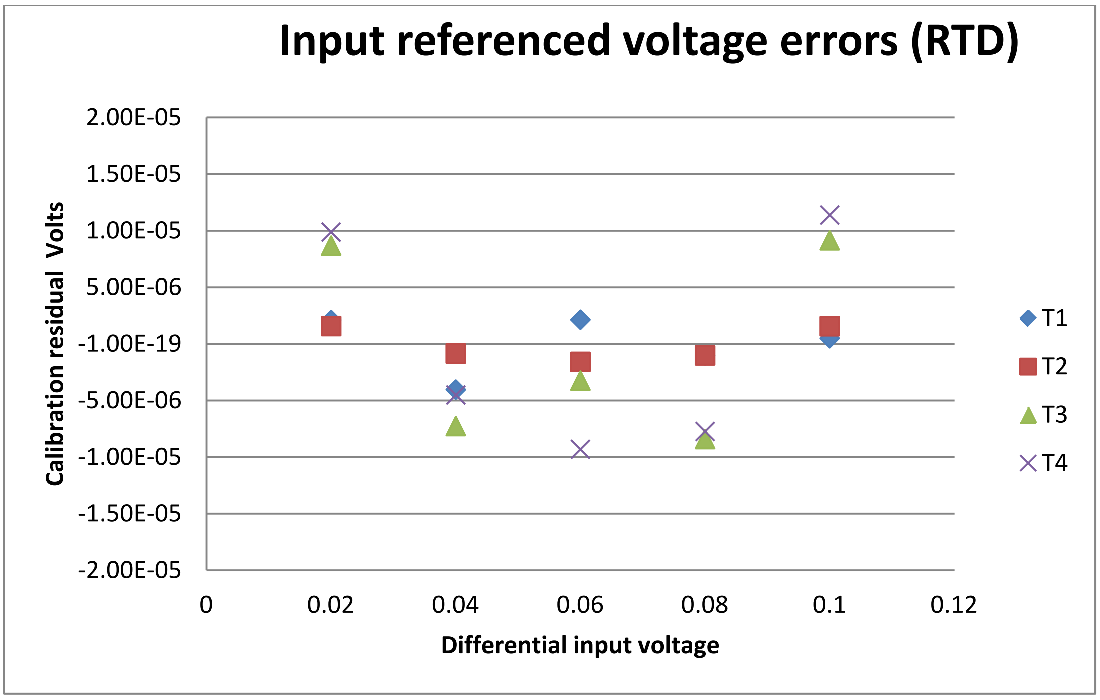


Figure 23. Residual calibration errors for each RTD input.

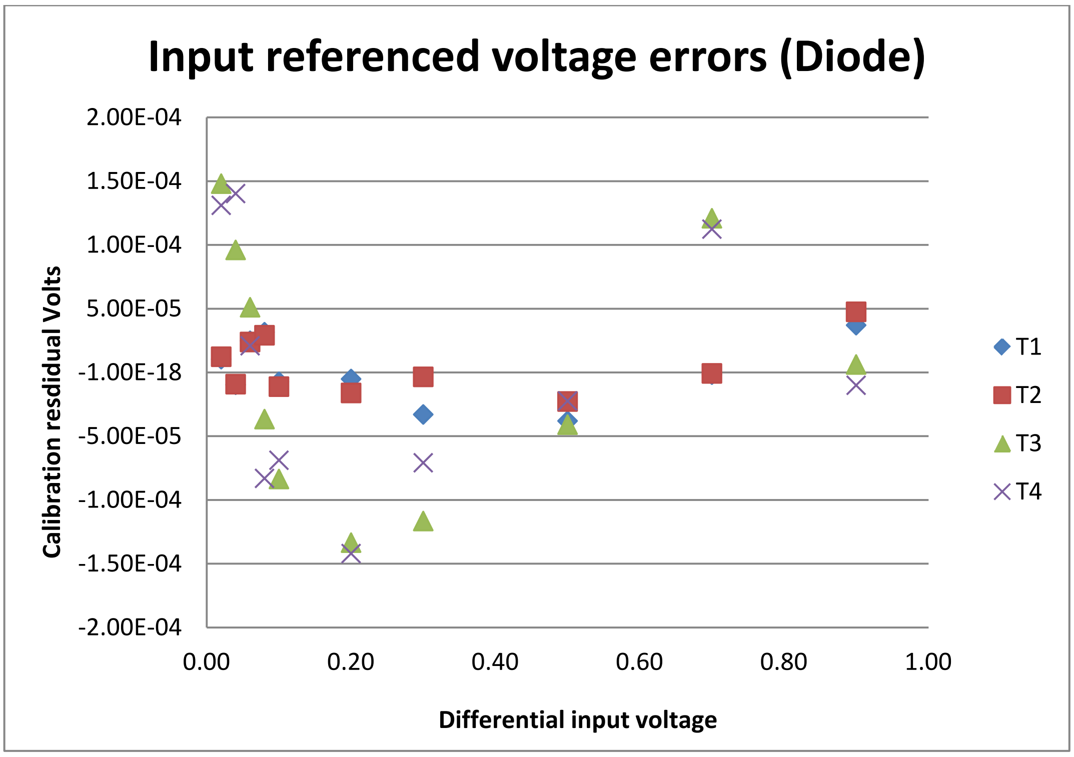


Figure 24. Residual calibration errors for each diode input.

Although every effort was made to select low temperature-coefficient components it was still necessary to test the temperature stability of the board as a whole. This was done by placing the board in a fridge and cooling to 15C. It was then placed in a insulating jacket and left to warm up slowly on the bench whilst recording the temperature of both an on-board temperature sensor and an external dummy resistor. The resistor was used to imitate a Pt100 sensor albeit one with negligible temperature coefficient. A graph of board temperature against indicated dummy-sensor temperature was then plotted to reveal any temperature coefficients in the board itself. The errors are consistent with the gain stability with temperature of the front-end instrumentation amplifiers.

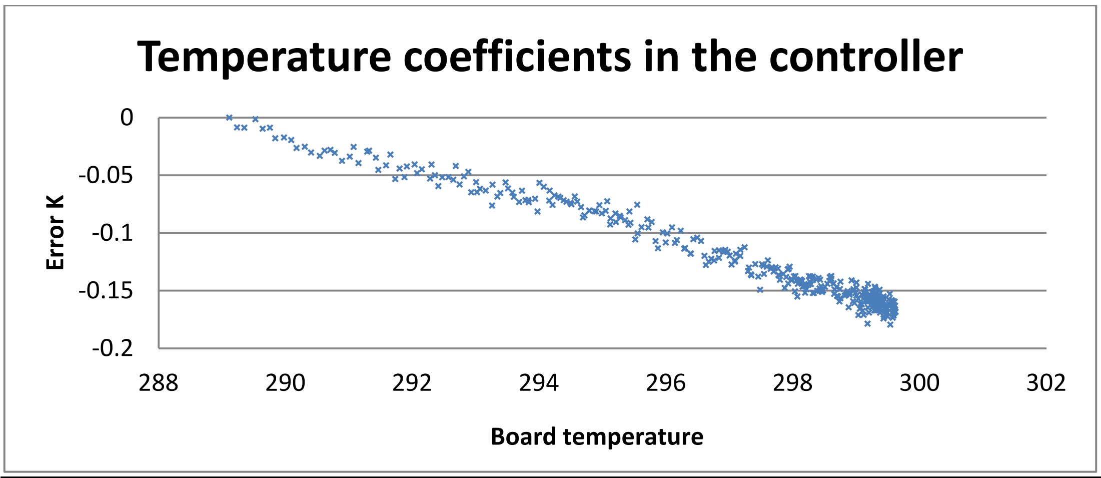


Figure 25. Board temperature sensitivity.

The amplifier noise on each sensor channel was measured and is shown in Table 5. The noise voltages are referenced to the inputs.

	Noise with Diode Sensor	Noise with Pt100 sensor
16-bit channels (1,2)	24μV RMS	5μV
12-bit channels (3,4)	50μV	5μV

Table 5. Input referenced amplifier noise in Volts

This result suggests we are ADC noise limited. In Table 6 the measured voltage noise is converted into units of Kelvin assuming conversion factors of 2.5mV/K for diode sensor and 385μV/K for Pt100 sensor.

	Noise with Diode Sensor	Noise with Pt100 sensor
16-bit channels (1,2)	10mK RMS	13mK
12-bit channels (3,4)	20mK	13mK

Table 6 Input referenced amplifier noise in Kelvin

6.3. Analogue Inputs

The two analogue inputs have at least a 10V linear input range. They hit the rail at approximately 12V. The input network is the same for both channels. It is shown in Figure 26. Both inputs can be used as general purpose, however, channel 1 has the added functionality that the on-board software can convert the measured voltage into a pressure if an MKS or Pfeiffer gauge is attached.

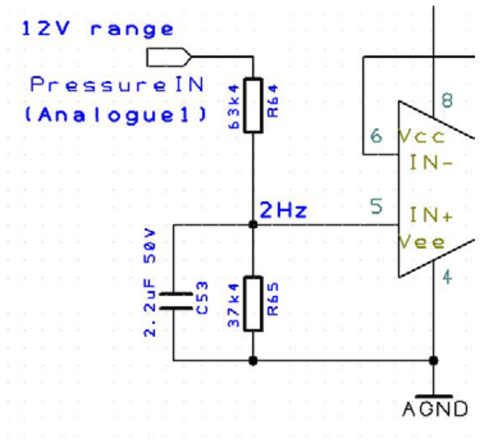


Figure 26. Analogue input channels (ANA-1 and ANA-2)

Resistor and op-amp tolerances meant that each channel had a slightly different offset and gain. This was calibrated out using the same custom calibration board used for the temperature channels. Calibration residuals of around 2mV across the whole 0-10V range were measured on the prototype board.

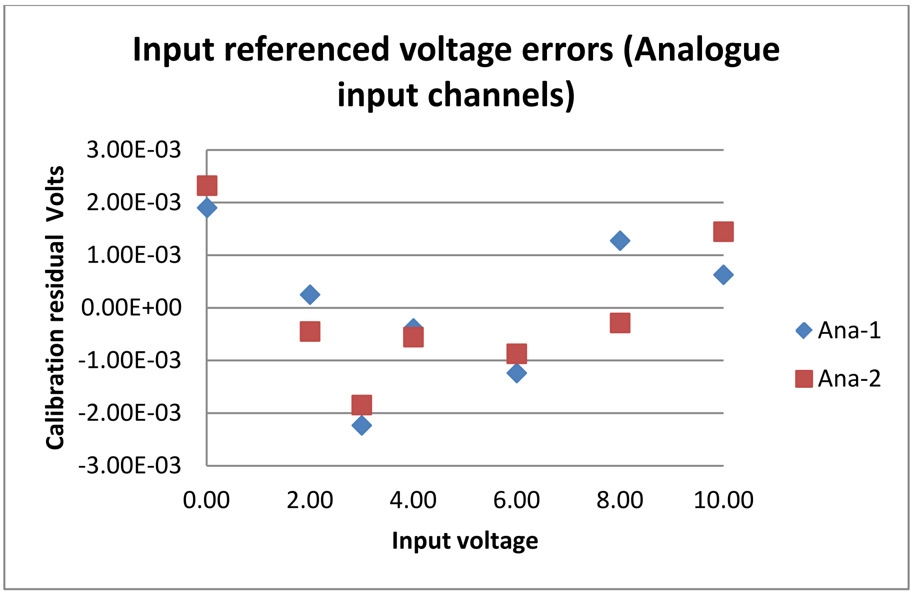


Figure 27. Analogue input channels residual calibration errors.

6.4. Digital IOs.

All digital IOs (except the open drain outputs) are 5V logic buffered with series 1K resistors for protection. The open drain (OD) outputs have 50R series current limiting resistors and 24V protection Zeners to deal with inductive loads. The OD outputs are switched on using the EXP command. The delay between the 'EXP' command being transmitted and the OD output transistor switching-on is 3.5ms.

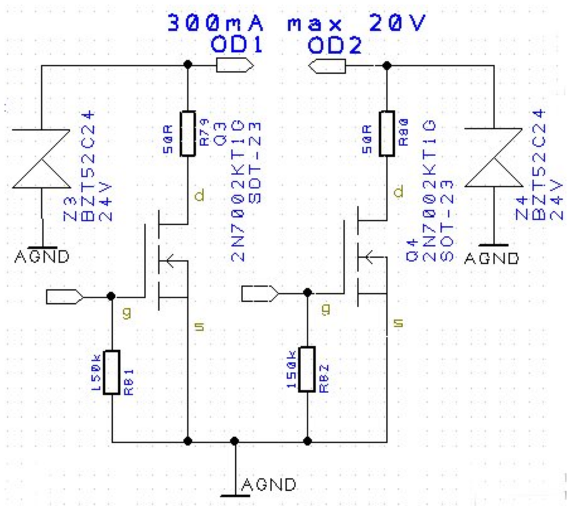


Figure 28. Open-drain digital outputs

The 8 GPIO (GPIO-0 to 7) pins are a bit tricky for users not familiar with quasi-bidirectional logic chips. The interface chip used is a Texas PCF8574. This device is accessed using just the 'GPO' and 'GPI' commands. each bit can be read or written. If a bit is required as an input it must first have a ´1´ written to it. On power-up all bits default as inputs. As can be seen from Figure29 below, the pull-up transistor to the output bit (PO-P7) is weak and can only supply 100uA. The pull down transistor, however, is full-strength limited only by the 1k series protection resistors that are present on the board. With a bit pulled weakly high the pin voltage simply follows any external input. Any bit used as an input must *not* have '0' written to it by the user (using the 'GPO' command) or the pin will have an indeterminate state.

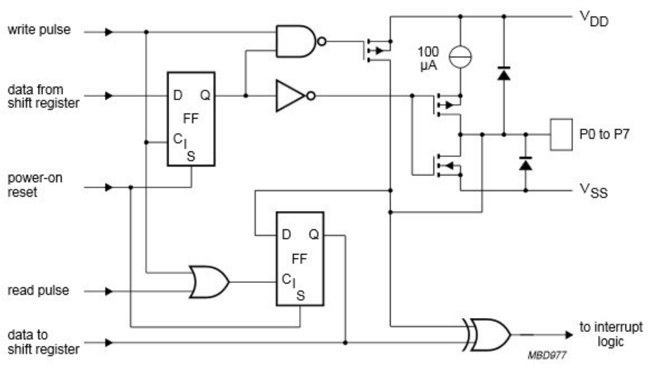


Figure 29. Texas PCF8574 data sheet extract.

There is a delay of 2.7ms between the GPO command being sent and the output pins of the PCF8574A changing state.

There are five other 5V digital output pins not under direct user control. These are the Alarm-A, Alarm-B, Alarm-P, Status-A and Status-B lines that are available on the 44-way D connector and are driven directly from the micro through 5k protection resistors. These lines can be directly connected to indicator LEDs. The two Status lines flash at 1Hz when a servo is active, becoming continuous when the temperature error is less than 1K. Rapid flashing indicates an unspecified board failure has caused a watch-dog timeout.

6.5. Servo performance.

The test heatsink described in 3.4. was servoed overnight to check its long term stability and noise. The noise about the set point was 40mK RMS. A 1N4148 diode sensor with 10mK RMS noise was used in this run.

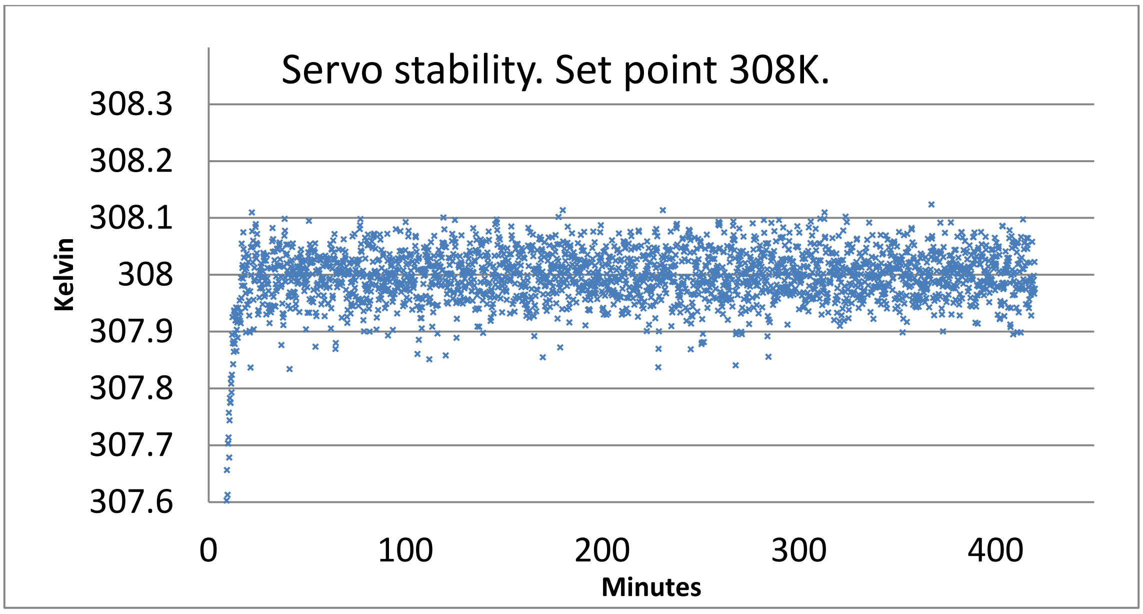


Figure 30. Overnight servo stability

Temperature as a function of time is shown in Figure 30. Figure 31 shows the variation in heater power over the same run.

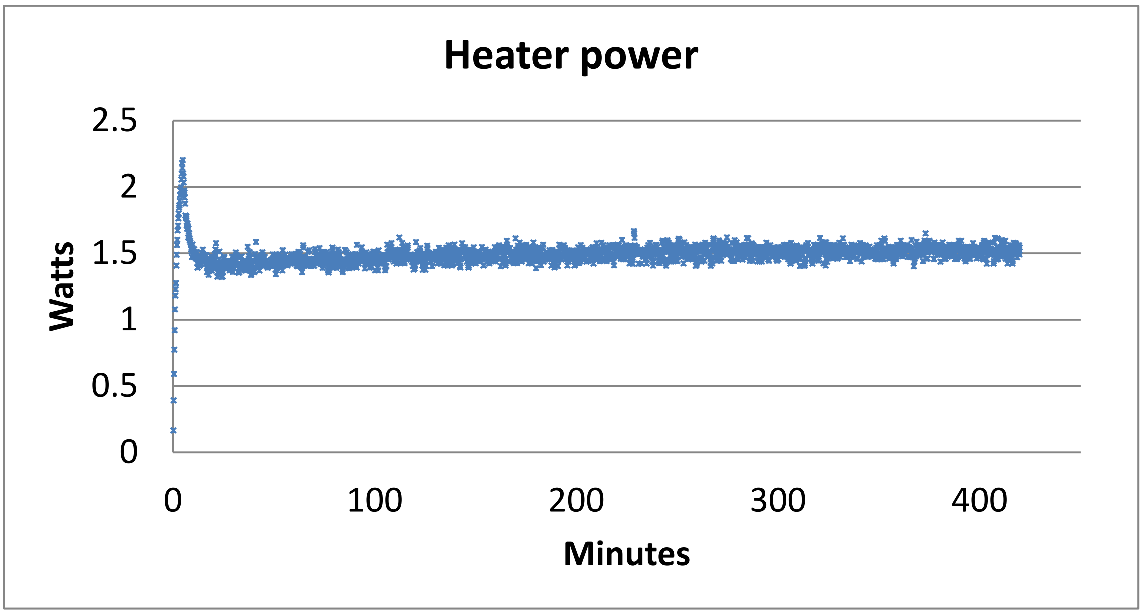


Figure 31. Heater power telemetry during run shown in Fig. 30.

6.6. Servo power outputs.

These are driven by hi-side N-channel MOSFETs. Assuming use of an external 15V power supply then the output voltage range is 10mV to 13.8V in high range and 10mV to 7 V in low range. Maximum output current is 700mA for each of the two servos. They can operate simultaneously at full power.

7. Command Vocabulary.

The vocabulary is described here in its ASCII representation. The commands, however, must be sent as parameters to the "Command" API function (see Section 4.2) as 24-bit integers. The Appendix shows the integer equivalents of each command. The commands are explained in detail in this section with some examples of how they are used in the API at the end.

General system information/status--

RID	Read serial number of board. Returns serial number as an integer.
SAV	Save current temperature controller setup to EEPROM. Returns "DON".

SYS	Read system status word. Returns an integer. See Appendix for details.
GSS n	Read status of servo n. For servo A n=1, for servo B n=2. Returns "ERR" or a 16-bit status word. See Appendix for details.
TDL n	Test Data Link. Returns n.
LED n	Enable/disable LEDs. 0=off, 1=on. Returns "DON" or "ERR".
LCD n	Select LCD display page 0-7. Returns "DON" or "ERR"
TCI n	Read temperature curve ID string. Where n= curve number of interest. returns "ERR" if n exceeds the number of curves stored, otherwise returns a 3 character ID string. The preprogrammed sensor curve EEPROM will return the following ID strings: 1 ="Pt1" i.e. Pt100 2 ="DT6" i.e. DT670 3 = "S90" i.e. S900 4 ="IN4" i.e. 1N4148

Digital control--
-

EXP n m	Do timed one-shot on Open-Drain digital output n for duration m. Immediately returns "DON" or "ERR". If n=1 then units of m are 100milli-seconds. If n=2 then units of m are 100micro-seconds. Maximum value of m is 32767.
ABR n	Abort timed one-shot on Open-Drain digital output n. Returns "DON" or "ERR".
SOD n m	Manually set state of Open-Drain digital output n. Returns "DON" or "ERR". n=1 or 2. m=0 or 1.

GPI	Read parallel input port. Returns an integer.
GPO n	Write parallel input port with value n. Returns "DON".

Analogue and pressure---

RPR	Read external power rail voltage. Returns integer with units of milli-Volts.
KEL n	Read a temperature channel. Returns integer with units of milli-Kelvin or "ERR". Can be used to read temperature of internal power FETs if n=5 (Servo A) or n=6 (Servo B). If any channel has a cable fault, the sensor is simply not present or the sensor is outside of its calibration range then the returned temperature will be 999999mK.
PRE	Read pressure on analogue channel 1. Returns integer that encodes an exponential-format pressure in units of mBar. For encoding see Section 4.3.
RAN n	Read analogue input channel 1 or 2. Returns "ERR" or an integer with units of milli-Volts.
NOI n	Read front-end noise on an analogue or temperature input channel. This is raw unfiltered noise. Returns "ERR" or an integer with units of micro-Volts RMS. Values for n are : 1,2,3,4 :for external sensor channels 5,6 : for on-board temperature sensors (power MOSFETs) 7,8 : for external analogue channels
SET PAL n	Set pressure alarm to value n. n is an integer encoding an exponential value. For encoding see Section 4.3. Returns "DON" or "ERR". Front panel LED activated if pressure above this trigger level.
GET PAL	Get pressure alarm value. Returns trigger pressure as an integer encoding an exponential-format value.

SET PSE n	Select pressure sensor n. n=1 means an MKS972 sensor is connected to Analogue input 1, n=2 means a PKR251 is attached. Returns "DON" or "ERR".
GET PSE	Get selected pressure sensor. Returns 1 for MKS972 or 2 for PKR251.

Servo loop configuration---

ENA n	Enable servo n. For servo A n=1, for servo B n=2. Returns "ERR" or "DON".
DIS n	Disable servo n. For servo A n=1, for servo B n=2. Returns "ERR" or "DON".
SET TAR n m	Set target temperature of servo n to value m. For servo A n=1, for servo B n=2. m has units of milli-Kelvin. Returns "DON" or "ERR".
GET TAR n	Get target temperature of servo n. Returns "ERR" or an integer with units of milli-Kelvin.
SET SEN n m	Select sensor m for use with servo n. For servo A n=1, for servo B n=2. m =1 or 2 only. Returns "DON" or "ERR".
GET SEN n	Get sensor used for servo n. Returns "ERR" or an integer=1 or 2 depending on currently selected temperature sensor
SET MAP n m	Map sensor n to calibration curve m. n=1,2,3 or 4. m=1,2,3 or 4, although more can be added with an updated EEPROM. Returns "DON" or "ERR".
GET MAP n	Get calibration curve used for sensor n. Returns "ERR" or an integer=1,2,3 or 4 depending on current sensor mapping.
SET PRO n m	Set proportional coefficient of servo n to value m. For servo A n=1, for servo B n=2. m is between 0 and 2000 Returns "DON" or "ERR".

GET PRO n	Get proportional coefficient for servo n. For servo A n=1, for servo B n=2. Returns "ERR" or the proportional coefficient.
SET INT n m	Set integral coefficient of servo n to value m. For servo A n=1, for servo B n=2. m is between 0 and 1000 Returns "DON" or "ERR".
GET INT n	Get integral coefficient for servo n. For servo A n=1, for servo B n=2. Returns "ERR" or the integral coefficient.
SET SLO n m	Set maximum slope of servo n to value m. For servo A n=1, for servo B n=2. m is an integer with units of milli-Kelvin per minute. Returns "DON" or "ERR".
GET SLO n	Get maximum slope for servo n. For servo A n=1, for servo B n=2. Returns "ERR" or the slope value in milli-Kelvin per minute.
SET IWI n m	Set integral window width of servo n to value m. For servo A n=1, for servo B n=2. m is an integer with units of milli-Kelvin per minute. Returns "DON" or "ERR". See Section 3.2 for detailed explanation.
GET IWI n	Get integral window width for servo n. For servo A n=1, for servo B n=2. Returns "ERR" or the width value in units of milli-Kelvin.
SET HLP n m	Set heater low-power range for servo n. For servo A n=1, for servo B n=2. m=1 to select low power, 0 for high power. Returns "DON" or "ERR". Selecting low power restricts the servo heater voltage to 7V.
GET HLP n	Get power-range setting for servo n. For servo A n=1, for servo B n=2. Returns "ERR" or the heater setting (1=low power, 0= high power).
SET LIM n m	Set temperature limit of servo n to value m. For servo A n=1, for servo B n=2. m is an integer with units of milli-Kelvin.

	Returns "DON" or "ERR". Heater is disabled above this temperature
GET LIM n	Get temperature limit for servo n. For servo A n=1, for servo B n=2. Returns "ERR" or the limit value in milli-Kelvin.
SET TRG n m	Set temperature alarm of servo n to value m. For servo A n=1, for servo B n=2. m is an integer with units of milli-Kelvin. Returns "DON" or "ERR". Front panel LED activated if temperature above this trigger level. Heater state unaffected.
GET TRG n	Get temperature alarm value for servo n. For servo A n=1, for servo B n=2. Returns "ERR" or the trigger value in milli-Kelvin.
SET FIL n m	Apply low-pass digital filter m to temperature channel n. n=1,2 ,3 or 4. m=0,1,2 or 3. Returns "DON" or "ERR". See Section 3.6 for detailed explanation.
GET FIL n	Get digital filter currently used by temperature channel n. n=1,2,3 or 4. Returns "ERR" or the filter in use (0,1,2 or 3).
HVO n	Read the voltage on a servo heater. n=1 for servo A, n=2 for servo B. Returns "ERR" or an integer with units of milli-Volts.
HCU n	Read the current through a servo heater. n=1 for servo A, n=2 for servo B. Returns "ERR" or an integer with units of milli-Amps.
HPO n	Read the power of a servo heater. n=1 for servo A, n=2 for servo B. Returns "ERR" or an integer with units of milli-Watts.
GST n	Get servo temperature. n=1 for servo A, n=2 for servo B. Returns "ERR" or an integer with units of milli-Kelvin. If the sensor has failed or if its temperature has fallen off the edge of the calibration range then a value of 999999mK will be returned.

Some examples of how to use the commands in the API function call follow:

Read Servo Status bits for Servo A.:

```
result= CameraAPI.Command( 3, GSS, 1);
```

Set target temperature for servo A to 158Kelvin:

```
result= CameraAPI.Command( 3, SET, TAR,1,158000);
```

Read back the limit temperature for servo B:

```
result= CameraAPI.Command( 3, GET, LIM,2);
```

or equivalently:

```
result= CameraAPI.Command( 3, 0x474554, 0x4c494d,2);
```

8. Typical setup recipe.

It is assumed that the temperature control board is being used to regulate the temperature of a cryogenic CCD camera. If the Java GUI is used from within the OWL DAS program then setup is quite intuitive but if the user writes their own scripts or uses the ARC API library as part of a third party DAS system then they should be aware of the following low-level command sequences.

-First the hardware connections need to be made. The servo channel A or B needs to be chosen (they are identical) and the servo heater attached to the relevant board edge 44-way connector pins. Likewise the temperature input channel needs to be chosen and the 4 sensor connections made. Only temperature channels 1 and 2 can be used as servo inputs. Then the links need to be set /cleared on the board (see Section 1.1) to select the correct current source and preamplifier gain for the chosen sensor type.

-Having set/cleared the links it is then necessary to let the boards micro 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 (1 for A or 2 for B) and m= temperature sensor channel (1or 2).

-Next we must tell the micro the sensor response curve that we will use with the selected sensor channel. This is done using the "SET MAP n m" command where n=sensor channel and m=curve number. If we need a reminder of which curve number refers to which sensor type we can use the "TCI n" command to read back a 3 character curve ID string.

-We then program the desired target temperature using the "SET TAR n m" command, where n= servo channel we are using (1 for A or 2 for B) and m is the desired temperature in milli-Kelvin.

-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 micro's non-volatile memory using the "SAV" command. The saved configuration is loaded after a power cycle or reset. Servos are always disabled after a power cycle or reset and must be explicitly enabled.

-

APPENDIX.

A. Pin-outs.

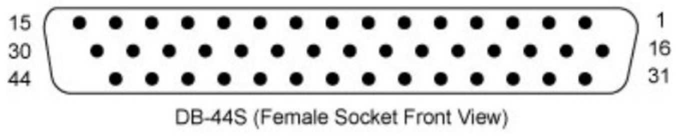


Fig 32. Edge connector pin-out

Pin	*Name*	*Description*
1	POW-B	*Heater power servo B*
2	GPIO-7	*Digital i/o bit 7*
3	GPIO-0	*Digital i/o bit 0*
4	GPIO-2	*Digital i/o bit 2*
5	GPIO-3	*Digital i/o bit 3*
6	OD-2	*Open-Drain output 2*
7	Alarm-A	*Servo A temp. alarm out*
8	V1+	*Sensor 1 connection*
9	I1+	*Sensor 1 connection*
10	V2+	*Sensor 2 connection*
11	I2+	*Sensor 2 connection*
12	V3+	*Sensor 3 connection*
13	I3+	*Sensor 3 connection*
14	V4+	*Sensor 4connection*
15	I4+	*Sensor 4 connection*
16	RET-A	*Power return servo A*
17	RET-B	*Power return servo B*
18	GPIO-5	*Digital i/o bit 5*
19	GPIO-1	*Digital i/o bit 1*
20	PGND	*Power ground*
21	OD-1	*Open-Drain output 1*
22	Alarm-B	*Servo B temp. alarm out*
23	ANA-1	*Analogue input 1 (pressure)*
24	GND	*System ground (not-power)*
25	ANA-2	*Analogue input 2*
26	GND	*System ground (not-power)*
27	GND	*System ground (not-power)*
28	GND	*System ground (not-power)*
29	GND	*System ground (not-power)*
30	GND	*System ground (not-power)*
31	POW-A	*Heater power servo A*
32	GPIO-6	*Digital i/o bit 6*
33	Alarm-P	*Pressure Alarm out*

34	STATUS-A	*Servo status A out*
35	STATUS-B	*Servo status B out*
36	GPIO-4	*Digital i/o bit 4*
37	V1-	*Sensor 1 connection*
38	I1-	*Sensor 1 connection*
39	V2-	*Sensor 2 connection*
40	I2-	*Sensor 2 connection*
41	V3-	*Sensor 3 connection*
42	I3-	*Sensor 3 connection*
43	V4-	*Sensor 4 connection*
44	I4-	*Sensor 4 connection*

Table 7. Edge connector pin-out.

Note that the identified power return pins must be used for the heater connections. Although it won't cause damage, use of the non-power system ground pins could induce voltage gradients in the board and disturb the temperature measurement accuracy.

B. Factory setup.

The board is supplied with the following configuration in non-volatile memory :

LEDs enabled and LCD page 1 selected.

Temperature channels:

1: 1N4148 sensor, 0.1Hz digital filter.

2: Pt100 sensor, 0.1Hz digital filter.

3: 1N4148 sensor, 0.1Hz digital filter.

4: 1N4148 sensor, 0.1Hz digital filter.

Temperature Servo A:	Temperature Servo B:
Set-point Target: 160K	Set-point Target: 160K
Alarm: 170K	Alarm: 170K
Temperature limit: 305K	Temperature limit: 305K
Proportional constant: 200	Proportional constant: 200
Integral constant: 80	Integral constant: 80
Integral window width 10K	Integral window width 10K
Slope: 4.5K/minute	Slope: 4.5K/minute
Temperature channel used: 1	Temperature channel used: 2
High-power mode	High-power mode

C. Status Bits.

Each of the servos has a status word (read by "GSS n") with the following encoding:

Bit	Name	Function	Notes
0	ENABLE	Servo is enabled	
1	SENSOR	0=sensor 1 selected 1=sensor 2 selected	
2	OVERHEAT	Sensor has been above limit temperature	Only cleared by re-enabling servo
3	ALARM	Sensor currently above Trigger temperature	
4	/	/	
5	WIREOFF	A sensor wire has broken or no sensor.	Disables servo
6	ATTEMPERATURE	Servo error<1K	
7	INTEGRALON	Integral term is on.	See Section 3.2.
8	OVERCURRENT	Heater resistor has drawn >700mA	Only cleared by re-enabling servo
9	MOSFETHEAT	MOSFET in power amplifier has overheated.	Only cleared by re-enabling servo
10	LOWPOWER	0=high power 1=low power	

Table 8. Servo status bits

There is also a system status word (read by "SYS") with the following encoding:

Bit	Name	Function	Notes
0	BACKPLANE	Detects setting of power link to 16.5V backplane.	
1	EXTPOWER	Set if we are using external power.	
2	EEPROM	Sensor curve EEPROM present.	
3	WDTERROR	Software error. Watch-dog timer has timed out.	For development
4	LEDENABLE	LEDs are enabled.	
5	PALARM	Pressure above alarm value.	
6	PRESSENSE	0=MKS972 sensor 1=PKR251 sensor	
7	OVERVOLTAGE	External power rail has exceeded 15.5V.	Only cleared by re-enabling servo
8	OD1	State of open drain output 1.	
9	OD2	State of open drain output 2.	
10	LCDPRESENT	LCD display plugged in.	
11	ISENSE_A_PRESENT	Current sensor present	For development
12	ISENSE_B_PRESENT	Current sensor present	For development
13	ARC_PRESENT	ARC controller present	Board can also be used without ARC with text-based USB interface.

Table 9. Servo status bits

D. Integer representations of commands.

RID	0x524944	**GPO**	0x47504f
SAV	0x534156	**RAN**	0x52414e
SYS	0x535953	**NOI**	0x4e4f49
ABR	0x414252	**TCI**	0x544349
RPR	0x525052	**SET**	0x534554
PCI	0x504349	**GET**	0x474554
GSS	0x475353	**TAR**	0x544152
TDL	0x54444c	**SEN**	0x53454e
LED	0x4c4544	**MAP**	0x4d4150
LCD	0x4c4344	**PRO**	0x50524f
ENA	0x454e41	**INT**	0x494e54
DIS	0x444953	**SLO**	0x534c4f
KEL	0x4b454c	**LIM**	0x4c494d
PRE	0x505245	**TRG**	0x545247
HVO	0x48564f	**PSE**	0x505345
HCU	0x484355	**FIL**	0x46494c
HPO	0x48504f	**IWI**	0x495749
EXP	0x455850	**ERR**	0x455252
GST	0x475354	**HLP**	0x484c50
GPI	0x475049	**DON**	0x444f4e