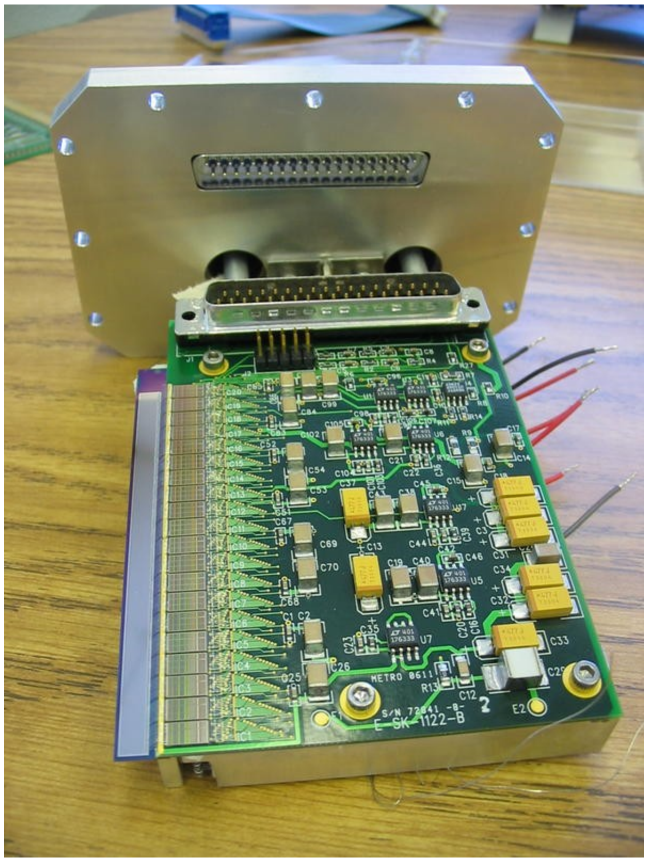
# Operating manual for HERMES-based photon-counting silicon microstrip detectors

## DRAFT

D. Peter Siddons

August 2006



## *Introduction*

The BNL silicon microstrip x-ray detectors are based on the HERMES application specific integrated circuit (ASIC) designed to read out the charge signals from silicon photodiodes with very low noise, and on custom photodiode arrays fabricated in-house at BNL. Each ASIC provides 32 channels of readout, with each channel consisting of a charge-sensitive preamplifier, a pulse shaping amplifier, a set of pulse-height discriminators and pulse counters. Higher channel counts are achieved by adding more ASICs. The diode arrays are low-leakage devices fabricated from high-resistivity silicon using a planar process, and are typically 0.4mm thick.

The ASIC requires various control signals and voltage levels to be provided for proper functioning, together with power supplies for the Peltier cooling elements, if required, and a high voltage bias supply for the photodiodes. These are all provided in a compact interface unit supplied with the detector. This unit also houses a microprocessor system which provides an EPICS interface to all of the detector functionality.

## *Initial setup*

For simplicity, this section will assume room-temperature operation of the detector. Details of the cooling system will be given in section XXX.

Initial setup of the control unit can be done without connecting the detector head. This should be done first, and then the control unit powered down before connecting the ribbon cable and ground strap as described below. The interface unit must be configured to be compatible with the local network environment. A static IP address must be provided, together with the address of an accessible NTP server. The device does not need a boot host, it boots from internal storage and provides its own persistent storage for databases etc. This initial configuration requires the use of the control unit debug port, an RS232 interface, and a terminal emulator. Full instructions for setting the required environment variables is given in Appendix A. The unit does not currently implement persistent storage of configuration parameters (save / restore), so the user must take steps to do so if required.

In order to operate the detector, the following prerequisites must be satisfied:

1. The coolant circulator must be operating. The circulator should be compatible with aluminum, since the internal cooled parts are fabricated from aluminum for lightness. This means that 'house water' in general cannot be used for this purpose since it will result in destructive electrochemistry causing severe corrosion and eventual failure. The temperature of the coolant depends on the needs of the experiment, but should be at most 22C. It is desirable for the water circulation to be interlocked to the control unit, such that the power is removed in case of a circulation failure. A coolant failure with power applied can cause overheating of the system which could result in permanent damage.
2. The vacuum system must be operating if the detector is to be operated cold (i.e. Less than room temperature). In this case the vacuum should be better than 1E-5 torr. If room temperature operation gives satisfactory performance, then the detector can be operated at atmospheric pressure, preferably purged and back-filled with dry nitrogen. For low-temperature operation it

is desirable for the vacuum system to be interlocked such that power is removed in case of a vacuum failure. This interlock should also cause the detector temperature to be returned to ambient. A vacuum vent while the system is cold can destroy the diode array.

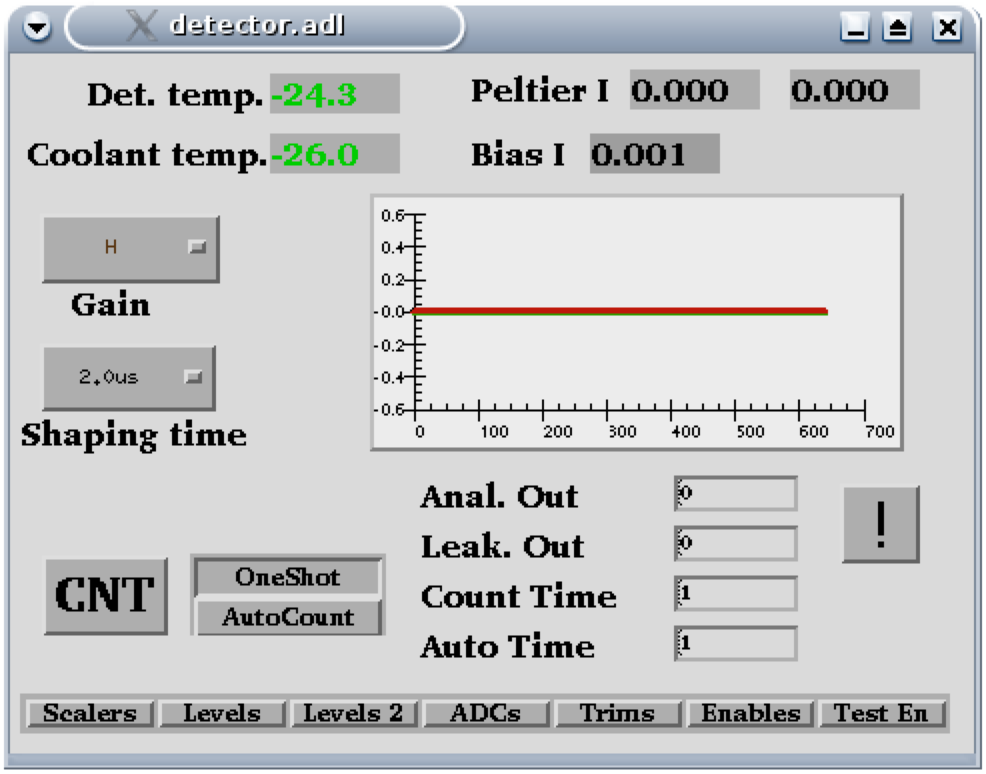
3. The detector head must be connected to the control unit via the supplied shielded ribbon cable. The additional ground strap connection must be securely attached at the detector head and at the control unit.
4. Power can now be applied to the control unit. It takes a few seconds for the system to boot. If the debug RS232 connection is operational, the boot-up commentary can be seen. The usual cause of a slow boot is the failure to find an NTP server. The system tries several times before giving up and setting some default time.
5. A standard EPICS installation is available.

It should now be possible to communicate with the detector via EPICS. Assuming a standard EPICS environment at a workstation on the network to which the detector is attached, executing

```
caget det1
```

should return three numbers. If so, then the system is operating correctly, and the MEDM operating screen can be launched.

Normal operation of the detector is via a GUI interface. This interface is a standard EPICS program which uses the MEDM application. This implies that MEDM must be available on the workstation to be used for operations. All of the detector functionality can be controlled from this interface. Some of the helper programs accessed from the MEDM screen are not MEDM applications, but shell scripts or C programs. These must also be installed for full functionality. Since the detector is a standard EPICS device, any standard method of accessing EPICS process variables will work. Since many potential users of the device will be familiar with the Spec diffractometer control software, a simple macro set is provided to allow its use in that environment. More detailed information on the process variables and their use is given in the reference manual for the EPICs record (Appendix B).



Det. temp. -24.3
Coolant temp. -26.0
Peltier I 0.000 0.000
Bias I 0.001
Gain H
Shaping time 2.0us
Anal. Out 0
Leak. Out 0
Count Time 1
Auto Time 1

The main screen is shown above. It consists of seven key elements:

- Informational displays showing the detector element temperature and the temperature of the cooled heat sink plate inside the detector housing
- Menus for setting the amplifier gain and shaping time
- A button for setting continuous counting or one-shot counting mode
- A button for initiating a count cycle
- A row of buttons for accessing subsidiary screens which access other features
- Text entry windows for controlling the counting time and some debug features
- A menu item for activating various external programs provided with the system, which are not MEDM applications.
- A plotting area, where counter contents are displayed.

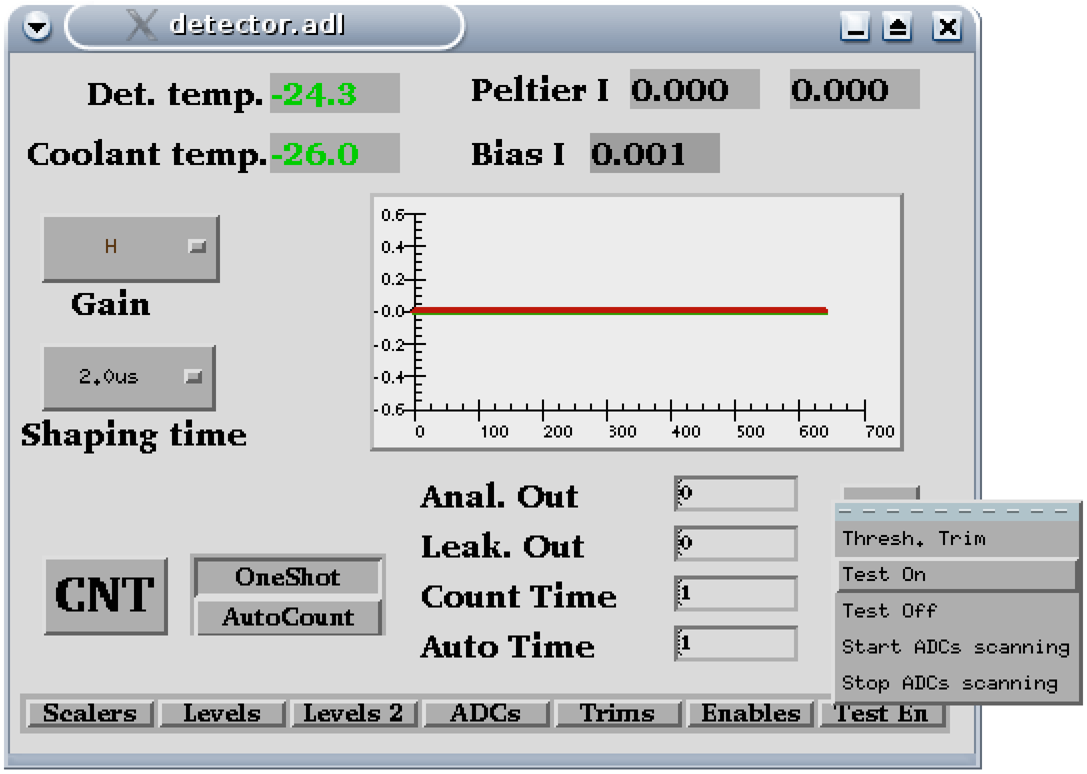
Although the detector record is very similar to (and is derived from) the Scaler record (part of the SynApps package from ANL), it has some key differences which are a consequence of the architecture of the detector ASIC:

- It is not possible to read the counter contents 'on the fly'. Thus, it is not possible to use this detector to 'count to monitor'. This is a consequence of the fact that reading the contents of the HERMES counters causes them to be reset to zero. Thus, reading counters in the middle of a long acquisition will generate incorrect data when the count period is over. In any case, digital activity in the ASIC during a count operation will cause significant disturbance to the sensitive amplifiers, with undesirable consequences.

- There are several 'results' available from the device. Whereas the standard Scaler record provides a separate field for each counter, in a system with many counters this is impractical. The detector record presents its scaler contents as vectors. The length of the vector depends on the size of the detector, but in any case can be determined by reading the NCH field of the record. Since each channel has three counters associated with its three discriminators, there are three results, located in fields S1, S2 and S3. In addition, the VAL field returns a three-element vector which contains the sum of all channel counters in each of the three groups.

Otherwise, the record behaves in the same way as the standard Scaler record.

For initial setup, it is helpful to have access to a test pulse generator and an oscilloscope. Connect the oscilloscope to the Analog Output connector on the control unit (near the power cord). Connect the output of the test pulse generator to the 'Cal' connector. If the test pulse generator is of the 'tail pulse' variety, select positive pulse polarity. A square wave generator will produce both polarities, you will have to ignore the negative-going outputs, but they will not cause any harm. Set the pulse rate to be around 1000Hz. Activate the 'Test On' command from the external commands menu (the button with the '!').



Det. temp. -24.3

Coolant temp. -26.0

Peltier I 0.000 0.000

Bias I 0.001

Gain H

Shaping time 2.0us

Anal. Out 0

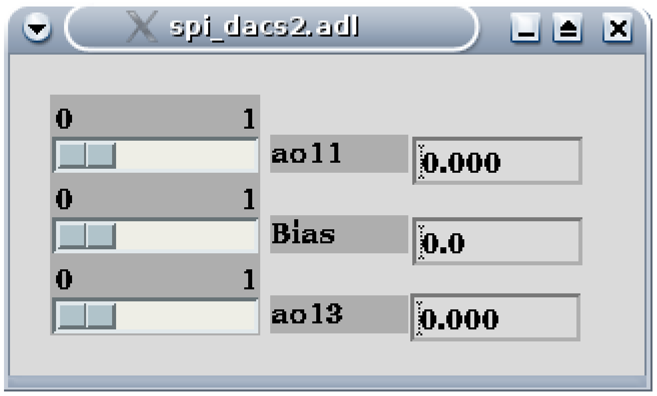
Leak. Out 0

Count Time 1

Auto Time 1

This enables the test pulses to be routed to every channel, so that we can verify operation of the electronics.

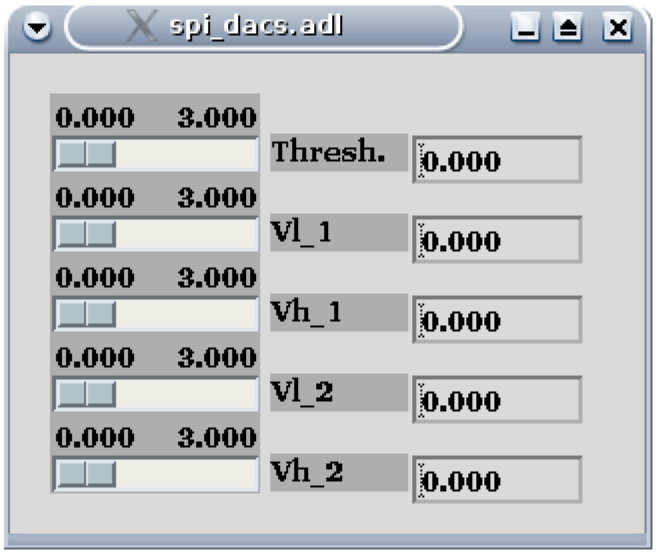
Next, we need to apply some bias to the diode array. With zero bias, the amplifiers are disabled. For this test, we need not apply full bias, merely sufficient to reduce the leakage current and diode capacitance to tolerable levels. A value around 20V is sufficient. The Bias control is part of the 'Levels 2' display. Pop it up by clicking on the button.



| | | |
|---|---|---|
| 0 – 1 | ao11 | 0.000 |
| 0 – 1 | Bias | 0.0 |
| 0 – 1 | ao13 | 0.000 |

Be careful! It is possible to destroy the detector by applying excessive bias, and the supply is quite capable of doing that. Adjust the value step-by-step by clicking in the slider window to the right of the slider. It should increase 1V per click. While adjusting this voltage, observe the analog output on the oscilloscope. As the bias increases beyond about 6V, there should be some signs of life on the output, if only broadband noise. The noise should decrease in amplitude as the bias is increased. If it does not do so, stop and investigate. At 20V the noise should be around 100mV RMS. At this point, pulses should be visible on the oscilloscope. Adjust the amplitude of the test pulse until output pulses of around 2V peak are obtained. This corresponds to an x-ray energy of around 12keV. We should point out that the signal available on the Analog Output connector is not the same as that internal to the ASIC. It is buffered with a x2 gain, and has a 0.3V baseline offset removed. This makes it easier to examine the pulse height spectrum with an MCA, but is confusing when thinking about the on-chip parameters. So, for a 2V peak output, the internal value should be 2/2+0.3 = 1.3V.

Pop up the 'Levels' window.



| | | |
|---|---|---|
| 0.000 – 3.000 | Thresh. | 0.000 |
| 0.000 – 3.000 | Vl_1 | 0.000 |
| 0.000 – 3.000 | Vh_1 | 0.000 |
| 0.000 – 3.000 | Vl_2 | 0.000 |
| 0.000 – 3.000 | Vh_2 | 0.000 |

Adjust the 'Threshold' slider to roughly 0.6V. This should allow pulses to be counted in the S1 scalers. Click on the Auto button to enable continuous count/read cycles. The default count time for both one-shot and continuous counting is 1 second. You can change either by typing values into the appropriate text entry box and hitting <enter>. Be sure to have the cursor in the window while you are making changes, MEDM has the annoying property that focus always follows mouse, and loss of focus looses any typing before <enter>.

You should now observe that the black line in the plotting window is showing a horizontal straight line at the 1000 level, indicating that it is counting the test pulses. Setting the VL1 and VL2 levels to 0.6V, and the VH1 and VH2 levels to 3V should show similar behaviour for the red and green traces.

Further testing requires a source of x-rays. Whether you are using a radioactive source or synchrotron radiation, be sure to verify that you are illumination the entire detector. In order to detect x-rays we need to increase the bias voltage to its correct operating value. This varies somewhat from detector to detector, but should be around 70V. Again, increase it slowly, observing the oscilloscope output as you do so. Verify that analog output pulses are visible across the detector by examining a few elements using the 'Anal. Out' text entry to choose a channel number for output to the oscilloscope. At this point, adjusting the Threshold slider should show counts in the plot window. Removing the radiation source should make the counts go to zero. It may be necessary to adjust the threshold carefully to get good behaviour.

## *Routine operations*

Once these initial tests are completed, the detector can be used directly. Startup involves the same initial steps, namely starting coolant circulation, establishing a suitable vacuum and powering up the control unit, setting levels and bias values as required. We recommend that the unit be kept under vacuum as much as possible, or backfilled with dry nitrogen and sealed for storage. Care should be taken when back-filling to use a low-pressure relief valve in the system so as not to over-pressure the detector housing, since damage may result to the beryllium window.

## *Appendix A: Setting environment variables*

The microprocessor system uses several bootloader environment variables rather than hard-coded parameters in order to simplify its configuration to a users local situation. Bootloader variables are set and read using the RS232 debug port. It communicates at 19200 baud, 8 data bits, 1 stop bit and no parity (8N1). We use the Minicom terminal emulator for this purpose.

On power-up, the bootloader issues various messages, ending with:

```
Autoboot in 10 seconds. Hit 'Esc' to enter monitor
```

So hit 'Esc'. At the monitor prompt, type

printenv

A list of existing environment variables will be printed. The ones we need to correct should already be defined. To change one of them, simply type

setenv VARIABLE value

Execute 'printenv' again to check that we did the right thing. A list of possible environment variables is in the table below. Of these, the ones which must be changed to suit your local network are NETMASK, IPADDR0, NTPSERVER and possibly HOSTNAME.

| ***Name*** | ***Default value*** | ***Comment*** |
|---|---|---|
| NETMASK | 255.255.255.0 | This is typical for a local network |
| SERVER | 172.16.1.1 | BNL standard private beamline network |
| IPADDR0 | 172.16.1.102 | The detector's IP address |
| NAMESERVER | Same as SERVER | Not usually needed |
| NTPSERVER | Same as SERVER | Needed if accurate timestamps are needed |
| GATEWAY | NULL | Used for access to different subnets |
| HOSTNAME | iocNobody | Only used in nameserver context |
| BOOTFILE | NULL | Only used for systems which boot externally |
| CMDLINE | /st.cmd | Looks for boot file in local flash memory |
| NFSMOUNT | NULL | Only used for systems which boot externally |
| TZ | NULL | Used by older EPICS systems |

At BNL we usually have all our beamline IOCs on a firewalled private local network using the 172.16.xx.xx group of addresses. Permanent beamline devices live at 172.16.1.1 – 172.16.1.100. Addresses above this range are free for other uses. HERMES uses 172.16.1.102.

## *Appendix B: Operating the detector cold.*

The detector operates with improved noise and better energy resolution if it is operated at a temperature of around -30 degrees C. In order to operate in this mode several conditions must be met:

1. The chiller / circulator used must be compatible with an aluminum system. The detector should never be connected to a copper-based “house water” circuit. Serious corrosion will result which can destroy the detector.
2. The coolant used must be suitable for operation near or below zero degrees C. Standard automotive antifreeze is suitable, diluted appropriately. We typically run the coolant at -10 degrees C.
3. The vacuum pump used must be oil-free, and the vacuum achieved must be better than $10^{-5}$ torr. Poorer vacuum will slowly deposit ice and other contaminants on the diode array, eventually destroying it.

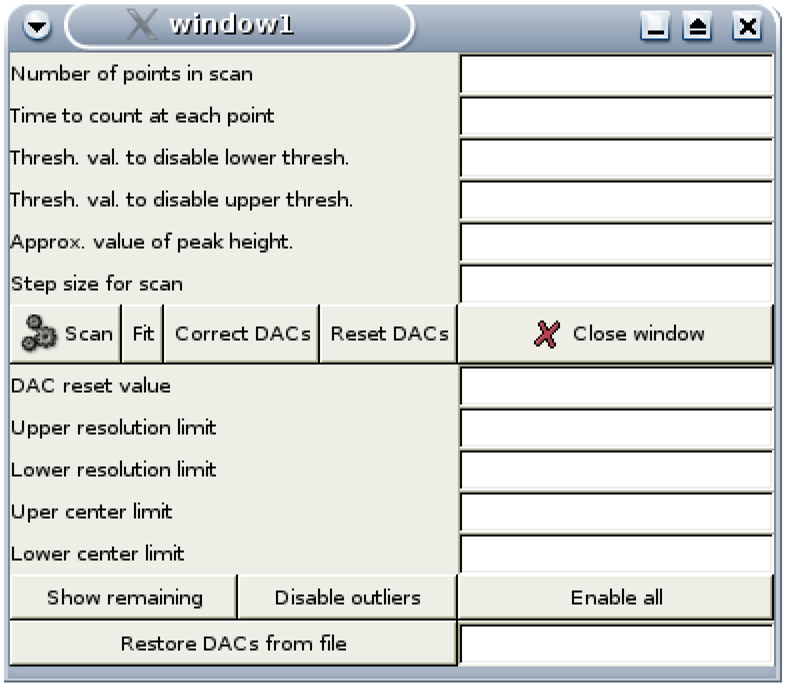
The detector display screen shows the temperature of the chilled heatsink internal to the vacuum chamber, and the temperature of the peltier-cooled block on which the diode array is mounted. The heatsink temperature is usually several degrees higher than that of the coolant supply, due to the thermal load of the ASICS and their power regulators which are thermally coupled to this block. This is normal.

## *Appendix C: Setting the trim DACs*

The HERMES ASIC is equipped with a set of DACs to correct any variations in the offsets of the analog comparators which form the discriminators in the chip. These allow a single reference voltage to control all discriminators and ensure consistent behavior between channels. Since these DACs and comparators are embedded in the ASIC, it is not possible to measure their properties directly. We provide programs which provide a way to infer the properties indirectly.

The requirement is that a given applied level reference causes identical behavior in all channels in response to a given pulse height. Our software takes the counting rate as a function of the externally applied threshold level and performs a least-squares fit to an appropriate function to establish the effective true threshold level for a given external level. The internal DACs are then adjusted to correct the difference between channels. This process is best done using a monochromatic x-ray signal uniformly illuminating the detector. This is not always easy to arrange, but is worth the effort if the best possible performance is required in terms of energy resolution and background suppression. For most diffraction experiments this procedure is not necessary at all. A fluorescent source is a reasonable compromise, and this can be arranged at most beamlines without too much effort. The influence of the beta line on the fitting (which assumes a single line, not a doublet) is not too disruptive.

The correction program is called asic_dac_trim, and is accessed from the external programs menu of the main MEDM screen. It is a Gtk/C application. Its window is shown below.



| | |
|---|---|
| Number of points in scan | |
| Time to count at each point | |
| Thresh. val. to disable lower thresh. | |
| Thresh. val. to disable upper thresh. | |
| Approx. value of peak height. | |
| Step size for scan | |
| DAC reset value | |
| Upper resolution limit | |
| Lower resolution limit | |
| Uper center limit | |
| Lower center limit | |

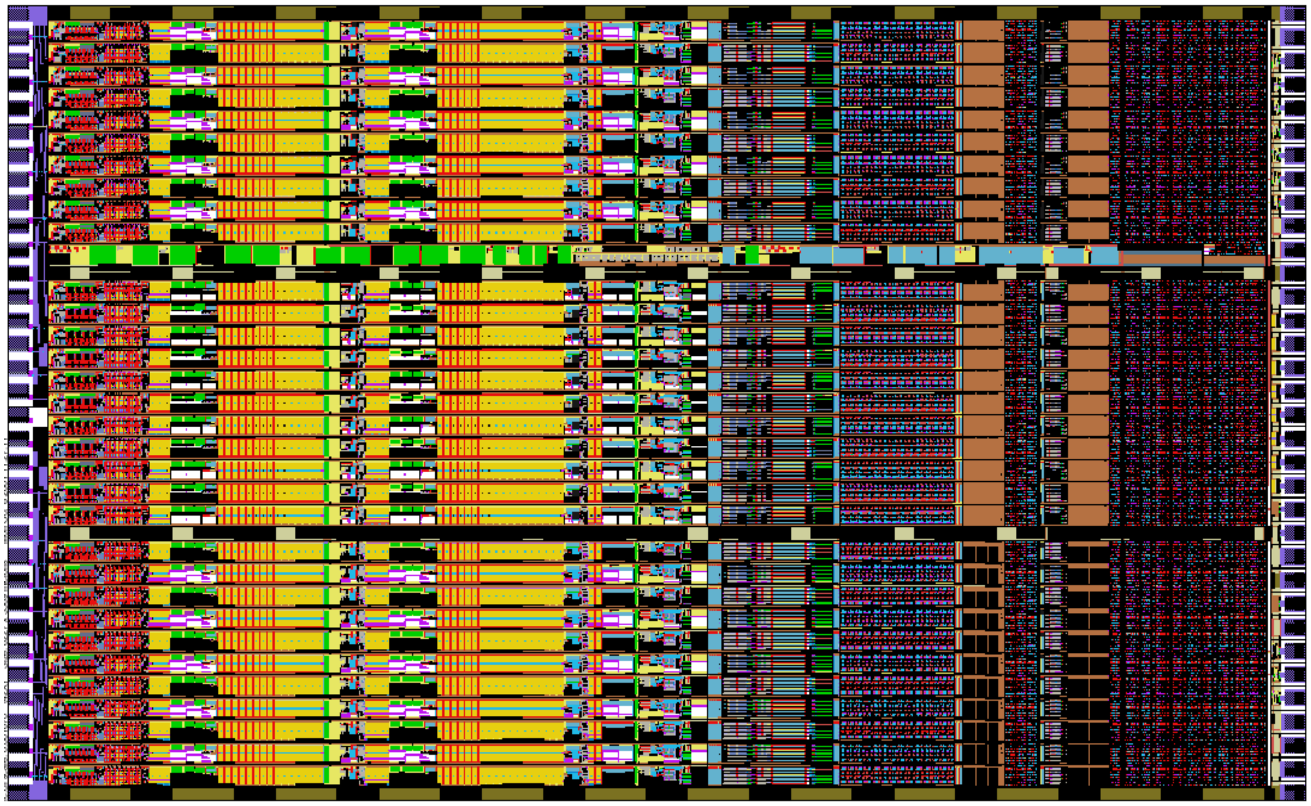
It sets up the threshold scan parameters, and allows us to do repeated scans until we have those parameters optimally set. The data is plotted as it is collected. It then takes the data collected and fits each curve to an erf() function (the pulse-hight distribution for this type of detector is closely Gaussian), and plots the fitted parameters. It then tries to set the trim DACs to minimize the resulting deviations between channels compared to the average. Some channels can have deviations too large to correct. These channels are brought as close as possible, but they will still be in error. It also provides the possibility of simply excluding the worst channels from the correction, possibly improving the correction for the rest. All of this can be done in an iterative manner until the user is happy with the result (or at least, as happy as possible!). The corrections are saved to a file, and can be restored using this program after a detector power-down, so the process need not be repeated unless some major change has occurred.

The process should be repeated after a major change of photon energy, since in fact it is not only correcting threshold offsets, but also gain dispersion among the channels. If it were a pure offset correction, it should be valid for any pulse height. Since it includes gain variations, the corrections will depend slightly on pulse height.

For most users, this is the whole story. However, one more step is necessary if starting from scratch (as we must do initially). The on-chip DACs are also subject to channel-by-channel variations in slope (i.e. Millivolts per DAC step). We therefore have a similar program (asic_dac_cal) which exercises these DACs and uses similar logic to establish the slope of each on-chip DAC. It is essentially the same program, but run five times with different DAC settings spanning their range. The shift in apparent position of the erf() curve can then be used to determine the slope. The results are saved in a file called 'factors.dat', and used by asic_dac_trim in making its corrections. If the file 'factors.dat' does not exist, a nominal constant value is assumed by the trim program.

## *Appendix D: The HERMES ASIC*

This section describes in detail the ASIC contents and signals. The casual user need not read this section since most of the details are taken care of by the microprocessor unit and its embedded software. It is here mainly for completeness.



**TECHNOLOGY :**

CMOS 0.35µm 2-poly, 4-metal

**CHANNELS :**

32 per ASIC

preamplifier / high order shaper

settable gain: 0.75*V/fC, 1.5V/fC*

settable peaking time: *0.5µs, 1µs, 2µs, 4µs*

dual window comparators plus single threshold comparator

comparator thresholds individually adjustable (6-bit DAC)

Three 24-bit counters per channel (one for each comparator)

Power per channel : 8m W

SPI compatible logic

Expected resolution : < 300 eV @ 1µs, 100kHz/pixel

## LOGIC

Chip select via CS:

Chip is selected when CS (chip select) is high.

Transition high to low of CS resets the counters (internal control CRST) and sets ELK and EAN (internal control SAN).

Chip select via token:

Chip selection can also be performed via token by connecting TO (token out) to CS of the next chip. This is the normal operating mode of the BNL multi-ASIC detectors.

Enabling Counters and data SDI/SDO interface

Counters are enabled/disabled through EN.

Read/write mode is selected through WR.

In write mode (WR=0) SCK and SDI are enabled through internal control WE (write enable).

In read mode (WR=1) SCK and SDO (tristated) are enabled through internal control RE (read enable).

If CS is low, WE and RE are forced to low.

## SUPPLIES

| | |
|---|---|
| Vdd | +3.3V analog |
| Vddb | +3.3V analog common bias |
| Vddp | +3.3V analog input MOSFET |
| Vddd | +3.3V digital |
| Vss | 0V analog |
| Vssb | 0V analog common bias |
| Vssd | 0V digital |
| Vsubd | 0V substrate digital |

## ANALOG INPUTS

| | |
|---|---|
| CAL | calibration input |
| VL0 | low threshold, window 0 |
| VL1 | low threshold, window 1 |

| | |
|---|---|
| VH1 | high threshold, window 1 |
| VL2 | low threshold, window 2 |
| VH2 | high threshold, window 2 |

**ANALOG OUTPUTS**

| | |
|---|---|
| OAN | analog output |
| OLK | leakage current monitor |

**ANALOG CONTROLS**

| | |
|---|---|
| BLK | internal leakage current control |
| BLN | output baseline control |
| BDAC | DAC range control |

**DIGITAL INPUTS**

SCK clock input

Positive edge shifts in write mode negative edge shifts in read mode (latch on positive edge), clock must be idling high

SDI 904 bit serial input

first bit written at first clock (positive edge), ending to Ch31

0,0,0,0,EBLK,T1,T2,G,[ECH,ECAL,ELK,EAN,D0-D5/VL1,D0-D5/VH1,D0-D5/VL2,D0-D5/VH2]x32,

first bit of data stream is D5/VH2 of Ch31, last bit of data stream is 0.

CS chip select

Active high

CS transition 1 to 0 resets the counters and sets EAN and ELK

EN counter enable/disable

EN=1 enables the counters

EN=0 disables the counters

WR write/read mode

WR=1 read mode

WR=0 write mode

RST global reset

Active low

OR-ed to internal power-on reset

|  |  |
|---|---|
| TO | token output |
|  | To be connected to CS of the next chip |
| TCK | token clock input |
|  | positive edge shifts token |

**DIGITAL OUTPUTS**

SDO Serial data out (clocked by SCK)

2304 bit counter serial output; first bit available before first clock, starting from Ch31 [C23-C0/W0, C23-C0/W1, C23-C0/W2]x32, first bit of data stream is C23/W0 of Ch31, last bit of data stream is C0/W2 of Ch0.

**DIGITAL CONTROLS**

These are controlled via bits in the serial command stream described above.

|  |  |
|---|---|
| EBLK | internal bias leakage enable |
|  | 0 enable |
|  | 1 disable |
| T2,T1 | peaking time controls |
|  | 00 4.0µs |
|  | 01 2.0µs |
|  | 10 1.0µs |
|  | 11 0.5µs |
| G | gain control |
|  | 0 1500mV/fC |
|  | 1 750mV/fC |
| ECH | channel enable |
|  | 0 enable |
|  | 1 disable |
| ECAL | calibration input enable |
|  | 1 enable |
|  | 0 disable |
| ELK | leakage current monitor enable |

|  |  |
|---|---|
|  | 1 enable |
|  | 0 disable |
| EAN | analog output enable |
|  | 1 enable |
|  | 0 disable |
| E4 | ch.4 EAN, ECAL, ELK enable (a debugging function, not normally used) |
|  | 1 enable |
|  | 0 disable |

**DIE SIZE / PADS**

|  |  |
|---|---|
| Layout size | 6197µm x 3568µm |
| Requested max die cut size | 6500µm x 3800µm |
| Confirmed die cut size | 6384µm x 3788µm |
| Pads | 34 left + 34 right, none top or bottom |
| Pad opening | Input: 60µm x 70µm – Output 70µm x 100µm |
| Pad pitch | 106µm |
| Fabrication | TSMC 0.35µm through MOSIS multi-project. |

|  |  |
|---|---|
| Vddp | Vdd |
| I31 | Vss |
| I30 | CS |
| I29 | SDI |
| I28 | SCK |
| I27 | TCK |
| I26 | WR |
| I25 | EN |
| I24 | RST |
| I23 | Vddd |

| | |
|---|---|
| I22 | Vssd |
| I21 | Vdd |
| I20 | Vss |
| I19 | VH2 |
| I18 | VL2 |
| I17 | VH1 |
| I16 | VL1 |
| I15 | VL0 |
| I14 | BDAC |
| I13 | BLN |
| I12 | BLK |
| I11 | OAN |
| I10 | OLK |
| I9 | Vss |
| I8 | Vdd |
| I7 | CAL |
| I6 | Vsubd |
| I5 | SDO |
| I4 | TO |
| I3 | E4 |
| I2 | Vddd |
| I1 | Vssd |
| I0 | Vss |
| Vddp | Vdd |

**COMPONENT COUNT / STATIC POWER**

Bias, Global Settings + Pads:

| | |
|---|---|
| MOSFETs | 708 |
| BJTs | 5 |
| Capacitors | 28 |
| Resistors | 34 |
| Static power | 3.25mW |
| Bias area | 5854µm x 102µm |

Channel (Analog+Windows+Settings+Counters):

| | |
|---|---|
| MOSFETs | 359 (analog) + 1986 (windows) + 200 (settings) + 3176 (counts) = 5721 |
| Capacitors | 87 (analog) + 17 (windows) = 104 |
| Resistors | 32 (analog) + 10 (windows) = 42 |
| Static Power | 5.68mW (analog) + 1.53mW (windows) + 0.85mW (DACs) = 8.06mW |
| Channel area | 5854µm x 102µm |

Total ASIC

| | |
|---|---|
| MOSFETs | 183780 |
| BJTs | 5 |
| Capacitors | 3356 |
| Resistors | 1378 |
| Static Power | 261.2mW |
| Chip area | 6280µm x 3568µm |

# Appendix E

**NSLS Detector Record (v 0.91) and related software**

*D. Peter Siddons, NSLS, BNL*
*Based on the Scaler Record v3.2, written by Tim Mooney, APS*

## *Contents*

- Overview
- Field Descriptions
- Files, device-support
- Restrictions

## *Overview*

This documentation describes version 0.91 of the EPICS nslsdet record, and related EPICS software required to build and use it. This version of the record is compatible with EPICS 3.14.7, and is incompatible with any earlier version of EPICS. It has only been built and tested on a Linux platform for use in an RTEMS application running on a Motorola Coldfire CPU. The software occupies a complete application TOP directory, and expects the apropriate cross-compilers and cross utilities to be installed, and a working RTEMS installation to be in place.

The nslsdet record resides within the embedded IOC which controls the NSLS High Energy Resolution Multi-Element Silicon detector (HERMES-X). This IOC is tightly coupled with the specific hardware for this detector, and is based on a credit-card sized computer which employs a Motorola Coldfire 32-bit microprocessor, which has ethernet and serial ports, on-chip timers, digital I/O bits, and an industry-standard SPI port to which are attached D/A and A/D converters, controlled by auxiliary standard ao and ai records in the IOC database.

HERMES consists of 32 * N identical channels, where N is an integer specific to the particular implementation in use (in this case, 20), each containing a complete solid-state detector processing chain. Each chain consists of a low-noise charge-sensitive preamplifier, variable-gain shaping amplifier (0.75 or 1.5V / fC) with selectable shaping time (0.5, 1.0, 2.0, 4.0 microseconds), a set of 5 pulse-height discriminators arranged as one simple threshold and two window-discriminators or single-channel analysers (SCA's) and three 24-bit counters, one each for the threshold and two SCA's. A built-in time base generator provides counter gating intervals in 1ms increments.

The Nslsdet record provides support for a bank of three sets of 32 x N 24-bit counters controlled by a common start/stop mechanism, all contained within HERMES. All counters are simple up-counters. When the counters overflow, they saturate at the maximum possible 24-bit value (16,777,215).

The count time interval is provided by one of the Coldfire processor's on-chip counters. It counts down from the programmed value using a clock derived from the CPU system clock, having a period of 1ms. The fields

- TP (time preset)
- PR1 (scaler channel 1 preset value), and
- FREQ (clock frequency)

satisfy the following equation:

```
TP = PR1 / FREQ
```

FREQ defaults to 1000, and should not be changed by the user.

This is how the nslsdet record is intended to be used:

1. The user sets TP (time preset)
2. The user sets CNT to "Count" (1), initiating counting.
3. The counters count until TP has expired, whereupon the record resets CNT to "Done" (0), or the user manually resets CNT to "Done" (0), causing counting to stop.

This version of the nslsdet record maintains two counting modes: normal and background (also called AutoCount). Normal counting is controlled by the CNT ("Count/Done") field, the time preset, and other presets, as described above. Background counting is controlled by the CONT ("OneShot/AutoCount") field, normally ignores user preset values, and is intended to give the user continuous updates whenever the scaler would otherwise be idle. This device differs from the standard scaler record in that the hardware cannot be read on the fly, so the scaler values are only updated at the end of the count period (normal or background).

When CONT = "AutoCount" (1), and no normal counting operation is already in progress, the scaler waits for the AutoCount delay (DLY1); counts for the AutoCount period (TP1); and displays the result. If the AutoCount period (TP1) is less than .001 s, we default to the normal criteria for determining the count period. Background counting is interrupted immediately whenever a normal counting operation is started with the CNT field, and the results of a normal counting operation are held for a defined time (default 3 seconds) after the scaler finishes that operation before background counting begins again. The hold time can be changed from the Coldfire console by modifying an EPICS variable (nslsdet_wait_time). Background counting never affects the state of the CNT field, so this field can always be used to determine when a normal counting operation has begun and finished.

The nslsdet record is unlike most other EPICS records in that its processing is neither "synchronous" nor "asynchronous", as these terms are used in the EPICS Record Reference Manual. Currently, the PACT field is always FALSE after record processing has completed, even though the scaler may be counting. This means the record always responds to channel-access puts, and that counting can be stopped at any time. The record's forward link is not executed until the scaler has stopped counting.

## *Field Descriptions*

In addition to fields common to all record types (see the EPICS Record Reference Manual for these) the scaler record has the fields described below.

- Alphabetical listing of all fields
- Control/status fields:
- Data fields
- Link fields
- Miscellaneous fields
- Private fields

---

## *Alphabetical list of record-specific fields*

NOTE: Hot links in this table take you only to the *section* in which the linked item is described in detail. You'll probably have to scroll down to find the actual item.

| ***Name*** | ***Access*** | ***Prompt*** | ***Data type*** | ***Comment*** |
|---|---|---|---|---|
| AOEN | R/W | Analog out enable | SHORT | Select which channel's analog signal is output. |
| CARD | r | Card Number | SHORT | Which detector. Currently only 0 valid. |
| CHEN | R/W | Channel Enable | CHAR | 32 x N element array of channel enable flags (0=enabled) |
| CNT | R/W* | Count | RECCHOICE | (0:"Done", 1:"Count") |
| CONT | R/W* | OneShot/AutoCoun t | RECCHOICE | (0:"OneShot", 1:"AutoCount") |
| DLY | R/W | Delay | FLOAT | before counting |
| DLY1 | R/W | Auto-mode Delay | FLOAT | before counting |
| EBLK | R/W | Enable bias current | SHORT | For use in testing |
| EGU | R/W | Units Name | STRING | 8 characters, default 'counts' |
| FREQ | R/W | Time base freq | DOUBLE | Always 1000 |

| | | | | |
|---|---|---|---|---|
| | | | | |
| G1 | R/W | Gate master/slave select | FLOAT | For multi-scaler systems |
| GAIN | R/W | Amplifier gain | RECCHOICE | (0:High, 1:Low |
| LOEN | R/W | Leakage out enable | SHORT | Select which channel's leakage monitor |
| NCH | R | Number of Channels | SHORT | from device |
| NM1...NM3 | R/W | Scaler n name | STRING | user's field, default 'Threshold', 'SCA1' and 'SCA2' |
| INP | R | Input Specification | INLINK | link to hardware |
| PCNT | r | Prev Count | RECCHOICE | private |
| PREC | R/W | Display Precision | SHORT | num decimal places |
| S1...S3 | R | Scaler n (32 x N)-element array | LONG | Vectors containing counter data. |
| SHPT | R/W | Shaping time | RECCHOICE | (0:"4us",1:"2us",2:"1us",3:"0.5us") |
| SS | r | Scaler state | SHORT | state of hardware |
| T | R | Timer | DOUBLE | elapsed time. Not implemented. |
| TP | R/W | Time Preset | DOUBLE | preset time |
| TP1 | R/W | Time Preset | DOUBLE | preset time (auto mode) |
| TR1 | R/W | Trim DACS 1 | CHAR | 32 x N element array of trim dacs |
| TR2 | R/W | Trim DACS 2 | CHAR | 32 x N element array of trim dacs |
| TR3 | R/W | Trim DACS 3 | CHAR | 32 X N element array of trim dacs |
| TR4 | R/W | Trim DACS 4 | CHAR | 32 x N element array of trim |

| | | | | |
|---|---|---|---|---|
| | | | | dacs |
| TSEN | R/W | Test pulse enable | CHAR | 32 x N element array of test pulse enable flags |
| US | r | User state | SHORT | state of user request |
| VAL | R/W | Result 3-element array | DOUBLE | Sums of all individual channels |
| VERS | R | Code Version | FLOAT | code version |

Note: In the **Access** column above:

| | |
|---|---|
| R | Read only |
| r | Read only, not posted |
| R/W | Read and write are allowed |
| R/W* | Read and write are allowed; write triggers record processing if the record's SCAN field is set to "Passive." |
| N | No access allowed |

## ***Control/status fields***

| ***Name*** | ***Access*** | ***Prompt*** | ***Data type*** | ***Comments*** |
|---|---|---|---|---|
| CNT | R/W* | Count | RECCHOICE | (0:"Done", 1:"Count") |
| User sets this field to "Count" (1) to start counting. When a preset is reached, counting will stop, this field will be reset to "Done" (0), and the forward link will be fired. If this field is set to "Done" (0) while counting is in progress, counting will be stopped, the accumulated counts will be reported, and the forward link will be fired. | | | | |
| CONT | R/W* | OneShot/AutoCount | RECCHOICE | (0:"OneShot", 1:"AutoCount") |
| User sets this field to "AutoCount" (1) to enable background counting. (See also autocount delay (DLY1), autocount display rate (RAT1), and autocount time presetTP1.) | | | | |

| | | | | |
|---|---|---|---|---|
| G1 | R/W | Gate control | RECCHOICE | (0:"N", 1:"Y") |
| This field controls the sense of the counter GATE signal. If it is 0, then HERMES-X is generating the gate signal, and the GATE connector is an output. If it is 1, then some other scaler is generating the gate signal, and the GATE connector is an input. Currently, the functionality for G1=1 is not implemented, and its assigned value is ignored and assumed 0. | | | | |
| TP | R/W | Time preset | DOUBLE | |
| This field specifies for how long, in seconds, the scaler is to count if no other preset stops it. | | | | |
| TP1 | R/W | AutoCount Time preset | DOUBLE | |
| This field specifies the background-counting period in seconds. | | | | |
| TR1..TR4 | R/W | Discriminator trim DAC's | CHAR | |
| (32 x N)-element arrays of values used to correct offset errors in the comparators used to form the SCA's.. | | | | |
| DLY | R/W | Delay (sec) | FLOAT | |
| The delay, in seconds, that the record is to wait after CNT goes to 1 before actually causing counting to begin. | | | | |
| DLY1 | R/W | AutoCount Delay (sec) | FLOAT | |
| The delay, in seconds, between successive background-counting periods. | | | | |
| | | | | |

## ***Data fields***

| ***Name*** | ***Access*** | ***Prompt*** | ***Data type*** | ***Comments*** |
|---|---|---|---|---|
| FREQ | R/W | Time base freq. (EGU) | DOUBLE | |

| | | | | |
|---|---|---|---|---|
| The frequency (in Hz) of the clock signal counted by the timebase generator. The record uses this field to convert between time values (T and TP, in seconds) and the value to be loaded into the timebase generator. The default value is 1000, and it should not be changed. | | | | |
| S1...S3 | R | Scaler n value (EGU) | LONG | |
| The counts accumulated by the scalers. They are (32 x N)-element arrays, corresponding to the 3 counters in each channel. These are posted once after counting stops, either normal or background. | | | | |
| VAL | R | Sum of equivalent scalers (EGU) | LONG | |
| The sum of all Sn channels, i.e. the total counts VAL[n] accumulated by all the scalers Sn. It is a 3-element array. It is posted once after counting stops, either normal or background. | | | | |

## *Link fields*

| ***Name*** | ***Access*** | ***Prompt*** | ***Data type*** | ***Comments*** |
|---|---|---|---|---|
| INP | R | Input Specification | INLINK | This field specifies the hardware to be controlled. |

## *Miscellaneous fields*

| ***Name*** | ***Access*** | ***Prompt*** | ***Data type*** | ***Comments*** |
|---|---|---|---|---|
| NM1... NM3 | R/W | Scaler n name | STRING | Names the user has given to the 3 scaler arrays. Default values are "Threshold", "SCA 1" and "SCA 2". |
| PREC | R/W | Display Precision | SHORT | The number of digits to the right of the decimal that are to be displayed by MEDM and other channel-access clients. |

|  |  |  |  |  |
|---|---|---|---|---|
| EGU | R/W | Engineering Units | STRING | String sent to channel-access clients who ask for engineering units. Default "Counts". |
| VERS | R | Code Version | FLOAT | Version number of the nslsdetRecord.c code. |
| CARD | R | Card Number | SHORT |  |
| Not used. Must be set to 0. |  |  |  |  |
| NCH | R | Number of channels | SHORT | The number of channels actually supported by the underlying hardware, as reported by device support. |

## *Private fields*

| ***Name*** | ***Access*** | ***Prompt*** | ***Data type*** | ***Comments*** |
|---|---|---|---|---|
| PCNT | R/W | Previous count | RECCHOICE | (0:"Done", 1:"Count") |
| Previous value of CNT. |  |  |  |  |
| SS | r | Scaler state | SHORT | state of hardware |
|  |  |  |  |  |
| US | r | User state | SHORT | state of user's request |
|  |  |  |  |  |

## *Files, device support*

The following table briefly describes all of the files required to implement and use the scaler record. The reader is assumed to be familiar with the IOC Applications: Building and Source/Release Control document which describes how to build an EPICS application tree into which these files are to be placed, and how to run "gnumake" to build record and device support. These files can all be obtained from the EPICS Software Distribution (in the custom-software section).

| ***SOURCE CODE*** ***files to be placed in `<top>/<app>App/src/`*** | |
|---|---|
| nslsdetRecord.c | Record support for the nslsdet record |
| devNslsdet.h | Header included by record and device support |
| devNslsdet.c | Device support for HERMES-X |
| nslsdetRecord.dbd | This file defines all of the fields menus, etc. for the nslsdet record. |
| *Include.dbd | This file is not included in the distribution. However, the user must edit this file and add the following lines: `# Database definition for scaler record` `include "scalerRecord.dbd"` `# Device support for scaler record` `device(scaler,VME_IO,devScaler,"Joerger VSC8/16")` |

| ***MEDM DISPLAY SCREENS*** ***files to be placed in `<top>/<app>App/op/adl/`*** | |
|---|---|
| detector.adl | Main operator screen for HERMES-X |
| spi_dacs.adl | DAC controls for SCA and threshold windows etc. |
| spi_dacs2.adl | DAC controls for Peltier current, bias voltage etc. |
| spi_adc.adl | ADC readout, for temperature monitoring, bias current etc. (not yet implemented). |
| These files build `medm` screens to access the detector record and related process variables in the database included with this distribution. Shell scripts are provided to launch these screens from the command line. | |

| ***EPICS STARTUP FILES*** ***files to be placed in `/tftpboot/epics/det1/`*** | |
|---|---|
| st.cmd | Startup script |
| This file is included in the distribution. It loads support for the SPI DACs and ADCs, and also the HERMES-X detector. | |

```
## Example RTEMS startup script

## You may have to change cfMemMap to something else
## everywhere it appears in this file

## Register all support components
dbLoadDatabase("dbd/cfMemMap.dbd")

cfMemMap_registerRecordDeviceDriver(pdbbase)

#initialize cfSPI interfaces
devcfSPIConfig(0,1,8)
devcfSPIConfig(1,1,8)
devcfSPIConfig(6,1,8)
#and nslsdet
devNslsDetConfig(1,1,384)

## Load record instances
dbLoadRecords("db/cfSpiDacs.db","user=peter")
dbLoadRecords("db/cfSpiDacs2.db","user=peter")
dbLoadRecords("db/cfSpiAdcs.db","user=peter")
dbLoadRecords("db/det1.db")

## Start EPICS
iocInit()
```

## ***BACKUP/RESTORE (BURT) REQUEST FILES***
## ***files to be placed in `<top>/<app>App/op/burt/`***

| | |
|---|---|
| scalerSettings.req | sample request file to save settings of one scaler database. Edit this file, supplying names of the scaler records whose settings you want saved. (The sample file also saves the states of other records in the sample database, Jscaler.db, that control calculations done when the scaler finishes counting.) |
| yyScalerSettings.req | save settings of a specified scaler record. This file is `#include`'d (once for each scaler) by scalerSettings.req. |

These files tell the backup/restore tool how to save scaler settings. To use them from the command line, type, for example

```
burtrb -f scalerSettings.req -o myScalerSettings.snap
```

To restore the scaler settings saved by the above commands, type

```
burtwb -f myScalerSettings.snap
```

## *Restrictions*

When AutoCount is enabled, with a very short counting time (TP1) and delay time (DLY1), the scaler can swamp the network with data.

*Suggestions and comments to:*
*D. Peter Siddons : (siddons#bnl.gov)*
*Last modified: July 20th, 2004.*