

NAME

hecus – Hecus ASA-32 MCA

INTRODUCTION

The ASA-32 is the ISA acquisition card for the Hecus SAX/SWAX X-ray system. The system has two detectors for obtaining both small- and wide-angle data with the position-sensitive detectors. The system can be operated in a pure small-angle (SAX) mode or a multiplexed small/wide angle (SWAX) mode, where the time slots assigned to each detector are independently adjustable from 0.01 to 2 seconds. The ASA-32 card can return both position and energy data for each of the two detectors.

FUNCTIONS

Since the MCA has two detectors and both energy and position data arrays are available for each, the usual numbering scheme in `spec` for selecting MCA devices has been extended. To access other than the default detector (detector 1, the SAX detector) or the default data (energy), the `mca_sget()` and `mca_spar()` functions must be used rather than `mca_get()` and `mca_par()`.

Normally, the first argument of `mca_sget()` and `mca_spar()` selects which MCA device, when more than one are configured. The ASA-32 is treated as one MCA device, but the first argument usage is extended. Instead of a simple number, the first argument can be passed as a string that contains first the number of the selected MCA, then a colon, then a number that selects the detector and/or the type of data. The lowest order hex digit of that second number is zero for detector one or one for detector two. The next hex digit is zero or one to code for energy or position, respectively. Decimal values are valid, too. For example,

```
mca_sget("0:0x10", data)
```

or

```
mca_sget("0:16", data)
```

reads the position array for detector one, while

```
mca_sget("0:0x11", data)
```

or

```
mca_sget("0:17", data)
```

reads the position data for detector two. The command

```
mca_sget("0:0x00", data)
```

reads the energy data for detector one, as does

```
mca_get(data)
```

With respect to parameters that can apply to either detector, a command such as

```
mca_spar("0:0x1", "energy_lower")
```

or

```
mca_spar("0:1", "energy_lower")
```

would return the parameter value for detector two.

The parameters where a detector designation makes a difference are:

```
"preset", but only in energy- or position-stop mode,  
"mplex_time",  
"energy_lower" and "energy_upper",  
"ADC_counts", "energy_counts", "position_counts" and "trigger_counts".
```

The standard MCA commands supported by `spec` are described in the `mca` help file. Of the standard functions, the ASA-32 MCA supports `mca_get()` but not `mca_put()`. The standard `mca_par()` options "info", "chans", "max_chans", "max_channels", "disable", "auto_run", "soft_preset", "auto_clear", "native_type", "preset", "run", "halt", and "clear", are supported by the ASA-32 MCA.

For options that apply to the "selected detector", detector 1 (the SAX detector) is selected by default, and the `mca_spar()` first argument can be used to select detector 2, as explained above.

`mca_par("preset")` – Returns the preset value for the current stop mode and the selected detector. The preset time parameter is returned in real-time and live-time stop modes. The preset count parameter is returned in energy and position stop modes.

`mca_par("preset", value)` – Sets the preset value for the current stop mode. For real-time and live-time stop modes, sets the number of seconds for acquisition. For energy and position stop modes, sets the total counts for the selected detector at which to stop acquisition.

`mca_par("mode")` – Returns either of the strings "sax" or "swax" to indicate whether the device is in pure SAX mode, or multiplexed SWAX mode.

`mca_par("mode", value)` – Sets the acquisition mode to SAX or SWAX respectively for the arguments "sax" and "swax".

`mca_par("mplex_time")` – Returns the time in seconds devoted to the selected detector in the multiplexed SWAX mode.

`mca_par("mplex_time", value)` – Sets the selected detector multiplex time to the specified value, which can be from 0 to 2 seconds, in increments of 0.01 second.

`mca_par("stop_mode")` – Returns one of the strings "realtime", "livetime", "energy" or "position" to indicate the current stop mode. Acquisition halts when the specified acquisition value reaches the preset value.

`mca_par("stop_mode", value)` – Sets the stop mode according to the argument, which can be "realtime", "livetime", "energy" or "position".

`mca_par("energy_lower")` – Returns the lower energy limit for the selected detector.

`mca_par("energy_lower", value)` Sets the lower energy limit for the selected detector to `value`. The valid range is from 0 to 4095. The default value is 2048.

`mca_par("energy_upper")` – Returns the upper energy limit for the selected detector.

`mca_par("energy_upper", value)` – Sets the upper energy limit for the selected detector to `value`. The valid range is from 0 to 4095. The default value is 3072.

`mca_par("high_voltage")` – Reports the status of the high-voltage supply. A positive return value indicates the supply is on.

`mca_par("high_voltage", value)` – Turns the high-voltage supply on if `value` is nonzero. Turns the high-voltage supply off if `value` is zero.

`mca_par("ratemeter")` – Returns either of the strings "trigger" or "position" to indicate the mode of the rate-meter output.

`mca_par("ratemeter", value)` – Sets the mode of the rate-meter output according to the argument which can be "trigger" or "position".

`mca_par("ADC_counts")` – Returns the total counts converted by the analog-to-digital converter.

`mca_par("energy_counts")` – Returns the total counts collected into the energy array.

`mca_par("position_counts")` – Returns the total counts collected into the position array.

`mca_par("trigger_counts")` – Returns the number of counts seen by the input trigger.

`mca_par("elapsed_time")` – Returns the elapsed time of acquisition.

`mca_par("dump")` – Displays current device settings for high-voltage status, ratemeter, position resolution and energy resolution.

`mca_par("reset")` – Resets the board, forcing a reload of the firmware. Currently nonfunctional.

`mca_par("min_delay")` – Returns the value for the parameter that is the minimum time between two pulses from the detector to be counted as separate pulses.

`mca_par("min_delay", value)` – Sets the above parameter to *value*. Note, this parameter is a constant for the detector and should not need to be changed. The default value, appropriate for the PSD-50M, is zero.

`mca_par("time_correction")` – Returns the value for the detector dead-time parameter.

`mca_par("time_correction", value)` – Set the above parameter to *value*. Note, this parameter is a constant for the detector and should not need to be changed. The default value, appropriate for the PSD-50M, is 0x573.