

NAME

Radicon – Radicon SCSD-3C scintillation detector with timer/counter

DESCRIPTION

`spec` supports the Radicon model SCSD-3C scintillation detector with embedded timer/counter over a serial interface. See www.radicon.spb.ru for details on the hardware.

The SCSD-3C device has one counter channel and can be operated in either preset-time mode or preset-count mode. In preset-time mode, the accumulated count value is available. In preset-count mode, the elapsed time is available. In both cases, results are only available after the preset is reached. No intermediate results are accessible. If the counting is interrupted, no hardware results are available. While counting or after an interrupted count, `getcounts` will fill the `S[sec]` counter array element with the estimated elapsed time.

The counter register is 32 bits. In preset-time mode, the register contents are scaled by $8e6$, so that the maximum programmable count time is just over 536 seconds. If the requested count time is longer than that, `spec` will automatically restart the counter as many times as necessary. In preset-count mode, if the time to accumulate the specified preset exceeds the hardware's maximum, the hardware counter overflows and loses the accumulated counts. `spec` will print an error message and set the elapsed time to 536.871 seconds and the elapsed counts to zero.

CONFIGURATION

The SCSD-3C should be configured for two channels. Channel 0 should be the time base and channel 1 should be configured as the monitor counter. The scale factors are built into the code for both channels ($1e6$ for the time channel and 1 for the monitor). Values in the `config` file are ignored.

PARAMETERS

The SCSD-3C has several configuration parameters. Although the parameters affect the controller as a whole, they should be configured for one of the individual channels. If the same parameter is given different values for both channels, the last parameter input will take precedence.

The parameters can be configured using `counter_par()` or as nonstandard optional parameters in the `config` file. To configure the latter, type a `p` while on the **Scaler** screen and enter the parameter name and value as described on the associated help screens. Values set using `counter_par()` will be reset to those in the `config` file on start up and on `reconfig`.

The following parameters are available:

`counter_par(mne, "high_voltage" [, arg])` – Returns or sets the high voltage value sent with the start command. Value should be from 0 to 818 volts.

`counter_par(mne, "use_sound" [, arg])` – Returns or sets the mode where the instrument generates sounds that reflect the intensity of the incident flux.

`counter_par(mne, "use_window" [, arg])` – Returns or sets windowed mode. When enabled, the device uses the upper threshold values to discriminate the input counts.

`counter_par(mne, "upper_threshold" [, arg])` – Returns or sets the value for the upper threshold in windowed mode. Value should be from 0 to 1023.

`counter_par(mne, "lower_threshold" [, arg])` – Returns or sets the value for the lower threshold. The lower threshold is always enabled and is used to cut off thermal noise and the intrinsic detector noise. Value should be from 0 to 1022.

`counter_par(mne, "window" [, arg])` – Returns or sets the value for the threshold window. Setting a value changes the upper threshold parameter. Will silently set the upper threshold to the maximum value if the window is too large. Note, the window

value equals upper – lower + 1.

The following commands are also available:

`counter_par(mne, "reset")` – Sends the reset command to the device.

`counter_par(mne, "sound_on")` – Sends the sound-on command to the device and turns on the "use_sound" mode.

`counter_par(mne, "sound_off")` – Sends the sound-off command to the device and turns off the "use_sound" mode.

`counter_par(mne, "dump")` – Displays the current values for the parameters formatted as follows:

```
upper_threshold : 800
lower_threshold : 204
    window      : 597
high_voltage    : 642
    use_sound    : 1
    use_window   : 1
```