

**NAME**

star1 – Photometrics STAR 1 CCD Camera and Controller

**DESCRIPTION**

The Photometrics STAR 1 System is supported as a GPIB device. The controller is selected in the *config* file with the line

```
GP_STAR1 = gpib_address
```

When running the configuration editor, switch to the device configuration screen to select the STAR 1 as an MCA device.

**PARAMETERS AND CONTROLS**

The `mca_par()` function controls the STAR 1 behavior as follows:

`mca_par("auto", value)` – Selects whether the lookup table will be automatically adjusted by the STAR 1 after each exposure. Valid arguments are 0 to turn auto mode off and 1 to turn it on.

`mca_par("auto")` – Returns the current auto mode (0 or 1).

`mca_par("bias")` – Initiates a bias integration. All charge is cleared from the CCD, and then it is immediately read out. Provides measure of the DC offset and inherent structure of the CCD array. Resets the ROI to the entire image area.

`mca_par("bins", value)` – Sets the binning value. Pixel values in the specified number of adjacent columns will be combined with the resulting image compressed horizontally.

`mca_par("bins")` – Returns the binning value.

`mca_par("clear")` – Clears the video image. Resets the ROI to the entire image area.

`mca_par("dark")` – Initiates a dark integration for the programmed exposure time. Resets the ROI to the entire image area.

`mca_par("dx", value)` – Sets the width of the box cursor.

`mca_par("dx")` – Returns the width of the box cursor.

`mca_par("dy", )` – Sets the height of the box cursor.

`mca_par("dy")` – Returns the height of the box cursor.

`mca_par("expose")` – Opens the shutter for the programmed exposure time. Resets the ROI (region of interest) to the entire image area.

`mca_par("expose_time", value)` – Sets the exposure time in units of tenths of seconds. Valid arguments are between 1 and 999 (0.1 and 99.9 seconds).

`mca_par("expose_time")` – Returns the current exposure time.

`mca_par("focus")` – Takes one 100x100 pixel exposure from the center of the image area. (Uses the *calibrate* STAR 1 command, rather than the *focus* command, as the latter starts a continuous acquisition mode that can't be halted from the remote computer.)

`mca_par("gain", value)` – Sets the gain. Valid arguments are 0 and 1, which set  $\times 1$  and  $\times 4$  gain, respectively.

`mca_par("gain")` – Returns the gain value (0 or 1).

`mca_par("getbox")` – Updates the retrievable box cursor coordinates within the display.

`mca_par("linear_lut")` – Creates a linear lookup table using the "lut\_min" and "lut\_max" values.

`mca_par("load")` – Reads in the image from the current data file and sends it to the STAR 1 controller to be displayed on the video monitor. Only full-screen images saved in the

"raw" format can be read and loaded.

`mca_par("lut#", value)` – Sets element number # of the user-defined lookup table to *value*. The user-defined lookup table is sent to the STAR 1 using the "send\_lut" command. Valid element numbers are from 0 to 4095. Valid lookup table values are from 0 to 255.

`mca_par("lut#")` – Returns the value of element # of the user-defined lookup table.

`mca_par("lut_max", value)` – Sets the maximum pixel value for the "linear\_lut" command. Valid arguments are from 0 to 4095.

`mca_par("lut_max")` – Returns the maximum pixel value.

`mca_par("lut_min", value)` – Sets the minimum pixel value for the "linear\_lut" command. Valid arguments are from 0 to 4095.

`mca_par("lut_min")` – Returns the minimum pixel value.

`mca_par("mouse_x")` – Sets the delta *x* value for the next cursor movement. The cursor is moved with the "mouse\_y" command.

`mca_par("mouse_y")` – Sets the delta *y* value for cursor movement, and moves the cursor using this value and the value set with the "mouse\_x" command.

`mca_par("roi", value)` – Sets the number of ROIs for the "roi\_acquire" command.

`mca_par("roi")` – Returns the number of ROIs.

`mca_par("roi_acquire")` – Initiates acquisition of the region defined by the box cursor.

Will acquire multiple regions if the "roi" value is greater than one and the size of the box cursor allows more than one region to be displayed on the video monitor.

`mca_par("roi_delay", value)` – Sets the delay (in tenths of a second) between ROI acquisitions when the "roi" value is greater than one, and the "roi\_acquire" command is sent.

`mca_par("roi_delay")` – Returns the ROI delay.

`mca_par("save")` – Reads the image from the STAR 1 (either the entire image area, or the ROI array(s), depending on the last used acquisition command), and writes to the current image file, using the current image format and the current save image parameters. Only one save image is allowed per file. The existing contents of the file will be erased.

`mca_par("save_using_lut")` – As above, but saves the image using the user-defined lookup table to transform pixel values.

`mca_par("send_lut")` – Sends the user-defined lookup table to the STAR 1. Element 0 is first set to zero and element 4095 to 255 in order to preserve the contrast in the annotation text of the display.

`mca_par("setbox")` – Displays the box cursor.

`mca_par("shade")` – Displays a test image on the video monitor.

`mca_par("trigger")` – Opens the shutter and continually clears the CCD array of charge until an external trigger signal is received. There is no way to reset the shutter from the remote computer if an external trigger signal is not received by the STAR 1.

`mca_par("x0", value)` Sets the lower left *x* coordinate of the box cursor.

`mca_par("x0")` – Returns the value of the lower left *x* coordinate of the box cursor.

`mca_par("y0", value)` Sets the lower left *y* coordinate of the box cursor.

`mca_par("y0")` – Returns the value of the lower left *y* coordinate of the box cursor.

`mca_par("zoom", value)` – Sets the zoom factor. Valid values are 0, 1 and 2 for  $\times 1$ ,  $\times 2$  and  $\times 4$  zoom factors, respectively.

`mca_par("zoom")` – Returns the current zoom value (0, 1 or 2).

`image_par("bits", value)` – Sets the number of bits per pixel to use when the image is saved to a (tiff) file. Valid arguments are 4, 8 or 12.

`image_par("bits")` – Returns the bits-per-pixel value.

`image_par("format", type)` – Sets the save file format to be used for the image data file. Valid formats are currently "tiff" and "raw".

`image_par("format")` – Returns the current save file format.

`image_par("file", file_name)` – Sets the data file name to be used for the next "save", "save\_with\_lut" or "load" command.