# MMPAD User Manual

# Overview

The MMPAD is a 384 x 256 element Pixel Array Detector based on the MMPAD x-ray detector chip developed by the Cornell X-ray Detector Group in collaboration with Area Detector Systems Corporation. The total charge per exposure is integrated.

The sensor is a 500 micron thick, fully-depleted silicon diode array bonded pixel by pixel to an underlying CMOS readout chip.

The detector has both high sensitivity (~ 1 keV read noise equivilent) and a high dynamic range (30 bits / frame).

Integration time per image is user programmable from 5 microseconds to >> 1 second. Frame readout take 860 microseconds, during which no integration of signal occurs.

Maximum rate for signal is ca. 4 \* 109 keV/s/pixel (roughly 4\*108 10 keV x-rays/s/pixel). Above this rate, signal will be lost.

System gain is roughly 0.75 keV per ADU (a better number will be forthcoming)

# Operation

# Camserver - talks directly with the hardware

To Start:

Open terminal window

**cd /home/username/tvx\_64/tvx/camera/camserver**

**./camserver**

**padcom power\_init -** (issue this **only** on the first connection after FPGA was powered up from off state)

camserver is now waiting for connection from TVX.

Camserver commands from TVX will show in camserver window as they are executed.

Note debugging commands could be typed in camserver window, but routine operation will be from TVX window.

# TVX - MMPAD-CdTe

## TVX is a tool to control image taking and provide display with some analysis of images.

**To Start:**

Open terminal window. Move to your desired data path.

**cd** /datapath/subpath

Create directory if needed with mkdir /datapath/subpath

**rtvx.sh**

A new TVX terminal window opens - NOTE commands are case INSENSITIVE - filenames are case sensitive

|  |  |
| --- | --- |
| **TVX startup and quick guide:**  **TVX command** | **Description** |
| * **startup** | connects to camserver program and initializes hardware |
| * **poweronv2** | turns on chip power =(**cam** **ldcmndfile ./mmpadpower\_v2.cmd**) |
| * **cam hvon** | turns on detector bias |
| * **resethv** | Sets HV to -2V for 60 s, then returns to +400V and waits 60 s to settle |
| * **settaken** exp\_time  added\_time  n | exp\_time - exposure time (in seconds) - note 858 microseconds overhead will be added for readout  added\_time in seconds - **total frame time = exp\_time + added\_time + 858 us**  n is the number of frames to take with a **taken** command |
| * **save\_blocks** | save data streams as raw 32 bit floats - one file per scan. **save\_tifs** will save frames as individual tif files |
| **\*\*\*\*\* Close x-ray shutter \*\*\*\*\*** | |
| * **bgmode** | software trigger mode for backgrounds |
| * **AvgNBkg** filename number\_to\_average | Take a new set of background images - take a new set (20-200 images) any time expt is changed |
| To align system - turn on videomode |  |
| * **von** | Turn on videomode - open x-ray shutter as needed - data not saved to disk |
| grab display window commands ([see below](#TVXMMPADCdTe-grabdisplay))  **log / lin**  **u / d**  **lu / ld** | change grab display to log or linear u (or d) = increase linear display range by 2x (or 1/2x) lu (or ld) = increase log display range by 10x (or 0.1x) |
| * **voff** | Turn off videomode |
| to capture last videomode frame   * **cam padcom storebuffers /path/filename tif\_last** | saves the last captured image from videomode - specify full path for this command |
| * **disp** /path/filename\_last.tif | [displays](#TVXMMPADCdTe-imagedisplay) image in a TVX window |
| choose box or annulus cursor in display window    -   setlect ROI with cursor **(Optional)** | The cursor type in the display window should match the desired ROI type |
| * **SetROI** roi\_type roi\_num         ([see below](#TVXMMPADCdTe-SetROI) for full syntax) | This passes ROI coordinates back to camserver for on-the-fly scan computation roi\_type = box\_ circle\_ annulus\_ all\_ (append quad\_ lr\_ ud\_ cross\_ optionally) roi\_num = 0 - 3 |
| set trigger mode |  |
| * + **bgmode** | software triggers - (cam padcom settrigger 4 0 0) |
| * + **hw\_mode1** | hw trigger - 1 trigger per **taken** - (cam padcom settrigger 2 0 0;cam padcom mtrigger 1 1) |
| * + **mtrigs**=m   + **hw\_modem** | hw\_trigger - m triggers per **taken** - (cam padcom settrigger 2 0 0;cam padcom mtrigger 1 [mtrigs])  if settaken time 0 N, then N x mtrigs frames will be output with **taken** (total must be < 40000 frames) |
| * **taken** filename | Takes a set of frames according to parameters given in **settaken** |
| * **avgn** filename  #\_frames\_to\_average | Takes a set of frames according to exposure time set by settaken but overrides # of frames  Stores individual images and filename\_avg.tif |

# Most common TVX commands - by Category:

|  |  |
| --- | --- |
| **Basic Functions** |  |
| **von / voff** | Turn videomode on/off - files are not stored to disk (but see [storebuffers](#TVXMMPADCdTe-storebuffers) below) |
| **setTakeN** exposuretime addedtime number\_frames | sets image exposure parameters        exptosuretime is the exposure time in seconds - (also sets variable expt)       addedtime is the additional frame time in seconds -(sets variable framet)            normally set to 0 unless you want to add extra delay between frames       number\_frames is the number of frames to take in sequence (1 - 10000) |
| **TakeN** imagename  **AvgN** imagename #frames  [*Notes on image names (below)*](#TVXMMPADCdTe-ImageNames) | Take and store a series of images - filename sets base filename |
| **disp** imagename | display image - create a new window if needed (up to 3 windows) |
| **disp1** imagename | displays using last window |
| **save\_blocks** | save one file per scan - multiple images per file - raw 32 bit floats |
| **save\_tifs** | save one file per scan point - not available for more than 100 \* 100 scan points |
| **VARIABLE**=value  **VARIABLE** | change value of tvx variable ( types = string / INT / FLOAT )  prints the value of the variable |
| **Variable Types:**   * Predefined:      TVXUserVAR = value * UserDefined:    define UserVAR = value * Temporary:      TEMPVAR = value | * built in variables - direct ties to various tvx functions * define a user variable which remains defined for the tvx session can be reset * if not previously defined - a temp variable which goes away after the current command line |
| **exposepath** /path\_for\_exposures/ | Set storage path for new exposures (default = path where rtvx.sh was started)      can be over-ridden by using explicit path with filename (e.g. TakeN /path/file) |
| **imagepath** /path\_for\_reading\_images/ | Path for reading images (default = path where rtvx.sh was started) |
| **move** im\_dest = im\_s1 OP im\_s2 **move** im\_dest=im\_s **move** im\_dest=im\_s OP CONSTANT **move** im\_dest = CONSTANT | General image manipulation  OP = + - \* /  .... |
| **Integrate** filename | Cursor specific integration Box cursor - gives min/max/sum/avg/st. dev Butterfly cursor - Wedge/Line integration |
| **Grab Display** |  |
| **log / lin** | change live grab display to change to log or linear scale |
| **lu / ld**  **u / d** | Change max log value to display ( lu =increase by 10x   ld = scale by 0.1x )  Change linear scale 2x ( u = 2x   d = 0.5x ) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TVX Image display** | |  | |  | | |
| **disp** filename low high scale | | opens a display window for interactive image viewing - creates up to 3 windows | | **low** = bottom value of display  **high** = top value to display  **scale** = power/15 for LUT (scale =15 is linear ) | | |
| **disp1** filename low high scale | | uses last image display window | |
|  | Display window has sliders to adjust low/high/scale and a 4th which scales #s > 65k A Zoom window can be opened Change grey/color/reverse | | |
|  | Cursors | Pointer | | Examines pixel values | | |
| Box | | define a box ROI | | |
| Annulus | | Define a circle/annulus ROI | | |
| Butterfly | | Wedge/Line ROI | | |
| **ROI usage** | | **roi\_type** | **type\_modifier** | | **roi\_num** | **Coordinates** |
| **SetROI roi\_type\_ *[type\_modifier*] roi\_num [coord1 coord2 coord3 coord4 ]** | | **circle\_** | * quad\_       - store quadrant info * lr\_            - store left half - right half * ud\_          - store upper half - lower half * cross\_     - store (UL+LR) - (UR+LL) * inv\_         - use pixels outside of region * nosum\_    - choose to disable sum store   any or all can be added to basic type | | 0 - 3  -1 = turn off all | center\_x center y radius |
| Coordinates can be given on command line                                   - or -    Coordinates can be taken from display cursor in TVX       Choose: **box cursor** (for box )                      **annulus cursor** (for circle or annulus) | | **box\_** | center\_x center\_y del\_x del\_y |
| **annulus\_** | center\_x center\_y radius\_1 radius\_2 |
| **all\_** | center\_x center\_y |
| Computed numbers get stored in metadata on last line of image [TakeXYScan](#TVXMMPADCdTe-xyscan) filename will create a scan image for sum, quadrants, differences as specified in type\_modifiers | | | | | | |
| **Examples:**  setroi circle\_lr\_ud\_ 0 56 85 10  setroi box\_ 1 setroi annulus\_quad\_ 2 55 72 10 15  define ROI # 0 to be a circle of radius 10 at (x,y) of (56,85) - compute left-right halves and up-down halves of this region as well define ROI # 1 to be a box using coordinates taken from TVX image display cursor define ROI # 2 to be an annulus of inner radius 10 and outer radius 15 at (55,72) - store quadrant info as well | | | | | | |

## cam padcom storebuffers parameters

***NOTES***:

* **save\_blocks must be on**
* **this command needs the explicit path in /path/filename**

|  |  |
| --- | --- |
| **parameters** |  |
| /path/filename **block\_all** | Uses last run takeXYScan parameters to determine buffers to output |
| /path/filename **block\_range** low high | Outputs buffer numbers low to high, inclusive |
| /path/filename **block\_single** frame\_number | Outputs a single buffer to the file |
| /path/filename **tif\_range** low high | Output a series of tiff files from frame buffer low to high |
| /path/filename **tif\_single** frame\_number | Output a single tiff file from frame buffer |
| /path/filename **tif\_last** | Output the last filled frame buffer  **Useful for outputting the last frame from von (if save\_blocks is on)** |
| /path/filename **tif\_all** | Output all frame buffers as tiff files |

## Notes on Image names

### Image name in tvx have some restrictions

* + Start with an Alpha character
  + No spaces
  + Underscore ( \_ ) is the only special character allowed
* /exposepath/ automatically added to exposure filenames as passed to camserver (which could be running on a different PC)
* /imagepath/ automatically added to display filenames

### Construction of filenames stored on disk.

|  |  |  |
| --- | --- | --- |
| **File Format** |  |  |
| save\_tifs | One image per file | 32 bit float with TIFF header |
| taken FileName | File(s) stored as |  |
| num\_fr = 1 | /exposepath/FileName.tif |  |
| num\_fr = m > 1 | /exposepath/FileName\_nnnnn.tif | nnnnn goes from 00000 to m (with leading zeros) |
| AvgNBkg FileName m | /exposepath/FileName\_nnnnn.tif  /exposepath/FileName\_avg.tif | nnnnn goes from 00000 to m (with leading zeros)  The average of the m files in the series      This file gets loaded as the new background to subtract |
| AvgN Filename N | /exposepath/FileName\_nnnnn.tif  /exposepath/FileName\_avg.tif | nnnnn goes from 0 to N-1 |
| save\_blocks | one file per series | 32 bit float \* 396 \* 266 pixels  \* number of frames  - little endian  * **first row is metadata** |
| taken FileName | /exposepath/FileName\_xxxxx.raw | where xxxx is num\_fr |
| AvgNBkg FileName m | /exposepath/FileName\_xxxxx.raw  /exposepath/FileName\_avg.tif | where xxxx is m  The average of the m files in the series      This file gets loaded as the new background to subtract |
| Avgn Filename N | /exposepath/FileName\_xN.raw  /exposepath/FileName\_avg.tif |  |