**hyperdaq.processing**

processing.py

hyperDAQ module for processing incoming data

Last Updated: January 2020

Trevor Arp

Gabor Lab

University of California, Riverside

All Rights Reserved

***class*hyperdaq.processing.Ringbuffer2D(*bufflength=100000*)**

Bases: **object**

A simple and efficient ring buffer for buffering 2D data from a larger numpy array The first index of the array is assumed to be a time coordinate Buffer\_time\_data.num is the number of elements in the array

**get()**

Returns the first element in the Queue

**getn(*number*)**

Returns the first number of elements from the buffer

**put(*x*, *ix*)**

Takes a multidimensional numpy array x and inserts x[:,0] and x[:,ix] into the buffer

***class*hyperdaq.processing.data\_image\_2(*data\_queue*, *channel\_ix*, *scan*, *initdata=None*, *message\_queue=None*)**

Bases: **threading.Thread**

Data Image class that processes Queued data from the DAQ card into a 2D image, taking the data in chunks not in a time series.

To change how data points in the image are inherit this class and override the self.process(x,y,data) function which processes a numpy array of data and assigns the value to (x,y) in the image. To change the way that the data from the line is processed override self.update\_line(data) where data is the raw\_input data array from the card.

**Parameters**

* **data\_queue** – the Queue containing data from the card
* **channel\_ix** – the column index of the data arrays (from the card) that is being tracked
* **scan** – the Scan object that is controlling the measurement

**data**

The current data image. READ ONLY

**current\_y**

The current line of data. READ ONLY

**current\_x**

The x axis points for the current line of data. READ ONLY

**process(*x*, *y*, *data*)**

Processes the data for the point at (x,y) and updates the image. Simply averages, but can be overridden for more complex processing

**Parameters**

* **is the x coordinate of the point to be updated in the image** (*x*) –
* **is the y coordinate of the point to be updated in the image** (*y*) –
* **is the data array, [time, data] to be processed** (*data*) –

**resize()**

Resizes the data image

**run()**

Main loop of thread, updates image

**scan\_update\_line(*y*)**

Updates the line plot with current data during a scan being scanned,

**Parameters**

**y** – The y-coordinate of the current data point i.e. the line currently being scanned

**set\_data\_queue(*data\_queue*)**

changes the data Queue

**set\_scanner(*scan*)**

Changes the scan object

**stop()**

Stops the process

**update\_line(*data*)**

Updates the current values while not writing into the data array matches in function with update\_line from previous iterations

**hyperdaq.processing.extract\_all(*q*)**

Removes all elements from the given data queue, returns data in a single array - q is the input queue, containing data in numpy format