**hyperdaq.utilities**

utilities.py

Various useful functions for hyperDAQ

Last Updated: January 2020

Trevor Arp

Gabor Lab

University of California, Riverside

All Rights Reserved

***Class* hyperdaq.utilities.Stopwatch**

Bases: **object**

A stopwatch for timing peripheral functions (i.e. those not from the DAQ card). Uses system time but can be synchronized by calling the zero() method.

11/8/2017 Replaced time.clock with time.perf\_counter, due to deprecation of time.clock in python 3

**time()**

**zero()**

**hyperdaq.utilities.brute\_diff\_min(*d1*, *d2*, *maxdrift=15*)**

Brute Force approach to reflection image drift correction, finds the minimum difference between the two (normalized) images by brute force.

**Parameters**

* **d1** (*numpy array*) – the prime image
* **d2** (*numpy array*) – the image to correct
* **maxdrift** (integer) – the maximum number of pixels to consider (*default:15*)

**hyperdaq.utilities.changeEntry(*entry*, *v*)**

Takes a Tkinter Entry object, deletes the old entry and inserts the given value

**Parameters**

**entry** (*tkinter entry*) – The new tkinter entry.  
**v** (*??*) – The value to be inserted.

**hyperdaq.utilities.checkEntryStar(*entry*)**

Checks if the value of a tkinter entry contains ‘\*’, returns ‘\*’ is true, returns the string if false

**Parameters**

**entry** (*tkinter entry*) – The tkinter entry to be checked.

**hyperdaq.utilities.dequeue\_all(*q*)**

Removes all elements from the given data queue, returns data in a single array

**Parameters**

**q** (*Queue*) – the input queue, containing data as numpy arrays

**hyperdaq.utilities.dequeue\_n(*q*, *n*)**

Removes n elements from the given data queue, and concatenates them. Will return early if the queue is empty, even if n elements have not been collected

**Parameters**

**q** (*Queue*) – the input queue, containing data as numpy arrays

**hyperdaq.utilities.dequeue\_str(*q*)**

Removes all elements from the given data queue

**Parameters**

**q** (*Queue*) – the input queue, containing data as strings

**hyperdaq.utilities.hl2int(*d*)**

Converts a high and low byte into an integer. Input is [high\_Byte, low\_Byte]

**hyperdaq.utilities.indexof(*list*, *item*)**

Searches list and returns the index of item. Returns -1 if it can’t find it. Compares as strings

**hyperdaq.utilities.isPosInt(*n*)**

Returns true is the number is a positive integer, false otherwise

**hyperdaq.utilities.remove\_values\_from\_list(*the\_list*, *val*)**

Removes all instances of a value from a list

**hyperdaq.utilities.tryfloat(*value*)**

Attempts to convert input to a float, returns None if it cannot

**hyperdaq.utilities.type\_index(*type*)**

Searches the global parameter DATA\_file\_types and returns the index corresponding to the input type. If it can’t find the type returns -1