Keithley6487-IOC

This IOC controls Keithley picoammeter Model 6487

The IOC gives the user full control over what filters to turn on, number fo samples to consider per time step, and the time per sample step.

PV List

The following are some of the important PVs ‘[1]‘:

PV Name	Description
K6487:1:ZeroCheck	Disable/Enable zero check (input clamp)
K6487:1:AutoZero‘[2]‘	Disable/Enable Auto Zero
K6487:1:AutoRange	Disable/Enable auto-range
K6487:1:IntegrationTime	Number Of Power Line Cycles‘[3]‘ (**NPLC**) (Range: 0.01 - 50)
K6487:1:AverageFilterEnable‘*‘	Disable/Enable Average filter
K6487:1:AverageFilterCount	Number of samples per time step (Range: 2 - 100)
K6487:1:MedianFilterEnable‘*‘	Disable/Enable Median filter
K6487:1:MedianFilterRank	Median Filter Rank for 2n+1 samples (n range: 1 - 5)
K6487:1:FilterType	Either Mov or REP - default is REP
K6487:1:Damping‘[4]‘	Disable/Enable damping

K6487:1:TimePerSampleStep	Time Per Sample Step
K6487:1:NumSampleSteps	Number Of Sample Steps

*(1) All the PVs above has readback version. Just add **RBV** at the end of PV name (e.g. K6487:1:ZeroCheckRBV)*

(2) Auto Zero Helps maintaining stability and accuracy over time and changes in temparature. When enabled, it reduces measurement speed by a factor of 3

(3) NPLC standard ranges are (fast: 0.1, medium: 1, slow: 5), where recommended values are in the range between 1 and 10

(4) Damping reduces noise from input capacitance. Maybe caused by long input cable or the capacitance of the source

(*) Average and Median filters stabilizes noise measurements caused by noisy input signals

Calculations

Some of the PVs above are connected with the following equation:

‘$3 * \frac{NPLC}{50} * (NumSamples * 1.0285) + \epsilon$‘

where ‘$\epsilon = 0.0076172$‘

Note That the green section collapses to 1 when average filter is disabled.

Also the factor 3 ‘$(in\ red)$‘ collapses to 1 when AutoZero is disabled.

Aqcuiring Data

The PV that fires the calculation process is K6487:1:Acquire.

All you have to do is: **caput K6487:1:Acquire.PROC 1**

All filters that are enabled will be used in this process. For example if both *median* and *average* filters are enabled, then the *median* filter operation is performed first. After the *median* filter yields a reading, it is sent to the stack of the *average* filter.

After the process finishes you can read the result by running: **caget K6487:1:Acquire**

Reference

For **full** reference, you can refer to the documentation in project source