QEDistribution Widget

Andrew Starritt

25th April 2019

Contents

Introduction

This document describes in detail the QEDistribution widget which is an EPICS aware widget provided by the EPICS Qt, aka QE, Framework.

This document was created as a separate widget specification document. The main reason for this is ease of maintenance and avoiding editing large and unwieldly word documents.

The QE Framework is distributed under the GNU Lesser General Public License version 3, distributed with the framework in the file LICENSE. It may also be obtained from here:
http://www.gnu.org/licenses/lgpl-3.0-standalone.html

Description

The QEDistribution widget provides both graphical and textual information about the statistical variation of a single scalar Process Variable (PV). See examples in Figure 1 and Figure 2 below.

Upon receiving a new value for the PV, the QEDistribution widget performs the following actions:

a) The value is stored. The number of values retained to provide the statistical information is not restricted, and is ultimately limited by the amount of available host memory.
Values with an alarm severity of INVALID are ignored for statistical analysis purposes.

b) The mean, minimum, maximum and standard deviation of the accumulated data is evaluated and displayed.

c) Using the number of accumulated values, the new mean and new standard deviation values together with the numberStdDevs : double property, the number of and size of each data bin is determined. The number of data bins increases as more values are accumulated, however is restricted to 128 bins. The accumulated data is then re-binned ready for plotting.
Note: each plotted bin shows the proportion of points in the bin, so is always strictly within the range 0.0 to 1.0

Note: currently the weight assigned to each value update is the same. This means that a value that has remained the same (perhaps due the MDEL value) for 5 minutes has the same statistical weight as a value that has remained the same for 5 seconds. This will be addressed at a later date.

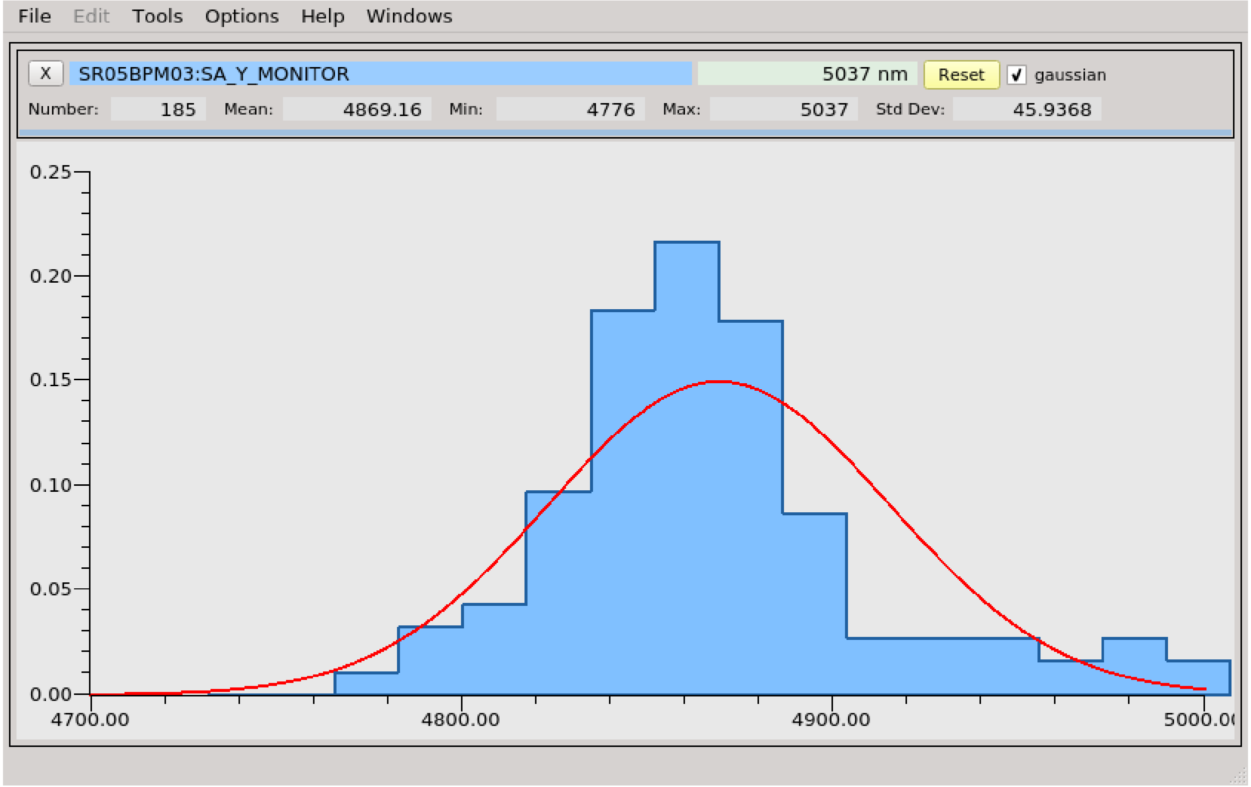
The widget provides a reset button. This clears the list of retained values and provided a "start again" capability.

The widget also provides a check box to allow the run time selection of the showGaussian : bool property value.

The PV name may be dragged and dropped onto the widget, copy and pasted to the widget, or entered using the standard PV name selection dialog accessed from the X button, or via the context menu associated with the PV name label. This is similar to the way PVs are added to QEStripChart and QEPlotter widgets.

This widget cannot retrieve values form the archive, however the QEStripChart widget can retrieve archived data and its Generate Statistic option provides a similar statistical analysis.

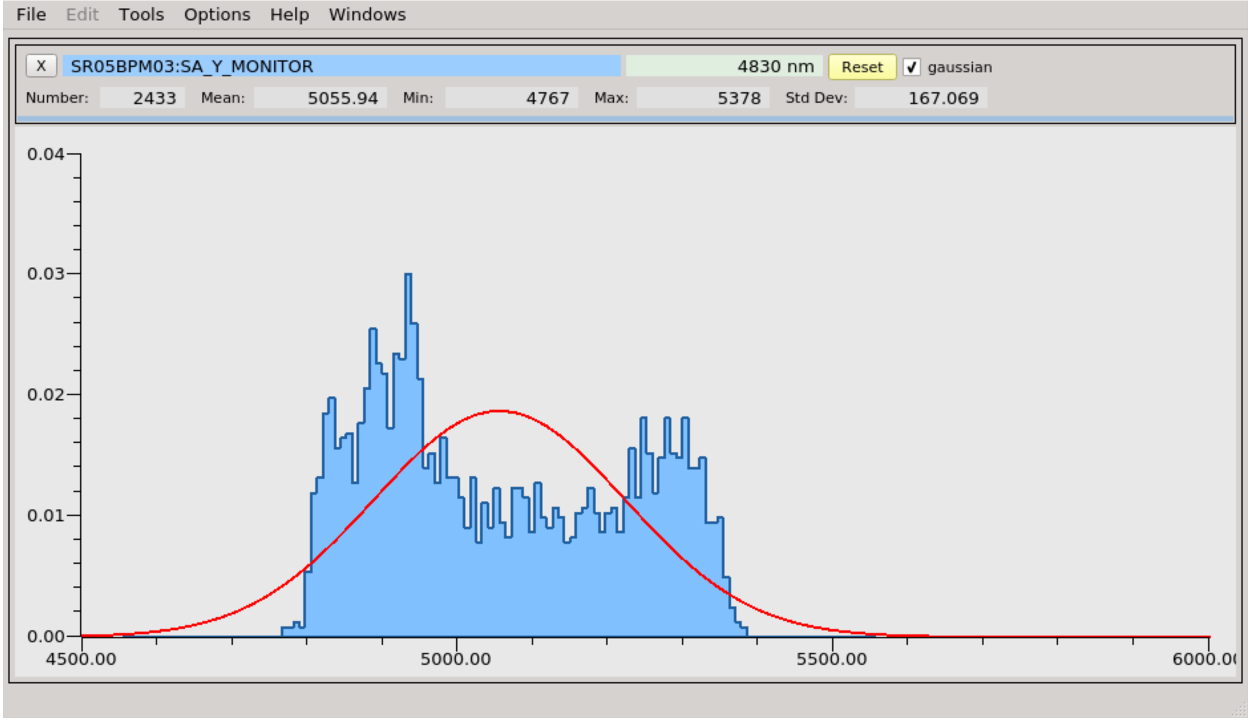
The functionality may be access via the QEGui built-in PV Distribution form (via menu entry **Tools | PV Distribution…**) or by placing a QEDistribution widget within a user defined form. In the latter case, the PV variable names can be set using the variable property accessible when using designer.



SR05BPM03:SA_Y_MONITOR 5037 nm ☑ gaussian

Number: 185 Mean: 4869.16 Min: 4776 Max: 5037 Std Dev: 45.9368

Figure 1 QEDistribution – few points



SR05BPM03:SA_Y_MONITOR 4830 nm

Number: 2433 Mean: 5055.94 Min: 4767 Max: 5378 Std Dev: 167.069

Figure 2 QEDistribution – many points

Properties

The QEDistribution inherits directly from QEFrame (and QEDynamicAbstractWidget) and as such inherits all the properties. The widget has the following class specific properties.

variable : QString

This defines the process variable protocol, the default is "ca://", and process variable name and to be used.

variableSubstitutions : QString

This defines the default substitutions that are applied to the variable name.

elementsRequired : int

default value: 0

For an array PV, this can be used to limit the number of elements subscribed for (0 implies all elements).

arrayIndex : int

default value: 0

For an array PV, this nominates the array element subject to the statistical analysis.

numberStdDevs : double

allowed range: 0.1 to 9.0

default value: 0

This defines the number of standard deviations, *n*, over which the data is binned, i.e.:
*mean - n*σ* to *mean + n*σ*.

isRectangular : bool

default value: true

This defines how the binned data is plotted. When true, the data is plotted like a histogram (as in the examples above). When this property is false, the data is plotted as a continuous curve.

edgeWidth : int

allowed range: 0 to 20

default value: 2

This controls the width of edge or boarder of the plotted data.

showGaussian : bool

default value: true

This defines whether the plotted binned data is overlayed with the nominal Gaussian distribution which is based on the mean and standard deviation of the accumulated data.

gaussianWidth : int

allowed range: 0 to 20

default value: 2

This controls the pen width of nominal Gaussian deviation.

fillColour : QColor

default value: pale blue

This specifies the brush colour of the plotted binned data.

edgeColour : QColor

default value: dark blue

This specifies the pen/edge colour of the plotted binned data.

gaussianColour : QColor

default value: red

This specifies the pen/line colour of the nominal Gaussian deviation.

backgroundColour : QColor

default value: light grey

This specifies the background colour of the plot area.

Note: the overall widget background colour may be set using the regular style sheet mechanisms.