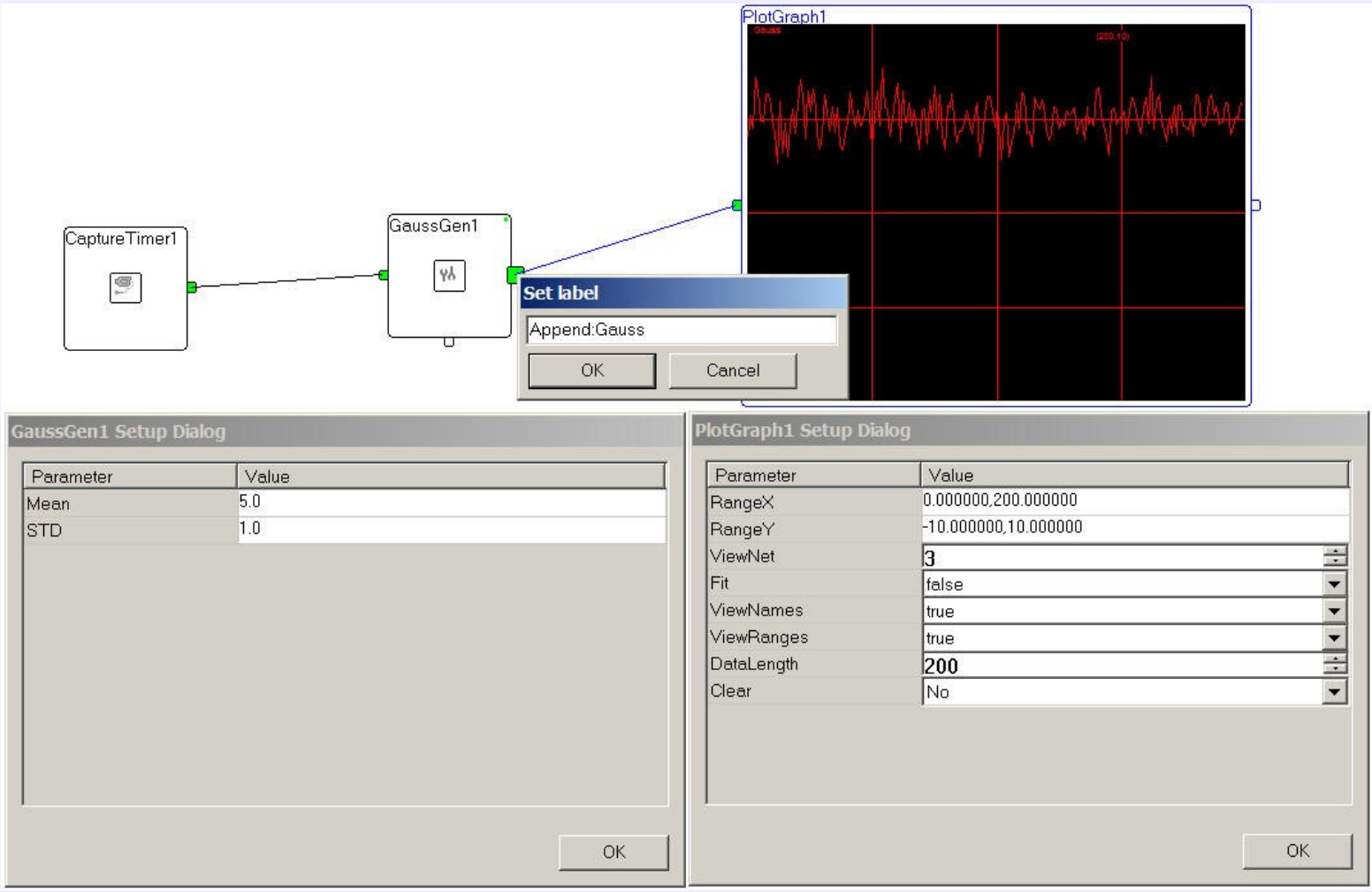
PlotGraph gadget

Input Formats:	CQuantityFrame, CFigureFrame or CArrayFrame
Outputs Formats:	CTextFrame
Duplex Formats:	NONE
Location in gadget tree:	Filters\Data\renderers for grouped by type or Data\renderers\ by default

PlotGraph gadget plots an input data as a 2D graph.

If mouse is in graph area and some click done, the coordinates of mouse cursor will be sent to output as text frame.

The sample of usage.



On this graph: GaussGen1 gadget produces values with gaussian distribution, PlotGrpah1 gadget does view of these values as graph.

Input pin does input frames receiving, decoding and data integration. Data arriving in Quantity Frame or Array Frame as integer or double numbers, will be shown as usual graphs, where is data value will be shown as Y (bigger is higher). X position of every new value is dependent on viewing mode, which is defined by arriving data frame label.

Label format for viewing control is "<Mode>:<GraphName>". GraphName is name for presentation on view in left upper corner of view on graph sample.

There are following viewing modes available (defined by frame label):

1. Append - figure, one value or array with "GraphName" will be appended to current data for graph. If data length of graph is more than "DataLength" gadget property graph will be shifted to the left (data in the beginning will be stripped). **Mode Append is used in graph sample above (graph is running from the right to the left)**.

2. Insert - figure, one value or array with "GraphName" will be inserted in the beginning to current data for graph. If data length of graph is more than "DataLength" gadget property graph will be shifted to the right (data in the end will be stripped).

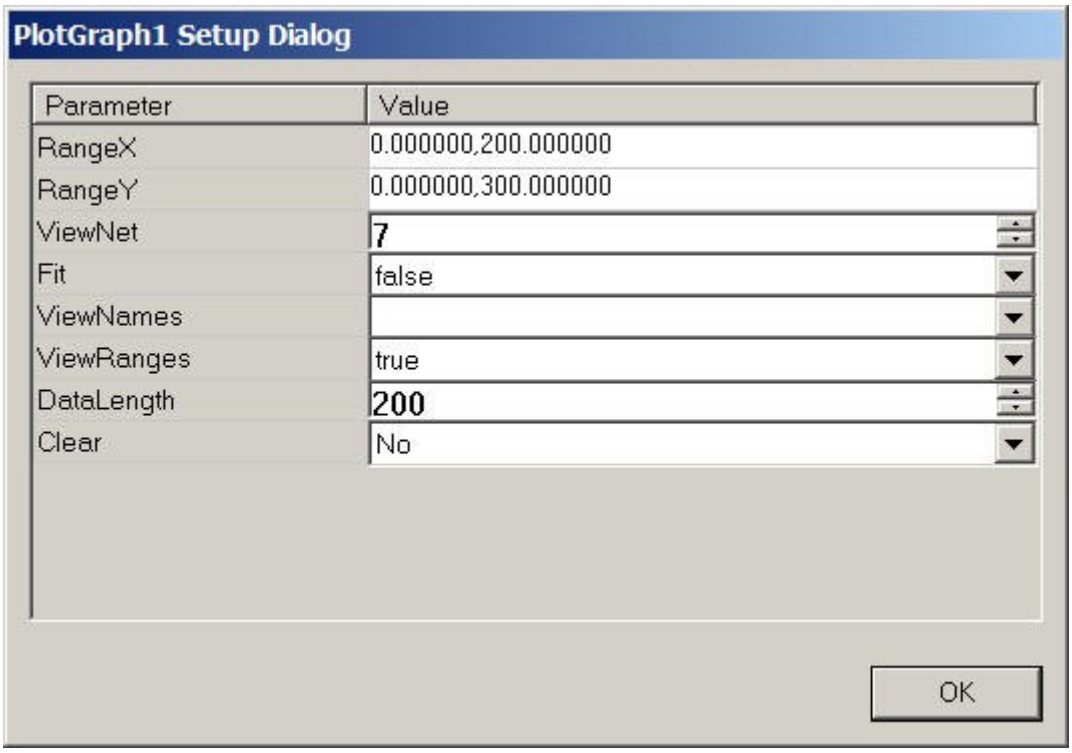
3. Replace - old graph "GraphName" data will be thrown, new graph will be fully taken from arrived data packet. This mode is for figures or arrays.

4. Remove - graph "GraphName" will be removed.

5. RemoveAll - all graphs will be removed. This mode is for graph initialization before processing sequence.

Output pin produces info about cursor position when cursor is moving over gadget view. Output data are in CTextFrame in format "x=<number.;y=<number>;

Gadget setup dialog:



RangeX - range of possible X values. If arrived or calculated X is not in this range, plot point will be out of view.

RangeY - range of possible Y values. If arrived or calculated Y is not in this range, plot point will be out of view.

ViewNet - how many vertical and horizontal lines will be drawn on graph for better points position estimation. Lines are used as markers on axes. If necessary to view marker lines every 25% of range, ViewNet should be equal to 3. ViewNet should be equal to 9 for 10% markers.

View Names - is it necessary to show names of graphs or not.

View Ranges - is it necessary to show X and Y ranges on the graph.

DataLength - how many numbers necessary to hold for each graph.

Clear - if changed, graph immediately will be cleared.

Fit - sets auto scale for Y coordinate (It's working not in all modes)

There is additional features for presentation control by arrived frame attributes: if attributes string is not empty gadget will analyze it for additional parameters. Keep in the mind, that parameters are case sensitive, i.e. capital letters are different from small. Following is available parameter list:

1. **color**—graph color setting. Format is "color=<color value>;", where color value is decimal or hexadecimal number for 3 bytes color value representation. The best is to use hexadecimal format "0xRRGGBB", where RR is two digit for red amplitude, GG—for green component, BB– for blue component. If color is not settled, gadget will automatically choose color.

2. **width**—line thickness in pixels. Format is "width=<thikness>;". Default thickness is 1.

3. **drawmode**– setting for draw mode. Format is "drawmode=<mode>;", where mode is number from 0 to 3. Available modes, are:

   ABSOLUTE(0) - where all received numbers will be used as is, without scaling.

   FIT_X(1) - X range of data will be fitted to the graph width.

   FIT_Y(2) - Y range of data will be fitted to the graph height.

   FIT_XY(3) - X and Y ranges will be fitted into width and height of drawing area.

4. **drawarea**– normalized draw area. Format is "drawarea=<left,right,top,bottom>;", where left, right, top, bottom is floating point numbers in range [0.,1.0]. For example, string "drawarea=0.,0.5,0.,0.5;" will lead to graph drawing in left upper quarter of drawing window. Different graphs could have different draw areas. Default draw area is full window client area.

5. **minmaxes**— view ranges for X and Y. Format is "minmaxes=<Xmin>,<Xmax>,<Ymin>,<Ymax>". For example, string "minmaxes=100,200,300,400" will lead to viewing of area between points (100,300) and (200,400). All numbers will be parsed as floating point.