

Benefits of EDD Enhancements

Christian Diedrich, Matthias Riedl
Institut für Automation und Kommunikation e.V. Magdeburg
Steinfeldstr. 3, 39179 Barleben, Germany
Web: <http://www.ifak-md.de/>
Email: {christian.diedrich|matthias.riedl}@ifak-md.de

1	Overview about the improved features	1
2	Graphical enhancements in detail	2
2.1	Charts	2
2.1.1	Gauges	2
2.1.2	Bargraphs	3
2.1.3	Strip	4
2.1.4	Sweep	5
2.1.5	Scope	6
2.2	Graphs	7
2.3	Images	8
2.4	Grids	9
3	Improved handling of instance data	10
3.1	Dynamic Storage	10
3.2	Persistence	10

1 Overview about the improved features

In the international standard IEC 61804-2, available since spring 2004, the unified Electronic Device Description Language EDDL is specified. This fulfills the end user requirement for a global interoperable device description which can operate in any kind of EDD application. Furthermore, the end user expects enhanced functionalities in the form of additional graphical elements and solutions for the instance data handling, described inside the device description. On the one hand, recent field devices continue to become more complex and require more complex tools for commissioning, maintenance etc. On the other hand, there are necessary resources inside the tools in order to visualise complex coherencies by using sophisticated graphical elements inside the user interface. In order to fulfill these requirements and possibilities, the EDD was enhanced by additional features, which are in keeping with the spirit of the EDD.

The device manufacturer can decide for himself which level of graphical representation the end user can use. Therefore several levels of views are introduced for PC based applications:

- Diagnostic
- Process variables
- Device
- Offline

In summary, the enhancements cover functionalities about the presentations of graphs, the presentation of trending information in the form of charts and the very important handling of persistent data. The following chapters give an overview about the new functionality, offered by EDD technology.

Very important is to mention that the following descriptions are related to the data interfaces of the graphical elements only. The software tool manufacturers are completely free to choose the shape and colour of the elements to be displayed on any terminal, i.e. following corporate design rules for an engineering tool chain. The big benefit of this solution is the unified

representation of information also independent of any communication system (i.e. HART, FF, PROFIBUS).

2 Graphical enhancements in detail

2.1 Charts

Charts permit the presentation of device data as a continuous process. They provide a view of a time varying trend. Therefore charts can relate to sources of the values to display. The values are sampled periodically.

2.1.1 Gauges

A gauge allows you to display one value. Therefore there is only one relationship to a source instance. The source instance also describes the Y-axis as device parameter to display. The axis description defines the minimum and maximum values of the gauge. The following example could produce a representation shown in Figure 1.

```
CHART GaugeChartNormalSize
{
    LABEL          "Gauge";
    TYPE GAUGE;
    MEMBERS
    {
        SOURCE1, chartsource1;
    }
}
SOURCE chartsource1
{
    LABEL "source 1";
    Y_AXIS chartyaxis1;
    MEMBERS
    {
        VALUE1, trans1_primary_value;
    }
}
AXIS chartyaxis1
{
    LABEL "yaxis 1";
    MIN_VALUE trans1_lower_sensor_limit;
    MAX_VALUE trans1_upper_sensor_limit;
}
```

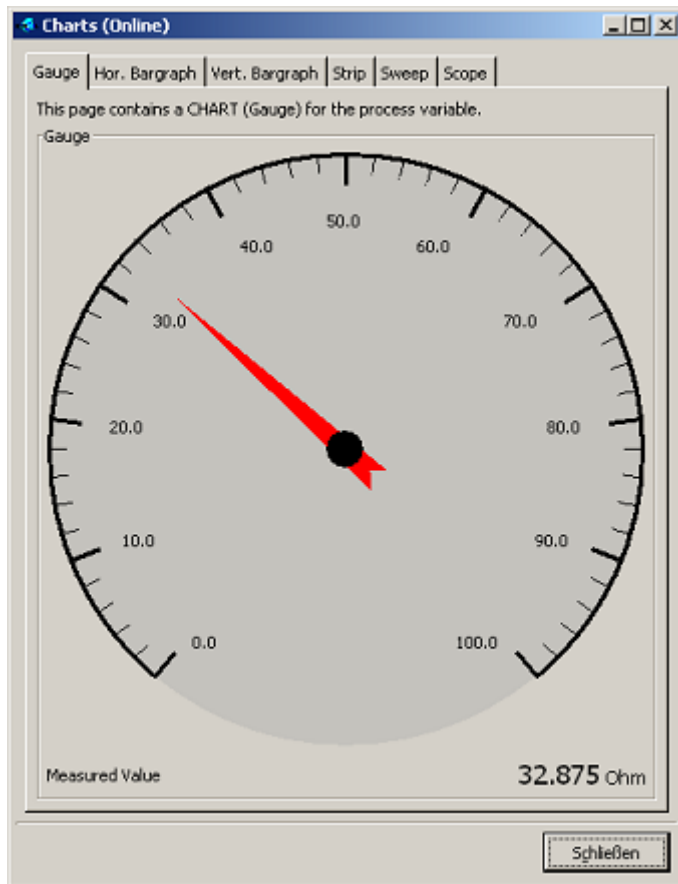


Figure 1: Representation of a Gauge

2.1.2 Bargraphs

Similar to the gauge, bargraphs allow the presentation of one or more continued varying values. The direction of the bargraph can be defined (HORIZONTAL_BAR or VERTICAL_BAR). Figure 2 shows a possible representation of such chart elements.

```

CHART VertChartNormalSize
{
    LABEL "Vertical Bargraph";
    TYPE VERTICAL_BAR;
    MEMBERS
    {
        SOURCE1, chartsource1;
    }
}
SOURCE chartsource1
{
    LABEL "source 1";
    Y_AXIS chartyaxis1;
    MEMBERS
    {
        VALUE1, trans1_primary_value;
    }
}
AXIS chartyaxis1
{
    LABEL "yaxis 1";
    MIN_VALUE trans1_lower_sensor_limit;
    MAX_VALUE trans1_upper_sensor_limit;
}

```

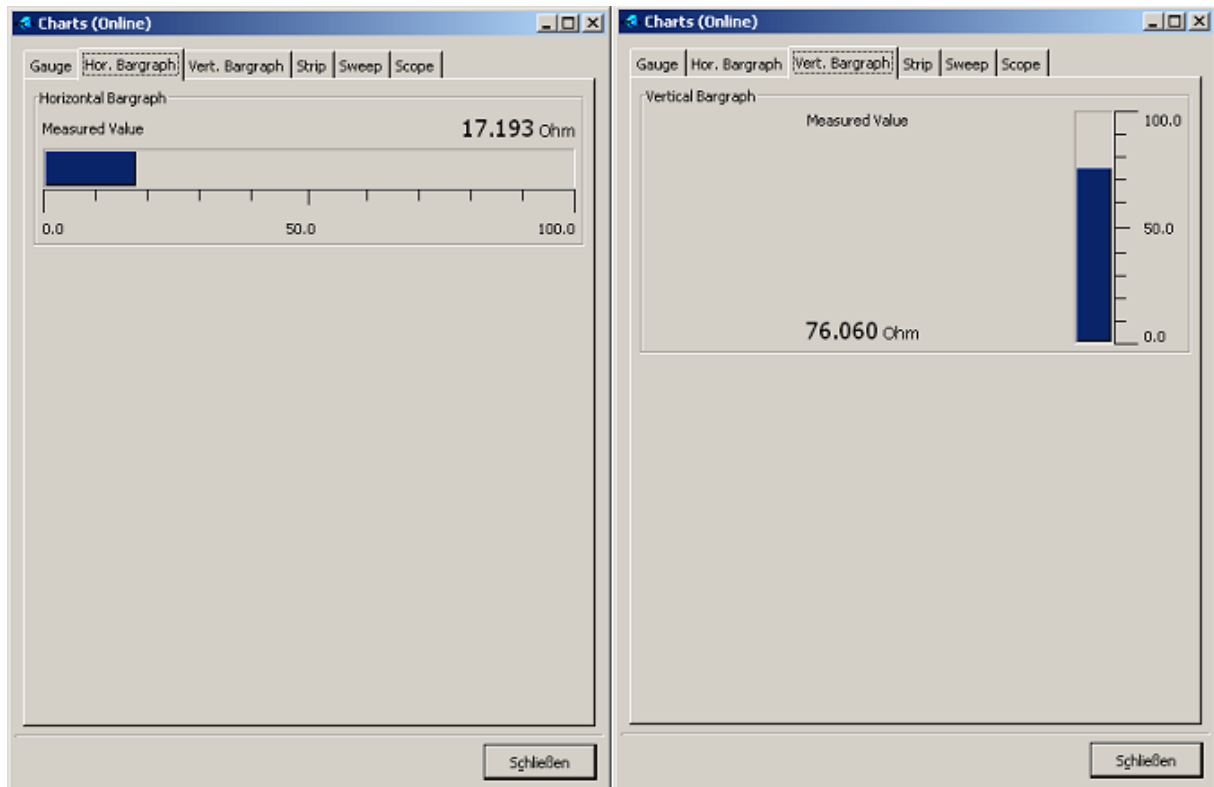


Figure 2: Horizontal and vertical Bargraphs

2.1.3 Strip

Additional to gauges, the following elements allow the representation of more than one device parameter. Therefore several axis and device parameters can be referenced. The following example could produce the chart representation, shown in Figure 3. The characteristics of a Strip is the following: when the display source value reaches the end of the display area, the complete area will be scrolled. The oldest values will be removed and the newest are visible on the right side.

```

CHART chartStrip1
{
    LABEL      "Strip";
    CYCLE_TIME 100;
    LENGTH     100 * 360;
    MEMBERS
    {
        SOURCE1, sourceStrip1; // first
    }
}
SOURCE sourceStrip1
{
    LABEL "source 1";
    Y_AXIS chartyaxis1;
    LINE_COLOR 0x8515c7;
    MEMBERS
    {
        VALUE1, trans1_primary_value;
    }
}
AXIS chartyaxis1
{
    LABEL "yaxis 1";
    MIN_VALUE trans1_lower_sensor_limit;
    MAX_VALUE trans1_upper_sensor_limit;
}

```

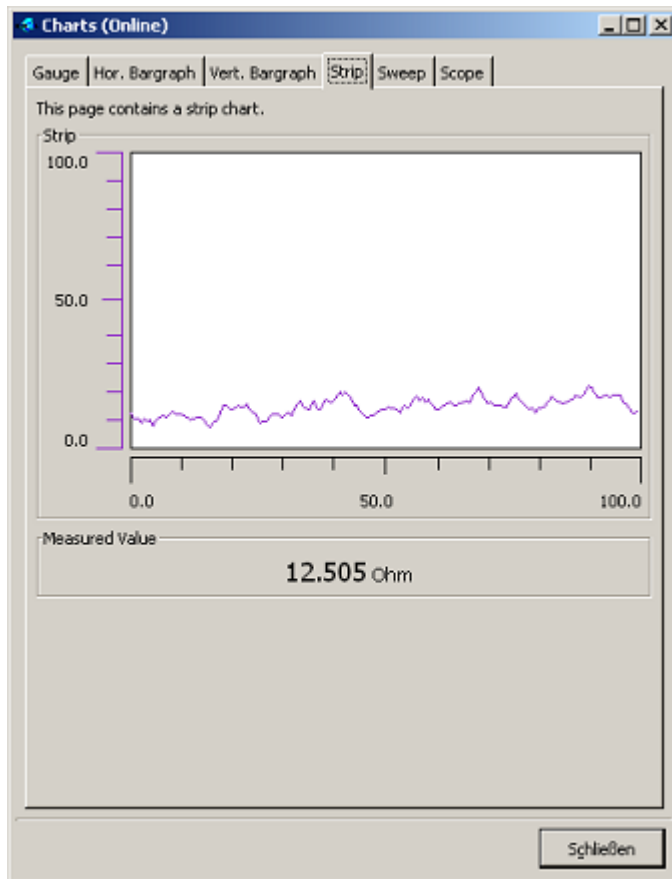


Figure 3: Chart as a Strip

2.1.4 Sweep

The Sweep characteristic is: when the display source value reaches the end of the display area, the new value is shown at the beginning of the display area. Thereby only the part of the display with the new value will be refreshed, see Figure 4.

The following example demonstrates the power of reusability of already defined EDD elements. Only the type of the representation is redefined, all other references will be used the same as in the original.

```
chartSweep1 LIKE chartStrip1
{
    REDEFINE TYPE SWEEP;
}
```

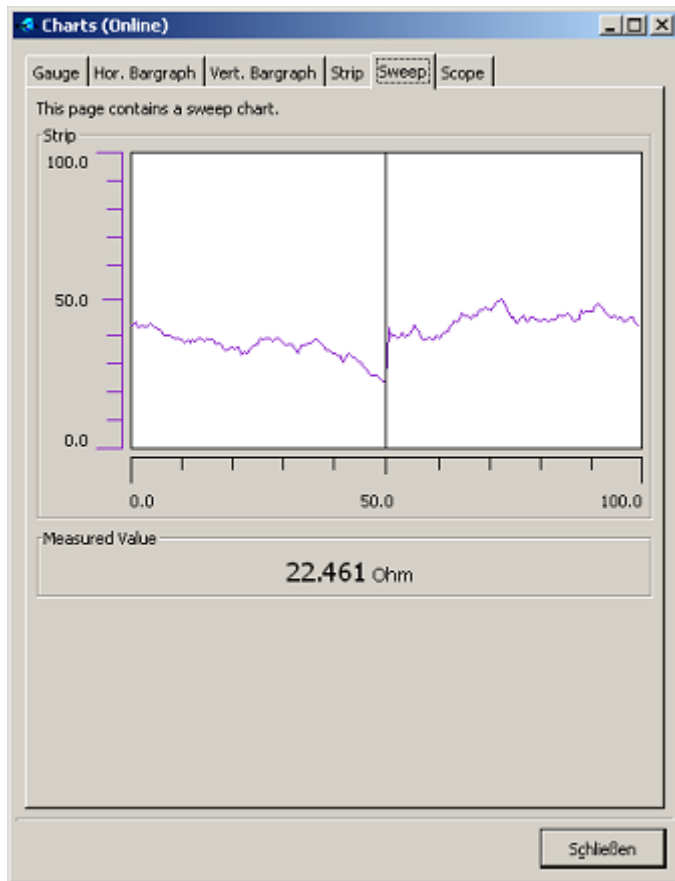


Figure 4: Chart as a Sweep

2.1.5 Scope

```
chartScope1 LIKE chartStrip1
{
    REDEFINE TYPE SCOPE;
}
```

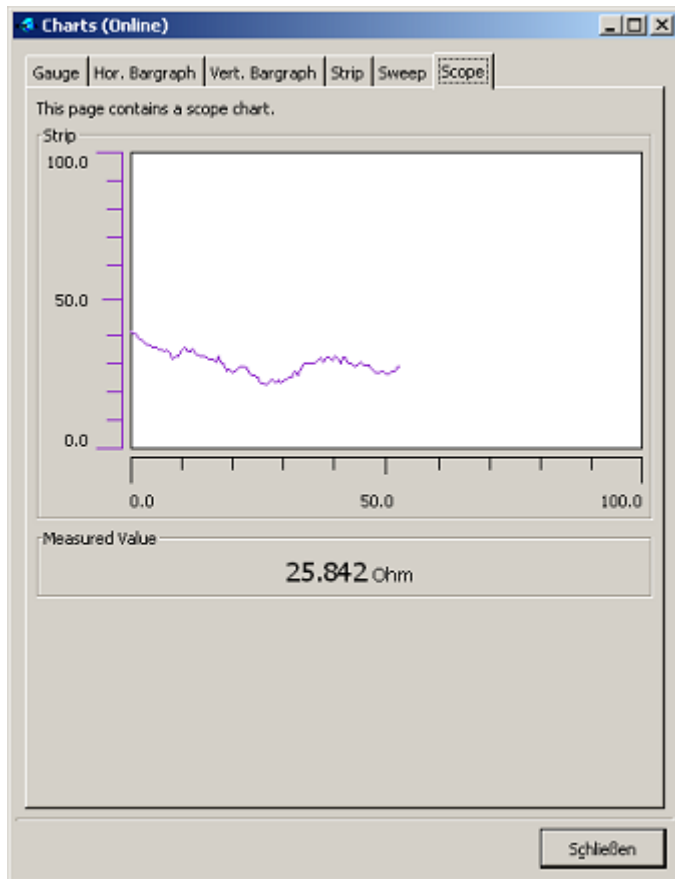


Figure 5: Chart as a Scope

2.2 Graphs

In contrast to the charts, a graph allows the representation of complete waveforms. The waveforms themselves could be sampled inside the device before the presentation or could be calculated inside the EDD. Furthermore several curves can be presented inside a graph object. Each of these curves has its own y-axis and colour. Additionally special points or lines (vertical and horizontal) can be defined.

GRAPH graph1

```
{
    LABEL          "Graph 1";
    X_AXIS         my_x_axis;
    MEMBERS
    {
        WAVEFORM1, waveformXY;
        WAVEFORM3, waveformHori;
        WAVEFORM4, waveformVert;
    }
}
WAVEFORM waveformXY
{
    LABEL This is the label";
    HELP "Help for WaveformXY";
    HANDLING READ_WRITE;
    Y_AXIS yaxis1;
    LINE_COLOR 0xff0000;
    TYPE XY
    {
        NUMBER_OF_POINTS 35;
        ...
    }
}
```

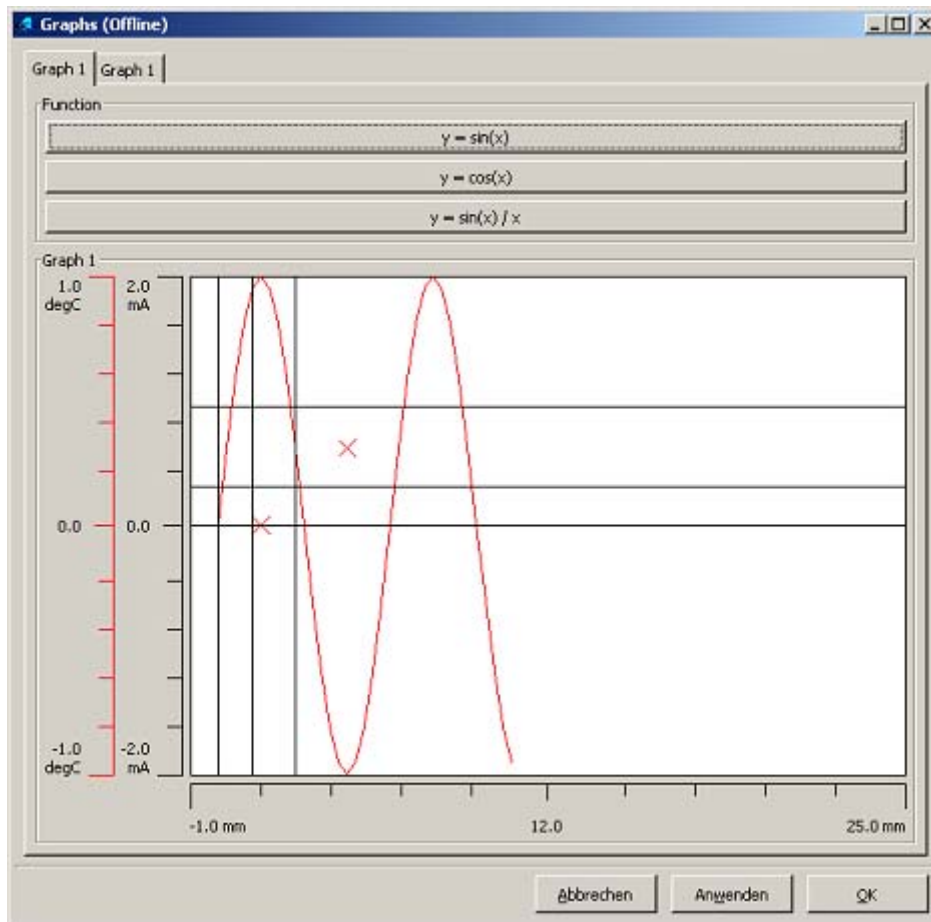


Figure 6: Graph with three curves and points

2.3 Images

From the device vendor point of view, the presentation of the device or the company label is very important. Therefore the EDD offers the possibility to prettify some windows or dialogs by graphics. The graphic can be positioned inside the window. This is helpful in order to describe the complex semantic of the device with a picture which belongs to parameters nearby. The following example demonstrates the presentation of two pictures of different formats. Both pictures obtained the possibility to call a method by means of the LINK statement.

MENU pageImage

```
{
  STYLE PAGE;
  LABEL "Image";
  ITEMS
  {
    imageLogo,
    imageBmp
  }
}
IMAGE imageLogo
{
  PATH "ifak_logo.jpg";
  LINK methodImageLink1;
}
IMAGE imageBmp
{
  PATH "TF12-Ex.bmp";
  LINK methodImageLink1;
  LABEL "TestLabel";
}
```



Figure 7: Images inside a window on a page

2.4 Grids

Another possibility for the representation of data is the definition of an own structured grid or table. Inside the grid any amount of columns and lines can be displayed. The example describes a grid of three columns in the form of vectors. The name of each referenced array inside is equivalent to the name of the column. The amount of the rows inside the columns is defined by the referenced arrays.

```

GRID GridFoundEcho
{
    LABEL "All detected echos are displayed below";
    VECTORS
    {
        { "Type", arrPeakType },
        { "Distance", arrPeakDistance },
        { "Amplitude", arrPeakAmplitude }
    }
}
VARIABLE PeakType
{
    LABEL "Peak Type";
    CLASS DEVICE;
    TYPE ENUMERATED
    {
        { 1, "False Echo" },
        { 2, "Button Echo" },
        { 3, "Unknown" }
    }
}
ARRAY arrPeakType
{
    NUMBER_OF_ELEMENTS 10;
    TYPE PeakType;
}

```

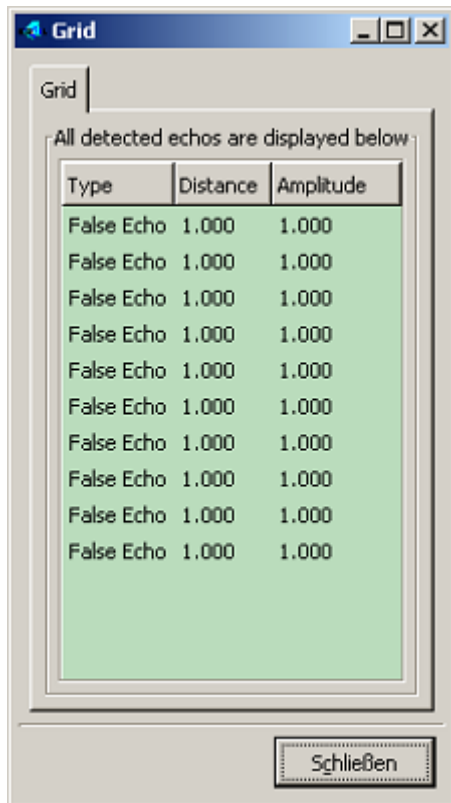


Figure 8: Example of a Grid

3 Improved handling of instance data

3.1 Dynamic Storage

The new statement LIST offers the possibility to store an unlimited amount of data in it. The access to the elements is done by using indexes. Therefore very efficient algorithms inside the methods can be implemented. The data type inside the dynamic storage is a reference of an EDD element, e.g. a variable, collection of something else. In order to add a new element into the storage, the access to a new index allocates automatically the needed memory. On the other hand, the garbage collection of the unused memory will be done also by the EDD host.

3.2 Persistence

The persistent storage of the device states is very important for the history/chronicle of the device inside the plant. The statement FILE offers the possibility to name such a persistent storage inside the EDD. The EDD host application is responsible for saving such constructs automatically using the environment specific data base interface, when the applications stop the handling with the device, and to load the correct data set, when the handling is continued. An example, which could be used very often, is to record the fading of the functionality of an actor or sensor. Therefore, this state can be stored in a dynamic storage and this storage could be referenced inside a FILE construct. Therefore the data will be continually stored inside the instance data which belongs to the specific device and can be displayed inside a graph, a grid or can be computed inside a method in order to calculate a derivation of a curve in order to signal the kind of fading.