

HMI-DesignBox Documentation

SYBERA GmbH © 2020

Release Date: July,24th.2023

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020

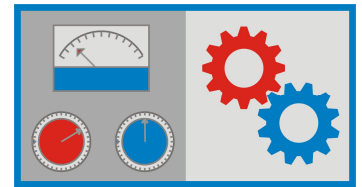
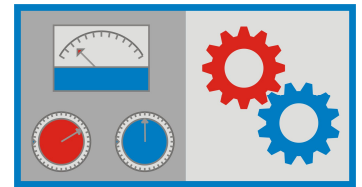


Table of content

1	INTRODUCTION	3
1.1	Features	4
1.2	Supported OS	4
1.3	Supported Development Environment	4
1.4	Default provided panel templates:	4
2	HMI DESIGNBOX	5
2.1	Panel Parameter	6
2.1.1	Panel Hotkeys:	6
2.2	AnalogControl	7
2.3	AnalogMeter	8
2.4	LinearMeter	9
2.5	ButtonControl	10
2.6	DigitalMeter	11
2.7	Graph	12
2.8	Indicator	13
2.9	InputValue	14
2.10	JoystickControl	15
2.11	LinearControl	16
2.12	DioModule	17
2.13	Motion	18
2.14	TeachAxis	19
2.14.1	X-Axis Parameter File (Dialog Type 1)	20
2.14.2	Additional Axis Parameter File (Dialog Type 2 – 4)	20
2.14.3	Additional Speed Parameter File (Dialog Type 5)	20
2.14.4	Additional Control Parameter File (Dialog Type 6)	20
2.14.5	TeachAxis Dialog	21
2.15	LoadAxis	22
3	SMART CAMERA EXTENSION	23
3.1	Creating a Camera Configuration File	24
3.1.1	GigE Accesspoint	25
3.1.2	GeniCam Register Groups	26
3.2	Creating a Processing Parameter File	27
3.3	Connect a PLC variable within X-GO Logic Control	29
3.4	Image Processing Functions	30
3.4.1	Function “Region Center”	30
3.4.2	Function “Region Contour”	31
3.4.3	Function “Match Pattern”	32
3.4.4	Function “Edge”	33
3.4.5	Function “Distance”	33
3.4.6	Function “User”	34
4	HMI-SDK	35
4.1	C++ Export functions	35
4.1.1	Sample function call C++	36
4.2	SDK template prototype	37
4.2.1	Parameter list structure	37
4.2.2	Template file	37
4.2.3	Sample	38

HMI-DesignBox SDK Documentation

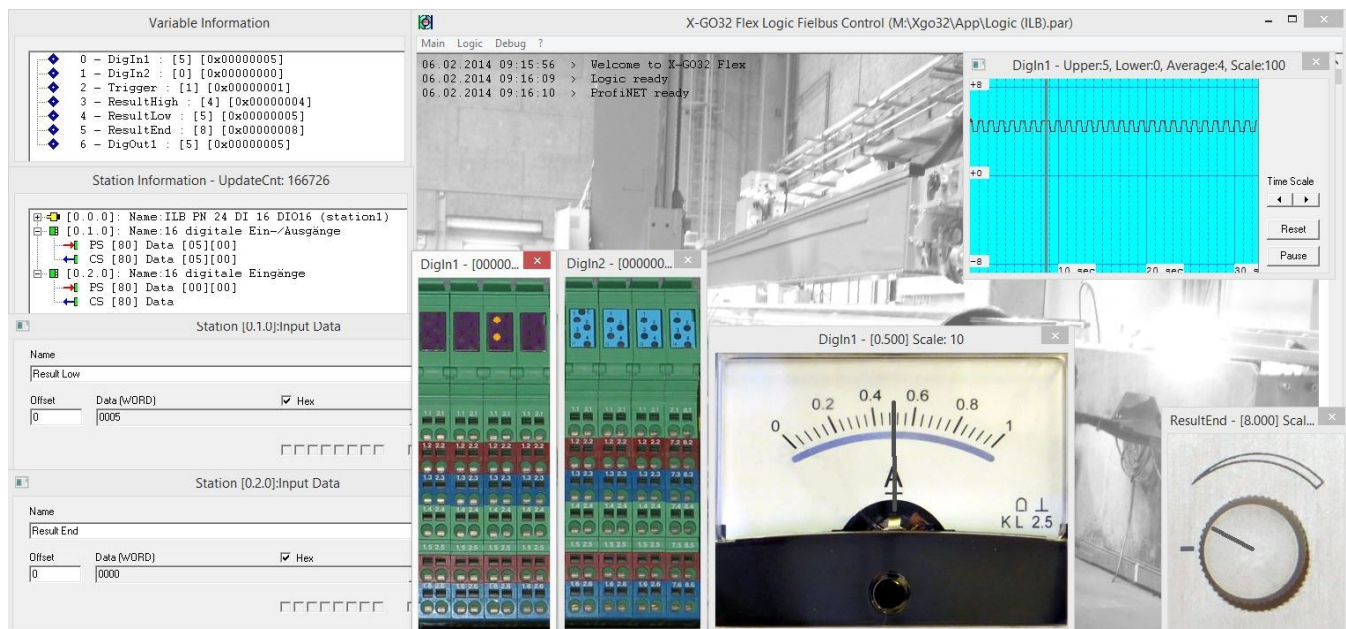
SYBERA GmbH Copyright © 2020



1 Introduction

Design your HMI design in a few simple steps. The HMI-DesignBox allows the design of photo-realistic surfaces with a simple data interface to fieldbus systems. The construction of the HMI design is done without programming. The SDK (Software Development Kit) enables the quick connection of input and output values to the HMI elements, both for control and for the simulation of fieldbus systems. Whether integrating measuring devices, input panels or backgrounds - the surface can be finished quickly and efficiently.

With pre-assembled templates, you can overlay images and graphics that correspond to your system with parameterizable control panels and input modules and link them with PLC variables. The HMI modules are overlaid by simply adjusting parameter files.

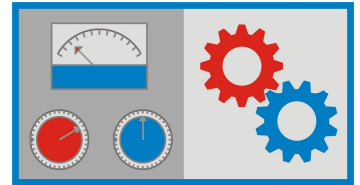


Especially in simulation technology, the fieldbus stacks (e.g. ProfiNET slave, EtherCAT slave or Ethernet / IP slave) can now be linked with a convenient interface. This turns the PC into a complete system simulation in real time.

The HMI-DesignBox is optimally matched to the fieldbus stacks from Sybera and thus enables a quick connection to the existing infrastructure. The HMI-DesignBox optimally complements the existing Sybera products

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



1.1 Features

- Photorealistic surface design
- Fast function connection with SDK
- HMI design without programming
- Function templates
- Simple data connection
- Extendable function modules
- Optimal adaptation to the master and slave stacks

1.2 Supported OS

- Windows 7
- Windows 8
- Windows 10

1.3 Supported Development Environment

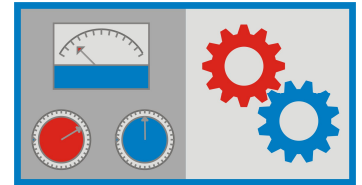
- VisualStudio (>10)

1.4 Default provided panel templates:

- AnalogControl
- AnalogMeter
- ButtonControl
- DigitalMeter
- DioModule
- Graph
- Indicator
- InputValue
- JoystickControl
- LinearControl
- LinearMeter
- Motion
- ShowEllypse
- ShowImage
- ShowLine
- TeachAxis
- LoadAxis

HMI-DesignBox SDK Documentation

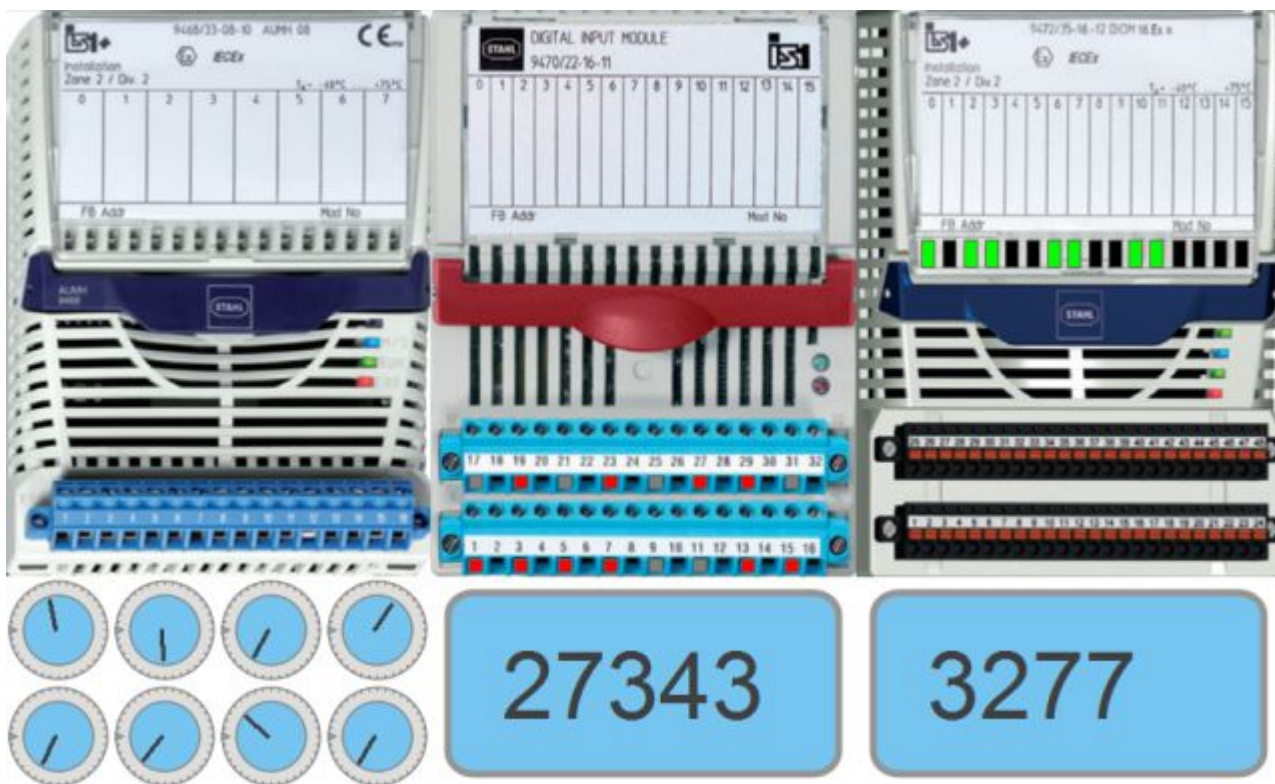
SYBERA GmbH Copyright © 2020



2 HMI DesignBox

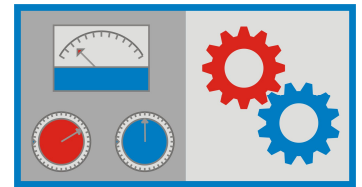
The HMI-DesignBox allows the photorealistic design and integration of machine panel instruments and controls, like meters, switches, modules and backgrounds - without programming. You can use images and graphics of your machine instruments, overlaid by corresponding software templates and simply linked to PLC program variables. The overlay of the software templates is carried out through simple adjustment of the parameter files to each HMI panel. Additionally new templates can be programmed and integrated with the SDK.

Sample: Panel Parameter Set



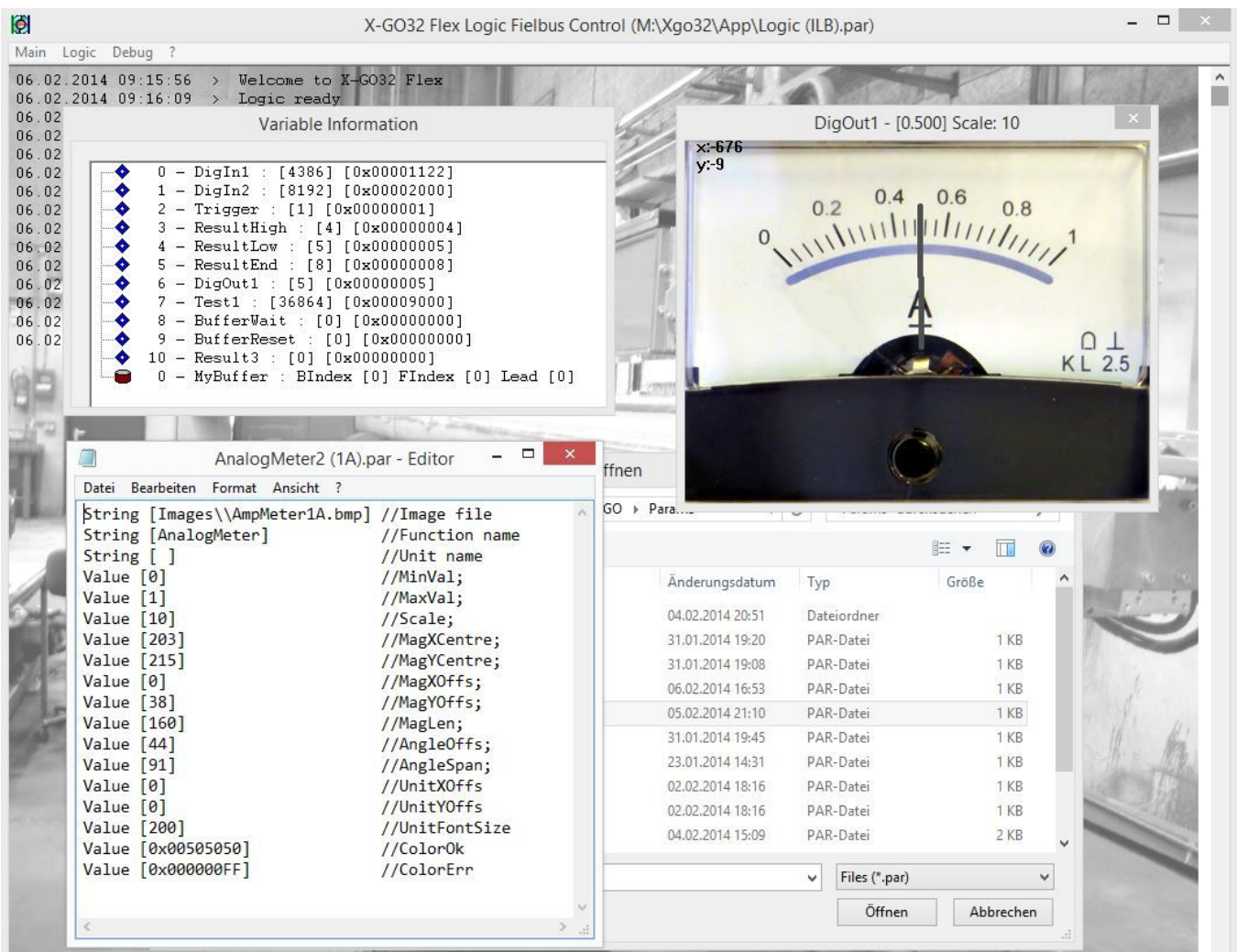
HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



2.1 Panel Parameter

To overlay a software panel template on a given image, simply adjust the parameter set stored in the parameter file. This can be done with any editor. The default parameter files are stored within the directory \PARAMS, while corresponding images of Windows bitmap type BMP are stored within \PARAMS\IMAGES. To adjust the panel element positions, like magnitude, unit offset, font size, ..., by pressing the key [F5] the coordinates are shown in the left-top corner inside the panel dialog. When pressing the key [F4] the movement of the panel is adjusted to a grid.

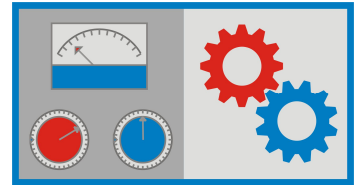


2.1.1 Panel Hotkeys:

- F4 : Toggle grid
- F5 : Print pixel coordinates
- F6 : Unlock position
- F7 : Delete panel
- F8 : Decrease Z-Order
- F9 : Increase Z-Order

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



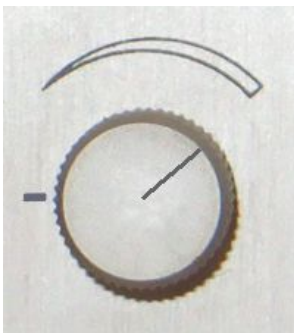
2.2 AnalogControl

The analog control panel overlays a magnitude to an image. Connected to an input value, it changes the value to its movement.

Sample: \Params\AnalogControl (1).PAR

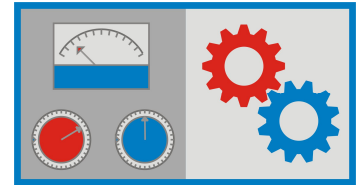
```
String [Images\\AnalogControl_2.bmp] //Image file
String [AnalogControl]                //Function name
String [ ]                             //Unit name
Value [0]                             //ShowTitle;
Value [0]                             //MinVal;
Value [10]                             //MaxVal;
Value [1]                              //Scale;
Value [95]                             //MagXCentre;
Value [130]                            //MagYCentre;
Value [0]                              //MagXOffs;
Value [0]                              //MagYOffs;
Value [50]                             //MagLen;
Value [3]                              //MagWidth;
Value [0]                              //AngleOffs;
Value [350]                            //AngleSpan;
Value [0x00646464] //ColorOk
Value [0x000000FF] //ColorErr
Value [0x00FFFFFF] //ColorMask (0xFFFFFFFF : OFF)
```

Sample Image



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



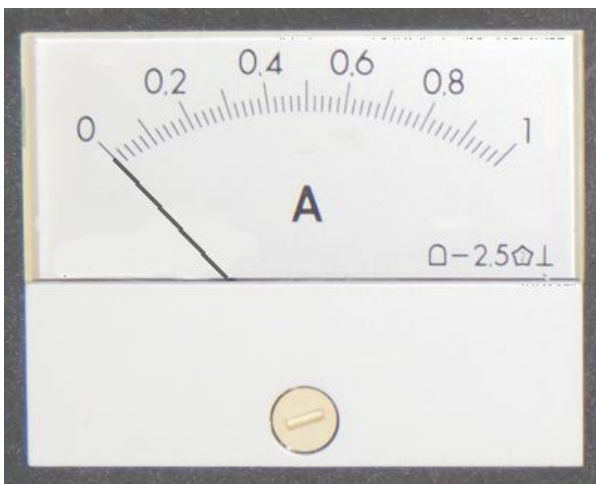
2.3 AnalogMeter

The analog meter panel overlays a magnitude to an image. Connected to an output value, it moves the magnitude depending on the value.

Sample: \Params\AnalogMeter (1A).PAR

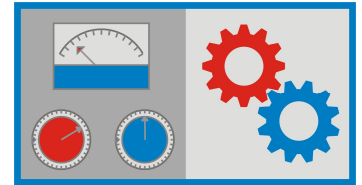
```
String [Images\\AmpMeter.bmp] //Image file
String [AnalogMeter]          //Function name
String [ ]                     //Unit name
Value [0]                     //ShowTitle;
Value [0]                     //MinVal
Value [1]                     //MaxVal
Value [10000]                 //Scale
Value [203]                   //MagXCentre
Value [240]                   //MagYCentre
Value [0]                     //MagXOffs
Value [60]                   //MagYOffs
Value [190]                   //MagLen
Value [3]                     //MagWidth;
Value [47]                   //AngleOffs
Value [86]                   //AngleSpan
Value [0]                     //UnitXOffs
Value [0]                     //UnitYOffs
Value [200]                   //UnitFontSize
Value [0x00646464]           //ColorOk
Value [0x000000FF]           //ColorErr
Value [0x00FFFFFF]           //ColorMask (0xFFFFFFFF : OFF)
```

Sample Image



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



2.4 LinearMeter

The linear meter panel overlays a horizontal or vertical bar to an image. Depending on the output value it increases/decreases the bar.

Sample: \Params\LcdMeter.PAR

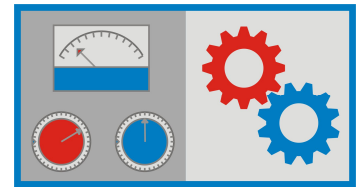
```
String [Images\\LcdMeter (V).bmp] //Image file
String [LinearMeter] //Function name
String [mA] //Unit name
Value [0] //ShowTitle;
Value [0] //MinVal
Value [100] //MaxVal
Value [80] //BarXOffs
Value [330] //BarYOffs
Value [20] //BarXSpan
Value [280] //BarYSpan
Value [7] //BarSize
Value [1] //Scale
Value [1] //ScaleVertical;
Value [70] //ScaleXOffs
Value [340] //ScaleYOffs
Value [5] //ScaleXSpan
Value [280] //ScaleYSpan
Value [10] //ScaleGrad
Value [25] //UnitXOffs
Value [340] //UnitYOffs
Value [150] //FontSizeUnit
Value [0x00323232] //ColorOk
Value [0x000000FF] //ColorErr
Value [0x00FFFFFF] //ColorMask (0xFFFFFFFF : OFF)
```

Sample Images (vertical / horizontal)



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



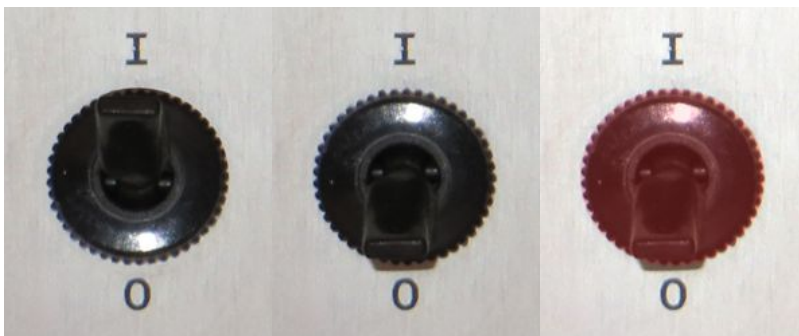
2.5 ButtonControl

The button control panel shows different images, depending on the input/output value. It may be designed as switch or as a button panel.

Sample: \Params\ButtonControl.PAR

```
String [Images\\SwitchControl.bmp] //Image file base name (3 images must exist:
//*_OFF.BMP, *_ON.BMP, *_ERR.BMP)
String [ButtonControl] //Function name
String [] //Unit name
Value [0] //ShowTitle;
Value [0] //Mode (0: SwitchMode, 1: ButtonMode)
Value [0] //MinVal;
Value [100] //MaxVal;
Value [1] //Scale;
Value [1] //ValOn;
Value [0] //ValOff;
Value [75] //ClickXPosOn
Value [10] //ClickYPosOn
Value [75] //ClickXPosOff
Value [190] //ClickYPosOff
Value [50] //ClickXSpanOn
Value [50] //ClickYSpanOn
Value [50] //ClickXSpanOff
Value [50] //ClickYSpanOff
Value [0x00646464] //ColorOk
Value [0x00646464] //ColorErr
Value [0x00FFFFFF] //ColorMask (0xFFFFFFFF : OFF)
```

Sample ButtonControl Images



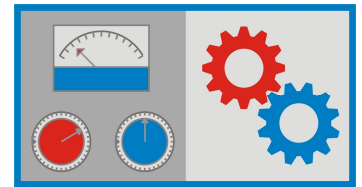
Switch_ON.BMP

Switch_OFF.BMP

Switch_ERR.BMP

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



2.6 DigitalMeter

The digital meter panel overlays the digital value to an image.

Sample: \Params\DigitalMeterLed (1V).PAR

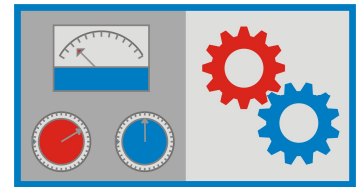
```
String [Images\\LedMeter.bmp] //Image file
String [LinearMeter]           //Function name
String [mA]                    //Unit name
Value [0]                      //ShowTitle;
Value [0]                      //MinVal
Value [1000]                   //MaxVal
Value [50]                     //BarXOffs
Value [60]                     //BarYOffs
Value [300]                    //BarXSpan
Value [40]                     //BarYSpan
Value [5]                      //BarSize
Value [100]                   //Scale
Value [0]                      //ScaleVertical;
Value [47]                    //ScaleXOffs
Value [110]                   //ScaleYOffs
Value [300]                   //ScaleXSpan
Value [5]                     //ScaleYSpan
Value [10]                    //ScaleGrad
Value [0]                     //UnitXOffs
Value [0]                     //UnitYOffs
Value [150]                   //FontSizeUnit
Value [0x000000FF]            //ColorOK = RGB(255, 0, 0);
Value [0x00C8C8C8]           //ColorErr = RGB(200,200,200);
Value [0x00FFFFFF]           //ColorMask (0xFFFFFFFF : OFF)
```

Sample



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020

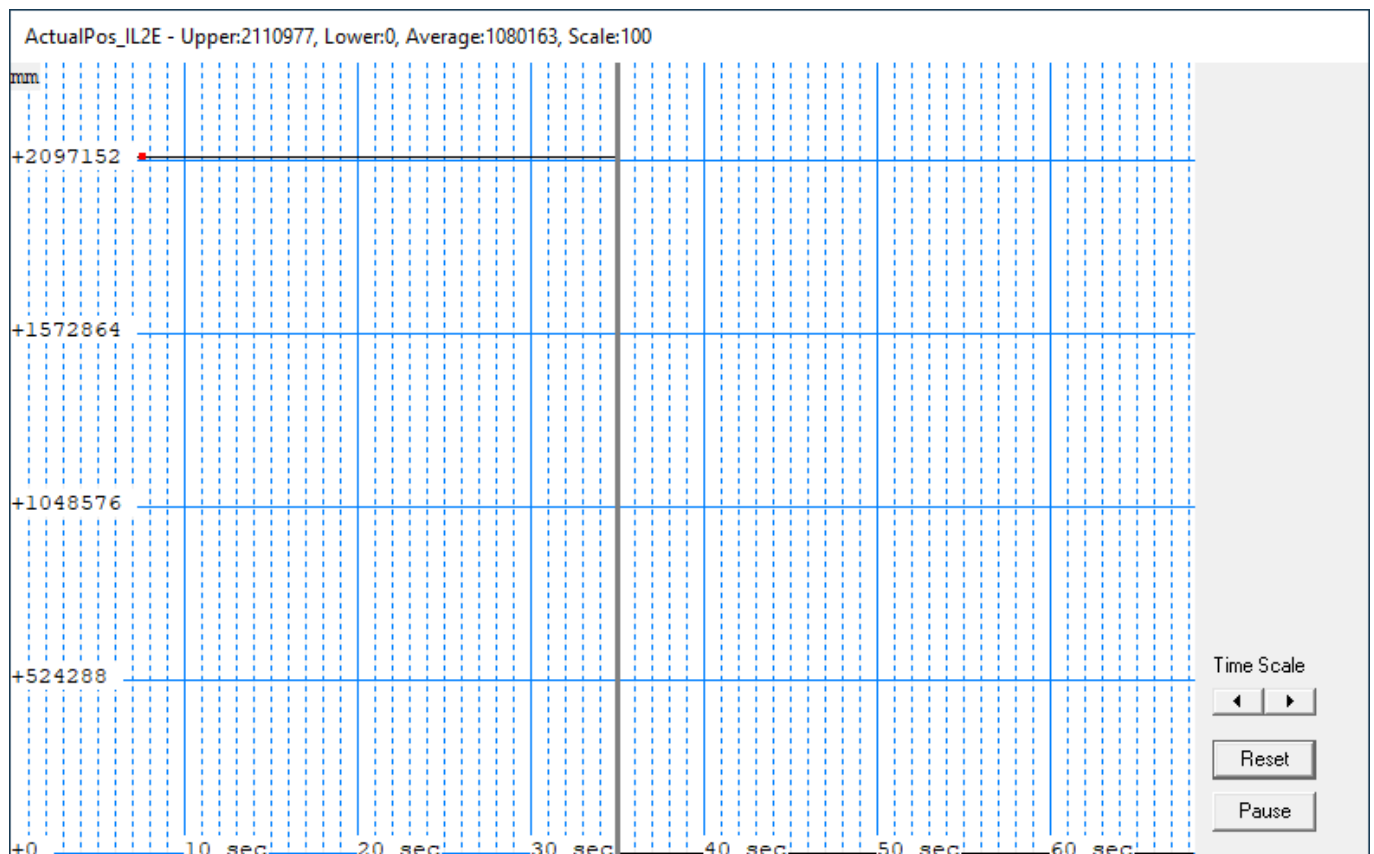


2.7 Graph

The graph panel shows the timing behavior of a value. Its resizable.

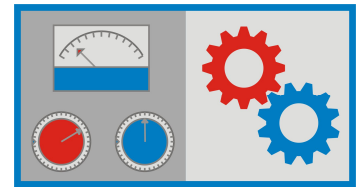
Sample: \Params\GenericGraph.PAR

```
String [] //Image file
String [GenericGraph] //Function name
String [mm] //Unit name
Value [0] //ShowTitle;
Value [800] //DlgWidth
Value [500] //DlgHeight
Value [1] //ScaleZero
Value [0] //UnitXOffs
Value [10] //UnitYOffs
Value [100] //UnitFontSize
```



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



2.8 Indicator

The indicator panel overlays a different colored circles or rectangle to an image, depending on the output value.

Sample: \Params\EMCA_State.PAR

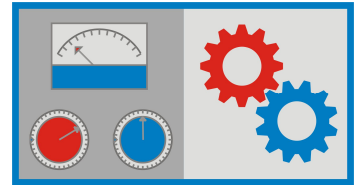
```
String [Images\\Led.bmp]           //Image file
String [Indicator]                 //Function name
String [EMCA]                      //Unit name
Value [0]                          //ShowTitle;
Value [0x0000FFFF]                 //ColorON           (color sheme: xxbbgrr)
Value [0x00808080]                 //ColorOFF        (color sheme: xxbbgrr)
Value [0x0000FF00]                 //ColorWarn       (color sheme: xxbbgrr)
Value [0x000000FF]                 //ColorERR        (color sheme: xxbbgrr)
Value [0x00808080]                 //ColorUndefined (color sheme: xxbbgrr)
Value [0x00FFFFFF]                 //ColorMask       (0xFFFFFFFF : OFF)
Value [0x0007]                     //ValOn           (Enabled)
Value [0x0000]                     //ValOff          (Disabled)
Value [0x1007]                     //ValWarn         (Homed)
Value [0x0008]                     //ValErr          (Error)
Value [0x100F]                     //ValMask
Value [0]                          //IndicatorForm   (0:rounded, 1:squared)
Value [50]                          //IndicatorWidth
Value [50]                          //IndicatorHeight
Value [35]                          //IndicatorXOffs;
Value [35]                          //IndicatorYOffs;
```

Sample image:



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



2.9 InputValue

The input value allows to set a digital input value. It overlays the text to an image.

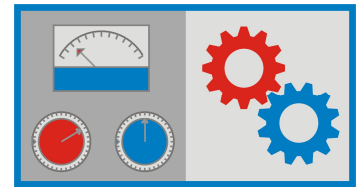
Sample: \Params\InputValue.PAR

```
String [Images\\Stahl_94xx_Meter.bmp] //Image file
String [InputValue] //Function name
String [DEC] //Unit name
Value [0] //ShowTitle;
Value [0] //MinVal;
Value [100] //MaxVal;
Value [0] //HEX Flag;
Value [30] //DigitXOffs;
Value [90] //DigitYOffs;
Value [30] //UnitXOffs
Value [40] //UnitYOffs
Value [100] //UnitFontSize
Value [400] //ValFontSize;
Value [0x00646464] //ColorOk
Value [0x000000FF] //ColorErr
Value [0x00FFFFFF] //ColorMask (0xFFFFFFFF : OFF)
```

Sample image:



HMI-DesignBox SDK Documentation



SYBERA GmbH Copyright © 2020

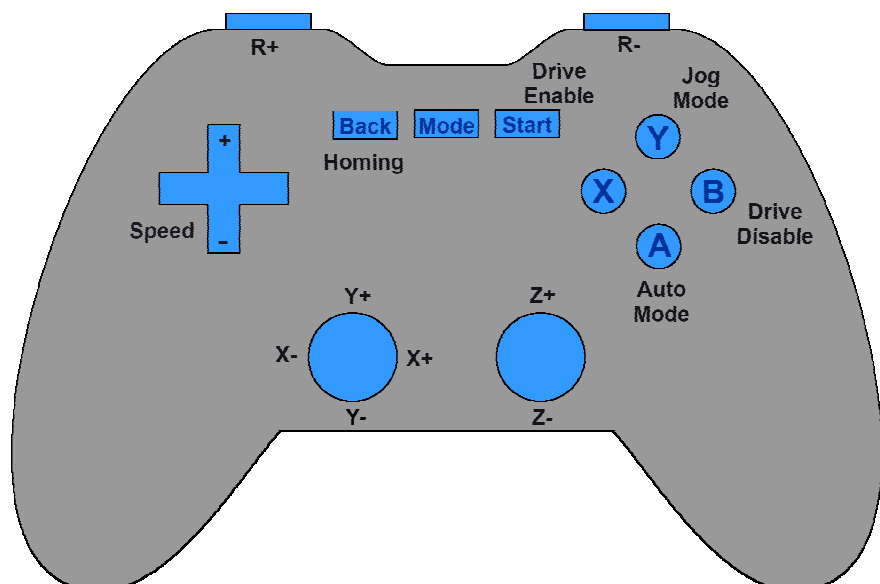
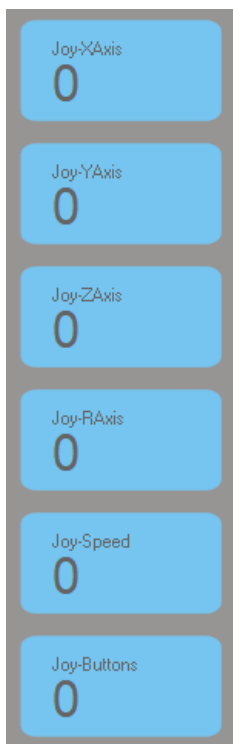
2.10 JoystickControl

The Joystick-Control Panel allows to make use of several joysticks or gamepads as input devices. Each Joystick Control Panel can be connected to a one joystick axis, or to a button (or a group of buttons).

Sample: \Params\InputValue.PAR

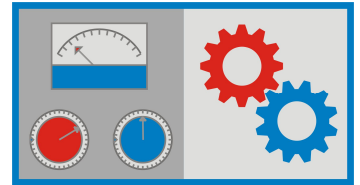
```
String [Images\\Joystick_Value.bmp]           //Image file
String [JoystickControl]                       //Function name
String [Joy-X-Axis]                            //Unit name
Value [0]                                     //ShowTitle;
Value [-100]                                  //MinVal;
Value [100]                                   //MaxVal;
Value [1]                                     //Axis Type (1:X, 2:Y, 3:Z, 4:R, 5:U, 6:V, 7:Buttons)
Value [500]                                   //Axis Scale
Value [1000]                                  //Axis Zero Threshold
Value [30]                                    //DigitXOffs;
Value [60]                                    //DigitYOffs;
Value [30]                                    //UnitXOffs
Value [30]                                    //UnitYOffs
Value [80]                                    //UnitFontSize
Value [250]                                   //ValFontSize;
Value [0x00646464]                           //ColorOk
Value [0x000000FF]                           //ColorErr
Value [0x00FFFFFF]                           //ColorMask (0xFFFFFFFF : OFF)
```

Sample: combination of 6 panels with gamepad for motion control



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



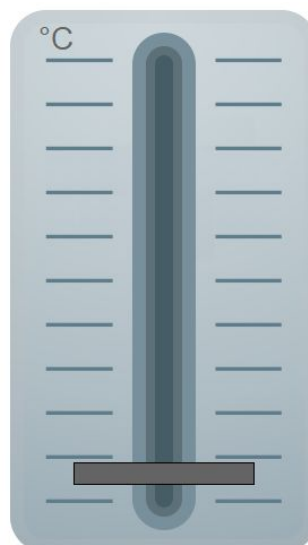
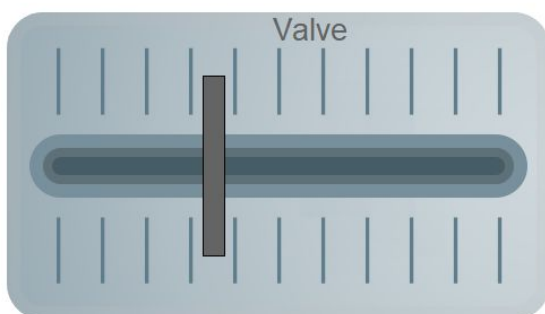
2.11 LinearControl

The linear control panel overlays a rectangle to an image, depending on the input/output value.

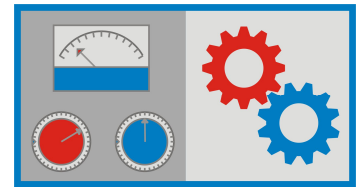
Sample: \Params\LinearControl (H).PAR

```
String [Images\\Schieberegler (H).bmp] //Image file
String [LinearControl] //Function name
String [Valve] //Unit name
Value [0] //ShowTitle;
Value [0] //MinVal;
Value [100] //MaxVal;
Value [1] //Scale;
Value [0] //SliderVertical;
Value [40] //SliderXOffs;
Value [136] //SliderYOffs;
Value [380] //SliderSpan;
Value [0] //ButtonXOffs;
Value [80] //ButtonYOffs;
Value [20] //ButtonWidth;
Value [160] //ButtonHeight;
Value [3] //ButtonBorder;
Value [235] //UnitXOffs
Value [25] //UnitYOffs
Value [200] //UnitFontSize
Value [0x00646464] //ColorOk
Value [0x000000FF] //ColorErr
Value [0x00FFFFFF] //ColorMask (0xFFFFFFFF : OFF)
```

Sample: horizontal and vertical slider



HMI-DesignBox SDK Documentation



SYBERA GmbH Copyright © 2020

2.12 DioModule

The DIO module overlays a colored rectangle or a circle (as LED) to an image. It may be used as output or input panel. Upto 32 LEDs may be overlaid.

`\Params\DioModule.PAR`

```
String [Images\\Stahl_9472_DIOM.bmp] //Image file
String [DioModule] //Function name
String [ ] //Unit name
Value [0] //ShowTitle;
Value [1] //ChangeLock (1: not user input allowed)
Value [0x0000FF00] //ColorON (color sheme: xxbbgrrr)
Value [0x00000000] //ColorOFF (color sheme: xxbbgrrr)
Value [0x000000FF] //ColorERR (color sheme: xxbbgrrr)
Value [0] //MinVal
Value [0xFFFF] //MaxVal
Value [1] //Scale
Value [1] //LedForm (0:rounded, 1:squared)
Value [10] //LedWidth
Value [20] //LedHeight
Value [16] //LedNum
```

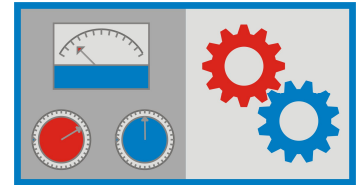
```
Value [51] //Led0XOffs;
Value [172] //Led0YOffs;
Value [66] //Led1XOffs;
Value [172] //Led1YOffs;
Value [81] //Led2XOffs;
Value [172] //Led2YOffs;
Value [96] //Led3XOffs;
Value [172] //Led3YOffs;
Value [110] //Led4XOffs;
Value [172] //Led4YOffs;
Value [125] //Led5XOffs;
Value [172] //Led5YOffs;
Value [140] //Led6XOffs;
Value [172] //Led6YOffs;
Value [154] //Led7XOffs;
Value [172] //Led7YOffs;
```

```
Value [169] //Led8XOffs;
Value [172] //Led8YOffs;
Value [183] //Led9XOffs;
Value [172] //Led9YOffs;
Value [197] //Led10XOffs;
Value [172] //Led10YOffs;
Value [212] //Led11XOffs;
Value [172] //Led11YOffs;
Value [228] //Led12XOffs;
Value [172] //Led12YOffs;
Value [242] //Led13XOffs;
Value [172] //Led13YOffs;
Value [257] //Led14XOffs;
Value [172] //Led14YOffs;
Value [272] //Led15XOffs;
Value [172] //Led15YOffs;
```



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



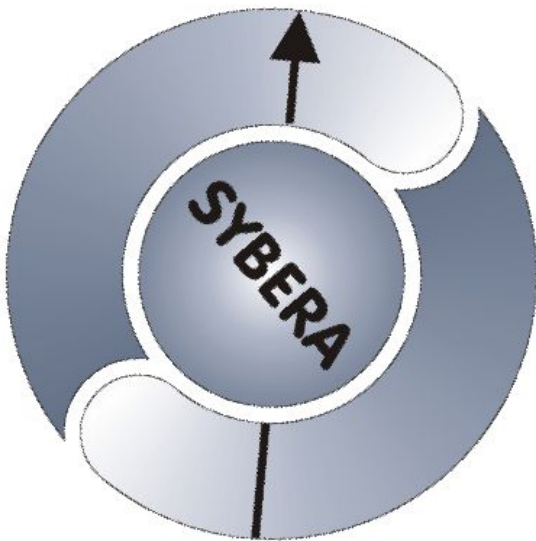
2.13 Motion

The motion panel allows a rotation image, depending on the output value. This allows e.g. to simulate the motion of a motor axis.

\Params\MotionModule.PAR

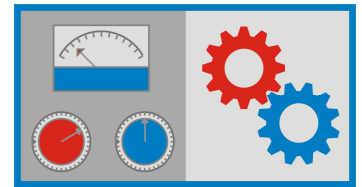
```
String [Images\\MotionModule.bmp] //Image file
String [Motion] //Function name
String [UPM] //Unit name
Value [0] //ShowTitle;
Value [1] //Rotation Direction;
Value [4096] //Rotation Inkrements
Value [0] //Degree Offset;
Value [0x00FFFFFF] //ColorMask (0xFFFFFFFF : OFF)
```

Sample: value dependend rotation image



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



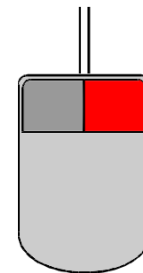
2.14 TeachAxis

The teach axis panel is to define a route of a multi axis device. Additionally it calculates the waypoints due to the given speed of each teach coordinate. The teach axis panel combines several axis position values and control values to teaching coordinates:

- X-Axis Actual Position
- Y-Axis Actual Position
- Z-Axis Actual Position
- R-Axis Actual Position
- Speed
- Control

DoubleClick on variable to select Panel Parameter File

Index	Variable Name	Value 1	Value 2
0	ActualPos_EMCA	[134003]	[0x00020b73]
1	ActualPos_IL2E	[1144428]	[0x0011766c]
2	ActualPos_BG75	[10995]	[0x00002af3]
3	ActualPos_PD4	[265]	[0x00000109]
4	Speed	[50]	[0x00000032]
5	Control	[0]	[0x00000000]
6	TargetPos_EMCA	[0]	[0x00000000]
7	TargetPos_IL2E	[0]	[0x00000000]
8	TargetPos_BG75	[0]	[0x00000000]
9	TargetPos_PD4	[0]	[0x00000000]

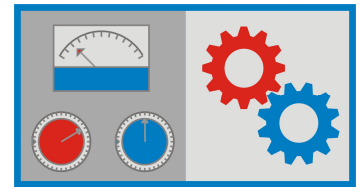


The Panel Parameter File describes one axis or control value for the TeachAxis Panel. Upto 4 axis value, one speed value and one control value can be selected as combination for the “TeachAxis” Panel

```
1 String [TeachCoord.par] //Teach Coordinate List Pathname
2 String [TeachAxis] //Function name
3 String [0.1 mm] //Unit name
4 Value [0] //Dialog Tag
5 Value [800] //Dialog Width
6 Value [500] //Dialog Height
7 Value [1] //Dialog Type (1:X-Axis, 2:Y-Axis, 3:Z-Axis, 4:R-Axis, 5:Speed, 6:Control)
8 Value [10000] //Unit Scale (per Meter)
9 Value [1000000] //Unit Steps (per Meter)
10
```

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



2.14.1 X-Axis Parameter File (Dialog Type 1)

```
String [TeachCoord.par] //Coordinate List Pathname
String [TeachAxis] //Function name
String [0.1 mm] //Unit name
Value [0] //Dialog Tag (used for additional teach dialogs)
Value [800] //Dialog Width
Value [500] //Dialog Height
Value [1] //Dialog Type (1:X-Axis, 2:Y-Axis, 3:Z-Axis, 4:R-Axis,
                    5:Speed, 6:Control)
Value [10000] //Unit Scale (per Meter)
Value [1000000] //Unit Steps (per Meter)
```

2.14.2 Additional Axis Parameter File (Dialog Type 2 – 4)

```
String [] //Reserved
String [TeachAxis] //Function name
String [] //Reserved
Value [0] //Dialog Tag (used for additional teach dialogs)
Value [0] //Reserved
Value [0] //Reserved
Value [2] //Dialog Type (1:X-Axis, 2:Y-Axis, 3:Z-Axis, 4:R-Axis,
                    5:Speed, 6:Control)
Value [10000] //Unit Scale (per Meter)
Value [2000000] //Unit Steps (per Meter)
```

2.14.3 Additional Speed Parameter File (Dialog Type 5)

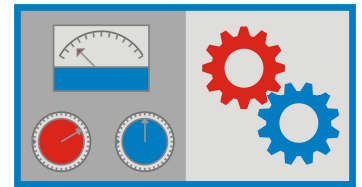
```
String [] //Reserved
String [TeachAxis] //Function name
String [] //Unit name
Value [0] //Dialog Tag (used for additional teach dialogs)
Value [0] //Reserved
Value [0] //Reserved
Value [5] //Dialog Type (1:X-Axis, 2:Y-Axis, 3:Z-Axis, 4:R-Axis,
                    5:Speed, 6:Control)
Value [1000] //Unit Scale (Speed [m/s])
Value [0] //Reserved
```

2.14.4 Additional Control Parameter File (Dialog Type 6)

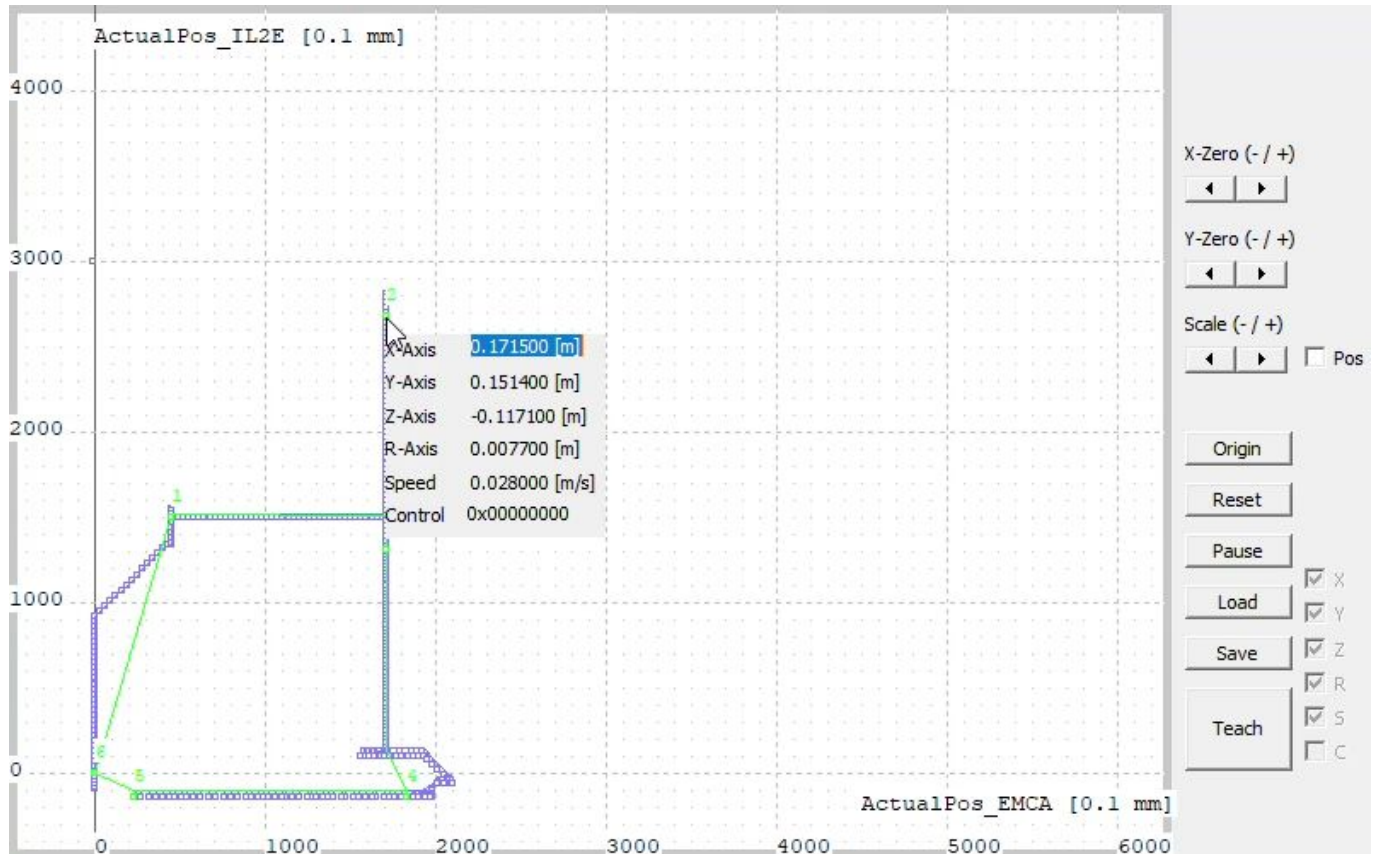
```
String [] //Reserved
String [TeachAxis] //Function name
String [] //Reserved
Value [0] //Dialog Tag (used for additional teach dialogs)
Value [0] //Reserved
Value [0] //Reserved
Value [6] //Dialog Type (1:X-Axis, 2:Y-Axis, 3:Z-Axis, 4:R-Axis,
                    5:Speed, 6:Control)
Value [0] //Reserved
Value [0] //Reserved
```

HMI-DesignBox SDK Documentation

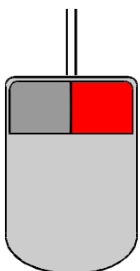
SYBERA GmbH Copyright © 2020



2.14.5 TeachAxis Dialog



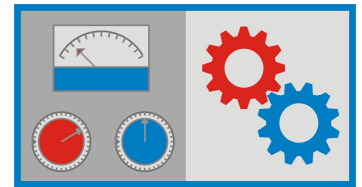
- Origin: Set origin to current axis position
- Reset: Delete current track points
- Load: Load coordinate list
- Save: Save coordinate list (with calculated waypoint list)
- Teach: Set current position as waypoint
- Scale: Change dialog scale
- X-Zero / Y-Zero: Move zero position



Right Mouseclick on TeachPoint:
Left Mouseclick on Teachpoint:

Delete Teachpoint
Connect to Teachpoint

HMI-DesignBox SDK Documentation



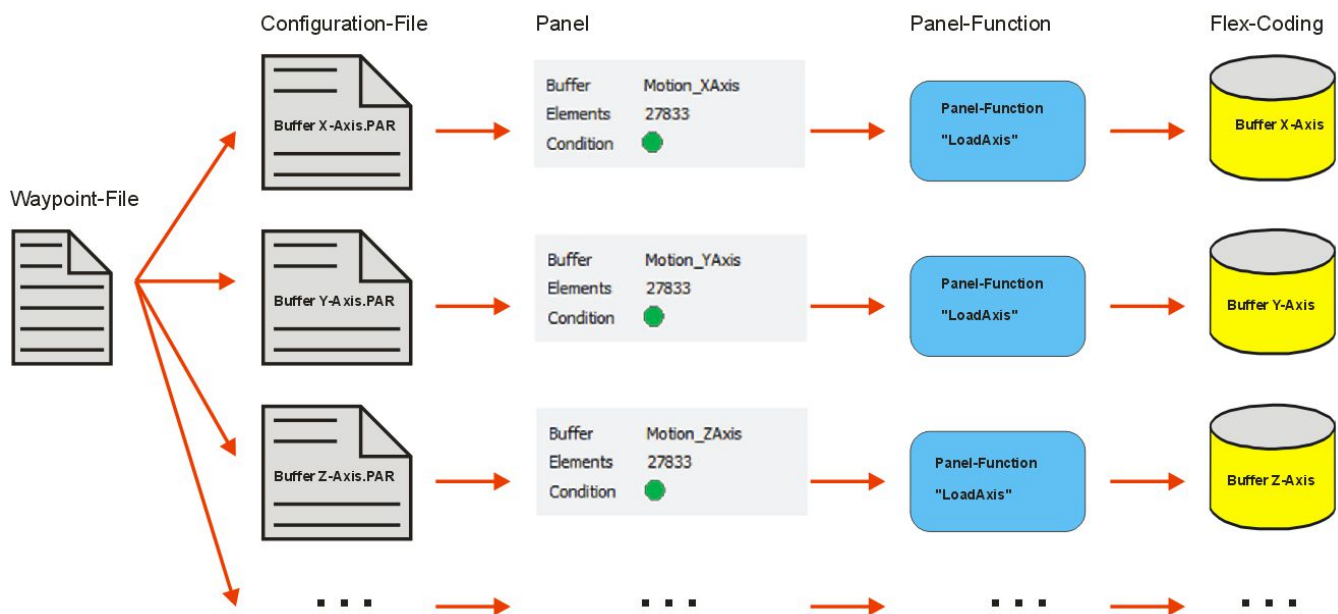
SYBERA GmbH Copyright © 2020

2.15 LoadAxis

The LoadAxis-Panel loads coordinates (X / Y / Z / R) from a calculated waypoint file (from TeachAxis) into a buffer. The calculated waypoint distance relates on the timing, the buffer is read at realtime, so the axis moves with the calculated speed.

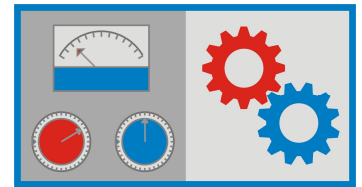
```
String [.\WeldInspect\TeachCoord.way] //Waypoint file
String [LoadAxis] //Function name
String [ ] //Reserved
Value [0] //ShowTitle
Value [1] //Axis Type (1:X-Axis, 2:Y-Axis, 3:Z-Axis, 4:R-Axis)
```

Sample of 3 axis buffers out of the waypoint file

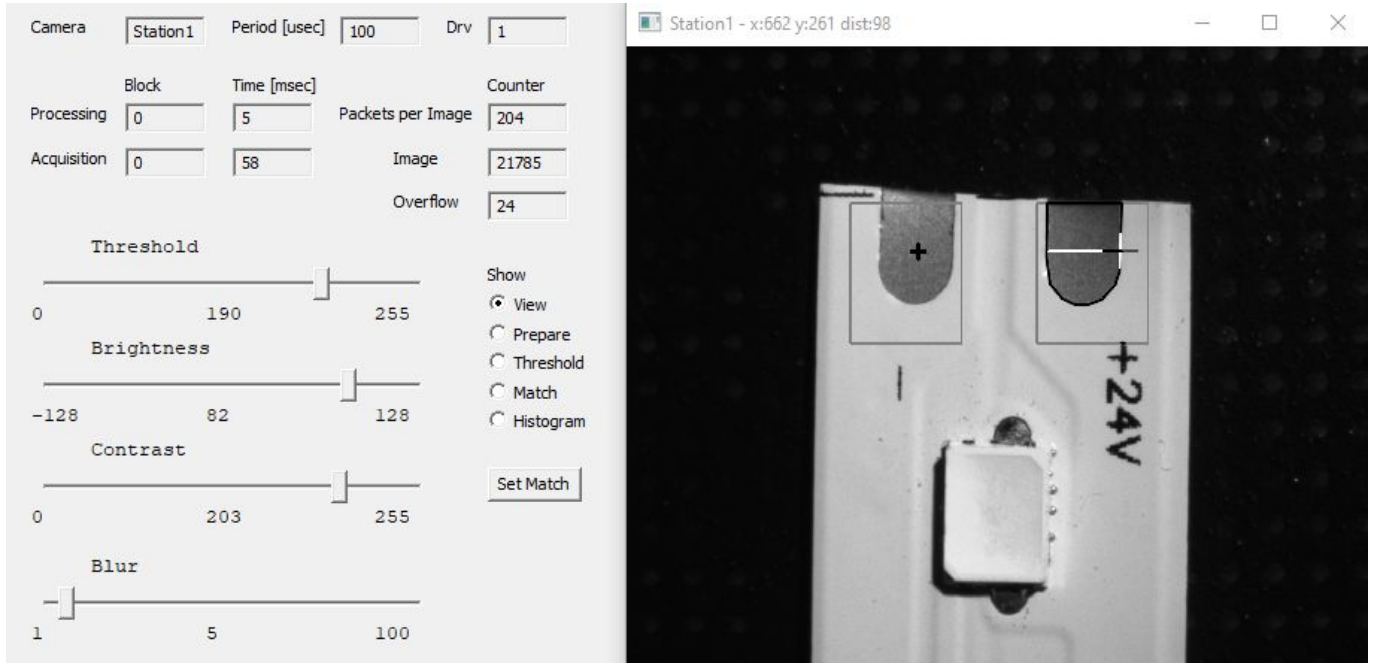


HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



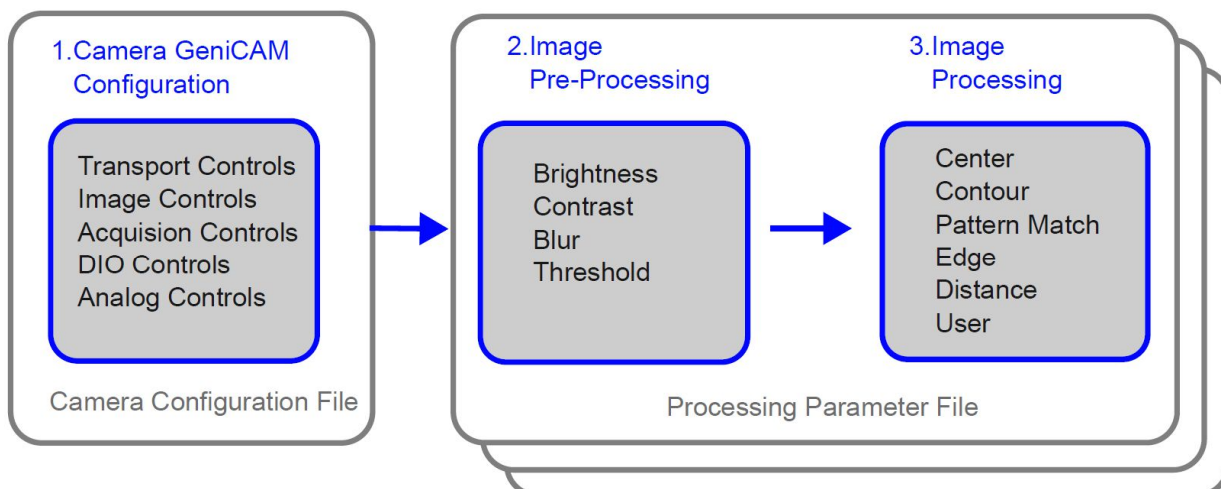
3 Smart Camera Extension



The smart camera extension is an additional package of the HmiDesignBox. It allows control industrial cameras based on the GigE-Vision standard. Image acquisition, as well as image processing will be done with easy to use control with direct connection to the Soft-PLC. With a basic set of image processing functions, like region center, region contour, pattern match, histogram the most important work can be done. Additionally a programming interface allows adding functions for image processing.

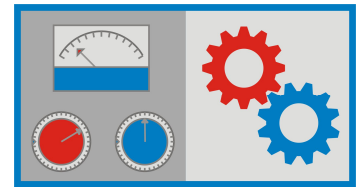
The smart camera extension is based on the GigE-Vision realtime core of Sybera, as well as on the OpenCV library. While the image acquisition is handled at realtime, the image processing is handled under windows in parallel. This allows a high speed industrial image processing connected to fieldbus systems.

There are 3 typical steps for image processing required:



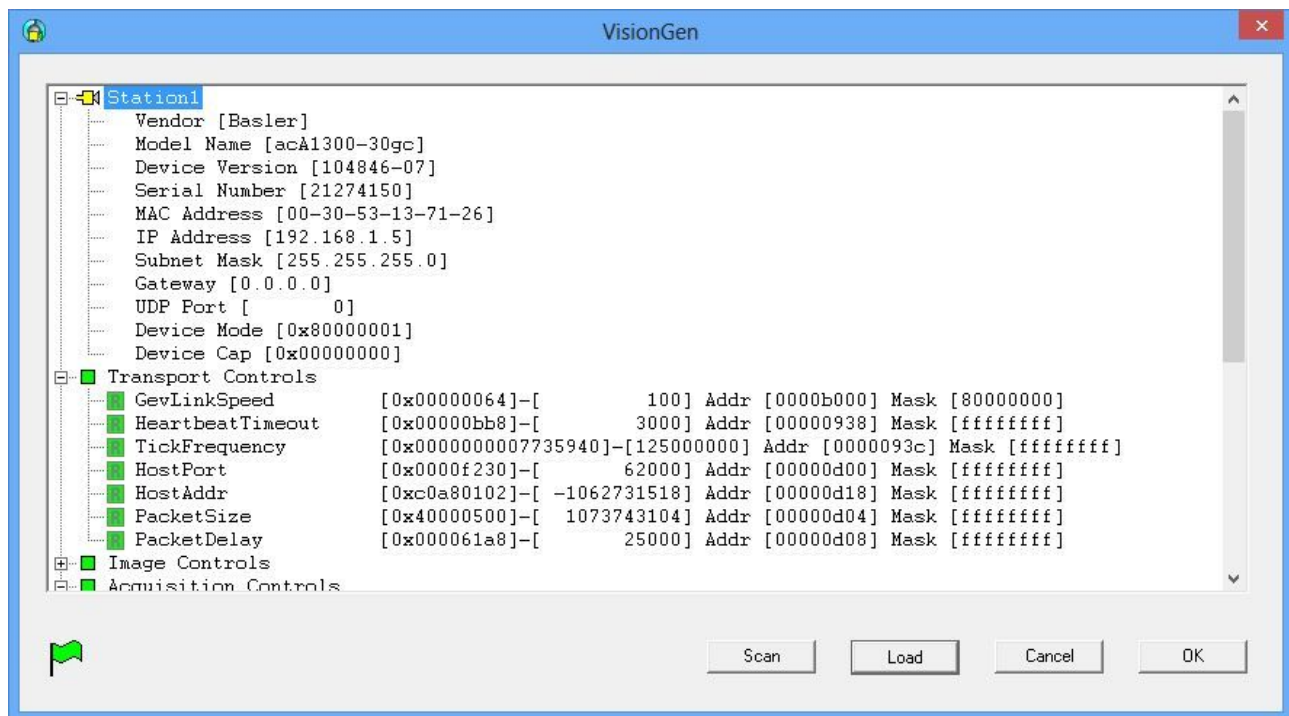
HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



3.1 Creating a Camera Configuration File



The configuration of a GigE vision camera is based on the GeniCam vision standard. A camera configuration consists at least of one accesspoint and several Genicam register groups. For proper operation of the SYBERA vision realtime core the GigE-Vision devices need first to be configured by creating a native configuration file. Therefore SYBERA provides a program called VISIONGEN.EXE.



Note: Make sure a valid IP address is provided for the network connection.

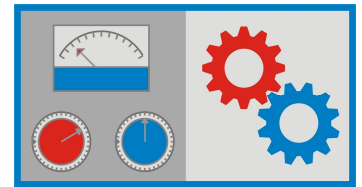
Note: If the application fails to run, check if the latest Microsoft XML Parser has been installed. If not, install in the directory \MSXML\MSXML6

VISIONGEN allows creating a native configuration file, with all required information for controlling the camera. This configuration is based on the GeniCam Standard, by selecting camera information from a provided XML camera file which must be present in the same directory as VISIONGEN. The creation of a configuration file can be done by scanning the bus and / or modifying the configuration manually. There are two types of information:

-  GigE Accesspoint
-  GeniCam Register Group

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



3.1.1 GigE Accesspoint

The accesspoint keeps all information required for connecting to the fieldbus, as station name, IP parameters, MAC address, identify parameters. Therefore first task is to collect information about the GigE-Vision configuration by scanning the bus. The scan gets information about manufacturer name and MAC address. Now individual assignment must set (e.g. IP address, station name). On a right button click to the selected GigE accesspoint the dialog appears:

Station Name: Station1
IP Address: 192.168.1.5 z.B. 192.168.0.2
Subnet Mask: 255.255.255.0 z.B. 255.255.255.0
Gateway: 0.0.0.0 z.B. 0.0.0.0

Buttons: Connect, Cancel, Set

3.1.1.1 Connecting a Camera

After scanning the bus, or after a configuration has been loaded, the selected camera may be connected

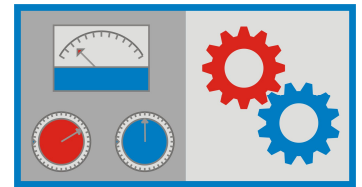
ImageCnt: 2097 PacketRate: 242 ImageFreq: 82400 [µsec]

Station1
Vendor [Basler]
Model Name [acA1300-30gc]
Device Version [104846-07]
Serial Number [21274150]
MAC Address [00-30-53-13-71-26]
IP Address [192.168.1.5]
Subnet Mask [255.255.255.0]
Gateway [0.0.0.0]
UDP Port [0]
Device Mode [0x80000001]
Device Cap [0x00000000]

Transport Controls
Image Controls
Acquisition Controls
DIO Controls
Analog Controls
Misc. Controls

Buttons: Scan, Load, Cancel, OK

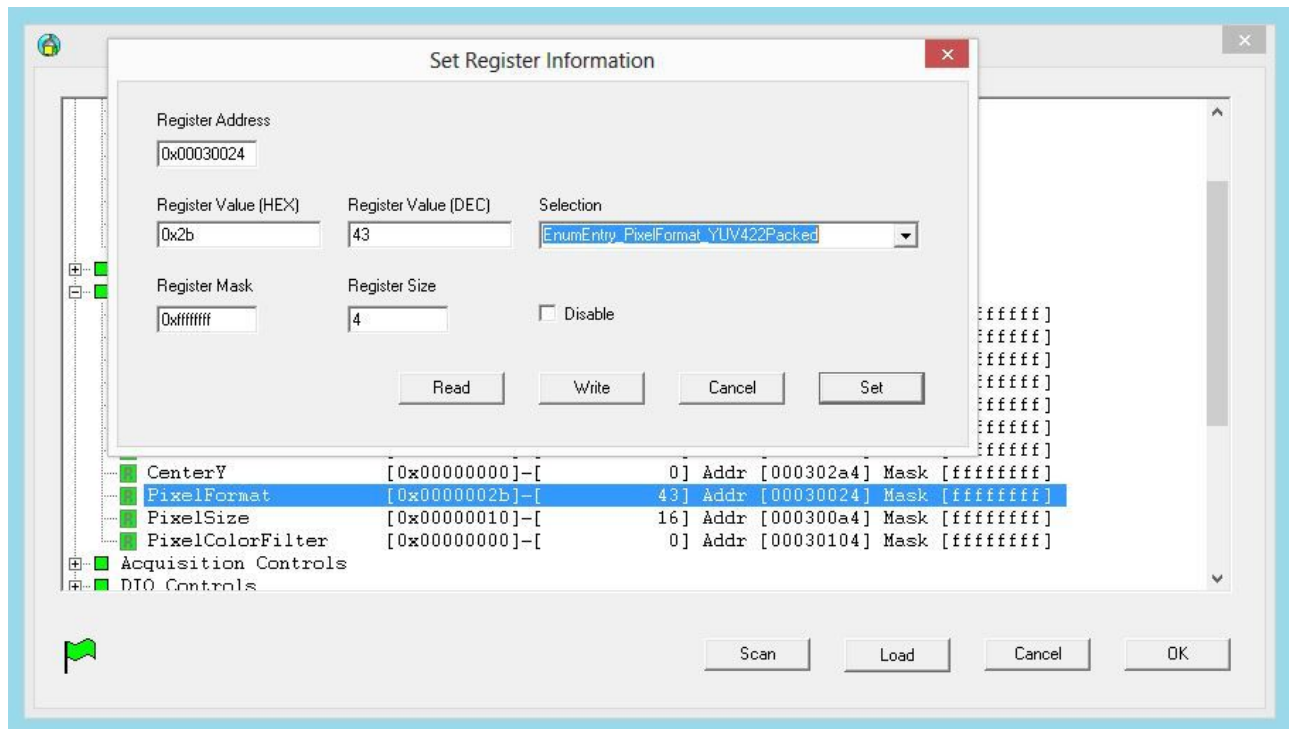
HMI-DesignBox SDK Documentation



SYBERA GmbH Copyright © 2020

3.1.2 GeniCam Register Groups

GeniCam is the standard for generic camera interfaces. Cameras, which are used in industrial image processing, are configurable more or less in the same way, by reading and writing of parameters to specified control register. However, protocols, formats, and addresses of these registers usually vary from manufacturer to manufacturer. The GeniCam registers are read or written via the acyclic GVCP protocol standard. On a right button click to the selected GeniCam register the dialog appears:



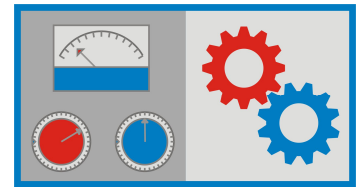
Note: A GeniCam register may also be disabled, so it will be ignored while initializing the camera. To disable a GeniCam register, either setting "Disable" or setting the register size to zero. Some GeniCam registers allow a control selection with a corresponding value. The GeniCam registers may be read or written while a camera is currently operating (camera is connected).

At least, typically following GeniCAM registers need to be configured to operate a camera:

- AcquisitionMode: 0x00040004
- Heartbeat Timeout: 0x0938
- Stream Channel Packet Size: 0x0D04
- Stream Channel Destination Address: 0x0D18
- Stream Channel Port: 0x0D00
- Pixel Format: 0x00030024
- Image Width: 0x00030204
- Image Height: 0x00030224
- Offset X: 0x00030244
- Offset Y: 0x00030264
- Acquisition Start: 0x00040024
- PacketSize: 0x40000500
- GainAuto: 0x00000002

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



Sample declaration of the smart camera configuration file within the smart camera parameter file

```
D:\Sybera\Xgo64\App\3D-Print\SmartCam (Contour).par - Notepad++
Datei Bearbeiten Suchen Ansicht Codierung Sprache Einstellungen Werkzeuge Makros Ausführen Plugins Fenster ?
SmartCam (Contour).par
1 String [c:\GEV\Basler acal300-30gc (00-30-53-13-71-26).par] //SmartCam Configuration File
2 String [SmartCam] //Panel function name
3 String [ ] //Unit
4 Value [0] //Dialog Tag
5 Value [1] //Item Index (0:First, 1:Second, 2:Third, ...)
6 Value [2] //Item Type (1:Center, 2:Contour, 3:Match, 4:User)
7 Value [1] //Driver Index
8 Value [100] //Period
9 Value [5] //Number of Image-Blocks
10 Value [0x300000] //Size of Image-Block
11 Value [100] //Image Window Position
```

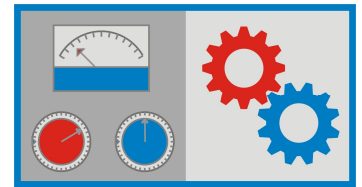
3.2 Creating a Processing Parameter File

The processing parameter file describes a region of interest (ROI) with the picture. For each ROI an additional parameter file is required. It defines:

- Smart camera configuration file based on the GENICAM standard
- ROI item index and type
- Camera driver index
- Acquisition sampling period
- Number of image acquisition blocks
- Size of image block
- Image window position and size
- ROI position and size
- Threshold slider
- Brightness slider
- Contrast slider
- Blur slider
- Item Parameters

The dialog tag defines a new item group. Each parameter file describes one item within a dialog group (with equal dialog tags). The item index describes the order of items, while one item must have index 0.

HMI-DesignBox SDK Documentation



SYBERA GmbH Copyright © 2020

Sample Processing Parameter file:

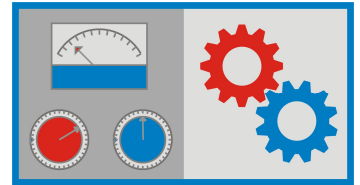
```
D:\Sybera\Xgo64\App\3D-Print\SmartCam (Contour).par - Notepad++
Datei Bearbeiten Suchen Ansicht Codierung Sprache Einstellungen Werkzeuge Makros Ausfuehren Plugins
Fenster ?
SmartCam (Contour).par x
1 |String [c:\\GEV\\Basler acal300-30gc (00-30-53-13-71-26).par] //SmartC
2 |String [SmartCam] //Panel function name
3 |String [ ] //Unit
4 |Value [0] //Dialog Tag
5 |Value [1] //Item Index (0:First, 1:Second, 2:Third, ...)
6 |Value [2] //Item Type (1:Center, 2:Contour, 3:Match, 4:User)
7 |Value [1] //Driver Index
8 |Value [100] //Period
9 |Value [5] //Number of Image-Blocks
10 |Value [0x300000] //Size of Image-Block
11 |Value [410] //Image Window X-Position
12 |Value [18] //Image Window Y-Position
13 |Value [500] //Image Window Width
14 |Value [300] //Image Window Height
15 |Value [550] //ROI X-Position (0 : no ROI)
16 |Value [200] //ROI Y-Position (0 : no ROI)
17 |Value [150] //ROI Width (0 : no ROI)
18 |Value [180] //ROI Height (0 : no ROI)
19 |Value [0] //Min. Value Slider1 (Threshold)
20 |Value [-128] //Min. Value Slider2 (Brightness)
21 |Value [0] //Min. Value Slider3 (Contrast)
22 |Value [1] //Min. Value Slider4 (Blur)
23 |Value [0] //Min. Value Slider5
24 |Value [255] //Max. Value Slider1 (Threshold)
25 |Value [128] //Max. Value Slider2 (Brightness)
26 |Value [255] //Max. Value Slider3 (Contrast)
27 |Value [100] //Max. Value Slider4 (Blur)
28 |Value [255] //Max. Value Slider5
29 |Value [180] //Preset Value Slider1 (Threshold)
30 |Value [95] //Preset Value Slider2 (Brightness)
31 |Value [153] //Preset Value Slider3 (Contrast)
32 |Value [20] //Preset Value Slider4 (Blur)
33 |Value [40] //Preset Value Slider5
34 |Value [1] //Show Slider1 (Threshold)
35 |Value [1] //Show Slider2 (Brightness)
36 |Value [1] //Show Slider3 (Contrast)
37 |Value [1] //Show Slider4 (Blur)
38 |Value [0] //Show Slider5
length: 1.726 lines: 39 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS
```

Note:

Dialog tag	Tag of dialog item group
Item index	Index of the item within the item group (starting with 0)
Item type	Item function type (region center, region contour, pattern match)
Driver Index	Index of the realtime driver (starting with 0)
Period	Realtime sampling period (typically 100 usec)
Number of Image Blocks	Ring buffer for image acquisition and processing in parallel
Image Block Size	Size of raw image format

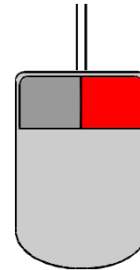
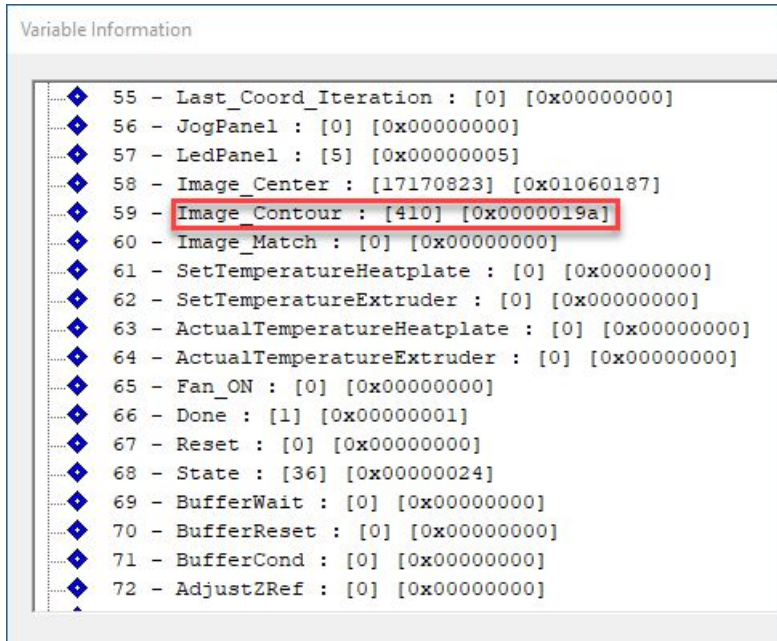
HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



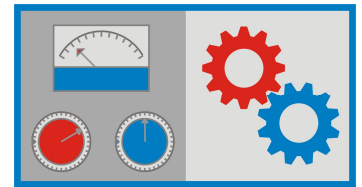
3.3 Connect a PLC variable within X-GO Logic Control

Right-Click on variable within to select smart camera parameter file



HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



3.4 Image Processing Functions

The smart camera extension provides a set of basic functions for image processing. Additionally a user interface is provided to extend the image processing by more complex operations. The image processing depends on the pre-setting of brightness and contrast, as well on threshold and blur.

3.4.1 Function “Region Center”

The Function “Region Center” gets the mass center of an object within a region of interest. The resulting X/Y coordinates are 16 bit values of pixel coordinates.

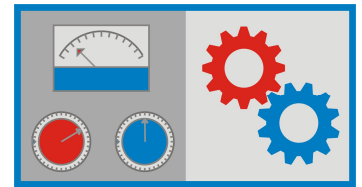
The screenshot displays the HMI-DesignBox SDK interface for the 'Region Center' function. The interface is divided into several sections:

- Camera Settings:** Station1, Period [usec] 100, Drv 1.
- Processing Parameters:** Block 3, Time [msec] 3, Packets per Image 204, Counter 3079.
- Acquisition Parameters:** Acquisition 4, Image 58, Overflow 45.
- Image Processing Sliders:** Threshold (0 to 255, value 224), Brightness (-128 to 128, value 42), Contrast (0 to 255, value 208), Blur (1 to 100, value 8).
- Show Options:** View (selected), Prepare, Threshold, Match, Histogram.
- Set Match:** A button to apply the current settings.
- Image View:** A window titled 'Station1 - x:688 y:552 dist:0' showing a grayscale image of a circular object with a crosshair indicating the region center.
- SmartCamTst Results:** A table showing the resulting X and Y coordinates.

	32-Bit	16-Bit (0)	16-Bit (1)
Value 1	27525635	515	420
Value 2	0	0	0
Value 3	0	0	0

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



3.4.2 Function “Region Contour”

The Function “Region Contour” gets the length of specified object contour within a region of interest. The resulting pixel length is a 32 bit value. The contour index is specified as item parameter 1 in the parameter file. If the contour index in the parameter file is defined as -1, then all contour lengths are summarized.

The screenshot displays the SmartCamTest software interface. On the left is a configuration panel with various settings:

- Camera: Station1, Period [usec]: 100, Drv: 1
- Block: Processing: 3, Time [msec]: 4, Packets per Image: 204, Counter: 204
- Acquisition: 3, Image: 4378, Overflow: 49
- Threshold: 0 to 255 (slider at 235)
- Brightness: -128 to 128 (slider at 95)
- Contrast: 0 to 255 (slider at 147)
- Blur: 1 to 100 (slider at 10)
- Show: View (selected), Prepare, Threshold, Match, Histogram
- Set Match button

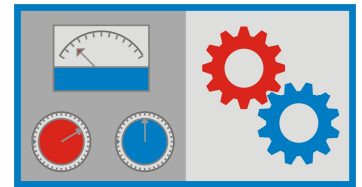
The main window shows a camera view titled "Station1 - x:993 y:795 dist:0" with a red contour index overlay. The contour index is labeled "Contour Index" and shows three contours with indices 1, 2, and 3. The innermost contour is labeled 1, the middle one 2, and the outermost one 3.

Below the camera view is a "SmartCamTest" window titled "Contour Length" with a table of results:

	32-Bit	16-Bit (0)	16-Bit (1)
Value 1	0	0	0
Value 2	3425	3425	0
Value 3	0	0	0

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



3.4.3 Function “Match Pattern”

The Function “Match Pattern” has two implementations:

- Match Template
- Match Shapes

While “Match Template gets the x/y coordinates of the matching object, “Match Shapes” gets a similarity value of two objects. The matching type is defined by Param1 in the parameter file.

Match Template:

Define matching template (once)

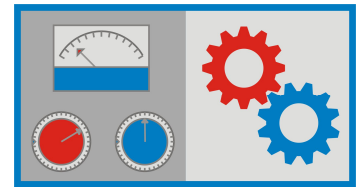
	32-Bit	16-Bit (0)	16-Bit (1)
Value 1	0	0	0
Value 2	0	0	0
Value 3	27853325	525	425

Match Shape:

	32-Bit	16-Bit (0)	16-Bit (1)
Value 1	0	0	0
Value 2	0	0	0
Value 3	5675	5675	0

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



3.4.4 Function “Edge”

The Function “Edge” gets the X / Y position of a contour edge on a line. The resulting X/Y coordinates are 16 bit values.

Station1 - cur:852,786 pt0:0,0 pt1:0,0 dist:0

Camera: Station1, Period [usec]: 100, Drv: 1

Block: Processing 0, Time [msec] 4, Packets per Image 204, Counter 204

Acquisition 0, Image 1736, Overflow 6

Threshold: 0 to 255 (241)

Brightness: -128 to 128 (63)

Contrast: 0 to 255 (186)

Blur: 1 to 100

Show: View, Prepare, **Threshold**, Match, Histogram

Set Match

SmartCamTst

Edge Point X / Y

	32-Bit	16-Bit (0)	16-Bit (1)
Value 1	0	0	0
Value 2	0	0	0
Value 3	27394347	299	418

3.4.5 Function “Distance”

The Function “Distance” gets the length of two contour edges on a line. The resulting pixel distance is a 32 bit value.

Station1 - cur:145,224 pt0:0,0 pt1:0,0 dist:0

Camera: Station1, Period [usec]: 100, Drv: 1

Block: Processing 2, Time [msec] 4, Packets per Image 204, Counter 204

Acquisition 2, Image 3506, Overflow 21

Threshold: 0 to 255 (241)

Brightness: -128 to 128 (63)

Contrast: 0 to 255 (186)

Blur: 1 to 100

Show: View, Prepare, **Threshold**, Match, Histogram

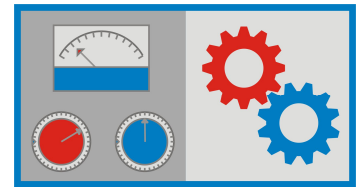
Set Match

SmartCamTst

Distance

	32-Bit	16-Bit (0)	16-Bit (1)
Value 1	0	0	0
Value 2	0	0	0
Value 3	117	117	0

HMI-DesignBox SDK Documentation



SYBERA GmbH Copyright © 2020

3.4.6 Function “User”

The Function “User” allows to implement an own image processing algorithm based on the functions of OpenCV. Therefore the C++ library “IMGPROCLIB” with its class CImgProc is provided as open source. The functional interface is as following:

CImgProc class members:

```
cv::Mat m_MatRaw;           //MAT with raw image
cv::Mat m_MatView;         //MAT for image window
cv::Mat m_MatPrep;         //MAT with adjusted image
cv::Mat m_MatThres;        //MAT with threshold image
PSLIDER_INFO m_pSliderList; //Slider values

ULONG CImgProc::GetUser (
    ULONG ROIXPath,           //ROI X-Position, out of parameter file
    ULONG ROIYPos,           //ROI Y-Position, out of parameter file
    ULONG ROIWidth,          //ROI Width, out of parameter file
    ULONG ROIHeight,         //ROI Height, out of parameter file
    PLONG pValue,             //Pointer to result value
    LONG Param1,              //Param1, out of parameter file
    LONG Param2,              //Param2, out of parameter file
    LONG Param3,              //Param3, out of parameter file
    LONG Param4,              //Param4, out of parameter file
    LONG Param5)              //Param5, out of parameter file
{
    //Define color
    cv::Scalar color(128,128,128);

    //Get ROI rectangle
    cv::Rect RectROI = __GetRectROI(
        m_MatView,           //MAT for image window
        ROIXPath,
        ROIYPos,
        ROIWidth,
        ROIHeight);

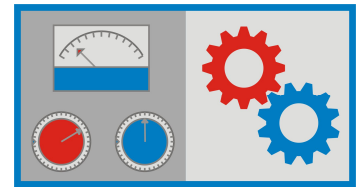
    //Get ROI matrix
    cv::Mat MatROI = m_MatPrep(RectROI);

    //*****
    //Todo : User implementation of OpenCV image processing
    //*****

    //Return function value
    If (pValue) { *pValue = 0; }
    return ERROR_SUCCESS;
}
```

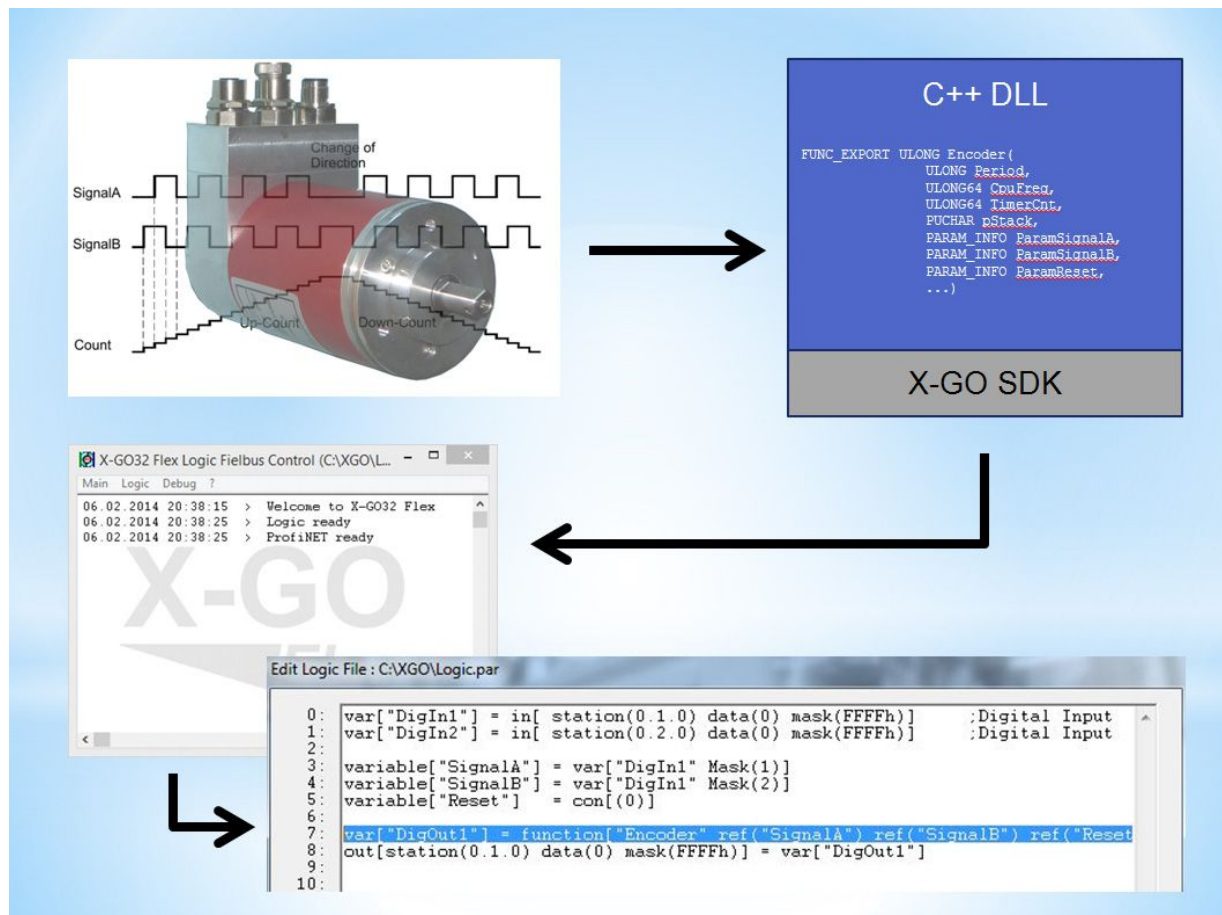
HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



4 HMI-SDK

With the SDK, new HMI function modules can also be developed and integrated or changed into the existing user HMI interface. With a simple C programming interface, additional HMI function modules (e.g. drive modules for complex curves or encoder modules) can be created and integrated into the HMI-DesignBox.

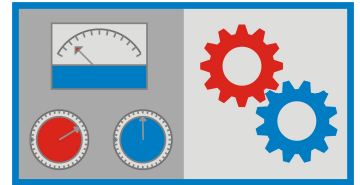


4.1 C++ Export functions

```
typedef struct _PANEL_INFO  
{  
    char szName[MAX_PATH];  
    char szParamPath[MAX_PATH];  
    char szParamFile[MAX_PATH];  
    LONG XPos  
    LONG YPos  
    LONG ZOrder  
    HANDLE hPanel  
  
} PANEL_INFO, *PPANEL_INFO;
```

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



```
FUNC_EXPORT ULONG LoadParams (
    LPCTSTR szParamFile,
    LPCTSTR szParamPath,
    LPSTR szImageFile,
    LPSTR szFunc,
    LPSTR szUnit,
    PLONG pParams,
    ULONG MaxParams)

FUNC_EXPORT ULONG CreatePanel (
    PPANEL_INFO pPanelInfo,
    ULONG Period,
    PLONG pValue)

FUNC_EXPORT void DestroyPanel (PPANEL_INFO pPanelInfo)

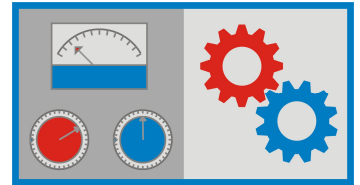
FUNC_EXPORT ULONG GetPanelPos (
    PPANEL_INFO pPanelInfo,
    PLONG pXPos,
    PLONG pYPos,
    PLONG pWidth,
    PLONG pHeight,
    PLONG pZOrder)
```

4.1.1 Sample function call C++

```
strcpy(m_PanelInfo[0].szName, "LcdMeter");
strcpy(m_PanelInfo[0].szParamPath, "..\\..\\..\\Params");
strcpy(m_PanelInfo[0].szParamFile, "LcdMeter.par");
m_PanelInfo[0].rect.top = 100;
m_PanelInfo[0].rect.left = 100;
CreatePanel(&m_PanelInfo[0], 2000, &m_Value[0]);
```

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



4.2 SDK template prototype

```
FUNC_EXPORT HANDLE MyPanel(  
    LPCTSTR szName,           //Panel name  
    LPCTSTR szUnit,          //Panel Unit  
    LPCTSTR szImageFile,     //Image file  
    PLONG pParams,           //Panel parameter list  
    ULONG ParamNum,          //Number of parameters  
    ULONG Period,            //Cycle Period  
    PLONG pValue)            //Current panel value
```

4.2.1 Parameter list structure

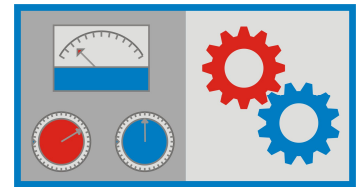
```
//Define parameter list within XGOPANEL64.H  
typedef struct _MY_PANEL_INFO  
{  
    LONG Param1;  
    LONG Param2;  
    LONG Param3;  
    ...  
} MY_PANEL_INFO, *PMY_PANEL_INFO;
```

4.2.2 Template file

```
String [Images\\MyPanel.bmp] //Image file  
String [MyPanel]             //Function name  
String [usec]                 //Unit name  
Value [0]                     //Param1;  
Value [100]                   //Param2;  
Value [0x0F]                  //Param3;
```

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020

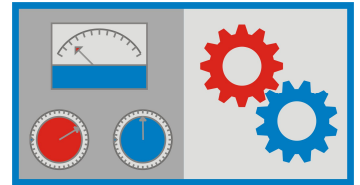


4.2.3 Sample

```
//*****  
//*** Export functions *****  
//*****  
  
FUNC_EXPORT HANDLE AnalogControl(  
    LPCTSTR szName,  
    LPCTSTR szUnit,  
    LPCTSTR szImageFile,  
    PLONG pParams,  
    ULONG ParamNum,  
    ULONG Period,  
    PLONG pValue)  
{  
    //Check parameter size  
    if (ParamNum != sizeof(ANALOG_CONTROL_INFO) / sizeof(LONG))  
        return NULL;  
  
    //Create dialog  
    CWnd wnd;  
    CAnalogControlDlg* pDlg = new CAnalogControlDlg(szName, szUnit, szImageFile,  
        pValue, (PANALOG_CONTROL_INFO)pParams);  
    pDlg->Create(IDD_ANALOG_CONTROL_DLG, &wnd);  
    return (HANDLE)pDlg;  
}  
  
//*****  
//*** Function Class *****  
//*****  
  
CAnalogControlDlg::CAnalogControlDlg(  
    LPCTSTR szName,  
    LPCTSTR szUnit,  
    LPCTSTR szImageFile,  
    PLONG pValue,  
    PANALOG_CONTROL_INFO pInfo,  
    CWnd* pParent /*=NULL*/) :  
    CDialog(CAnalogControlDlg::IDD, pParent)  
{  
    //Init member elements  
    m_szImageFile = szImageFile;  
    m_szName = szName;  
    m_szUnit = szUnit;  
    m_pValue = (PLONG)pValue;  
    m_bPrintPos = FALSE;  
    m_bGridPos = FALSE;  
    m_bUnlock = FALSE;  
  
    //Init analog meter information  
    memcpy(&m_Info, pInfo, sizeof(ANALOG_CONTROL_INFO));  
}
```

HMI-DesignBox SDK Documentation

SYBERA GmbH Copyright © 2020



```
CAnalogControlDlg::~CAnalogControlDlg()
{
    //Stop the timer
    KillTimer(NULL);
}

void CAnalogControlDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAnalogControlDlg, CDialog)
    ON_WM_ERASEBKGND()
    ON_WM_PAINT()
    ON_WM_TIMER()
    ON_WM_LBUTTONDOWN()
    ON_WM_KEYDOWN()
    ON_WM_EXITSIZEMOVE()
END_MESSAGE_MAP()

//*****
//*** CAnalogControlDlg message handlers *****
//*****

BOOL CAnalogControlDlg::OnInitDialog()
BOOL CAnalogControlDlg::OnEraseBkgnd(CDC* pDC)
void CAnalogControlDlg::OnPaint()
void CAnalogControlDlg::OnTimer(UINT_PTR nIDEvent)
void CAnalogControlDlg::OnLButtonDown(UINT nFlags, CPoint point)
void CAnalogControlDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
void CAnalogControlDlg::OnExitSizeMove()
```