

# DACHSview

the graphical way of real-time programming  
for rapid time to market

## DACHSview-SDL

is the first »all-in-one« programming tool with  
seamless development environment for different  
Real-Time Linux (kernel 2.6.3x)-  and

**QNX**-based event-oriented applications in real-  
time. One tool for accessing I/O ports, physical  
memory, fieldbus I/O points, SQL data bases, and  
single pixels on the screen of a visualization.



ASi

ETHERNET  
POWERLINK

MODBUS

EtherCAT



CANopen

CAN

LON



**STEINHOFF**  
AUTOMATION & FIELDBUS-SYSTEMS



**DACHS**  
...the better Idea!

MADE IN GERMANY

[www.steinhoff-automation.com](http://www.steinhoff-automation.com)

## DACHSview-SDL

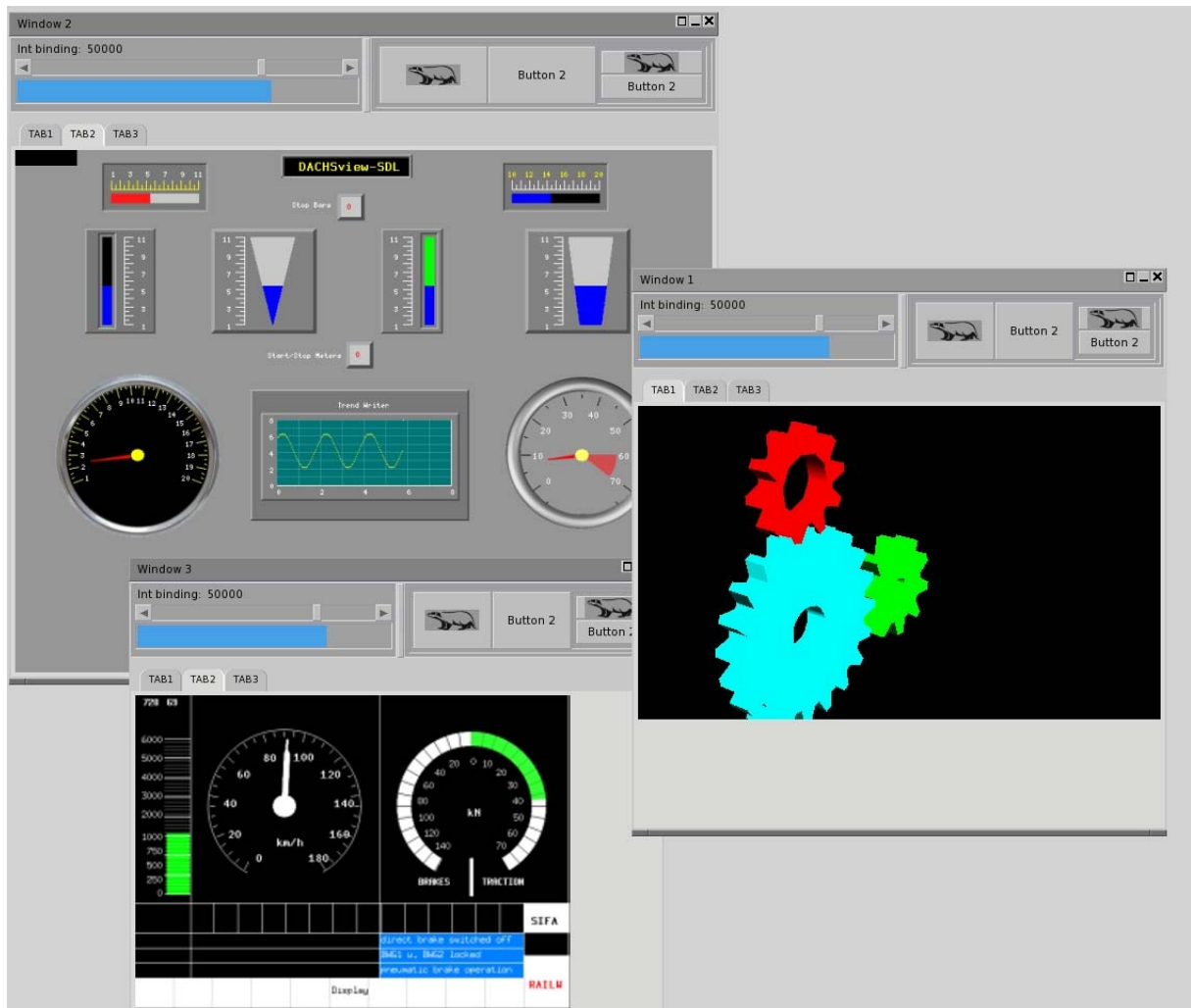
THE INNOVATIVE GRAPHICAL AND REAL-TIME PROGRAMMING & GUI TOOL  
WITH THE POWER OF C/C++ PROGRAMMING

### DACHSview-SDL

is the answer on increasing complexity of development for industrial automation applications. It's an universal and modular Real-Time Soft-PLC which includes additional a GUI-development environment. Comfortable development by using pre-defined and user-defined Function Blocks for **rapid time to market!**

### DACHSview

is a graphical function block language which covers all aspects of today's industrial automation by its data flow- and function block orientation. **Uniform programming environment** with support of I/O ports, physical memory, fieldbus I/O points, SQL data bases, and for development of complex animated graphics. **C-applications can be integrated easily by message-passing.**

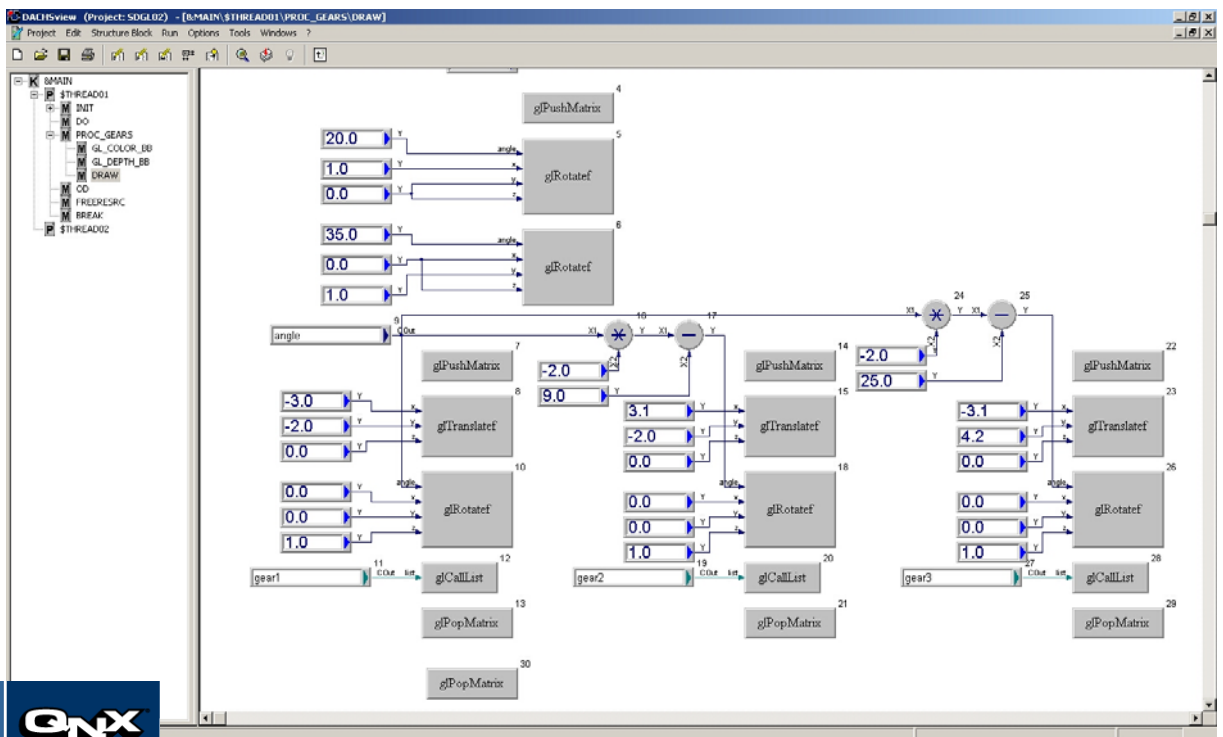
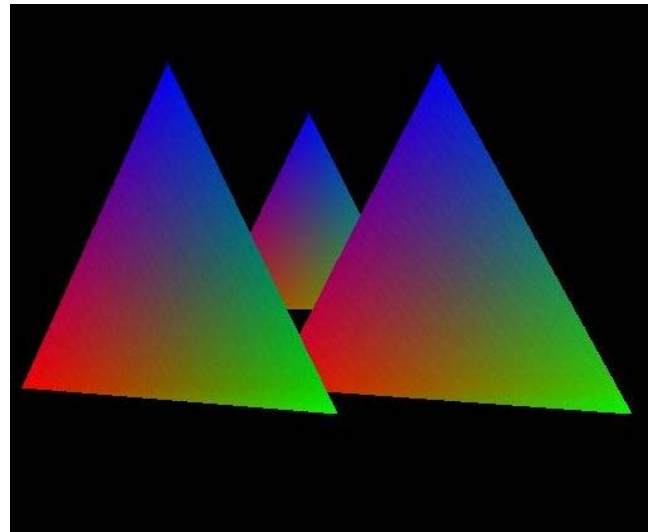
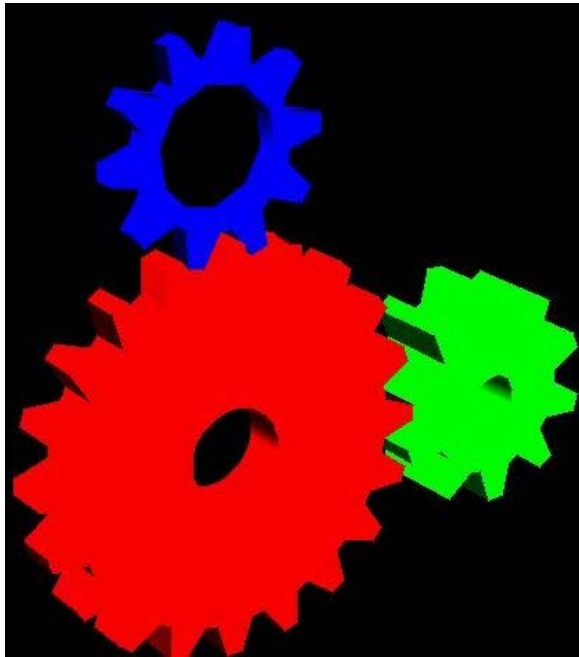


Demo Applications: Real-Time Linux Target or QNX 6.x Target running under QNX Photon



## DACHSview-SDGL - OpenGL for 3D graphics in real-time

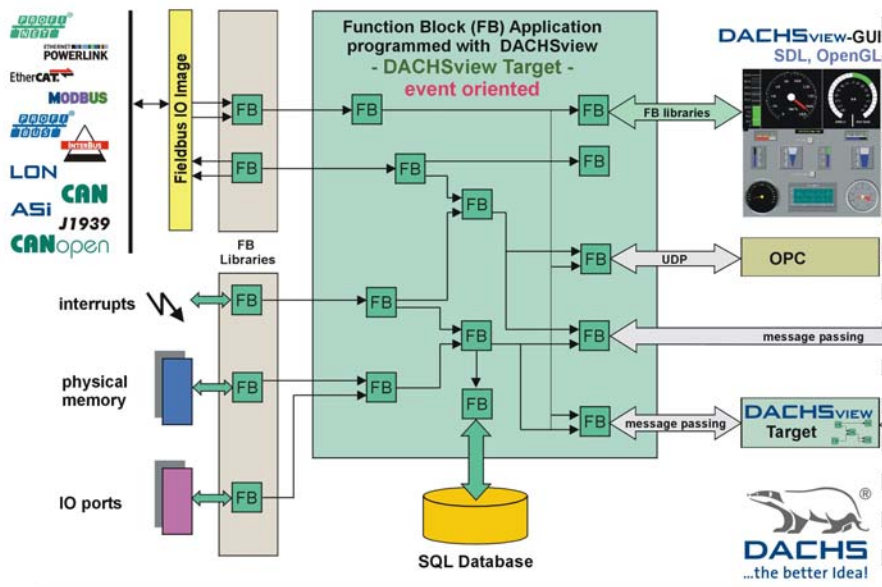
Works with low level graphical interfaces like the VESA or SVGA interface



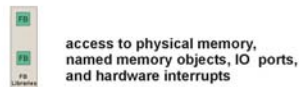
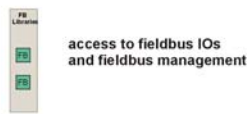
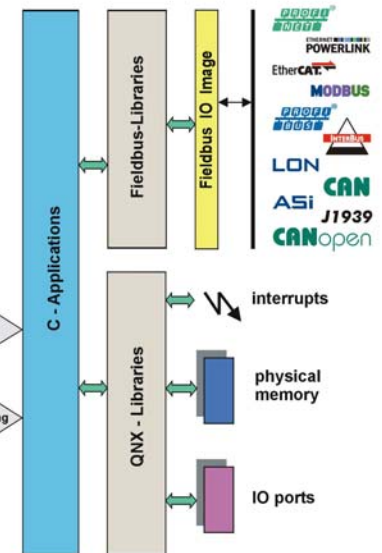
**DACHS** - Components for QNX Neutrino RTOS v. 6.x   or Real-Time-Linux  
for PC and Fieldbus based Real-Time and GUI Applications



**FUNCTION BLOCK PROGRAMMING,  
IEC61131-3 AND IEC61499 BASED**



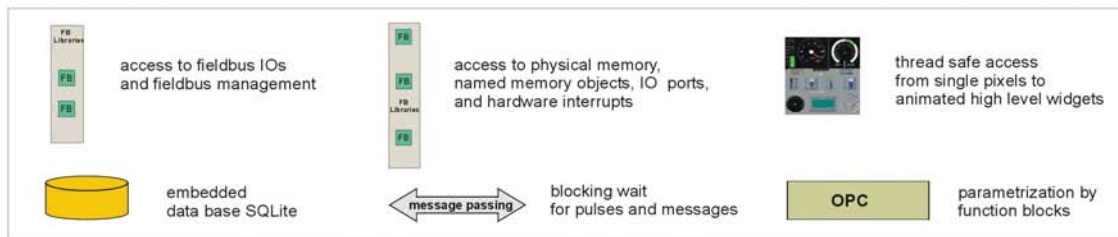
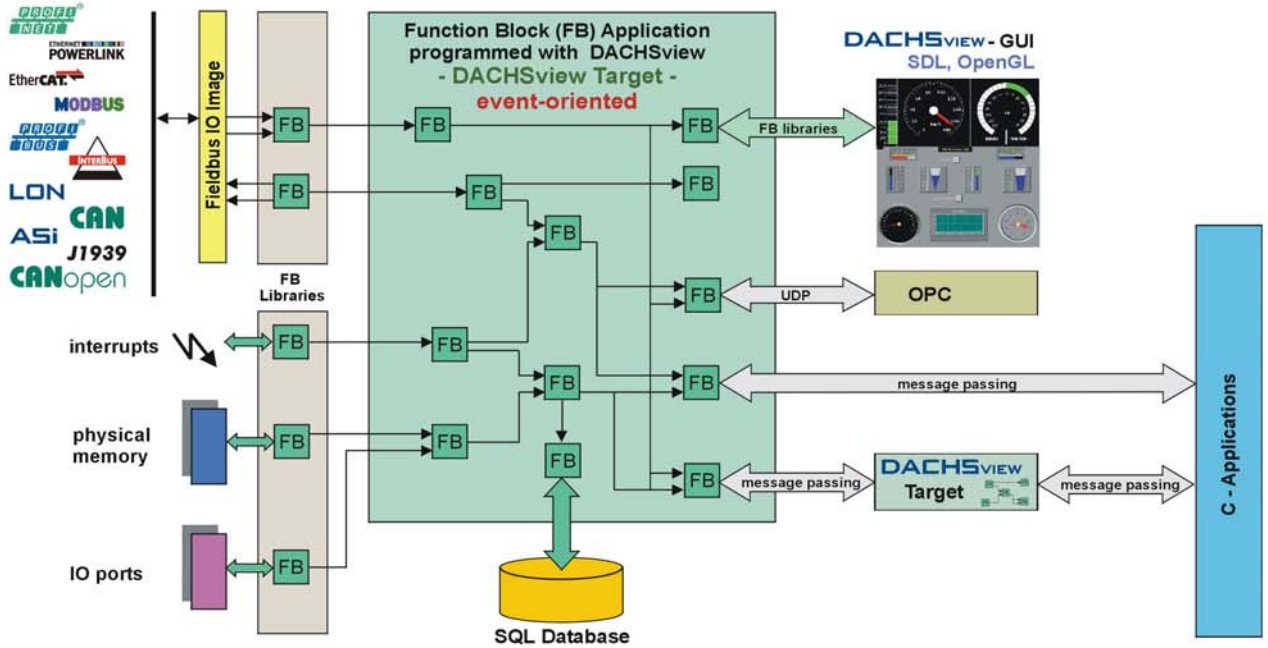
**C-PROGRAMMING**



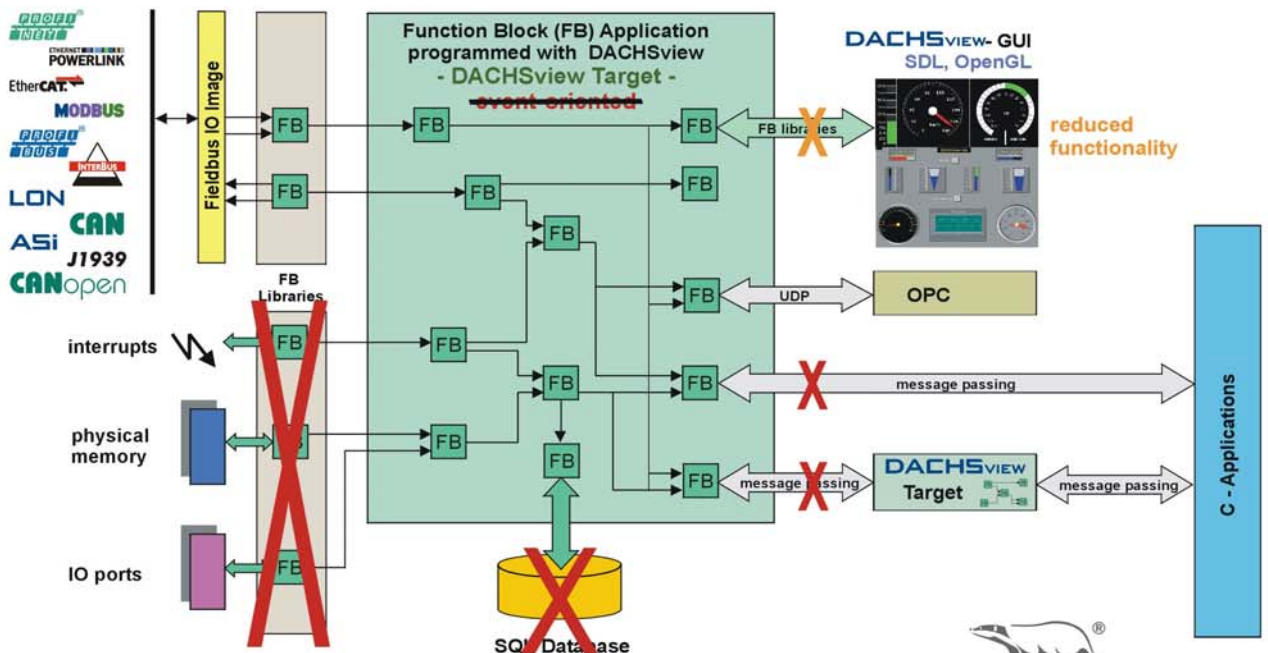
thread safe access from single pixels to animated high level widgets



## ADVANCED FUNCTION BLOCK PROGRAMMING, IEC61131-3 AND IEC61499 BASED

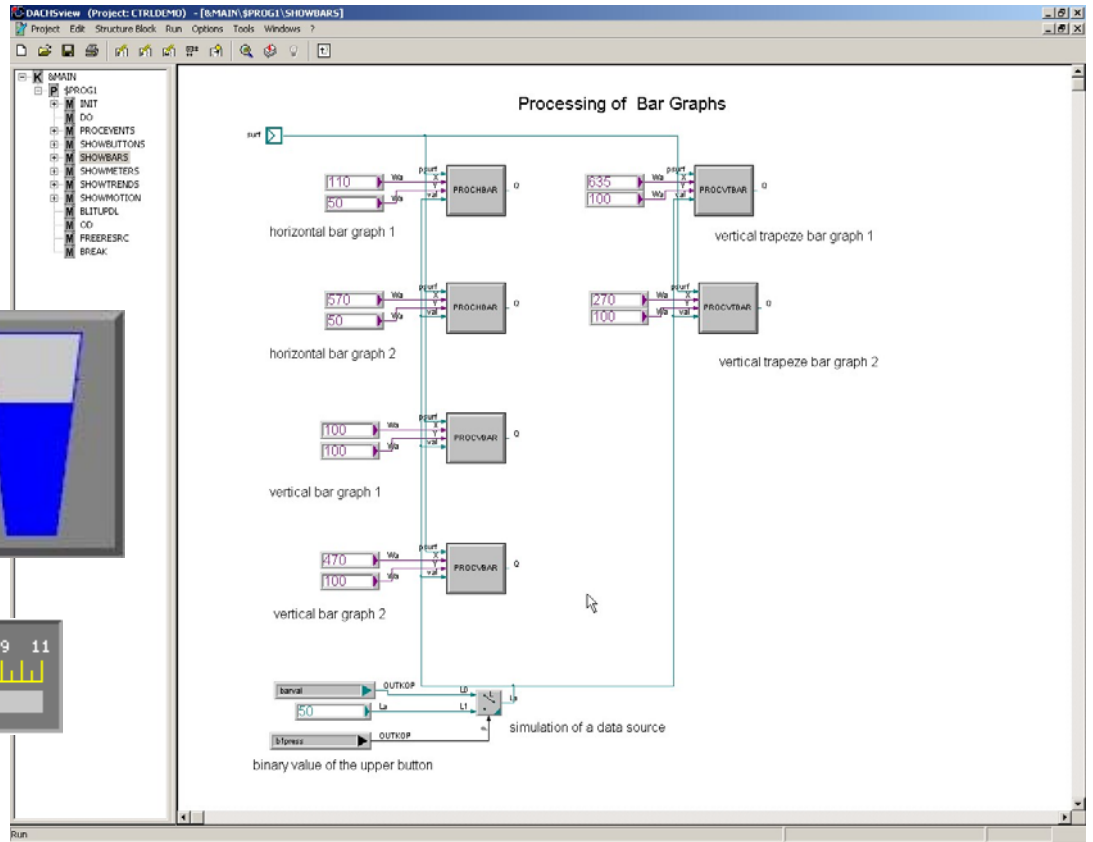
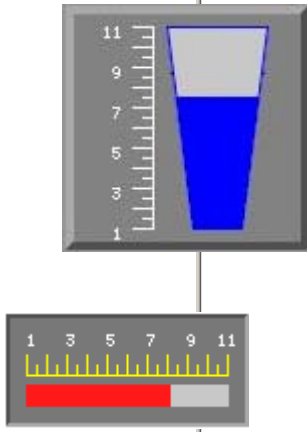


## ONLY IEC61131-3 BASED

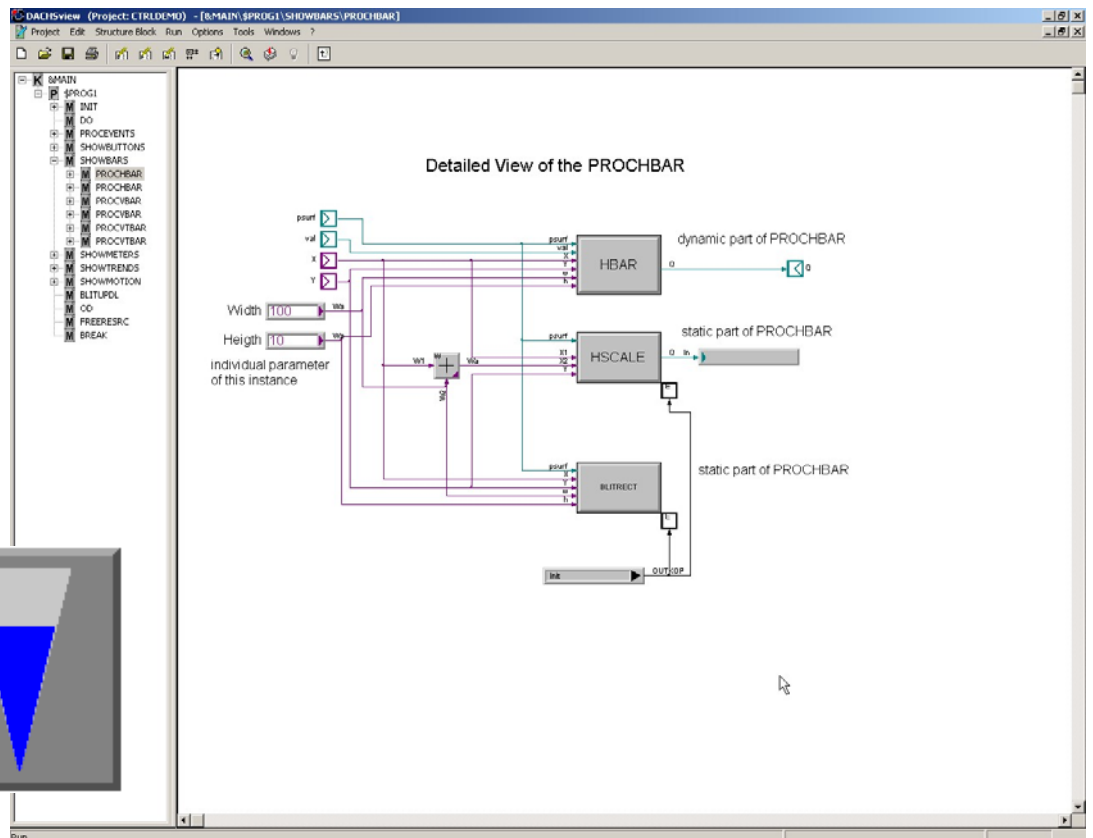
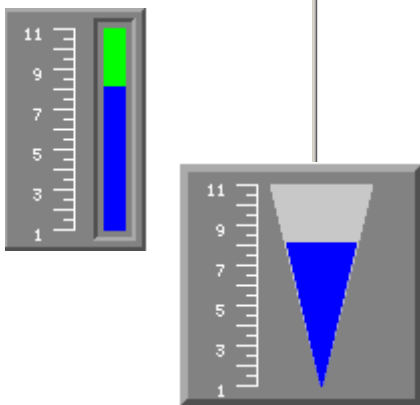


# Bar Graph Function Blocks of a Demo Application

composite function blocks of the bar graphs in the demo application



detailed level of a horizontal bar graph and its individual instance parameters

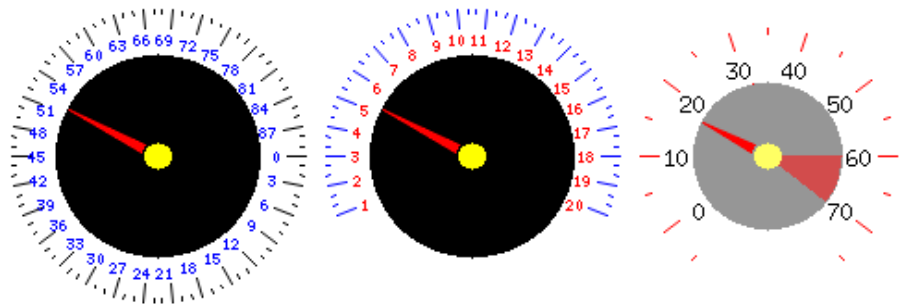
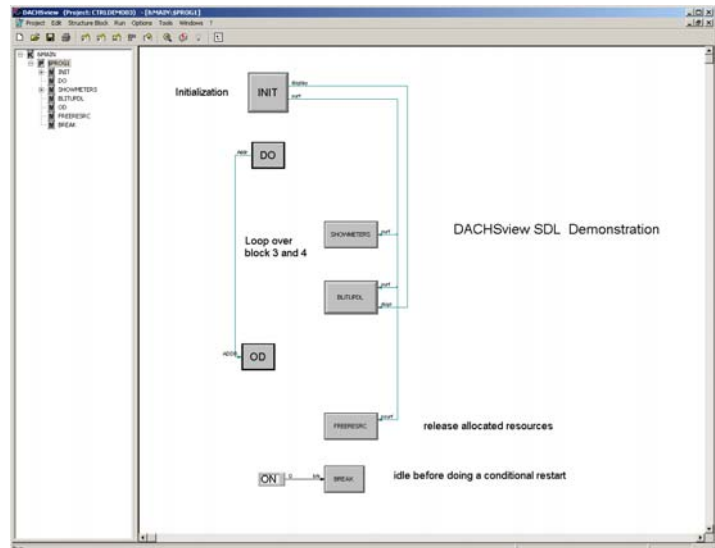


# Program Structure of Meter Instruments

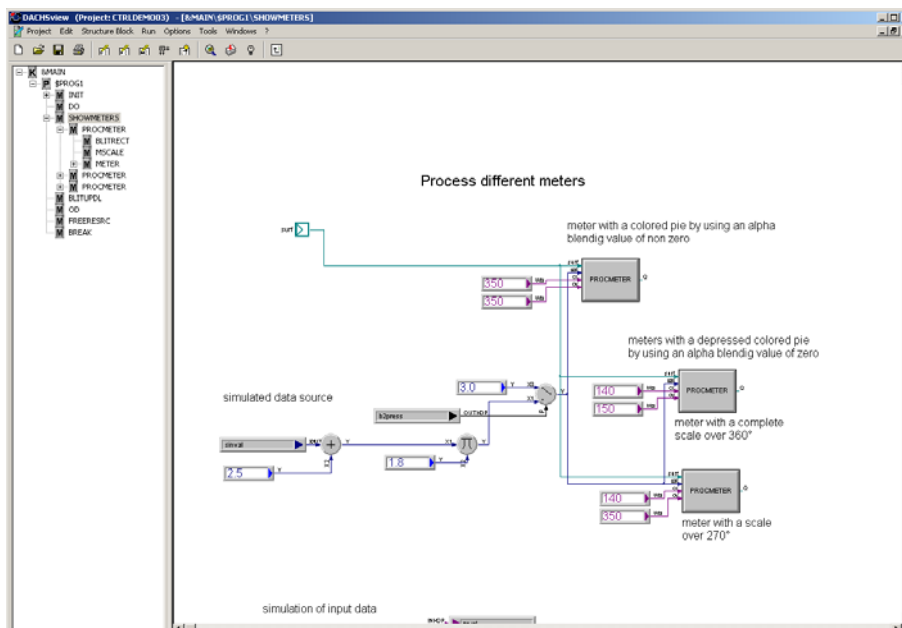
## 1. Highest Program Level with the lowest level of details

Structure of the function block diagram for displaying needle meters

After executing the initialization block **INIT** the blocks **SHOWMETERS** and **BLITUPDL** are processed in the cycle of the defined frame rate.



## 2. SHOWMETERS



internal structure of this function block (FB):

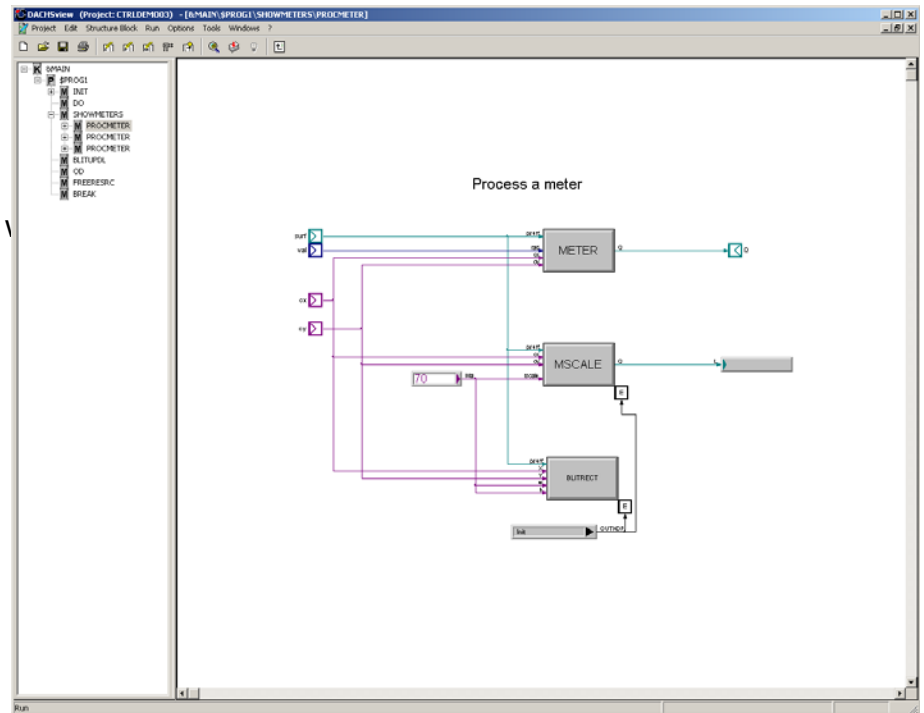
SHOWMETERS contains three instances of the block **PROC METER** for 3 different needle meters.

The positions of these three meters are defined by their individual instance data

### 3. PROC METER

internal structure of this FB

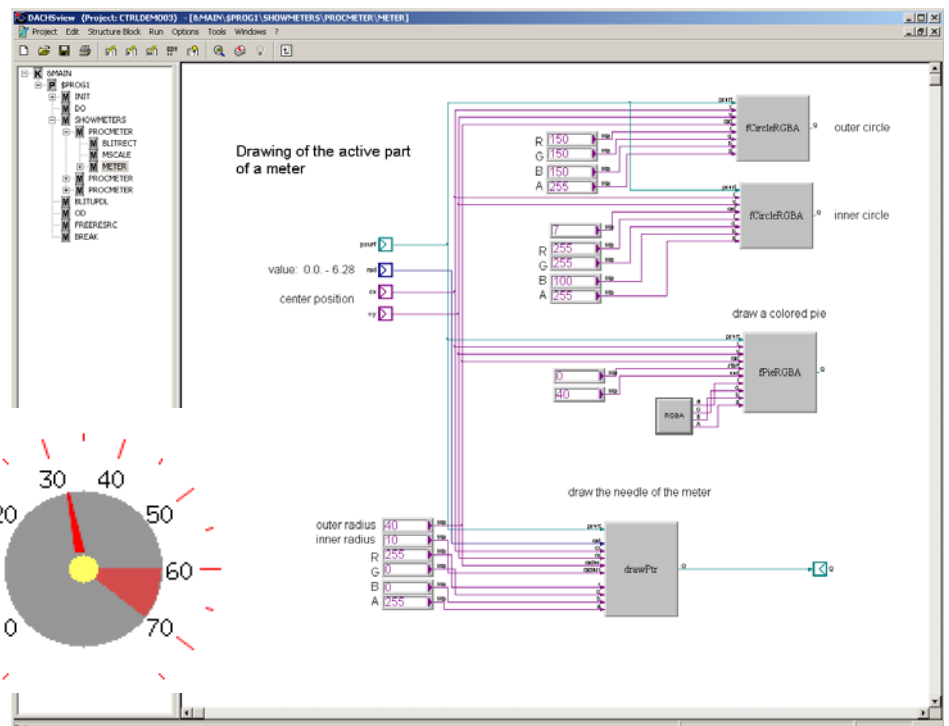
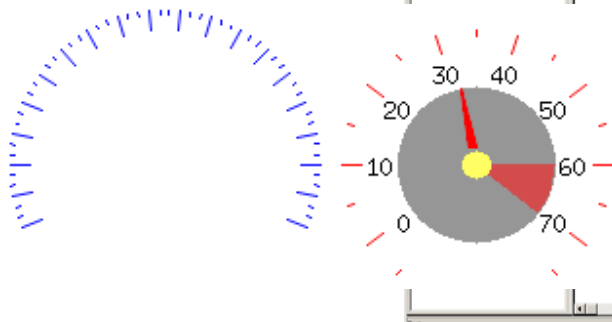
The picture elements will be processed conditionally in order to draw the instrument



- Block **METER** - draws the needle together with the background graphic
- Block **MSCALE** - draws a statically scale with its scale values
- Block **BLITRECT** - responsible for the definition of the data for optimized displaying of the instrument.

### 4. METER

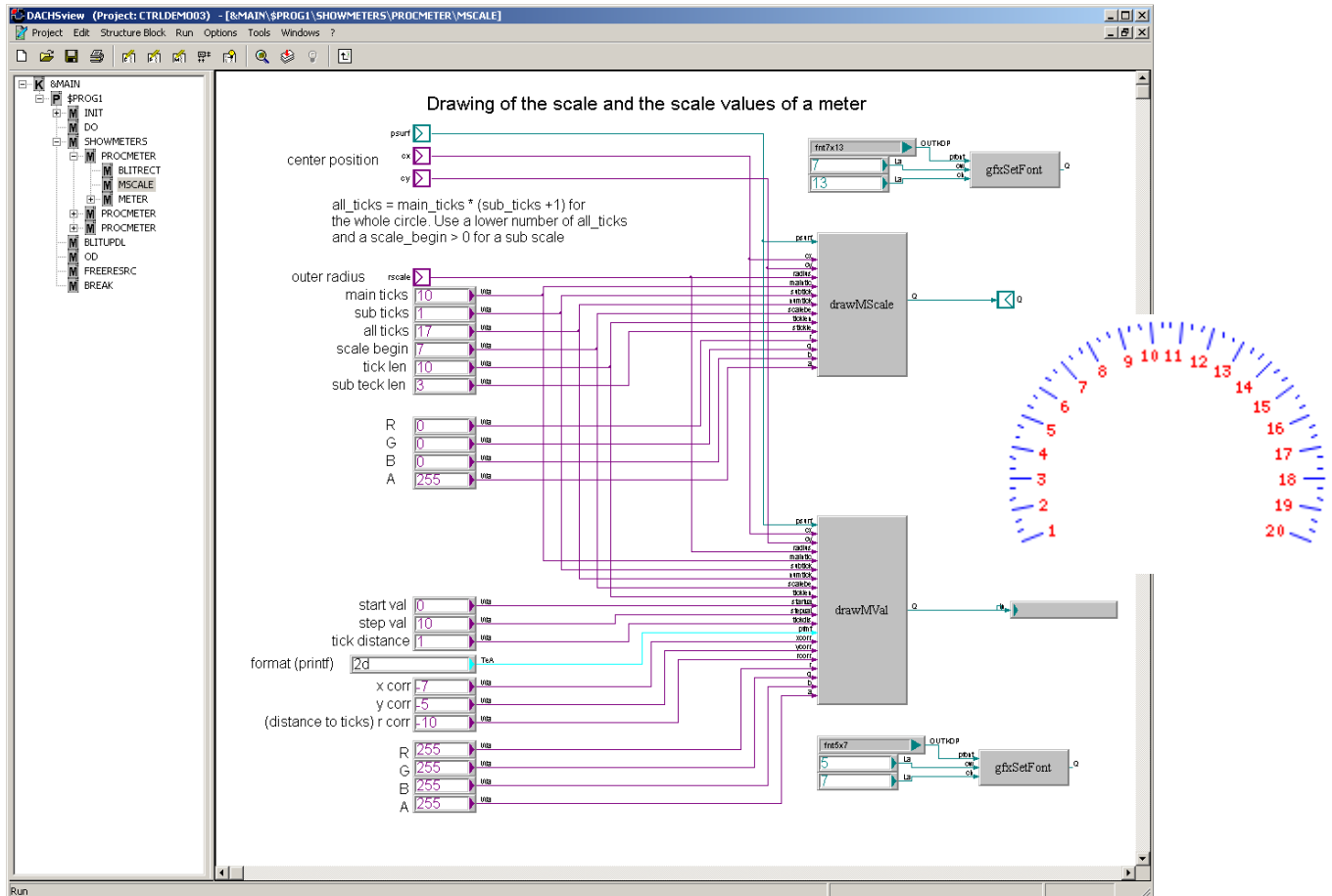
This FB is mainly assembled out of library blocks:



The geometry and colors become defined by individually instance parameters. After displaying the background graphics the needle of the meter will be drawn on its current position by the block **drawPtr**. Its color is defined with RGBA-values and can even be defined in dependency of the position of the needle.

## 5. SCALE

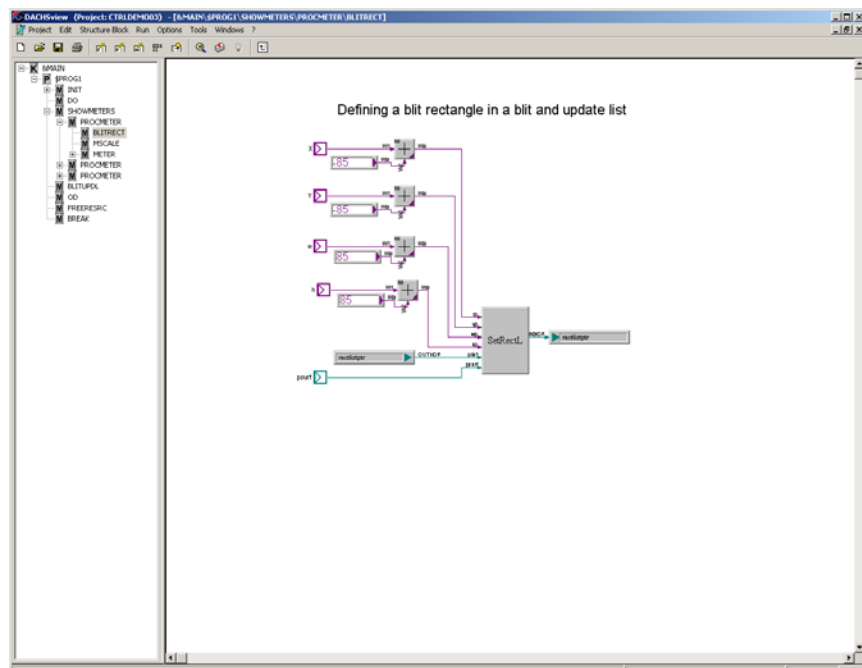
This FB draws the scale and its scale values



The geometry as well as the colors of the scale and its values can be defined by individual instance parameters.

## 6. BLITRECT

this FB defines the 'blit'- and 'update'-rectangle of the meter and adds it into the list of the rectangles of the picture which are to draw.



**DACHSview-SDL** is an **innovative IDE-tool** which is **based on a graphical function block language** and covers all aspects of today's industrial automation. DACHSview-SDL applications are data flow- and function block oriented and are developed **within an uniform programming environment**.

### DACHSview-SDL comes with Function Block Libraries for:

- **2D graphics:** calls of the libraries SDL and SDL-Gfx, SDL\_ttf, TinySDGL and Agar as pre-defined function blocks (FBs) incl. higher level FBs for easy handling of complex visualization tasks
- **3D graphics:** OpenGL (SDGL)
- **Standard controls:** GUI buttons, control- and input-elements, etc.
- **SQL, historical data base:** SQLite for extending DACHSview-SDL for processing mass data or for SCADA requirements
- **International fonts:** TTF support

### Options, FB Libraries for:

- **Fieldbuses** - for processing I/O-data with EtherCAT, Ethernet POWERLINK, PROFINET Modbus, PROFIBUS-DP, CAN, CANopen, INTERBUS, etc.
- **additional interfaces** like OPC and TCP/IP-based middleware (SDL-net, PVM e.g.)

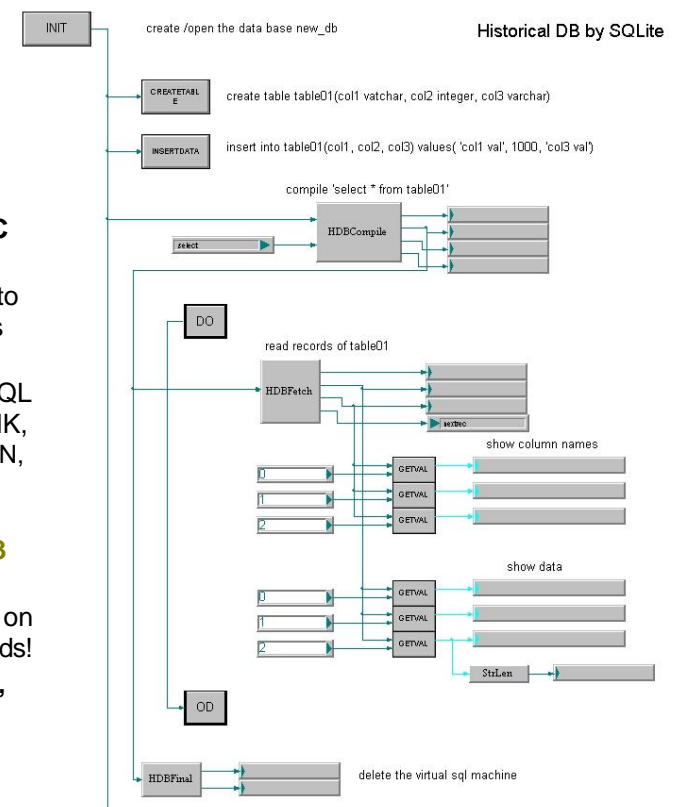
All these FBs are provided by the libraries of the MS-Windows based Workbench for the programming tool DACHSview-SDL, and are included in a specific DACHSview-SDL Target for QNX Neutrino RTOS 6.x.

**User-defined function blocks can be built, too.**

**C-applications can even be integrated by message-passing.**

### Important Features of DACHSview-SDL:

- **by mouse-click definition of complex and re-usable GUI-components**, based on pre-defined and user-defined function blocks.
- **access from a single pixel up to a complex widget or virtual instrument** by graphical libraries
- **minimized complexity** by abstraction of details of interfaces in high-level-function blocks
- **higher efficiency in development and shorter time to market** because of developers have no longer to care for details of the system
- **hierarchical definitions of composite, and application specific function blocks**
- **instantiation** of library- and composite function blocks.
- **easy recycling of function blocks.**
- **minimized coding errors** by code-reuse
- **support of semaphores**
- **freely parameterizable function blocks** for bar graphs, meters, trend graphs.
- **GUI buttons, control- and input-elements** as provided by library AGAR.
- **visualization functions combined with Soft-PLC tasks**
- **development of control applications** by access to hardware interfaces, fieldbus systems and libraries for graphical elements.
- **seamless data flow between subsystems** like SQL data base or fieldbus-I/Os like Ethernet POWERLINK, EtherCAT, PROFINET, Modbus, PROFIBUS-DP, CAN, CANopen, INTERBUS, etc.
- **processing in threads: system threads** and internal managed **threads according IEC 61131-3**
- **blocking function blocks in system threads (IEC 61499-processing)**, therefore response time on real time events in microseconds and not in milliseconds!
- **access to shared-memory, I/O ports, interrupts, and fast message-passing**
- **development of device drivers** by hierarchical defined function blocks



**STEINHOFF Automation & Fieldbus-Systems**  
was founded in 1992 by Armin Steinhoff.

**STEINHOFF is working with QNX since 18 years and is specialized in:**

- Real-Time Software for
  - QNX 4 RTOS and QNX Neutrino RTOS v.6.x
  - Linux (e.g. RTAI) added in 2001, other RTOS' only for specific projects
  - Real-Time Linux with preemptive patch
- Fieldbus Technologies
  - PROFIBUS, CAN/CANopen, PROFINET, Ethernet POWERLINK, EtherCAT, Modbus, INTERBUS, LonTalk, ASi, ...
- Technology Transfer for OEMs
- IEC 61131-3 and IEC 61499 based Systems
- GUI
- System Integration for distributed and embedded PC-based OPEN CONTROL SYSTEMS.
- Consulting, Custom Engineering

**Standard and OEM products for Industrial Automation and Embedded Systems**

**Product Suite DACHS<sup>®</sup>, Fieldbus Packages including:**

- Starter Kits
- Drivers (resource managers) , Target-Systems
- C-APIs, C-Talk, Python, and PVM APIs, Functionblock Libraries for DACHSview
- Configuration Tools
- CANopen Master Stack
- OPC

**Visual Programming**

- IEC 61131-3 or IEC 61499 based programming tools
- Soft PLCs
- **DACHSview-SDL** (event driven) for 2D and 3D GUI & Realtime applications with integrated SQL data base
- Functionblock Libraries for QNX6, SQL data base, 2D/3D graphics, standard control elements and complex widgets, TTF support, fieldbuses, etc.



Fieldbus technologies for Open Control Architecture together with Real-Time Systems, C-, Python or visual Programming (IEC 61131-3 or IEC 61499 based), HMI, and industrial hard- and software standards meet the requirements of industrial and embedded automation from today and tomorrow. This open system is cost-effective because of the minimization of costs of ownership. Advantages of the automation system DACHS<sup>®</sup> are high speed and shortest cycle times less than 1ms which allow the realization of applications with hard real-time requirements. Our product suite DACHS<sup>®</sup> supports well tested Fieldbus and I/O boards and comes with a **single unified data interface for all user level applications: the Process Image.**

**DACHS**

Distributed Automation Control & Host System based on industrial standards, QNX or other RTOS, Fieldbuses and new software technologies.

**DACHS<sup>®</sup> real-time solutions are used worldwide for various kinds of reliable industrial automation applications, embedded systems, as well as in mission critical control systems:**

- |                                    |   |
|------------------------------------|---|
| Automotive Control Systems         | Material Flow Control                                 |
| Automatic Storage Management       | Oil & Gaz SCADA Applications                          |
| Autonomous Vehicle Control Systems | Power Plant Systems (water, wind, ...)                |
| Building Automation Systems        | Power Generating Systems                              |
| Energy Supply Optimization         | Process Control Systems, Process Data Acquisition     |
| Funpark Control System             | Transportation Systems, Traffic Control Systems       |
| Environmental Technique Control    | Train Station Management                              |
| Factory Automation                 | Test Stands and Test Equipment Control                |
| Healthcare Applications            | ... and many other and complex graphical applications |
| Machine Control Systems            |   |

DACHS<sup>®</sup> is installed with and without GUI or SCADA systems e.g. in driver display systems, power plants, power station control, oil platforms, sorter devices, robots, fork lifts, factories, airports, automated parking houses, cars, trains, ships, airplanes, fun park plant (EPCOT, Disney), medical devices, etc.

QNX is a registered trademark of QNX Software Systems Ltd. | DACHS is a registered trademark of Steinhoff A.  
All other trademarks belong to their respective owners. 03/2010



**DACHS**  
...the better Idea!

