CODESYS
Features and Improvements
CODESYS V3.5 SP16
Overview

- Device User Management
- Integrated web browser
- Package Manager
- New libraries
- New language features
- Smart Coding and Usability
- Memory consumption in CODESYS
- CODESYS Professional Developer Edition
  - CODESYS Static Analysis: Major improvements
  - CODESYS Profiler
  - CODESYS SVN
  - CODESYS Test Manager
  - CODESYS UML
Device User Management

- Secure, encrypted transmission of user names and passwords
- New services: asymmetrical procedure for the transmission of passwords at login
- Forwarding of client type to the controller (e.g. CODESYS Development System or CODESYS Automation Server)
- Now only possible online: handling of users, passwords, groups
- Export/import: Still possible - password required
- User Interface: Almost unchanged
- Workflows: Slightly different

Benefit for CODESYS users:
Secured passwords - even without encrypted communication
Integrated web browser: Chromium Embedded Framework (CEF)

- Security update
- Used for access to CODESYS Store, library documentation, and overlay visualization
- No change to the user interface

Benefit for CODESYS users:
Reduced risk of attacks when surfing, e.g. in the CODESYS Store
Package Manager

- Faster package installation
- Installation of interface components directly through a package
- New hooks for device manufacturers for rejecting a package

Benefit for CODESYS users:
Faster package installation
New library: IIoT Libraries SL

- IIoT communication / reading and writing of data structures

- Included libraries with former workstation licensing:
  - Web Client SL (Communication via http, https)
  - MQTT Client SL (Communication via MQTT)
  - Mail Service SL (Sending/receiving e-mails)
  - SMS Service SL (Sending/receiving SMS)
  - SNMP Service SL (Supervision of device states via SNMP agent and manager)
  - SNTP Service SL (SNTP server and client for time queries)
  - AWS IoT Core Client SL (Communication with AWS IoT Core based on MQTT)
  - Azure IoT Hub Client SL (Communication with MS Azure IoT Hub based on MQTT and https)
  - CSV Utility SL (Reading/writing of CSV files)
  - INI File Utility SL (Reading/writing of INI files)
  - JSON Utilities SL (Reading/writing of JSON files / strings)
  - XML Utility SL (Reading/writing of XML files / strings)

- Sample projects included
New library: Control Loop Library

- Closed loop controllers / filters for process control / signal processing
- Included function blocks for closed loop controllers / add-ons:
  - P-, PD-, PI-, PID-, two-point and three-point controllers
  - Function blocks for integral estimation
  - Function blocks for derivation estimation
  - Function blocks for anti-windup strategies
    (Different strategies: Prevent integrator overflow in case of a longer lasting control deviation)
  - Filter
    (Finite impulse response filters, infinite impulse response filters, second order section filters)
  - PWM generator
  - Abstract basic function blocks for the creation of individual components
- Sample projects included
- Download and usage for free – no licensing necessary
New library: Net Base Services 2

- Function blocks for the communication via TCP/IP and UDP
- Included in the setup of the CODESYS Development System
- TLS support for secure TCP/IP communication
- Pure IEC 61131-3 implementation ➔ Portable to any CODESYS platform)
- Support of multitasking / multicore
- Usage of IEC Tasks for asynchronous calls
- Usage of optional libraries (e.g. CmpTls, CmpCrypto) ➔ Support of many different runtime systems
- Usage in POUs with graphical programming languages and "synchronously" via corresponding methods (transmission/reception in one cycle)
- Sample projects in the CODESYS Store
New language features: Optional Inputs of methods and functions

- Known from other programming languages
- CODESYS-specific extension of the IEC 61131-3
- Inputs with initial value: No need to pass variables in the call
- Consistently integrated in the user interface

FUNCTION Increment: INT
VAR_INPUT Val INT
VAR_INPUT By INT := 1;

\[ \text{Result} := \text{Increment}(\text{Val} := \text{SomeValue}, \text{By} := 2); \]
\[ \text{Result} := \text{Increment}(\text{Val} := \text{SomeValue}); \]
\[ \text{Result} := \text{Increment}(\text{SomeValue}, 3); \]
\[ \text{Result} := \text{Increment}(\text{SomeValue}); \]
\[ \text{Result} := \text{Increment}(\text{Val} := \text{SomeValue}, \text{By} := 2); \]
\[ \text{Result} := \text{Increment}(\text{Val} := \text{SomeValue}); \]
\[ \text{Result} := \text{Increment}(\text{SomeValue}, 3); \]
\[ \text{Result} := \text{Increment}(\text{SomeValue}); \]
New language features: 64 bit data types for time date

- 64 bit data types
  - LDATE
  - LDATE_AND_TIME
  - LTIME_OF_DAY
  - analog to the 32bit variants DATE, TIME_OF_DAY, DATE_AND_TIME
- Realization compliant to the IEC 61131-3
Smart Coding and Usability: New Autodeclare options in SmartTags

- Common operations already offered as SmartTag (“light bulb”), e.g. declaration of (local) variables
- Input in dialog no longer required for execution
- Input in auto declare dialog not necessarily required for execution
Smart Coding and Usability: Improved Cross Reference View

- **New column: Usage context**
  - Can be filtered
  - Find very precisely relevant places of use

### Cross Reference List

<table>
<thead>
<tr>
<th>Symbol</th>
<th>POU</th>
<th>Variable</th>
<th>Access</th>
<th>Context</th>
<th>Type</th>
<th>Location</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>error4</td>
<td>assign_lint_bit</td>
<td>error4</td>
<td>Declaration</td>
<td>.ro3, error4: BOOL = TRUE;</td>
<td>BOOL</td>
<td>Line 3 (Decl)</td>
<td>CrossRef_Demo [Device: PLC Logic: Application]</td>
</tr>
<tr>
<td>error4</td>
<td>assign_lint_bit</td>
<td>error4</td>
<td>Write</td>
<td>error4 := FALSE;</td>
<td>BOOL</td>
<td>Line 20, Column 16 (Impl)</td>
<td>CrossRef_Demo [Device: PLC Logic: Application]</td>
</tr>
<tr>
<td>error4</td>
<td>assign_lint_bit</td>
<td>error4</td>
<td>Write</td>
<td>somePtr := ADDR (myvar);</td>
<td>INT</td>
<td>Line 58, Column 16 (Impl)</td>
<td>CrossRef_Demo [Device: PLC Logic: Application]</td>
</tr>
<tr>
<td>error4</td>
<td>assign_lint_bit</td>
<td>error4</td>
<td>Write</td>
<td>somePtr := ADDR (myvar);</td>
<td>INT</td>
<td>Line 96, Column 16 (Impl)</td>
<td>CrossRef_Demo [Device: PLC Logic: Application]</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>myvar</td>
<td>CrossRef_Demo</td>
<td>myvar</td>
<td>Declaration</td>
<td>myvar : INT;</td>
<td>INT</td>
<td>Line 134, Column 16 (Impl)</td>
<td>CrossRef_Demo [Device: PLC Logic: Application]</td>
</tr>
<tr>
<td>myvar</td>
<td>CrossRef_Demo</td>
<td>myvar</td>
<td>Write</td>
<td>somePtr := ADDR (myvar);</td>
<td>INT</td>
<td>Line 172, Column 16 (Impl)</td>
<td>CrossRef_Demo [Device: PLC Logic: Application]</td>
</tr>
<tr>
<td>myvar</td>
<td>CrossRef_Demo</td>
<td>myvar</td>
<td>Write</td>
<td>somePtr := ADDR (myvar);</td>
<td>INT</td>
<td>Line 211, Column 16 (Impl)</td>
<td>CrossRef_Demo [Device: PLC Logic: Application]</td>
</tr>
</tbody>
</table>
Smart Coding and Usability: Improved Library Manager

- Better navigation in the Library Manager Editor with linked identifiers
  - Find referenced libraries or data types
  - Browse the referenced links forward and backward

- Specific deactivation of repositories possible
Smart Coding and Usability: Improved Library Manager

- Recursive reloading of libraries
- No more popup window for missing libraries
- Quickfix for missing libraries
Memory consumption in CODESYS

- **Reduction of memory consumption for the compiler:**
  - In total about 50% less RAM needed
  - 25% less RAM for the CODESYS Development System
  - May save up to hundreds of megabytes of RAM

- **Further improvements of compile / generate code:** Pending (SP17)
CODESYS Static Analysis: New configuration

- Faster configuration with better overview due to categories, filters, and sorting
CODESYS Static Analysis: Automatic execution

- Execution of Static Analysis during coding
- Quickfix of results
  - By SmartTag (“light bulb”)
  - In the message window
CODESYS Static Analysis: Clone Detection (from version 4.4.0.0)

- Check entire project for structurally identical code (ST code only)
- List view
- Comparison in separate window
- Complexity filter
- Object name filter
- Summary with clone ratio
CODESYS Static Analysis: Extract Methods

- Automatic parameter detection
- Preview in the editor
- Preview of new method

- Watch new video under https://youtu.be/Q7lf5ceYBc8
CODESYS Profiler

- **New measuring method: Sampling**
  - Minor influence on the cycle times of the application
  - Prevents Profiler to trigger watchdogs
  - Process flow (almost) not disturbed
  - Measurements much more meaningful / less falsified
  - Prerequisite: Controller supports multicore
    - Profiler task on different core than monitored task

- **Profiler Watch List**
  - Real time information on cycle times of POU's

- New video in the CODESYS Channel on YouTube shortly!
CODESYS SVN

- Conversion to core separation
- Performance improvement: Compare of working project with base project speeded up
CODESYS Test Manager: Improvements for users

- **TestDriverDevices:**
  - Set communication with IP/DNS:
    - Scan for devices:
      - In addition to Hostname / CODESYS address **now** for IP address or DNS name
      - No filtering for Target ID and Target Type or Name
  - Set communication, filtering by Target ID and target type:
    - only with host name
CODESYS Test Manager: Improvements for users

- **TestDriverTestReport:**
  - Report installed CODESYS packages, report Windows platform (32/64 bit)

- **TestManagerEditor:**
  - Generate command line template for executing a particular script
CODESYS Test Manager: Improvements for CODESYS Automation Platform users

- **TestDrvFileTransfer:**
  - Recursive directory transfer

- **TestDrvVisu:**
  - Scripts without COMPARE statement
  - Requires change in runtime system (CmpTargetVisuAutoTest) for SP16

Can now be left out.
CODESYS UML

- Refactoring UML Class Diagram
  - Changes in the UML Class Diagram (e.g. renaming): Execution of the CODESYS refactoring functionality
  - Display of the refactoring preview: Can be switched off
1. Engineering
2. Runtime
3. Visualization
4. Motion CNC Robotics
5. Fieldbus
6. Safety
Overview

- OPC UA Client
- Secure client access to runtime systems
- Optional libraries for optional components
- Runtime system documentation
- CODESYS Control for Linux ARM SL (Demo)
- CODESYS Control for WAGO Touch Panels 600 SL
- CODESYS Safety SIL2 PSP
- CODESYS Control RTE: Support of real-time IP communication
- CODESYS Control VxWorks: Support of LLVM Compiler
- Further improvements
OPC UA Client: Architecture overview
OPC UA Client: Supported features

- **Create / Delete Instance**
- **Discovery** (GetEndpoints, FindServers)
- **Session Management** (CreateSession, CloseSession, ActivateSession)
- **Browsing** (Browse, BrowseNext, Translate, RegisterNodes, UnregisterNodes)
- **Attributes** (AttributeRead, AttributeWrite)
- **Subscriptions** (CreateSubscription, DeleteSubscription, ModifySubscription, SetPublishingMode)
- **MonitoredItems** (CreateMonitoredItem, DeleteMonitoredItem, ModifyMonitoredItem, SetMonitoringMode)
Secure client access to runtime systems

- Identification of connected client types and users on the controller
  - More flexibility for future authentication methods

- **Secure authentication of users:**
  - Stronger obfuscation during password transfer
  - Stronger hash while saving the passwords on the controller
  - Improved infrastructure:
    - Easier integration of future authentication methods
    - Better separation of user administration and user configuration (for device manufacturer or OS connection)
Secure client access to runtime systems: Client types

- CODESYS Development System
- CODESYS Development System via CODESYS Automation Server
- PLCHandler
- Edge Gateway
- Data Sources
- WebVisu
- Remote Target Visu
- OPC UA Client

Query session information from IEC code, runtime system and PlcShell command
Secure client access to runtime systems: Improved infrastructure for user management in the runtime system
Secure client access to runtime systems: Future usage (outlook)

- Simultaneous login of CODESYS / CODESYS Automation Server into an IEC application:
  - Deny access
  - Report client / optional user and coordinate access

- Recording of user actions:
  - To a separate log file
  - Audit Trail (audit-proof storage)
Optional libraries for optional components

- **Optional runtime system components:**
  - Resource consumption or OS requirements
    - Components cannot be installed in every runtime system

- **Typical examples of optional runtime system components:**
  - CmpOpenSSL
  - CmpOPCUA::Client + CmpOPCUA::Server
  - CmpEventMgr
  - SysSocket
  - SysSem
  - SysShm
  - CmpRedundancy
  - …

- **Optional external libraries:**
  - Component with IEC interface
Optional libraries for optional components

- **Divided into three libraries:**
  - Container library (fixed version)
  - Interface library (* newest)
  - Implementation library (is resolved by placeholders from the DevDesc)

- **Library placeholder in the Device Description:**
  - Created by the DeliveryManager: If the corresponding component is installed in the runtime system
  - No more misconfiguration!
  - Conditional compilation in user code against optional library!
Runtime system documentation for device manufacturers

- **Runtime system online help significantly improved**
  - New: table of contents with interactive unfolding/folding of chapters and tile-based entry page
  - Consistent full-text search: Documentation in HTML wherever possible
  - New main chapters: "Runtime Variants" and "Runtime Adaptation"
  - Significant improvement of the reference documentation:
    - New chapters for compiler switches, defines, settings, events, tasks, logger entries, etc.
    - Significantly improved documentation of features (generated links to components, target settings, necessary features, etc.)
  - Workshop presentation: Now also part of the online help

- **For device manufacturers:**
  Available in the customer portal
  (most recent released version)
CODESYS Control for Linux ARM SL (Demo)

- Integrated Features
  - MultiCore
  - CODESYS WebVisu
  - Security Manager
  - OPC UA Server
  - SocketCAN
  - Common Fieldbus Systems, e.g. EtherCAT, EtherNet/IP, PROFINET

- Generic SoftPLC for ARM devices with Linux (Debian and derivatives)
CODESYS Control for WAGO Touch Panels 600 SL

- **HW-Support: WAGO Touch Panel Serie**
  - Webpanel: 10.1“ / 7“ / 5.7“ / 4.3“
  - Ethernet / USB / CAN / Serial / (DIO)
  - RT-Preempt Linux
  - 2 Core ARM Cortex A9
  - 2GB RAM / 60MB Flash + SD

- **CODESYS Features**
  - Single License per device (per CodeMeter Runtime: Dongle / SoftContainer)
  - CODESYS MultiCore
  - CODESYS WebVisu
  - Security Manager
  - CODESYS OPC UA Server
  - SocketCAN
  - Common Fieldbus Systems, e.g. EtherCAT, EtherNet/IP, PROFINET
  - Optional: Support for Motion CNC Robotics
CODESYS Safety SIL2 PSP

- **PSP**: Platform Support Package for manufacturers of safety controllers
- **Pre-certified SIL2 runtime system toolkit for a specific platform**
  - Already released: TI RM48 and TMS570
  - New: Infineon AURIX TC29X
- **Contains**
  - System adaptation to the platform
  - Error analysis of the modules used
  - Pre-certified parts with already fulfilled requirements from CODESYS Safety SIL2
  - Simplified interface (RTSSIL2PSP.Itf)
  - Tool validations of the tools used
  - Integration tests
- **Reduced development and certification efforts for device manufacturers**
  - Dimension for savings: ≥4 man-years
CODESYS Safety SIL2 PSP: Architecture / Implementation for device manufacturers

Device manufacturer

System extension

- Init (Core, RAM, ...)
- Interrupt / Trap entries
- ...

CODESYS

CODESYS Safety SIL2 Runtime (PSP)

- Flash driver
- Memory management
- Debugging
- CPU handling
- Exception handling
- MPU handling
- System timer
- ...

MCU selftests

- RAM tests
- Timer tests
- ...

Device manufacturer

Runtime extension

- CAN
- SPI
- Ethernet
- Local I/Os
- ...

RUNTIME
CODESYS Control RTE: Support of real-time IP communication

Basic principle

CODESYS Control RTE

Windows IP  IP-based communication  LwIP

Network adapter driver

Network
CODESYS Control RTE: Support of real-time IP communication

- **LwIP: Lightweight IP:**
  - Widely used open source TCP/IP stack developed for embedded systems

- **Main focus: Reduction of resource consumption with the simultaneous presence of a fully functional TCP stack**

- **Real-time capability: Based on use of CODESYS network adapter drivers (Intel and Realtek) to access the network directly**

- **Benefits**
  - Real-time capable IP-based communication (without using the Windows IP driver)
    - e.g. for network variables, EtherNet/IP, Modbus
    - Even possible: Usage simultaneously to fieldbus systems e.g. EtherCAT or PROFINET
**CODESYS Control RTE: Support of real-time IP communication**

**Performance measurements**

<table>
<thead>
<tr>
<th></th>
<th>V3.5 SP15</th>
<th>V3. SP16 (without LwIP)</th>
<th>V3.5 SP16 (with LwIP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download, 3.1 MB</td>
<td>6s</td>
<td>6s</td>
<td>1s</td>
</tr>
<tr>
<td>File Transfer, 10 MB</td>
<td>15s</td>
<td>15s</td>
<td>3s</td>
</tr>
<tr>
<td>OPC UA, Browse 1000 vars</td>
<td>35s</td>
<td>4s*</td>
<td>1s</td>
</tr>
</tbody>
</table>

*General revision of the memory management of CODESYS Control RTE (heap management) for V3.5 SP16
CODESYS Control VxWorks: Support of LLVM Compiler

- **LLVM: Low Level Virtual Machine**
- **Compiler backend**
  - Collection of compiler and toolchain technologies (C, C++, Objective-C and OpenCL) with a comprehensive translation concept
- **Compiler frontend: Clang**
- **Benefits of LLVM/Clang**
  - Required for the use of VxWorks 7
  - Uniform parser for C-based languages
  - Compatible with GCC compiler
  - High performance: Faster translation of the sources with less memory consumption
  - Often smaller executable programs
Further improvements

- **CmpSettings**: Separation of read-only and writable cfg-file
- **CmpBlkDrvUdp**: Blacklist / Whitelist for Ethernet adapter
- **CmpCrypto**: Support for asymmetric crypto operations
- **OPC UA Client**: Debugging of callback functions in IEC code (infrastructure)
- **OPC UA Server**: Debugging of callback functions in IEC code (infrastructure)
- **OPC UA Server**: Support for multitasking / multicore
- **OPC UA Server**: Access to comments of variables now possible
- **OPCServer/PLCHandler - Interfaces Gateway and ARTI**: Support of V2 PLC password
AGENDA

1. Engineering
2. Runtime
3. Visualization
4. Motion CNC Robotics
5. Fieldbus
6. Safety
Overview

- WebVisu overlay
- Distributed alarm management for CODESYS HMI
- Alarm table
- Remote TargetVisu
- Keyboard operation
WebVisu overlay

- Now released
- New drawing logic
- Elements as separated objects in the client
- Each element draws independently
- Arbitrary overlaps with native controls possible
WebVisu overlay: New features

- Dynamic movement for all elements
  - Inner rotation for all elements (also for group, frame, or native control)
- Time-controlled animations independent of VISU-TASK cycle
  - Smooth moving of menus
  - Smooth transitions when showing/hiding dialogs
  - Update of animated images such as GIFs/SVGs
  - No load on the controller / PLC
Distributed alarm management for CODESYS HMI

- Individual alarm management for each controller
- HMI: Monitoring of remote alarms via data source connection
- Central display of active and historical alarms from different controllers in the HMI alarm table
- Monitoring of connected controllers / PLCs possible
- Offline configuration possible
Distributed alarm management for CODESYS HMI

- Data sources: Provide information on remote alarm groups and classes
- New object "Remote Alarms": User configuration: Which remote alarm groups and classes are also to be monitored

- Successfully tested setup:
  - Limit: 32 controllers / PLCs with 2,000 alarms each
  - Message burst of 1,000 alarms from different controllers: Displayed centrally in the alarm table of the HMI within ~ 1-2 seconds
Alarm table

- Filtering of alarms over time range via IEC variables (data type: DATE_AND_TIME / DT)
- Activation / deactivation of the filtering: via "Filter type"
  ➔ Finding alarms faster

<table>
<thead>
<tr>
<th>Alarm configuration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm groups</td>
<td>All</td>
</tr>
<tr>
<td>Priority from</td>
<td>0</td>
</tr>
<tr>
<td>Priority to</td>
<td>255</td>
</tr>
<tr>
<td>Alarm classes</td>
<td>All</td>
</tr>
</tbody>
</table>

- Filter by latch 1
  - Filter variable: HMI_PRG.@Filter
  - Filter type: HMI_PRG.eFiltType

- Filter by time range
  - Filter variable, from: HMI_PRG.dtFrom
  - Filter variable, to: HMI_PRG.dtTo
  - Filter type: HMI_PRG.eFiltTarget
Remote TargetVisu: Extensions

- **Encrypted communication**
  - Activation via entry in `targetvisuremote.cfg`
    
    ```
    [CmpVisuHandlerRemote]
    Communication.EncryptionMode=1
    ```

- Installation of the certificate by the following call:
  
  "D:\Presentation\RemoteTargetVisu.exe" --installTrustedCert=D:\Presentation\cert\export\abc.cer"

- **Network scan for Linux Remote TargetVisu**
  - Already available under Windows/WinCE
  - Included in every delivery of the Linux Remote TargetVisu
Keyboard operation: Freely configurable tab order

- **Tab sequence independent of element order**
  - Elements removable from tab order
  - Tab order relevant for standard keyboard operation

<table>
<thead>
<tr>
<th>Type</th>
<th>X</th>
<th>Y</th>
<th>Width</th>
<th>Height</th>
<th>Id</th>
<th>Name</th>
<th>Acces...</th>
<th>Tab Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>#0 Button</td>
<td>10</td>
<td>720</td>
<td>101</td>
<td>30</td>
<td>30</td>
<td>GenEl...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#1 Button</td>
<td>900</td>
<td>720</td>
<td>101</td>
<td>30</td>
<td>30</td>
<td>GenEl...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#2 Spin Box</td>
<td>21</td>
<td>350</td>
<td>55</td>
<td>30</td>
<td>0</td>
<td>GenEl...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#3 Spin Box</td>
<td>450</td>
<td>510</td>
<td>55</td>
<td>30</td>
<td>89</td>
<td>GenEl...</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>#4 Group Box</td>
<td>493</td>
<td>60</td>
<td>531</td>
<td>424</td>
<td>90</td>
<td>GenEl...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#5 Button</td>
<td>153</td>
<td>148</td>
<td>150</td>
<td>30</td>
<td>96</td>
<td>GenEl...</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>#1 Button</td>
<td>153</td>
<td>193</td>
<td>150</td>
<td>30</td>
<td>90</td>
<td>GenEl...</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>#3 Rectangle</td>
<td>10</td>
<td>10</td>
<td>260</td>
<td>30</td>
<td>94</td>
<td>GenEl...</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>
Overview

- Basic motion
- Multicore support for CNC and robotics
- Robotics
Basic motion: Buffer mode for PLCopen Part 1 FBs

- Single axis FBs like `MC_MoveAbsolute` as well as Master/Slave FBs like `MC_CamIn`
- New and easier possibility to command movements one after the other
- New input `BufferMode` with three modes: Aborting, **Buffered**, Blending.
Basic motion: Buffer mode for PLCopen Part 1 FBs

Blending from velocity 100 mm/s to velocity 200 mm/s with mode BlendingHigh at position 100 mm
Multicore support for CNC and Robotics

- **CNC:**
  - Reading of G code and preprocessing of the path (corner rounding, tool radius correction, ...) on separated core possible

- **Robotics:**
  - Trajectory planning on separate cores possible
  - Flexible distribution of robots among cores possible

- **Significant performance advantages, especially on common Arm/Linux-based PLCs** (not so powerful, but often with 2 cores)
Robotics: Path fidelity during jogging

- New block SMC_GroupJog2
- Easier to use (no virtual axes required)
- Moves the robot "true to track" in any case, i.e. on a straight line
Robotics: Continuing the path

- Former situation:
  After error of the axis group or after MC_GroupHalt/MC_GroupStop:
  all commanded movements removed from axis group
- New: Optional saving of the status of the axis group, later continuation
- Based on MC_GroupContinue
Robotics: Improved movements

- Improved pick & place movements

Previous trajectory for a pick & place application
Position (blue), velocity (green), acceleration (red), jerk (violet)
Robotics: Improved movements

- Improved pick & place movements
Robotics: Improved movements

- Improved pick & place movements
- Improved tracking (synchronization with conveyor belt or rotary table)

Improved trajectory, compare the red acceleration curve.
Robotics: Improvements MC_GroupHalt/MC_GroupStop

- Braking ramp starts immediately (previously up to 50 ms delay)
- Set ramp parameters taken into account (previously the ramp of the path was used)
Robotics: Simplified tool change

- New FB for setting a tool
- Terms clarified: tool vs. orientation kinematics
AGENDA

1. Engineering
2. Runtime
3. Visualization
4. Motion CNC Robotics
5. Fieldbus
6. Safety
Overview

- **Diagnosis**
- **Updates of**
  - CODESYS EtherCAT
  - CODESYS PROFINET
  - CODESYS ETHERNET/IP Scanner/Adapter
  - CODESYS CAN
  - CODESYS Modbus
- **Further improvements**
Diagnosis: Recap of SP15

- Error indicator, error cleared and logger page for fieldbus devices
## Diagnosis: Status of Implementation SP16

<table>
<thead>
<tr>
<th>Fieldbus</th>
<th>Logger page</th>
<th>Diagnostic cleared</th>
</tr>
</thead>
<tbody>
<tr>
<td>EtherCAT</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ProfiNet Controller (IEC)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ProfiNet Controller (NetX)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ProfiNet Device (IEC)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ProfiNet Device (NetX)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Profibus</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet/IP Scanner (IEC)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ethernet/IP Scanner (NetX)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ethernet/IP Adapter</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CANopen Master</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CANopen Slave</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>J1939</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Sercos</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modbus TCP/Serial</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Green color: Implemented with SP16
EtherCAT

- Standalone API allows for programmatic fieldbus configuration
  ➔ No master in the device tree necessary
  ➔ Examples available

- Support of Beckhoff AX8xxx drives

- Diagnosis
  - Status page: Time measurements for SendEthFrame and GetEthFrame
    - Recv Time (Avg): LTIME#30us201ns Average Time for receiving Ethernet Frames per BusCycle
    - Recv Time (Max): LTIME#2ms235us600ns Max Time for receiving Ethernet Frames per BusCycle
    - Send Time (Avg): LTIME#45us171ns Average Time for sending Ethernet Frames per BusCycle
    - Send Time (Max): LTIME#335us480ns Max Time for sending Ethernet Frames per BusCycle
### PROFINET controller

- **Reconfigure extensions**
  - Changing IP and station name
  - Reading and changing module settings
- **Less jitter: Reduced semaphore blocking times**
- **MRP Configurator: Configuration of redundant Ethernet ring**

<table>
<thead>
<tr>
<th>PN10 I/O Mapping</th>
<th>PN10 I/O Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Station Name</td>
</tr>
<tr>
<td>1</td>
<td>scalce</td>
</tr>
<tr>
<td>2</td>
<td>wago1</td>
</tr>
<tr>
<td>3</td>
<td>ET200</td>
</tr>
</tbody>
</table>

- **Configurator: Unicode station names**
- **Diagnosis**
  - Further diagnostic outputs for controller FB
PROFINET Device

- New FB: DeviceAR-FB
  - Control all phases of the connection establishment and parameterization
EtherNet/IP: General information

- **Conformance Test**
  - First customer with CT16 certification for scanner and adapter (possible from 3.5 SP15 P1)
  - As of SP16: CT17 protocol test on Linux successful

- **Significant performance improvement on CODESYS Control RTE with lwIP**
  - Average bus cycle time for scanner without slaves, sample measurements:
    - SP15: 4ms
    - SP16: 160us
  - 25 times faster!
  - EtherNet/IP test project with test stand, sample measurements:
    - SP15: 16ms
    - SP16: 300us
  - > 50 times faster!
EtherNet/IP

- **Scanner:**
  - Icons: extracted from EDS and displayed

- **Adapter:**
  - Address Conflict Detection (ACD)

- Support of Connection Tags

- New service ‘Unconnected Send’
Various diagnostic enhancements:
- Errors in the configuration phase: Displayed as diagnostic messages and in the logger
- EMCY and abortion codes: Displayed textually
- Runtime errors: Displayed textually in the logger
- Diagnostic display: now also with SIL2
### J1939

- 64 bit support
- Additional log message for address claiming

<table>
<thead>
<tr>
<th>Severity</th>
<th>Time Stamp</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.594</td>
<td>NetID 1, ECU NAME 16#10800A000000 (current address: 0): ECU Address defended against ECU with NAME 16#8000010800A000000.</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.574</td>
<td>NetID 1, ECU NAME 16#10800A000000 (current address: 0): Operational</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.574</td>
<td>NetID 1, ECU NAME 16#10800A000000 (current address: 254): Address claiming in progress...</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.574</td>
<td>NetID 1, ECU NAME 16#10800A000000 (current address: 254): ECU is initializing...</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:18.169</td>
<td>NetID 0, ECU NAME 16#8000010800A00000 (current address: 128): Operational</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.594</td>
<td>NetID 0, ECU NAME 16#5000010800A00000 (current address: 254): Address claiming in progress...</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.594</td>
<td>NetID 0, ECU NAME 16#6000010800A00000 (current address: 254): Address claiming lost against ECU with NAME 16#10800A000000.</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.574</td>
<td>NetID 0, ECU NAME 16#8000010800A00000 (current address: 0): Operational</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.574</td>
<td>NetID 0, ECU NAME 16#8000010800A00000 (current address: 254): Address claiming in progress...</td>
</tr>
<tr>
<td>1</td>
<td>26.03.2020 14:53:17.574</td>
<td>NetID 0, ECU NAME 16#8000010800A00000 (current address: 254): ECU is initializing...</td>
</tr>
</tbody>
</table>
Modbus

- **Master**
  - Better I/O mapping performance through own I/O copy functions (factor 3!)
  - Improved diagnostics on status page

- **Slave**
  - Modbus TCP Device Extension for RTU Gateway

- Reconfigure: Change serial port settings via application code
- Possibility of separating coil and holding registers into different data areas
Further improvements

- Device Repository: Reduced memory consumption through hard links
- Compile warning: Mapping of wrong variable types
- Update device on several devices simultaneously
Further improvements

- **Ethernet device**
  - API to change parameters via application code:
    - IP address / subnet mask / gateway address / interface name

- **BACnet**
  - Version 1.5.1 released! Feature Complete

- **PLCNext**
  - Axioline Module Scan

- **Redundancy Editor**
  - Fast switching of the online mode between active and passive PLC
Engineering
Runtime
Visualization
Motion CNC Robotics
Fieldbus
Safety
Overview

- CODESYS Safety for EtherCAT Safety Modules (4.1.0)
- CODESYS Safety Runtime Toolkit / Qualification-Kit (1.6.0)
CODESYS Safety for EtherCAT Safety Modules: Process improvement

- **Up to now:**
  - Test on special CODESYS version ➔ Report to TÜV ➔ Release
  - Use with other CODESYS versions: No evaluation / Test / Report to TÜV

- **New: Extended qualification test**

- **New: Information for users**
  - Tab "All Versions" / "Compatibility" on product page in the Store
  - Information update with every successful qualification test
  - User manual: Reference to this information
CODESYS Safety for EtherCAT Safety Modules: Check of project boundaries

- Better messages:
  - E.g. "More than 255 function blocks defined", or "No function blocks defined"

- Checked limits for specific derivatives of the EtherCAT Safety modules:
  - Max. POUs: for EK1960: 128
  - Max. I/O image: for EK1960: 24,576
  - Max. data exchange with non-safety application: Bit limit of 256 / 1024

- Testing of further limit values:
  - Unsupported slaves: Too many device-specific parameters, image too large
  - Too many devices with device-specific parameters, too many SC devices
  - Process image too large, too many data points used
  - Too much (monitorable) data in the application
CODESYS Safety for EtherCAT Safety Modules: Faster monitoring cycle

- So far:
  - From application data size > 2 kByte
    ➞ Reading of 13 * 2k blocks from controller (via EtherCAT)
  - To ensure complete reading, editor only updated every 2 s(!)

- New: Editor calculates application data size
  - Reading only required 2k blocks
  - Update rate accelerated accordingly
CODESYS Safety for EtherCAT Safety Modules: Support of old projects

- Manually created device description included for compatible safety modules controllers including onboard IO
  - Some improvements require a new device description:
    - E.g. standard EtherCAT diagnostic tab, new driver version for faster monitoring, improved description of device-specific parameters of onboard IOs

- So far:
  - Package update: uninstall old device descriptions
    - Uninstalled device descriptions missing in old projects
    - Manually perform Update Device command for the controllers
    - Copy configuration and manual mapping of onboard IOs to new version (no update device on logical devices)

- New: Old device descriptions installed additionally ➔ Old projects still work
- New: Old projects updateable via Update Environment (incl. onboard IO)
CODESYS Safety Runtime Toolkit / Qualification-Kit: Boot applications

- Up to now: Login with (temporary) download + command “Create bootapplication”
- New: Login download dialog extended by option to create the bootaplication
CODESYS Safety Runtime Toolkit / Qualification-Kit: Boot applications

- **Up to now:**
  Multiple download not offered for Safety-PLC

- **New: Safety-PLC also selectable for multiple download**
  1. Multiple download selection dialog
  2. Safety dialogs for each Safety PLC, all dialogs carried out in a row in advance
  3. Actual download to selected controllers without safety dialogs
  4. Safety feedback signals per Safety PLC all given in a row at the end
  5. Result overview of the multiple download