CODESYS Motion + CNC:
Your motion control, quickly developed

CODESYS Users' Conference 2014, Tim van Heijst
1. Motion task

2. Realization of the motion task

3. Basic information on CODESYS Motion + CNC

4. CODESYS SoftMotion CNC

5. Typical use cases for CODESYS Motion + CNC
Given system

- Notebook PC with Ethernet and CODESYS for programming the logic and motion control
- Test: for a start with Raspberry Pi
- Servo drive controlled via EtherCAT
Task

- Motor M1: rotate with constant motor speed
- Motor M2: rotate with scaled motor speed depending on motor M1 (gear factor adjustable)
- Programming and test: initially without physical drives
- Motor operation via visualization
- Replace virtual drives by physical drive(s)

... and all this with CODESYS SoftMotion!
1. Motion task
2. Realization of the motion task
3. Basic information on CODESYS Motion + CNC
4. CODESYS SoftMotion CNC
5. Typical use cases for CODESYS Motion + CNC
New CODESYS SoftMotion project

- Start CODESYS and create a new project for Raspberry Pi
- Establish connection to the controller
- Configure virtual axes
Prepared CODESYS SoftMotion project
Switch on an control motors: Programming

- **Switch on**
  - with PLCopen POU MC_Power

- **Control**
  - M1: with PLCopen POU MC_MoveVelocity
  - M2: with PLCopen POU MC_GearIn

- **Prepare reset (just in case)**
  - with PLCopen POU MC_Reset

- **For all axes:**
  - Setpoint value, parameter etc. with visualization
    - Merely the axes M1 and M2 have to be passed
    - Declaration of POUs as function block instances
Programmed motion application

Realization of the motion task
Visualization of the motion application

- Add visualization in the project
- Suitable visualization templates for motion POUs
  - Insert frame objects and connect to templates
  - Pass FB instance or pass axis (rotatory drive)
Realization of the motion task

Test and commissioning

- Login and download of the application
- Operation of the virtual drives via visualization
  - Switching the drives on via MC_Power
  - Default value for MC_MoveVelocity and MC_GearIn
  - In case of error: Use "Reset" POU

- Virtual axis M1 rotates
- When virtual axis M2 starts: gearing function
- Rotatory motors with blue arrow
Realization of the motion task

Test and commissioning
1. Motion task
2. Realization of the motion task
3. Basic information on CODESYS Motion + CNC
4. CODESYS SoftMotion CNC
5. Typical use cases for CODESYS Motion + CNC
Realization of MotionControl

- Separation of logic and motion control
  - Expensive, as two devices are needed.
  - Synchronization/communication needs substantial resources
  - Engineering in two different tools

- One device, but different tools:
  - Engineering in two different tools
  - Synchronization/communication needs substantial resources
Realization of MotionControl

- One device, one tool: CODESYS Motion + CNC
- Inexpensive
- Easy engineering in one tool
- No resources needed for synchronization/communication, both systems run in one process.
Why Motion + CNC in CODESYS?

- State-of-the-art controllers have power reserves.
- MotionControl or CNC on the PLC save hardware costs.
- Project planning of motion tasks in the PLC tool allows for flexible engineering.
- Well-engineered infrastructure supports design of drive configuration: from motion planning to commissioning.
Motion + CNC realized in IEC 61131-3

- IEC 61131-3 runtime system controls motion.
- State-of-the-art programming system offers flexibility for editing the motion sequence.
- Motion control modules are library functions, developed in the languages of the IEC 61131-3.
- PLC programming system, extended by editors for motion planning
- Fieldbus and I/O configuration are also used for drive configuration.
Basic information on CODESYS Motion + CNC

Structure overview

- CAM objects
- CNC objects

**Development environment**
- SM3_Basic.library (DriveInterface, PLCopen MC, additional motion FBs)
- SM3_CNC.library (CNC, kinematics)

**IEC 61131-3 user application**
- Drive Interface
  - drive specific driver (library)
  - standard drivers (CAN/ETC/...)
- CODESYS standard I/O image
  - CANopen
  - EtherCat
  - local I/Os
  - local I/Os

**Motion + CNC**

**Motion design**

**Motion execution**

**Machine connection**
Hardware requirements

- Standard requirements for the implementation of the CODESYS Control Runtime System
- Hard realtime properties of the device (small jitter)
- Suitable communication channels to drives required
  Standard fieldbuses: EtherCAT and CANopen, further on demand
- Power reserves (CPU performance, memory) of the device in accordance with the application controller, 32-bit CPU and FPU strongly recommended

Raspberry Pi is not suitable for industrial motion tasks, at least not without individual extensions!
Available control methods

- For servo drives

- For stepper drives

- For frequency converters (FC)
1. Motion task
2. Realization of the motion task
3. Basic information on CODESYS Motion + CNC
4. CODESYS SoftMotion CNC
5. Typical use cases for CODESYS Motion + CNC
- CODESYS SoftMotion:
  - For coordinated single and multi-axis movements
  - Includes the complete motion functionality without CNC

- CODESYS SoftMotion CNC:
  - Includes CODESYS SoftMotion
  - Includes many kinematics
  - Additional 3D CNC editor and CNC interpolation
Definition: Coordinated movement

- "Coordinated movement":
  - Controlled movement, executed by one or more axes
  - Movement definition relates to the POI of the machine or of the kinematic movement.

- CNC (Computer Numerical Control):
  - Subset of coordinated movement
  - Movement definition with exact definition of movement path and speed
  - Path definition per G-code (DIN 66025) is widely spread.
Programming the motion functionality with FBs
Integrated 3D CNC editor according to DIN 66025 (G-code)

- Graphical and textual input of 3D G-code
- Path design in table form
- DXF import
Kinematic transformations

- Gantry systems
- Robot
Suitable visualization elements

- For 3D CNC handling and/or online editing
  - Select and edit available CNC movements
1. Motion task
2. Realization of the motion task
3. Basic information on CODESYS Motion + CNC
4. CODESYS SoftMotion CNC
5. Typical use cases for CODESYS Motion + CNC
CNC application
- Using the CNC editor
- Jogging the axes with PLCopen POUs
- Portal kinematics with stepper motors
- Teaching function

Pick & Place application
- Using PLCopen POUs to position (pick) and belt-synchronously place an object on a moving target
- Displaying the process with visualization templates
Typical use cases

- **Tripod robot**
  - Using the CNC editor
  - Parallel kinematics (transformation)
  - Visu template for parallel kinematics
  - Configuration of EtherCAT drives

- **Labelling / CAM application**
  - Using the cam editor
  - PLCopen POUs and virtual axis as electronic line shaft (ELS)
  - Different visualization templates
  - CANopen servo drives
  - Using the Touch Probe/Latching function
Generally: Whenever...

- movements shall be flexibly adapted using PLC data.
- the motion system shall only be limited by computing capacity, the used fieldbus and memory capacity.
- motion programming shall be done driver-independently.
- motion shall be executed on different PLCs.
- logic application and motion shall be realized in one tool.
Thank you for your attention.