

PLCcore-E660 CODESYS with EtherCAT-SoftMotion

Document version: 1.0
Date: 11/3/2014 11:43:00 AM
Author: -np

1 Preface

To control drives is a common task in automation projects. Frequently the matter is to control drives from different manufacturers and different by type (steppers, AC and DC servo drives etc.) together in the same automation equipment. In addition, the drives are often spatially distributed. Thus a field bus solution for the drive control is recommended. This document describes a solution for the control of two type-different servo drives from different manufacturers on a single PLC in the form of a SYS TEC PLCcore-E660 with CODESYS Runtime System, where the drives are connected to the PLC by the field bus EtherCAT. For effort reduction in programming, the SoftMotion libraries of the CODESYS-Runtime System are used in the project.

2 Principle

2.1 Communication in EtherCAT

The drives as well as the PLC itself in the project have a field bus interface. In case of EtherCAT Ethernet interfaces are used. These interfaces drive the EtherCAT protocol (see http://www.ethercat.org/pdf/english/EtherCAT_EAP_EN.pdf).

Normally, the EtherCAT interfaces of the drives have two connection slots for an Ethernet cable. This enables to use several network topologies (dependent on the demands), such as:

- Star topology by using a hub or a switch
- Line topology – the Ethernet connection goes from the PLC to the first drive in the network, further from this drive to the next one by using the second Ethernet slot and so on
- Mixed topologies – star and line topologies together in the network

Control information from the PLC to the drives and status information from the drives to the PLC is transferred by the EtherCAT protocol. There is a strong master-slave principle in the communication control, where the drives are EtherCAT slaves and the PLC is the EtherCAT master.

All well-known drive manufacturers provide special drive libraries for the CODESYS Runtime System, which can be implemented in the automation project. But these libraries differ extremely – dependent on the drive type and the manufacturer. That's why the usage of different drive types is quite sophisticated to the user who has to establish a respective automation project. However, CODESYS provides a possibility to unify and simplify the access to all drives used in the automation project. Therefore the CODESYS SoftMotion libraries are available. Further the principle of these libraries will be explained.

2.2 SoftMotion

In SoftMotion each drive has to be considered as an axis with respective parametrizable properties (e.g. rotatory or linear with respective coordinates). It is not up to the user to care about hardware details – that's the task for the SoftMotion libraries available in CODESYS. The SoftMotion libraries themselves use the manufacturer-dependent hardware-specific libraries and hardware drivers for the particular drives.

The SoftMotion function blocks, which are available for the user, are based on the standard PLCopen (see also http://www.plcopen.org/pages/tc2_motion_control/) and simplify the handling of the different axes in the PLC program significantly.

CODESYS SoftMotion provides furthermore the possibility to program CNC equipment (see also <http://www.codesys.com/products/codesys-motion-cnc/softmotion.html>).

2.3 What's up to the user

In case of using the CODESYS SoftMotion libraries, the following tasks are up to the user to design his automation equipment with field-bus-controlled drives:

- Insert the field bus into the device tree of the PLC
- Parameterize the field bus
- Insert the drives (axes) into the device tree below the field bus
- Parameterize the axes (type, scaling, limit parameters etc.. – dependent on the drive to be used)
- Configure the tasks
- Program the drives, based on the SoftMotion function blocks in CODESYS

A simple example (see below) has been realized to explain the principle for designing complex drive systems. Therefore two type different servo drives from different manufacturers have been connected over the EtherCAT field bus to a PLC with a PLCcore-E660 CODESYS. The drives are controlled with SoftMotion.

3 Realization

3.1 Hardware

Two drives are connected to a SPS SYS TEC PLCcore-E660 CODESYS via (see Figure 1). The EtherCAT field bus has a line topology. An AC servo drive with resolver and a DC servo drive (electronic commutation) with encoder are connected to their special servo controller. The servo controllers are connected to the PLC via EtherCAT.

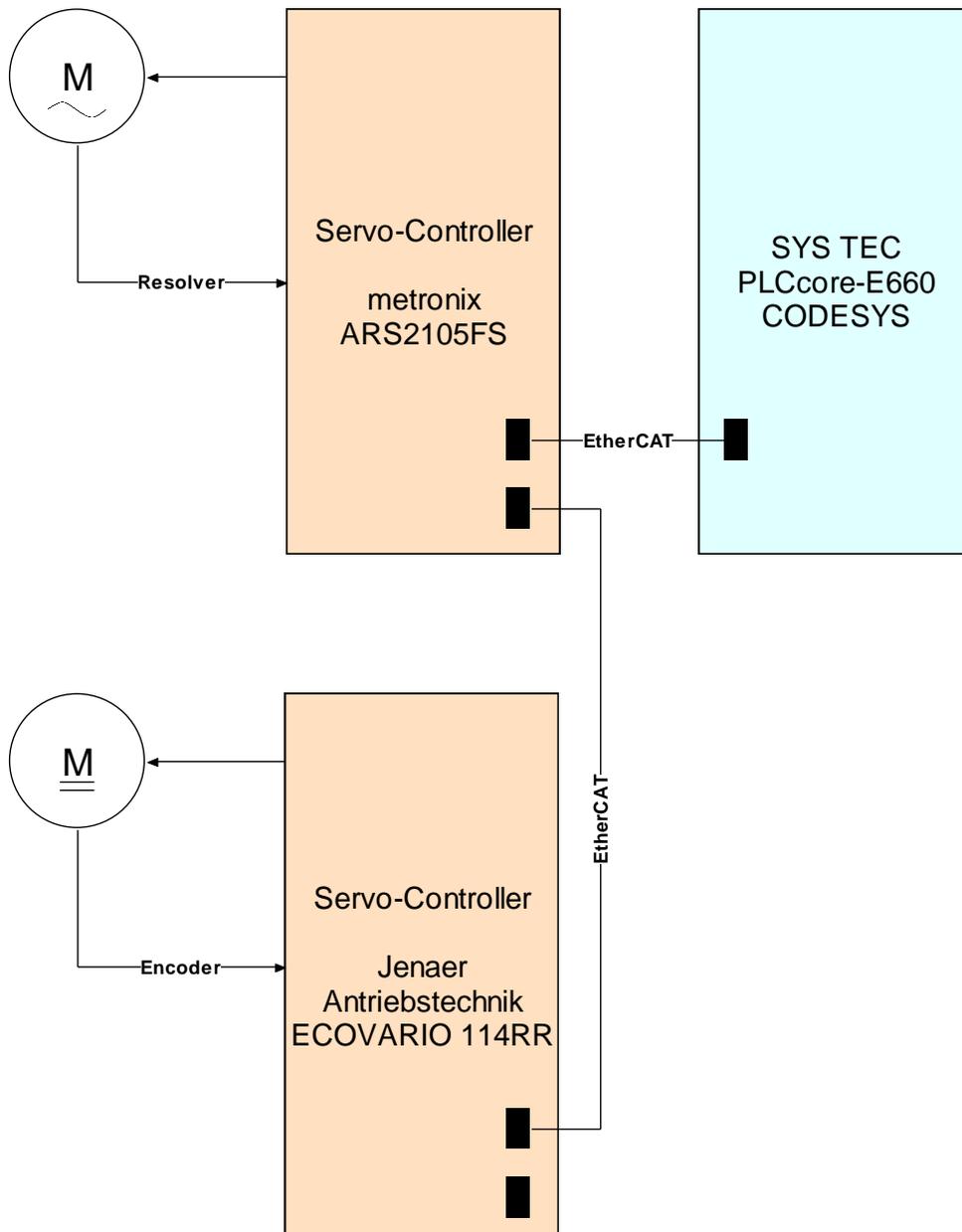


Figure 1: Hardware structure of the example

3.2 Software Project

The example project is realized in CODESYS V3. The software components and tools are:

- CODESYS V3.5 SP4 Patch 2 IDE (newer versions available as free download at <http://www.codesys.com/>)
- Runtime System for Linux on x86 processors "CODESYS Control x86 SYS TEC electronic target" statically linked SoftMotion library – a Demo-Version with 2 hours operation limit for the Runtime System and 30 minutes for the EtherCAT SoftMotion application
- Device descriptor for the PLC "CODESYS Control x86 SYS TEC electronic target" – to be installed into the device repository of the IDE
- Operating system Linux on the SYS TEC PLCcore-E660 CODESYS

As shown in Figure 2, the project structure has to fulfill the following rules:

- The device named "CODESYS Control x86 SYS TEC electronic target" has to be selected for PLC.
- The EtherCAT_Master has to be inserted in the device tree below the PLC "CODESYS Control x86 SYS TEC electronic target" – below the EtherCAT_Master the drives from metronix and Jenaer Antriebstechnik (bottom section of the figure, red-marked) have to be inserted.
- The task configuration has to be designed in the manner, that the tasks for both drive controllers are inserted below the "EtherCAT_Master", whereby the "EtherCAT_Master.EtherCAT_Task" has been inserted automatically during the project setup (middle section of the figure, also red-marked).
- The task program code files for the drive from Jenaer Antriebstechnik and the drive from metronix (EtherCAT_JAT_PRG and EtherCAT_PRG) must be inserted in the device tree below "Application" (upper section of the figure, green-marked). The user-specific code for the particular drives must be placed ST-coded inside these two programs.
- The program PLC_PRG is also automatically inserted during project setup. It contains the program code in ST for the "MainTask". In this program the user can place additional program code (e.g. control sequences and program code for the synchronization of the drives). In the given example this program is empty and doesn't contain any code.

The way, how the usage of the SoftMotion function blocks simplifies the program design, is shown in the following ST code snippets.

Code snippet for reading the current position of an axis

```
(* Read actual position of the drive*)
ReadActPosition(
    Enable := TRUE,
    Axis   := SM_Drive_ETC_JAT_ECOVARIO
);
fReadDone := ReadActPosition.Valid;
fReadErr  := ReadActPosition.Error;
wReadErrId := ReadActPosition.ErrorID;
lActPos   := ReadActPosition.Position;
```

Code snippet for a drive movement profile of an axis

```
(* Perform drive movement profile *)
...
...
CASE iStatusMove OF
  0:
    Power(Enable:=TRUE, bRegulatorOn:=TRUE, bDriveStart:=TRUE,
          Axis:=SM_Drive_ETC_JAT_ECOVARIO);
    fPowerStatus := Power.Status;
    fPowerErr    := Power.Error;
    iPowerErrID  := Power.ErrorID;
    IF fPowerStatus
    THEN
      iStatusMove := iStatusMove + 1;
    END_IF;

  1:
    p := 100;
    MoveAbsolute(Execute:=TRUE, Position:= p, Velocity:=1000,
                Acceleration:=100,
                Deceleration:=100, Axis:=SM_Drive_ETC_JAT_ECOVARIO);
    fMoveDone := MoveAbsolute.Done;
    fMoveErr  := MoveAbsolute.Error;
    iMoveErrId := MoveAbsolute.ErrorID;
    IF fMoveDone
    THEN
      MoveAbsolute(Execute:=FALSE, Axis:=SM_Drive_ETC_JAT_ECOVARIO);
      iStatusMove := iStatusMove + 1;
    END_IF;

  2:
  ...
  ...
```

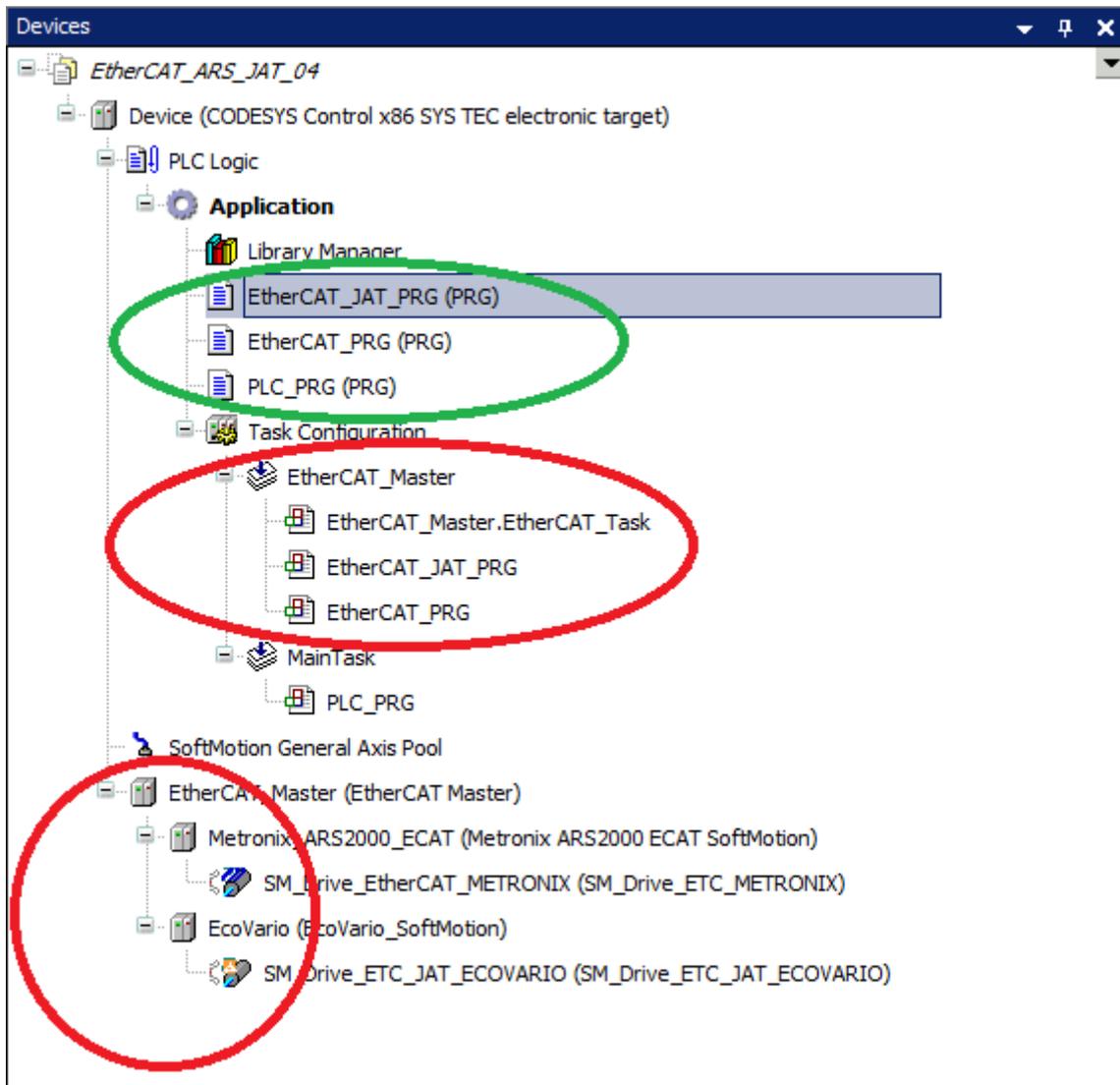


Figure 2: Device tree of the example

4 Postface

The usage of SoftMotion function blocks reduces the effort for drive programming significantly to the user. In principle, it is possible to substitute the EtherCAT connection by another field bus connection (e.g. CANopen). The user specific program parts for drive control remain the same in principle – only the necessary drive specific hardware drivers and libraries have to be substituted by the field-bus-specific ones, and the identifiers (names) for the axes change respectively.