



Industrial Control Design as

Contents:	CDPCoDeSys
Contact:	lcd@icd.no www.icd.no
Version:	CDP 2.2.0.8
Date:	23.03.2006
Document:	CDPCoDeSys Manual.sxw



CDPCoDeSys

CDPCoDeSys soft- plc users manual

The content of this document is confidential information not to be published without the consent of ICD AS.

Contents

1. GENERAL INFORMATION.....	3	2.1.5. AutoUpdateDisabled functionality.....	6
1.1. About Component Descriptions.....	3	2.2. Interface.....	7
2. CDPCODESYS.....	4	2.2.1. Messages.....	7
2.1. Inside.....	4	2.2.2. XML Attributes.....	7
2.1.1. Based on.....	4	2.2.3. XML Values.....	7
2.1.2. Description.....	4	2.2.4. Parameters.....	7
2.1.3. Usage.....	5	2.2.5. Signals.....	7
2.1.4. States.....	6	2.2.6. Example .xml.....	8
		2.3. Specification.....	9
		2.3.1. CoDeSys version.....	9
		2.3.2. CDP version.....	9
		2.3.3. Signal types.....	9
		2.3.4. Operating systems.....	10
		2.3.5. Performance.....	10

1. General Information

1.1. About Component Descriptions

The description explains how the component works and how to use it:

- Based on: Ancestor model type. Components inherit properties from their ancestors, most components will receive and process all messages supported by ancestor model types.
- Description: The purpose of the component, what it is used for.
- Usage: How to use it.
- Members: Subcomponents, alarms, signals, parameters.
- States: State machine states briefly described.
- Messages: Message interface. List of commands with parameters.
- XML: Attributes, values, example .xml for component configuration.

Please refer to the CDP System Manual for a description of the CDP system and the component model.

2. CDPCoDeSys

2.1. Inside

2.1.1. Based on

The CDPCoDeSys is based on CDPCComponent.

2.1.2. Description

CDP CoDeSys soft plc engine allows programming in IEC61131 plc languages while still having the benefits of the CDP architecture.

CDPCoDeSys runs CoDeSys soft- plc in a CDPCComponent.

CoDeSys communicates through callback wrapper methods and calls from this CDP component.

Signals declared as export to CDP in CoDeSys are created automatically.

All export signals are saved in the component .xml file.

CoDeSys signals are of ordinary CDP Signal types.

References to CoDeSys signals are contained in coDeSysIo.

Current implementation runs CoDeSys on RTOS32 and Win32 platforms.

CoDeSys export signals can only be single variables, not arrays or records. Array elements will be exported as single signals.

Current implementation limitations

- The signal update frequency is the CDPCoDeSys component fs. This can be changed run- time, for instance from the web interface. On RTOS32, 166MHz cpu, 2000 signals at 10 Hz gives 18% cpu load.
- Also due to the signal copy method, since the CDPCoDeSys runs on low priority, signal update may be interrupted by other components.

CoDeSys state signal

This signal indicates the current run state of the soft plc.

CoDeSys run state definitions:

```
#define RTS_VOID                ((unsigned char)-1)
#define RTS_RUN                 0
#define RTS_STOP_SRV           1
#define RTS_STOP_BP            2
#define RTS_RUN_STEP_IN        3
#define RTS_RUN_STEP_OVER      4
#define RTS_RUN_WATCHDOG       5
```

2.1.3. Usage

Include in binary build

The CDPCoDeSys is provided in its own dedicated library. The CoDeSys library must be included in your .exe/.rtb binary build.

The CDPCoDeSys library must be compiled using the same header files and library files as in your current CDP developer installation.

If you have source code files for CDPCoDeSys you can compile the library for your CDP version.

Instantiate

Create the CDPCoDeSys component by adding it to the application as a subcomponent, in Application.xml:

```
<Subcomponents>
  <Subcomponent Name="CDPEngine" Type="CDPEngine" src="Components\CDP\CDPEngine.xml">
</Subcomponent>
  <Subcomponent Name="Messenger" Type="Messenger" src="Components\CDP\Messenger.xml">
</Subcomponent>
  <Subcomponent Name="WebServer" Type="WebServer" src="Components\CDP\WebServer.xml">
</Subcomponent>
  <Subcomponent Name="MessengerIOserver" Type="MessengerIOserver"
src="Components\CDP\MessengerIOserver.xml"> </Subcomponent>
  <Subcomponent Name="CDPCoDeSys" Type="CDPCoDeSys" src="Components\CDP\CDPCoDeSys.xml">
</Subcomponent>
</Subcomponents>
```

Only one CoDeSys instance can run inside one CDP application.

Run CoDeSys

You can now run CoDeSys soft- plc to create and upload plc programs to the CDP application.

Please refer to the CoDeSys documentation provided with the CoDeSys soft- plc for information on how to use and program CoDeSys.

Create CoDeSys global signals

Signals declared in CoDeSys as global variables are exported as CDP signals, available to other CDP components.

To limit the number of exported signals, set the `ExportOnlyIN_OUT_Prefix` parameter to '1'. Then only variables starting with `IN_` or `OUT_` prefix will be created as CDP signals.

Signals starting with `IN_` prefix will be declared as input signals.

Signals starting with `OUT_` prefix will be declared as output signals. This is also the default behaviour when no prefix is given.

Connect to CoDeSys signals

Globals created as CDP signals can be routed and connected like normal CDP signals.

Run on Windows workstation

CDPCoDeSys supports Win32 so you can run CoDeSys soft- plc programs on your windows workstation. This will prove convenient for development and testing.

The CDPCoDeSys is distributed with a project that can be started by clicking on the `CDP.exe` in the Application- folder.

Use CoDeSys to connect to local CDP application running CDPCoDeSys and upload new or updated CoDeSys software.

All global CoDeSys signals are automatically updated when a new soft plc program is uploaded.

CoDeSys version 2.3 requires full Clean and Rebuild of soft- plc program to update signal configuration.

Run on controller

Use CoDeSys to connect to CDP controller running CDPCoDeSys and upload new or updated CoDeSys software.

All global CoDeSys signals are automatically updated when a new soft plc program is uploaded.

CoDeSys verion 2.3 requires full Clean and Rebuild of soft- plc program to update signal configuration.

2.1.4. States

The state machine controls synchronization with the CoDeSys run- time.

States

State	Description
WaitingForSymbols	Waiting for CodeSys to finish update of symbols table.
Run	SoftPLC may run, signals are updated.

2.1.5. AutoUpdateDisabled functionality

If `AutoUpdateDisabled` is on, the signal configuration will not be updated automatically if the softplc program is changed.

To update the signal configuration it is then necessary to send a `Updatesignals` message to the CDPCoDeSys component.

When CDPCoDeSys starts and AutoUpdateDisabled is on, it will use the signal configuration present in the CDPCoDeSys component .xml file. If AutoUpdateDisabled is off, the signal configuration is always read from softplc configuration on startup.

2.2. Interface

2.2.1. Messages

Also see base component specification for message interface description.

Message	Description
Updatesignals	Update CoDeSys signals. Sometimes necessary to do after updating softplc program.

2.2.2. XML Attributes

Attribute	Description
Name	CDPCoDeSys component name, default 'CDPCoDeSys'.
Type	'CDPCoDeSys'

2.2.3. XML Values

Value	Description
fs	Global signal update, default 10 Hz.

2.2.4. Parameters

Value	Description
Debug messages on	Set to '1' to enable CoDeSys debug messages printout on CDPMessages.
ExportOnlyIN_OUT_Prefix	Set to '1' if only CoDeSys signals with name prefix IN_, OUT_ should be exported as CDP signals.
AutoUpdateDisabled	Set to '1' to disable automatic update of CoDeSys signal configuration.

2.2.5. Signals

Value	Description
CoDeSys state	CoDeSys plc run state.
CoDeSys global exports	All signals exported as global in CoDeSys soft- plc are automatically created. The custom attribute CoDeSys set to '1' identifies a CoDeSys exported signals.

2.2.6. Example .xml

```

<?xml version="1.0" ?>

<!-- CDPCoDeSys component model. -->
<Component Name="CDPCodeSys" Type="CDPCoDeSys">

  <Activate>1</Activate>
  <fs>10 </fs>

  <InstanceHelp></InstanceHelp>

  <Signals>
    <Signal Name="CoDeSys state" Value="0" Type="int" RouteType="Variable" Routing="No routing" Description="CoDeSys plc run state."></Signal>

    <Signal Name='a[1]' CoDeSys='1' Type='short' Input='0' Value='0' Routing='No routing' Description='CoDeSys signal.'></Signal>
    <Signal Name='bitOut4' CoDeSys='1' Type='bool' Input='1' Value='0' Routing='No routing' Description='CoDeSys signal.'></Signal>
    <Signal Name='byt' CoDeSys='1' Type='unsigned char' Input='0' Value='0' Routing='No routing' Description='CoDeSys signal.'></Signal>
    <Signal Name='di' CoDeSys='1' Type='double' Input='0' Value='0' Routing='No routing' Description='CoDeSys signal.'></Signal>
    <Signal Name='double' CoDeSys='1' Type='double' Input='0' Value='0' Routing='No routing' Description='CoDeSys signal.'></Signal>
  </Signals>

  <Alarms>
  </Alarms>

  <Parameters>
    <Parma Name="Debug messages on" Value="1" DefaultValue="1" PreviousValue="1" TimeLastChanged="0" Description="Enable CoDeSys debug messages.">5/Parma>
    <Parma Name="ExportOnlyIN_OUT_Prefix" Unit="" Value="0.0" DefaultValue="0.0" PreviousValue="0.0" TimeLastChanged="0" Description="Only CoDeSys signals with name prefix IN_, OUT_ are exported as CDP signals." CppName="m_exportOnlyIN_OUT_Prefix"></Parma>
    <Parma Name="AutoUpdateDisabled" Unit="" Value="1" DefaultValue="0" PreviousValue="0" TimeLastChanged="Fri Feb 17 17:57:50 2006" Description="Disable automatic update of CoDeSys signal configuration."></Parma>
  </Parameters>

</Component>

```

2.3. Specification

2.3.1. CoDeSys version

This version CDP 2.2.0.8 is compiled and tested for CoDeSys version 2.3 modified for CDPCoDSys. Source code for CoDeSys runtime is distributed as part of the CDPCoDeSys installation.

2.3.2. CDP version

This version CDP 2.2.0.8 is compiled and tested for CDP Release 2.2.0.8.

CDPCoDeSys version CDP 2.2.0.8 requires CDP version 2.2.0.8.

If you have source code files for CDPCoDeSys you can compile for other CDP versions CDP Release 2.2.0.8 or newer.

2.3.3. Signal types

CDPCoDeSys supports many CoDeSys signal types. Global CoDeSys signals of unsupported types will not be available as CDP signals.

CDPCoDeSys does not support arrays as array type or strings. Array elements will be created as individual signals.

Supported CoDeSys signal types

CoDeSys type	Description
Byte BOOL	Boolean mapped to 8- bit unsigned, maps to CDP type bool.
Bit BOOL	Boolean mapped to bit, maps to CDP type bool.
BYTE	8 bit unsigned integer, represented as unsigned char.
SINT, USINT	8 bit signed, unsigned integer, represented as signed, unsigned char.
WORD	16 bit unsigned integer ("Wort"), represented as unsigned short.
INT, UINT	16 bit signed, unsigned integer ("Wort"), represented as signed, unsigned short.
DWORD	32 bit unsigned integer, represented as unsigned int.
DINT, UDINT	32 bit signed, unsigned integer, represented as signed, unsigned int.
REAL	32 bit floating point, represented as float.
LREAL	64 bit floating point, represented as double.
TIME	Handled as 32- bit unsigned int.
LWORD	64 bit unsigned integer, represented as double.
LINT, ULINT	64 bit signed, unsigned integer, represented as double.

2.3.4. Operating systems

Supported operating systems

Operating system	Description
Win32	Supports soft- real- time implementation of CoDeSys soft- plc inside CDPCoDeSys on Windows NT 2000, Windows Xp Professional.
On- Time RTOS32	Supports CoDeSys soft- plc inside CDPCoDeSys on controllers running RTOS32, version 4.16 with RTOS 32 ip-stack fix.

2.3.5. Performance

To achieve the desired flexible functionality, some performance issues must be considered when creating CoDeSys soft- plc applications. Load time and CPU load while running depends on number of global CoDeSys signals, so it is a good idea to limit the number of global signals. This will also make it easier to find

Internal performance of CoDeSys soft- plc is not affected by the performance issues described here, still it is important to note that CoDeSys soft- plc will also consume CPU time. The cpu load which can be read from the application CDPULoad signal can be used to determine the total load of the application, including both CDP components and the CoDeSys soft- plc.

Win32 on PIII 1.4 MHz computer

Value	Specification
Max recommended global signals	2000
CPU load	< 1% for 2000 signals, 10Hz.
Config update time	< 1.0 s. for 2000 signals.

RTOS32 on ICOP PII 166MHz controller

Value	Specification
Max recommended global signals	2000
CPU load	CDPCoDeSys component: 6% for 2000 signals, 10 Hz. Total: 22% for 2000 signals, 10 Hz.
Config update time	8.0 s.