



Product:	Hydac CS1000 Decoder
Product version:	v1.1
Document ID:	UM-Hydac CS1000 Decoder
Doc revision:	A1
Written/Aprr.:	NPE/RE
Date:	07.10.2008

Industrial Control Design AS



Hydac CS1000 Decoder v1.1

User Manual

The content of this document is confidential information not to be published without the consent of Industrial Control Design AS.

Industrial Control Design AS, www.icd.no, support@icd.no, forum.icd.no

Contents

1. INTRODUCTION.....	3
1.1. About.....	3
1.2. Overview.....	3
1.2.1. Features.....	3
1.2.2. Decoder illustration.....	3
2. APPLICATION CONFIGURATION.....	4
2.1. About.....	4
2.2. How to add the CS1000 Decoder to a CDP Application.....	4
2.3. How to add the CS1000 Simulator to a CDP Application.....	4
3. CS1000 DECODER CONFIGURATION.....	5
3.1. About.....	5
3.2. DecodeMethod.....	5
3.2.1. Description.....	5
3.2.2. Example XML.....	5
3.2.3. Elements.....	5
3.3. InputSignal.....	5
3.3.1. Description.....	5
3.3.2. Example XML.....	5
3.3.3. Elements.....	5
3.4. Signals.....	6
3.5. Parameters.....	6
4. CS1000 SIMULATOR CONFIGURATION.....	7
4.1. About.....	7
4.2. Decode Method.....	7
4.2.1. Description.....	7
4.2.2. Example XML.....	7
4.2.3. Values.....	7
4.3. Signals.....	7
4.4. Parameters.....	8
5. APPENDIX.....	9
5.1. Example CS1000Decoder Component XML-file.....	9
5.2. Example CS1000Simulator Component XML-file.....	10

1. Introduction

1.1. About

This document describes how the Hydac CS1000 Decoder component works, and how to set it up and use it with the CDP system. See the following section for an overview of the component.

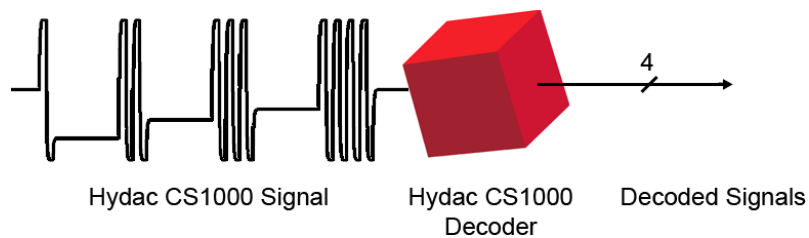
1.2. Overview

1.2.1. Features

The Hydac CS1000 Decoder component has the following features:

- Decodes the time-coded analog signals from a Hydac CS 1000 Particle Counter and gives out the SAE class (AS 4059), ISO code (4406:1999), ISO code (4406:1987) or NAS 1638.
- Setting the decoding type and input signal source is done in the component's XML-file (CS1000Decoder.xml).

1.2.2. Decoder illustration



2. Application Configuration

2.1. About

The Hydac CS1000 Decoder library includes two components. That is the actual decoder component and a simulator component that enables simulated CS1000 signals. This chapter describes how to instantiate the components within a CDP application.

2.2. How to add the CS1000 Decoder to a CDP Application

Add the following to your project's Application.xml:

Inside the <Components> element, add an instance of a CS1000Decoder component, for instance:

```
<Component Name="CS1000Decoder" src="Components/CS1000Decoder.xml"></Component>
```

Or, inside the <Subcomponents> element, add:

```
<Subcomponent Name="CS1000Decoder" Model="CS1000Decoder" src="Components/CS1000Decoder.xml">
</Subcomponent>
```

This will tell CDP to initialize a component named “CS1000Decoder” from a component file located at “Components/CS1000Decoder.xml”. Make sure that your Models\ folder contains an CS1000Decoder.xml model file, or the component will not be initialized correctly.

Configuration is done by modifying the component xml file. It should not be necessary to modify the model xml file. An example of CS1000Decoder.xml file (component xml) is found in 5.1.

2.3. How to add the CS1000 Simulator to a CDP Application

Add the following to your project's Application.xml:

Inside the <Components> element, add an instance of a CS1000Simulator component, for instance:

```
<Component Name="CS1000Simulator" src="Components/CS1000Simulator.xml"></Component>
```

Or, inside the <Subcomponents> element, add:

```
<Subcomponent Name="CS1000Simulator" Model="CS1000Simulator" src="Components/CS1000Simulator.xml">
</Subcomponent>
```

This will tell CDP to initialize a component named “CS1000Simulator” from a component file located at “Components/CS1000Simulator.xml”. Make sure that your Models\ folder contains an CS1000Simulator.xml model file, or the component will not be initialized correctly.

Configuration is done by modifying the component xml file. It should not be necessary to modify the model xml file. An example of CS1000Simulator.xml file (component xml) is found in 5.2.

3. CS1000 Decoder Configuration

3.1. About

This chapter describes the various configuration parameters of the Hydac CS1000 Decoder.

3.2. DecodeMethod

3.2.1. Description

Defines which decoding method should be used for the incoming signal.

3.2.2. Example XML

```
<DecodeMethod>SAE</DecodeMethod>
```

3.2.3. Elements

Element	Description
DecodeMethod	Defines which decoding method should be used for the incoming signal. Alternatives are 'SAE', 'ISO 1999', 'ISO 1987' and 'NAS'. 'SAE' is the default if not specified.

3.3. InputSignal

3.3.1. Description

Defines the component from which the input signal will be read.

3.3.2. Example XML

```
<Signals>
  <Signal Name="InputSignal" Input="1" Type="double" Unit="" Value="" Routing="CS1000Simulator.Output"
    Description="The input signal.">
  </Signal>
</Signals>
```

3.3.3. Elements

Element	Description
Routing	Defines the component from which the input signal should be read.

3.4. Signals

The following two signals are available in the CS1000 Decoder no matter the decode method. In addition, there are one output signal for every decoded class value. These signals are named differently depending on the decode method. Note that when SAE or NAS is selected there are four such output signals, while only three is available when one of the ISO classes is selected.

Signal Name	Description
InputSignal	Coded input signal. See section 3.3 for more details.
Temperature	Decoded temperature output signal.

3.5. Parameters

The following parameters are in the CS1000 Simulator:

Parameter Name	Description
Read Value Length	Time to read the value signal. Should always be $\frac{1}{2}$ of the ValueTime.
mA High	High value for sync
mA Low	Low sync value

4. CS1000 Simulator configuration

4.1. About

The Hydac CS1000 Decoder library includes a simulator to test the decoder with various inputs. This chapter describes how to configure and use the simulator.

4.2. Decode Method

4.2.1. Description

The decode method of the simulator can be changed by editing the CDP Parameter named “Decode Method”. Set the parameter to 1 for SAE, 2 for ISO 1999, 3 for ISO 1987 and 4 to use the NAS decode class. Note that the decode method must be identical for both the simulator and the decoder.

4.2.2. Example XML

```
<Parameters>
  <Parma Name="Decode Method" Unit="" Value="3" Min="1" Max="4" DefaultValue="0" PreviousValue="1"
    TimeLastChanged="Thu Oct 02 11:23:16 2008"Description="1 = SAE, 2 = ISO 1999, 3 = ISO 1987, 4 = NAS"
    CppName="m_decodeMethod">
  </Parma>
</Parameters>
```

4.2.3. Values

Element	Description
values	Set to 1 for SAE, 2 for ISO 1999, 3 for ISO 1987 and 4 to use the NAS decode class.

4.3. Signals

The following signals are in the CS1000 Simulator:

Signal Name	Description
Output	Time Coded 4-20 mA output signal
CLASS_A	Input signal for the first class value. (The class type depends on the decode method.)
CLASS_B	Input signal for the second class value. (The class type depends on the decode method.)
CLASS_C	Input signal for the third class value. (The class type depends on the decode method.)
CLASS_D	Input signal for the second class value. Note that this signal is not in use when selecting ISO class types.
CLASS_T	Calculated from Temperature signal
Temperature	Temperature signal
OutputNoise	Output noise signal

4.4. Parameters

The following parameters are in the CS1000 Simulator:

Parameter Name	Description
Decode Method	Select decode method by entering a number between 1 and 4. See section 4.2 for more details.
Noise	Parameter for adding noise to the coded output signal.
SyncTime	SyncTime in seconds (Time to hold sync pulse)
ValueTime	Time to hold value signal
HighSyncValue	High value for sync
LowSyncValue	Low sync value
OutputFilter_b0	Filter coefficient

5. Appendix

5.1. Example CS1000Decoder Component XML-file

```

<?xml version="1.0" encoding="iso-8859-1"?>
<Component Name="CS1000Decoder" Model="CS1000Decoder">
  <Activate>1</Activate>
  <fs>100</fs>
  <InitialState>Null</InitialState>

  <Description><![CDATA[
    ]]></Description>

  <DecodeMethod>SAE</DecodeMethod>

  <Signals>
    <Signal Name="InputSignal" Input="1" Type="double" Unit="" Value="" Routing="CS1000Simulator.Output"
      Description="The input signal."></Signal>
    <Signal Name="Temperature" Input="0" Type="double" Unit="" Value="" Routing="No routing"
      Description=""></Signal>
  </Signals>

  <Alarms>
  </Alarms>

  <Parameters>
    <Parma Name="mA High" Unit="" Value="19.2" Min="" Max="" DefaultValue="0.0" PreviousValue="0.0"
      TimeLastChanged="" Description=""></Parma>
    <Parma Name="mA Low" Unit="" Value="4.8" Min="" Max="" DefaultValue="0.0" PreviousValue="0.0"
      TimeLastChanged="" Description=""></Parma>
    <Parma Name="Sync Detail" Unit="" Value="0.9" Min="" Max="" DefaultValue="0.0" PreviousValue="0.0"
      TimeLastChanged="" Description=""></Parma>
    <Parma Name="Read Value Length" Unit="" Value="1.5" Min="" Max="" DefaultValue="0.0"
      PreviousValue="0.0" TimeLastChanged="" Description=""></Parma>
  </Parameters>

  <Subcomponents>
  </Subcomponents>

  <RemoteComponents>
  </RemoteComponents>
</Component>

```

5.2. Example CS1000Simulator Component XML-file

```

<?xml version="1.0" encoding="iso-8859-1"?>
<Component Name="CS1000Simulator" Model="CS1000Simulator">
  <Activate>1</Activate>
  <fs>100</fs>
  <InitialState>Null</InitialState>

  <Description><![CDATA[
    Simulator for a Hydac CS1000
  ]]></Description>

  <Signals>
    <Signal Name="Output" Input="0" Type="double" Unit="mA" Value="" Routing="No routing"
      Description="Time Coded 4-20 mA output signal"></Signal>
    <Signal Name="CLASS_A" Input="1" Type="double" Unit="Class" Value="4" Routing="No routing"
      Description="CLASS A Signal"></Signal>
    <Signal Name="CLASS_B" Input="1" Type="double" Unit="Class" Value="7.3" Routing="No routing"
      Description="CLASS B Signal"></Signal>
    <Signal Name="CLASS_C" Input="1" Type="double" Unit="Class" Value="9.2" Routing="No routing"
      Description="CLASS C signal"></Signal>
    <Signal Name="CLASS_D" Input="1" Type="double" Unit="Class" Value="13.8" Routing="No routing"
      Description="CLASS D Signal"></Signal>
    <Signal Name="CLASS_T" Input="0" Type="double" Unit="Class" Value="38" Routing="No routing"
      Description="Calculated from Temperature signal"></Signal>
    <Signal Name="Temperature" Input="1" Type="double" Unit="DegC" Value="38" Routing="No routing"
      Description="Temperature signal"></Signal>
    <Signal Name="OutputNoise" Input="0" Type="double" Unit="mA" Value="2" Routing="No routing"
      Description="output noise signal"></Signal>
  </Signals>

  <Alarms>
  </Alarms>

  <Parameters>
    <Parma Name="Decode Method" Unit="" Value="2" Min="1" Max="4" DefaultValue="0"
      PreviousValue="1" TimeLastChanged="Thu Oct 02 11:23:16 2008" Description="1 = SAE, 2 = ISO 1999,
      3 = ISO 1987, 4 = NAS" CppName="m_decodeMethod"></Parma>
    <Parma Name="Noise" Unit="mA" Value="0" DefaultValue="0.1" Description="" Min="" Max=""
      PreviousValue="1" TimeLastChanged="Thu Oct 02 11:18:13 2008"></Parma>
    <Parma Name="SyncTime" Unit="s" Value="0.3" DefaultValue="0.3" Description="SyncTime in seconds
      (Time to hold sync pulse)" Min="1.e-002" Max="0.5" PreviousValue="0.2" TimeLastChanged="Wed Oct
      01 16:34:40 2008"></Parma>
    <Parma Name="ValueTime" Unit="s" Value="3" DefaultValue="3" Description="Time to hold value signal"
      Min="0.5" Max="1.e+020" PreviousValue="1" TimeLastChanged="Wed Oct 01 16:37:04 2008"></Parma>
    <Parma Name="HighSyncValue" Unit="mA" Value="19.2" DefaultValue="19.2" Description="High value for
      sync" Min="" Max="" PreviousValue="14" TimeLastChanged="Tue Sep 30 18:45:20 2008"></Parma>
    <Parma Name="LowSyncValue" Unit="mA" Value="4.8" DefaultValue="4.8" Description="Low sync value" Min=""
      Max="" PreviousValue="5" TimeLastChanged="Wed Jan 09 16:41:53 2008"></Parma>
    <Parma Name="OutputFilter_b0" Value="0.3" DefaultValue="0" PreviousValue="0.4" TimeLastChanged="Tue
      Sep 30 18:36:00 2008" Description="filter coefficient" Unit=""></Parma>
  </Parameters>

  <Subcomponents>
  </Subcomponents>

  <RemoteComponents>
  </RemoteComponents>
</Component>

```