



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

## Industrial Control Design AS



# Hydac CS1000 Decoder v1.1

## Programmers 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The content of this document is confidential information not to be published without the consent of Industrial Control Design AS.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](http://www.icd.no), [support@icd.no](mailto:support@icd.no), [forum.icd.no](http://forum.icd.no)

# Contents

---

<b>1. INTRODUCTION.....</b>	<b>3</b>
1.1. About.....	3
1.2. Overview.....	3
1.2.1. Features.....	3
1.2.2. Decoder illustration.....	3
<b>2. INSTALLATION.....</b>	<b>4</b>
2.1. Prerequisites.....	4
2.2. How to add the CS1000Decoder library in a new or existing project.....	4
2.3. Modify the project xml files.....	7

# 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. Details regarding configuration and setup is located in the User Manual.

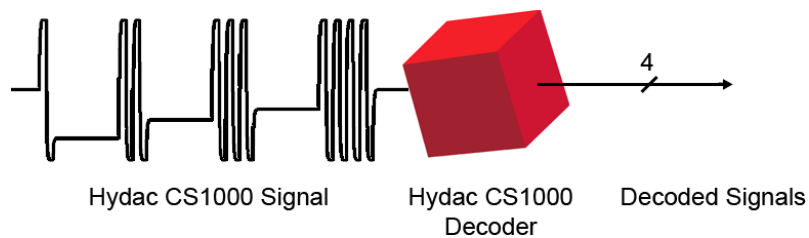
## 1.2. Overview

### 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. Installation

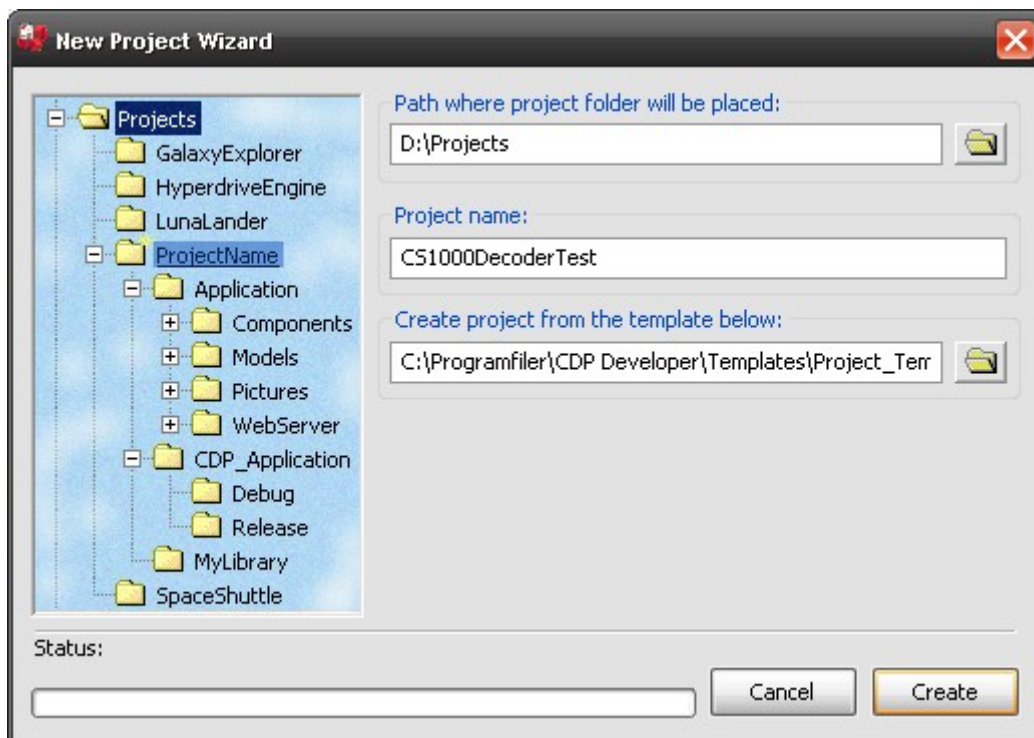
This chapter explains how to install the CD1000Decoder library.

## 2.1. Prerequisites

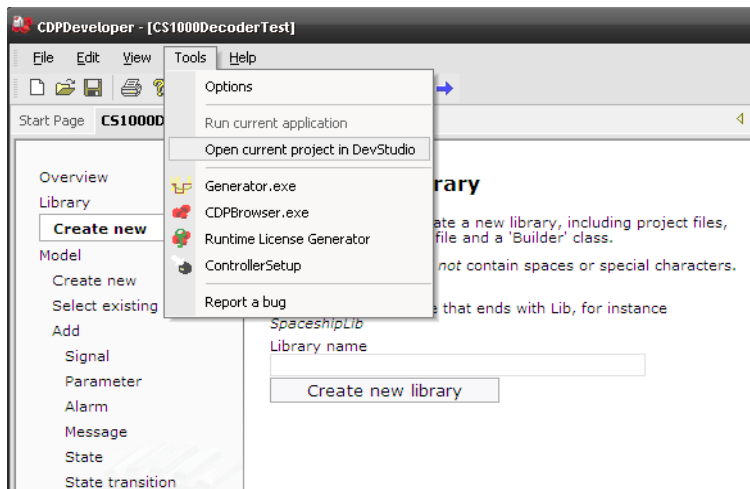
- A valid CDP license
- Familiar with CDP
- CDP version 2.3.0.0/RTOS version 5.06

## 2.2. How to add the CS1000Decoder library in a new or existing project

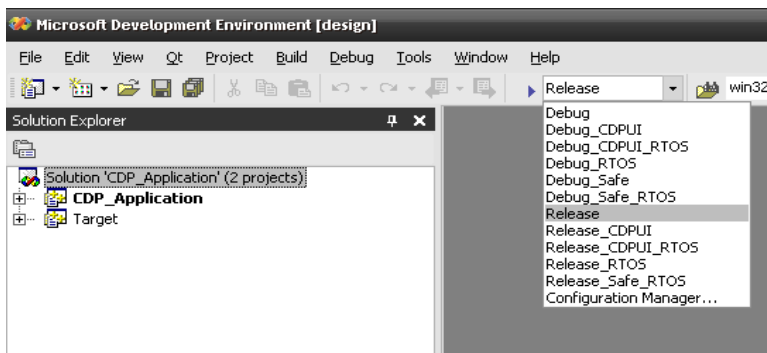
- Start CDPDeveloper located at Start Menu > Programs > CDP > CDPDeveloper
- Make a new project by Selecting File->New, type in project name and click Create (or skip four steps forward and open an already existing project in Visual Studio)



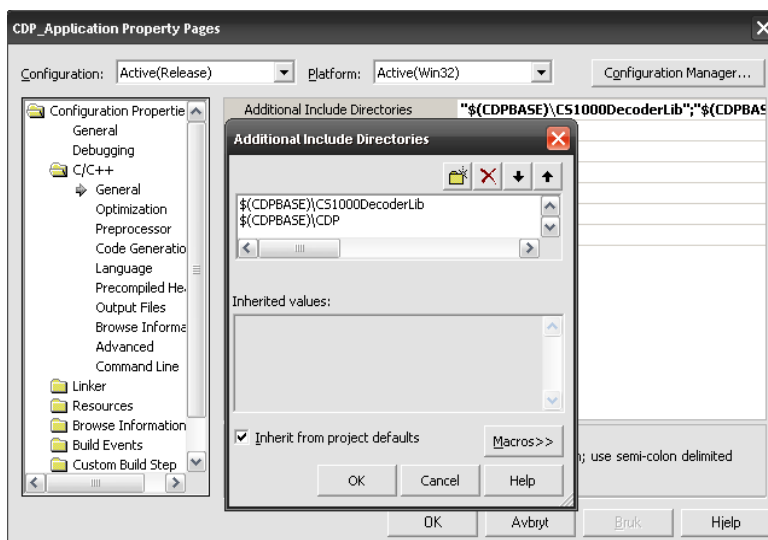
- Choose 'Open current project in Devstudio' from the Tools menu:



- If Visual Studio asks to convert the project, accept this by selecting 'Next'/'Finish'/'Close' until done. Close the conversion report.
- In Visual Studio, select the configuration that is right for your target platform.

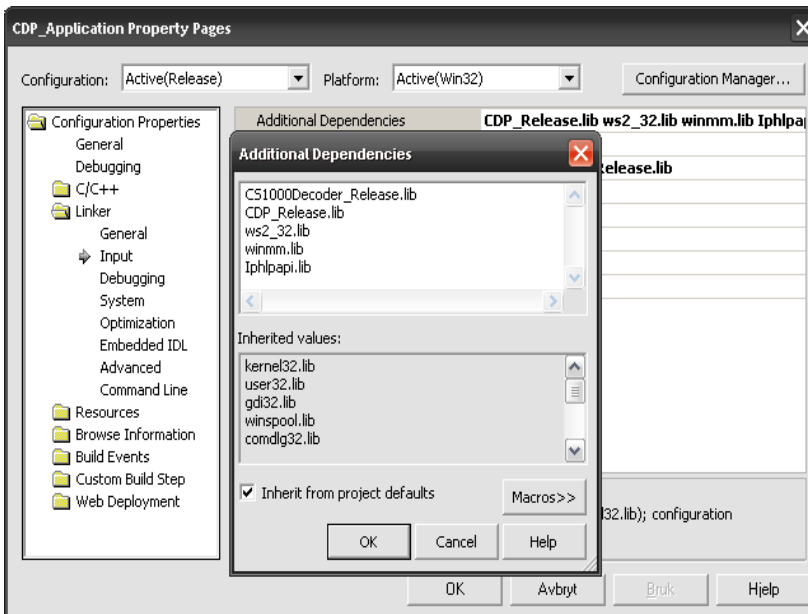


- Select 'CDP\_Application' from the 'Solution Explorer', right-click and select Properties.
- In C++/Additional include Directories, make sure it says: “\$(CDBASE)\CS1000DecoderLib”; “\$(CDBASE)\CDP”. When compiling for Ontime RTOS, the following include is also required: “;”\$(RTTarget)\Include”.

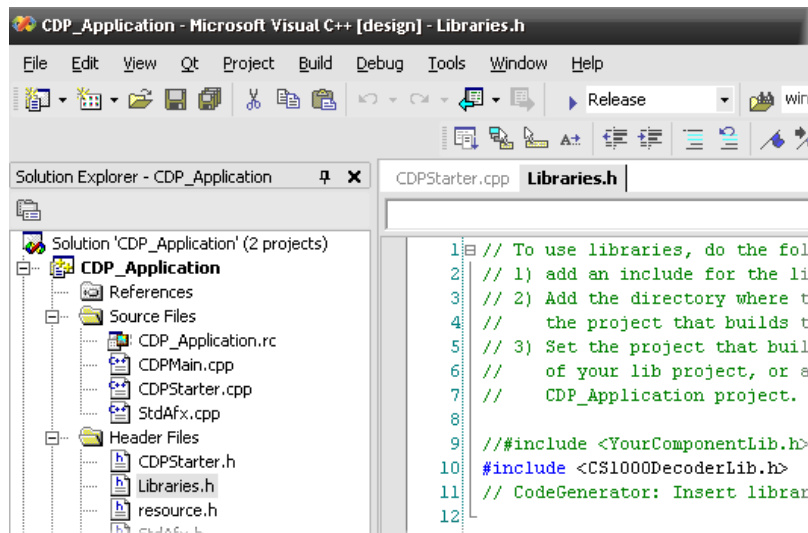


The compiler will look in the directories specified in 'Additional Include Directories' for files that you #include in your .cpp and .h files. If you get an error 'Can not open include file ...' then it is most likely caused by a missing include directory, or that the file you #include does not exist.

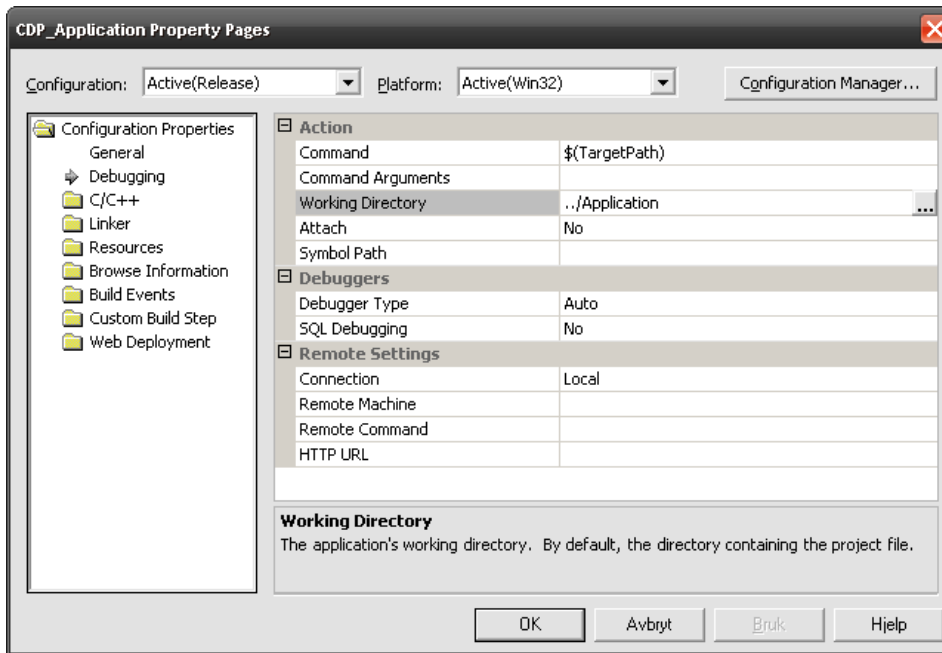
- In Linker->Input, make sure you list CS1000DecoderLib\_Release.lib and CDP\_Release.lib in addition to all the win32 libraries you need. This will ensure that the linker finds all the functions that is referenced in the code. Note that when compiling for other Operating Systems you will need to replace these libraries with the ones for the target OS.



- In the 'Solution Explorer', inside the 'CDP\_Application' project, locate the Header file Libraries.h, and add the line #include <CS1000DecoderLib.h>:



- Finally, make sure that the working directory of the project is set to ../Application. This will ensure that the executable is started where the configuration files are located.



## 2.3. Modify the project xml files

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.