



Product:	Phoenix IB IL 24 BK Modbus Setup
Product version:	V1.1
Document ID:	UM-Phoenix IB IL 24 BK Modbus
Doc revision:	A
Written/Appr.:	RE / SL
Date:	10/03/2008

Industrial Control Design AS



Phoenix IB IL 24 BK Modbus Setup 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The content of this document is confidential information not to be published without the consent of Industrial Control Design AS.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
<hr/>	
2. CONFIGURATION.....	4
2.1. ErrorPackets.....	4
2.1.1. Description.....	4
2.1.2. XML Example.....	4
2.2. Packet setup.....	4
2.3. Phoenix Analog Input Settings.....	5
2.3.1. Common values.....	5

1. Introduction

1.1. About

This document describes how to set up the XML for a Phoenix Contact IB IL 24 BK. For more information about Phoenix Contact Modules / devices, please visit <http://www.phoenixcontact.com/>.

This document requires some knowledge about Modbus (<http://www.modbus.org/>), Packets and FunctionDefinitions. Please see the CDP documentation 'UM-Modbus Setup' and 'UM-Packets and FunctionDefinitions ' for more information.

The Phoenix Contact IB IL 24 BK is an Input/Output module that can communicate using Modbus request/reply packets. It is connected via TCP/IP and can be operated at speeds up to approximately 200 Hz.

2. Configuration

Configuration is done by modifying the component xml file inside the Application\Components\ folder. It should not be necessary to modify the model xml file. Please see the document 'UM-Modbus Setup' for how to set up a ModbusTCPIOServer. The following section assumes that you have read that document, and will only describe what you need to set up in addition for the Phoenix IO to work properly.

2.1. ErrorPackets

2.1.1. Description

You must set up an ErrorPacket to clear the phoenix error code on startup. Errorpackets are sent sequentially when a modbus error response has been received.

2.1.2. XML Example

```
<ErrorPackets>
  <Packet Name="ClearPhoenixError" FunctionCode="MODBUSClearPhoenixError" NetworkConvert="1"></Packet>
</ErrorPackets>
```

Element	Description
ErrorPackets	<p>Surrounding element for packets to send to the IONode when a modbus error occurs. Each packet listed here is sent sequentially in the order they are defined in the XML file, with approximately 100 milliseconds time-delay between each packet.</p> <p>The above ErrorPackets can be used on a Phoenix I/O module, where a 'clear error status' packet is required on first-time-communication after a communication failure.</p>

Please note that the FunctionCode(s) specified in the packets in <ErrorPackets> **must** have been read by the DefinitionManager.

2.2. Packet setup

You can access the I/O via Modbus/TCP in two ways.

The most efficient way is to use the FunctionCode 'ReadWriteMultipleRegisters'. This will send the data to write and request the data to read in just one Modbus request. You then specify just one packet with inputs and outputs. The inputs are listed first, and start at offset 0x8000, and the Outputs start at the last Input Offset + Register size of the last input (for Analog channelgroups this is the same as the NumberOf attribute of the last input, for digital channelgroups this is the same as RoundUpToInteger(NumberOf/16)).

The other way to set up communication is to have 4 packets; one for Digital Inputs, one for Analog Inputs, one for Digital Outputs and one for Analog Outputs. If you use this method you must set up 4 packets: Digital Inputs from offset 0, Analog Inputs from Offset 192, Digital Outputs from offset 384 and Digital Outputs from offset 576. This results in three more requests and three more responses than the preferred method. You can use FunctionCodes MODBUSReadHoldingRegisters and MODBUSWriteMultipleRegisters for reading and writing values.

2.3. Phoenix Analog Input Settings

The Phoenix contact I/O needs to have Analog Inputs set up to specify the format for some Analog output modules. The input registers of the analog input module is set up by setting bits in corresponding output registers. Please see how this is set up by searching www.phoenixcontact.com/download for 'IB IL AI 2/SF-PAC', where the various formats are described. You will then have to set up Output channels with a Value="" the correct format, to get the input channel to function correctly.

2.3.1. Common values

Value	Description of format
0x8120 = 33056	0..10V, no filter, IB RT numeric representation
0x8121 = 33057	+/-10V, no filter, IB RT numeric representation
0x812A = 33066	4..20mA, no filter, IB RT numeric representation