DeviceNet Encoder Gateway

Application and Installation information

1. Introduction

This document contains application and installation information on the Leine & Linde DeviceNet Encoder Gateway based on the device profile, Encoder Profile. The gateway is the interface between the CAN bus and the encoder. The communication between the gateway and the encoder is handled through a fast Bidirectional Synchronous-Serial Interface called Endat. In the EnDat encoder, advanced diagnostics are integrated together with the position sensing parts to make it possible to check the correctness of the position value very thoroughly.

1.1 DeviceNet communication model

DeviceNet is a low-level network that provides connections between simple industrial devices (sensors, actuators) and higher-level devices (controllers). DeviceNet provides Master/Slave and Peer-to-Peer capabilities over the CAN bus.

DeviceNet has two primary purposes:

- Transport of control-oriented information associated with low-level devices
- Transport of other information, which is indirectly related to the system being controlled, such as configuration parameters.

DeviceNet makes use of abstract object modelling to describe:

- The suite of communication services available
- The externally visible behaviour of a DeviceNet node
- A common means by which information within DeviceNet products is accessed and exchanged

A DeviceNet node is modelled as a collection of Objects. An Object provides an abstract representation of a particular component within a product. The realization of this abstract object model within a product is implementation dependent. In other words, a product internally maps this object model in a fashion specific to its implementation.
The Encoder Profile is a description of the objects and functions available to the user of the Leine & Linde DeviceNet Encoder Gateway.

Objects implemented in the Encoder Gateway:

- Identity Object
  - Provides identification of and general information about the device. Also the reset function is implemented here.

- Message Router
  -The Message Router Object provides a messaging connection point through which a Client may address a service to any object class or instance residing in the physical device.

- Connection Class Object
  - Provides the configuration and status of a poll connection port.

- DeviceNet Object
  - Provides the configuration and status of a DeviceNet port.

- Assembly Object
  - Binds attributes of multiple objects, which allows data to or from each object to be sent or received over a single connection. By addressing different instances of the Assembly object, different collections of data are returned in each poll cycle.

- Position Sensor Object
  - Provides the configuration and status of the Encoder

- Acknowledge Object
  - Used to manage the reception of message acknowledgments. This object communicates with a message producing Application Object within a device. The Acknowledge Handler Object notifies the producing application of acknowledge reception, acknowledge timeouts, and production retry limit.

1.2 DeviceNet's use of the CAN identifier field

The 11 CAN Identifier bits available on DeviceNet are subdivided into four separate message groups: Group 1, Group 2, Group 3, and Group 4.

Identifier Bits											*Hex Range*	*Identity Usage*
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 Message ID				Source MAC ID						0x000 – 0x3FF	Message Group 1
1	0	MAC ID						Group 2 Message ID			0x400 – 0x5FF	Message Group 2
1	1	Group 3 Message ID			Source MAC ID						0x600 – 0x7BF	Message Group 3
1	1	1	1	1	Group 4 Message ID						0x7C0 – 0x7EF	Message Group 4
1	1	1	1	1	1	1	X	X	X	X	0x7F0 – 0x7FF	Invalid CAN ID's

- Message ID - Identifies a message within a Message Group inside a particular end-point.
- Source MAC ID - The MAC ID assigned to the transmitting node.
- Destination MAC ID - The MAC ID assigned to the receiving device. Message Group 2 allows the specification of either Source or Destination within the MAC ID portion of the CAN Identifier Field

The MAC ID uses 6 bits, which makes it possible to have 0-63 node addresses. Address 0 is not allowed. The Leine & Linde Encoder gateway uses Message Group 2.

For further information please see the DeviceNet Specification [4].

1.3 Profile overview

The Encoder Profile defines the functionality of encoders connected to DeviceNet. In the Encoder Profile are all Objects described that are used from DeviceNet Object library. Particular interesting is the Position Sensor Object (0x23 Hex). It describes the services that are available for fetching positions, scaling of position values and other useful info.

For further information please see the Encoder Profile specification [6].

1.4 References

Document Name:	*Author:*
[1] http://www.odva.org	ODVA
[2] CAN Specification Version 2.0 A	Robert Bosch GmbH
[3] 82527 Architectural Overview	Intel
[4] ODVA DeviceNet Specification 2.0	ODVA
[5] Statement of Conformance	Leine & Linde AB
[6] Encoder Profile Revision 6	ODVA / Leine & Linde AB
[7] Encoder Gateway EDS-File	Leine & Linde AB

2. Functional overview

2.1 DeviceNet supported functionality

The Leine & Linde DeviceNet Encoder Gateway supports the Predefined Master/Slave Connection Set (Group 2 only server). There are two possibilities to retrieve the position and/or velocity value. One way is to directly read the position attribute in the Position Sensor object or to allocate a poll channel. A poll channel always returns the active assembly object instance (see below). The advantage to use a poll channel instead is that you can use different poll allocation modes and also return more data on a single request. For gathering of position data / velocity value, the following allocation modes are supported:

- Poll I/O Connection
  - When the master sends a poll request, the Encoder Gateway immediately returns a poll response holding the latest calculated position value (returns the active assembly instance). See section 2.3.

- Bit–Strobe I/O Connection
  - When sent by a master, all units affected (multi cast) respond with a bitstrobe response (returns the active assembly instance). This can be used to get a synchronised reading. Remember that the CAN-bus always priorities the node with the lowest address to send data before a node with higher address.

- Change of State
  - Returns a change of state response (returns the active assembly instance), when a change of data in the assembly is detected or a max timeout have occurred. Use the COS/delta attribute in the Position Sensor object to specify the number of position steps that must occur before a Change of State response is triggered. If e.g. the value 100 is used, the position value must be 100 steps larger or smaller compared to the position value last reported to the master before a new Change of State is triggered. *Can not be combined with Cyclic I/O Connection.*

- Cyclic I/O Connection
  - Returns a Cyclic I/O response (returns the active assembly instance), when the specified periodic time has occurred. *Can not be combined with Change of State.*

To specify which assembly instance that should be active when a poll response is returned, use the ‘Set produced connection path’ attribute of the Connection Class object.

The Encoder Gateway supports the following three Assembly instances:

Instance 1:

Byte 0	Position Data (Low byte)
Byte 1	Position Data
Byte 2	Position Data
Byte 3	Position Data (High byte)

Instance 2:

Byte 0	Position Data (Low byte)		
Byte 1	Position Data		
Byte 2	Position Data		
Byte 3	Position Data (High byte)		
Byte 4	Bit 7-2 Unused	Bit 1 Warning Flag	Bit 0 Alarm Flag

Instance 3:

Byte 0	Position Data (Low byte)
Byte 1	Position Data
Byte 2	Position Data
Byte 3	Position Data (High byte)
Byte 4	Velocity value (Low byte)
Byte 5	Velocity value
Byte 6	Velocity value
Byte 7	Velocity value (High byte)

The following attributes are supported in the Position Sensor object (0x23 Hex):

Attribute	Attribute Name	Data Type
3	Position Value (Unscaled)	UDINT
10	Position Value (Scaled)	DINT
11	Position sensor type	UINT
12	Code sequence	BOOL
13	Commissioning diagnostic control	BOOL
14	Scaling function control	BOOL
15	Position format	ENGUNITS
16	Measuring units per revolution	UDINT
17	Total measuring range	UDINT
18	Position measuring step	UDINT
19	Preset value	DINT
20	COS/delta	UDINT
21	Work area state register	BYTE
22	Work area low limit	DINT
23	Work area high limit	DINT
24	Velocity value	DINT
25	Velocity format	ENGUNITS
26	Velocity Resolution	UDINT
41	Operating status	BYTE
42	Physical resolution	UDINT
43	Number of distinguishable revolutions	UINT
44	Alarms	WORD
45	Supported alarms	WORD
46	Alarm flag	BOOL
47	Warnings	WORD
48	Supported warnings	WORD
49	Warning flag	BOOL
50	Operating time	UDINT
51	Offset value	DINT

BOOL	1 Byte
BYTE	1 Byte
WORD	2 Byte
UINT	2 Byte
ENGUNITS	2 Byte
UDINT	4 Byte
DINT	4 Byte

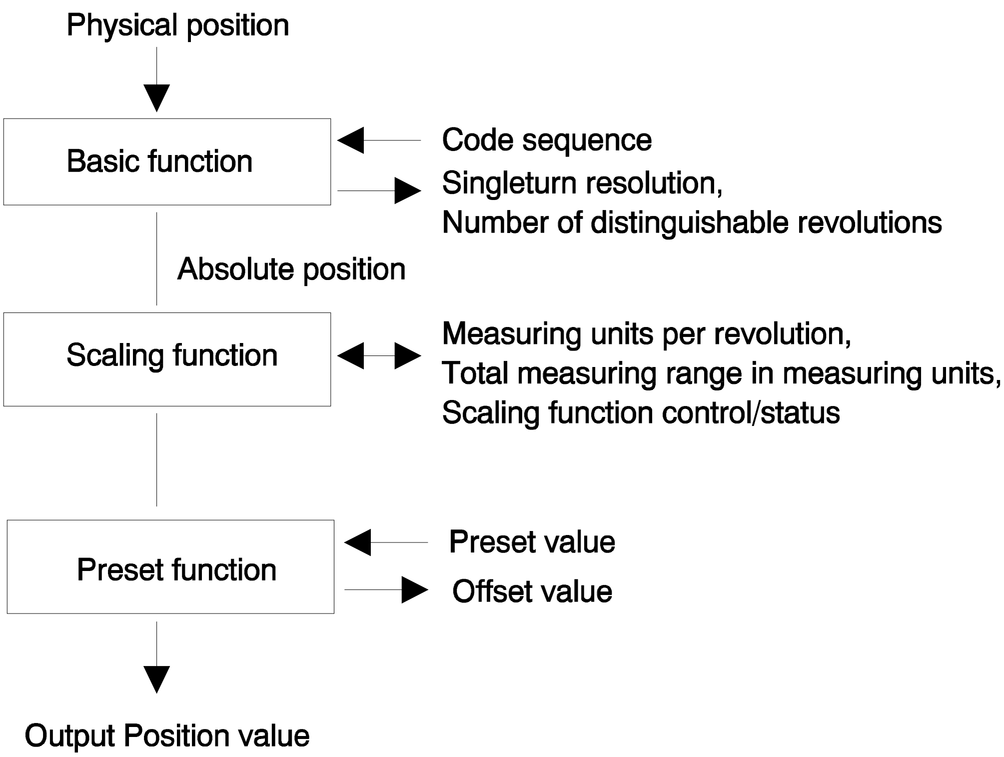
NOTE! The 'Data Type' Engineering Units (ENGUNITS) might not be common knowledge for the majority of people. For a description of how it works, please see Appendix K-2.30 in the DeviceNet Specification [4].

- Attribute 15 "Position format" is a read only attribute in the Leine & Linde Encoder gateway. It will return value 0x1001 (counts) or 0x2203 (mm).
- Attribute 25 "Velocity format" is a read/write attribute. It will accept value 0x1F04 (counts), 0xDF0F (RPM) or 0xDF0E (RPS).

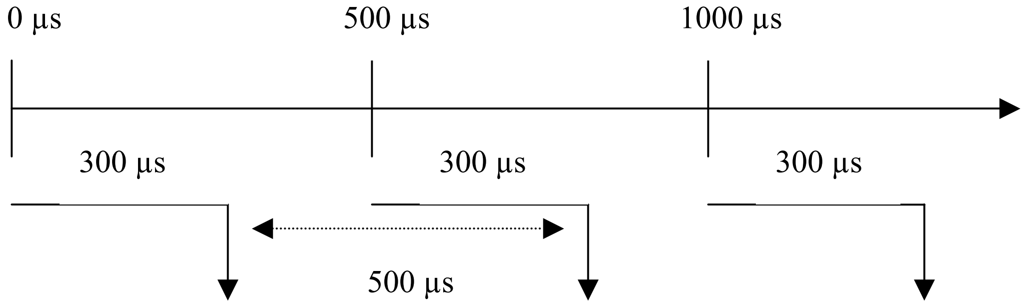
For further information on supported DeviceNet functionality please see the Statement of Conformance [5] document and the Encoder Profile [6].

2.3 Encoder functionality

The figure below gives an overview of the Encoder functions.



The Encoder Gateway fetches a new position value from the Encoder every 500 µs. In a worse case situation, when assembly instance 3 is used, a 300 µs scaling calculation takes place before the output value is stored in the assembly data. This means that a position read from the CAN bus will at a worse case report a 800 µs (500µs + 300µs) + CAN transfer time old position value.



3. Scaling function

3.1 Overview

With the scaling function the Encoder internal numerical value is converted in software to change the physical resolution of the Encoder.
The parameters ”Measuring units per revolution” and ”Total measuring range in measuring units” are the scaling parameters set to operation with the scaling function control bit.

NOTE! When scaling a Multiturn encoder the parameter “Measuring units per revolution” must be sent (downloaded) before the parameter “Total measuring range in measuring units”.

The data type for both scaling parameters is unsigned 32 with a value range from 1 to 2^{32} limited by the encoder resolution. For a 25 bit encoder with a singleturn resolution of 13 bits the permissible value for the ”Measuring units per revolution” is between 1 and 2^{13} (8192) and for the ”Total measuring range in measuring units” the permissible value is between 1 and 2^{25} (33 554 432). The scaling parameters are securely stored in case of voltage breakdown and reloaded at each start-up.

Scaling parameter format:

Byte	**3**	**2**	**1**	**0**
Bit	**31 - 24**	**23 - 16**	**15 - 8**	**7 - 0**
Data	2^{31} - 2^{24}	2^{23} - 2^{16}	2^{15} - 2^{8}	2^{7} - 2^{0}
	Attribute 16 - Measuring units per revolution			

Byte	**3**	**2**	**1**	**0**

Bit	31 - 24	23 - 16	15 - 8	7 - 0
Data	2^{31} - 2^{24}	2^{23} - 2^{16}	2^{15} - 2^{8}	2^{7} - 2^{0}
	Attribute 17 - Total measuring range in measuring units			

3.2 *Scaling formulas*

The scaling function used in the DeviceNet encoder gateway is limited to a singleturn resolution within one step. After downloading new scaling parameters the preset function should be used to set the encoder starting point to the required starting point.

NOTE! Changing the scaling function parameters should only be used at encoder standstill.

In the following formulas a 25 bit multiturn encoder with a singleturn resolution of 13 bits is used as an example.
Formula for the multiturn scaling function:

A = measuring_units_per_revolution * singleturn_position / 8192
output_position = (revolution_number * measuring_units_per_revolution) + A

Where:

singleturn_position =	the Absolute position value of the encoder singleturn disk
revolution_number =	the Absolute revolution number of the encoder multiturn disks

3.2.1 Measuring range

The measuring range is set by the parameter, “Total measuring range in measuring units”. The encoder has two different operating modes depending on the specified measuring range. When the encoder receives new scaling parameters it checks the values for binary scaling and chooses operating mode A if binary scaling detected, see explanation below.

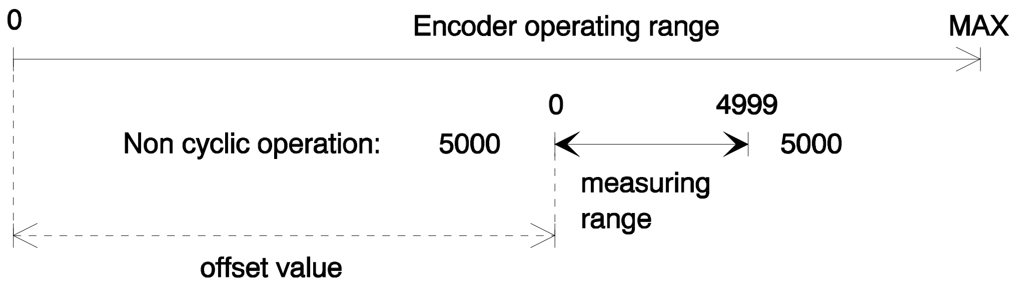
A. Cyclic operation (Binary scaling)
Used when operating with $\mathbf{2^x}$ number of turns (2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096 number of turns).
If the desired measuring range is equal to the specified singleturn resolution $\mathbf{* 2^x}$ (where x <= 12) the encoder operates in endless cyclic operation (0 - max - 0 - max ...). For example: If the position value increases above the maximum value (measuring range-1) by rotating the encoder beyond the maximum value the encoder continues from 0.
Example of a cyclic scaling:

Measuring units per revolution	= 1000
Measuring range	= 32000 ($\mathbf{2^5}$ = 32 number of turns)

B. Non cyclic operation
If the measuring range is used to limit the encoder value range to a value not equal to the specified singleturn resolution *** 2^x** the output position value is limited within the operating range. If the position value increases or decreases outside the measuring range by rotating the encoder beyond the maximum value (measuring range-1) or below 0 the encoder outputs the total measuring range value, see figure below.

Example of a non-cyclic scaling:

Measuring units per revolution: 100
Total measuring range: 5000 steps (50 turns)



4. Preset value

4.1 Overview

The preset function supports adaptation of the encoder zero point to the mechanical zero point of the system. The preset function sets the actual position of the encoder to the preset value. The preset function is used after the scaling function, which means that the preset value is given in the current measuring units.
The encoder handles a preset in the following way:
The encoder reads the current position value and calculates an offset value from the preset value and the read position value. The position value is shifted with the calculated offset value. The offset value can be read from the 'Offset value' parameter and is securely stored in case of voltage breakdown and reloaded at each start-up.

NOTE! The preset function should only be used at encoder standstill.

Preset value format:

Byte	3	2	1	0
Bit	**31 - 24**	**23 - 16**	**15 - 8**	**7 - 0**
Data	2^{31} - 2^{24}	2^{23} - 2^{16}	2^{15} - 2^{8}	2^{7} - 2^{0}
	Attribute 19 - Preset Value			

4.2 *Preset formula*

An offset_value is calculated when the encoder receives the preset_value, see setup calculation below. The offset_value is then used during runtime to shift the current position to get the required output position, see runtime calculation below.

NOTE! In the formulas below the current_position is the Absolute position of the encoder disk after the scaling function. The calculations are made with signed values.

Setup calculation:
offset_value = preset_value - current_position

NOTE! A previously set offset_value is not included in the current position.

Runtime calculation:
output_position = current_position + offset_value

5. Installation

This section handles the installation issues of the Encoder Gateway.

5.1 *Power supply cable*

An external power supply cable can be used if the power supply for some reason can not use the DeviceNet cable.
The gateway should be supplied with 9-30VDC through the integrated screw terminal block. A shielded power cable must be used.

Installation:
1. Remove the cover of the gateway box.
2. Strip the cable ends to the appropriate length, leave app. 15mm of the shield for connection to the cable gland.
3. Insert the power cable through the cable gland.
4. Connect the cables to the +E and 0V screw terminal block. Tighten the terminal screws.
5. Tighten the cable gland and make sure the shield is connected to the gland.

6. Close the cover of the gateway box.

5.2 Bus cable

The Encoder gateway has a non-isolated bus interface. The power supply 0V and the CAN transceiver 0V have equal potential. The CAN transceiver ground is available at the screw terminals marked CAN_GND.
A standard 5 line shielded DeviceNet cable must be used.
The gateway includes a build in T-coupling with a screw terminal block for CAN_H, CAN_L, CAN_GND, E+ and 0 V. Make sure that the incoming cable is connected to the terminal marked IN and the outgoing cable is connected to the terminal marked OUT. If the Encoder Gateway is the last, first or only node connected, make sure that the terminal marked OUT, have a bus terminator connected. A bus terminator is a 121 Ω resistor connecting the CAN_H with CAN_L.

Installation:
1. Remove the cover of the gateway box.
2. Strip the cable ends to the appropriate length, leave app. 15mm of the shield for connection to the cable gland.
3. Insert the bus cable through the cable gland.
4. Connect the cable conductors to the screw terminal block marked BUS (IN or OUT). Make sure that the same conductor is always connected to the same terminal (e.g. white conductor always connected to CAN_H, blue conductor always connected to CAN_L). Tighten the terminal screws.
5. Tighten the cable gland and make sure the shield is connected to the gland.
6. Close the cover of the gateway box.

5.3 Encoder cable

Installation:
1. Make sure the power supply to the gateway is switched off.
2. Connect the male cable-connector to the gateway.
3. Connect the female cable-connector to the encoder.
4. Switch on the gateway power supply.

5.4 Address setting

The physical address (node number) of the gateway must be set between 1 – 63 with the address switch inside the gateway. The gateway reads the address switch only at power-up.

Setting of the address:
1. Make sure the power supply to the gateway is switched off.
2. Set the address with the dipswitches.
3. Switch on the gateway power supply.

5.5 Baud rate setting

DeviceNet uses the 125, 250 or 500Kbit mode of the CAN bus. The Encoder Gateway uses an auto baud rate algorithm to set the baud rate. No DIP-switches needs to be set. The baud rate is detected at power on. To be able to detect the baud rate there must be at least one other node communicating on the CAN bus. It is not possible to only connect two or more auto baud rate nodes to a network. Since all units wait for network traffic to detect baud rate, all units will fail to communicate with one another. Typical a network will have a PLC or other master devices connected. They will periodical send dummy data or connection requests on the network to allow auto baud rate nodes to detect the baud rate.

Another scenario is when a configuration tool like RSNetWorx with only the configuration hardware module (1770-KFD) and an auto baud rate node (Encoder Gateway) is used. Since the configuration hardware does not send any data on the network without being told, a user of RSNetWorx must initiate an operation that generates traffic on the network so that the auto baud rate node can startup.

Setting of the transmission rate:
1. Make sure the power supply to the gateway is switched off.
2. Make sure that there is network traffic.
3. Switch on the gateway power supply.

5.6 EDS files

The EDS-file allows a configuration tool to automate the device configuration process.

The EDS-file serves two main purposes by providing information needed to:
- Describe each device parameter, including it’s legal and default values
- Provide the user with a selection of choices for each configurable parameter in a device.

NOTE! The EDS Installation procedure depends on your configuration tool, please consult your tool supplier if you run into problems.

Installation example using RSNetWorx:
1. Use the ‘Tools’ menu and select ‘EDS Wizard’.
2. Choose ‘Register an EDS file’.
3. Browse for the Leine & Linde DeviceNet Gateway EDS file and select ‘Next’ to continue.
4. No error should occur when RSNetWorx parses the EDS file. Use ‘Next’.
5. Now use ‘Change icon’ to choose an icon that better describe what kind of device the EDS file describes (preferably the Icon included with the EDS file). Then use ‘Next’.
6. Sometimes a restart of RSNetWorx is needed to detect the newly added device.

5.7 Parameter settings

The parameters are usually set from the configuration tool when the device is in the Pre-Operational state using the objects obtained from the EDS-file. The parameters can also be changed during runtime (Operational state), please be careful as the position data is directly effected by some parameters and will change directly following such parameter message. Therefore *changing the scaling function parameters and the code sequence should only be used at encoder standstill.*

NOTE! The parameterisation procedure depends on your configuration tool, please consult your tool supplier if you run into problems.

Setting parameters using RSNetWorx:

1. Right click on the icon for the Encoder Gateway and choose ‘properties’.
2. Select tab ‘Parameters’ and then ‘Upload’.
3. Now you can edit all parameters that are allowed to be changed. Right click on the parameter you want to change and select ‘Edit’.
4. Then use the icon ‘Download parameter’ with the parameter you changed selected.
5. Now you can use the ‘Upload parameter’ to verify that your value was downloaded and accepted by the device.

6. LEDs

The DeviceNet Encoder Gateway has two LEDs, Module and Network Status LED. The Network Status LED indicates the condition of the CAN-bus (DeviceNet Network) as seen by the Encoder Gateways point of view. The Module Status LED shows the internal status of the Encoder Gateway software.

Module Status LED

For this state:	*LED is:*	*To indicate:*
No Power	Off	There is no power applied to the device.
Device Operational	Green	The device is operating in normally.
Device in Standby	Flashing Green	The device needs commissioning due to configuration. Missing, incomplete or incorrect.
Minor Fault	Flashing Red	Recoverable Fault. Means that the Encoder connected to the Encoder Gateway has signal an Alarm state. You may need to replace the connected Encoder.
Unrecoverable Fault	Red	The device has an unrecoverable fault; may need replacing. Means that the Encoder Gateway can not establish contact with the Encoder. Replace Gateway or Encoder.

Network Status LED

For this state:	*LED is:*	*To indicate:*
Not Powered/Not On–line	Off	Device is not on–line.
On–line, Not Connected	Flashing Green	Device is on–line but has no connections in the established state.
On–line, Connected	Green	The device is on–line and has connections in the established state.
Connection Time–Out	Flashing Red	One or more I/O Connections are in the Timed–Out state.
Critical Link Failure	Red	Failed communication device. The device has detected an error that has rendered it incapable of communicating on the network (Duplicate MAC ID, or Bus–off).

Some other useful LED indications:

- If Module Status LED is red at power on, check the cable to the Encoder.
- If both LEDs are off and the Encoder Gateway is connected to a DeviceNet Network, the module is trying to detect the Networks baud rate.

7. Manufacturer specific functions

7.1 Gateway Serial number

The Encoder Gateway has one Manufacturer specific parameter. The Gateway Serial Number parameter fetches the serial number of the Encoder connected to the Encoder Gateway. This is used to keep record of when a new Encoder is connected. If a new Encoder is connected, the stored parameters in the EEPROM memory of the Encoder Gateway are invalidated.

8. Encoder types

A number of different EnDat encoders can be connected to the DeviceNet gateway. This section of the document gives you information on the supported functionality.

8.1 Singleturn Encoders 13 bit

The Scaling function accepts all values between 2 and 8192 measuring units per revolution. The total measuring range is equal to the scaled singleturn resolution.

8.2 Multiturn Encoders 25 bit

The Scaling function parameter ”measuring units per revolution” accepts all values between 2 and 8192. The encoder has two different operating modes

(Cyclic or Non-cyclic) depending on the value of the parameter “total measuring range in measuring units”, please check the section Scaling formulas above for further information.

8.3 Angle Encoders

All EnDat singleturn Angle encoders are supported. The Angle encoders support the scaling function limited to binary scaling. This means that the parameter “measuring units per revolution” accepts only values which are equal to $\mathbf{2^x}$ where x is a value from one to 20 (for a 20 bit encoder). Scaling a 20 bit encoder to 18 bit gives the parameter “measuring units per revolution” the value $\mathbf{2^{18}}$ = 262144.

8.4 Linear Encoders

EnDat Linear encoders are supported with the preset function but currently no change of code sequence is supported. However access to all diagnostic objects as the operating time monitor, alarms, warnings and the encoder serial number is supported. For Linear encoders the measuring step in nm is output by the encoder.

9. Encoder setup example

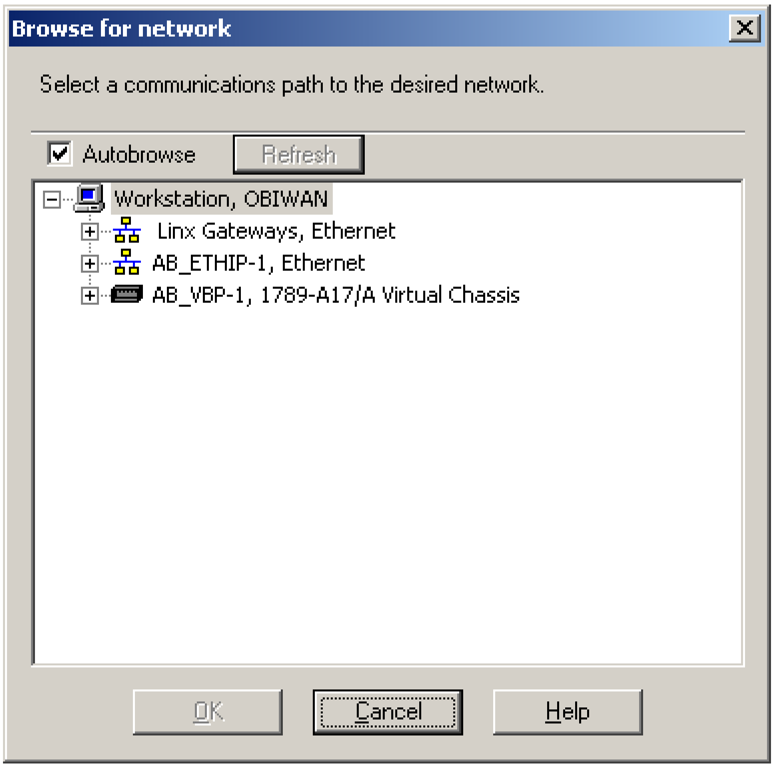
This example shows a simple setup of the Encoder Gateway for cyclic transmission of the Position value. Read section 2 of this Application note before proceeding.

NOTE! Since this is very tool dependent, the steps below must be adapted to the environment used. Also this only shows how the scanner part is made. Later you would probably use a PLC programming environment.

1. Set the physical address (Node Number) of the Encoder Gateway by Dip-switches, see section 5.4 for further information.

2. Power up the Encoder Gateway connected to the CAN bus.

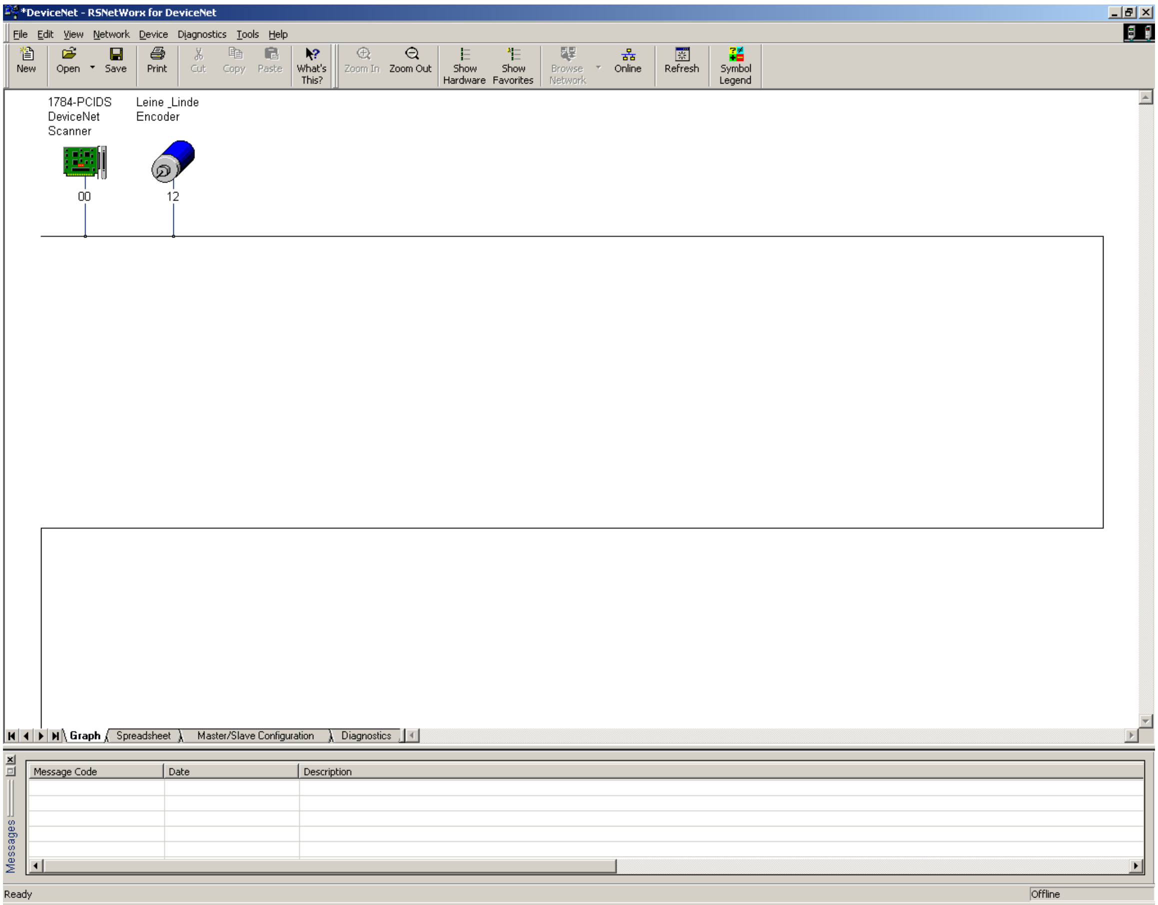
3. In RSNetworx, click the go on-line button to connect to the DeviceNet Network.



Browse to your DeviceNet network and click OK, to go on-line.

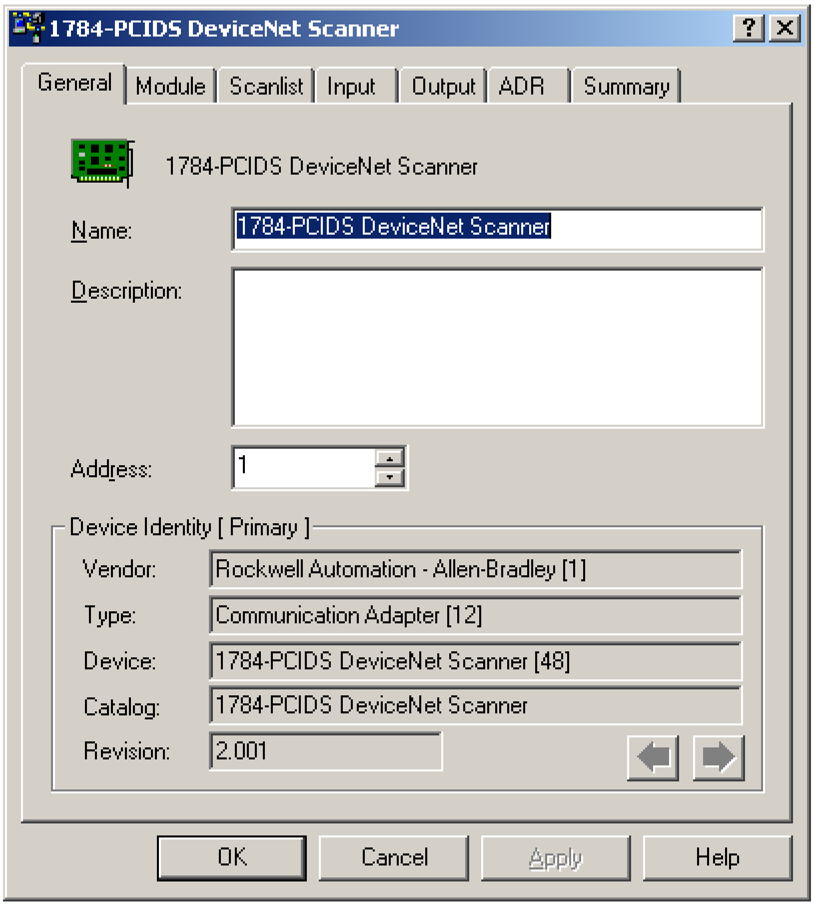
4. The Encoder Gateway will send a Boot-up message (Duplicate MAC message) when the module have detected the baud rate. If no other node responded on the MAC request, the Encoder Gateway is online and waiting for a master to connect. If another node responded on the MAC request, the Encoder Gateway will light the fail LEDs and go offline (repeat step 1 using a different address).

5. Scan the network.



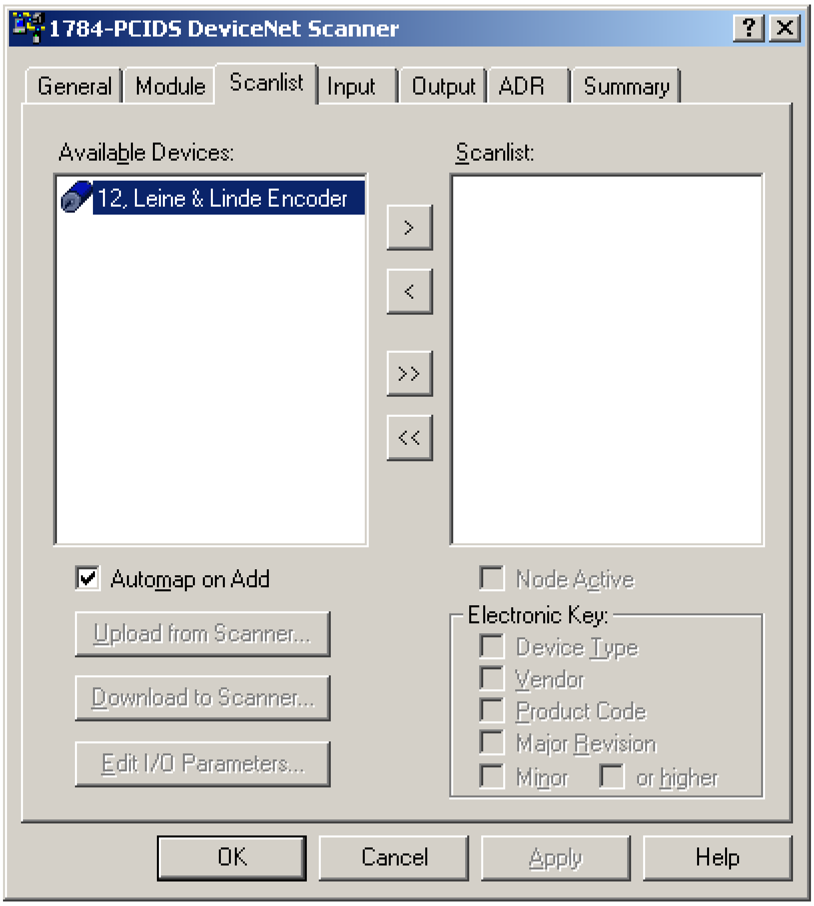
When all expected units on the Networks have appeared, double click the scanner icon to bring up its property page.

6. Select the Scanlist tab.

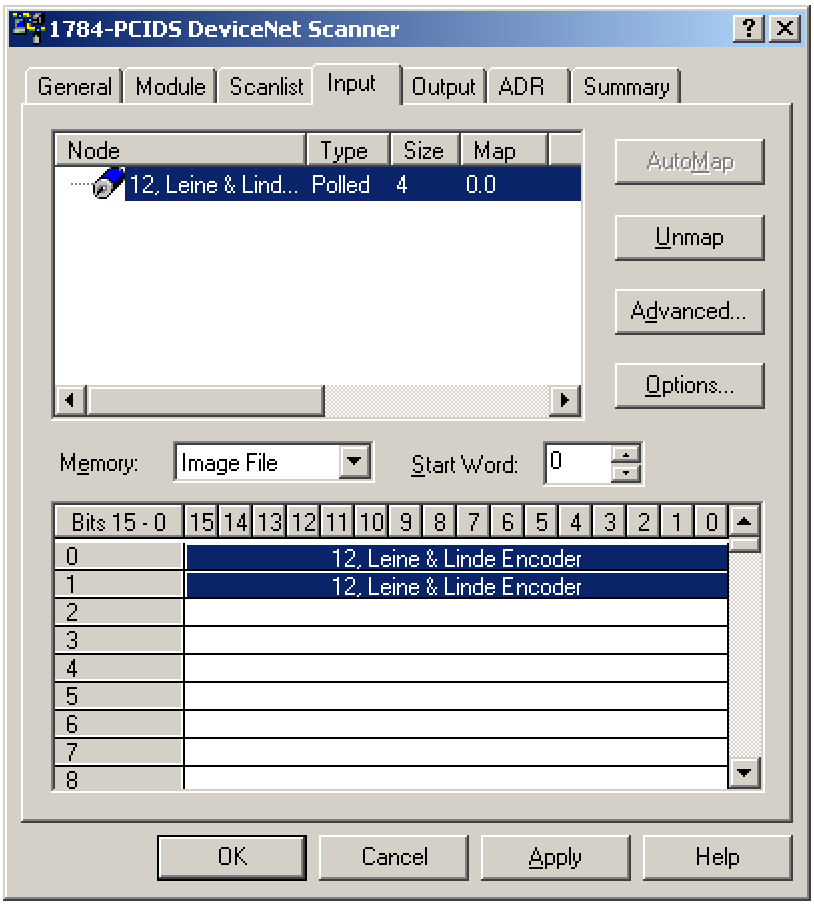


All units added to the scanlist will be polled by the Scanner/PLC.

7. Click the “>” button to add the encoder to the scan list.



8. Select the Input tab.



9. Confirm the mapping of the encoder value in the PLC Image file. Click OK to close the property page. And download the changes to the scanner device.

10. Now the scanner will gather all position information from all units in its scanlist and put it in the Image file.

Appendix A, History

Revision	Date	Changes
Rev. 0.9	03-06-25	First draft
Rev. 0.91	03-07-02	Second draft. Added info of implemented standard objects, better references, description of allocation modes and more LED info.
Rev. 0.92	03-07-07	Third draft. Added Connection class object description, EDS-file reference, more allocation mode info, position value fetch figure and more module LED info.
Rev. 0.92	03-08-14	Grammatical corrections.
Rev. 0.92	03-08-19	5.2, correction: non isolated bus interface.
Rev. 0.93	03-08-22	2.1, Added info regarding ENGUNITS.