DeviceNet configuration instructions for VAMP devices Introduction

1.1 Document overview

The purpose of this document is to describe the configuration and testing of the DeviceNet protocol in VAMP 50, VAMP 200 and VAMP 300 series devices.

In this document, the word “device” refers to VAMP protection relays, VAMP measuring units and similar types of equipment.

In order to configure a VAMP device, the Vampset relay setting and configuration tool is needed. Vampset can be downloaded from the website (http://www.schneider-electric.com/products/ww/en/2300-ied-user-software/2320-vamp-user-software/62050-vamp-software/?xtmc=vamp&xtcr=2).

The configurator of a device must be logged in with *Configurator* access level in order to be able to make any changes to protocol settings. This is done by entering the appropriate password when connecting to a device with Vampset. The default password for configurator access level is “2”.

This document assumes that the reader has some previous knowledge of the DeviceNet protocol.

Note: This document applies fully only to firmware versions v.10.134 and newer.

1.2 References

[1] Application Note DeviceNet and EtherNetIP Data Model, “DeviceNet and EtherNet/IP data model in VAMP devices”, 30.10.2014, V1.1

2 DeviceNet

This section will give a brief overview of the DeviceNet protocol in general, and in VAMP devices.

2.1 DeviceNet protocol overview

DeviceNet is the first adaptation of CIP, Common Industrial Protocol, which is an open standard, designed for vendor independent device interoperability that is strictly object oriented protocol. CIP can use different transport layers. In the case of DeviceNet, the CAN (Controller Area Network) protocol is used.

DeviceNet is a digital, multi-drop network that connects and serves as a communication network between industrial controllers and I/O devices. Each device and/or controller is a node on the network. DeviceNet is a producer-consumer network that supports multiple communication hierarchies and message prioritisation.

DeviceNet systems can be configured to operate in a master-slave or a distributed control architecture using peer-to-peer communication. DeviceNet also has the unique feature of delivering power on the network, which allows devices with limited power requirements to be powered directly from the network.

2.1.1 Messaging

DeviceNet supports two modes of messaging, unconnected and connected messaging.

Unconnected messaging refers to peer-to-peer communication, where opening and closing of connections is allowed via unconnected messaging. This is handled by the Unconnected Message Manager (UCMM).

Connected messaging, on the other hand, is dedicated to a particular purpose, such as frequent explicit message transactions or real-time I/O data transfers. Connection resources are reserved and configured using communications services available via the UCMM.

To ensure uniqueness of each connection ID, DeviceNet uses the 11-bit CAN identifier to define the connection ID and divides the CAN identifier into four groups.

There are two types of connections, explicit and implicit.

Explicit connections refer to request-response messages which are general purpose messages. In explicit connections, acknowledgments of the messages are used.

In *implicit* connections, only application data is contained within the messages. Implicit data may be polled, cyclic or COS (Change of State) messages. Acknowledgments of messages are not used in implicit connections.

2.1.2 Objects

Objects in CIP (and thereby DeviceNet) are defined by:

- A description – a description of an object being specified
- A class code (Class ID) – a hexadecimal identifier assigned to each CIP object.
- Attributes – data elements associated with the object.
- Common services – a list of the common services defined for the object.
- Object-specific Services – the full specifications of any services unique to the object.
- Connections – connections supported by the object.
- Behaviour – the relationship between attribute values and services.

The object model of CIP is shown in Figure 2.1.2-1 below. A description of the objects is given in

Table 2.1.2-1.

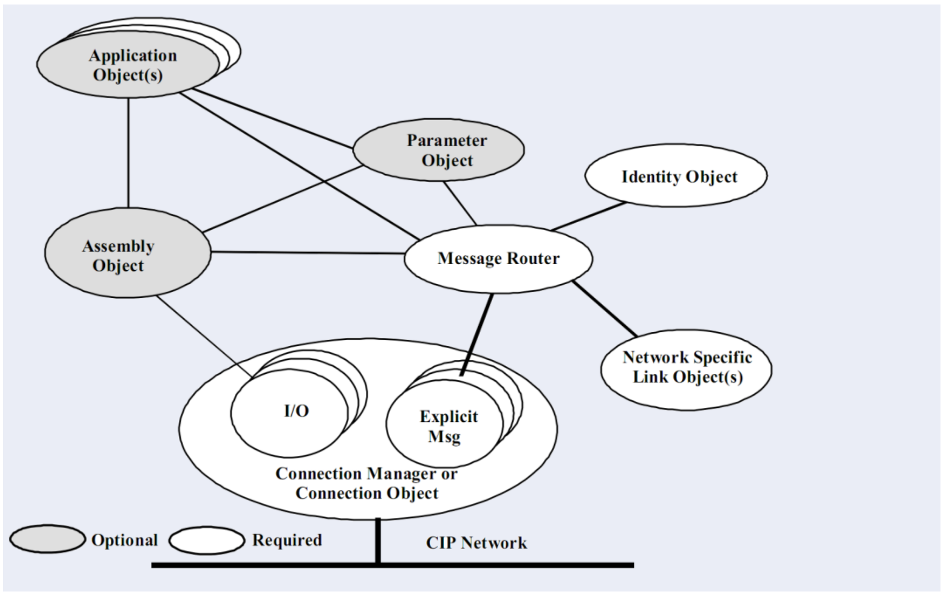


Figure 2.1.2-1: The CIP object model.

The CIP specification also includes an Object Library, which is a set of standardized objects that covers protocol objects such as the Identity Object, Assembly Object, etc. as well as Application Objects. The Object Library covers basic industrial automation building blocks and some more complex devices, including Digital Input, Digital Output, Analog Input, Analog Output, Position Sensor, Position Controller, AC/DC Drive, Overload, Softstart and Motion Axis.

Table 2.1.2-1: CIP object description.

Object	Description
Connection Object	The CIP Communication Object manages and provides the run-time exchange of messages
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. It routes explicit messages over requested paths.
Assembly Object	The Assembly Object binds attributes of multiple objects, which in allows data to or from each of these objects to be sent or received over a single connection (like a data set). Assembly objects can be used to group input data (producing instance of an Assembly Object – information transmitted to the network) or output data (consuming instance of an Assembly Object – information received from the network). I/O connections are established between Assembly Object instances of the devices – between inputs and outputs. Assembly object instances are accessible via explicit messaging.
Identity Object	Provides device identification and general information about the device, such as vendor identifier, product code, name, status, etc. The Identity Object shall be present in all CIP products.

Application Objects should be based on the standard objects from the Object Library if possible, if not, vendors can define their own, private (vendor specific) Application Objects. CIP specifies the Class ID ranges for that purpose. See Table 2.1.2-2.

Table 2.1.2-2: CIP Class ID ranges.

Range (hexadecimal)	Meaning	Quantity
0x00…0x63	Open	100
0x64…0xC7	Vendor specific	100
0xC8…0xEF	Reserved by CIP for future use	40
0xF0…0x2FF	Open	528
0x300…0x4FF	Vendor specific	512
0x500…0xFFFF	Reserved by CIP for future use	64256

2.1.3 Device profile

The series of application objects for a particular device type is known as the device profile. A large number of profiles for many device types have been defined. An example of a device profile is shown in Table 2.2.1-1.

2.1.4 Electronic Data Sheet (EDS)

DeviceNet vendors are required to provide some type of documentation specifying how their device is configured. The document may be a set of printed instructions or some electronic file. An electronic listing of the attributes that configure a DeviceNet device is usually provided by a vendor.

An Electronic Data Sheet (EDS) is a textural description of a device profile. An EDS specifies the accessible parameters and the content of Assemblies. The format of an EDS (file) is similar to Windows .INI files to ensure machine readability. EDS file mainly contains:

- Device's identity information – main configuration parameters described in section 3
- Parameters list – all of the attributes available in the data model; see section 5
- Assemblies info – current configuration of producing and consuming assemblies.

2.2 DeviceNet in VAMP devices

The DeviceNet protocol is available on VAMP devices with an optional external VSE 009 DeviceNet module. The protocol can be used to read/write data from/to a VAMP device using request-response communication and via cyclic I/O messages. Data to be transmitted is assigned to Assemblies. A VAMP device with the DeviceNet protocol selected on the Remote port serves as a *DeviceNet Adapter*, which means that it is not able to initiate communication with other devices on the network.

2.2.1 Objects and Messaging

The DeviceNet implementation on VAMP devices supports all required standard objects with their required attributes. There are also total of 10 application objects from which 8 are private. A list of VAMP device's objects and their classes is shown in Table 2.2.1-1.

Table 2.2.1-1: Device profile of a VAMP 257 protection relay in motor manager mode

Class	Object	Object Category	
0x01	Identity	Protocol	Standard
0x03	DeviceNet		
0x04	Assembly		
0x05	Connection		
0x29	Control Supervisor	Application	
0x2C	Overload		
0x64	Digital		Private (vendor specific)
0x65	Analog		
0x66	StgProtCurrent		
0x67	StgProtEF		
0x68	StgProtOther		
0x69	StgGeneral		
0x70	Analog2		
0x71	Special		

The DeviceNet implementation in VAMP devices supports two types of communication:

- Explicit Requests and Responses – used mainly for establishing I/O connections, but can also be used for one time requests to attributes of data model objects.
- I/O Messaging connections – used for very frequent exchange of process data. Polled, Change of State and Cyclic I/O Messaging types are supported.

2.2.2 Supported services

VAMP device support following services for objects:

- GAS = Get Attribute Single
- SAS = Set Attribute Single

GAS service is available for all attributes with the GET or GET | SET access type and the SAS service is available for all attributes with the GET | SET or SET access type

A list of VAMP device's services for objects is shown in the

Table 2.2.2-1: Supported services for objects in VAMP device

Class	Object	Supported services	
		Get	Set
0x01	Identity	GAS	-
0x02	Message Router	-	-
0x03	DeviceNet	GAS	SAS
0x04	Assembly	GAS	SAS
0x05	Connection	GAS	-
0x29	Control Supervisor	GAS	SAS
0x2C	Overload	GAS	SAS
0x64	Digital	GAS	SAS
0x65	Analog	GAS	SAS
0x66	StgProtCurrent	GAS	SAS
0x67	StgProtEF	GAS	SAS
0x68	StgProtOther	GAS	SAS
0x69	StgGeneral	GAS	SAS
0x70	Analog2	GAS	SAS
0x71	Special	GAS	SAS

2.2.3 I/O messaging assemblies

The DeviceNet implementation on VAMP devices includes total of two producing assemblies (Tx, Target → Originator) and two consuming assemblies (Rx, Target ← Originator); see Table 2.2.3-1.

Table 2.2.3-1: Available assemblies in VAMP devices

Instance no.	Type	Description
2	Producing	Static Basic Output Image
50	Consuming	Static Basic Input Image
100	Producing	Configurable Output Image (dynamic)
150	Consuming	Configurable Input Image (dynamic)

Assemblies have to be configured during the device setup. Configuring assemblies involve selecting the producing and consuming instances to be used.

If dynamic assemblies (instance numbers 100 & 150) are used it is also needed to configure the contents of both assembly. By default both assemblies are configured with one byte of data each. By default producing assembly is configured with 'Control Supervisor Object' / 'Faulted attribute' and consuming assembly with 'Control Supervisor Object' / 'FaultRst'.

2.2.4 Electronic Data Sheets

Every change to main configuration parameters or assemblies configuration requires a new EDS file to be generated (once all changes are made and the device is about to be used in the network).

Some of the configuration tools are capable of simplifying device configuration based on the EDS file. In the current implementation the EDS file can only be generated from Vampset tool – EDS file extraction over the EtherNet/IP network is not supported in VAMP device.

EDS file cannot be extracted from VAMP devices over the DeviceNet network, rather, it must be generated with Vampset. This operation is explained in section 0 of this document.

2.2.5 Events

VAMP device events are available under the following attributes of the VAMP Digital Object (0x64):

- Attribute 147 – Event Code (bits 0-5: code, bits 6-15: channel)
- Attribute 148 – Event Milliseconds And Seconds (bits 0-5: seconds, bits 6-15: milliseconds)
- Attribute 149 – Event Min And Hour (bits 0-7: hour, bits 8-15: minutes)
- Attribute 150 – Event Day And Month (bits 0-7: month, bits 8-15: day)
- Attribute 151 – Event Year.

Events are read starting from the oldest one in the Event Buffer on the VAMP device. Events are read sequentially, the next event is read when the previous one is acknowledged. Acknowledgement is done by setting attribute 152 of the VAMP Digital Object (0x64) – Event Ack. When all events have been read and the event buffer thus is empty, the attributes will contain zero-data (zeroes). This zero-data will automatically be replaced with the data of a new event when one is registered.

2.2.6 Fault codes

Table 2.2.6-1 below contains a translation of VAMP protection stages to DeviceNet Fault Codes.

Table 2.2.6-1: DeviceNet fault code to VAMP protection stage translation.

DeviceNet fault code	Description	VAMP protection stage	
20	CURRENT TRIP	Overcurrent Stage I> Overcurrent Stage I>> Overcurrent Stage I>>>	50/51 50/51 50/51
21	THERMAL OVERLOAD	Thermal Overload Stage T>	49
26	PHASE IMBALANCE	Unbalance Stage I2>	46
27	GROUND FAULT	Earth Fault Stage Io> Earth Fault Stage Io>> Earth Fault Stage Io>>> Earth Fault Stage Io>>>>	50N/51N 50N/51N 50N/51N 50N/51N
29	UNDERLOAD	Under Current Stage I<	37
31	STALL	Stall Protection Stage Ist>	48
51	UNDERVOLTAGE	U< U<< U<<<	27 27 27
52	OVERVOLTAGE	U> U>> U>>>	59 59 59
54	PHASE REVERSAL	Unbalance Stage I2>>	47
55	FREQUENCY	f< f<< f>< f>><<	81L 81L 81 81
73	START/HOURS EXCEEDED	Frequent Start Protection N>	66

3 Configuration

This section will explain how to configure a VAMP device to use the DeviceNet protocol.

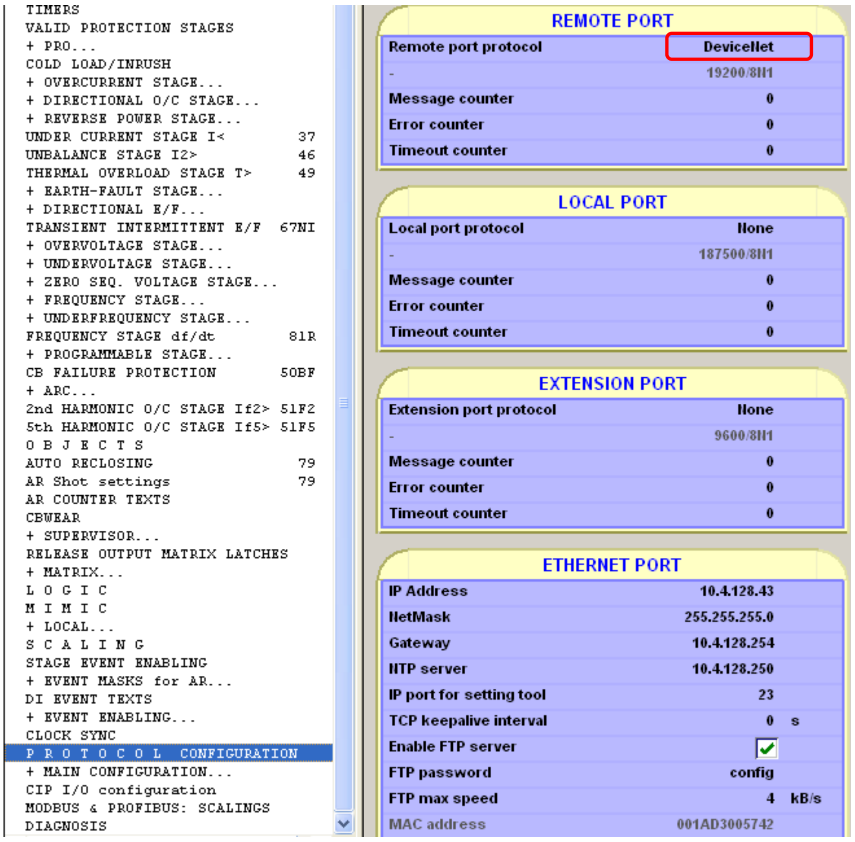
Note: The configuration and features may vary slightly between different VAMP devices and between different firmware versions.

3.1 General

First, Vampset must be connected to the device. This is typically done by connecting a USB to RS-232 converter to the PC and then connecting a VX003 cable from the converter to the front port of the device. Alternatively, an Ethernet connection can be used. See the device manual for more information on the different ports on the device.

The DeviceNet protocol is activated by setting it as the port protocol for a serial port on the device at hand. This setting can be found by navigating to the *PROTOCOL CONFIGURATION* menu in the Vampset Group List (the list to the left in the Vampset window). The protocol is set by clicking on the highlighted part and selecting “DeviceNet” in the drop-down list that will appear.

Figure 3.1-1 features an example of DeviceNet chosen as protocol on the Remote port of a VAMP device (the relevant portion of the screenshot is highlighted).



```
TIMERS
VALID PROTECTION STAGES
+ PRO...
COLD LOAD/INRUSH
+ OVERCURRENT STAGE...
+ DIRECTIONAL O/C STAGE...
+ REVERSE POWER STAGE...
UNDER CURRENT STAGE I<        37
UNBALANCE STAGE I2>           46
THERMAL OVERLOAD STAGE T>     49
+ EARTH-FAULT STAGE...
+ DIRECTIONAL E/F...
TRANSIENT INTERMITTENT E/F   67NI
+ OVERVOLTAGE STAGE...
+ UNDERVOLTAGE STAGE...
+ ZERO SEQ. VOLTAGE STAGE...
+ FREQUENCY STAGE...
+ UNDERFREQUENCY STAGE...
FREQUENCY STAGE df/dt         81R
+ PROGRAMMABLE STAGE...
CB FAILURE PROTECTION        50BF
+ ARC...
2nd HARMONIC O/C STAGE If2>  51F2
5th HARMONIC O/C STAGE If5>  51F5
O B J E C T S
AUTO RECLOSING                79
AR Shot settings              79
AR COUNTER TEXTS
CBWEAR
+ SUPERVISOR...
RELEASE OUTPUT MATRIX LATCHES
+ MATRIX...
L O G I C
M I M I C
+ LOCAL...
S C A L I N G
STAGE EVENT ENABLING
+ EVENT MASKS for AR...
DI EVENT TEXTS
+ EVENT ENABLING...
CLOCK SYNC
P R O T O C O L  CONFIGURATION
+ MAIN CONFIGURATION...
CIP I/O configuration
MODBUS & PROFIBUS: SCALINGS
DIAGNOSIS
```

REMOTE PORT

Remote port protocol	DeviceNet
-	19200/8N1
Message counter	0
Error counter	0
Timeout counter	0

LOCAL PORT

Local port protocol	None
-	187500/8N1
Message counter	0
Error counter	0
Timeout counter	0

EXTENSION PORT

Extension port protocol	None
-	9600/8N1
Message counter	0
Error counter	0
Timeout counter	0

ETHERNET PORT

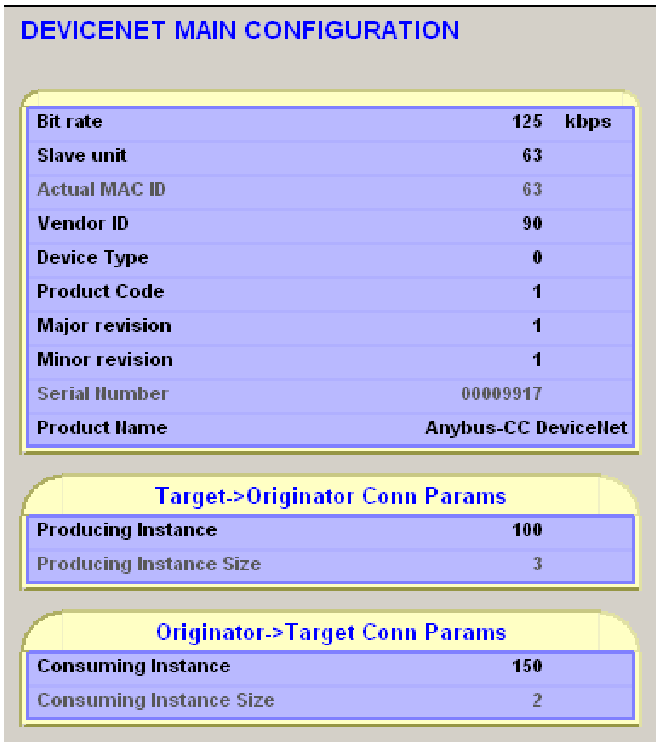
IP Address	10.4.128.43	
NetMask	255.255.255.0	
Gateway	10.4.128.254	
NTP server	10.4.128.250	
IP port for setting tool	23	
TCP keepalive interval	0	s
Enable FTP server	✓	
FTP password	config	
FTP max speed	4	kB/s
MAC address	001AD3005742	

Figure 3.1-1: Protocol configuration menu in Vampset.

Note: Setting a protocol on any port will require a reboot of the device before the changes will take effect. Vampset will prompt for a device reboot when a change has been made.

3.2 DeviceNet main configuration

The configuration of the DeviceNet settings is done in the *DEVICENET MAIN CONFIGURATION* menu in Vampset. The available settings are shown in Figure 3.2-1 and explained in Table 3.2-1.


DEVICENET MAIN CONFIGURATION

Bit rate	125	kbps
Slave unit	63	
Actual MAC ID	63	
Vendor ID	90	
Device Type	0	
Product Code	1	
Major revision	1	
Minor revision	1	
Serial Number	00009917	
Product Name	Anybus-CC DeviceNet	

Target->Originator Conn Params

Producing Instance	100
Producing Instance Size	3

Originator->Target Conn Params

Consuming Instance	150
Consuming Instance Size	2

Figure 3.2-1: DeviceNet main configuration menu in Vampset.

Table 3.2-1: Description of the DeviceNet main configuration parameters.

Parameter	Range	Description
Bit rate (kbps)	125, (default) 250, 500	The bit rate of the DeviceNet network. More detailed information, see 3.2.1
Slave unit	0…63	The slave unit address of the device on the network. More detailed information, see 3.2.2
Actual MAC ID	0…63	Displays the current MAC address of the device which uniquely identifies the device in the network (not editable)
Vendor ID	1…65535	Identification of a vendor by number (not editable)
Device Type	0…65535	Identification of a general type of product (not editable)
Product Code	1…65535	Identification of a particular product of an individual vendor (not editable)

Major revision	1…127	Major revision number of the item the Identity Object represents (not editable)
Minor revision	1…255	Minor revision number of the item that the Identity Object represents (not editable)
Serial Number	0…4294967295	Serial number of the device (not editable)
Product name	32 character string	Human readable identification (not editable)
I/O assembly instances in use	2+50, (default) 100+150, 101+151	Instance numbers of producing and consuming assemblies being used
Producing Instance	[producing instance number]	Instance number of the producing assembly (not directly editable) *
Producing Instance Size	[byte]	The size of the producing assembly (not directly editable) **
Consuming Instance	[consuming instance number]	Instance number of the consuming assembly (not directly editable) *
Consuming Instance Size	[byte]	The size of the consuming assembly (not directly editable) **

* Automatically updated according to the value of ‘I/O assembly instances in use’ parameter.

** Automatically updated as the assemblies are configured.

3.2.1 Bit rate

This is a baud rate of the DeviceNet network. For the device to work properly in the network, its Bit rate parameter must match baud rate used on the network. The Bit rate parameter can be set to one of three possible values: 125, 250 or 500 kbps. Default value of this parameter is 125 kbps as required by the DeviceNet specification.

3.2.2 Slave unit

This is a MAC ID of the device. It uniquely identifies the device in the DeviceNet network. Main range of this parameter is 0 to 63 and it has to be set to a value that will not cause a duplicate MAC ID problem. There is also a possibility to set this parameter to -1 (only available via GETSET) to allow MAC ID setting from the network. Default value of this parameter is 63.

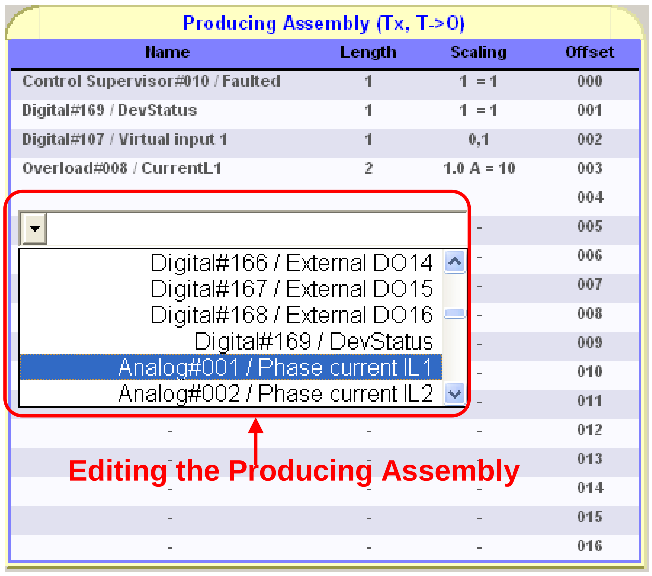
3.3 Data point configuration

Available data items, that is, the contents of the Producing Assembly and the Consuming Assembly can be viewed / configured in the following Vampset menus:

- DeviceNet I/O 2+50 (static)
- DeviceNet I/O 100+150 (dyn.)

Assembly 2+50 is static, meaning user cannot make changes to the contents of assembly.

Assembly 100+150 is dynamic, meaning user can select data items to the assembly by clicking on a row and selecting a desired data point. An example of this is shown in Figure 3.3-1 below. The attributes of data items are described in Table 3.2.2-1.



Producing Assembly (Tx, T->O)

Name	Length	Scaling	Offset
Control Supervisor#010 / Faulted	1	1 = 1	000
Digital#169 / DevStatus	1	1 = 1	001
Digital#107 / Virtual input 1	1	0,1	002
Overload#008 / CurrentL1	2	1.0 A = 10	003
			004
		-	005
		-	006
		-	007
		-	008
		-	009
		-	010
		-	011
-	-	-	012
-	-	-	013
-	-	-	014
-	-	-	015
-	-	-	016

Digital#166 / External DO14
Digital#167 / External DO15
Digital#168 / External DO16
Digital#169 / DevStatus
Analog#001 / Phase current IL1
Analog#002 / Phase current IL2

Figure 3.3-1: Configuration of the DeviceNet Producing Assembly in Vampset.

Table 3.2.2-1: Description of Assembly configuration table contents.

Attribute	Description
Name	Type and name of the data item
Length	Length of the data item in bytes
Scaling	The scaling used for the data item
Offset	The offset of the data item in the assembly

Note:

1. Making changes to the Assemblies will require a device reboot for the changes to take effect.
2. A list of the set of available data items in Ethernet/IP is available in a different document on the VAMP website.

3.4 Generating an EDS file with Vampset

An EDS file can be generated with Vampset. This is done by navigating to "*Communication*" → "*Get DeviceNet EDS…*", see Figure 3.4-1. Selecting this option will generate the EDS file and bring up a file browser window asking where to save the generated file, see Figure 3.4-2**Error! Reference source not found.**. After clicking *Save* the generated EDS file will be stored at the selected location.

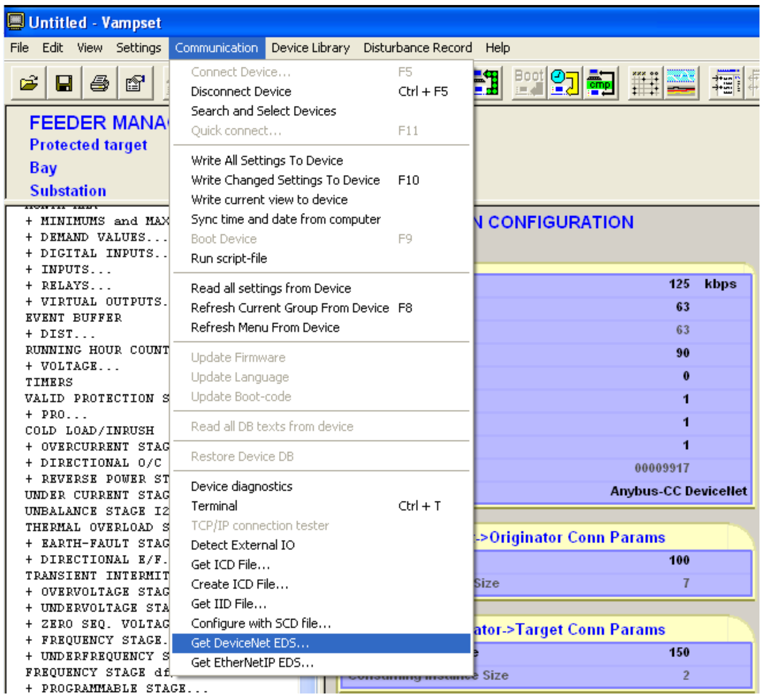


Figure 3.4-1: Creating an EDS file with Vampset.

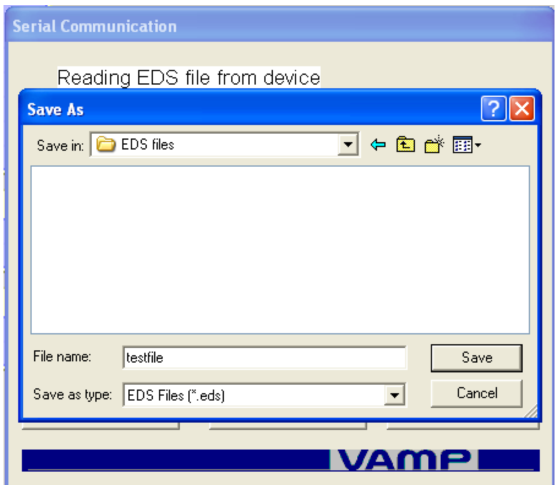


Figure 3.4-2: Saving the EDS file.

4 Testing with DeviceNet Master Simulator

This section will describe how to perform some test that a configured device is working properly using the *DeviceNet Master Simulator* software.

4.1 General

The DeviceNet Master Simulator is a piece of made by HMS, a Swedish company, and is available for download on the Internet (not on the VAMP website). It can be used to check that the communication to and from a VAMP device is working properly.

The following is needed to connect the VAMP device to a PC:

1. A PC with a free USB port.
2. A VAMP device.
3. A VAMP VSE 009 DeviceNet module.
4. A DeviceNet interface circuitry-Dongle.
5. A DeviceNet cable with data and power lines.
6. A 24 V power supply.

The test system setup is schematically shown in Figure 4.1-1.

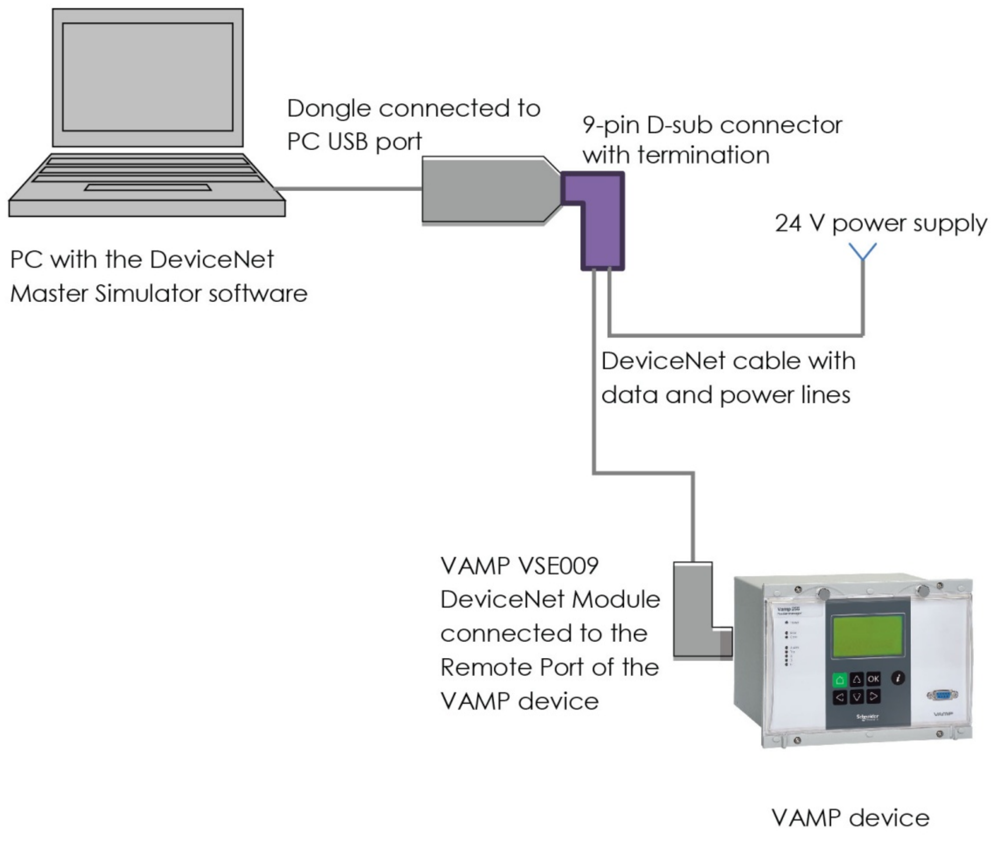
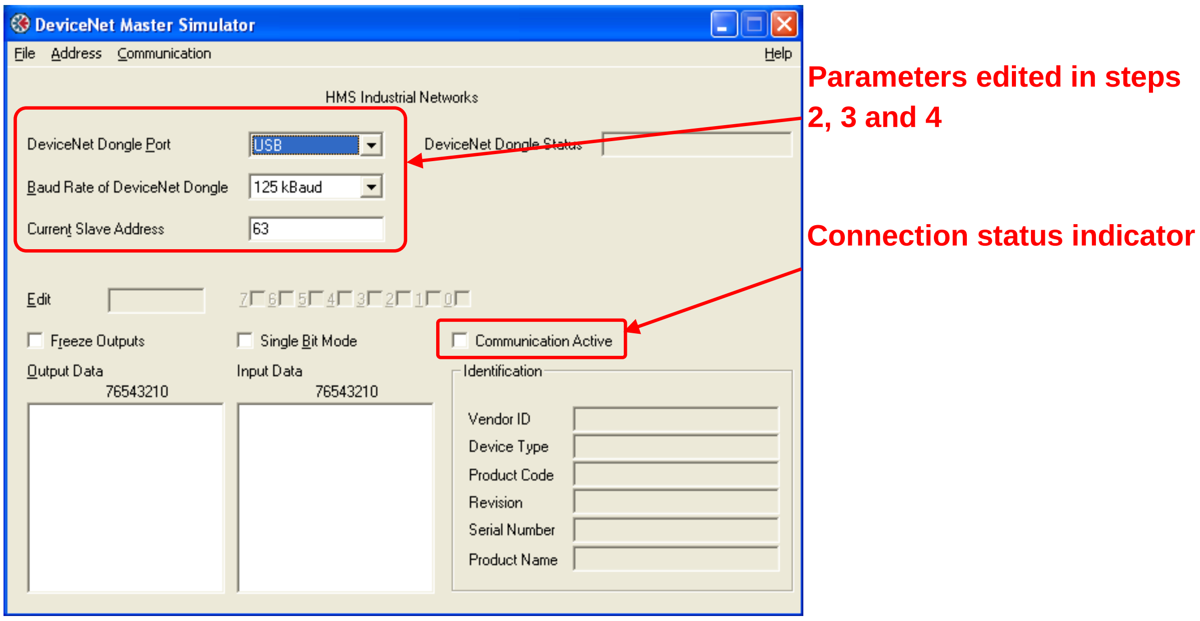

Dongle connected to PC USB port
9-pin D-sub connector with termination
24 V power supply
PC with the DeviceNet Master Simulator software
DeviceNet cable with data and power lines
VAMP VSE009 DeviceNet Module connected to the Remote Port of the VAMP device
VAMP device

Figure 4.1-1 Test system setup

4.2 Connecting to the device with DeviceNet Master Simulator

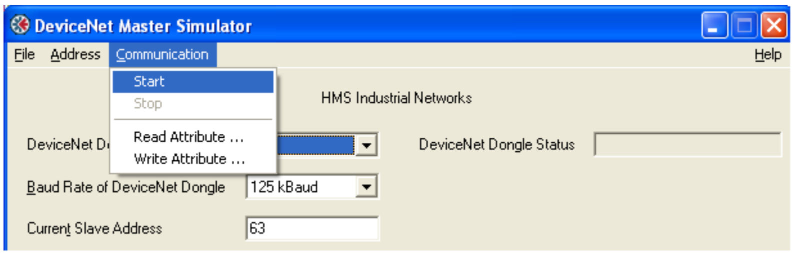
The following steps explain how to connect to the device:

1. Start the application on the PC. A screenshot of the main window is shown in Figure 4.2-1.
2. Set the *DeviceNet Dongle Port* to “USB”.
3. Set the *Baud Rate of DeviceNet Dongle* to the same settings as on the VAMP device.
4. Set the *Current Slave Address* to the same value as the *Slave unit* is set to on the device.
5. Open the *Communication* menu up in the menu bar and select *Start.*
6. A warning dialog, “*Warning! Outputs may be modified!*”, will be shown. Dismiss the dialog by pressing *OK.* See Figure 4.2-2**Error! Reference source not found.** and Figure 4.2-3**Error! Reference source not found.** for screenshots of this.
7. A successful connection will be indicated by a check-mark in the *Communication Active* checkbox smaller highlight in Figure 4.2-1.



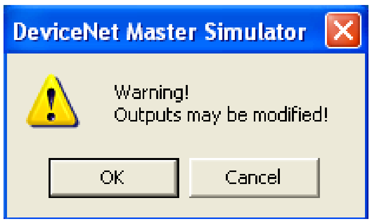
DeviceNet Dongle Port: USB
Baud Rate of DeviceNet Dongle: 125 kBaud
Current Slave Address: 63

Figure 4.2-1: The DeviceNet Master Simulator user interface.



Baud Rate of DeviceNet Dongle: 125 kBaud
Current Slave Address: 63

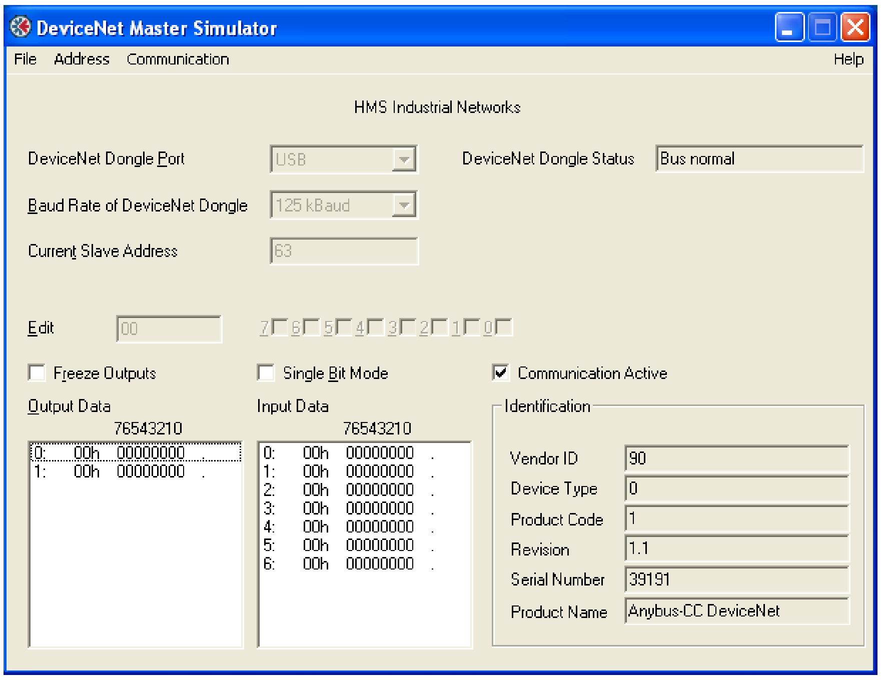
Figure 4.2-2: Connecting with DeviceNet Master Simulator to a device.



Warning!
Outputs may be modified!

Figure 4.2-3: Warning dialog shown when connecting.

Figure 4.2-4 shows a screenshot of DeviceNet Master Simulator connected to a device.



DeviceNet Dongle Port: USB
DeviceNet Dongle Status: Bus normal
Baud Rate of DeviceNet Dongle: 125 kBaud
Current Slave Address: 63
Edit: 00 7 6 5 4 3 2 1 0

☐ Freeze Outputs ☐ Single Bit Mode ☑ Communication Active

Output Data
```
          76543210
0:   00h  00000000  .
1:   00h  00000000  .
```

Input Data
```
          76543210
0:   00h  00000000  .
1:   00h  00000000  .
2:   00h  00000000  .
3:   00h  00000000  .
4:   00h  00000000  .
5:   00h  00000000  .
6:   00h  00000000  .
```

Identification

Vendor ID	90
Device Type	0
Product Code	1
Revision	1.1
Serial Number	39191
Product Name	Anybus-CC DeviceNet

Figure 4.2-4: DeviceNet Master Simulator when connected.

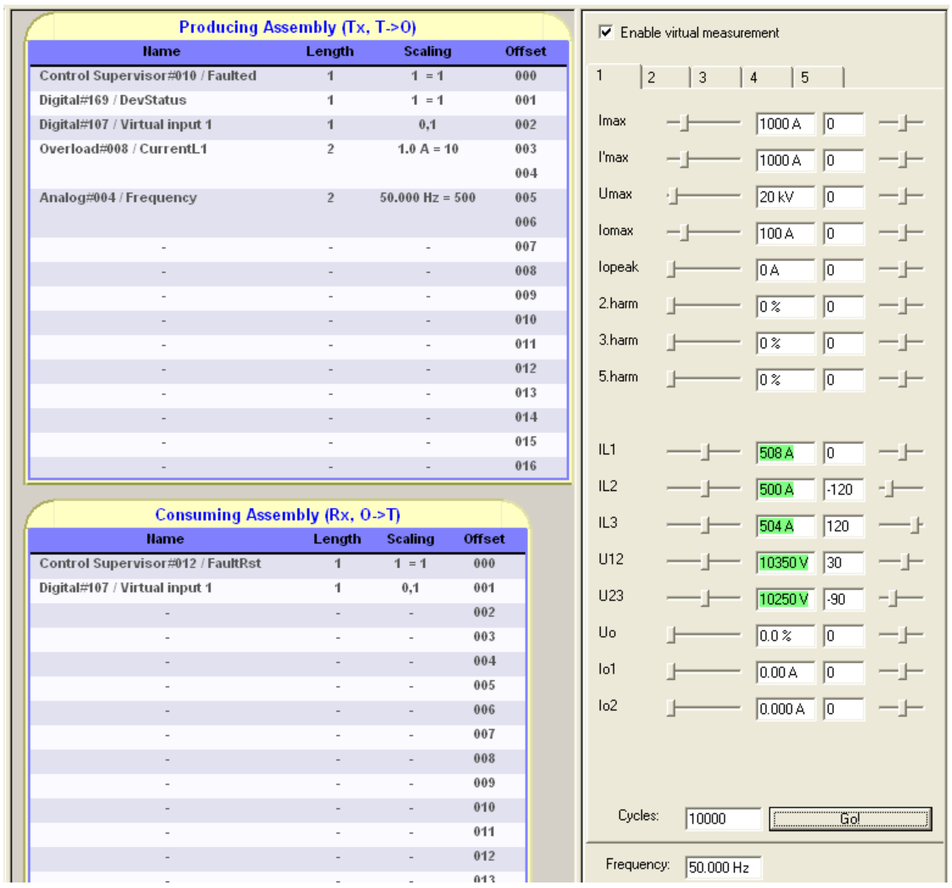
4.3 Reading data

The data points defined in the Producing Assembly of the device are received by the DeviceNet Master Simulator and shown in the *Input Data* field. The data defined in the Consuming Assembly, (the data sent from the master), of the device are shown in the *Output Data* field.

An example of how to read some particular data follows:

Some virtual measurements are sent to the device with Vampset. These, as well as the Assembly configurations are shown in Figure 4.3-1.

The values of Virtual Input 1, Line 1 Current and Frequency are defined in the Producing Assembly at Offsets 2, 3-4 and 5-6, respectively, and can be read from the *Input Data* field in DeviceNet Master Simulator, see Figure 4.3-2.



Producing Assembly (Tx, T->O)

Name	Length	Scaling	Offset
Control Supervisor#010 / Faulted	1	1 = 1	000
Digital#169 / DevStatus	1	1 = 1	001
Digital#107 / Virtual input 1	1	0,1	002
Overload#008 / CurrentL1	2	1.0 A = 10	003
			004
Analog#004 / Frequency	2	50.000 Hz = 500	005
			006
-	-	-	007
-	-	-	008
-	-	-	009
-	-	-	010
-	-	-	011
-	-	-	012
-	-	-	013
-	-	-	014
-	-	-	015
-	-	-	016

Consuming Assembly (Rx, O->T)

Name	Length	Scaling	Offset
Control Supervisor#012 / FaultRst	1	1 = 1	000
Digital#107 / Virtual input 1	1	0,1	001
-	-	-	002
-	-	-	003
-	-	-	004
-	-	-	005
-	-	-	006
-	-	-	007
-	-	-	008
-	-	-	009
-	-	-	010
-	-	-	011
-	-	-	012
-	-	-	013

☑ Enable virtual measurement

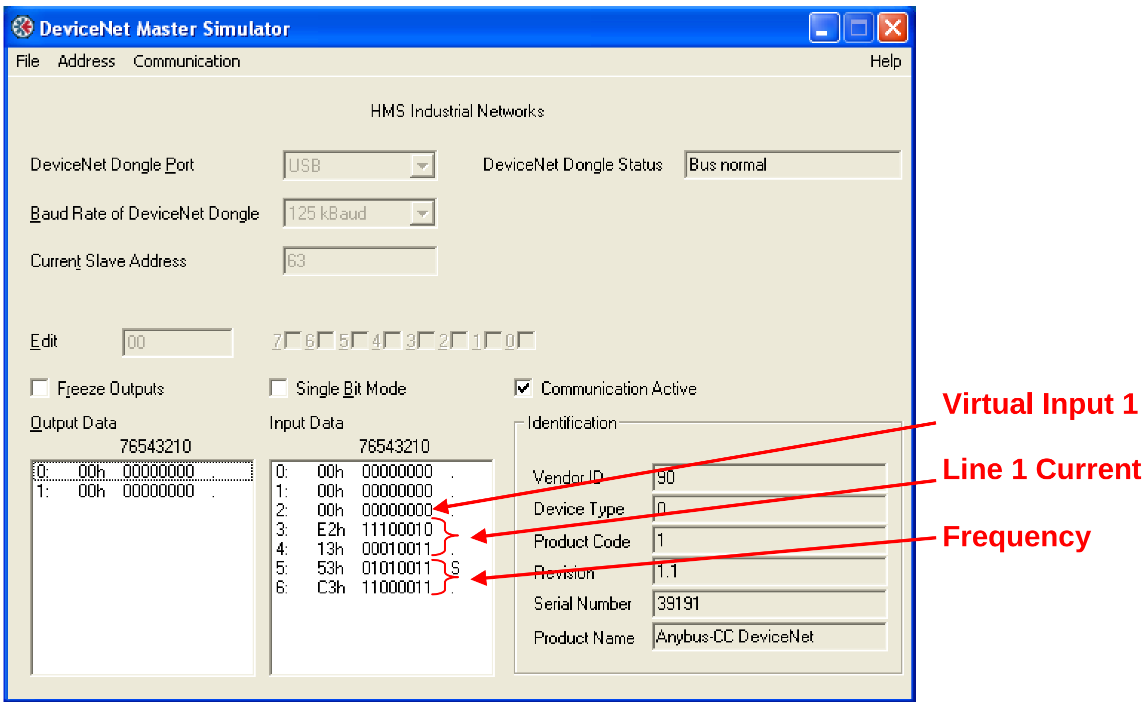
Tabs: 1 | 2 | 3 | 4 | 5

Imax	1000 A	0
I'max	1000 A	0
Umax	20 kV	0
Iomax	100 A	0
Iopeak	0 A	0
2.harm	0 %	0
3.harm	0 %	0
5.harm	0 %	0
IL1	508 A	0
IL2	500 A	-120
IL3	504 A	120
U12	10350 V	30
U23	10250 V	-90
Uo	0.0 %	0
Io1	0.00 A	0
Io2	0.000 A	0

Cycles: 10000 Go!

Frequency: 50.000 Hz

Figure 4.3-1: Data configuration and virtual measurements in Vampset.



HMS Industrial Networks

DeviceNet Dongle Port: USB
DeviceNet Dongle Status: Bus normal
Baud Rate of DeviceNet Dongle: 125 kBaud
Current Slave Address: 63

Edit: 00 7☐ 6☐ 5☐ 4☐ 3☐ 2☐ 1☐ 0☐

☐ Freeze Outputs ☐ Single Bit Mode ☑ Communication Active

Output Data
```
         76543210
0:   00h 00000000 .
1:   00h 00000000 .
```

Input Data
```
         76543210
0:   00h 00000000 .
1:   00h 00000000 .
2:   00h 00000000 .
3:   E2h 11100010 .
4:   13h 00010011 .
5:   53h 01010011 S
6:   C3h 11000011 .
```

Identification

Vendor ID	90
Device Type	0
Product Code	1
Revision	1.1
Serial Number	39191
Product Name	Anybus-CC DeviceNet

Figure 4.3-2: DeviceNet Master Simulator when receiving data.

Now, the received data needs to be interpreted:

1. The value of Virtual Input 1 is zero (00h as hexadecimal or 00000000 as binary), which can be seen at offset 2 in the Input Data field.

2. The value of Line 1 Current is divided into two bytes, at offsets 3 and 4. These are in the order "low byte, high byte", so the value should be read as 13E2h which has the decimal value of 5090.
   This is in accordance with the scaling shown for the value in Vampset: "1.0 A = 10" represents 509.0 A, which corresponds to the virtual value, 508 A sent from Vampset.

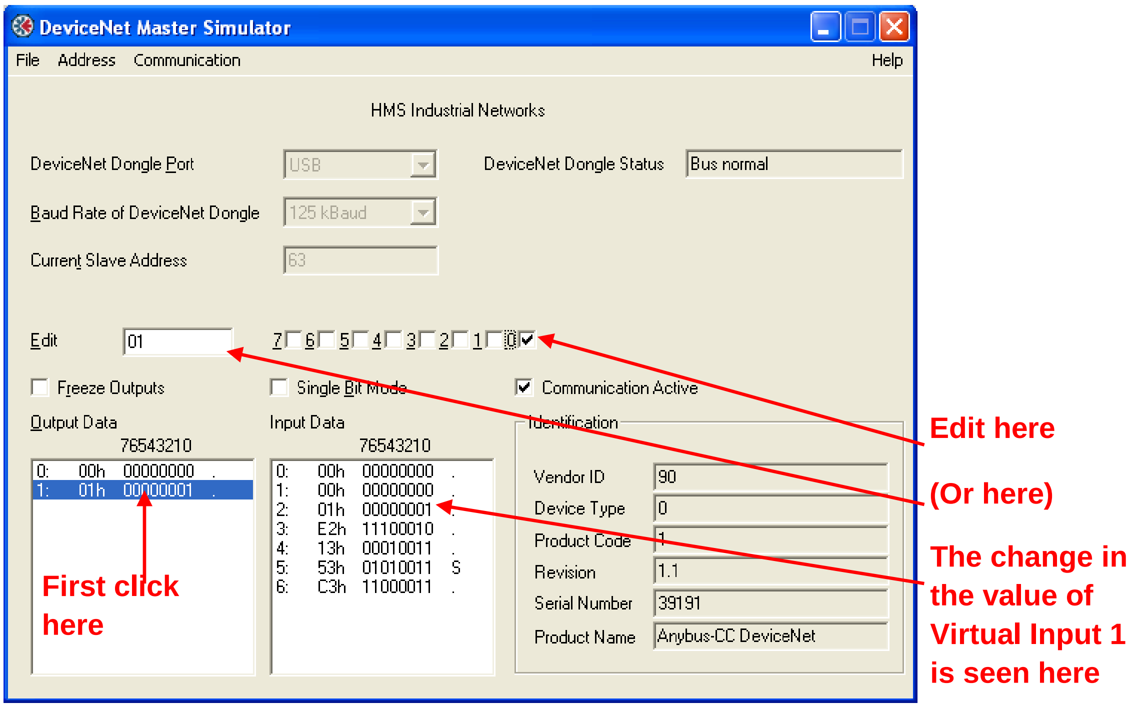
3. The value of the frequency is found at offsets 5 and 6 in the Input Data field, again "low byte, high byte", so the value is C353h = 50003_{10}, which is the internal representation of the frequency in the device. Not according to the scaling given in the data configuration table in Vampset.

4.4 Writing data

This section will feature an example of writing to Virtual Input 1.

Virtual Input 1 has been defined at offset 1 in the Consuming Assembly of the VAMP device (Figure 4.3-1), and is thus written to by editing the value at offset 1 in the Output Data field in DeviceNet Master Simulator. This is done by clicking on the item in the Output Data list and checking the box for bit0 (alternatively by writing a "1" in the field next to the *Edit* label. A screenshot of this is shown in Figure 4.4-1.

Since Virtual Input 1 was also defined in the Producing Assembly at index 2 (Figure 4.3-1), the current value of it can be seen at offset 2 in the Input Data field in DeviceNet Master Simulator.


DeviceNet Master Simulator
File Address Communication Help
HMS Industrial Networks
Edit here
(Or here)
The change in the value of Virtual Input 1 is seen here
First click here

DeviceNet Dongle Port	USB
DeviceNet Dongle Status	Bus normal
Baud Rate of DeviceNet Dongle	125 kBaud
Current Slave Address	63
Edit	01

7☐ 6☐ 5☐ 4☐ 3☐ 2☐ 1☐ 0☑

☐ Freeze Outputs ☐ Single Bit Mode ☑ Communication Active

```
Output Data
       76543210
0:  00h 00000000 .
1:  01h 00000001 .
```

```
Input Data
       76543210
0:  00h 00000000 .
1:  00h 00000000 .
2:  01h 00000001 .
3:  E2h 11100010 .
4:  13h 00010011 .
5:  53h 01010011 S
6:  C3h 11000011 .
```

Identification	
Vendor ID	90
Device Type	0
Product Code	1
Revision	1.1
Serial Number	39191
Product Name	Anybus-CC DeviceNet

Figure 4.4-1: Write to Virtual Input 1.

5 Data model

Please refer to [1]

Schneider Electric

35 rue Joseph Monier

92506 Rueil-Malmaison

FRANCE

Phone: +33 (0) 1 41 29 70 00

Fax: +33 (0) 1 41 29 71 00

www.schneider-electric.com

Publishing: 11/2014