Application Note

Anybus Communicator CAN communicating with SEW Drives using MOVILINK Profile

Contents

1 Document Revision History

Version	Description	Date	Author
0.90	First draft	2011-10-04	KJO
0.91	New template	2011-10-13	KAD
1.00	Added information, pictures and some modifications	2012-03-30	JHN
1.01	Added information and changed layout	2012-04-20	JHN
1.02	Some changes	2012-05-09	MSt
1.03	New screenshots, some minor modifications	2012-05-22	JHN
1.04	Minor modifications and added information to appendix	2012-05-25	JHN
1.05	Rewrite for ABC-CAN Modbus-TCP; included configurations for eight drives; screenshots of newest ACM version; some more modifications	2013-03-07	ALK

2 Overview

The Anybus Communicator CAN can be configured to communicate with all SEW drives talking the MOVILINK profile via CAN, like the MOVIDRIVE or MOVITRAC series of drives. This enables connectivity with the networking protocols supported by the Anybus Communicator CAN. These are:

Fieldbus Network	Industrial Ethernet
CANopen	EtherCAT
CC-Link	EtherNet/IP
ControlNet	Modbus-TCP
DeviceNet	Profinet RT
Modbus-RTU	Profinet IRT
Profibus	

For this application note a Modbus-TCP slave to CAN gateway will be used.

This document is available from HMS with sample configuration files for Modbus-TCP. If these files are not included, please find them at the HMS website (http://www.anybus.com) or contact HMS support (by e-mail: support@hms.se).

For information on how to commission an Anybus Communicator Modbus-TCP server with a Modbus-TCP client please download the application note “How to configure an Anybus Modbus-TCP slave module with Unity Pro L” (http://www.anybus.com/support/support.asp?PID=480&ProductType=Anybus Communicator CAN) and read the included PDF file.

It is assumed that the reader is familiar with industrial communication and the Anybus Communicator CAN.

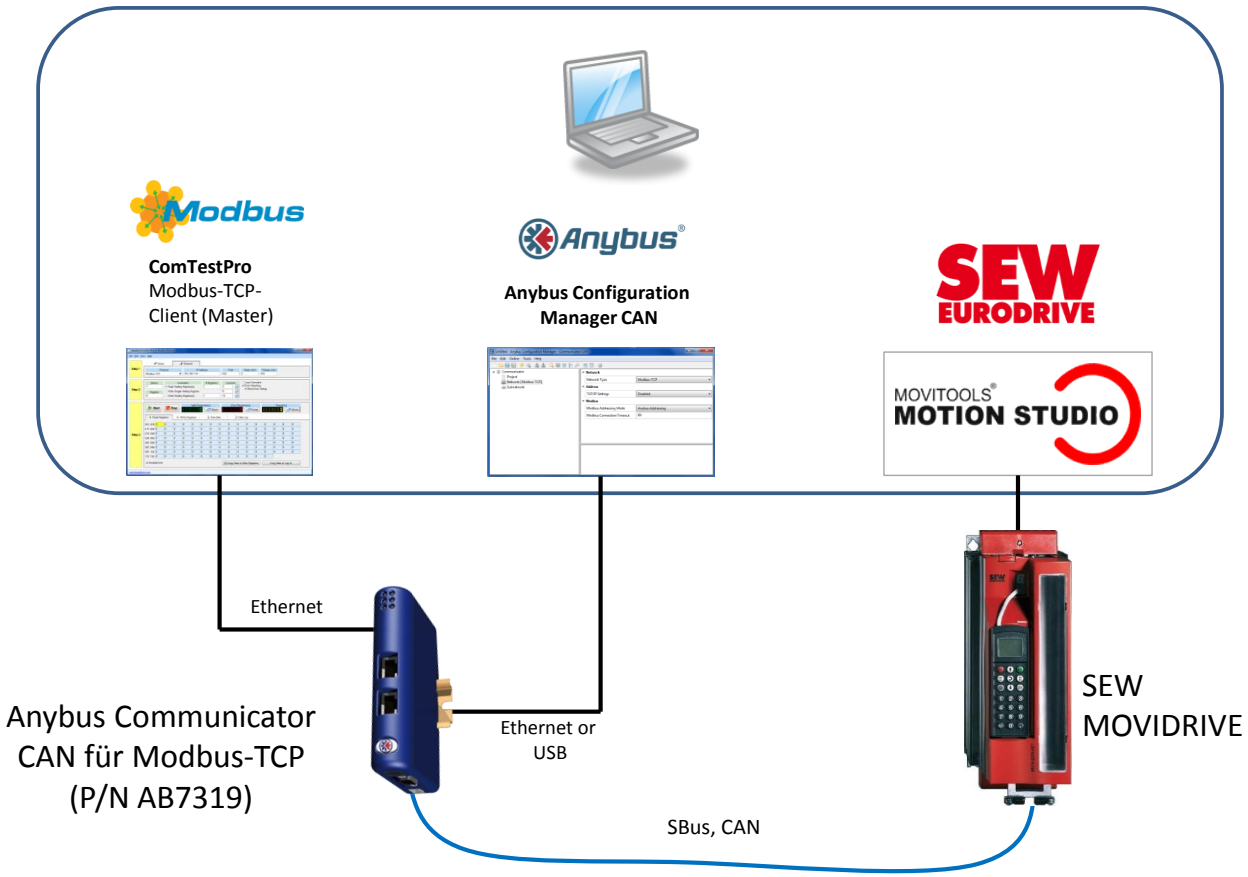
For more in depth information explaining the CAN interface of SEW drives please refer to the file CAN_interfaces_of_SEW_Drives.pdf file and the documentation available via SEW’s webpage.

3 Physical connections

Connect your PC via a switch with the Modbus-TCP fieldbus connector of the Anybus Communicator CAN (Ethernet). This suits both for the configuration with the Anybus Configuration Manager CAN and for testing with the Modbus-TCP master simulator software ComTestPro. For configuration of Anybus Communicator CAN units that are not Ethernet based a USB cable may alternatively be used.

Connect the subnet port of the Anybus Communicator CAN to the CAN network.

Connect the SEW drive to the CAN network.



3.1 Communication settings used in this configuration

Anybus Communicator CAN:

IP address 192.168.1.58 (your PC has to be in the same subnet)
CAN Baud Rate: 500 kbit/s
CAN Identifier: 11 bit

SEW drive e.g. MOVIDRIVE MDx60/61B (Parameter-Group 88x/89x):

SBus Address: 2
CAN Baud Rate: 500 kbit/s
CAN profile: MOVILINK

3.2 Anybus Configuration Manager CAN

System requirement: Windows 2000/NT/XP/7

Install from CD or download from HMS website (http://www.anybus.com).

3.3 Modbus-TCP Master Simulator ComTestPro

System requirement: Windows XP/7

Download from BaseBlock website (http://www.baseblock.com)

4 Configuration

In this application note four different communication setups between SEW´s MOVIDRIVE (using the MOVILINK protocol) and the Anybus Communicator CAN will be shown.

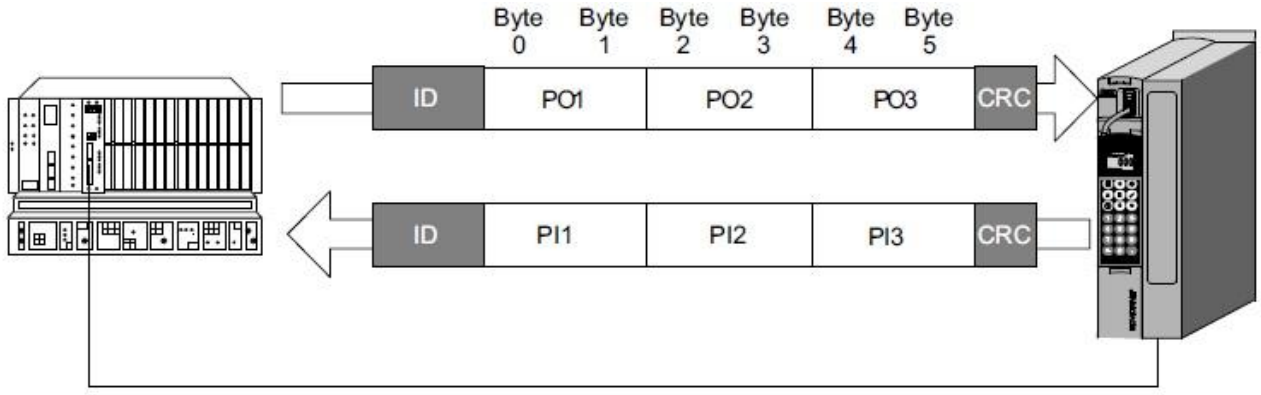
1. Exchange **three words** as process data cyclically **for one drive**. See Configuration 1: One Drive Exchanging Process Data Cyclically.
2. Exchange **three words** as process data cyclically **for eight drives**, respectively. See Configuration 2: Eight Drives exchanging Process Data Cyclically.
3. Exchange **ten fragmented process data words** cyclically for **one drive**. See Configuration 3: One Drive Exchanging Ten Process Data Words using Fragmentation.
4. Exchange **ten fragmented process data words** cyclically for **eight drives**, respectively. See Configuration 4: Eight Drives Each Exchanging Ten Process Data Words using Fragmentation.

Reading and writing of the drives parameter is possible.

4.1 Configuration 1: One Drive Exchanging Process Data Cyclically

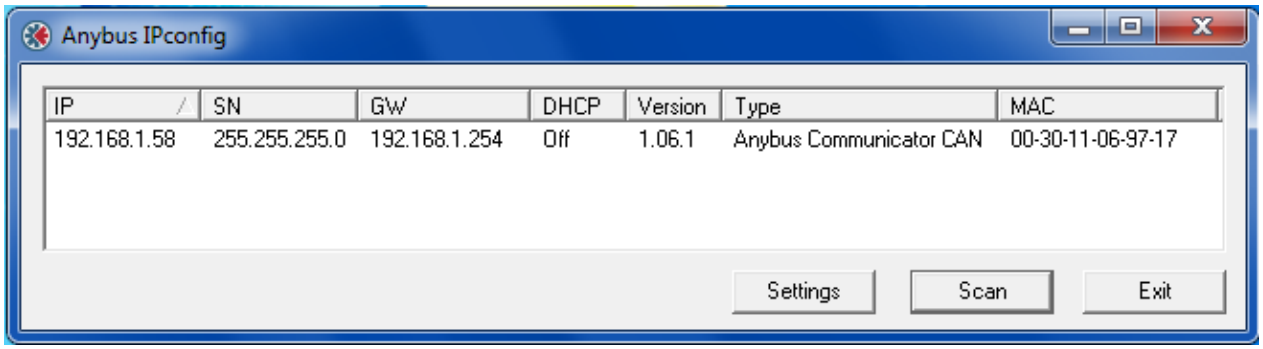
In this example three words (16 bit each) of process data will be exchanged cyclically.

The image below shows an extract from the Appendix CAN_Interfaces_of_SEW_Drives.

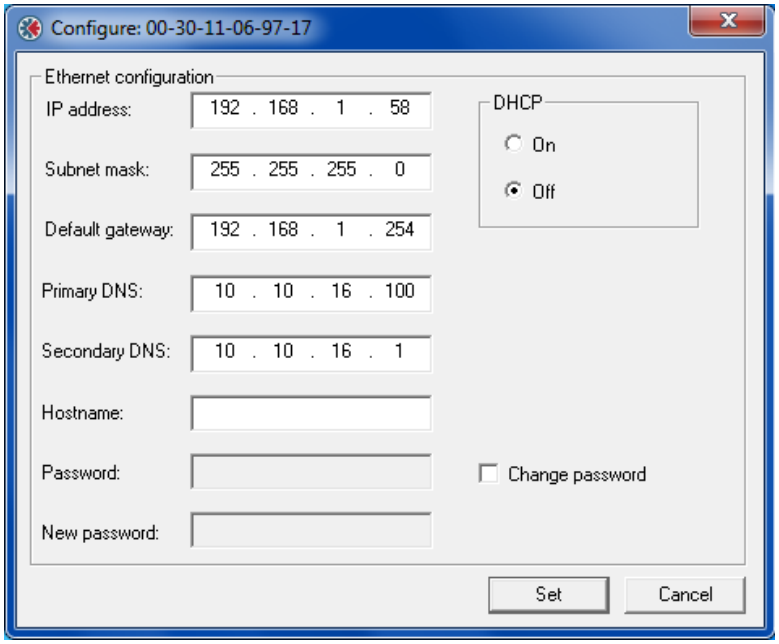


4.1.1 Assigning an IP Address

Launch the Anybus IPConfig tool in order to assign an IP address to the Ethernet port of your Anybus Communicator CAN. The tool will find all Anybus modules in your network. By double-clicking an entry inside the window you will be able to configure its network settings.



Hit the *Set* button in order to permanently store the settings inside the module.

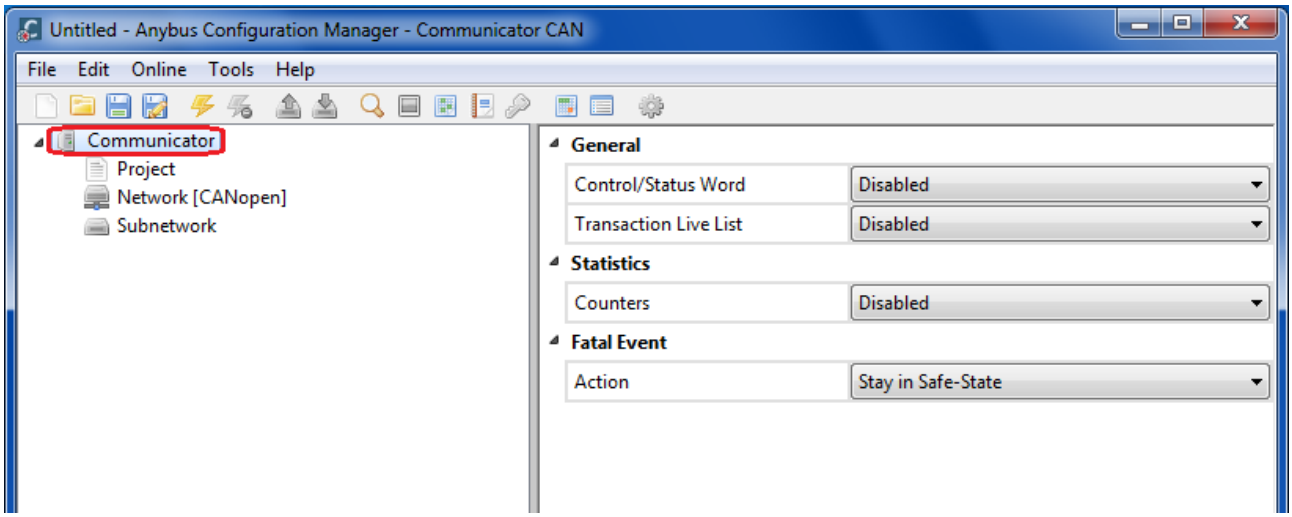


4.1.2 Configuration of the Anybus Configuration Manager – Communicator CAN

Settings for this configuration can be found in the file **AppNote_1_Axle_3_Words_ProcessData.hcg**

Step 1:

Start the application *Anybus Configuration Manager - Communicator CAN*. For easy start-up leave all settings under *Communicator* as default.



Optionally the behaviour of the gateway can be monitored and controlled by setting the following parameters:

Control/Status Word: Information can be mapped to the network I/O data. If enabled, the control word is mapped to the first two bytes in the output data area, and the status word is mapped to the first two bytes in the input data area.

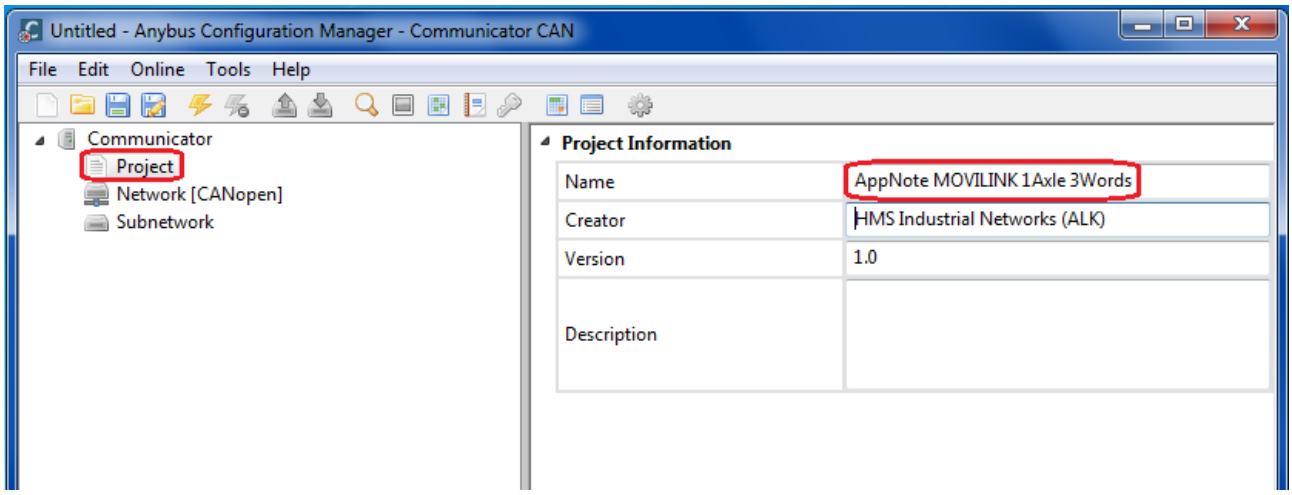
Transaction Live List: A transaction live list, telling the control system if a certain transaction is running or not, can be mapped to the network input data area. Each transaction is represented by a bit in the transaction live list. It is possible to select the size of the transaction live list in this parameter. The transaction live list is mapped from the beginning of the input area. If the Control/Status word is enabled, the transaction live list is mapped after the Status word.

Counters: Transmit and receive counters count successful CAN messages on the sub network. The counters can be mapped to the input data area.

Action: The action in case of a fatal software event can be defined here. If “Stay in safe-state” is selected, the Communicator will be locked in fatal state. If “Software Reset” is selected, the software will be reset and the Communicator will be restarted automatically.

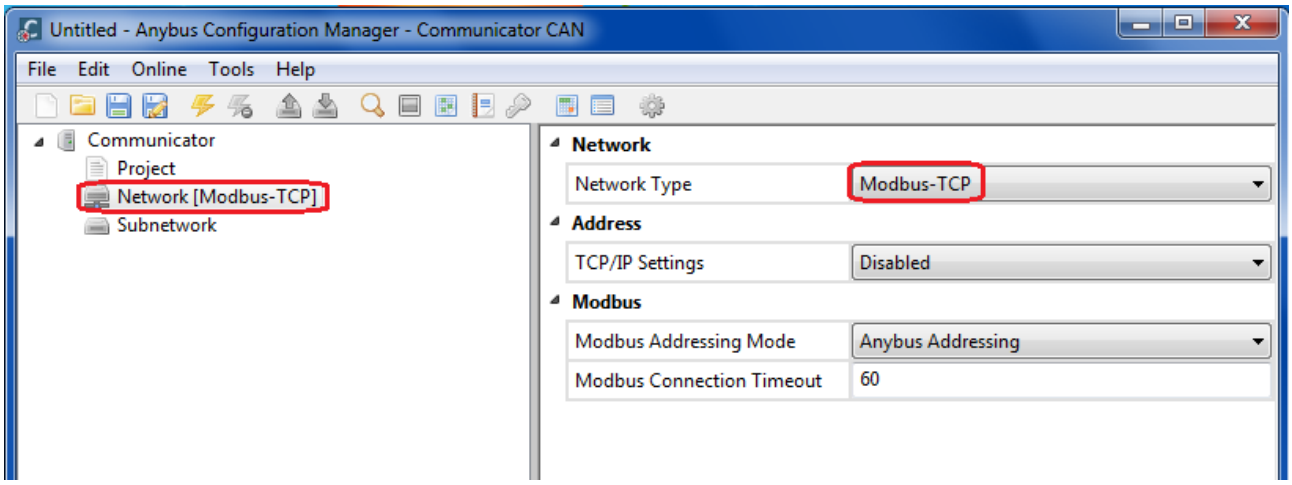
Step 2:

Set the project name to *AppNote MOVILINK 1Axis 3Words*



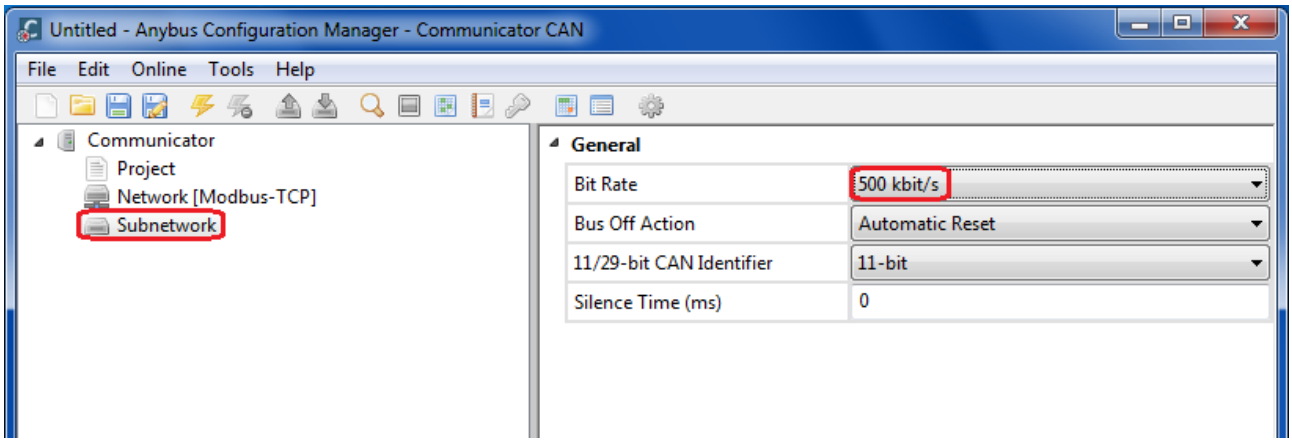
Step 3:

Set the *Network Type* to *Modbus-TCP* under *Network* and leave all other parameters untouched.



Step 4:

Set the *Bit Rate* under *Sub network* to *500 kbit/s* and leave the rest as default.



Bit rate: Select the desired CAN bit rate on the sub network. It must be equal on all nodes in the network

Bus Off Action: Select what will happen to the CAN controller if the CAN network goes Bus Off. (available only when Control/Status word (see step 1) is not used).

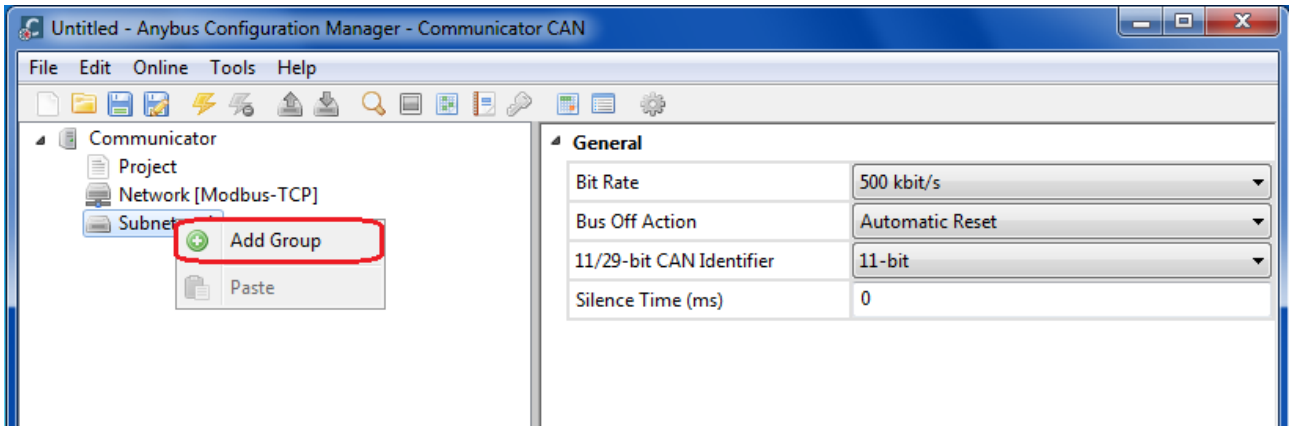
11/29-bit CAN Identifier: Select if the CAN identifier on the sub network shall be 11 bits (CAN 2.0A) or 29 bits (CAN 2.0B). SEW MOVILINK defines the usage of 11 bit identifier, but modern SEW drives won't fail if messages using 29 bit identifier exist in parallel.

If transactions exist in the transaction tree when this parameter is changed, the following will happen. Change from 11 to 29 bits: The 11 bits identifier will be kept, and the IDs will be padded with zeros. Change from 29 to 11 bits: WARNING! The higher bits will be deleted and only the lower 11 bits will be kept. This may in some cases cause faulty CAN identifiers.

Silence Time: Defines the minimum time interval that has to elapse between outgoing CAN frames, e.g. if your CAN node is slow or has a small CAN input buffer. The Anybus Communicator CAN may pause outgoing communication in order to observe the Silence Time.

Step 5:

Right-click *Sub network* in the left menu and select *Add Group.*

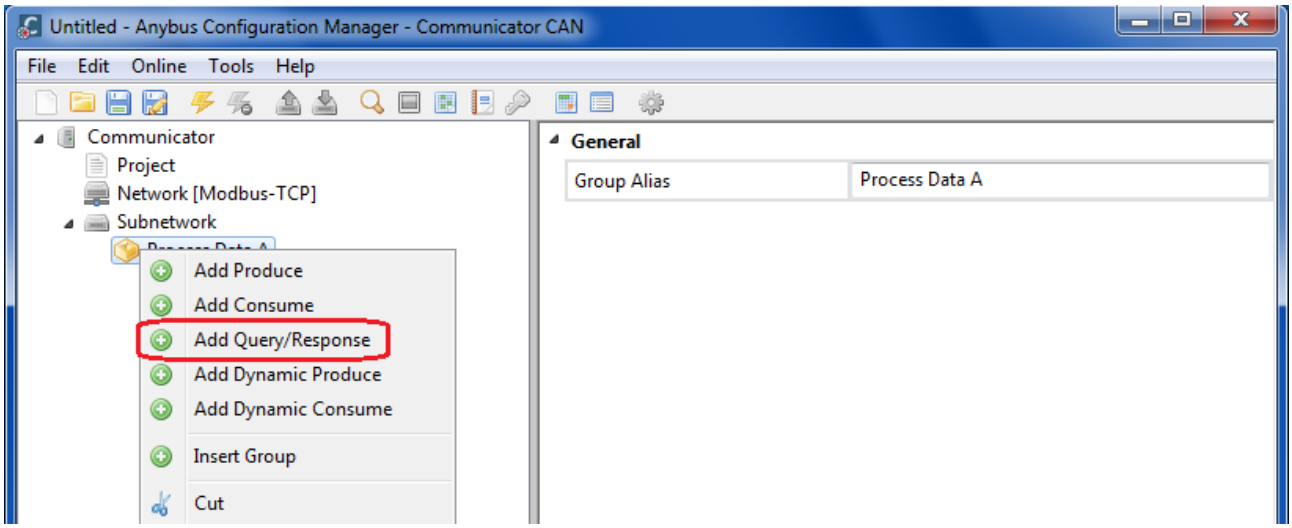


It is not necessary to change the name of the group but it is recommended to choose an appropriate name for it. Select the group and enter the desired name in the *Group Alias* field. In this configuration the name is set to *Process Data A.*

In one group several transactions can be defined. Also several groups can be configured. Transferred data configured in one group will always be copied to and from the upper network in one consistent data block. Data configured in different groups may be received or transmit in different messages on the upper network.

Step 6:

Right-click on the group and select *Query/Response.*



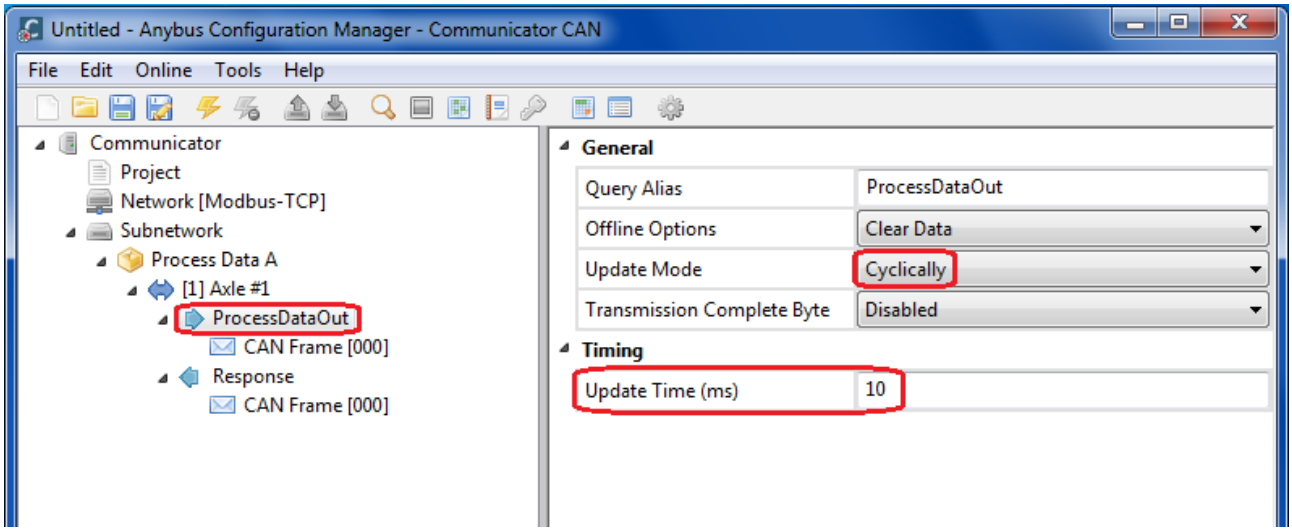
Select Transaction #1 and set the *Transaction Alias* to *Process-Data*.

There are two different transaction types:

- Query/Response:
  After sending a request message, a response is required, so monitoring (see step 3) is done for the entire connection. Request/Response fit more to the master slave approach of SEW MOVILINK.

- Produce/Consume:
  Produces and consumes are totally independent from each other, there is individual monitoring for each Rx and Tx transaction (see step 3). Produce/Consume fits more to the multi master approach of CAN networks.

To get a predictable data transfer via CAN a cyclic communication should be selected:



The selected *Update Time* must fit to the number of defined CAN messages and the selected baud rate:

At a baud rate of 500 kbit/s less than four CAN messages can be transferred within 1 ms. In addition the maximum busload should not exceed 70 % of the available bandwidth to allow transfer of automatic generated error frames.

For example: Having defined eight query/response transactions, one CAN message each, @500 kbit/s gives 16 messages in total. A cycle time of 4 ms would result in 100 % bandwidth. Selecting a cycle time of 6 ms would reduce the busload below 70 %.

A suitable timeout time in the drive then would be 30 ms (3 * cycle-time)

The configuration parameters for the transaction dialog in detail:

Offline Options: Select what will happen with the output data if the network goes offline.

Update Mode: Defines how the transmission of the transaction is triggered.

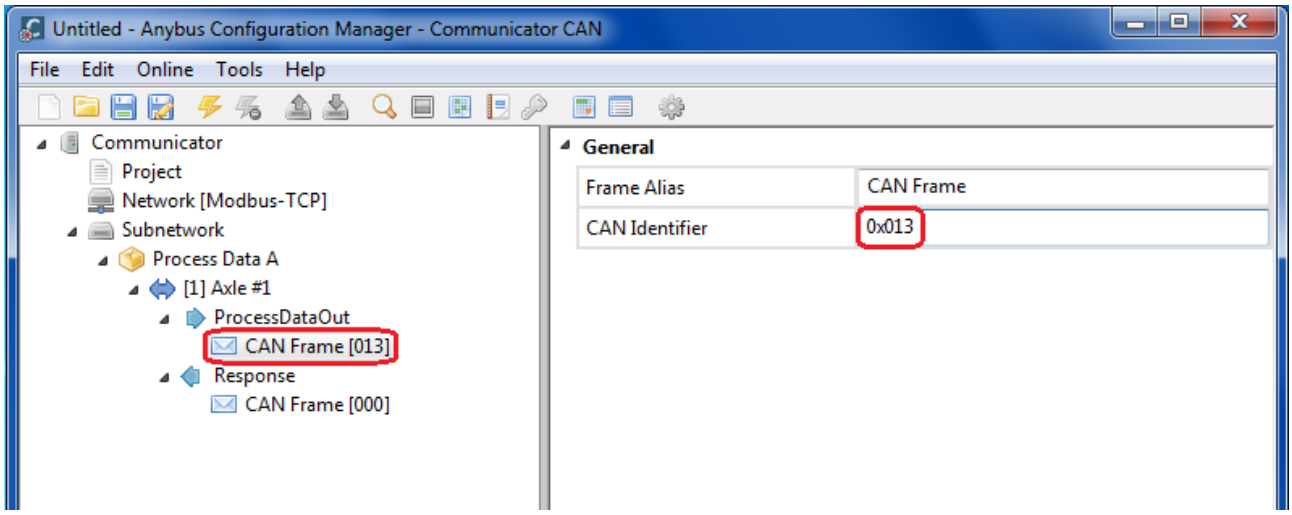
Update on RTR: If a message on the configured CAN ID for a produce transaction is received with the RTR-bit set (Remote Transmission Request), the produce transaction is triggered to be sent. Only one CAN frame is allowed in the transaction for this functionality to work. RTR cannot be used with SEW drives in MOVILINK-Profile

Transmission Complete Byte: Select whether the Transmission Complete Byte shall be enabled or disabled. When a query transmission is completed, the Transmission Complete Byte will be increased by one.

Update Time (ms): Defines the interval time (in ms) between two transmissions when update mode "cyclically" is selected. Valid Range: *5-65535 ms.*

Now the CAN frames must be configured for each transaction:

Set the CAN Frames *CAN Identifier* value to *0x013.*



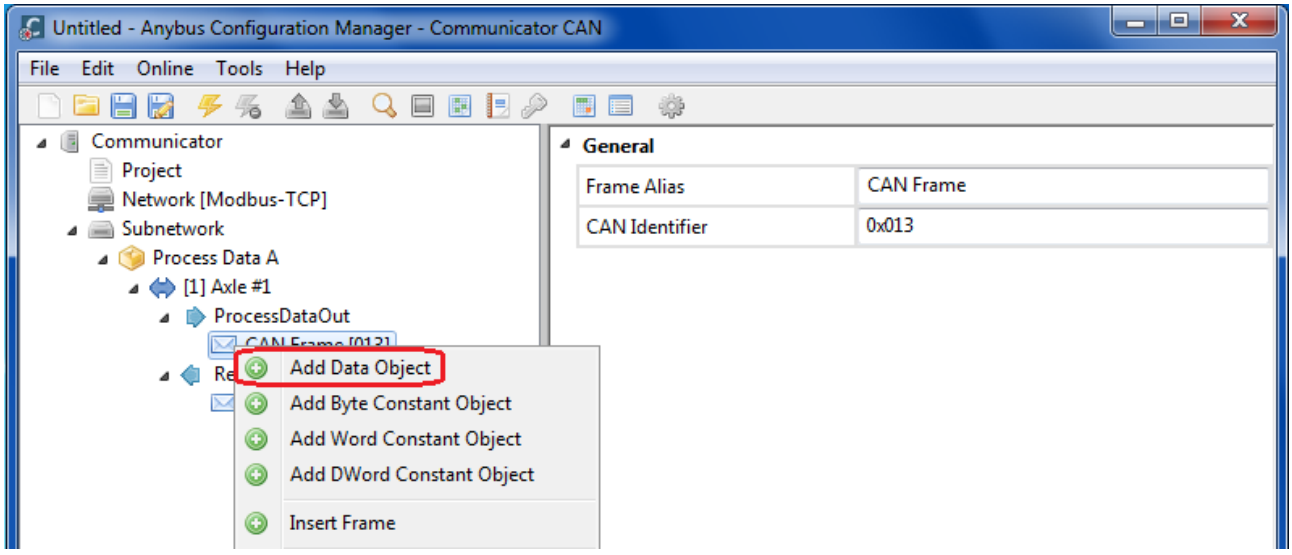
CAN Identifier: Select the identifier for the CAN frame. The valid range depends on whether 11-bit or 29-bit identifiers are selected in the Communicator settings. Valid Range: 0x000-0x7FF.

In the SEW MOVILINK profile the CAN identifier for process output data messages (PO) is
8 * SBus address + 3.

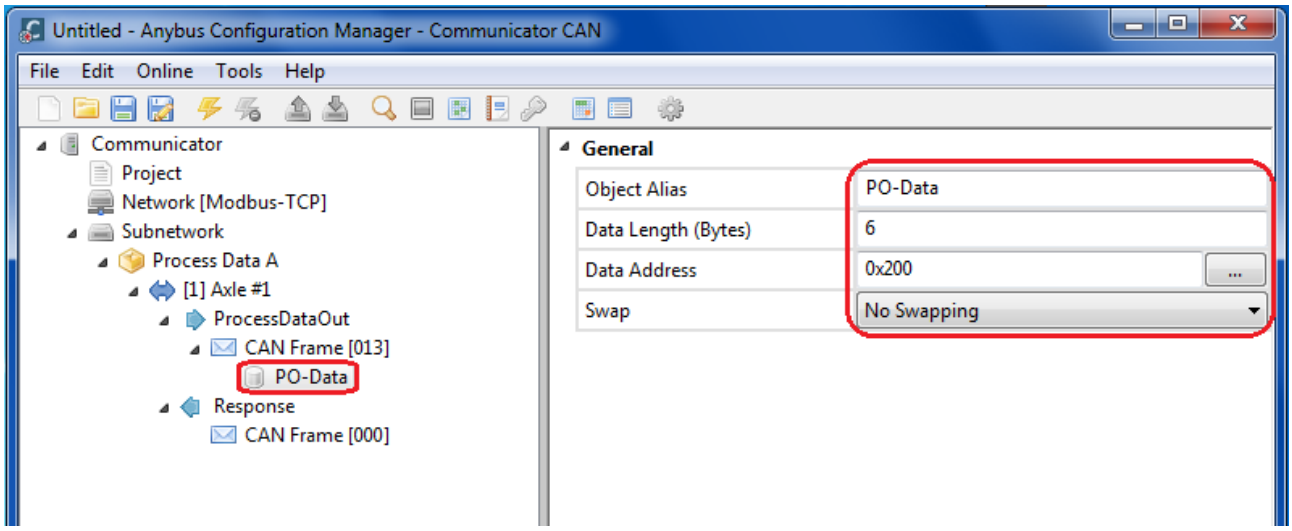
Example: SBus address = 2 → CAN identifier = 8 * 2 + 3 = 0x013

Step 7:

Add Data Object to the CAN Frame by right-clicking the CAN frame and select *Add Data Object.*



Select the data object and assign the *Object Alias* "PO-Data". An SEW drive always offers 3 PD-words in each direction so set *Data Length (Bytes)* to 6, *Data Address* to *0x200* with *No Swapping.*



Data Length (Bytes): Length in bytes of the data object. Possible values are 1-8 bytes. Note that 8 bytes is the total size of the data area in the CAN frame, i.e. if other objects are added the maximum size of the data object will decrease. Valid Range: 1-8.

Data Address: Address in the data area where the data object shall be mapped. Valid range: 0x200-0x3FF or 0x400-0x7FF. The data block beginning at 0x200 contains the data received by the upper network.

Assuming individual data to all drives connected via CAN, the data to the drives should be mapped in consecutive order with no overlapping (and no gaps in-between).

Swap: Depending on the data length value it is possible to word swap or double word swap the data.

SEW MOVILINK defines 16-bit-PD in Motorola data format, so if the upper network (or PLC) also works in Motorola data format (as PROFIBUS, PROFINET or Modbus) "No Swapping" should be selected.
If the upper network (or PLC) works in Intel data format (as DeviceNet, EtherNet/IP or CANopen) "Word Swap" should be selected.

Step 8:

Leave all settings under *Response* as default.

Offline Options: Select what will happen with the input data if the transaction on the subnetwork is stopped.

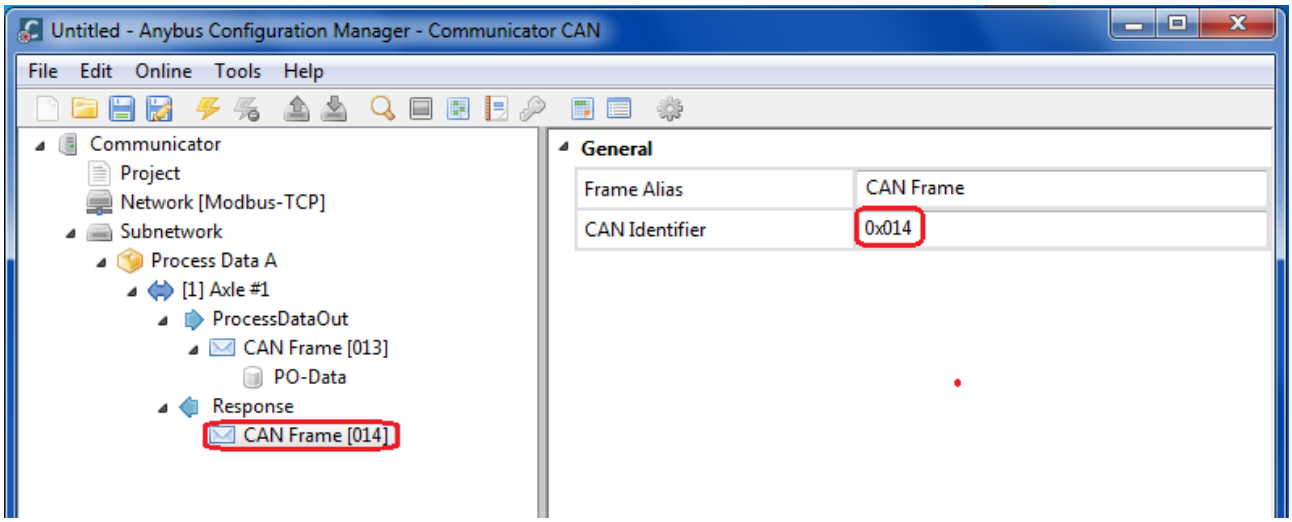
Consistency Check: Only applicable if a response includes more than one CAN frame. In this case all defined CAN frames have to be received for the input data area to be updated.

Offline Timeout (ms): The maximum time before the transaction is considered to be lost. Use 0 to disable the timeout. Valid Range*: 0, 10-65535.*

Reception Trigger Byte: Select whether the Reception Trigger Byte shall be enabled or disabled. The Reception Trigger Byte is incremented each time a response is received.

Transaction Status Byte: if enabled an additional byte is mapped to the input data area storing the current status of this transaction (timeout, data errors, execution yes/no).

Set the CAN Frames *CAN Identifier* value to *0x014.*



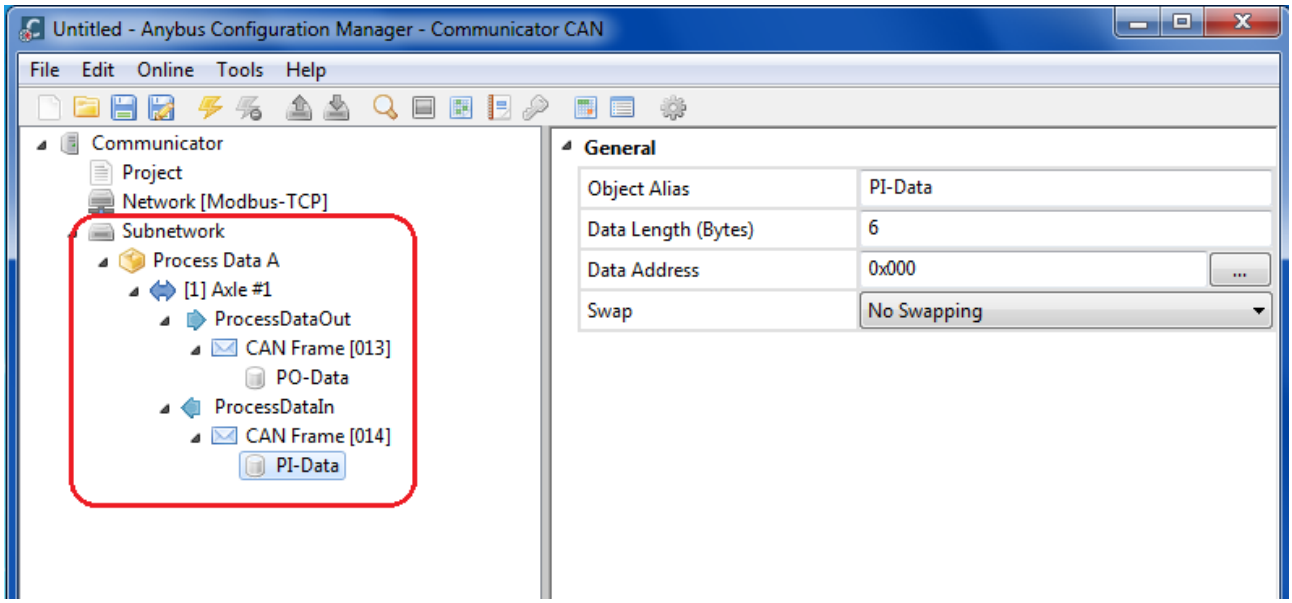
In SEW MOVILINK-Profile the CAN identifier for process input data messages (PO) is 8 * SBus address + 4.

Example: address = 2 → CAN identifier = 8 * 2 + 4 = 0x014

Step 9:

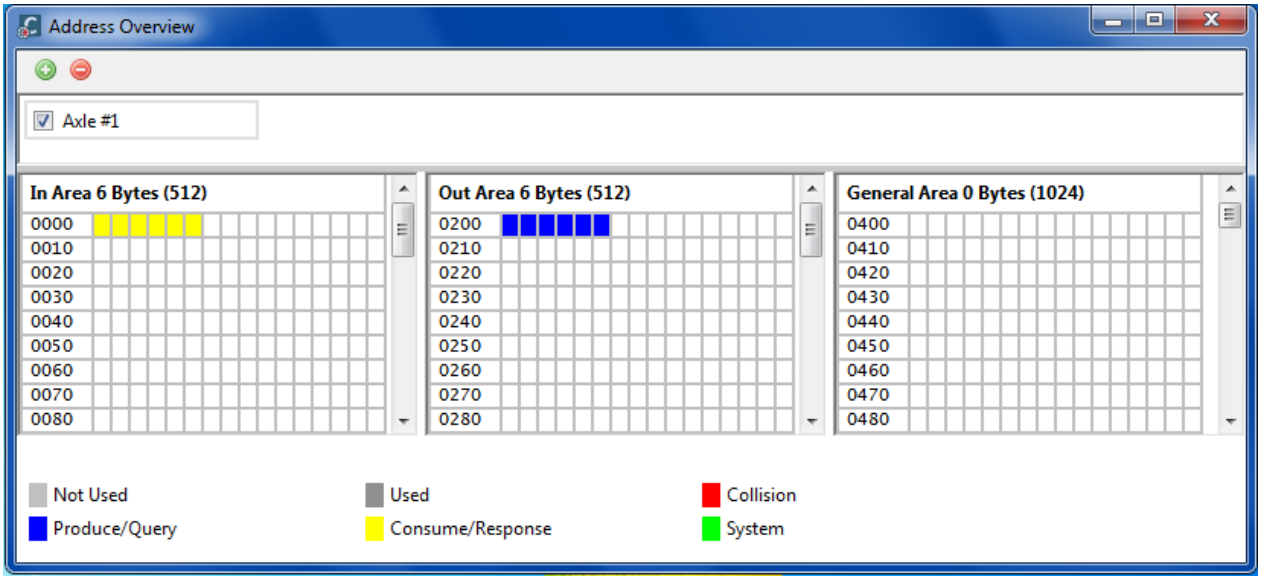
Add *Data Object* to the CAN frame and assign the *Object Alias* "PI-Data", set *Data Length (Bytes)* to 6, *Data Address* to *0x000* and *No Swapping.*

When the configuration is done it should look like this:



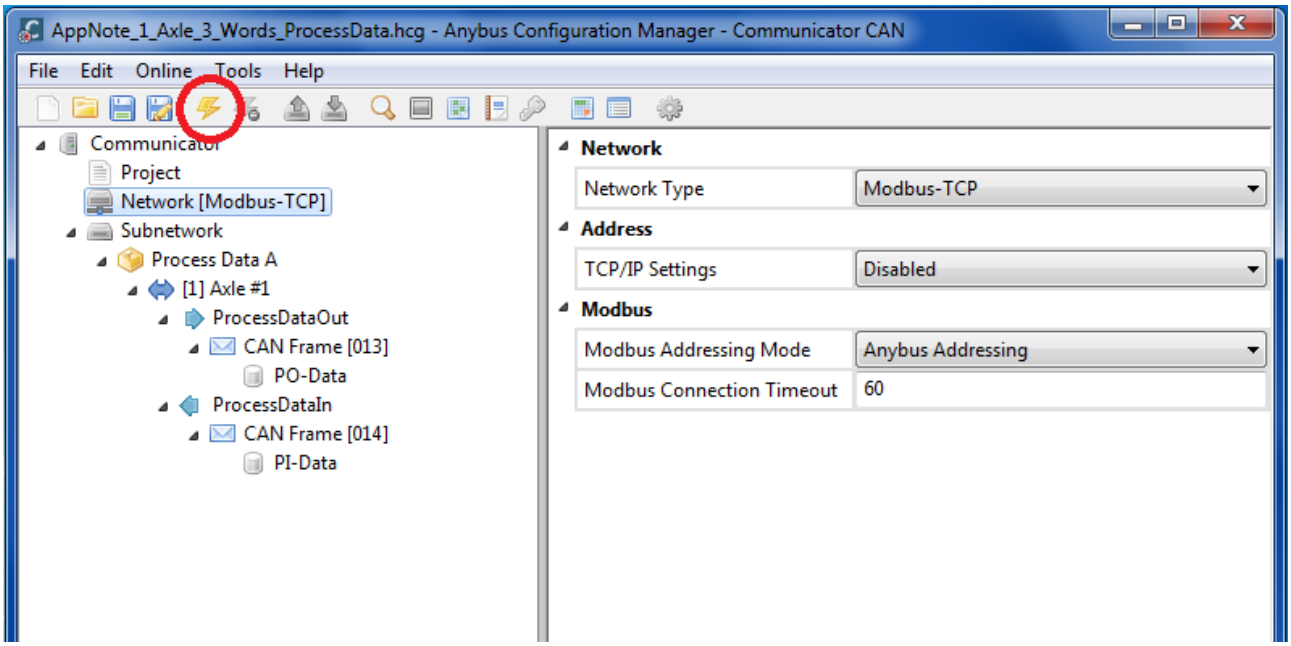
Step 10:

In the menu go to *Tools* > *Address Overview*. You see that six bytes are mapped in the *In-* and *Out-Areas*, respectively.



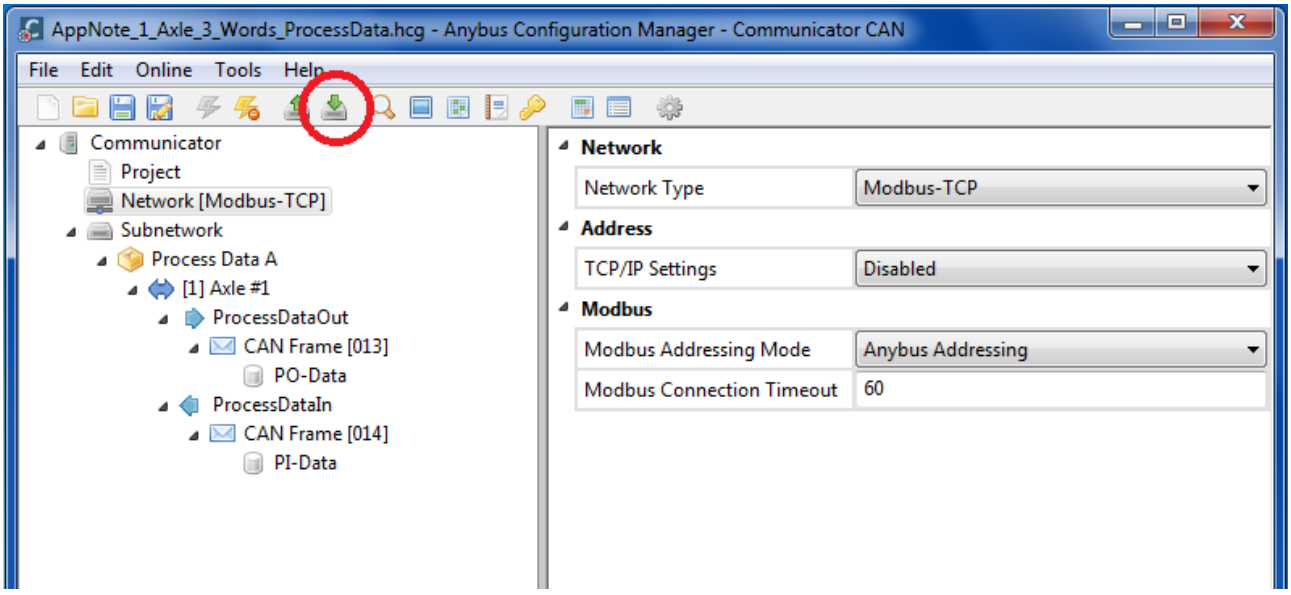
Step 11:

Establish a connection by clicking on the *Connect* icon or by selecting *Online->Connect* in the top menu.



Step 12:

Download the configuration by clicking on the *Download Configuration to Communicator* icon or by selecting *Online>Download Configuration* from the menu.



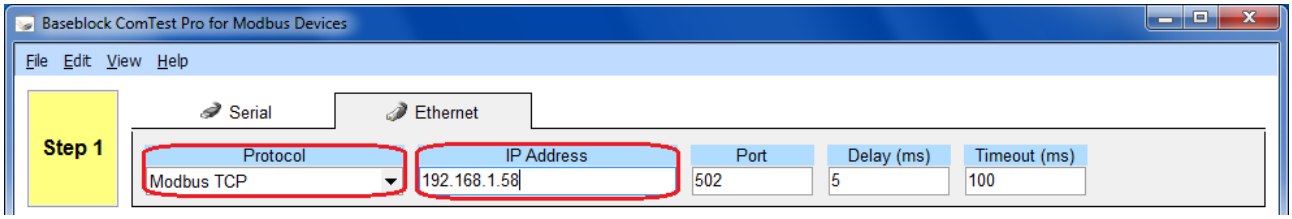
Now the configuration of *Anybus Communicator CAN* for exchanging three words as process data cyclically is done.

4.1.3 Testing the Configuration with ComTestPro

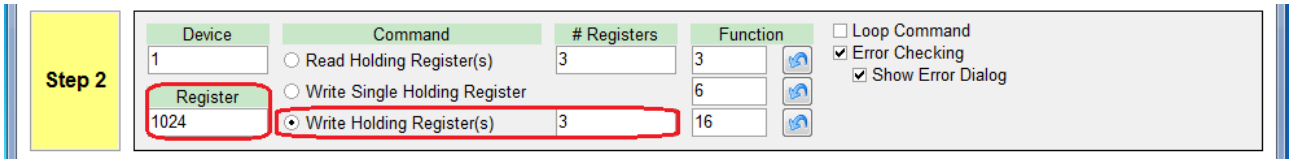
Start the application *ComTestPro* which works as a Modbus-TCP master simulated in software. We will use it to write to the Anybus Communicator CAN's output area and read from its input area. The three steps marked yellow in the user interface will guide you through the setup.

4.1.3.1 Writing to the Output Area

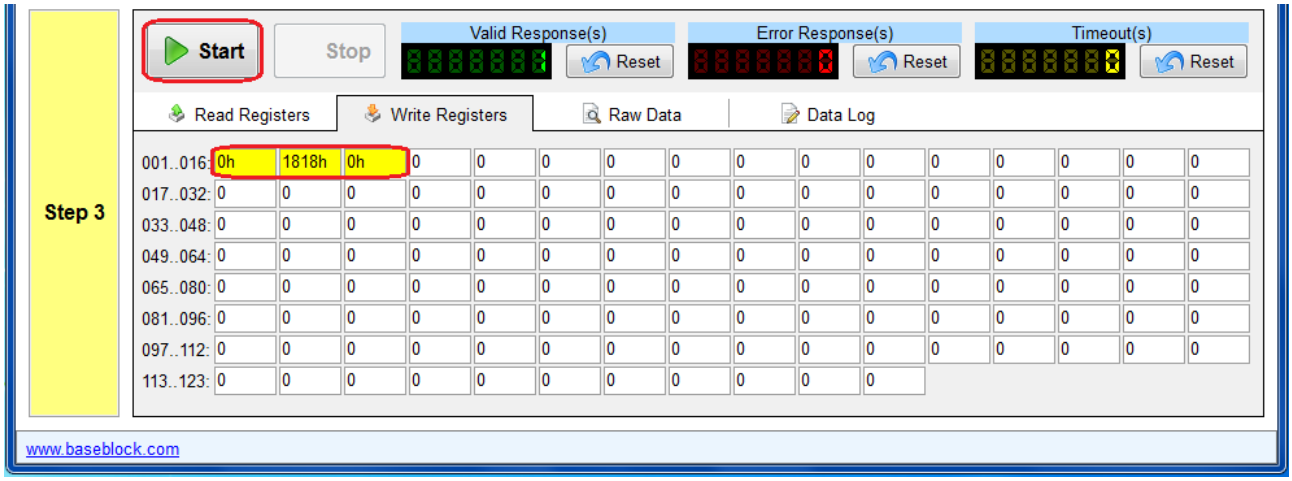
Step 1: Click on the Ethernet tab and make sure that Modbus-TCP protocol is selected. Then, enter the IP address of the Anybus Communicator CAN:



Step 2: In the *Register* field the register offset of the output area is expected. Enter the value 1024 marking the beginning of the output area. Note that the output area is mapped to registers 4x1025-4x1280. The register offset is computed by subtracting one (1). Then choose the Modbus function code *Write Holding Registers* and enter 3 for the number of registers to be written.



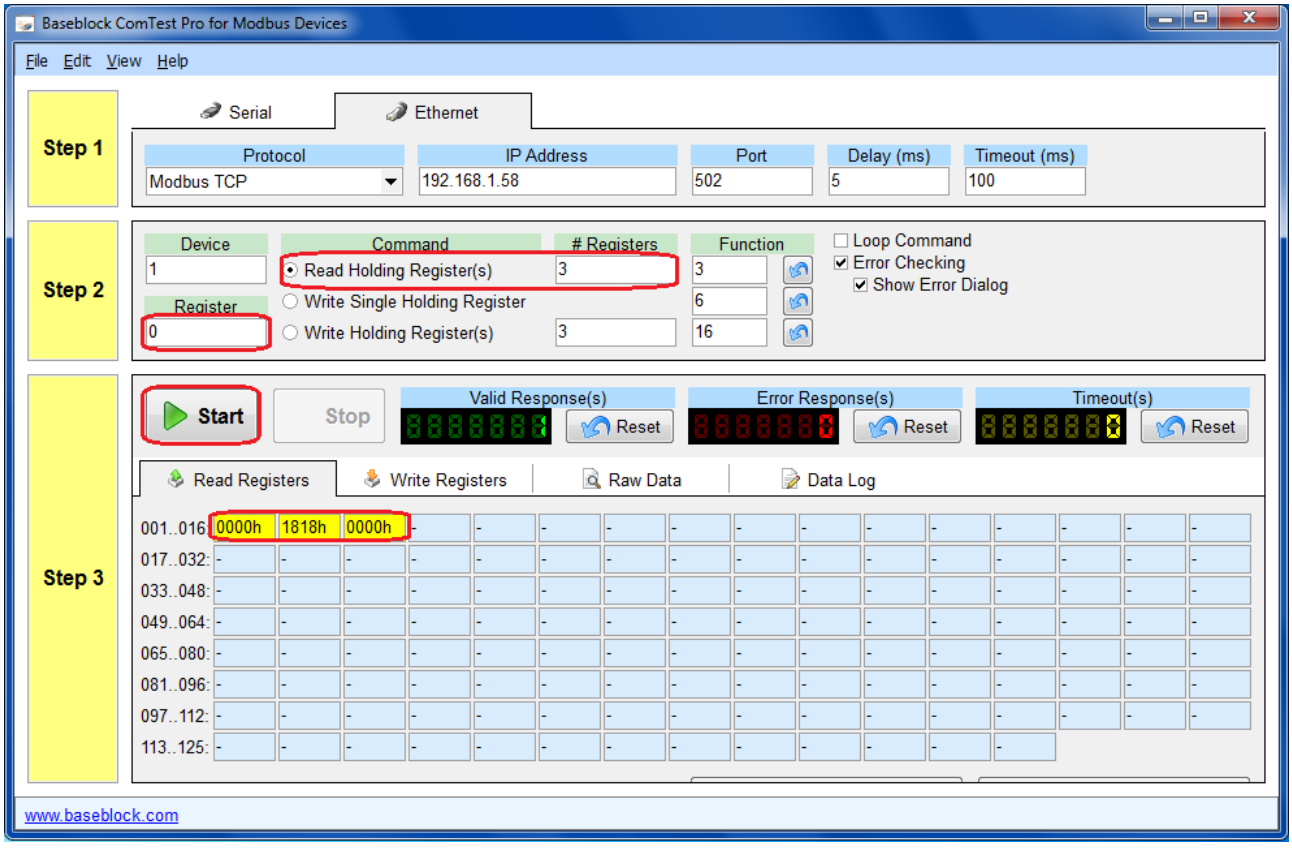
Step 3: Enter some PO data into the three fields highlighted in yellow and hit the *Start* button to send out the data. Note that Modbus registers are 16 bits wide.



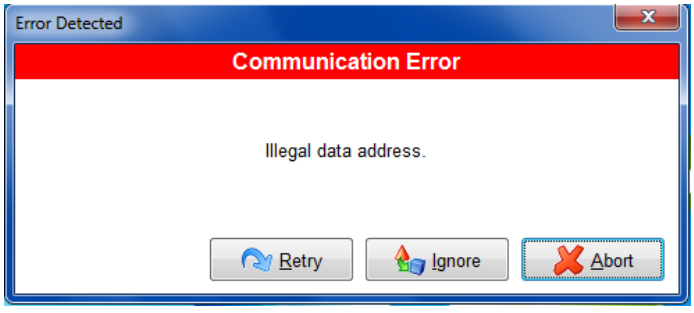
4.1.3.2 Reading the Input Area

In order to read from the input area choose the Modbus function code *Read Holding Registers*, type in 3 for the number of registers to read, enter register offset 0 and hit the *Start* button. The three yellow fields hold the data. The input area is mapped to registers 4x0001-4x0256.

In this example the drive is set to loop incoming data back. This is done as a verification that everything works.



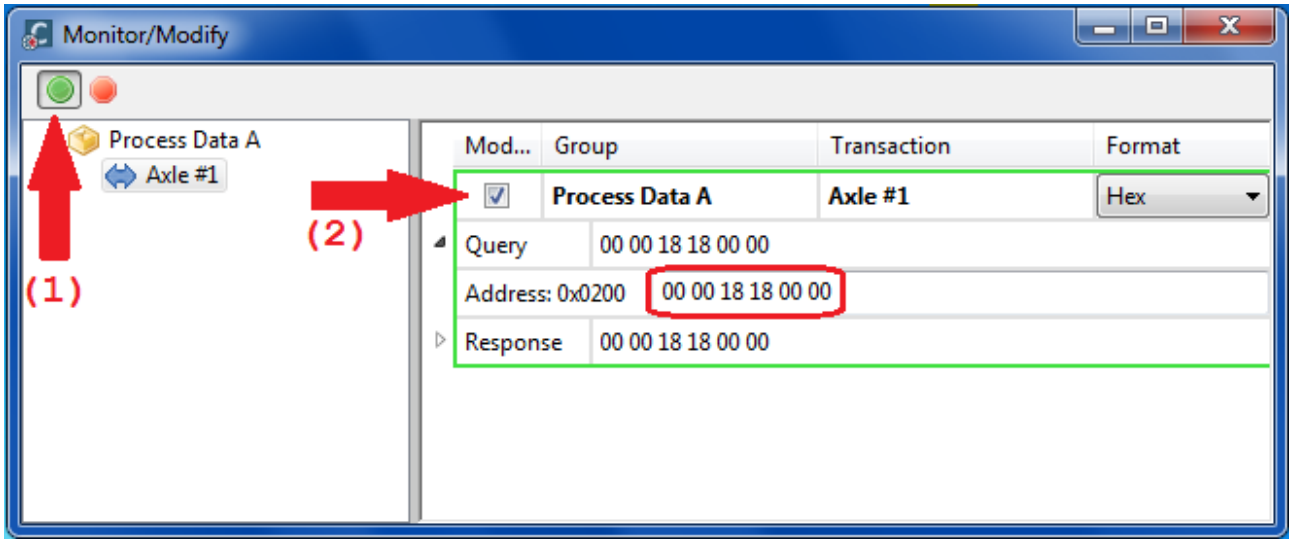
If you get this communication error you've entered an invalid register offset:



4.1.3.3 Using Monitor/Modify in Anybus Configuration Manager

It is also possible to monitor and modify the input and output area via Anybus Configuration Manager.

To open the Monitor/Modify window click the magnifying glass icon or select Tools > Monitor/Modify from the menu. Next, hit the green circle to enable Monitor/Modify and the checkbox next to the "*Process Data A"* transaction. Now you are able to edit values in the output data area of the Communicator CAN.



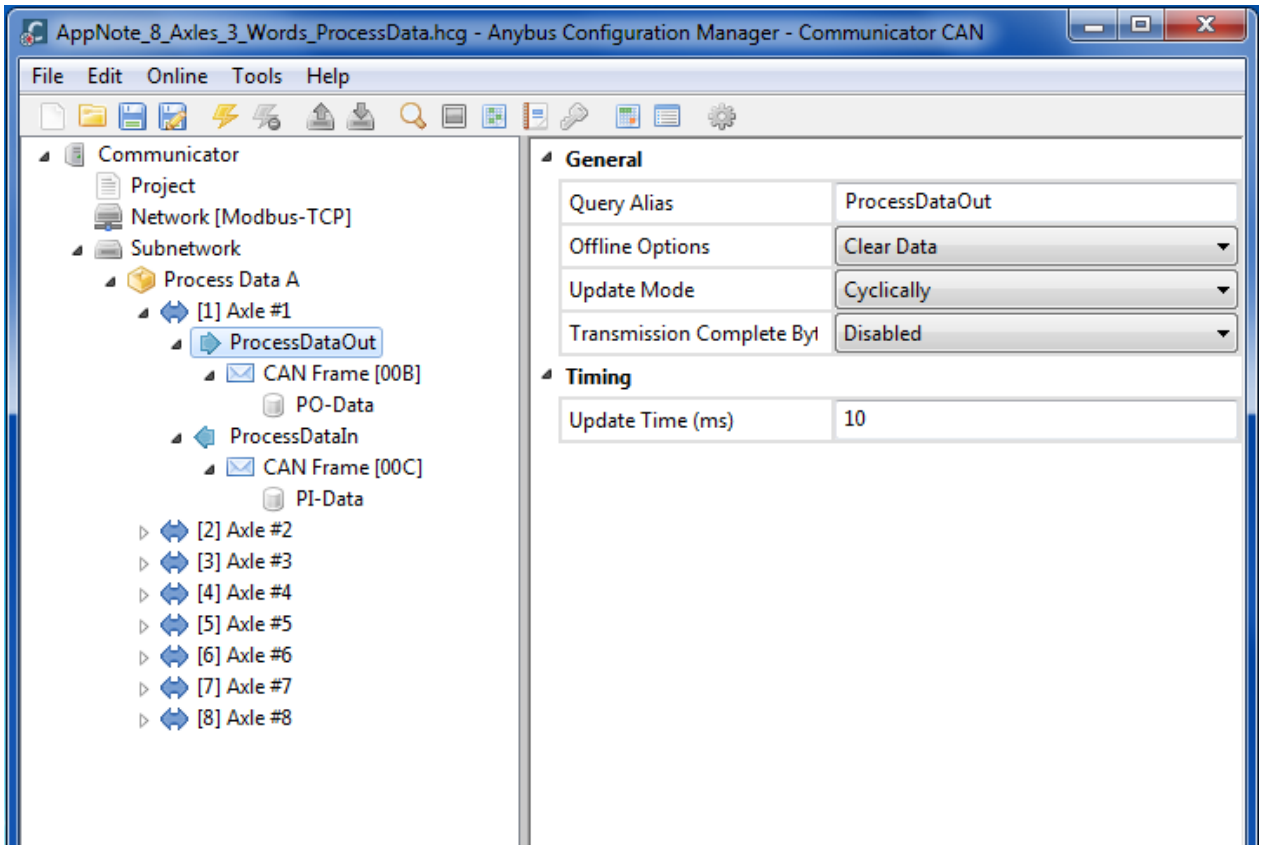
The transferred data can be monitored with SEW engineering software MOVITOOLS MotionStudio in e.g. Parameter-tree or Fieldbus-Monitor.

Content and scaling of the transferred process data words is not subject of this application note, please refer to the drives manuals available from SEW.

4.2 Configuration 2: Eight Drives exchanging Process Data Cyclically

The settings for this configuration can be found in the file **AppNote_8_Axles_3_Words_ProcessData.hcg**.

In this example eight drives will be exchanging three process data words similar as in chapter 4.1. The drives are assumed to have SBus addresses 1 through 8. An update time of 10 ms is still sufficient for a total of 16 CAN frames being exchanged. For this configuration no detailed information will be given.



4.3 Configuration 3: One Drive Exchanging Ten Process Data Words using Fragmentation

In this example ten process data words will be exchanged cyclically with one drive using fragmentation over four CAN frames.

There will be no extended information for the configuration in this chapter. Please see chapter 4.1, Configuration 1, for a more in detail description of the setup.

4.3.1 Assigning an IP Address

Please refer to chapter 4.1.1 for the network setup.

4.3.2 Configuration of the Anybus Configuration Manager – Communicator CAN

Settings for this configuration can be found in the file **AppNote_1_Axle_10_Words_ProcessData.hcg**.

Step 1:

Start the application *Anybus Configuration Manager - Communicator CAN*. For easy start-up leave all settings under *Communicator* as default.

Step 2:

Set the project name to *AppNote MOVILINK 1Axis 10Words.*

Step 3:

Set *Network Type* to *Modbus-TCP* under *Network.*

Step 4:

Set *Bit Rate* under *Subnetwork* to *500 kbit/s* and leave the rest as default.

Step 5:

Add Group under *Subnetwork* and change the *Group Alias* to *Process Data A.*

Step 6:

Add Query/Response-Transaction. In the Query properties set the *Update Time* to 10 ms. This is still fine for the amount of CAN frames. Set the *CAN identifier* value to *0x013* for the CAN frame.
CAN identifier for process output data telegram (PO): 8 * SBus address + 3 = 8 * 2 + 3 = 0x013

Step 7:

Add three more CAN frames under *Query* by right-clicking *Query* and selecting *Add Frame three times.* Set the *CAN identifier* value to *0x013* for all produced CAN frames.

Step 8:

The four CAN frames should be configured according to the list below:

First frame:

1. *Add Byte Constant Object.* Set *Value* to *0x00.*
2. *Add Byte Constant Object.* Set *Value* to *0x0A.*
3. *Add Data Object.* Set *Data Length (Bytes)* to 6, *Data Address* to *0x200* and *No Swapping.*

Second frame:

1. *Add Byte Constant Object.* Set *Value* to *0x41.*
2. *Add Byte Constant Object.* Set *Value* to *0x0A.*
3. *Add Data Object.* Set *Data Length (Bytes)* to 6, *Data Address* to *0x206* and *No Swapping.*

Third frame:

1. *Add Byte Constant Object.* Set *Value* to *0x42.*
2. *Add Byte Constant Object.* Set *Value* to *0x0A.*
3. *Add Data Object.* Set *Data Length (Bytes)* to 6, *Data Address* to *0x20C* and *No Swapping.*

Fourth frame:

1. *Add Byte Constant Object.* Set *Value* to *0x83.*
2. *Add Byte Constant Object.* Set *Value* to *0x0A.*
3. *Add Data Object.* Set *Data Length (Bytes)* to 2, *Data Address* to *0x212* and *No Swapping.*
4. *Add Double-Word- Constant Object.* Set *Value* to *0x00000000.*

Step 9:

Set the *CAN identifier* value, under *Response,* to *0x014* for the CAN frame.
CAN identifier for process input data telegram (PO): 8 * SBus address + 4 = 8 * 2 + 3 = 0x014

Step 10:

Add three more CAN frames under *Response* by right-clicking *Response* and select *Add Frame three times.*
Set the *CAN identifier* value to *0x014* for all produced CAN frames.

Step 11:

The four CAN frames should be configured according to the list below:

First frame:

4. *Add Byte Constant Object.* Set *Value* to *0x00.*
5. *Add Byte Constant Object.* Set *Value* to *0x0A.*
6. *Add Data Object.* Set *Data Length (Bytes)* to 6, *Data Address* to *0x000* and *No Swapping.*

Second frame:

4. *Add Byte Constant Object.* Set *Value* to *0x41.*
5. *Add Byte Constant Object.* Set *Value* to *0x0A.*
6. *Add Data Object.* Set *Data Length (Bytes)* to 6, *Data Address* to *0x006* and *No Swapping.*

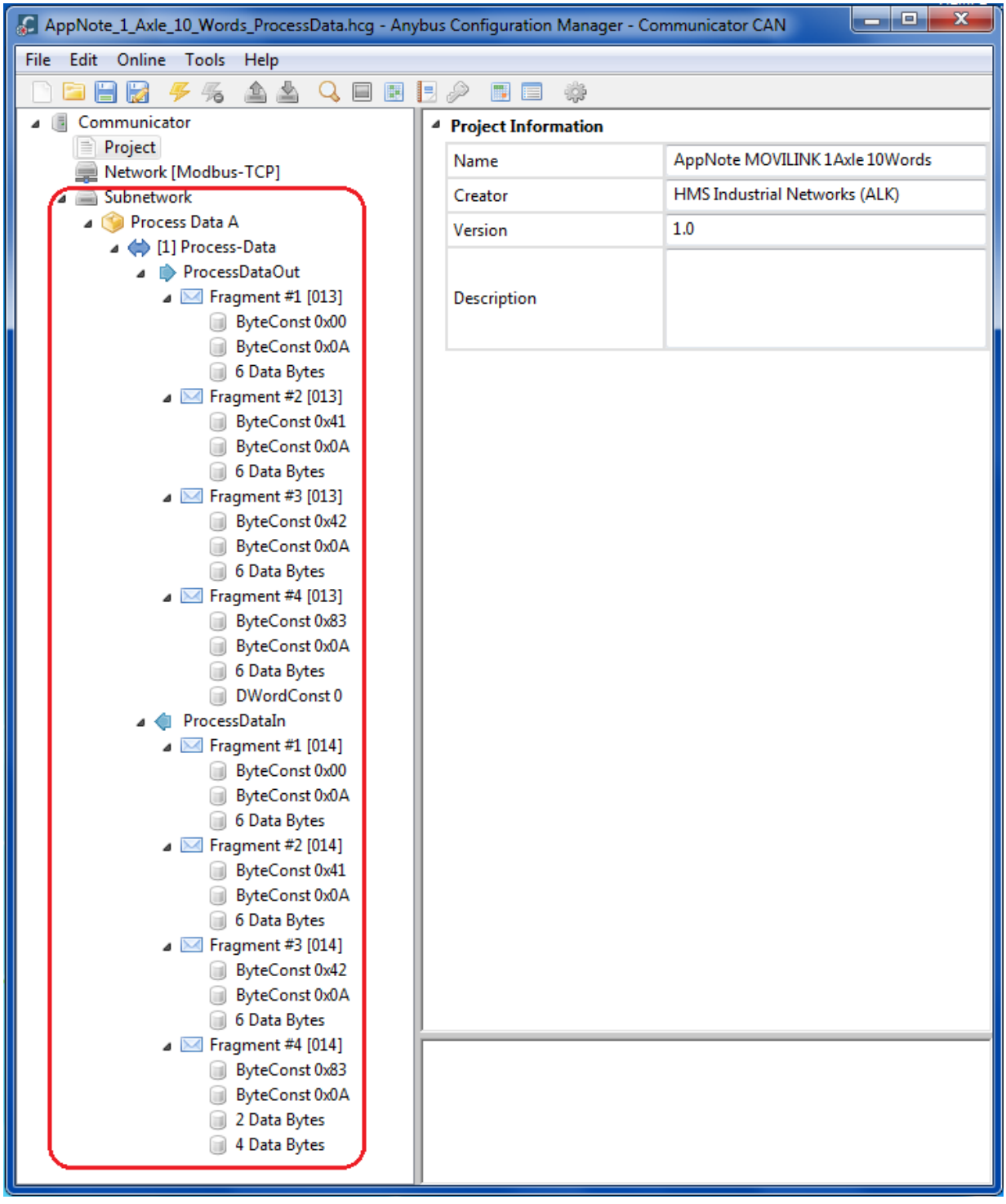
Third frame:

4. *Add Byte Constant Object.* Set *Value* to *0x42.*
5. *Add Byte Constant Object.* Set *Value* to *0x0A.*
6. *Add Data Object.* Set *Data Length (Bytes)* to 6, *Data Address* to *0x00C* and *No Swapping.*

Fourth frame:

5. *Add Byte Constant Object.* Set *Value* to *0x83.*
6. *Add Byte Constant Object.* Set *Value* to *0x0A.*
7. *Add Data Object.* Set *Data Length (Bytes)* to 2, *Data Address* to *0x012* and *No Swapping.*
8. *Add Data Object.* Set *Data Length (Bytes)* to 4, *Data Address* to *0x400* and *No Swapping.* (Setting data address ≥ 0x400 means the data won't be transferred to the primary network.)

When the configuration is done it should look like this:



4.4 Configuration 4: Eight Drives Each Exchanging Ten Process Data Words using Fragmentation

The settings for this configuration can be found in the file **AppNote_8_Axles_10_Words_ProcessData.hcg**.

In this example eight drives will be exchanging ten process data words cyclically using fragmentation over four CAN frames similar as in chapter 4.3. The drives are assumed to have SBus addresses 1 through 8.

There are eight transactions with four transmitted and four received CAN frames totalling in 64 CAN frames. At a baud rate of 500 kbit/s about 4 frames/ms can be exchanged loading up the bus to 100 %.

➔ 64 $frames$ / 4 $\frac{frames}{ms} = 16\ ms$ cycle time.

To keep the bus load below 70 % we will choose an update time (cycle time) of 25 ms for all eight transactions.