Emotron Fieldbus
HMS Gateway Anybus Communicator

Quick Start Guide

English

emotron
DEDICATED DRIVE
A **CG** Product

Quick start to HMS Gateway Anybus Communicator

This document provides a short example on how to configure the Anybus Communicator Gateway from HMS with Emotron MSF 2.0 and Emotron VFX/FDU 2.0.

NOTE: All support/questions regarding configuration/ installation of the gateway are handled by HMS Industrial Networks, http://www.hms.se/

Questions regarding the Softstarter or Variable Speed Drive, VSD, are handled by CG Drives & Automation.

The following examples show how to configure the telegrams on the sub network side where the gateway works as a modbus-rtu master.

Examples of configuration files for the HMS gateway are found under FDU/VFX and MSF at: http://www.emotron.com/services-support/file-archive.

The files can be downloaded and reconfigured to your specific application/needs.

Hint! Keep the following manuals nearby before continuing:

For VSD:

- Emotron VFX/FDU 2.0 Instruction manual

- Emotron I/O board 2.0 Option (describes process data mapping)

- Emotron Isolated RS232/485 2.0 Option (describes modbus rtu protocol)

For Softstarters:

- Emotron MSF 2.0 Instruction manual

- Emotron MSF 2.0 Fieldbus Option (describes process data mapping)

- Emotron MSF 2.0 Serial Communication Option (describes modbus protocol)

In the following examples the modbus registers for the 4 bytes process data (described in each corresponding fieldbus manual) are mapped to the double port memory of the gateway. This enables access (read/write) of selected modbus registers from the fieldbus side.

General configuration Gateway

NOTE: Please refer to the Anybus Communicator manual on www.hms.se for detailed information.

The following text describes step by step how to configure the gateway and sub network.

NOTE: Preconfigured examples both for MSF 2.0 and VSD 2.0 may be downloaded from www.emotron.com under File archive.

1. Start ABC Config Tool program.
2. Choose Blank Configuration and press OK.

NOTE: The wizard does not allow byte swapping, therefore the manual configuration mode is recommended.
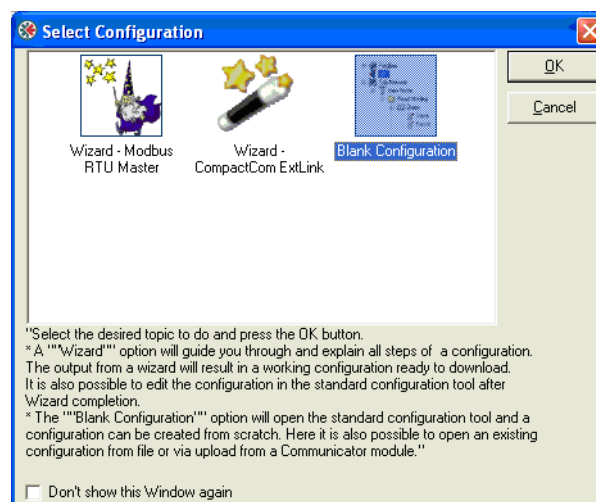

Fig. 1

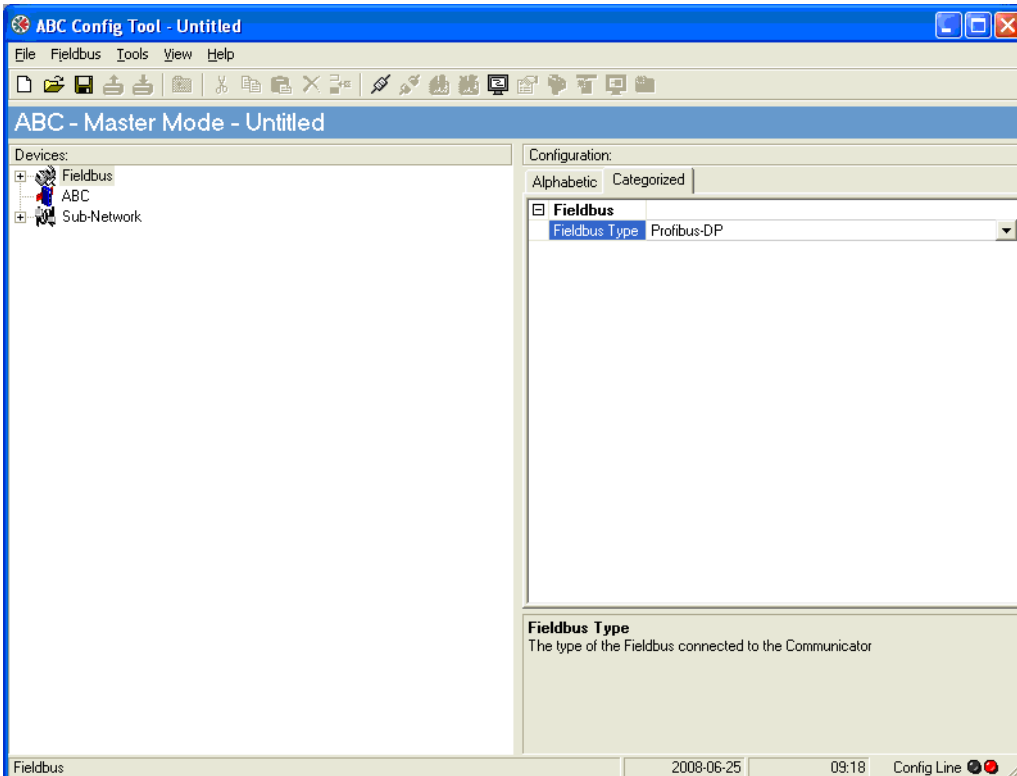3. Click on Fieldbus to the left and choose Fieldbus type to the right.

Fig. 2

4. Click on Sub-Network to the left

5. Make appropriate settings for baudrate, data bits (8), parity (none), physical interface and stop bits (2) matching the settings in MSF/FDU/VFX.
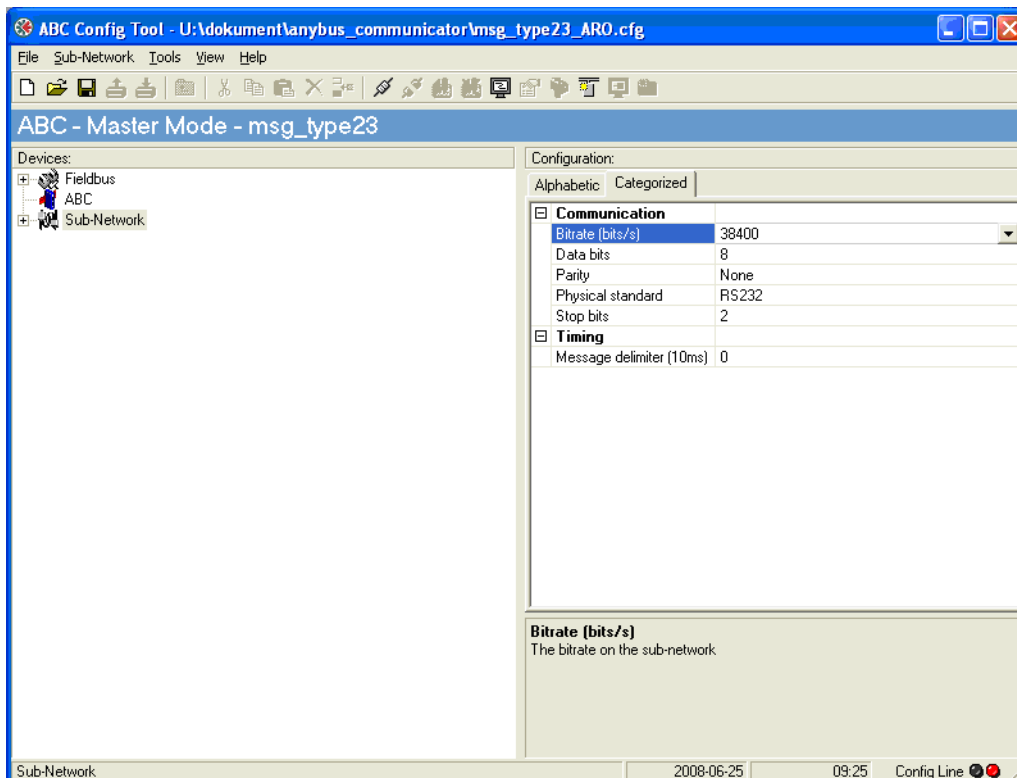

Fig. 3

CG Drives & Automation 01-5971-01r1

6. Expand Sub-Network structure (with +) to the left and add the number of slave nodes you want on the sub network. Click on the Sub-Network and select "Add node" to add a new node. You may rename the name of the "New Node" to something more appropriate, e.g. like "Inverter2".

7. For each node you add you must select a unique address, matching the address setting in the product (MSF, FDU or VFX). To do this, click on the Node name and select Slave address to the right.
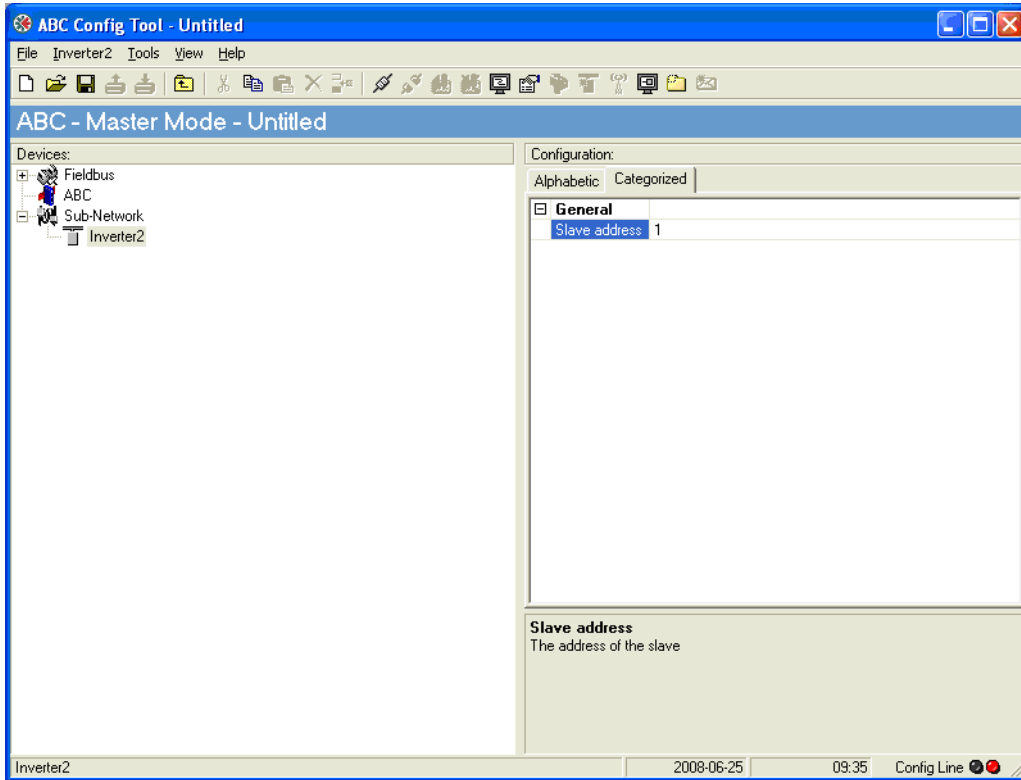

Fig. 4

8. Right click on the node name again and select "Add transactions".

9. Continue to chapter Setup sub network for your specific product.

Setup sub network for MSF 2.0

First setup the MSF menus:

For detailed menu information, please check manuals listed above.

Setup the following menus:

[200] Selection of control sources – set to 3 = Serial communication control

[270] Serial comm unit address

[271] Serial comm baudrate

[272] Serial comm parity – set to 0 = No parity

[273] Serial comm contact broken

NOTE: The exact same address, baudrate etc. should be configured for the gateway.

We will now continue our configuration of the gateway to setup a modbus rtu message that will read and write modbus registers containing the 4 byte fieldbus process data.
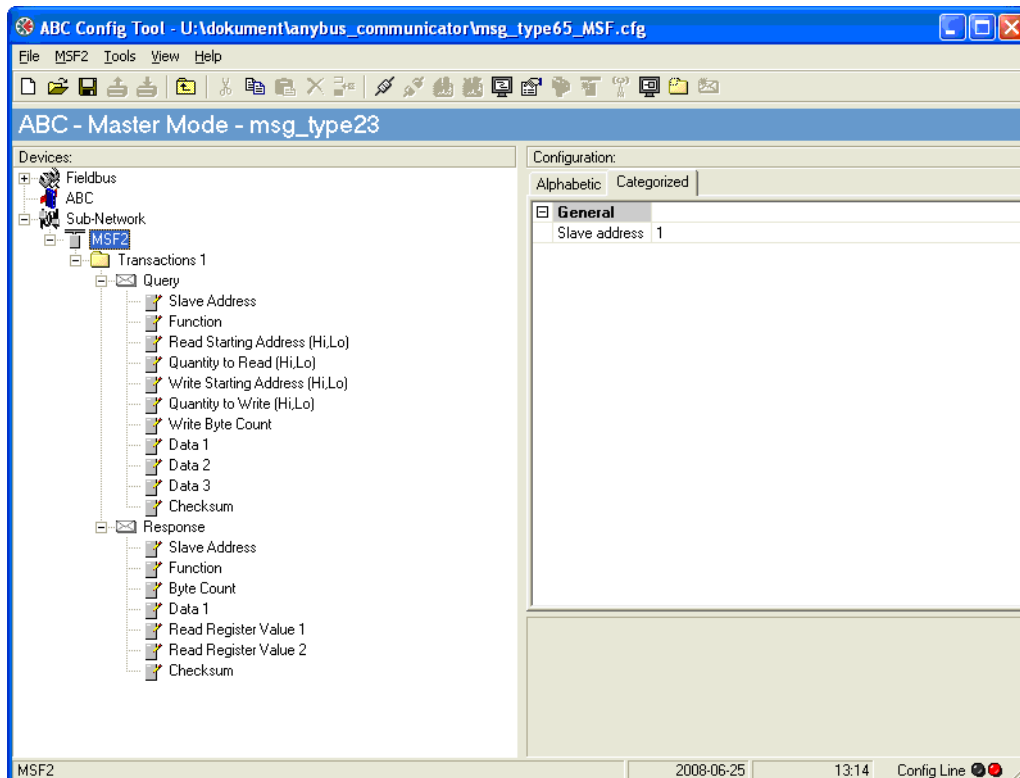

Fig. 5

Under "transactions" you can now fill in the following data (right click on the Query or Response field and add Byte constant/Word constant or Data according to the table below).

Query msg MSF

Table 1

Field name Query	Set to	Right click on query and add as...
Slave Address	Same as in MSF	Byte constant
Function	Set to 0x41 for MSF (Emotron specific function code for process data registers only)	Byte constant
Read Starting Address (Hi,Lo)	0x26F1	Word constant
Quantity to Read (Hi,Lo)	0x0003	Word constant
Write Starting Address (Hi,Lo)	0x26F1	Word constant
Quantity to Write (Hi,Lo)	0x0003	Word constant
Write byte count	0x06	Byte constant
Data 1	Data length 0x0002 Data location 0x0400 (This status reg must be read but is not mapped to process data area) Byte swap: Don't care	Data
Data 2	Data length 0x0002 Data location 0x0202 (optional) Byte swap: Swap 2 bytes (optional)	Data
Data 3	Data length 0x0002 Data location 0x0200 (optional) Byte swap: Swap 2 bytes (optional)	Data
Checksum	Use default values	Checksum

Response msg MSF

Table 2

Field name Response	Set to	Right click on response and add as...
Slave Address	Same as in MSF	Byte constant
Function	Set to 0x41 for MSF (Emotron specific function code for process data registers only)	Byte constant
Byte Count	0x06	Byte constant
Data 1	Data length 0x0002 Data location 0x0400 (This status reg must be read but is not mapped to process data area) Byte swap: Don't care	Data
Read Register Value 1	Data length 0x0002 Data location 0x0002 (optional) Byte swap: Swap 2 bytes (optional)	Data
Read Register Value 2	Data length 0x0002 Data location 0x0000 (optional) Byte swap: Swap 2 bytes (optional)	Data
Checksum	Use default values	Checksum

Setup sub network for FDU/VFX 2.0

First setup the VSD menus:

For detailed menu information, please check manuals listed above.

Setup the following menus:

[214] Reference control

[215] Run/Stp control

[216] Reset control

[261] Com type – Set to RS232/485

[2621] Baud rate

[2622] Address

NOTE: The exact same address, baud rate etc. should be configured for the gateway.

We will now continue our configuration of the gateway to setup a modbus rtu message that will read and write modbus registers containing the 4 byte field bus process data.
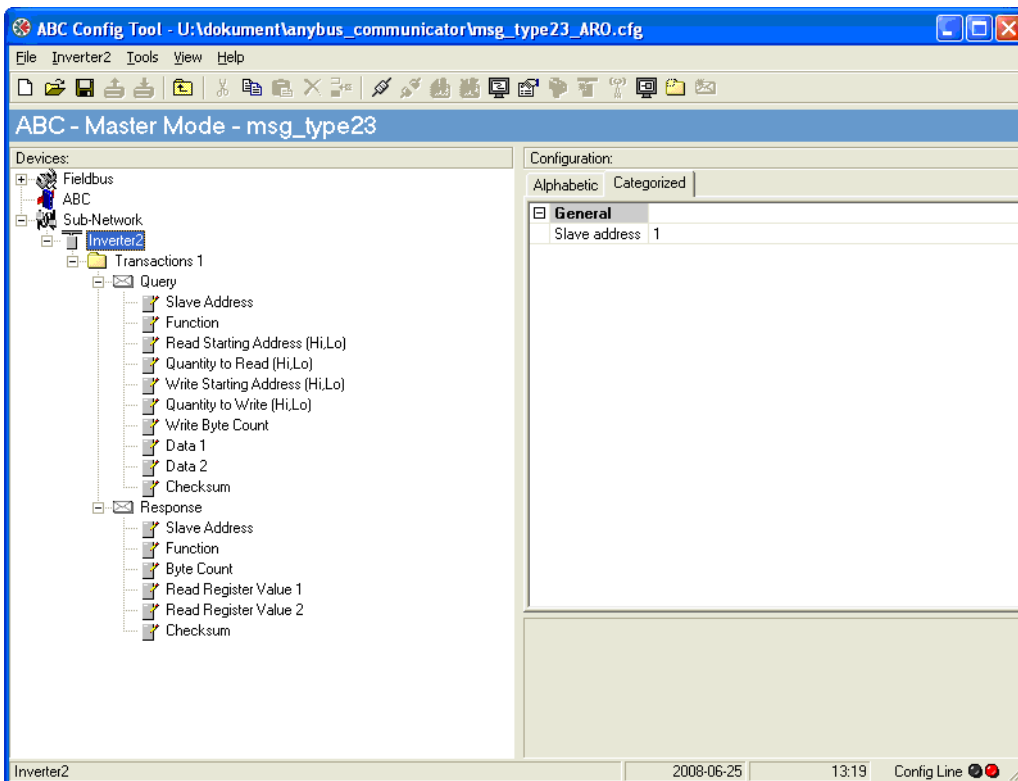

Fig. 6

Under "transactions" you can now fill in the following data (right click on Query or Response field and add Byte constant/Word constant or Data according to table below).

Query msg Inverter

Table 3

Field name Query	Set to	Right click on query and add as...
Slave Address	Same as in Inverter	Byte constant
Function	Set to 0x17 Read/Write Multiple registers	Byte constant
Read Starting Address (Hi,Lo)	0x26F2	Word constant
Quantity to Read (Hi,Lo)	0x0002	Word constant
Write Starting Address (Hi,Lo)	0x26F2	Word constant
Quantity to Write (Hi,Lo)	0x0002	Word constant
Write byte count	0x04	Byte constant
Data 1	Data length 0x0002 Data location 0x0202 (optional) Byte swap: Swap 2 bytes (optional)	Data
Data 2	Data length 0x0002 Data location 0x0200 (optional) Byte swap: Swap 2 bytes (optional)	Data
Checksum	Use default values	Checksum

Response msg Inverter

Table 4

Field name Response	Set to	Right click on response and add as...
Slave Address	Same as in Inverter	Byte constant
Function	Set to 0x17 Read/Write Multiple registers	Byte constant
Byte Count	0x04	Byte constant
Read Register Value 1	Data length 0x0002 Data location 0x0002 (optional) Byte swap: Swap 2 bytes (optional)	Data
Read Register Value 2	Data length 0x0002 Data location 0x0000 (optional) Byte swap: Swap 2 bytes (optional)	Data
Checksum	Use default values	Checksum

General hints

Speeding up the sub network

The speed on the sub network depends on several issues:

- Baud rate

- Telegram length/amount of data

- Number of slave nodes on the RS485 sub network (sharing bandwidth to master)

- Update time (i.e. minimum time between two transactions)

In general it is preferred to achieve a fast response. Try to keep the time between two transactions of a kind as short as possible. This can be altered by marking the Query to the left and altering the Update time (10 ms) value to the right, see Fig. 7.
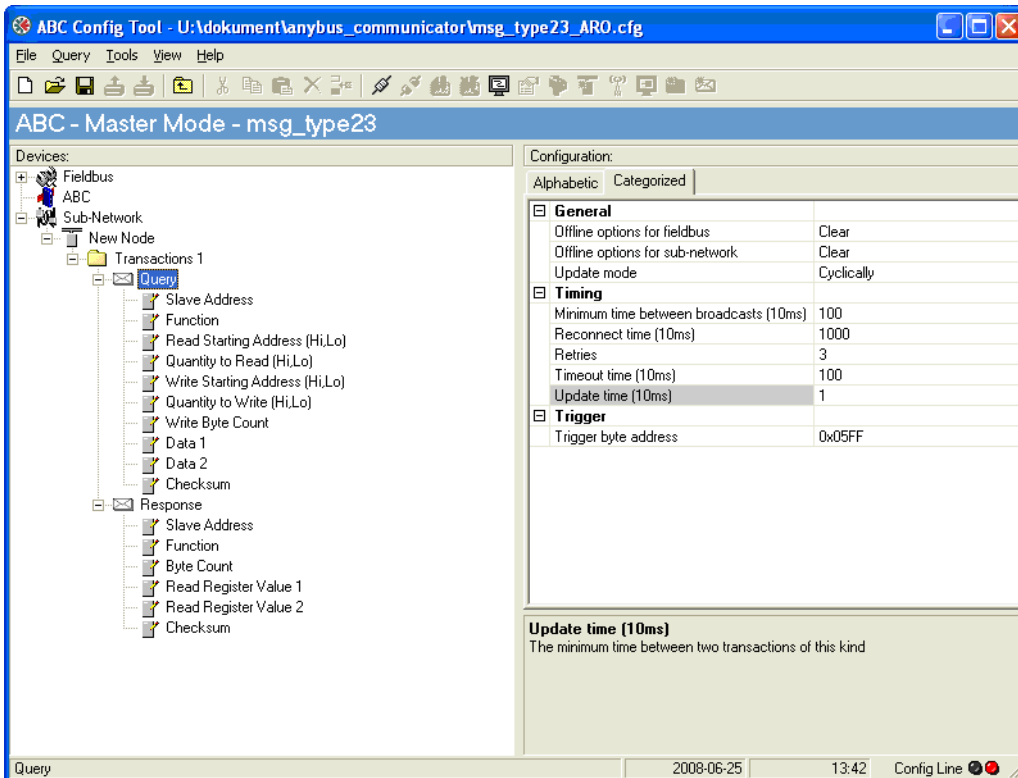

Fig. 7

Data location & byte swapping

PLC's of different brands may use different data formats (little endian or big endian). Depending on this you may have to perform byte swapping, either in the PLC or in the Gateway.

NOTE: The Data length value must be set to 0x0002 before you can alter the Byte swap setting.

You may also swap registers (words) if necessary by altering the Data location in the DP memory.

Example for Profibus:

The Anybus communicator gateway (Profibus DP) has been configured for a VSD FDU/VFX 2.0 exactly as in the example above. This means that the process data mapping will be exactly the same with the gateway as for the Fieldbus I/O board 2.0 option (standard Profibus option).
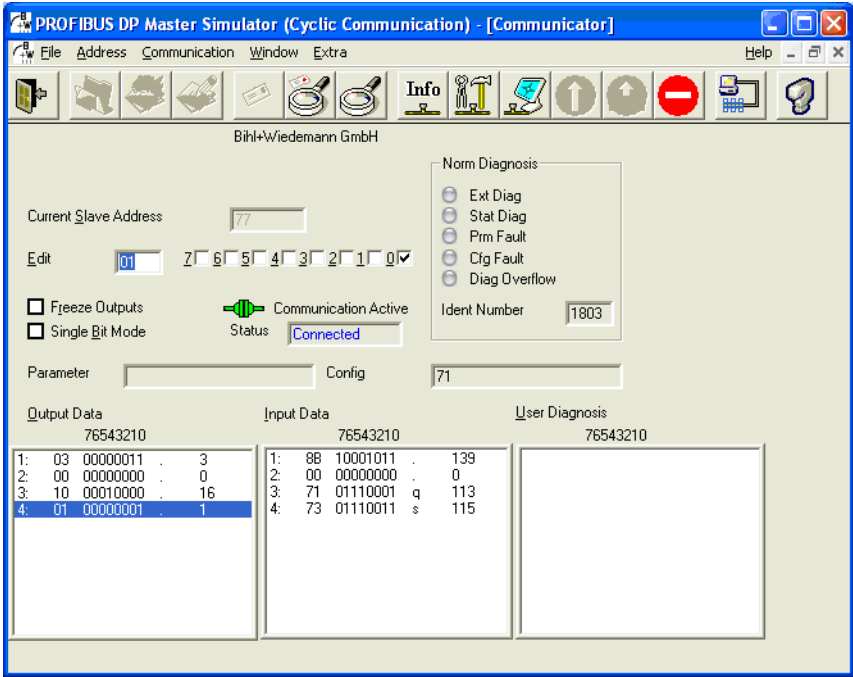

Fig. 8   Using a profibus DP Master simulator with 4 bytes of process data in/out.

The first output data
Byte 1: is set to 0x03=Run and Run to the right
Byte 2: Drive mode = speed, and parameter set A
Byte 3 and 4: Reference Signal.

According to Fig. 8, the first output data:
Byte 1: is set to 0x03=Run and Run to the right
Byte 2: is set to 0x00 which corresponds to Drive mode=speed, and parameter set A
Byte 3 and 4: Reference Signal.

Input data:
Byte 1: is 0x8B=1000 1011b=Running, run right, Relay 1 on , Run Ctrl=run/stp by com.
Byte 2: is 0x00=Drive mode speed, no warning/error status.
Byte 3: AnOut1 value
Byte 4: AnOut2 value.

Input data
Byte 1: is 0x8B = 1000 1011b = Running, run left, Relay1 on, Run Ctrl=1=run/stp by com.
Byte 2: is 0x00 = Drive mode speed, no warning/error status.
Byte 3: AnOut1 value
Byte 4: AnOut2 value.

The mapping is 1:1 compared to the Fieldbus I/O board 2.0 Option manual, i.e. same byte order.

In the same manner the MSF example above will give a 1:1 mapping according to the process data in the Emotron MSF 2.0 Fieldbus Option manual.

Network comparison chart

The example above is a configuration for a Profibus network. Profibus uses the Big endian format, which means that the most significant byte (MSB) comes first (on the lowest memory address).

If you use another network which uses the little endian format, you may change the byte swapping in the example above, either in the gateway or in your PLC.

Table 5   Example data format network comparison list

Network	Data format
Profibus	MSB first
Ethernet/IP	LSB first
CC-Link	LSB first
DeviceNet	LSB first
CANopen	LSB first
Modbus RTU	LSB first
Modbus/TCP	LSB first