



***CAN_AC_PCI - Driver of CANBUS
Protocol for CAN ACx PCI Card
User's Manual***

Doc. No. ENP4011
Version: 29-08-2005

ASKOM[®] and **asix**[®] are registered trademarks of ASKOM Spółka z o.o., Gliwice. Other brand names, trademarks, and registered trademarks are the property of their respective holders.

All rights reserved including the right of reproduction in whole or in part in any form. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without prior written permission from the ASKOM.

ASKOM sp. z o. o. shall not be liable for any damages arising out of the use of information included in the publication content.

Copyright © 2005, ASKOM Sp. z o. o., Gliwice



ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,
tel. +48 (0) 32 3018100, fax +48 (0) 32 3018101,
<http://www.askom.com.pl>, e-mail: office@askom.com.pl

1. CAN_AC_PCI - Driver of CANBUS Protocol for CAN ACx PCI Card

1.1. Driver Use

The CAN_AC_PCI driver is used for data exchange between SELECONTROL MAS PLCs of Selectron Lyss AG and **asix** system computers by using the CAN network. The **asix** system computer must be provided with a card of CAN_AC1 or CAN_AC2 communication processor of Softing GmbH.

1.2. Declaration of Transmission Channel

The full syntax of declaration of transmission channel operating according to the CAN_AC_PCI driver is given below:

logical_name=CAN_AC_PCI,*interface_no*

where:

interface_no - number identifying the CAN_AC1/CAN_AC2 card interface by means of which the transmission with the CAN network is executed. In the CAN_AC1 card the interface no. 1 may be used exclusively.

The CAN_AC_PCI driver is loaded as a DLL automatically.

1.3. Addressing the Process Variables

Values of process variables are transferred in telegrams sent by CAN based controllers connected to a CAN network. Each telegram consists of at most 8 bytes that may be identified as:

- bytes indexed 1 – 8 (type BY),
- 16-bit numbers indexed 1 – 4 (type WD),
- 32-bit numbers indexed 1 – 2 (type DW).

The CAN_AC_PCI driver distinguishes the following types of access to process variables:

- read-only (type R_),
- write-only (type W_),
- read/write (type RW_).

The addressing of process variables consists in indication of:

- access type (R_, W_ or RW_);
- variable type (BY, WD, DW);
- telegram no. (for variables with RW_ access type it is the telegram number used to read the variable),
- index within the telegram (for variables with RW_ access type it is the index in the telegram used to read the variable),
- for variables with RW_ access type it is necessary to declare in addition:
 - a/ telegram no. used to read the variable,
 - b/ index in the telegram used to write variable.

The syntax of symbolic address which is used for variables belonging to the CAN_AC_PCI driver channel is as follows:

<access_type><variable_type><tel>.<index>[.<tel>.<index>]

where:

<i>access_type</i>	- access type to the process variable:
R_	- read-only,
W_	- write-only,
RW_	- read/write,
<i>variable_type</i>	- process variable type:
BY	- variable of the byte type,
WB	- variable of the 16-bit number type,
DW	- variable of the 32-bit number type,
<i>tel</i>	- telegram no.,
<i>index</i>	- index within the telegram.

EXAMPLE

X1, byte no 2 of telegram 31,	R_BY31.2,	NONE, 1, 1, NOTHING_BYTE
X2, word no 3 of telegram 31,	R_WD31.3,	NONE, 1, 1, NOTHING
X3, burner state,	RW_BY31.1.35.3,	NONE, 1, 1, NOTHING_BYTE
X4, valve setting,	RW_WD32.1.34.1,	NONE, 1, 1, NOTHING

Value of the variable X3 is transferred to the **asix** system by means of the byte no. 1 of the telegram no. 31. The transfer of the variable X3 consists in sending from **asix** a telegram no. 35, the byte no. 3 of which includes the required state of the variable X3.

1.4. Driver Configuration

The CAN_AC_PCI protocol driver may be configured by using the [CAN_AC_PCI] section of the application INI file. Individual parameters are given in separate items of the section. Each item has the following syntax:

item_name=[number [,number]] [YES] [NO]



TRANSMISSION_SPEED=interface_no,baud_id

Meaning - the item is used to declare a transmission speed in the CAN network.

Default value - by default, the transmission speed is assumed to be 1 MB.

Parameter:

interface_no - interface no. of the CAN_AC card (for the CAN_AC1 card always 1),

baud_id - identifier of transmission speed in the CAN network:

- 1 - 1 MB
- 2 - 500 kB
- 3 - 250 kB
- 4 - 125 kB
- 5 - 100 kB
- 6 - 50 kB
- 7 - 20 kB

EXAMPLE

An example of declaration of transmission speed of 20 kB (the CAN network numbered 1):

TRANSMISSION_SPEED=1,7



REFRESH_CYCLE=number

Meaning	- the item is used to declare a time interval between two successive signals allowing the CAN_AC card driver to read data from the CAN network.
Default value	- by default, the CAN_AC_PCI driver reads data every 0.5 second.
Parameter: <i>number</i>	- number of 0.5-second intervals, which must pass between two successive signals allowing the CAN_AC board driver to read data from the CAN network.

EXAMPLE

An example of declaration of data reading every 1 second:

REFRESH_CYCLE=2



NETWORK_CONTROL=number

Meaning	- the item allows to test the reception of telegrams from the CAN network. It defines the maximal time (in seconds) between reception of successive telegrams with the same number. In case of exceeding this time the process variables from such telegram will be provided with an error status. If additionally in the same time any telegram wasn't received from the CAN network the message about a lack of telegrams in the network is generated in ' <i>Control Panel</i> '.
Default value	- by default, the CAN_AC_PCI driver doesn't check reception of telegrams.
Parameter: <i>number</i>	- maximal number of seconds, which may pass between successive telegrams of the same number.

EXAMPLE

An example of checking the reception of telegrams every 5 seconds:

NETWORK_CONTROL=5



TELEGRAM_TRACE=YES/NO

Meaning	- the item controls transferring to the operator panel the messages about telegrams that have been received from the CAN network. A message includes the number of CAN network, the number of telegram, number of bytes and telegram contents in hexadecimal form.
---------	--

Default value - by default, the contents of telegrams are not displayed.



CONTROL_TRACE=YES/NO

Meaning - the item controls transferring to the operator panel the messages about control telegrams that have been sent from the **asix** system computer to controllers. A message includes the number of CAN network, the number of telegram, number of bytes and telegram contents in hexadecimal form.

Default value - by default, the contents of telegrams are not displayed.



LOG_FILE=file_name

Meaning - the item allows to define a file, to which all messages, describing the telegrams received from the CAN network, will be written. If LOG_FILE does not define the full path, then the log file will be created in the current directory. The log file should be used only while the **asix** start-up.

Default value - by default, the log file is not created.



USE_CONTROL_VALUE=YES/NO

Meaning - the driver has two pools of telegrams: sending and receiving ones. The sending telegrams are used by **asix** to send control data whereas the receiving telegrams contain actual copies of telegrams sent from controllers and are a source of values of process variables for **asix**. USE_CONTROL_VALUE allows to copy a value of the process variable (of W type) from a sending telegram directly to a buffer of the receiving telegram. The copying concerns the receiving telegram with the same number as the sending telegram assigned to the control variable and is executed only after that have done the control operation correctly. In this way the values of process variables are used by driver as actual values of process variables. This state lasts up to the moment the real values of process variables under consideration will be read from a controller.

The CONTROL_VAR_CHECK item allows to change the copying mode of control values.

Default value - by default, the control values are not copied to the buffers of receiving telegrams of the driver.

Checking Control Variables

The declaration of W_ type variables (control variables) are checked by default. It is possible to use only one variable of this type in one telegram.



CONTROL_VAR_CHECK =YES/NO

Meaning - the item permits to change default settings and enables using telegram to send control value by means of more than one variable of W_ type. Individual control values are send sequentially, i.e. by

sending a separate telegram transferring a value of one control variable only, the other parts of the telegram are filled in with zeroes.

Change of Default Settings for Updating Mode of Receiving Buffers by Means of Control Variables



USE_RW_DECLARATION=YES/NO

Meaning	- the item allows to change the mode of copying control variables to buffers of driver receiving telegrams. The item is import for W_ type variables only, and is effective if it is used together with the CONTROL_VAR_CHECK=YES item. The result of using the item under consideration is copying the control variable to the buffer of receiving telegram (with the number specified in the process variable declaration in the position of telegram that is used to read the variable value).
Default value	- by default, the driver buffers are not updated according to declarations of RW_ type variables.

1.	CAN_AC_PCI - DRIVER OF CANBUS PROTOCOL FOR CAN ACX PCI CARD	3
1.1.	DRIVER USE.....	3
1.2.	DECLARATION OF TRANSMISSION CHANNEL	3
1.3.	ADDRESSING THE PROCESS VARIABLES.....	3
1.4.	DRIVER CONFIGURATION.....	4