



***CtSNPX - Driver of SNPX Protocol for GE  
Fanuc PLCs  
User's Manual***

Doc. No. ENP4055  
Version: 29-08-2005

**ASKOM**<sup>®</sup> and **asix**<sup>®</sup> 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](mailto:office@askom.com.pl)

# 1. CtSNPX - Driver of SNPX Protocol for GE Fanuc PLCs

## 1.1. Driver Use

The driver of the SNPX protocol (Series Ninety Protocol) is used for data exchange between **asix** system computers and GE\_FANUC 90-30 PLCs as well as GE\_FANUC 90 CMM and PCM modules. The communication is executed by means of serial links.

The driver allows access to the following controller variable types:

- Discrete Inputs (%I),
- Discrete Outputs (%O),
- Discrete Internals (%M),
- Analog Inputs (%AI),
- Analog Outputs (%AO),
- Registers (%R),
- Genius Global Data (%G).

The driver does not handle the following controller variable types (system and temporary types):

- %SA Discrete,
- %SB Discrete,
- %SC Discrete,
- %S Discrete,
- Discrete Temporary (%T).

## 1.2. Declaration of Transmission Channel

The syntax of declaration of transmission channel operating according to the SNPX protocol is as follows:

```
Channel=UNIDRIVER, CtSNPX, Port=number; [Baudrate=number;]
[ParityBit=check_parity_name;]
[TimeSynchrX=address[:period];] [T4=timeout_break;] [T2=response_timeout;]
[TBroadCast=timeout_broadcast]
```

where:

UNIDRIVER	- universal <b>asix</b> system driver;
CtSNPX	- driver name;
Port	- number of the COM serial port;
BaudRate	- transmission speed between the computer and the controller; there are the following acceptable values: 300, 600,1200,2400, 4800, 9600, 19200 Bd; a default value – 19200 Bd;
ParityBit	- determines the method of frame parity check; there are the following acceptable values: <i>NONE</i> , <i>ODD</i> , <i>EVEN</i> ; default value – ODD (odd parity check);
T4	- timeout (in milliseconds) between sending BREAK and BROADCAST ATTACH; a default value – 50 milliseconds;

<i>TBroadcast</i>	- timeout between sending BROADCAST ATTACH and sending the first request to the controller; a default value – 2000 milliseconds;
<i>T2</i>	- timeout (in milliseconds) for receiving the first bit of the response; a default value – 2000 milliseconds;
<i>SynchrCzasuX</i>	- cyclic (every <i>period</i> in seconds) date & time frame write at the given address in the controller; there are 99 positions of time synchronization from the range of names from <i>SynchrCzasu1</i> to <i>SynchrCzasu99</i> ; if the parameter <i>period</i> is not given, the synchronization is performed every 60 seconds by default; the date & time frame synchronization is compatible with <i>SVCREQ 7</i> – the procedure of date & time write:

```

struct    dateTime
{
    byte    Year;
    byte    Month;
    byte    Day;
    byte    Hour;
    byte    Minute;
    byte    Second;
    byte    DayOfWeek;
    byte    NotUsed;    // always 0
    word    wSynchr;    // set to 1 at new date & time
frame write
};

```

**NOTICE**

*The parameters given in the channel declaration must be compatible with the parameters set for communication ports of the controllers handled by this channel.*

**EXAMPLE**

An exemplary declaration of the channel, in which the controllers with identifiers A123 and B456 are handled, is given below:

- 1/ for the controller with the A123 identifier - time is synchronized by writing to the register area beginning with R10 (every 25 seconds),
- 2/ for the controller with the B456 identifier - time is synchronized by writing to the register area beginning with R10 (with default frequency).

The communication with the controllers is performed over COM2 by means of standard transmission parameters, i.e. 19200 Bd, odd parity control, the first bit of stop and standard timeouts of the SNPX protocol.

```

CHANNEL = UNIDRIVER, CtSNPX, Port=2; TimeSynchr1=A123.R10:25;
TimeSynchr2=B456.R20

```

### 1.3. Declaration of Variables

The driver makes the following variable types available:

I	- Discrete Input (%I) in BIT mode,
IB	- Discrete Input (%I) in BYTE mode,
IW	- Discrete Input (%I) in WORD mode,
Q	- Discrete Output (%I) in BIT mode,

QB	- Discrete Output (%I) in BYTE mode,
QW	- Discrete Output (%I) in WORD mode,
M	- Discrete Internal (%I) in BIT mode,
MB	- Discrete Internal (%I) in BYTE mode,
MW	- Discrete Internal (%I) in WORD mode,
G	- Genius Global Data (%G) in BIT mode,
GB	- Genius Global Data (%G) in BYTE mode,
GW	- Genius Global Data (%G) in WORD mode,
AI	- Analog Input (%AI) in WORD mode,
AO	- Analog Output (%AO) in WORD mode,
R	- Register (%R) treated as WORD,
RL	- two following Registers (%R) treated as DWORD,
RF	- two followin Registers (%R) treated as FLOAT,

The syntax of the variable address is as follows:

[<CpuID>.<Type><Index>

where:

<i>CpuID</i>	- CPU identifier;
<i>Type</i>	- variable type name;
<i>Index</i>	- variable address within the framework of the variable type <i>Type</i> .

#### NOTICE

1. *CpuID* may be omitted in the variable address exclusively when only one controller is connected to the serial link. In such case, commands sent to the controller contain the identifier set at NULL (the real identifier set in the controller is unimportant).

2. *Index* indicates bit number, from which the range of bits escribed to the variable begins, for discrete variables. Index may have one of the following values (by pattern of addressing used in VersaPro), depending on mode of making discrete variables available.

- any value w.e.f. 1,
- value 1, 9, 17, i.t.d (numbers of the first bit of successive bytes),  
c/ for WORD mode - wartości 1, 17, 33, i.t.d (numbers of the first bit of successive words).

#### EXAMPLE

An examplary variable declaration (the variable values come form the controllers identified by A123 and B456):

```
JJ_01, Rejestr R3,      A123.R3,      CHANNEL, 1, 1, NOTHING
JJ_02, Analog Input 1,  A123.AI1,     CHANNEL, 1, 1, NOTHING
JJ_03, Discrete Input 3, B456.I3,     CHANNEL, 1, 1, NOTHING
JJ_04, Discrete Output Byte 9 , A123.QB9, CHANNEL, 1, 1, NOTHING_BYTE
JJ_05, Genius Global Word 17 ,      A123.GW17, CHANNEL, 1, 1, NOTHING
JJ_06, Discrete Internal Word 33,    B456.MW33, CHANNEL, 1, 1, NOTHING
```

## 1.4. Driver Configuration

The driver configuration is performed by using the separate section named [CTSNPX]. By means of this section it is possible to declare:

- log file and its size,
- log of telegrams.

***LOG\_FILE=file\_name***

Meaning - the item allows to define a file to which all the diagnostic messages of the driver will be written.  
Default value - by default, the log file is not created.

***LOG\_FILE\_SIZE =number***

Meaning - the item allows to define the size of the log file in MB.  
Default value - by default, the item assumes that the log file has a size of 1 MB.  
Parameter:  
*number* - size of the log file in MB.

***LOG\_OF\_TELEGRAMS =YES / NO***

Meaning - the item allows writing to the log file (declared with use of the LOG\_FILE item) the contents of telegrams transmitted during the data exchange between the **asix** system and controllers.  
Default value - NO.

#### **EXAMPLE**

An exemplary driver section:

```
[CTSNPX]
LOG_FILE=d:\tmp\ctLG\LG.log
LOG_FILE_SIZE=3
LOG_OF_TELEGRAMS=YES
```

<b>1.</b>	<b>CTSNPX - DRIVER OF SNPX PROTOCOL FOR GE FANUC PLCS</b>	<b>3</b>
1.1.	DRIVER USE	3
1.2.	DECLARATION OF TRANSMISSION CHANNEL	3
1.3.	DECLARATION OF VARIABLES	4
1.4.	DRIVER CONFIGURATION	5