



CtSi400 - *Driver of Protocol for Sintony Si 400 Alarm
Central of SIEMENS*

User's Manual

*Doc. No. ENP4067
Version: 05-07-2006*

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 company.

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 © 2006, 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

Table of Contents

1. CTSI400 - DRIVER OF PROTOCOL FOR SINTONY SI 400 ALARM CENTRAL OF SIEMENS	5
1.1. DRIVER USE.....	5
1.2. DECLARATION OF TRANSMISSION CHANNEL.....	5
1.3. ADDRESSING PROCESS VARIABLES.....	6
1.4. TYPES OF VARIABLES USED TO MONITOR.....	6
1.5. TYPES OF VARIABLES USED TO CONTROL.....	8
1.6. TIME STAMP.....	10
1.7. DRIVER CONFIGURATION	10

1. CtSi400 - Driver of Protocol for Sintony Si 400 Alarm Central of SIEMENS

1.1. Driver Use

The CtSi400 driver is used for data exchange between **asix** computers and a Sintony Si 400 alarm central of SIEMENS. The communication is executed by means of serial links in the RS-232 standard.

Additional recommendations:

For proper operation of the alarm system it is demanded to connect only one visualization computer to a Sintony central (despite the central is equipped with 3 interfaces, during the system operation it is possible to use only one of them). The best solution is to connect a computer to the J7 interface with use of a SAQ11 cable recommended by SIEMENS.

1.2. Declaration of Transmission Channel

The CtSi400 driver is loaded by the universal **asix** system driver - UNIDRIVER.

The syntax to declare the transmission channel operating with the CtSi400 driver is given below:

```
Channel=UNIDRIVER, CtSi400, Port=port_nr [; BaudRate=baud]  
[; Timeout=ms_number] [; AlarmOffset=offset] USERID=user_id
```

where:

UNIDRIVER	- the universal asix system driver;
CtSi400	- the driver used for communication with a SINTONY SI 400 central;
<i>port_nr</i>	- serial port number (for COM1 it is passed 1, for COM2 it is passed 2, etc.);
<i>baud</i>	- transmission speed passed in bauds. By default, it is assumed 9600 Bd;
<i>ms_number</i>	- timeout of waiting for the controller response (in milliseconds); it is passed 1000 milliseconds by default;
<i>offset</i>	- offset added to the number of each alarm sent from a SINTONY SI 400 central;
<i>user_id</i>	- user identifier sent to SINTONY SI 400 in control frames. Proper values are included in the range 0-500.

By default, it is assumed:

- number of character bits - 8,
- without parity check (NONE),
- number of stop bits - 1.

EXAMPLE

The exemplary declaration of the channel using CtSi400 on the serial port COM2 with the receiving timeout that equals 2000 ms and the offset, added to each alarm number, equaling 1500 is as follows:

PLC1 = UNIDRIVER, CTSI400, Port=2; Timeout=2000; OffsetAlarm=1500

1.3. Addressing Process Variables

The syntax of the symbolic address for all variables used to monitor the central state is given below:

<type><index>

and for variables used to control:

<type>

or

<type><index>

or

<type><p_index>.<r_index>

where:

type	- variable type,
index	- index within the type,
p_index	- partition number (0-15),
r_index	- room's number within the partition (0-7).

1.4. Types of variables used to monitor

There are several types of variables used to monitor, which are listed below. The driver permits readout operations to be performed on these variables, while all write operations end with the OPC_E_BADRIGHTS error.

The bit 0 means the least significant bit in the following list.

- **IS<index>** - input status

PState	- bits 7-6
Lstate	- bits 5-3
Reserved	- bits 2-0

- **IP<index>** - input parameters:

PSL	- bits 15-14
V.Address	- bits 13-8
Type	- bits 7-4,
SubType,	- bits 3-1

- **IA<index>** - input address

room nr - bits 12-9
 partition nr - bits 4-0

- **IN<index>** - input name

- **OS<index>** - output status

State - bit 7
 OutInTest - bit 6
 BlinkBit - bit 5
 NbrPulse - bit 4
 Reserved - bits 3-0

- **OP<index>** - output parameters

Ltype - bits 7-0

- **OA<index>** - output address

room nr - bits 12-9
 partition nr - bits 4-0

- **ON<index>** - output name

- **RP<index>** - room parameters

Rset - bits 15-8 (room1 is in the bit no. 15, room2 is in the bit no. 14, etc.)
 Rarm - bits 7-0 (room1 is in the bit no. 7, room2 is in the bit no. 6, etc.)

- **RS<index>** - room status

room1 - bits 15-14
 room2 - bits 13-12

etc.

- **PS<index>** - partition status

TAMP - bit 15
 PA - bit 14
 Fire - bit 13
 AM - bit 12
 Chime - bit 11
 Mess - bit 10
 Reserved - bits 9-8
 Pstate - bits 7-6
 PFA - bit 5
 PPA - bit 4
 PFS - bit 3
 PPS - bit 2
 PUS - bit 1
 BA - bit 0

- **SS<index>** - system status

ParamCh	- bit 15
Evt Overflow	- bit 14
Avzone	- bits 13 - 8
Mains	- bit 7
Batt	- bit 6
Fuse	- bit 5
Line	- bit 4
CMS1	- bit 3
CMS2	- bit 2
InComm	- bit 1
Evt queue	- bit 0

1.5. Types of variables used to control

There are several types of variables used to control, which are listed below. The driver permits write operations to be performed on these variables, while all readout operations end with the OPC_E_BADRIGHTS error.

- **cIB<index>** - input bypass (index = 0..0xFFFF)

Writing any value to the variable will cause the performance of the command for the input with the *index* number.

- **cIS<index>** - input in soak test (index = 0..0xFFFF)

Writing any value to the variable will cause the performance of the command for the input with the *index* number.

- **cOT<index>** - set output in test mode (index = 0..0xFFFF)

Writing any value to the variable will cause the performance of the command for the output with the *index* number.

- **cTO<index>** - toggle output (index = 0..0xFFFF)

Writing any value to the variable will cause the performance of the command for the output with the *index* number.

- **cPP<p_index>** - partition part-set (p_index = 0..0xF)

Writing any value to the variable will cause the performance of the command for the partition with the *p_index* number.

- **cPF<p_index>** - partition full-set (p_index = 0..0xF)

Writing any value to the variable will cause the performance of the command for the partition with the *p_index* number.

- **cPU<p_index>** - partition unset (p_index = 0..0xF)

Writing any value to the variable will cause the performance of the command for the partition with the *p_index* number.

- **cCA<p_index>** - clear alarm memory in partition (p_index = 0..0xF)

Writing any value to the variable will cause the performance of the command for the partition with the *p_index* number.

- **cRF<p_index>.<r_index>** - room full-set (*p_index* = 0..0xF, *r_index* = 0..0xF)

Writing any value to the variable will cause the performance of the command for the room with the *r_index* number from the partition with the *p_index* number.

- **cRU<p_index>.<r_index>** - room unset (*p_index* = 0..0xF, *r_index* = 0..0xF)

Writing any value to the variable will cause the performance of the command for the room with the *r_index* number from the partition with the *p_index* number.

- **cSR<p_index>** - partition reset sounders/bells (*p_index* = 0..0xF)

Writing any value to the variable will cause the performance of the command for the partition with the *p_index* number.

- **cIP<index>** - input activation pulse (*index* = 0..0xFFFF)

Writing any value to the variable will cause the performance of the command for the input with the *index* number.

- **cIL<index>** - input activation latch (*index* = 0..0xFFFF)

A written value must convert to a double type. If the value 0 is received after conversion, for the input with the *index* number there will be performed the command with the parameter *commutation level* equaling 0; otherwise the parameter will be equal to 1.

- **cAV** - new current A/Video zone

A written value must be an integer from the range <0; 32>. Writing the value 0 will activate the *autoswitch* option.

- **cTC** - transmission to CMS

A written value must have the value 1 or 2 and indicate CMS1 or CMS2 suitably.

- **cER** - reset the pending event buffer

Writing any value to the variable will cause the performance of the command.

IN and ON variables are 16-element BYTE type arrays. Other variables are WORD type values.

All the variables used to control are of the WORD type.

EXAMPLE

Exemplary variable declarations:

JJ_00, input status 1,	IS1,	PLC1, 1, 1, NIC
JJ_01, output status 10,	OS10,	PLC1, 1, 1, NIC
JJ_02, nazwa input 2,	IN2,	PLC1, 16, 1, NIC_BYTE
JJ_03, system status,	SS1,	PLC1, 1, 1, NIC
cIB3, input 3 bypass,	cIB3,	PLC1, 1, 1, NIC
cRU1.2, part 1 room 2 unset,	cRU1.2,	PLC1, 1, 1, NIC
cAV, current A/V zone,	cAV,	PLC1, 1, 1, NIC

1.6. Time Stamp

All values of variables read from a SINTONY SI 400 central are stamped with the local PC time.

1.7. Driver Configuration

The driver configuration is performed by using the separate section named **[CTSI 400]**. By means of this section it is possible to declare:

- log file and its size,
- log of telegrams,
- delay between opening a session and data exchange.



LOG_FILE = file_name

Meaning - the item allows to define a file to which all diagnostic messages of the driver will be written. If the item does not define the full path, then the log file is created in a current directory. The log file should be used only while the **asix** system start-up.

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

Defining - manual.



LOG_FILE_SIZE = number

Meaning - allows to specify the log file size.

Default value - 1MB.

Defining - manual.

Parameter:

number - number in MB.



LOG_OF_TELEGRAMS=[YES/NO]

Meaning - item allows to write to the log file (declared by use of the item LOG_FILE) the contents of telegrams received by the driver. Writing the contents of telegrams to the log file should be used only in the stage of the **asix** system start-up.

Default value - NO.

Defining - manual.

***DELAY_AFTER_OPEN_SESSION=number***

- Meaning - allows to define a timeout between opening a session and sending the first query about data within this session.
- Default value - by default, the value is set to 100.
- Defining - manual.
- Parameter:
number - timeout value in milliseconds.