



***CtPmc4000 – POLON 4800 Fire Protection Station Driver
According to Protocol PMC-4000
User's Manual***

*Doc. No. ENP5068
Version: 01-02-2008*

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 © 2008, 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. CtPmc4000 – POLON 4800 fire protection station driver according to protocol PMC-4000

- | | |
|---|---|
| <input type="checkbox"/> Driver designation | <input type="checkbox"/> Driver parameters |
| <input type="checkbox"/> Transmission channel declaration | <input type="checkbox"/> Log file |
| <input type="checkbox"/> Example | <input type="checkbox"/> log file size |
| <input type="checkbox"/> Variable declaration | <input type="checkbox"/> telegram log |
| <input type="checkbox"/> Example | <input type="checkbox"/> Example of driver section: |

Driver designation

The CtPmc4000 protocol driver is used for data exchange between the **asix** system and POLON 4800 fire protection station according to protocol PMC-4000. The communication is executed with the use of a serial link in RS-232 standard.

The parameterization of the CtPmc4000 driver is performed by the Architect application.

Transmission channel declaration

The declaration of the transmission channel operating according to CtPmc4000 protocol requires a channel with the following parameters to be added to the *Current data* module:

Standard tab:

Name: logical name of the transmission channel
Driver: CtPmc4000

CtPmc4000 tab:

Channel parameters:
Port=number; StationNr=number; Baud=number

where:

<i>Port</i>	- computer's COM serial port number,
<i>StationNo</i>	- station number,
<i>Baud</i>	- transmission speed (2400, 4800 or 9600 Bd). By default, 9600 Bd.

EXAMPLE

An example of the declaration of the transmission channel communicating with the fire protection station numbered 2 with the use of the COM2 serial port at the default transmission speed (9600 Bd):

Name: CHANNEL
Driver: CtPmc4000
Channel parameters: Port=2; StationNo=2

Variable declaration

The variable address has the following syntax:

V.<LineNo>.<ElemNo>.<IONo>.<ZoneNo>[.<baseAlarmNo>]

where:

<i>V</i>	- variable name,
<i>LineNo</i>	- line number,
<i>ElemNo</i>	- given line element number,
<i>IONo</i>	- input/output number of the given element,
<i>ZoneNo</i>	- zone number,
<i>AlarmNo</i>	- base alarm number.

All variables return values of WORD type.

Line, element, IO and zone numbers are determined by the fire protection station configuration.

The variable value is the bitmap of the fire protection station's element status whose address was given in the variable declaration. Bits have the following meaning:

bit 0	- fire alarm status;
bit 1	- control status;
bit 2	- test status;
bit 3	- blocking status;
bit 4	- technical alarm status;
bit 5	- failure status;
bit 6	- unmaskable failure status;
bits 7 – 15	- are not used.

If the bit value is 1, the given bit is active.

For each variable the user can define a set of alarms assigned to specific variable bitmap bits. Alarms are reported, as soon as the status of any bit is changed. To guarantee the correct driver operation, the aforementioned alarms should occupy a coherent alarm number area, starting from *baseAlarmNo*. Subsequent alarm numbers, starting from *baseAlarmNo*, have the following meanings:

+ 0	- fire alarm;
+ 1	- control performed;
+ 2	- test;
+ 3	- blocking;
+ 4	- technical alarm;
+ 5	- failure;
+ 6	- unmaskable failure;

Asix alarms are reported according to the following rule: **start** (change from 0 to 1) and **end** (change from 1 to 0). The time at which the alarm occurred is marked with a fire protection station timestamp. It is not possible to automatically pass on the alarm text sent from the fire protection station to the **asix** alarm. It is recommended that the alarm texts defined in the **asix** system should be compliant with the alarm texts defined in the fire protection station.

Specific supported variables

The variables of the following addresses are specifically supported:

V.0.0.0.0.baseAlarm (line no. 0)

and

V.1.0.0.0.baseAlarm (line no. 1)

The status of **V.0.0.0.0.baseAlarm** contains information about the 2nd level alarm status.

The bits have the following meaning:

bit 0 - 2nd level alarm status;

The base alarm number of the variable in question is used for generating the following alarms:

+ 0 - 2nd level alarm signaling;
 + 1 - receipt of DELETE information from the station,
 + 2 - receipt of CONFIRMATION information from the station,

The variable can be used to send orders to the fire protection station:

- recording the value 1 to the variable results in sending a DELETE order to the station,
 - recording the value 2 to the variable results in sending a CONFIRMATION order to the station,

V.1.0.0.0.baseAlarm variable status is used to pass on alarms about station failures.

The base alarm number of the variable is used to generate 107 alarms according to the specification given in the station failure code table of POLON unit ("PMC-4000 – Digital monitoring protocol for POLON 4800 standard"). The first alarm in the specification will be assigned to *baseAlarm* number, the second one – to *baseAlarm + 1*, and so on.

The variable value is accidental.

EXAMPLES

Examples of variables declaration:

```
# variable used to display the 2nd level alarm status and to send orders
# DELETE and CONFIRMATION. It generates alarms of the numbers starting
from 1
```

```
JJ_01, , V.0.0.0.0.1, PMC, 1, 1, NIC
```

```
# line 1001, element 2, I/O 3, zone 4, alarms starting from number 10
```

```
JJ_02, , V.1001.2.3.4.10, PMC1, 1, 1, NIC
```

```
# line 3002, element 4, I/O 0, zone 0, alarms starting from number 20
```

```
JJ_03, , V.3002.4.0.0.20, PMC1, 1, 1, NIC
```

```
# variable used to pass alarms about station failures:
```

```
JJ_04, , V.1.0.0.0.1000, PMC, 1, 1, NIC
```

Alarm definitions for JJ_01 variable:

1, al, 2nd level fire alarm

- 2, al, information on deleting fire alarm from station
- 3, al, information on confirmation of event or failure

Alarm definitions for JJ_02 variable:

- 10, al, fire alarm - line 1001 elem 2 io 3 zone 4
- 11, al, control - line 1001 elem 2 io 3 zone 4
- 12, al, test - line 1001 elem 2 io 3 zone 4
- 13, al, blocking - line 1001 elem 2 io 3 zone 4
- 14, al, technical alarm - line 1001 elem 2 io 3 zone 4
- 15, al, failure - line 1001 elem 2 io 3 zone 4
- 16, al, unmaskable failure - line 1001 elem 2 io 3 zone 4

Alarm definitions for JJ_03 variable:

- 20, al, fire alarm - line 3002 elem 4 io 0 zone 0
- 21, al, control - line 3002 elem 4 io 0 zone 0
- 22, al, test - line 3002 elem 4 io 0 zone 0
- 23, al, blocking - line 3002 elem 4 io 0 zone 0
- 24, al, technical alarm - line 3002 elem 4 io 0 zone 0
- 25, al, failure - line 3002 elem 2 io 0 zone 0
- 26, al, unmaskable failure - line 3002 elem 2 io 0 zone 0

Alarm definitions for JJ_04 variable:

- 1000, al, uP1 microprocessor system
- 1001, al, uP2 microprocessor system
- 1005, al, uP1 processor EPROM memory
- 1006, al, configuration memory - uP1 setup

Driver parameters

The CtPmc4000 driver parameters are declared in the *Miscellaneous* module, on *Directly entered options* tab.



- Section name: CtPmc4000**
- Option name: LOG_FILE**
- Option value: log_filename**

Meaning - the diagnostic purposes are fulfilled by a text log file, where the driver operation status messages are entered.

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



- Section name: CtPmc4000**
- Option name: LOG_FILE_SIZE**
- Option value: number**

Meaning - option is used to determine the size of the log file defined with the use of LOG_FILE option. When the size exceeds the amount declared, the driver keeps the current log file contents in the file named *log_filename.old* (the name is created from *log_filename* given in the LOG_FILE declaration and extension *.old*). In this manner, the user is always able to view records of size equal to at least LOG_FILE_SIZE MB.

Parameter:

number - log file size in MB.
Default value - default log file size is 10 MB.



- Section name: CtPmc4000**
- Option name: LOG_OF_TELEGRAMS**
- Option value: YES / NO**

Meaning - option allows the contents of telegrams sent between the driver and controllers to the log file (declared with use of LOG_FILE option) to be saved. The subject option should be used only during the start-up of the **asix** system.

Default value - by default, the option value is set to NO.

EXAMPLE

Example of driver parameters:

Section name: CtPmc4000
Option name: LOG_FILE
Option value: d:\tmp\CtPmc4000\pmc.log

Section name: CtPmc4000
Option name: LOG_FILE_SIZE
Option value: 20

Section name: CtPmc4000
Option name: LOG OF TELEGRAMS
Option value: YES