

PowerCube parser software manual

1. In root directory of PwrCube.exe (usually C:\Program Files\PowerCube) create *parsers* directory
2. There is built-in Pwrcube.exe parser for PowerCube charger which has top priority in data processing
3. If coming RS232 data don't comply with built-in parser, parsers in *parser* directory are processed
4. Parsers are processed in alphabetic order
5. If condition HEAD match incoming data, data are processed by parser
6. If condition HEAD doesn't match incoming data, DATA condition is evaluated. If condition DATA match incoming data, data are processed by parser. If even DATA doesn't match, next parser (in alphabetic order) is tested
7. Some basic facts about Regular Expression you can find forexample here http://www.zvon.org/other/PerlTutorial/Output_cze/contents.html#26 or anywhere on internet

Basic set of commands working with regular expressions in version PwrCube.exe 1.1.21

TITLE is name of parser which is displayed in Protocol Editor and Block name

Example 1.
TITLE = Hello

HEAD is file header. If first row of incoming data meets HEAD condition, remaining data will be processed by this parser

Example 2.
HEAD = ^([XYZ])\s+(\S*)\s*\$

HEAD_IX_NAME is index of value in brackets which will be added to block name

Example 3.
HEAD_IX_NAME = 1
In example 2. is index1=XYZ and index2=\S* (both expressions are in brackets), Characters XYZ will be used, while they are in first set of brackets (index1)

HEAD_IX_BATT is index of value in brackets. This value is translated to string which is defined in BATT1_IF and BATT1_THEN. BATT1_THEN string is added to block name

Example 4.
HEAD_IX_BATT = 1
In example 2. is index1=XYZ

BATT1_IF=X
BATT1_THEN=NiCd

```
BATT2_IF=Y
BATT2_THEN=NiHM
BATT3_IF=Z
BATT3_THEN=LiXxx
```

If X character will come, NiCD is added to block name

special function has operand * (asterix). If you will use it in expression

```
BATT1_IF=. *
BATT1_THEN=*
```

whole string will be transfered to block name

DATA is format of row where all key values (time, voltage and current) are evaluated

Example 5.

```
DATA = ^\s*([0-9.]+)\s+([0-9.]+)\s+([0-9.]+)\s*$
```

DATA_IX_T,U, I key value index

Example 6.

```
DATA_IX_T = 1
DATA_IX_U = 2
DATA_IX_I = 3
```

DATA_EXPRESSION_T,U,I key values can be modified using regular math operands. If this parameter is not defined, direct RS232 data are used instead. T [sec], U [V] a I [A]

Example 7.

```
DATA_EXPRESSION_T = t / 10
DATA_EXPRESSION_U = u * 10
DATA_EXPRESSION_I = (i + 5) * 3
```

#Endblock and Endsession formats. If this match, block or session is closed

Example 8.

```
ENDBLOCK = endblock
ENDSESSION = endsession
```

#TIMEOUT=X. If there are no data for X sec, session is closed

Example 9.

```
TIMEOUT=5
```

IGNORE1, IGNORE2

Example 10.

```
IGNORE1=^\*\S
```

Rows matching this criteria are ignored. Anyhow NO(!) new parser is evaluated, i.e. data are still processed by current parser.

Version 1.0 from 21/11/2005

© Roman Vojtěch, www.Lomcovak.cz