

Version 1.0
10 / 2019

PMON User Manual

A PACTOR® Monitoring Utility for Linux

SCS Special Communications Systems



pmon(1) PMON User Manual

SCS GmbH & Co. KG info@scs-ptc.com

NAME

PMON - a PACTOR(R) monitoring utility

SYNOPSIS

```
pmon [-a | -audio <audio_device> | ?] [-o | -port <port_number>] [-p | -packets <0|1>] [-h | -hex <0|1>] [-v | -verbose <0|1>] [-t | -test <0|1>] [-n | -nostdout] [-V | -version] [-L | -license] [-? | -help | -usage] [<config_file>]
```

To quit PMON press CTRL-C.

DESCRIPTION

PMON allows the thorough observation and documentation of all presently available PACTOR-1/2/3 transmissions. PMON covers all PACTOR levels with the appropriate Speedlevels and packet variations. In the operating mode *PACTOR-3* PMON will read in parallel also PACTOR-2 and PACTOR-1. The very wide receiving range (frequency offset ± 200 Hz), as well as the automatic sideband recognition, ease routine operation of PMON with PACTOR-2 and PACTOR-3 considerably.

PMON also searches for *Connect Packets*; this search comprises the *Normal Connect*, the *Robust Connect* as well as the *Free Signal*.

PMON is strictly for private use only! This means non-commercial applications, by private persons, e.g. radio amateurs. The monitored data may be published without restriction by those users - if not prohibited by law!

The **audio center frequency** of the monitored signal should be tuned to **1500 Hz**. On the upper sideband (USB), the displayed dial frequency (VFO) on the receiver must be set 1500 Hz below the HF center frequency of the signal to be monitored.

OPTIONS

The following command line arguments are recognized:

- **-a, -audio:** Specify the audio device to work with. Or with ? pmon lists all available audio devices.
- **-o, -port:** Port number for the TCP/IP connection. Default: 3301.
- **-p, -packets:** Packet types to display (0 or 1). Default: 0.
- **-v, -verbose:** Verbose formatted output (0 or 1). Default: 0.
- **-h, -hex:** Hex conversion if **Verbose 0** (0 or 1). Default: 1.
- **-t, -test:** Enables or disables Audio Level Display (0 or 1). Default: 0.
- **-n, -nostdout:** Disables the output to console.
- **-?, -help, -usage:** Help message.
- **-V, -version:** Show program version.
- **-L, -license:** Show license information.
- **<config_file>:** Name of a config file if don't want to use the default name and search pattern.

The only parameter you really have to specify is the audio device pmon should work with.

Example:

```
pmon -a hw:0,0
```

To get a list of available audio devices enter

```
pmon -a ?
```

AUDIO CONNECTION

For proper operation you need of course an audio cable from your radio to your sound device. If the audio device has stereo input we only use the left channel which is normally the tip of the audio jack.

Nevertheless you may simply apply the audio signal to both channels (tip and ring) of the audio jack.

CONFIG FILE

PMON tries to read a config file on startup.

If not given on the command line PMON searches the config file in following locations in the given order:

1. Location pointed to by the environment variable PMON_CFG.
2. .pmon.cfg in the current directory.
3. .pmon.cfg in the users home directory.

Parameters given on the command line always take precedence over parameters read from the config file!

The format of the config file is very simple and can edited with any text editor e.g. nano.

Here is the general structure of the file:

```
[global]
port = <portnumber>      ; default: 3301
console = false           ; disable console output

[audio]
device = <devicename>     ; default: not defined

[pmon]
packets = <0/1>
verbose = <0/1>
hex = <0/1>
test = <0/1>
```

Comments start with a ; or #.

All key words are case insensitive!

A simple configuration contains only the audio device:

```
[audio]
device = hw:1,0
```

To get a list of available audio devices enter

```
pmon -a ?
```

A bit more sophisticated example:

```
[global]
port = 4000

[audio]
device = hw:0,0

[pmon]
verbose = 1
```

This sets the TCP/IP port to 4000, the audio device to hw:0,0 and sets *Verbose* to 1.

PARAMETERS

Audio

Value range: ALSA raw audio device

Default : none

Defines the audio device pmon should work with. This is the most important parameter. With

`pmon -a ?`

or

`pmon --audio ?`

PMON outputs a list of available audio devices.

An audio device is not only a typical (USB) *sound card* but can be for example a modern radio with build in USB sound capability like ICOM IC-7300.

Port

Value range: 0-65535

Default : 3301

PMON can output data to a raw TCP/IP connection. The Parameter **Port** specifies the TCP/IP port PMON listens for an incoming connection.

Packets

Value range: 0-1

Default : 0

Defines which packet variation should be given out.

0: Prints TRAFFIC packets only.

1: Prints TRAFFIC packets and REQUEST packets (repetitions).

Verbose

Value range: 0-1

Default : 0

The Verbose command defines if additional status information and framing is used in the output.

0: Prints the actual Payload only. No additional status information or framing is added.

1: Formatted output, beginning with status information followed by the actual Payload. See below for a detailed description.

Hex

Value range: 0-1

Default : 1

This controls the automatic hex translation of unprintable characters if **Verbose 0**.

Hex 0 disables the automatic hex translation. This gives a pristine copy of a PACTOR transmission, even if characters in the range of 128-255 (dec.) were used.

With **Verbose 1** this parameter has no effect!

Test

Value range: 0-1

Default : 0

Measures the input amplitude and outputs the result every second if set to 1.

FUNCTION

Note: in the following text, <...> (statement in pointed brackets) represents a *Byte*, e.g. <CR> means *Carriage return* (decimal value 13), <0xFE> shows a Byte with the decimal value 254.

If PMON is ready to decode PACTOR traffic it will output the message:

```
PACTOR-1/2/3 Monitor started:
=====
```

Output format

In **Verbose 1** mode various additional status information is displayed along with the payload:

```

<0xFA>
###PLISTEN: Level: 3:
###STATUS: SL: 2, CYC: 1, RQ: 0, REV: 1, LSB: 0, dF: -2.5, FRNR: 30
###PAYLOAD1: LEN: 77, TYPE: 1
###PAYLOAD2:
This is a test. The actual useable information is displayed here... ###PAYLOAD_END
<0xFE>

```

Every output starts with the byte <0xFA> as the unique output start character (*Start of Frame*). This character cannot appear in the following data set, also not in the payload as unprintable characters are automatically printed in hexadecimal form.

The first line

```
<CR><LF>###PLISTEN: Level: 3:<CR><LF>"
```

shows the current PACTOR-Level (1-3).

The second line shows the status:

```
###STATUS: SL: 2, CYC: 1, RQ: 0, REV: 1, LSB: 0, dF: -2.5, FRNR: 30
```

The fields are:

SL: 1-10 Speedlevel of the packet.

CYC: 0 this is a **short** packet.
1 this is a **long** packet.

RQ: 0 the packet is not a REQUEST packet, but a normal TRAFFIC packet
(with a new 2-bit packet counter or new CRC in comparison with the previous packet).
1 the packet is a REQUEST packet. This is detected by comparison of
the actual CRC with the CRC of the previously received packet.

REV: 0 Paket with the original tone-adjustment.
1 Paket with reversed tone-adjustment.

LSB: 0 the packet is received in the sideband mode that the radio is set to.
1 the packet is received in the opposite sideband to that which the radio is set to.
E.g. in LSB if the receiver is set to USB.

dF: -250...+250.0 Measured frequency error in Hertz. One significant decimal digit.

FRNR: 1...X frame number counting from 1, to ease documentation.

After the status line the actual payload follows:

```

###PAYLOAD1: LEN: 77, TYPE: 1
###PAYLOAD2:
This is a test. The actual useable information is displayed here... ###PAYLOAD_END

```

The line

```
###PAYLOAD1: LEN: 77, TYPE: 1<CR><LF>
```

precedes the output and contains the length of the following output as well as the type. The length always refers to the actual payload bytes and so does not change for hexadecimal output.

For the data type indicator the following values are possible:

- 0: ASCII (8 Bit, transparent)
- 1: Huffman, text data
- 2: Huffman, swapped (reversed capitals and small letters), text data
- 3: Reserved
- 4: PMC, German, text data
- 5: PMC, German, swapped, text data
- 6: PMC, English, text data
- 7: PMC, English, swapped, text data
- 8: ASCII (8 Bit, transparent), automatic hexadecimal output as the data contains unprintable characters.

These are characters between decimal values <32 and >127 except for <CR> and <LF>.
9: LZHUF, data from the LZHUF decoder

The last character of a packet data set is always the unique end character <0xFE> (*Frame End*) except with **Verbose 0**.

Output of received connect-packets

The output of detected connect-packets always starts with the sequence

```
<0xFA><CR><LF>###CONNECT:
```

Followed by the packet type and the destination callsign (max. 8 characters – alphanumeric), both enclosed in right angled brackets. The output is always finished by <CR><LF><0xFE>

Example:

```
<0xFA>
###CONNECT: [Robust Call: DL6MAA]
<0xFE>
```

The following packet types are possible:

```
[Normal Call:    ]
[Longpath Call:  ]
[Robust Call:    ]
[Free Signal Normal:  ]
[Free Signal Encrypted:  ]
```

LZHUF decoder

PMON automatically decompresses LZHUF compressed messages *on the fly*. This is very useful for monitoring Winlink email transfers. On mode **Verbose 1** or higher, the status line is headed by *###LZHUF* instead of *###PLISTEN*, the LZHUF decompressed data is displayed as data type 9.

Note: LZHUF compression is not inherent to PACTOR, it is not the internal PACTOR Huffman/PMC compression but an external compression variant utilized by some application software, e.g. Winlink. Nevertheless, as LZHUF is widely used, PMON supports decompression even of that third party compression type.

Allowed frequency offset

PACTOR-1

Generally: ca. ± 100 Hz

PACTOR-2

Generally: ca. ± 240 Hz

PACTOR-3

Generally: ca. ± 200 Hz

Delay of the data output

PMON uses sometimes very difficult iterative searching algorithms in order to detect packets under the worst possible SNR. This means that very many *False Synchronizations* are produced and must be evaluated. The resulting computational effort can give a maximum of 6 seconds delay between the physical end of a packet at the AF input and the display of the packet on the user interface. The average value is generally only 1 to 2 seconds however.

Main Limitations

CS3-Packets (Changeover) are currently not detected. These packets however contain only idle bytes in virtually all practical cases.

NETWORKING

PMON offers a simple networking interface. PMON listen for an incoming TCP/IP connection on the given port (default: 3301). If the connection is successfully established, all output is also passed to the network.

You can switch off the output to the local console with the command line parameter `-n` or `-nostdout` or with the *console* keyword in the *Global* section of a config file.

PMON recognizes a few very simple command on the network connection:

- P0<CR> - Packet 0
- P1<CR> - Packet 1
- V0<CR> - Verbose 0
- V1<CR> - Verbose 1
- H0<CR> - Hex 0
- H1<CR> - Hex 1

All this can easily be tested with any Telnet client.

```
telnet <ip_of_pmon> <port>
```

If your computer which runs PMON has the IP address 192.168.0.10 and PMON uses the default port:

```
telnet 192.168.0.10 3301
```

ENVIRONMENT VARIABLE

PMON_CFG holds the file name and full path of a PMON config file.

Set an environment variable with:

```
export PMON_CFG="/path/to/file/my_pmon.cfg"
```

Remove the environment variable with:

```
unset PMON_CFG
```

COMPATIBILITY

PMON was throughly test with the following hardware:

- Raspberry Pi 4 Model B
- Raspberry Pi 3 Model B+
- HiFiBerry DAC+ ADC
- microHAM USB Interface III
- various cheap USB sound cards

AUTHOR

SCS GmbH & Co. KG
Roentgenstrasse 36
63454 Hanau
GERMANY

LICENSE

PMON - a PACTOR(R) monitoring utility
(C) Copyright 2019 SCS GmbH & Co. KG, Hanau, GERMANY

Permission is hereby granted to use this software free of charge to monitor PACTOR(R) transmissions for **PRIVATE USE** only, subject to the following conditions:

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



Spezielle Communications Systeme GmbH & Co. KG
Roentgenstrasse 36
63454 Hanau
GERMANY

Internet: www.p4dragon.com
Email: info@p4dragon.com

FON: +49 (0)6181 85 00 00
FAX: +49 (0)6181 99 02 38