PXdragon DR-9400

Firmware Update 1.01.02



Description of modifications and improvements

Update from Version 1.00.00

Firmware Update Information for PXdragon DR-9400

Firmware 1.01.02 replaces firmware 1.00.00. This document describes the new features and improvements available when updating directly from 1.00.00 to 1.01.02. The update introduces Robust Unproto (RUP), extends Hostmode, and improves reception, monitoring, maintenance and operating controls.

The update keeps the established PACTOR operation and hostmode workflows. Existing applications can continue to use the normal PACTOR functions; the new commands extend the available operating modes and diagnostic options.

Several features introduced in this release were developed in close cooperation with SafecomLink[1], the first application to put them into day-to-day use.

1 New Features

1.1 Robust Unproto (RUP)

RUP is a new robust unconnected short-message mode for the PXdragon DR-9400. It is intended for short text or data messages that should be transmitted without setting up a connected PACTOR link first.

RUP was designed for difficult real radio channels. Its main channel properties are:

Property	RUP behavior
Occupied bandwidth	About 2.4 kHz, like PACTOR-4.
Multipath propagation	Highly robust against delayed signal components and delay spread.
Narrowband interference	Very robust against narrowband interferers, because localized interference usually affects only a limited part of the RUP signal bandwidth.
Non-flat transceiver paths	Tolerant of audio paths that are too narrow or not flat.
Frequency offset	Can handle large offsets of about ±1 kHz.
Very weak signals	Demonstrated SL1 tests reached about 90% packet decode rate at approximately −20 dB SNR @ 3 kHz under suitable channel conditions.
Partial-spectrum reception	At good SNR, successful decoding can still be possible when only part of the transmitted spectrum is received cleanly.
Low crest factor	The transmitted signal has a crest factor of only about 3 dB.

RUP provides four selectable speedlevels:

- **Super Robust** for difficult paths.
- **Robust** as the normal robust operating level.
- **Fast** for improved throughput on usable paths.
- **Very Fast** for good paths and higher payload efficiency.

When RUP is enabled, the receiver searches all regular RUP speedlevels in parallel, so the receiving station does not need to know the speedlevel selected by the sender. Received RUP messages are delivered as complete messages in terminal mode and in hostmode.

[1] SafecomLink is a trademark of its publisher. See SafecomLink for Pactor for product information.

The default setting is `RUP 0`. In this setting, RUP receive output is off and the RUP receiver is kept out of the active receive path.

One possible application of RUP is compact network management for SafecomLink systems. SafecomLink already uses RUP for one-to-many broadcasts, group chats, and other coordination functions. The unconnected short-message transport is well suited for small management messages, status exchange, station announcements, and other lightweight network coordination tasks that should not require a full PACTOR connection.

1.2 RUP Beacons

RUP beacons provide compact station identification. Depending on the selected mode, the beacon can contain MYCALL only, MYCALL with GPS position, or MYCALL with the configured fixed position. Received beacons are displayed in a readable format and include the local receive time and estimated RUP SNR. Beacon transmission observes the channel busy indication, including the RUP busy state.

Beacons carry an explicit type identifier, allowing the receiver to distinguish them from ordinary RUP messages. The compressed callsign is also checked for correct encoding.

Interoperability: For mutual beacon recognition, use firmware supporting the explicit beacon identifier on both stations. RUP-capable firmware predating this identifier sends beacons that are not recognized as beacons by this version.

SafecomLink uses a separate beacon format that complements the modem's standard beacon with additional information, such as the station locator and available services, including email and AI services.

1.3 Extended Status and Logging

The update adds extended status polling and a configurable system log level. This makes it easier to obtain useful diagnostic information when needed, while keeping normal operation quiet.

1.4 RUP Data Identification and Waterfall Display Support

Host applications can distinguish received RUP data from PACTOR connection data using the new Hostmode extension setting `%M 2`. The new Hostmode command `@W` supplies compact spectrum data for waterfall displays in supporting applications. Command usage and data formats are described below.

2 New and Extended Commands

cmd: RUP

Syntax:

```
RUP <0..4>
```

Selects the Robust Unproto speedlevel.

`0` RUP off. RUP receive output is disabled and the RUP receiver is kept out of the active receive path.

`1` Super Robust.
`2` Robust.
`3` Fast.
`4` Very Fast.

If the command is entered without parameter, the current RUP setting is displayed.

Default: `0` (RUP off).

Examples:

```
RUP
RUP 1
RUP 2
```

cmd: RP

Syntax:

```
RP <text>
```

Transmits a Robust Unproto text message from terminal mode. A `#` character in the text is transmitted as a line break.

Example:

```
RP CQ from PXdragon#Second line
```

There is no stored default value for this command; it transmits the supplied text immediately.

cmd: RUP Beacon

Syntax:

```
RUP Beacon <0..3> [seconds]
```

Configures compact RUP beacon transmission.

`0` Beacon off.
`1` Beacon with current GPS position.
`2` Beacon with configured fixed position.
`3` Callsign-only beacon.

The optional second parameter sets the beacon interval in seconds.

Defaults: Beacon mode `0` (off), interval `900` seconds.

Mode `2` reuses the APRS fixed position. This avoids entering the same station position twice. The fixed position is configured with the APRS `POsition` subcommand in APRS position format:

```
APRS POsition <latitude> <longitude>
```

Example:

```
APRS POsition 4810.30N 01030.25E
```

To clear the stored fixed position:

```
APRS POsition NONE
```

Examples:

```
RUP Beacon
RUP Beacon 2 300
RUP Beacon 1 300
RUP Beacon 3 900
```

Received RUP beacons are shown as formatted text. The receive time is generated locally by the receiving modem and is not transmitted over the air. The `SL` field shows the speedlevel of the received RUP packet. The actual beacon information is framed by `###` to make it stand out.

Receive examples:

```
RUP-Beacon 14:23:07 SL1 ### DL6MAA ###
RUP-Beacon 14:23:07 SL1 ### DL6MAA 52.12345N 009.87654E ###
```

The same formatted beacon line is also delivered in hostmode on the configured PACTOR hostmode channel.

cmd: RUP Filter

Syntax:

```
RUP Filter <0..100>
```

Controls the terminal display filter for RUP text output. The filter helps keep unlikely binary-looking output away from the terminal screen. A value of `0` disables the filter.

Default: `30` percent.

Examples:

```
RUP Filter
RUP Filter 30
RUP Filter 0
```

cmd: RUP MH

Syntax:

```
RUP MH
```

Shows the local RUP heard list. The list is built from received RUP beacons and contains the receive date, receive time, callsign, and estimated RUP SNR for the most recent beacon heard from each callsign.

cmd: RUP Remote

Syntax:

```
RUP Remote <0..1>
```

Controls replies to simple RUP remote query messages.

`0`	Remote query replies off.
`1`	Remote query replies on.

Default: `0` (off).

When enabled, a received RUP message containing `+?snr` requests a short SNR reply. A received RUP message containing `+?mh` requests the RUP heard list. Replies are sent after a short randomized delay and wait until the channel is no longer busy.

%RP

Hostmode command for binary-transparent RUP transmission. All bytes following the `%RP` command prefix are transmitted as one RUP message.

There is no stored default value for this command.

G4 / G5

Extended hostmode status polling commands for applications that need current modem status and receive-quality information.

`G4` returns the extended status information including the normal status bytes plus two additional bytes: the last measured SNR value and the age of that SNR value in seconds.

`G5` returns only the compact two-byte SNR status: SNR and age in seconds. This is intended as a fast polling alternative when the application only needs receive-quality information.

These are polling commands and have no stored default value.

Examples:

```
G4
G5
```

cmd: SNR

Terminal-mode command that displays the last measured SNR value and its age. This provides a convenient alternative to hostmode polling during tests and manual operation.

This is a display command and has no stored default value.

Example:

```
SNR
```

%U

Hostmode command that sets the unproto timeout in seconds.

Default: `240` seconds.

Examples:

```
%U
%U 240
```

%X

Syntax:

```
%X <0..2>
```

Hostmode command that controls PTT inhibit operation.

`0`	Normal operation.
`1`	Inhibit the next PTT activation.
`2`	Inhibit PTT immediately.

Default: `0` (normal operation).

Examples:

```
%X
%X 1
```

This gives host applications direct control over transmitter inhibit, enabling clean operator aborts and deliberate receive-only intervals. SafecomLink uses this mechanism, for example, to listen briefly after a frequency change before transmitting ("QSY sniffing").

cmd: TZone

Syntax:

```
TZone
TZone <number>
```

Without parameter, the command lists the available time zones with their selection numbers. With a numeric parameter, it stores the selected time zone in non-volatile memory. Example: `TZone 1` selects `Europe/Berlin`. A modem reboot is requested after changing the time zone.

The default setting is `Europe/Berlin`.

Available time zones:

`1`	Europe/Berlin
`2`	Europe/London
`3`	Europe/Dublin
`4`	US/Eastern

`5` US/Central
`6` US/Mountain
`7` US/Pacific
`8` US/Alaska
`9` Canada/Atlantic
`10` Canada/Newfoundland
`11` Australia/North
`12` Australia/South
`13` Australia/Queensland
`14` Australia/NSW
`15` Australia/West
`16` Brazil/East
`17` Asia/Hong_Kong
`18` Asia/Singapore
`19` Asia/Jakarta
`20` Asia/Ulan_Bator
`21` Pacific/Auckland
`22` Pacific/Honolulu
`23` Europe/Moscow

Examples:

```
TZone
TZone 1
```

cmd: SYS LOG LEVEL

Syntax:

```
SYS LOG LEVEL <0..255>
```

Sets the system log verbosity. Detailed TRX data logging is now tied to this level, reducing normal log traffic while still allowing detailed diagnostics when required.

Default: `0`.

Examples:

```
SYS LOG LEVEL
SYS LOG LEVEL 1
SYS LOG LEVEL 5
```

%M – Separate RUP Reception in Hostmode

Syntax (Hostmode command):

```
%M <0..2>
```

- `0`: Existing standard receive format, without delayed transmit echo.
- `1`: Existing receive format with delayed transmit echo.
- `2`: Separate RUP receive frames (code 9), with delayed transmit echo (code 8).

To enable: A supporting host application sends `%M 2` after starting Hostmode. The setting is reset to 0 whenever Hostmode is initialized, so the application must set it again at each Hostmode start. RUP reception must also be enabled; for example, enter `RUP 2` in terminal mode before starting Hostmode.

Use setting 2 only with applications supporting frame type 9. Existing applications can retain setting 0 or 1. With setting 2, RUP receive output, including beacon output, uses type 9 instead of ordinary connection-data type 7; PACTOR connection data continues to use code 7. Delayed transmit echo remains enabled and continues to use code 8.

@W – Waterfall Spectrum Data

Syntax (Hostmode command):

```
@W
```

Returns the latest spectrum line for a waterfall display. A supporting application polls this command and renders successive lines. The reply contains binary data; the command does not draw a waterfall in a text terminal. No separate modem-side waterfall enable command is required.

To use: Enable the waterfall feature in a host application supporting `@W`. This feature requires application support for polling and displaying the returned data.

Reply format:

- Hostmode connection-data reply (type 7), with 128 binary payload bytes.
- 256 frequency bins in ascending frequency order, starting at 300 Hz. The spacing is 9.375 Hz, covering a 2400 Hz band up to approximately 2700 Hz (last bin centre: 2690.625 Hz).
- Two four-bit amplitude values per byte. The lower-frequency bin is in **the least significant nibble (bits 0–3)**; the next, higher-frequency bin is in the most significant nibble (bits 4–7).
- Values range from 0 to 15 on a base-2 logarithmic amplitude scale, with approximately 6 dB per step. Values below amplitude 1 map to 0; otherwise the level is $\min(15, \lfloor \log_2 A \rfloor + 1)$, where A is the normalized FFT amplitude in sample units. Values are not calibrated dBm readings.

The command returns a snapshot of the latest available line, rather than a queue of earlier lines; repeated polls may return the same line until it is refreshed.

3 Further Improvements

3.1 PACTOR Reception and Synchronization

- PACTOR-I call synchronization was hardened to attribute a detection to the actual winning candidate. Improved callsign discrimination reduces false responses to very similar callsigns while retaining Memory-ARQ weak-signal gain.
- PACTOR-4 control-signal processing now applies the correct frequency-offset compensation in the real-part detector, improving control-signal reception in the presence of a carrier offset.
- The transmit level of the PACTOR-4 Speed Level 1 chirp waveform was corrected to match the intended modem output level.
- PACTOR-4 Short Robust operation and changeover behavior were refined, and Short Robust Memory-ARQ support was restored.
- Timing tracking for long-running PACTOR operation was improved.
- The PACTOR-4 Turbo decoder was optimized for ARM processors with NEON support, reducing decoding CPU load without changing the on-air protocol.

3.2 Hostmode and Monitoring

- PACTOR call frames detected by ADEC are now delivered as ordinary hostmode listen text. They can therefore be displayed without enabling PMON.
- PMON processing no longer delays normal PACTOR synchronization and remains available across call detection.
- Hostmode `%W` channel-busy polling now also accounts for active RUP transmission and RUP reception.
- Main command help was updated and sorted for easier command discovery.

3.3 Network and Maintenance

- Network modem discovery and startup handling were made more reliable, including restored NetDetect UDP processing and improved embedded-system startup ordering.
- Fast block-based modem log download was added while retaining automatic compatibility with the existing legacy log-transfer protocol.
- Firmware update validation, diagnostics, and update-task preparation were improved.

4 Companion Windows Tools

4.1 PX Fast Updater

The new portable PX Fast Updater supports firmware updates through a Windows COM port or the modem's Telnet interface. It discovers network modems automatically, verifies the modem before starting an update, uses the fast update protocol when available, and falls back to the legacy protocol for older firmware. Successful fast updates are reported only after the modem has validated and stored the firmware file.

The updater can also download the modem system log. Current firmware transfers the log in blocks of up to 65,536 bytes for substantially shorter download times, while older firmware remains supported through automatic fallback to the legacy 256-byte transfer.

4.2 PX HostTerminal

PX HostTerminal provides a simple entry point for setup, testing, and chat using extended CRC Hostmode (`JHOST5`) over Windows COM or Ethernet/Telnet. Separate panes show received text, confirmed transmitted text (delayed echo), and transmit input, with link status alongside.

It supports PACTOR ARQ calls, changeover/break-in, disconnect, text files, and P1/P2 Unproto broadcasts. Two compatible instances can exchange binary files using optional whole-file DEFLATE compression and CRC-32 verification; a transfer can be aborted. The optional `startup.txt` can set the interface and start Hostmode automatically.

5 Application Support

The Hostmode interfaces described here are available to host application developers. The features shown to an operator depend on support in the application being used. SafecomLink helped introduce the RUP and operating extensions in version 1.01.00; support for the newer RUP frame type and waterfall command must be checked in the chosen application. Companion Windows tools are separate downloads with their own version numbers.