Document Status

Published status	Confidentiality status
☐ Draft	☐ Internal
☒ Released	☐ NDA
☐ Canceled	☒ Public

Version Control

Ver.	Date	Summary of changes
0.9	27.09.2023	Final Draft
0.91	27.10.2023	Define forbidden string characters, unify representation of decimal elements in `VHF`, `SENS` and `AIRPT` messages, add airport status to `AIRPT` message
0.92	06.11.2023	Redefine text fields in periodic messages
0.93	03.07.2024	Add information about FLARM Classic devices
1.00	16.12.2024	Changed license and minimal FW version requirements

Scope and Summary

This document describes the FLARM Messaging functionality introduced with PowerFLARM firmware release 7.24. This functionality entails a dataport interface, and a set of RF messages to exchange information between FLARM devices.

Table of Contents

1 Introduction 4
2 Overview 5
2.1 System 6
2.2 Requirements 6
2.3 Document Conventions 7
3 Dataport Interface 8
3.1 Receiving Messages 9
3.2 Sending Messages 9
3.3 FLARM Messaging Status 9
3.4 String Encoding 10
3.4.1 Simple String 10
3.4.2 Hex-encoded UTF-8 String 10
4 Message Types 12
4.1 Aircraft Registration 12
4.2 Pilot Name 12
4.3 Aircraft Type 13
4.4 Aircraft Callsign 13
4.5 Version message 14
4.6 VHF Radio Frequency 14
4.7 Team Name 15
4.8 Sensor Measurements 16
4.9 Airport Information 17
4.10 Airport Weather 18
4.11 Open Broadcast 19
4.12 Open Unicast 20
4.13 Message Overview 21
4.14 Error Codes 21
4.15 Configuration Interface 22
4.15.1 Transmission of Periodic Messages 22
4.15.2 Enabling of Message Functionality 22
4.15.3 Licensing 23

4.15.4 FLARM Classic 23

5 Implementation Considerations 24

5.1 Data Acquisition 24

5.2 Message Scheduling 24

5.3 Messages Frequency 25

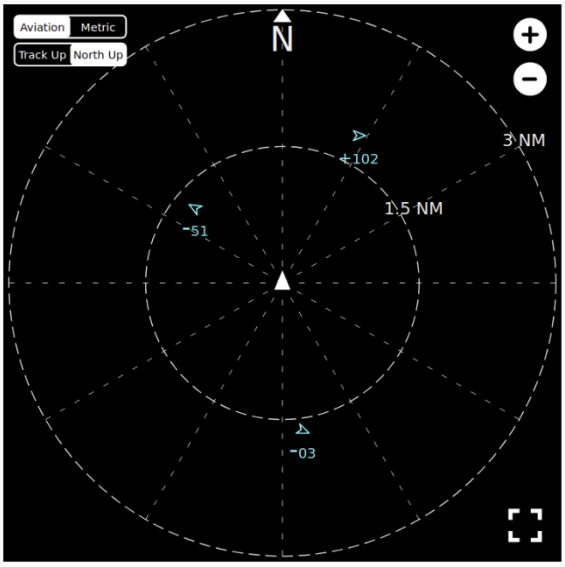
5.4 Checksum 25

1 Introduction

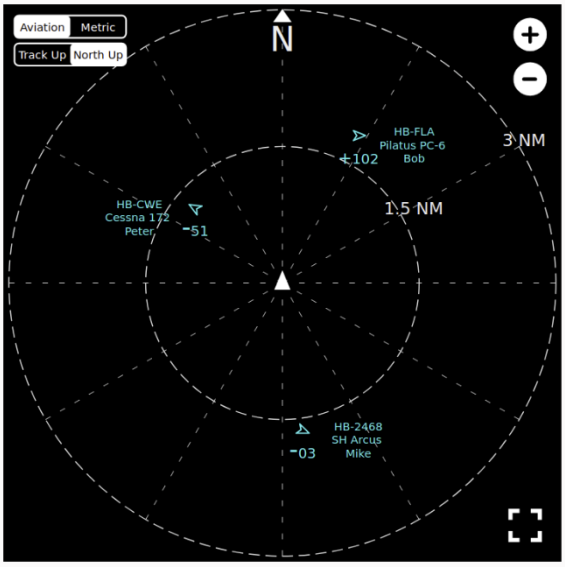
In this document, the FLARM Messaging functionality and its interface are specified. The new functionality allows for the exchange of various types of data between aircraft and/or ground stations. This information can be received by other units and then e.g., be used by displays or other accessories to show additional information about an aircraft or its pilot.

An example use case is augmenting situational awareness display with details of each received target:

Without the Messaging functionality:



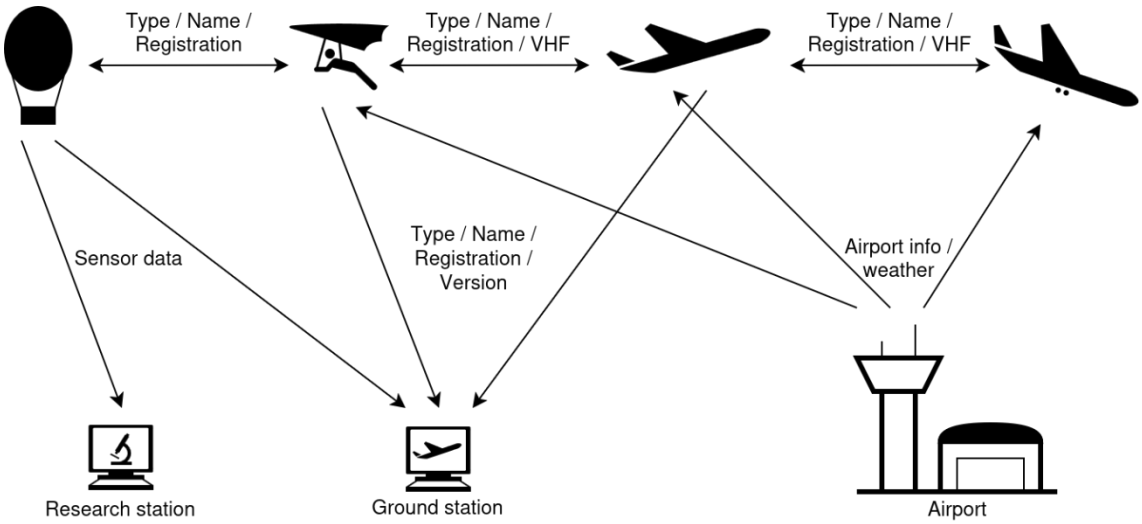
With the Messaging functionality:



This removes the need for a dedicated database (e.g., FLARMnet) that associates FLARM IDs with aircraft type, registration ID etc. and which is hard to keep up-to-date and accurate.

In this document, the Messaging functionality refers to the overall system – including transmission and reception of RF messages, scheduling and generating periodic messages etc. The Messaging interface refers to the set of dataport commands used to interact with the functionality.

2 Overview



FLARM Messaging adds communication capabilities between FLARM devices, as well as compatible ground infrastructure. It extends the FLARM communication by information that is not strictly relevant for safety or situational awareness purposes.

Users can select which information is being sent out and how the reception is handled. FLARM devices, however, do not store any information received through the Messaging functionality. E.g., it is up to a display manufacturer to implement a data store that associates received data from a specific FLARM target through the unique radio ID and display it accordingly.

Three classes of functionality are offered:

- Periodical messages are transmitted automatically. They contain information derived from the static state of the sending unit, in particular, pilot name, aircraft registration, aircraft callsign and aircraft type.
- Standard messages that can be invoked via the dataport interface by connected equipment. For example, with an interface to a VHF radio, the currently used frequency can be transmitted, which can then again be shown in a display.
- Open messages provide a data-exchange functionality without a predefined structure or semantics, also to be used by connected equipment via the dataport interface. Data can either be broadcast or transmitted to a specific aircraft (unicast). This can be used, e.g., for instrument manufacturers to provide additional functionality for data exchange.

In this document an extension of the dataport interface for sending and receiving Messaging data is described, along with configuration details.

2.1 System

The Messaging functionality consists of FLARM devices in the air and/or on the ground communicating via RF, and access to the data over the dataport interface.

For full use of the system, the interface must be implemented on a host system. For example, the periodic messages are intended to be shown on a traffic display, replacing the need for a dedicated database such as FLARMnet on the display that would need to be updated periodically. In that case, the individual messages received through the dataport interface need to be consolidated for each aircraft. In another example, the airport and weather information could be shown on a moving map display. Furthermore, the currently used frequency by a VHF radio connected to a host system can be received by a similar system on another aircraft. That information can be shown, and the user can be aided in the setup of their own VHF radio.

Finally, user-defined messages can be used for communication with more complex user interfaces.

2.2 Requirements

Hardware/software requirements:

- PowerFLARM[1] (Portable, Core, Fusion, Atom) and newer.
- FW 7.40 or later.
- License installed[2]

Necessary configuration:

- Messaging functionality enabled.
- Stealth mode disabled.
- No-track mode disabled.
- ICAO or FLARM address set (no Random ID).
- Baud rate of at least 57.6 kBaud.

[1] FLARM Classic provides limited Messaging functionality since FW version 7.25. See chapter 4.15.4 for more details.
[2] License is not required for transmission or reception of periodic messages

- Dataport configured with FLARM protocol of at least version 6 with FLARM messages enabled e.g., `$PFLAC,S,NMEAOUT1,91`.
- GPS fix acquired.

2.3 Document Conventions

To differentiate between user input and FLARM device output, all commands sent to the device are preceded by the prompt character: `>`

Example

- User → FLARM

  `> $PFLAM,S,VHF,118.455,,,*CS`

- FLARM → User

  `$PFLAM,A,OK,VHF,118.455,,,*CS`

Most of the command examples are trailed with a dummy checksum denoted as `*CS`. In a real implementation, these need to be replaced by correct checksums.

3 Dataport Interface

The Messaging functionality is accessed via a dataport interface. Whenever a message is received over the RF channel, an unsolicited message is emitted. Furthermore, to invoke sending out a message, the dataport interface with send, acknowledge, or error feedback is used.

Syntax:

NMEA Sentence	Description
`$PFLAM,U,<IDType>,<ID>,<MsgType>,<Payload>*CS`	Incoming message – sentence generated by FLARM device
`> $PFLAM,S,<MsgType>,<Payload>*CS`	Request to send a message
`$PFLAM,A,<Response>,<OptionalDetails>*CS`	FLARM answer to request
`> $PFLAM,R*CS`	Querry the status of Messaging functionality
`$PFLAM,R,<Queued>,<Sent>,<QueueCapacity>*CS`	Response for the status query

Description:

The `PFLAM` sentences are the interface to messaging RF messages transmitted or received to/from other FLARM devices. The FLARM device emits a `PFLAM` sentence when an RF message is received from another FLARM device and sending a `PFLAM` message to a FLRAM device triggers transmission of an RF message. The ID of an RF message received from another device is uniquely defined through both ID and IDType.

Parameters:

<ID>	6-digit hexadecimal value (e.g., "5A77B1") of sender device
<IDType>	Refer to FTD-012 for more details
<MsgType>	Type of message, as defined in the following sections
<Payload>	Type-specific payload
<Response>	Response from a FLARM device (`OK` or `ERROR`)
<OptionalDetails>	Parsed arguments or error details
<Queued>	Number of messages requested by user with `$PFLAM,S`
<Sent>	Number of messages transmitted
<QueueCapacity>	Remaining number of free slots in message transmission queue
***CS**	NMEA 0183 checksum

3.1 Receiving Messages

When an RF message has been received, a message of the following form will be emitted on the dataport interface:

```
$PFLAM,U,<IDType>,<ID>,<MsgType>,<Payload>*CS
```

The payload field is specific for the different message types described in the following sections.

3.2 Sending Messages

To send messages, the syntax known from `$PFLAC` command is used. In general, the message sending command syntax is as follows:

```
> $PFLAM,S,<MsgType>,<Payload>*CS
```

The `S` query is used to indicate transmission of an RF message. The `<MsgType>` and `<Payload>` are defined in the following sections.

If the message is successfully parsed and stored in the transmission queue the device answers with:

```
$PFLAM,A,OK,<MsgType>,<ParsedValues>*CS
```

Otherwise, an error is generated:

```
$PFLAM,A,ERROR,<OptionalErrorDetails>*CS
```

If the command was executed successfully, the response contains the same arguments, as parsed by the device.

3.3 FLARM Messaging Status

Status of Messaging can be checked with `$PFLAM,R` command.

As a response the FLARM device will return the number of packets scheduled for transmission (since the last system startup), the number of messages transmitted and the number of free slots in the transmission queue.

Example

Since the system startup transmission of 349 messages were requested, 390 were sent already and there are 6 more slots in the transmission queue.

```
> $PFLAM,R*CS
```

`$PFLAM,R,394,390,6*CS`

3.4 String Encoding

There are two types of string fields used in the message payload:

3.4.1 Simple String

Used as an ICAO identifier in the `AIRPT` message and to transmit extra weather information in `METAR`.

Contains only alphanumeric characters ([A-Z][a-z][0-9) and a limited set of special characters:

Character	Hex	Dec
Space	0x20	32
(	0x28	40
)	0x29	41
+	0x2B	43
-	0x2D	45
.	0x2E	46
_	0x5F	95

Basic strings are passed directly using the dataport interface.

Examples

`$PFLAM,U,2,DG0000,1,AIRPT,`**`LSZF`**`,47.443333,8.233888,1300,26,121.555,1013*CS`

`$PFLAM,U,2,DG0000,METAR,260,7,,190,280,9999,SCT,1200,21,18,`**`-TSRA`**`*CS`

3.4.2 Hex-encoded UTF-8 String

String payload in messages `AREG`, `PNAME`, `ATYPE`, `ACALL` and `ATEAM` are UTF-8 encoded and passed over the dataport interface as hex values (each byte is represented by two hexadecimal characters ([0-9][A-F]). NUL-termination is implicit.

In periodically generated messages: `AREG`, `PNAME`, `ATYPE`, `ACALL`, if the configured data source field is longer than 17 characters, the string is cropped to the last complete UTF-8 character within that limit.

Examples

Data cropping:

Configured string	Cropped string
Leon Riemenschneider	Leon Riemenschnei
Paul Christoph Wöhler	Paul Christoph W

Pilot name *Andrée Müller* as seen in the `PNAME` message:

```
$PFLAM,U,2,DF2000,PNAME,416E6472C3A965204DC3BC6C6C6572*35
```

UTF-8	A	n	d	r	é	e		M	ü	l	l	e	r
Hex	41	6E	64	72	C3A9	65	20	4D	C3BC	6C	6C	65	72

4 Message Types

All message types supported by the Messaging functionality are defined below.

Some arguments in messages generated by `$PFLAM,S` command are marked as optional and can be left empty (they still need to be delimited by commas (,)).

4.1 Aircraft Registration

Aircraft registration is taken directly from the FLARM device configuration (`GLIDERID` configuration item). If the configured field is longer than 17 characters, all additional characters are discarded. If the corresponding configuration item is not set, then the message is not sent.

Type Specifier	Data Source	Periodicity
`AREG`	Device configuration (`GLIDERID`). Not possible to invoke via `$PFLAM`.	More than once per minute if bandwidth allows

Arguments

Payload	Mandatory	Details
`<name>`	yes	Hex string (max 17 bytes)

Example

Message received from a FLARM target `DF2000` containing aircraft registration `HB-SIA`:

```
$PFLAM,U,2,DF2000,AREG,48422D534941*CS
```

4.2 Pilot Name

Pilot name is taken directly from the FLARM device configuration (`PILOT` configuration item). If the configured field is longer than 17 characters, all additional characters are discarded. If the corresponding configuration item is not set, then the message is not sent.

Type Specifier	Data Source	Periodicity
`PNAME`	Device configuration (`PILOT`). Not possible to invoke via `$PFLAM`.	More than once per minute if bandwidth allows

Arguments

Payload	Mandatory	Details
<name>	yes	Hex string (max 17 bytes)

Example

Message received from a FLARM target DF2000 containing pilot name Orville Wright:

```
$PFLAM,U,2,DF2000,PNAME,4F7276696C6C6520577269676874*CS
```

4.3 Aircraft Type

Aircraft type is taken directly from the FLARM device configuration (ATYPE configuration item). If the configured field is longer than 17 characters, all additional characters are discarded. If the corresponding configuration item is not set, then the message is not sent.

Type Specifier	Data Source	Periodicity
ATYPE	Device configuration (GLIDERTYPE). Not possible to invoke via $PFLAM.	More than once per minute if bandwidth allows

Arguments

Payload	Mandatory	Details
<name>	yes	Hex string (max 17 bytes)

Example

Message received from a FLARM target DF2000 containing aircraft type Cessna 172:

```
$PFLAM,U,2,DF2000,ATYPE,436573736E6120313732*CS
```

4.4 Aircraft Callsign

Aircraft callsign is taken directly from the FLARM device configuration (COMPID configuration item). If the configured field is longer than 17 characters, all

additional characters are discarded. If the corresponding configuration item is not set, then the message is not sent.

Type Specifier	Data Source	Periodicity
ACALL	Device configuration (COMPID). Not possible to invoke via $PFLAM.	More than once per minute if bandwidth allows

Arguments

Payload	Mandatory	Details
<name>	yes	Hex string (max 17 bytes)

Example

Message received from a FLARM target DF2000 containing e.g., the competition ID "ZM":

```
$PFLAM,U,2,DF2000,ACALL,5A4D*CS
```

4.5 Version message

Internal FLARM message that contains FW/HW version information. This message does not generate any $PFLAM,U sentences.

4.6 VHF Radio Frequency

Frequency currently used by the VHF radio. Up to four frequencies can be specified, but only the first one is mandatory.

Type Specifier	Data Source	Periodicity
VHF	User provided data in form $PFLAM,S,VHF,<Payload>	Once, after $PFLAM,S message

Arguments

Payload	Mandatory	Details	Resolution	Min	Max
<frequencyA>	yes	Frequency expressed in MHz	0.001 MHz	0	40000.000
<frequencyB>	no	Frequency expressed in	0.001 MHz	0	40000.000

		MHz			
<frequencyC>	no	Frequency expressed in MHz	0.001 MHz	0	40000.000
<frequencyD>	no	Frequency expressed in MHz	0.001 MHz	0	40000.000

Example

Sending an information about currently configured radio frequencies: 118.455 MHz and 121.500 MHz:

```
> $PFLAM,S,VHF,118.455,121.500,,*CS
$PFLAM,A,OK,VHF,118.455,121.500,,*CS
```

Message received from a FLARM target DF2000 that shows which frequencies they can receive:

```
$PFLAM,U,2,DF2000,VHF,118.455,121.500,,*CS
```

4.7 Team Name

This message can be used to notify surrounding targets about the aircraft team membership – e.g., for paraglider groups.

Type Specifier	Data Source	Periodicity
TEAM	$PFLAM,S,TEAM,<Payload>	Once, after $PFLAM,S message

Arguments

Payload	Mandatory	Details
<name>	yes	Hex string (max 17 bytes)

Example

Sending an information that aircraft is a member of WWGCAUS team:

```
> $PFLAM,S,TEAM,57574763415553*CS
$PFLAM,A,OK,TEAM,57574763415553*CS
```

This example shows error generated in case the team name exceeds 17 bytes:

```
> $PFLAM,S,TEAM,414141414141414141414141414141414141*CS
$PFLAM,A,ERROR,PAYLOAD TOO LARGE*CS
```

Message received from a FLARM target `DF2000` that shows team WWGCAUS membership:

```
$PFLAM,U,2,DF2000,TEAM,57574763415553*CS
```

4.8 Sensor Measurements

Measurements taken by the airplane and received either by other aircraft, or by ground stations for meteorological data collection. All data is optional.

Type Specifier	Data Source	Periodicity
`SENS`	Manually sent from a ground station with `$PFLAM,S,SENS,<Payload>` command	Once, after `$PFLAM,S` message

Arguments

Payload	Mandatory	Details	Resolution	Min	Max
`<ias>`	no	Integer in m/s, positive values only	`1 m/s`	0	65534
`<altimeter>`	no	Altitude AMSL	`1 m`	-32768	32765
`<vario>`	no	In m/s	`0.1 m/s`	-1000.0	1000.0
`<temperature>`	no	Integer in °C	`0.1°C`	-273.1	3276.5

Example:

Reporting that sensors measure 7.2°C when flying at the altitude of 2999 m at 105 m/s with no vario information.

```
> $PFLAM,S,SENS,105,2999,,7.2*CS
$PFLAM,A,OK,SENS,105,2999,,7.2*CS
```

A message received from FLARM target `DF2000`. The current measured sensor values reported are the indicated air speed (IAS) of 62 m/s, altimeter value of 3052 m, variometer value of 4.1 m/s climb, and the current temperature is 4.3°C.

```
$PFLAM,U,2,DF2000,SENS,62,3052,4.1,4.3*CS
```

4.9 Airport Information

FLARM device can be installed on the ground and broadcast weather/airport information. If the airport has no ICAO code, additional airport name can be transmitted using the `TEAM` message.

Type Specifier	Data Source	Periodicity
`AIRPT`	Manually sent from a ground station with `$PFLAM,S,AIRPT,<Payload>` command	Once, after `$PFLAM,S` message

Airport information frame uses geographical position compression, therefore positions in Tx and Rx packets may differ by up to 0.5m. Additionally the airport must be in a range of 300km for the position to be decoded correctly.

Arguments

Payload	Mandatory	Unit	Resolution	Min	Max
`<ICAO code>`	yes	Basic string (4 characters)	-	-	-
`<lat>`	yes	Degrees, north positive	See the above note		
`<lon>`	yes	Degrees, east positive	See the above note		
`<alt AMSL>`	yes	Feet	`1 ft`	`-16384`	`16383`
`<runway in use>`	no	Tens of degrees	`10°`	`0`	`36`
`<vhf frequency>`	no	In MHz	`0.001 MHz`	`0`	`40000.000`
`<altimeter QNH>`	no	Integer in hPa	`1 hPa`	`700`	`1200`
`<airport status>`	no	See table below	-	`0`	`3`

Airport status

Value	Status	Description
`0`	None	No status information available
`1`	Red	Airport closed and should not be operated
`2`	Yellow	Airport open but caution is required
`3`	Green	Airport is safe to use

Example

Message received from airport LSZF containing information about active runway, frequency in use, local barometric pressure and the airport status:

```
$PFLAM,U,2,DG0000,1,AIRPT,LSZF,47.443333,8.233888,1300,26,121.555,10
13,3*CS
```

Similar as above but no information about frequency is being transmitted:

```
$PFLAM,U,2,DG0000,AIRPT,LSZF,47.443333,8.233888,1300,26,,1013,3*CS
```

Similar as above but only information about airport position is provided:

```
$PFLAM,U,2,DG0000,AIRPT,LSZF,47.443333,8.233888,1300,,,,*CS
```

Example how AIRPT message is triggered on the dataport (position rounding visible):

```
> $PFLAM,S,AIRPT,LSZF,47.4433333,8.2338888,1300,,,,*CS
$PFLAM,A,OK,AIRPT,47.4433336,8.2338872,1300,,,,*CS
```

Non-mandatory items can be left empty, but still must be delimited by commas (following example is missing one coma that separates empty arguments):

```
> $PFLAM,S,AIRPT,LSZF,47.4433333,8.2338888,1300,,,*CS
$PFLAM,A,ERROR,INVALID DATA*CS
```

4.10 Airport Weather

Weather information of airport. Allows to send current METAR (FM-15) data in condensed form. Must be accompanied by Airport Information message.

Type Specifier	Data Source	Periodicity
METAR	Manually sent from a ground station with $PFLAM,S,METAR,<Payload> command	Once, after $PFLAM,S message

Arguments

Payload	Mandatory	Unit	Resolution	Min	Max
<wind direction>	yes	Degrees	10°	0	360
<wind speed>	yes	Knots	1 kt	0	126
<wind gusts>	no	Knots	1 kt	0	126

<wind variation below>	no	Degrees	10°	0	360
<wind variation above>	no	Degrees	10°	0	360
<visibilty>	yes	Meters	1 m	0	9999
<sky condition>	no	See table below	-	-	-
<base height>	no	Meters	1 m	0	16383
<temperature>	yes	°C	1°C	-128	126
<dew point>	yes	°C	1°C	-128	126
<present weather>	no	Basic string (7 characters)		-	-

Type	Description
FEW	Few
SCT	Scattered
BKN	Broken
OVC	Overcast
CB	Cumulonimbus
TCU	Towering Cumulonimbus

Example

Message received from FLARM target DG0000 (airport LSZF like received in accompanying AIRPT message).

```
$PFLAM,U,2,DG0000,METAR,260,7,,190,280,9999,SCT,1200,21,18,-TSRA*CS
```

Sending METAR LSZH 061220Z 26007KT 190V280 9999 -TSRA FEW050 SCT060CB 21/18 Q1021 as a PFLAM message:

```
> $PFLAM,S,METAR,260,7,,190,280,9999,SCT,1200,21,18,-TSRA*CS
$PFLAM,A,OK,METAR,260,7,,190,280,9999,SCT,1200,21,18,-TSRA*CS
```

4.11 Open Broadcast

Arbitrary binary data broadcast to every receiver within range. Allows user-specific use of the network.

Type Specifier	Data Source	Periodicity
BCST	Manually sent with $PFLAM,S,BCST,<Payload> command	Once, after $PFLAM,S message

Arguments

Payload	Mandatory	Details
<data>	yes	17-byte hex (34 characters [0-9A-F])

Example

Sending of an open broadcast message with arbitrary data (note trailing zeros to match 17-byte hex requirement):

```
> $PFLAM,S,BCST,476F696E6720746F20454E53423F000000*CS
$PFLAM,A,OK,BCST,476F696E6720746F20454E53423F000000*CS
```

Attempt to send a message that does not contain exactly 17 hex bytes:

```
> $PFLAM,S,BCST,476F696E6720746F20454E53423F*CS
$PFLAM,A,ERROR,INVALID DATA*CS
```

Message received from a FLARM target `DF2000` that contains arbitrary data:

```
$PFLAM,U,2,DF2000,BCST,6E6F2E20686174652062656172730000000*CS
```

4.12 Open Unicast

Type Specifier	Data Source	Periodicity
`UCST`	Manually sent with `$PFLAM,S,UCST,<Payload>` command	Once, after `$PFLAM,S` message

Arguments

Payload	Mandatory	Details
<ID>	yes	24bit ID
<IDType>	yes	Refer to FTD-012 for more details
<data>	yes	13-byte hex (26 characters [0-9A-F])

Example

Sending a message destined to FLARM target `DF2000` (note trailing zeros to match 13-byte hex requirement):

```
> $PFLAM,S,UCST,2,DF2000,476F696E6720746F2045000000*C
$PFLAM,A,OK,UCST,2,DF2000,476F696E6720746F2045000000*CS
```

Attempt to send a message that does not contain exactly 13 hex bytes:

```
> $PFLAM,S,UCST,2,DF2000,476F696E6720746F2045*CS
$PFLAM,A,ERROR,INVALID DATA*CS
```

4.13 Message Overview

ID	Trigger source	Data source
AREG	Periodic	FLARM config
PNAME	Periodic	FLARM config
ATYPE	Periodic	FLARM config
ACALL	Periodic	FLARM config
VER	Periodic	FLARM config
VHF	User	$PFLAM,S…
TEAM	User	$PFLAM,S…
SENS	User	$PFLAM,S…
AIRPT	Ground station	$PFLAM,S…
METAR	Ground station	$PFLAM,S…
BCST	User	$PFLAM,S…
UCST	User	$PFLAM,S…

Note: Periodic messages are generated automatically and cannot be triggered by the $PFLAM command.

4.14 Error Codes

Code	Error String	Description
0	OK	No error
1	PAYLOAD TOO LARGE	The payload supplied in the message is too large, e.g., due to UTF-8 conversion
2	BANDWIDTH EXCEEDED	The RF bandwidth has been exceeded, e.g.., the transmit queue is full
3	INVALID DATA	The supplied data is invalid, e.g., when a number is expected, but text is supplied, or a number exceeds limits specified in this document.
4	MISSING LICENSE	License for sending a particular type of message is missing e.g., user tries to send METAR message without the GND license.
5	INVALID CHECKSUM	Provided checksum is invalid

4.15 Configuration Interface

4.15.1 Transmission of Periodic Messages

Transmission of periodic messages: `AREG`, `PNAME`, `ATYPE`, `ACALL` and `VER` is enabled by default and does not require a license – every FLARM device with recent enough firmware will transmit those configured fields. Transmission and reception of periodic messages can be disabled in configuration.

Periodic messages are disabled automatically in stealth (PRIV) mode, with NOTRACK or Random ID enabled.

4.15.2 Enabling of Message Functionality

The transmission and reception of RF messages are enabled and disabled via the `MSG` configuration item as a bitfield.

The syntax is as follows:

```
$PFLAC,S,MSG,<Bitfield>
```

The bitfield consists of the following items:

Type	Direction	Value	Default
Enable periodic messages (AREG, PNAME, ATYPE, ACALL, VER)	Rx+Tx	1 – enabled 0 – disabled	1
Enable reception of standard messages (VHF, TEAM, AIRPT, METAR, TASK, SENS)	Rx	2 – enabled 0 – disabled	0
Enable reception of open messages (BCST, UCST)	Rx	4 – enabled 0 – disabled	0

The bitfield can be assembled by addition (logical OR) of all values. The bitfield is transmitted without leading 0x.

Example (Activation)

```
> $PFLAC,S,MSG,7
$PFLAC,A,MSG,7
```

This activates reception and transmission of all messages.

Example (Deactivation)

```
> $PFLAC,S,MSG,0
$PFLAC,A,MSG,0
```

Reception and transmission of all messages is disabled.

Example (Only periodic)

```
> $PFLAC,S,MSG,1
$PFLAC,A,MSG,1
```

Only reception and transmission of periodic messages is enabled.

4.15.3 Licensing

A FLARM Messaging license is required to receive messages, exceptions are the periodic `AREG`, `PNAME`, `ATYPE` and `ACALL` messages that can be received without any additional license.

No license is required for transmission.

4.15.4 FLARM Classic

On FLARM Classic devices functionality is limited to periodic (`PNAME`, `ATYPE`, `AREG` and `ACALL`) message transmission only – no reception is possible.

Accordingly, configuration options are also limited:

Type	Direction	Value	Default
Enable periodic messages (AREG, PNAME, ATYPE, ACALL, VER)	Tx	1 – enabled 0 - disabled	1

5 Implementation Considerations

This chapter contains helpful tips for users implementing the Messaging functionality in their products.

5.1 Data Acquisition

FLARM devices do not store data received by Messaging functionality. It is up to the user to keep a data store that associates received names, callsigns etc. with specific FLARM targets. Each message contains the ID/IDType, and the targets can be associated with this unique identifier.

5.2 Message Scheduling

User should be aware of following limitations:

- There is no guaranteed timeframe between a `$PFALM,S` command and the actual transmission of the message. In optimal conditions the delay should be less than 1s but in case of alarm/high traffic situations, the transmission may be delayed. Users can use the `$PFLAM,R` command to check the status of queued messages.
- `$PFLAM,A` response is generated as soon as the message is successfully inserted into the queue.
- FLARM device can buffer a finite number (10^3) of messages. When this limit is exceeded `$PFLAM,A,<MsgType>,BANDWIDTH EXCEEDED` error will be reported.

Examples

After filling the queue with 10 messages an error response is received:

```
> $PFLAM,S,BCST,476F696E6720746F20454E53423F*CS
$PFLAM,A,OK,BCST,476F696E6720746F20454E53423F*CS
> $PFLAM,S,BCST,476F696E6720746F20454E53423F*CS
$PFLAM,A,OK,BCST,476F696E6720746F20454E53423F*CS
[… few more]
> $PFLAM,S,BCST,476F696E6720746F20454E53423F*CS
$PFLAM,A,OK,BCST,476F696E6720746F20454E53423F*CS
> $PFLAM,S,BCST,476F696E6720746F20454E53423F*CS
$PFLAM,A,OK,BCST,476F696E6720746F20454E53423F*CS
> $PFLAM,S,BCST,476F696E6720746F20454E53423F*CS
$PFLAM,A,ERROR,BANDWIDTH EXCEEDED*CS
```

[3] Exact number may vary between devices.

```
> $PFLAM,S,BCST,476F696E6720746F20454E53423F*CS
$PFLAM,A,ERROR,BANDWIDTH EXCEEDED*CS
```

5.3 Messages Frequency

The number of messaging frames sent per second depends on the current amount of traffic detected by the FLARM device. No more than one message per second is sent from each device. This amount may be further reduced in high traffic/alarm situations.

5.4 Checksum

The Messaging interface uses NMEA 0183 checksums to ensure correctness of data exchanged between user and the FLARM device.

`$PFLAM,U` messages received from FLARM device are always trailed with a checksum. For the testing purposes it is possible to omit the checksum in `$PFLAM,S` command, however it is highly encouraged to use checksums in production systems.

Example

Message reception:

```
$PFLAM,U,2,DF0000,AIRPT,LSZF,47.443333,8.233888,1300,26,121.555,1013
,3*40
```

Message transmission:

```
> $PFLAM,S,UCST,2,DF2000,476F696E6720746F2045000000*76
$PFLAM,A,OK,UCST,2,DF2000,476F696E6720746F2045000000*4C

> $PFLAM,S,UCST,2,DF2000,476F696E6720746F2045000000*75
$PFLAM,A,ERROR,INVALID CHECKSUM*3B

> $PFLAM,S,UCST,2,DF2000,476F696E6720746F2045000000
$PFLAM,A,OK,UCST,2,DF2000,476F696E6720746F2045000000*4C
```