TRX-1090 - ADS-B Receiver Data Port Specification

Record of Revisions

Date	Revision	Description of Change	Inserted by
01 APR 2010	0.1	Initial draft	JG
18 AUG 2010	0.2	NEW: raw mode UPD: updated absolute mode	GG
02 NOV 2010	0.3	NEW: Specification of $FLAA, $FLAU sentences with TRX specific modifications	JG
01 DEC 2010	0.4	NEW: baudrate specifier for raw commands NEW: raw-binary protocol description	GG
07 APR 2011	0.5	UPD: spec of $PGAV sentence (additional fields)	GG
09 SEP 2011	0.6	• UPD: fixed incorrect chapter numbering • NEW: section about compatibility and versions • UPD: spec of $PGAV sentence (additional fields, new field names, more defined content) • NEW: $PGAVC configuration commands • NEW: description of NMEA flight data input • NEW: definition of raw and preprocessed Mode-C data output sentences	GG

Table of contents

1. Protocol / Firmware compatibility issues

This manual is up-to-date with the protocols present in TRX firmware version 28.

If you want to use all of the features described here, you must update your existing TRX to firmware version 28 or newer.

If, for any reason, you install a firmware version newer than 28, please also check for a new version of this dataport manual, as there might be changes in the protocols which are described there, and which affect compatibility to your application software written for older firmware version / protocol specification.

We do our best to make any future changes to any protocol described in this manual as backward compatible as possible, but sometimes, minor changes causing small incompatibilities to software developed for an older firmware version cannot be avoided.

We reserve the right to add new $PGAV sentences and/or additional data fields at the end of previously defined $PGAV sentences, at our descretion. Please provide enough robustness in your application code to cope with this new sentences or fields, at least ignoring them without crashing.

2. System description

Automatic Dependent Surveillance – Broadcast (ADS-B) is a modern ATC system enhancement for broadcasting aircraft position data. Transponders which are connected to a GPS system transmit their own position and other flight data, like call sign, Mode-S address, speed and altitude as well as track and vertical speed. The transponder transmits these data periodically – typically once or twice per second – like a radio station (Broadcast).

TRX-1090 contains a sensitive 1090 MHz receivers with complex signal processing unit. Transponder signals broadcast by other aircraft are received, processed and decoded.

The unit will be connected between FLARM and a FLARM® compatible external display unit, to simultaneously show FLARM-equipped as well as Mode-S transponder equipped aircraft with ADS-B output capability.

The presence of transponder equipped aircraft not broadcasting ADS-B output will be detected and indicated on the connected display as a non directional target.

It is not essential to connect the TRX-1090 to a transponder. The system comes with its own 1090 Mhz receiver. To operate a TRX-1090, no transponder installation is required.

The TRX-1090 provides GPS data received from the FLARM via a dedicated port to supply an ADS-B out capable Mode-S transponder. (e.g. Garrecht Avionic VT-01, VT-02). This device broadcasts the current position message every secound - the message can be received by ADS-B receivers installed in other aircraft, as well as with receivers installed on ground.

The unit can be used as on ground ADS-B receiver for realtime aircraft tracking. The maximum receving range is up to 300 km, depending on the antenna type and location. In the on-ground mode, received data will be provided via the unit's USB interface.

3. Enabling Ground Mode output of the TRX-1090

Usually the TRX-1090 is installed in aircraft for collision avoidance purposes. However, it might be necessary to use it for other application (i.e. aircraft tracking from ground or other aircraft).

For using the TRX-1090 as a ADS-B ground station or for special purposes in airborne application, the unit can be configured to output absolute position data for each aircraft (absolute mode), or 2 different formats of raw data as received over the radio data link.

In airborne mode, position of other aircraft are provided in relative format. This is required to be compatible with the popular FLARM collision avoidance system and related dispays (CDTI), but requires input of GPS position. Due to the limited resolution and range of the relative data format, it is not recommended to use it for ground applications.

System behavour in on ground modes:

- The receiver's sensitivity will be set to maximum value.
- No collision avoidance data will be supplied.
- Data will only be sent via USB port of the TRX-1090. Do not connect any device on the unit's serial ports 1-4. OEMs needing a different port usage scheme, please contact the manufacturer for special configuration options.

To enable one of the ground modes (absolute data, raw ascii or raw binary), please follow the steps below:

- Connect the device to the USB Port of your PC (FTDI USB driver needs to be installed)
- No device may be connected on the serial ports 1-4 !
- Check the Windows hardware manager, if the unit has been detected (COM and LPT interfaces.FTDI COM Serial converter must be detected, e.g. “USB Serial Adapter”)
- Determine the virtual com port the TRX-1090 has been connected to
- Open the serial port of your application (115200 bps, 8N1, no flowcontrol)
- As soon as you receive the string **trxsync**, send <return> (CR-LF)
- send one of the following commands to enable the desired output mode
    - for 115200 bps absolute data mode, send “ABS” <return>
    - for RAW ASCII mode, send “RAWASC<baudrate>” <return> (e.g. RAWASC115200)
    - for RAW binary mode, send “RAWBIN<baudrate>” <return>
    - The <baudrate> number (only supported in RAW modes) is optional. If specified, only 115200, 230400, 460800 or 921600 bps is accepted. If baudrate is not or not correctly specified, 921600 is used by default.
- Send “QUIT” <return>
- change your application’s baudrate to the one needed for the selected protocol

The unit starts broadcasting the selected data sentences now. Please check the description of each output protocol below for further details.

NOTE:

Setting up one of the absolute / raw output format protocols is required after every reset / power down. Otherwise, the unit ALWAYS starts in airborne mode by default !!!

For permanent configuration to absolute protocols, contact the manufacturer.

4. General

4.1. Description of signal strength values

At several places, this manual references signal strength values <rssi> or minimum triggering level <MTL..>.

These values are represented in a unit as directly measured by the receiver ADC (analog-digital-converter) inside the TRX.

In order to calculate dBm values from the <rssi> value, the following formula must be applied:

*signalStrength[dBm] = ((<rssi> – <rssim57dbm> * 6 / 74) – 57*

The calibration constant <rssim57dbm> must be taken once from the $PGAV4 sentence.

During manufacturing, each TRX is calibrated with a 1090 MHz/ –57dBm RF-Source connected to its RF input, and the resulting <rssi> value is permanently stored inside the unit.

4.2. Restrictions

All commands / outputs referencing Mode-A / Mode-C are only present of effective in Mode-A/C enabled firmware versions.

5. Output Protocols

5.1. Absolute Data Output

5.1.1. General

Baudrate : 115200 bps, 8N1, no flow control

5.1.2. sentence $PGAV4, Diagnostic ouput

Syntax:
```
$PGAV4,<val_1>,<val_2>,<val_3>,<val_4>,<val_5>,
<rssi><rssim57dbm>*<nmeachecksum>
```

Meaning:

Outputs calibration data and proprietary information of the TRX-1090 to observe the receiver's performance and failure analysis. This sentence is subjected to change by the manufacturer anytimes without further notice.

NOTE: THIS SENTENCE IS FOR INTERNAL USE BY THE MANUFACTURER ONLY.

Do not use any of this data in your application unless referenced to in any section of this manual.

The <rssim57dbm> field is needed to calculate the signal strength of each aircraft in any of the absolute or raw format modes.

Input / Output: only sent by TRX-1090

Periodicity: sent once per second, for calibration purposes

5.1.3. sentence $PGAV5, Absolute Traffic Positions

Syntax:

```
$PGAV5,
<age_of_position_data>,<hexaddress>,<lat>,<lon>,<gnssAlt>,<baroAlt>,
<flightid>,<track>,<groundspeed>,<verticalspeed>,<rssi>,
<category>,<dataSource>,<rssiDistance>,
<ageMDS>,<ageVelocity>,<agePosition>,<squawk>
*<nmeachecksum>
```

Meaning:

Outputs aircraft's data in absolute coordinates the “on-ground mode” of the TRX-1090. Data will be sent in the absolute coordinates with a speed of 115.200 bps. The receiving range will be set to maximum value.

Data on other aircraft around, intended for use of the TRX-1090 as on ground ADS-B receiver with sensitivity set to maximum value (receiving range up to 300 km, depending on antenna, antenna location etc.).

No collision avoidance will be supplied.

Input / Output: only sent by TRX-1090

Periodicity: sent when available, can be sent several times per second with information on several (but maybe not all) targets around.

Values:

<age_of_position_data>	It's not a timestamp, it's an "age"-stamp which tells you how old the position data in the data stream is. Neglecting communication buffering latency, you can recalculate position of aircraft by extrapolating movement and position to the actual position by way of the age, value in sec.
<hexaddress>	24-Bit Aircraft address (Mode-S Adress), hex 000000 for a nearby aircraft carrying only a Mode-C transponder
<lat>	Aircraft's latitude [degrees], if position is available (from connected Flarm, or from ADSB-out). Empty otherwise.
<lon>	Aircraft's longitude [degrees], if position is available. Empty otherwise.
<gnssAlt>	The aircraft's GPS altitude above WGS-84 ellipsoide, [integer in ft]. Only available for Mode-S equipped aircraft with ADSB-out and GPS altitude enabled.
<baroAlt>	The aircraft’s pressure altitude above 1013.2 hPa pressure level, [integer in ft]
<flightid>	Company callsign for commercial aircraft operating with flight plan Aircraft's registration for non commercial operating aircdraft Depends on data entered into the aircraft's transponder. Only available for Mode-S Transponders with ADSB-out enabled and configured to output Flight-ID.
<track>	The target’s true ground track [degrees]. Integer between 0 and 359. The value 0 indicates a true north track. Only available for Mode-S Transponders with ADSB-out enabled and configured to output GPS speed and track.
<groundSpeed>	The target’s ground speed [kts]. Only available for Mode-S Transponders with ADSB-out enabled and configured to output GPS speed and track.
<verticalspeed>	The target’s vertical speed . Positive values indicate a climbing aircraft. [integer in ft/min].
<rssi>	Strength of received signal [integer]. Can be used for estimating the distance between receiver and target, by using the far-field formula, knowing all antenna parameters. See Chapter 4 for more information.
<category>	Aircraft category value as described in DO260, in hexadecimal format

<dataSource>	0: Aircraft has no (operational) Flarm. Position and velocity data in all fields is derived from ADS-B out only 1: Aircraft is equipped with Flarm (provided that Flarm data is input into the TRX). Position and velocity data in all fields are derived from ADS-B out and Flarm, with individual data fields from Flarm having priority. 8: (only used with address 000000 / Mode A/C data) Received signal contains a code which represents a meaningful (from -1000 to 43000ft) altitude within the predefined altitude band (commands $PFLAC,S,ALTBAND and BAROALT) 9: (only used with address 000000 / Mode A/C data) Received signal contains a code which does not represent a meaningful altitude or is outside the predefined altitude band
<rssiDistance>	Distance between aircraft (target) and receiver (TRX) in [meters], calculated by fieldstrength and stored calibration, and corrected by a tuning factor from config (see chapter configuration commands). Because this value if of poor quality, don't use it as soon as you find position information in the <lat> and <lon> fields. If only Mode-S data without ADS-B position information, or only Mode-C is received from an aircraft's transponder, this value can be used to indicate nearby traffic to the pilot's attention. The signal is unfiltered, so system integrators can do their own filtering.
<ageMDS>	Age [ms] of last received Mode-S signal
<ageVelocity>	Age [ms] of last received ADS-B velocity information
<agePosition>	Age [ms] of last received ADS-B position information
<squawk>	Mode-A squawk selected on aircraft Transponder (4 octal digits)
<nmeaChecksum>	NMEA checksum, calculated as defined in the NMEA specification

<u>Example:</u>

```
$PGAV5,00.1,4780BD,59.103278,10.213035,21585,21110,SAS4744 ,19,383,-
1856,250,20,0,15136,121,580,3516,3702*F7
```

This sentence denotes a target aircraft transmitting the following data

- 24 Bit adress 4780BD, Flight-ID/Callsign: "SAS4744"
- in extrapolated position 59.103278°N, 10.213035°E
- altitude 21585ft above WGS-84 ellipsoid, 21110ft above 1013,2 hPa reference pressure altitude
- true ground track 19°, groundspeed 383kts
- descending with -1856ft/min
- signal strength 250 ADC units
- category 0x20 (defined, but unknown)
- no Flarm data is received from this aircraft
- distance estimated from signalstrength 15136m (example value)
- last Mode-S signal received 121ms ago
- last ADS-B Velocity message received 580ms ago
- last ADS-B Position message received 3516ms ago
- sqawk 3702

5.1.4. sentence $PGAV7, Raw Mode C signal output

$PGAV7,<ver>,<altitude>,<rssi>*<checksum>

field	description of field content
ver	Version of this protocol, currently 2. Future enhancements will have higher numbers
rssi	Received signal strength indication in ADC units (refer to Chapter 4 for more information)
squawk	Received Mode-A squawk (4 digits, octal, with leading zeros)
altitude	Received Mode-C altitude (100ft steps) from Mode A/C detector. If own or other aircraft's transponder replies Mode-A codes that also constitute valid Mode-C codes, false alarms can be triggered. Therefore, it's recommended to also check the signal strength for further evaluation.
checksum	NMEA conformant checksum over the sentence

Note:
When operating the TRX in an airspace area with high traffic density, or with overlapping coverage of several ATC ground stations, this sentence generates a high amount of traffic on the serial interface.

It's recommended to throttled the serial traffic down to the minimum required for proximity detection by raising the Mode-A/C receiver threshold with the $PGAVC,,MTLAC command.

Example:

$PGAV7,2,240,7000,*34
$PGAV7,2,275,7120,19000*09

5.2. Raw ASCII Data output

5.2.1. General

Baudrate : at least 115200; default 921600 bps
Data format : asynchronous, 8 databits, no parity, 1 stopbit, no flow control

5.2.2. Raw ASCII Sentence $PCKOA

Syntax :
$PCKOA,<type><data><nlcb><ts><rssi><res1><res2>

Note :
The PCKOA sentence contains no checksum !
Errors should be primarily detected by the length of the $PCKOA sentence. If it deviates from the default length, some data transmission error might have occured in the TRX device. In this case, ignore the complete sentence and start parsing again when a new $PCKOA sentence starts.

If reliable checksummed data is needed, use Raw Binary Data output format instead.

Example :
$PCKOA,018D06A04560C381467BC4F966736C0016A48C2F00E20300

Each 2 characters of the hexdata represent one byte of data.

Depending on the type of data, the Mode-S standard knows long and short sentences. These can be distinguished by the MSB of the first byte of <data>.
If MSB=1, then it's a long format, if 0, it's a short format.

The long / short determination must be done first, in order to interpret the next 10 bytes of <data> (see below).

example of a long message, already separated into fields
$PCKOA, 01 8D06A045 60C381467BC4F9 66736C 00 16A48C2F 00E2 03 00

example of a short message, already separated into fields
$PCKOA, 01 5D06A045 7305E6 00000000000000 00 16B077EA 00F0 03 00

Data fields:

<type>, 1 byte
example : 01
type of raw data : 1 = Mode-S message, ignore others

<data>, 14 bytes
The <data> field must be subdivided into 3 separate fields, depending on the MSB of the first byte (data[0].

pseudo-code :

```
if (MSB(data[0]) == 1)
        <data> := <DF,CA,AA><ME><CHECKSUM>
else
        <data> := <DF,CA,AA><CHECKSUM><00000000000000>
```

example : 8D06A045
= first 4 bytes of Mode-S datagram (DF,CA,AA)
In DF=11, 17, 18, 19, byte 2..4 contain the ICAO address in plaintext.
first 5 bits of first byte 0x8D is DF (downlink format), here decimal 17 => DF=17; also 0x88 .. 0x8F would all be DF=17
DF=18 could start with bytes 0x90 .. 0x97, and so on.

60C381467BC4F9
= next 7 bytes of message (ME), only available in long messages.
In short messages, this field would not be there, instead the checksum field would directly follow the first 4 bytes and 7 zero bytes would follow after the 3 checksum bytes

66736C
= checksum/address
ignore this data for DF=11, 17, 18 or 19
ICAO address for all other DF messages (DF=0, 4, 5, 16, 20 or 21)

<nlcb>, 1 byte
example : 00
number of low confidence bits (receive quality)

<ts>, 4 bytes
example : 16A48C2F
microsecond timestamp (MSBfirst 32-bit) from power-up of unit

<rssi>, 2 bytes
example : 00E2
received **s**ignal **s**trength **i**ndication (MSBfirst 16-bit); refer to Chapter 4 for more information

<res1>, 1 byte
reserved field, ignore

<res2>, 1 byte
reserved field, ignore

now some sample messages, all from aircraft address 06A045

extended squitter, DF=17:
$PCKOA,018D06A04560C381467BC4F966736C0016A48C2F00E20300

short surveillance reply, DF=4:
$PCKOA,012000183806A0450000000000000000016AE921600E50300

long surveillance reply, DF=20
$PCKOA,01A0001838801BA9352004EF06A04500169DF31A00EC0300

acquisition squitter or allcall reply, DF=11
$PCKOA,015D06A0457305E60000000000000000016B077EA00F00300

TCAS short reply, DF=0
$PCKOA,0102E6183806A0450000000000000000016D7247300DF0300

5.3. Raw BINARY Data Output

5.3.1. General

Baudrate : at least 115200; default 921600 bps
Data format : asynchronous, 8 databits, no parity, 1 stopbit, no flow control

5.3.2. Binary frame structure

The RAWBINARY datastream contains all the data mentioned in 2.2.2, but without the leading NMEA-like $PCKOA header, without comma as byte separators, and with bytes binary instead of printable Hex format. This about doubles the speed and capacity with which data can be transferred.

In order to enable an application to accurately and correctly synchronize to the data stream, each data packet is encapsulated into a frame by means of binary control characters, and additionally protected by a position-sensitive checksum.

In the following, constant message bytes are denoted by their hex value for better readability.

Abstract syntax notation (might not be fully standard conform, feedback is welcome)

STX = 0x02
CKS = 0xFE
DLE = 0x10
data = one or more consecutive databytes (presently 18 bytes)
checksum1 = <SUM8 all data bytes; startvalue=0>
checksum2 = <SUM8 of all checksum1's after each data byte; startvalue=0>
edata = <data, with escaped control-characters, see below for explanation>
echecksum1 or 2 = <checksum1 or 2 with escaped control-characters, see below>

frame = <STX> 0x81 <edata> <CKS> <echecksum1> <echecksum2>

Escaping / Transparency:
If any of the original databytes or the calculated checksum bytes to be transmitted by the TRX has the value of one of the three control bytes (0x02, 0xFE or 0x10), it is inverted (xor'ed with 0xFF) and preceded by the <DLE> control character (0x10) before being transmitted. This has to be done in reverse in the receiver !
This method ensures that all 256 possible values of a byte can occur in the data packet to be encapsulated, without compromising the frame structure by data bytes being falsely interpreted as control characters.

5.3.3. Additional information

1) The checksum algorithm is derived from, but not equal to, the well-known fletcher checksum.

2) For reference only, an implementation of the receiving algorithm in ANSI-C language can be found in the files

trx-rawbinary-reference-parser.c
trx-rawbinary-reference-parser.h

which can be obtained from Garrecht Avionik via a NDA, and integrated into applications free of charge.

5.4. Relative Data Output

5.4.1. General

The TRX-1090 outputs collision avoidance data in FLARM® compatible format. So any of the available compatible FLARM® displays (such as Butterfly Display, V2, V3, V4) can be connected as well as moving map systems (such as FlyMap L, Moving Terrain, LX8000/9000). Data of ADS-B targets are provided in the same manner as FLARM® targets in a common data stream.

The TRX-1090 also provides data from aircraft carrying a transponder without ADS-B capability. So, a new mode, called "undirected warning", has been introduced. No bearing data for such targets are available, but decoded altitude information and an estimated distance. This required a modification of the PFLAU sentence. Depending on the connected display type, two different modes for `<RelativeBearing>` value are available:

1. Empty field:
`<RelativeBearing>` field will be empty, no data at all is provided for this field. To make parser of your system software working properly, be sure that an empty field is not interpreted as zero degree.

2. Bearing jumps in 90° steps:
To make it compatible to existing basic CDTI (such as V2, V3, V4) which are installed in large quantities already and where firmware upgrade is hard to perform, a special format of data output has been introduced. The `<RelativeBearing>` field contains a value which changes in 90° steps in 500 msec intervals.

5.4.2. Sentence **PFLAU**

Syntax:
`PFLAU,<RX>,<TX>,<GPS>,<Power>,<AlarmLevel>,<RelativeBearing>,<AlarmType>,`
`<RelativeVertical>,<RelativeDistance>,<ID>`

Meaning: Operating status and high priority intruder and obstacle data, especially on the most relevant target. This is the main sentence to be used for 3rd party applications and more or less shows what is being visible on the FLARM user interface. PFLAU sentences are not affected by user actions like a temporary suppression, mode or volume selection. This sentence is especially designed for 3rd party applications with very limited CPU performance. Do always track and parse this sentence as it is given the highest priority. Important information might be lost if you only parse PFLAA sentences. Obstacle warnings are currently only given in PFLAU. Note that no mode information is communicated from FLARM to 3rd party devices (e.g. warning vs. nearest mode, sound volume, suppression modes), i.e. 3rd party devices must maintain their own user dialogue for these settings and can do mode-switching regardless of FLARM's mode setting. Inform the user when PFLAU is not received regularly, i.e. not for more than 3s.

Input / Output: only sent by FLARM

Availability on the extension port: always available, no configuration

Availability on the data port: depending on configuration (PFLAC,NMEAOUT)

Periodicity: sent once every second (1.8s at maximum)

Values:

<RX>	Number of devices with unique ID's currently *physically* received regardless of the horizontal or vertical separation, an integer from 0 to 99. Because the processing might be based on extrapolated historical data, *<Rx>* might be lower than the number of aircraft in range, i.e. there might be other traffic around (even if the number is zero). Do not expect to receive *<Rx>* PFLAA sentences, because the number of aircraft being processed might be higher or lower.
<TX>	Transmission status, 1 (hex31) for OK and 0 (hex30) for no transmission
<GPS>	GPS status: 0 (hex30) for no GPS reception, 2 (hex32) for 3d-fix when

	moving and 1 (hex31) for 3d-fix on ground, i.e. not airborne. If *<GPS>* goes to 0, FLARM does not operate as warning device, nevertheless wait for some seconds to issue any warning to 3rd party application's users.
<Power>	Power status, 1 (hex31) for OK and 0 (hex30) for under- or over-voltage
<AlarmLevel>	Alarm level as assessed by FLARM 0 = no alarm (used for no-alarm traffic information) 1 = low-level alarm 2 = important alarm 3 = urgent alarm >3 reserved, ignore
<RelativeBearing>	Relative bearing in degrees from the own position and true ground track to the intruder's / obstacle's position, an integer from -180 to 180. Positive values are clockwise. 0° indicates that the object is exactly ahead. An empty field as well as a change in 90° steps in 500 msec. interval indicated a non directed target (data provided by transponder without ADS-B capability).
<AlarmType>	Type of alarm as assessed by FLARM 0 = aircraft traffic (used for no-alarm traffic information) 1 = silent aircraft alarm (displayed but no alarm tone) 2 = aircraft alarm 3 = obstacle alarm 4 = "INFO alert" >4 reserved, ignore
<RelativeVertical>	Relative vertical separation in Meter above own position, negative values indicate the other aircraft is lower, signed integer. Some distance-dependent random noise is applied to altitude data if stealth for the target is active.
<RelativeDistance>	Relative horizontal distance in m, unsigned integer.
<ID>	6-digit hex value (e.g. "5A77B1") as configured in the target's `PFLAC,ID` sentence. The interpretation is delivered in `<ID-Type>`

Example:

```
$PFLAU,3,1,1,1,2,-30,2,-32,755*
```

FLARM is working properly and currently receives 3 other aircraft. The most dangerous of these aircraft is at 11 o'clock position 32m below and 755m away. It is an important alarm.

```
$PFLAU,2,1,1,1,0,,0,,*
```

FLARM is working properly and receives two other aircraft. They are both out of range.

5.4.3. Sentence `PFLAA`

Syntax:
`PFLAA,<AlarmLevel>,<RelativeNorth>,<RelativeEast>,<RelativeVertical>,<ID-Type>,<ID>,<Track>,<TurnRate>,<GroundSpeed>,<ClimbRate>,<AcftType>`

Meaning: Data on other aircraft around, intended for 3rd party devices with sufficient CPU performance. This sentence should be treated with utmost flexibility and tolerance on a best effort base: Individual fields can be omitted. This sentence is only delivered if the data-port Baud rate is 19.2kBaud or higher. In case of serial port congestion or high CPU load this sentence may be omitted for several objects independent of the alarm level. Obstacle information is not delivered with this sentence. Use a combination of <ID-Type> and <ID> to track the same target, as it might not appear every second. Note that in case of many targets within range, individual targets including the most dangerous one might not be delivered every second, not regularly and maybe not even at all due to less strict priority handling for the PFLAA sentence. Always use PFLAU as primary alarm source. Usually, but not always, the last PFLAA sentence is the one causing the PFLAU content. The other PFLAA sentences are not ordered. Do not expect to receive PFLAU *<Rx>* times PFLAA sentences, because the number of aircraft being processed might be higher or lower. PFLAA sentences can be based on extrapolated historical data. PFLAA sentences are limited to other aircraft with a horizontal distance of less than the configured range (default is 3km) and a vertical separation of less than 500m. Non-moving aircraft are suppressed.

Input / Output: only sent by FLARM

Availability on the extension port: not available, no configuration

Availability on the data port: depending on configuration (PFLAC,NMEAOUT and PFLAC,BAUD)

Periodicity: sent when available and port Baud rate is sufficient, can be sent several times per second with information on several (but maybe not all) targets around.

Values:

<AlarmLevel>	Alarm level as assessed by FLARM 0 = no alarm (pure traffic, limited to configured range (default is 3km) and 500m altitude difference) 1 = low-level alarm 2 = important alarm 3 = urgent alarm
<RelativeNorth>	Relative position in Meter true north from own position, signed integer If last digit is even, the position from the aircraft is based on ADSB-out signals only. If odd, the position is based on values received from the other aircrafts Flarm.
<RelativeEast>	Relative position in Meter true east from own position, signed integer
<RelativeVertical>	Relative vertical separation in Meter above own position, negative values indicate the other aircraft is lower, signed integer. Some distance-dependent random noise, changing every second is applied to altitude data if stealth for the target is active.
<ID-Type>	Defines the interpretation of the following field <ID> 1 = official ICAO aircraft address 2 = stable FLARM pseudo-ID (chosen by FLARM)
<ID>	6-digit hex value (e.g. “5A77B1”) as configured in the target’s PFLAC,ID sentence. The interpretation is delivered in <ID-Type>
<Track>	The target’s true ground track in degrees. Integer between 0 and 359. The value 0 indicates a true north track. This field is empty if stealth for the target is active.
<TurnRate>	The target’s turn rate. Positive values indicate a clockwise turn. Signed decimal value in °/s. Currently omitted. Field is empty if stealth for the target is active.
<GroundSpeed>	The target’s ground speed. Decimal value in m/s. The field is forced to 0 to indicate the aircraft is not moving, i.e. on ground. This field is empty if stealth for the target is active while the target is airborne.
<ClimbRate>	**The target’s climb rate. Positive values indicate a climbing aircraft. Signed decimal value in m/s. This field is empty if stealth for the target is active.**
<AcftType>	Up to two hex characters showing the aircraft type

0 = unknown 1 = glider / motor-glider 2 = tow / tug plane **3 = helicopter / rotorcraft** **4 = parachute**	5 = drop plane for parachutes 6 = hang-glider (hard) 7 = para-glider (soft) 8 = powered aircraft **9 = jet aircraft**	10 = flying saucer (UFO) 11 = balloon 12 = airship 13 = unmanned aerial vehicle (UAV) **15 = static object**

Example:

```
$PFLAA,0,-1235,600,220,2,DD8F12,180,-4.5,30,-1.4,1*
```

There is a glider in the south-west direction, 1.7km away (1.2km south, 0.6km east), 220m higher flying on south track with a ground speed of 30m/s in a slight left turn with 4.5°/s turning rate, sinking with 1.4m/s. Its ID is a static FLARM-ID “DD8F12”. The aircraft has operational Flarm onboard, whose signals are received.
There is no danger.

6. Input Protocols

6.1. General

Data encoding
All input (flight data or configuration setting) during normal operation of the TRX is performed by NMEA styled commands.

Checksum
An optional NMEA compatible checksum can be used in all data sent to the TRX. If a sentence to the TRX contains a checksum, it MUST be valid for the command to be accepted by the TRX.

NMEA coded output from the TRX always contains a checksum.

6.2. Configuration Commands

The TRX can be temporarily reconfigured away from the default configuration specified by the manufacturer, by the below defined commands.

In order to safeguard against accidential or nonrecoverable misconfiguration, and to reduce excessive wear of the non-volatile configuration memory device, the changed settings are lost after the next restart.

At each startup, an OEM application should, if needed, reconfigure the module to the settings required for proper operation. Most of the frequently needed settings (e.g. the MTL settings) are of dynamic nature anyway, so this shouldn't cause big problems.

Also, the currently active configuration settings can be read back from the TRX for whatever purpose.

- Each command / answer has the following structure:

  $PGAVC,*<action>*,*<setting>*[,*<value>*,*<value> ...*]*<*checksum*>

  and is / must be followed by a <CR/LF> (end-of-line) character combination.

  All characters must be in upper-case, and all setting names must be fully given; no abbreviations are allowed.

- Commands must be sent to the TRX in same baudrate and serial parameters as is active for the output protocol

- command parameters in brackets [ ] are optional; e.g. a request for readback of a setting's value needs not be given a value.

Explanation of the command parameters:

<action>

S	request to write a setting into TRX. Value required.
R	request to read a setting from TRX. No value required.
A	answer from TRX An answer is given to both S and R requests, containing the most recent content of a setting.

<checksum>

NMEA compatible checksum

<value>

value of a setting to be written into or read out of the TRX. This can be multiple values, separated by commas, if the command can modify multiple settings at once

<setting>

setting	valid values (integer)	description
ACFTADDR	hex 000001 to EFFFFF	ICAO 24-bit aircraft address of own aircraft. In the $PGAV5 protocol, Mode-S derived data from aircraft with this address is not output.
NMEAPGAV <sentencenr>, <activation>	sentencenr : 7 to 9 activation : 0 and 1	Activation (1) or deactivation (0) of experimental $PGAV sentences ($PGAV7, 8 or 9). Example : $PGAVC,S,NMEAPGAV,7,1 activates $PGAV7 sentences (raw Mode-A/C output).
MTLS	64 to 800	Mode-S and ADS-B minimum triggering level (sensitivity) of the TRX in ADC units. The higher, the less sensitive.
MTLAC	64 to 800	Mode-A/C minimum triggering level.
BAROALT	-1000 to 50000	Sets the barometric (referenced to 1013.2 hPa) altitude (in feet) of the device to a fixed value. This value overrides any value previously set by $PGRMZ data input. Setting a value of –1500 cancels the previously programmed altitude and reverts to receiving altitude from $PGRMZ data input. The altitude is used in conjunction with ALTBAND to form an altitude filter for data output.
ALTBAND	-50000 to 50000	All aircraft having a baro altitude differing more than this value (in feet) from own baro altitude are categorized “out-of-band”. In $PGAV5 sentences, out-of-band • Mode-S signals are not output, unless baro altitude is not transmitted by the transponder, or own altitude is not valid. • Mode-C signals (address = 000000) are classified into dataSource 8 or 9 (inband, outband) For safety reasons, no altitude discrimination is done when this value is zero, other aircraft’s baro altitude is unknown, or own altitude is unknown. In this case, all received Mode-C signals are used for the detection algorithms, regardless of their altitude content.
EXTRAPOL	0 and 1	Disable (0) or enable (1) position extrapolation algorithms inside the TRX. If an application wants to do its own extrapolation / tracking based on the fix coordinates and ages contained in the $PGAV5 sentence, extrapolation must be disabled.
SWVER	N/A (readonly)	Software version information
HWVER	N/A (readonly)	Hardware version information
RXSTUNE	-128 to 127	74 units represent a change of 6dB (distance factor 2) applied to the received signal strength before estimating a distance from it. This setting influences the <rssiDistance> field in $PGAV5 sentences, but not any raw <rssi> fields of any protocol.
CAL57	0 to 1023, readonly	Receiver calibration value (ADC units) stored in the TRX, measured during factory testing with a CW signal of 1090 MHz, –57dBm applied to the receiver input. This value is needed by an application doing its own distance estimations from the received signal strength from each aircraft.
RESTART	1	The TRX will be immediately restarted in its factory configuration (from programmed configuration file). All temporary settings made by $PGAVC,S, commands are lost.

6.3. empty page

This page has intentionally been left blank

6.4. Flight Data

Note:
By default, flight data is input by an original or OEM Flarm into the dedicated Flarm Port (port 4) of the TRX.

To enable acceptance of flight data sensed by a non-Flarm GPS device or on a different serial port (e.g. the same port as used for output of absolute traffic data), you MUST obtain a special configuration file from the manufacturer.

6.4.1. Altitude

Baro (pressure) - Altitude is input into the TRX via the following NMEA sentence

$PGRMZ,<altitudeFt>,F,2[*<checksum>]

<altitudeFt>
Altitude in feet

<checksum>
optional, NMEA compatible checksum

Input of baro altitude is required for

- proper decoding and filtering of signals from Mode-C Transponders. Without baro altitude input, no filtering will be performed.
- suppression of signals received from other aircraft separated in altitude so much that they are not a danger for the own aircraft (not needed for OEM applications which use the TRX in sensor-only mode).
- precise operation of collision detection algorithms (not needed for OEM applications which use the TRX in sensor-only mode)

Recommended update rate is once per second, timeout occurs after 60 seconds.

6.4.2. GPS Fix

GPS fix input is not needed for OEM applications which use the TRX in sensor-only mode.

$GPGGA (for GPS altitude)

$GPRMC (for GPS position fix, movement vector and geoidal altitude separation)