# MANUAL / USER GUIDE / HANDBOOK

## OPS-SAT UHF specifications

European Space Agency
Agence spatiale européenne

# APPROVAL

| | | |
|---|---|---|
| **Title** OPS-SAT UHF specifications | | |
| **Issue Number** 1 | **Revision Number** 0 | |
| **Author** Benjamin Fischer | **Date** 14/07/2019 | |
| **Approved By** | **Date of Approval** | |
| | | |

# CHANGE LOG

| Reason for change | Issue Nr. | Revision Number | Date |
|---|---|---|---|
| | | | |

# CHANGE RECORD

| Issue Number 1 | | Revision Number 0 | |
|---|---|---|---|
| Reason for change | Date | Pages | Paragraph(s) |
| | | | |

# DISTRIBUTION

| Name/Organisational Unit |
|---|
| |

**Table of contents:**

# 1      REFERENCE DOCUMENTS AND SOURCES

- **GomSpace Nanocom AX100 datasheet**
  https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanocom-ax100-33.pdf

- **AX.25 Link Access Protocol for Amateur Packet Radio**
  https://www.tapr.org/pdf/AX25.2.2.pdf

- **High-Level Data Link Control**
  https://en.wikipedia.org/wiki/High-Level_Data_Link_Control

- **Cube Sat Protocol (CSP)**
  https://github.com/libcsp/libcsp

- **CCSDS Blue Book Telemetry Channel Encoding**
  https://public.ccsds.org/Pubs/101x0b6s.pdf

- **CCSDS Blue Book Space Packet Protocol**
  https://public.ccsds.org/Pubs/133x0b1c2.pdf

- **CCSDS Blue Book MAL Space Packet Transport Binding**
  https://public.ccsds.org/Pubs/524x1b1.pdf

- **CCSDS Red Book Mission Operations – Common Services**
  https://public.ccsds.org/Lists/CCSDS%205220R1/522x0r1.pdf

- **GNU Radio Manual and C++ API Reference**
  https://www.gnuradio.org/doc/doxygen/

- **GPredict User Manual**
  http://www.w1npp.org/events/2010/2010-f~1/SATELL~1/GPREDI~1.PDF

## 2    INTRODUCTION

This document describes the UHF demodulation and decoding specifications for the ESA OPS-SAT mission. OPS-SAT is going to use the UHF radio for transmitting the UHF beacon as a heartbeat signal, nominal operations are done on a dedicated S-Band frequency.

## 3    UHF RADIO SPECIFICATIONS

| Name | Specification |
|---|---|
| Space segment transceiver | GomSpace NanoCOM AX100 |
| Frequency | UHF – 437.2 MHz |
| Modulation | GMSK |
| Occupied Bandwidth | 25 kHz |
| Baudrate | 9k6 |
| Modulation Index | 0.5 |

## 4    BEACON SPECIFICATIONS

The NanoCOM AX100 is configured in mode 6, which means that the data is encapsulated in a HDLC + AX.25 frame.

### 4.1    Frame encoding

The format is as follows, the number in the brackets represent the number of bytes being used:

| | [110] NanoCOM AX100 frame | | | |
|---|---|---|---|---|
| [1] Flag | [16] AX.25 HDR | [94] Data Field | [2] CRC – 16 CCIT | [1] Flag |

The whole frame is
- NRZI encoded,
- G3RUH/K9NG scrambled,
- LSB encoded (apart from the 16 bit CRC),
- and has a size of 110 bytes (without the flags).

The flag being used to separate frames is 0x7E (01111110). The NanoCom also sends 50 of those in the preamble for receiver synchronization.

After NRZI decoding and descrambling the AX.25 header becomes readable and does not require any further processing. The call signs are
- "**DL0ESA**" for the ground station, and
- "**DP0OPS**" for the satellite.

## 4.2    Data field encoding

The data field ends with a 32 bit cyclic redundancy check (Castagnoli), followed by a Reed-Solomon (223,255) code block, and is finally being scrambled with the CCSDS polynomial

$$h(x) = 1 + x^3 + x^5 + x^7 + x^8.$$

After descrambling, Reed-Solomon decoding, and the removal of the CRC, the data field becomes 58 bytes, as shown below.

| [94] Data field | | |
|---|---|---|
| [58] Payload data | [4] CRC-32C | [32] RS code block |

The payload data itself is a Cube Sat Protocol (CSP) packet and starts with a four byte header with the following content :

- Priority: 3
- Source: 5
- Destination: 10
- Destination Port: 31
- Source Port: 0
- Flags: 0

The remaining 54 bytes contain the telemetry table #4 of the NanoCom AX100 as shown in the following table.

| Offset | Description | Data type |
|---|---|---|
| 0x00 | Board temperature (near MCU) | Signed 16 bit integer |
| 0x02 | PA temperature (near PA) | Signed 16 bit integer |
| 0x04 | Last received RSSI | Signed 16 bit integer |
| 0x06 | Last received RF error | Signed 16 bit integer |
| 0x08 | Number of TX packets since reboot | Unsigned 32 bit integer |
| 0x0C | Number of RX packets since reboot | Unsigned 32 bit integer |
| 0x10 | Number of TX bytes since reboot | Unsigned 32 bit integer |
| 0x14 | Number of RX bytes since reboot | Unsigned 32 bit integer |
| 0x18 | The currently active system configuration | Unsigned 8 bit integer |
| 0x19 | The number of reboots | Unsigned 16 bit integer |
| 0x1B | The cause of the reboot (AVR32) | Unsigned 32 bit integer |
| 0x1F | The timestamp of the last valid packet | Unsigned 32 bit integer |
| 0x23 | The current background RSSI level | Signed 16 bit integer |
| 0x25 | Total TX duty time since reboot | Unsigned 8 bit integer |

| 0x26 | Number of TX packets (total) | Unsigned 32 bit integer |
|---|---|---|
| 0x2A | Number of RX packets (total) | Unsigned 32 bit integer |
| 0x2E | Number of TX bytes (total) | Unsigned 32 bit integer |
| 0x32 | Number of RX bytes (total) | Unsigned 32 bit integer |

# 5    NON-BEACON SPECIFICATIONS

OPS-SAT is not planning to downlink any housekeeping data on the UHF frequency, except for the beacon signal. Nominal operations are done on a dedicated S-Band frequency. Only in case of a S-Band radio failure, the UHF link might be used. For decoding those packets the same instructions as before can be followed. The payload data will also be a CSP packet, but the actual data field of the CSP packet itself will contain a Mission Operations (MO) Service Space Packet Protocol packet instead of the beacon table. The applicable standards are described in the CCSDS 'Blue books' referred to in the Referenced Documents section.

For an SPP over CSP packet, the CSP header has the following values:
- Priority: 2
- Source: 1
- Destination: 10
- Destination Port: 10

# 6    APPLICATION EXAMPLE

A clean UHF beacon recording and GnuRadio applications for receiving, demodulating, and decoding it, as well as a full GUI Desktop application can be found in the GitHub repository https://github.com/esa/gr-opssat.