This document contains the specification of commands and status information for the on-board computer and C-band communication module interface.

Contents

List of acronyms

BB	Base-band
OBC	On-board Computer
CBTM	C-band Transmitter Module
CS	CubeSat
HW	Hardware
KPI	Kyiv Polytechnic Institute
LSS	Long Synchronization Sequence
RF	Radio Frequency
SR	SatRevolution
RGS	CBSR Ground Station
CGS	Control Ground Station
SS	Space Segment
SP	Session Plan
RS	Retransmission Schedule
SI	Status Info

1. Commands

Commands are messages sent from the OBC to the CBTM via a set of interfaces (pin/UART/ETH) to control the settings and operation of the communication module. The commands may be generated by OBC and/or transmitted from the CGS and transferred via the OBC to the CBTM. Typically, though, commands are embedded in a session plan (SP), which allows them to be executed on a specific time epoch.

1.1. Operation of the communication module

1.1.1. Power on/off

Interface: RGS > CGS > OBC > pin

Switches the transmitter module on/off, based on the session start and end time.

Note: The powering off is realized in a somewhat distributed manner, i.e., the physical *plugging out the cable* is made by the OBC. Albeit, the time at which the power off operation should be made is based on a particular session plan, which in turn is delivered from Earth.

1.1.2. Modes of operation

Set during communication session

Controls mode of operation of the transmitter module

1.1.2.1. Module shutdown-ready

Interface: OBC > UART

Prepare module to be electrically cut-off, which include (not exhaustive):

- Stop software TX-DATA plane component from running [Note: software TX-DATA plane component should send EoT information to the FPGA or inform FPGA that transmission is about to end, in which case it is the FPGA's responsibility to send EoT]
- Stop scheduler (if applicable, i.e., scheduler hasn't finished running)
- Stop FPGA
- Other (e.g., un-mounting data partition)
- Only serial communication may work at this point and only with a limited support, i.e., we may allow reading the mode of operation in order to provide to the OBC way of ensuring that the CBTM entered the shutdown-ready mode. **No other commands / features are supported when entering this mode**. This mode should be solely used to prepare CBTM for electrical cut-off. **After entering this mode no communication will take place until the module restart**.

1.1.2.2. Retransmission subsystem reset (internal-RS)

Interface: RGS > CGS > OBC > ETH

Restarts the retransmission state machine

1.1.3. Status information

1.1.3.1. Send status information via OBC (internal-SP/RS)

Interface: RGS > CGS > OBC > ETH/UART

CBTM sends status information via TT&C downlink:

- on predetermined epochs in the SP

1.1.3.2. Send status information via CBSR downlink (internal-SP/RS)

Interface: RGS > CGS > OBC > ETH/UART

CBTM sends status information to the ground station:

- on predetermined epochs in the SP

Note: The 'internal' commands affects the CBTM operation only; they can be included in SPs and RSs and, therefore, are transferred transparently to other subsystems.

2. Parameters

Parameters are used to control the settings of CBTM and passed to the module using the SP file sent from the CGS via its TT&C uplink. In case no file exists for current communication session, the session uses default settings.

2.1. Communication module settings (all internal-SP)

Interface: RGS > CGS > OBC > UART

2.1.1. Carrier frequency

Set on system start-up

Sets the operating frequency of the transmitter module

Default value: 5840 MHz

x [MHz]

Can be set only via selecting appropriate filter file
It is static configuration, i.e., cannot be changed during TX session.
Access: **N/A set via filter file!**

2.1.2. Signal bandwidth

Set on system start-up

Sets the bandwidth of the transmitted signal (via sampling frequency)

Default value: B1 (1 MHz)

Can be set only via selecting appropriate filter file
It is static configuration, i.e., cannot be changed during TX session.
Access: **N/A set via filter file!**

Note: Since signal bandwidth is set via filter file there is no corresponding HW value.

The following bandwidths are available:

- B1 (1 MHz)
- B2 (1.25 MHz)
- B3 (5 MHz)
- B4 (10 MHz)
- B5 (20 MHz)

2.1.3. Channel coding rate

Set during communication session

Sets the turbo coder rate

Default value: R7 (0.19)

Access: `fpgatrx/code_rate`

2.1.3.1. R1 (0.91) – HW: 0

2.1.3.2. R2 (0.83) – HW: 1

2.1.3.3. R3 (0.70) – HW: 2

2.1.3.4. R4 (0.57) – HW: 3

2.1.3.5. R5 (0.41) – HW: 4

2.1.3.6. R6 (0.285) – HW: 5

2.1.3.7. R7 (0.19) – HW: 6

2.1.4. F_AMB length

Set on system start-up

Sets the length of the frequency synchronization preamble (used also as the length of the AGC preamble)

Default value: M3 (1040 S) [symbols]

Decisive are values from FPGA implementation (provided by Marcin)

Access: `fpgatrx/frequency_offset_estimation_preamble_length`

2.1.4.1. M1 (272 S) – HW: 0

2.1.4.2. M2 (528 S) – HW: 1

2.1.4.3. M3 (1040 S) – HW: 2

2.1.5. Shaping filter alfa

Set on system start-up

Sets the roll-off factor of the shaping filter

Default value: 0.35

2.1.5.1. A1 (0.35) – HW: 0

2.1.5.2. A2 (0.22) – HW: 1

Access: `fpgatrx/shaping_filter_aplha`

2.1.6. Frame length

Set on system start-up

Sets the frame length, interpreted as the number of code words between two consecutive time synchronization preambles (1-65535)

Defaut value: 100

X

Access: `fpgatrx/number_of_subframes_in_rf_frame`

2.1.7. AGC preamble repetition

Sets the number of repetitions of gain preamble (1-15).

Defaut value: 3

X

Access: `fpgatrx/gain_preamble_number`

2.1.8. RF Transmit power

Set on system start-up

Sets the transmit power level, relative to +33dBm (x=0)

Default value: 3 (30dBm)

x [dB]

MINUS (we can set attenuation only!!!)
Access: `ad9361/out_voltage0_hardwaregain`

2.1.9. P_AMB interval

Set on system start-up

Sets the number of PC_WORDs between two consecutive P_AMB midambles (1-3).

Default value: 2

X

Access: `fpgatrx/p_amb_interval`

Important: value “3” can not be used together with the highest (0.91) coding rate

2.2. Operation of the communication module (all internal-SP)

Interface: RGS > CGS > OBC > UART

2.2.1. Data transfer

Set on system start-up [**With the exception of EoT – see below**]

Controls the type of transmitted data: continuous T_AMB, regular frame with PN data, regular frame with coded PN data, regular frame with user data

It should be possible to by-pass software TX-DATA plane component. However, regardless of that, software TX-DATA plane component should be somehow informed about the fact that no more data should be sent to the FPGA.

Alternatively, FPGA implementation can stop receiving data from software TX-DATA plane component, which in turn may wait until situation changes.

2.2.1.1. Send continuous preamble(s) – HW: 3

2.2.1.2. Send uncoded PN data – HW: 2

2.2.1.3. Send coded PN data – HW: 1

2.2.1.4. Send user data – HW: 0

2.2.2. EoT

This should be set by the software TX-DATA plane component in the following cases:

- There is no more data to be sent (a.k.a. low data supply)
  [**Note 1**: it might be impossible to reliably detect such a situation, since OBC may decide to send more data to CBTM during the transmission window / session]
  [**Note 2**: Enforcing detection of such a state will prevent OBC from sending more data to CBTM during transmission window / session]
  [**Conclusion**: due to the note 2 it is advised to **not detect** situation of low data supply and **not send EoT** in case of the alleged low data supply]
- There is an information from the OBC that the CBTM is about to be shutdown

2.3. Transfer control

CBTM sends information as fast as it possible in independent and numbered segments. On RGS they are collected and their sequence numbers are analyzed.

Missing parts are identified and feedback information is prepared.

Return information contains commands witch confirmations of received parts of data or request of retransmissions. Confirmation of data with some id represents also correct receiving earlier data parts.

The format of confirmation is a binary stream of information packed with gzip and BASE64 code.

Each command contains two parts - id of command and its parameters.

Possible types of commands:

1 - Confirmation of one data segment - param: id of confirmed segment of data

2 - confirmation of range of data segments - params: id of start and end of segment of data

3 - request of repetition of transmission of one segment - param: id of segment data to be retransmitted

4 - request of repetition of range of transmitted segments - params: id of start and end of range data to retransmission.

Commands are packed into binary file with size no grater then 160 bytes and sent IP_RS interface from GSR to CGS.

They are loslessly transmitted to OBC and received by CBTM.

3. Status info

Status info are messages sent from the CBTM to the OBC (and forwarded to CGSA on the TT&C downlink) or sent by the CBTM directly to the RGS (via C-band downlink) to inform about the current status of the transmitter module.

3.1. Communication module status

Interface: UART > OBC > CGS > RGS or internal buffer

3.1.1. UART-level realization

Module status information are delivered from the CBTM to the OBC via UART interface. An umbrella of file-exchange related commands specified as protocol buffers message is used to ensure a correct file-exchange between these two entities. The umbrella of these messages follows the CRD rather than CRUD pattern, i.e., create, read, delete; there is no update of the whole file. Such an implementation is not a constraint, since any complete UART file exchange is followed by a DL transmission (via low-speed OBC-to-earth link), thus updating such files serves little purpose.

3.1.2. File format

Status information is gathered in a YAML file

3.1.3. Example file

An example of YAML file containing status information is shown below.

```
cbtm_status_info:
  time: 1970-01-08 10:56:09
  uptime: 7 [d] 10 [h] 56 [m] 10 [s]
  trx_subframe_all: 3841
  xadc_vccint: "1.000488 [V]"
  xadc_temp: 52.538922 [C]
  ad9361_temp: <READING AD9361 TEMPERATURE FAILED!>
  trx_power_level: "[WHICH POWER LEVEL? NOT IMPLEMENTED YET]!!!"
  sd_card_usage:
    /dev/mmcblk0p1:
      size: 511.0M
      used: 6.9M
      free: 504.1M
      usep: 1%
    /dev/root:
      size: 13.7G
      used: 80.6M
      free: 12.9G
      usep: 1%
  extra: none
```

3.1.4. Module status

Sent during communication session (when requested)

3.1.4.1. module internal time (source: PS)

Access: `e.g.: date +"%Y-%m-%d %H:%M:%S"`

3.1.4.2. time since module wakeup (source: PS)

Access: `e.g.: uptime | sed 's/^.*up *//' | sed 's/,.*//'`

3.1.4.3. number of frames transmitted since wakeup (source: PL)

Access: `fpgatrx/trx_subframe_all`

3.1.4.4. module internal voltage (source: ZYNQ)

Access: `e.g.: cat xadc/in_temp0_raw`

3.1.4.5. module internal temperature (source: Zynq)

Access: `e.g.: cat xadc/in_voltage0_vccint_raw`

3.1.4.6. module internal temperature (source: AD9361)

Access: `e.g.: cat ad9361/in_temp0_input`

AD9361 die temperature: https://wiki.analog.com/resources/tools-software/linux-drivers/iio-transceiver/ad9361#die_temperature_reading

3.1.4.7. transmit power level (source: AD9361)

Access: **`fpgatrx/????`**

3.1.4.8. SD card usage (optional)

Access: `e.g.: df -h /dev/mmcblk0p* | tr '\n' ':'`

3.1.4.9. Extra information (optional)

Access: `unspecified`

4. Session control file format

Satellite link budget varies from session to session as well as during every communications session and is determined by the relative position of the satellite and the ground station and other parameters, e.g. weather conditions. To maximize the radio link efficiency some transmission parameters (see Section 2.1) can be changed during the mission. This is implemented using session control file which is sent from the RGS via the CGS and its TT&C uplink. The file is downloaded by the CBTM on its startup. The communication module parses the file and sets the required parameters according to the current epoch.

4.1. File name

File name is related to the communication session, it indicates date, ground station number and session number.

4.2. File format

Session control is defined in a YAML file (an example of which is provided below).

This file is composed of two main section, mainly static configuration (field: `static_config`) and session plan (field name: `session_plan`).

```
static_config: |-
   --c_freq=2.2
   --bandwidth=24

session_plan:
  - epoch: 2021-08-19 16:35:30
    config:
      - 1 > fpgatrx/code_rate
      - 1240000000 > ad9361/sampling_frequency
  - epoch: 2021-08-19 13:55:55
    config:
      - 4 > fpgatrx/code_rate
      - 1230000000 > ad9361/sampling_frequency
  - epoch: 2021-08-19 13:56:30
    config:
      - 0 > fpgatrx/code_rate
      - 1250000000 > ad9361/sampling_frequency
```

Session configuration is a normal multiline string that contains options, which are in turn passed to the AD9361 chip configuring application (`/usr/local/bin/ad9361-config.run`). Parameters defined here are typically only those that cannot be changed during the communication window, e.g., frequency carrier, FIR filter configuration, bandwidth, etc.

Session plan is a list of dictionaries that each one of contains two keys: `epoch` and `config`.

Key `config` defines the list of configuration entities that are changed during the communication window. In the example above two parameters are changed, the code rate and the sampling frequency. Note that in the final system only code rate will be changed (the sampling frequency is used solely to highlight syntax). Note also that although they might look like, the entries of the list of configuration entities are not passed to the echo program via std::system or an analogous call. **Any similarities between the syntax provided for the session configuration and the syntax of the echo program are purely incidental**.

Under the epoch key is defined the time at which a given configuration is applied. In theory, this time should be the GMT time, but in reality whatever time passed to the CBTM from the OBC is fine as long as the CBTM is configured with the exact same time. Format of the epoch key is fixed and it should be **YYYY-MM-DD HH:MM:SS**.

4.3. Control commands

Control commands are realized as a part of a session plan specification; since it is the most vital communication between CBTM and OBC the file-exchange mechanism is implemented.

Non-file-exchange related UART commands are:

- power off request / response (re-named in protocol buffers as (Get)ExperimentTime),
- get time request / response (re-name in protocol buffers as (Get)Time

UpAPI	Request	Response
Message	GetExperimentTime	ExperimentTime
File	requests.proto	responses.proto

4.4. Time configuration

Time configuration is realized via UART’s Up API v4 protocol buffer(s) specification. Specific Up API messages and protocol buffers files with their definitions are coarsely delineated below.

UpAPI	Request	Response
Message	GetTime	Time
File	requests.proto	responses.proto

5. Retransmission schedule file format

Due to the satellite radio link variations some of the transmitted packets may become corrupted and erroneously decoded in the receiver. The retransmission mechanism helps to minimize the number of lost files, however, it requires feedback information at the transmitter. This information is sent as so called 'retransmission schedules', using the TT&C uplink.

5.1. File name

Files are named automatically and placed on OBC to be read by CBTM. Name of them represents their order.

5.2. File content

The content of RS files is a binary stream of data packed by gzip. Data contains stream of commands defined in "Traffic control" chapter of this document (2.3)

5.3. Size of file

Each portion of data in one RS file is no greater than 160 bytes (due to TT&C uplink parameters)

…