OPEN
SOURCE
SATELLITE

OSSAT Mercury Ground Segment Requirements Specification

VERSION 08
RELEASED

Reference:	KS-DOC-01104
Date:	20th September 2021
Author:	Paul Madle

Table of Contents

1 Document History

See the following record of revisions:

Document Revision	Document Status	Change Description	Sharepoint version
01	DRAFT	Initial document, for review by R Mota, A Brown, V Anderson.	2.0
02	DRAFT	Revised following review by R Mota. A Brown and V Anderson. Major revisions: Add GUI wireframes to guide development. Minor revisions, addition of an abort feature on file upload and clarifications throughout.	3.0
03	RELEASED	Revised following review of initial Mercury SS spec. Changes include: - Rebranding to Mercury GS - Introduction of Test Interface section needed in order to test Mercury SS (new section 5.7). Also PLAT_COMMS_00045 was altered to exclude test interface. - Altered PLAT_COMMS_00055 telecommand rejection frame to include more detailed list of validation failures. - Added a telemetry request rejection, this has altered requirements PLAT_COMMS_00160, PLAT_COMMS_00105 (additions to GUI). Also added PLAT_COMMS_00165.	6.0
04	RELEASED	Altered requirements according to Rel2 requirements that were actually fulfilled	7.0

		in Rel1. Also broke Rel2 down into Rel2.1, 2.2 and 2.3 to make collaboration easier.	
05	RELEASED	Added format of telemetry request rejection response to req PLAT_COMMS_00161 so that it is clear what the format is. Altered PLAT_COMMS_00165 to also reference the frame format for tlm rejection frames.	8.0
06	RELEASED	Introduced RFM69HCW requirements.	11.0
07	RELEASED	Introduced Mercury GS Web information and requirements (sections 4.1 and 5.8)	13.0
08	RELEASED	Refined Rel1.0_Web requirements to more carefully describe this functionality.	15.0

2 References

The following references are applicable to this document. The versions of these documents can be found in [OSSAT Baseline] revision 02.

Document Reference	Document Title	Reference in this document
KS-DOC-01073	OSSAT Baseline document revision reference	[OSSAT Baseline]
KS-DOC-01078	OSSAT Platform Software High level Requirements	[OSSAT Plat sw requirements]
KS-DOC-01077	SPRINT 1040: Flight Software Benchmarking Spec	[SPRINT1040 Flight SW Benchmark spec]
KS-DOC-01056	OSSAT Glossary	[OSSAT Glossary]
KS-DOC-01061	OSSAT Software Development Process	[OSSAT Software Dev Process]
KS-DOC-01158	OSSAT Mercury Space Segment Requirement Specification	[OSSAT Mercury SS]

3 Introduction

The Open Source Satellite (OSSAT) Programme is intending to build an open source spacecraft platform design. At the time of writing, some code has been created that demonstrates various functions of the platform computer and we need to expand upon this to enable a Telemetry and Telecommanding function (TTC).

In the longer term, OSSAT will engage with various ground station infrastructure providers to identify ground segment(s) to operate with. This will require a full communications protocol stack implementation.

In the short term, the OSSAT engineers wish to have a means to emulate a groundstation that can provide a few different functions:

- Telecommand transmission
- Telecommand acknowledgement handling
- Telemetry reception and display
- Telemetry logging
- Time synchronisation.

A draft set of platform computer requirements are available in [OSSAT Plat sw requirements]

Also, in order to downselect a processor as a core for our platform computing resource, OSSAT are running a SPRINT project in conjunction with the University of Surrey. This involves a benchmarking exercise specified in [SPRINT1040 Flight SW Benchmark spec]. Both of these requirements are driving the requirements in this document.

For reference, see the [OSSAT Glossary] to demystify terminology.

This specification should be read in conjuction with [OSSAT Mercury SS] which specifies the other half of the base TTC system (SS for Space Segment).

4 Scope

This document gives a specification of functions that Mercury GS shall fulfil.

The requirements in this specification are tagged against releases. The first release follows the needs set out in [SPRINT1040 Flight SW Benchmark spec].

Further releases follow the more general requirements documented in [OSSAT Plat sw requirements].

All code following this spec will be developed in accordance with [OSSAT Software Dev Process] as a "Type 1" development.

See [OSSAT Mercury SS] for further information on the use cases of the software and how it integrates with the wider system.

4.1 Remote collaborator use case

The below diagram gives an overview of the Mercury GS system from the perspective of a remote collaborator and a system administrator.

A remote collaborator can perform actions through a web browser to interact with a spacecraft as if they were in a mission control room.

In order to secure this link, remote collaborators are authenticated to use the system through a login screen and can then send telemetry requests and telecommands etc. From this point forward. Once authenticated, communication from the web browser to the Mercury GS Web Middleware is encrypted.

The Mercury GS Middleware acts as a bridge from the internet through to an existing Python application that performs the direct communication (either through radio or via UART) to the Spacecraft (that is running a complimentary piece of software called Mercury SS).

Since the Mercury GS Web Middleware and the Mercury GS Software both exist within the same local network, communications between them does not need to be secure.

An local administrator is capable of adding, removing are observing a list of current Remote Collaborators.

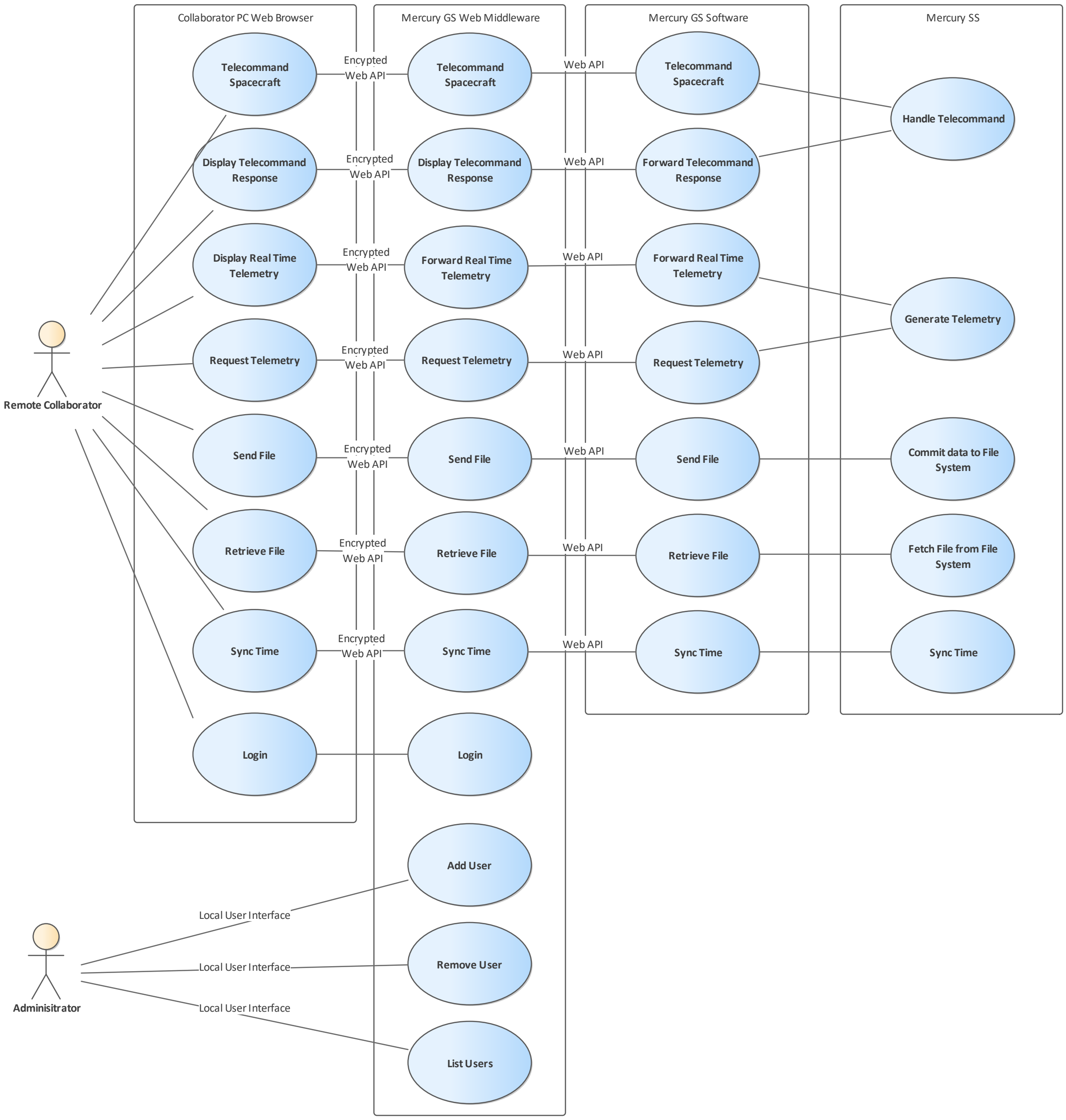


Figure 1: 4.1 Remote collaborator use case

5 Mercury GS Requirements

This section breaks down the specification into atomic requirements, tagged against a release and given a rationale. Unless otherwise stated data of multi byte fields (e.g. 32 bit integers/floating point numbers) are transmitted Most Significant Byte First (e.g. 32 bit integer "1" is transmitted as bytes of value and order 0,0,0,1). Strings are transmitted in the order of the ASCII characters from left to right (i.e. "Hello" would be transmitted 'H', 'e', 'l', 'l', 'o').
These requirements express a number of frame data types, the references of which are summarised in the following table:

DataType	Specific Number
Telecommand requests	0x01
Telecommand response	0x02
Telemetry Data	0x03
Telemetry Request	0x04

File Upload	0x05
File Download	0x06
Telemetry Request Rejection	0x07

5.1 Frame Format

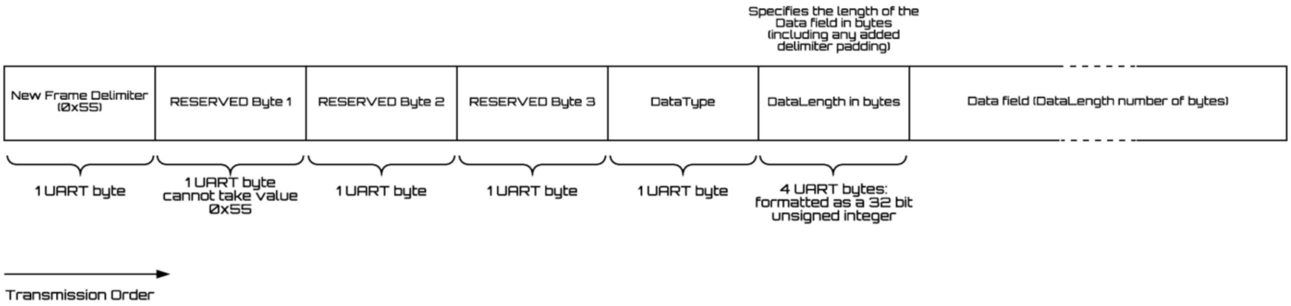
This section breaks down the format of transmissions to and from the platform computer.

PLAT_COMMS_00040

All data both transmitted and received by Mercury GS shall be formatted as follows:

Rationale: The definition includes reserved fields for future expansion (e.g. commanding multiple computers). Since the DataLength is limited to 4 bytes, the maximum frame size is approx 4 billion bytes)

Release: Rel1


Specifies the length of the Data field in bytes (including any added delimiter padding)
New Frame Delimiter (Ø x55)
RESERVED Byte 1
RESERVED Byte 2
RESERVED Byte 3
DataType
DataLength in bytes
Data field (DataLength number of bytes)
1 UART byte
1 UART byte cannot take value Øx55
1 UART byte
1 UART byte
1 UART byte
4 UART bytes: formatted as a 32 bit unsigned integer
Transmission Order

PLAT_COMMS_00045

Except where transmitted in the Test Interface page, if the data to be transmitted in any of the following fields contains the bit pattern 0x55 in any complete byte, Mercury GS shall transmit an additional 0x55 after the intended 0x55:

All RESERVED fields, DataType field, DataLength field, Data field

Rationale: This ensures that the first delimiting character can be distinguished from the rest of the data. The test interface allows for invalid data to be transmitted to the Mercury SS software, this will allow testing of invalid input.

Release: Rel1

PLAT_COMMS_00046

If Mercury GS receives a byte of value 0x55 followed by a byte other than 0x55, it shall interpret this as the start of a new data frame

Rationale: This ensures that the first delimiting character can be distinguished from the rest of the data and allows recovery should data be lost for whatever reason.

Release: Rel1

PLAT_COMMS_00047

If Mercury GS software receives two consecutive 0x55, it shall strip one of the 0x55 bytes before interpreting data further.

Rationale: 0x55 can equally appear to data fields. Therefore, this handles the stripping of delimiter padding characters part way through the data

Release: Rel1

5.1.1 UART Transmission

Communication can occur over a single UART (RS232) channel, this section gives details of the configuration and function of this method of communication.

PLAT_COMMS_00010

Mercury GS shall communicate with the platform computer over UART with the following configuration:
8 Data Bits
No parity bit
1 Stop Bit
No Hardware Flow Control

Rationale: This is a standard configuration, no parity bit is used since the data is travelling over a short, reliable link and this is a lab only interface.

Release: Rel1

PLAT_COMMS_00020

Mercury GS shall communicate with the platform computer over UART at a baud rate selectable from the GUI configuration options as follows:
9600
38400
57600
115200
1280000
2560000

Rationale: It is useful to be able to select a data rate to aid software performance tests.

Release: Rel1

PLAT_COMMS_00025

Mercury GS shall communicate with the platform computer over UART at a baud rate detected by analysis of the data transmitted by Mercury SS.

Rationale: This is a nice-to-have and is therfore not required on the first release.

Release: Rel3

5.1.2 RFM69 Transmission

Communication can occur over a radio transceiver (RFM69HCW) channel, this section gives details of the configuration and function of this method of communication.

PLAT_COMMS_00030

Mercury GS shall communicate with the platform computer over a RFM69HCW radio transceiver with the following configuration:
4 bytes Preamble
2 bytes Sync Word (2d, d4)
2 bytes CRC CCITT
Frequency 434.0
Power +13dBm
GFSK 250Kb/s
250KHz Deviation
Bitwhitening
No AES

Rationale: At the moment this is a fixed configuration. We may want to expand upon this and implement configurability at a later date.

Release: Rel2.1_RF

PLAT_COMMS_00031

Mercury GS shall communicate with the RFM69HCW radio transceiver over SPI with the following configuration:
8 bit Data Size
CPOL = 0
CPHA = 0
Motorola

Rationale: At the moment this is a fixed configuration. We may want to expand upon this and implement configurability at a later date.

Release: Rel2.1_RF

5.2 Telecommanding

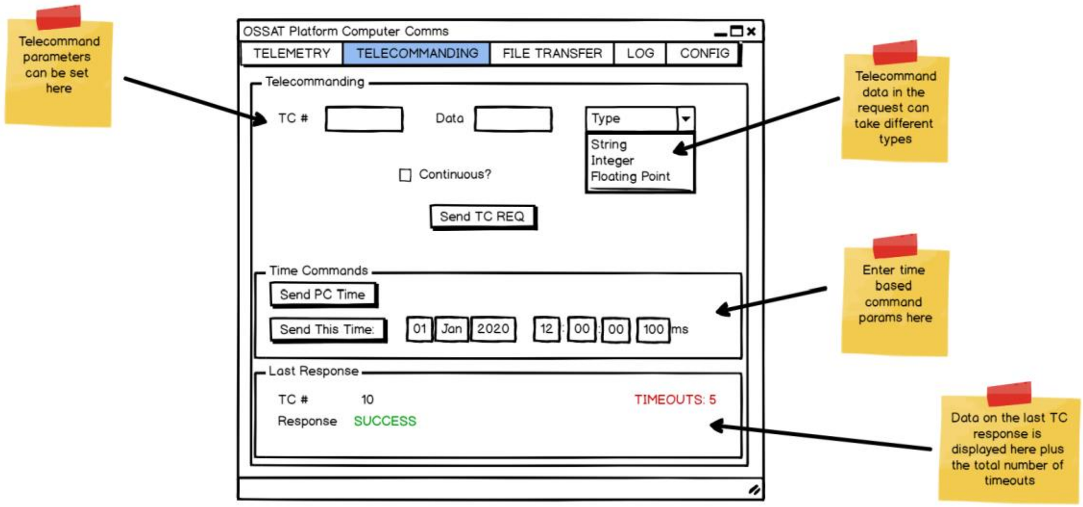
The following requirements apply to telecommanding, mimicking the commanding from the ground segment to the space segment. NOTE: telecommand requests are to be implemented as a 1 off transmission, automatic retries are not required.

PLAT_COMMS_00048

Mercury GS shall incorporate a GUI that is represented using the following wireframe.

Rationale: This guides the developer concerning the look and feel.

Release: Rel1



TC # 10

Response SUCCESS

TIMEOUTS: 5

Telecommand parameters can be set here

Telecommand data in the request can take different types

Enter time based command params here

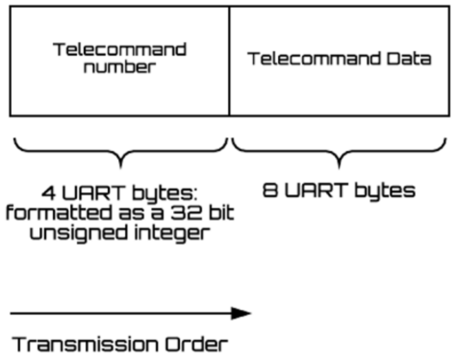
Data on the last TC response is displayed here plus the total number of timeouts

PLAT_COMMS_00050

Telecommand requests shall be sent by Mercury GS with a Data field formatted as specified below and where DataType is set to 0x01. (NOTE: this is the data field as part of the frame defined by PLAT_COMMS_00040)

Rationale: This gives a large telecommand address space and ample scope for telecommand data.

Release: Rel1

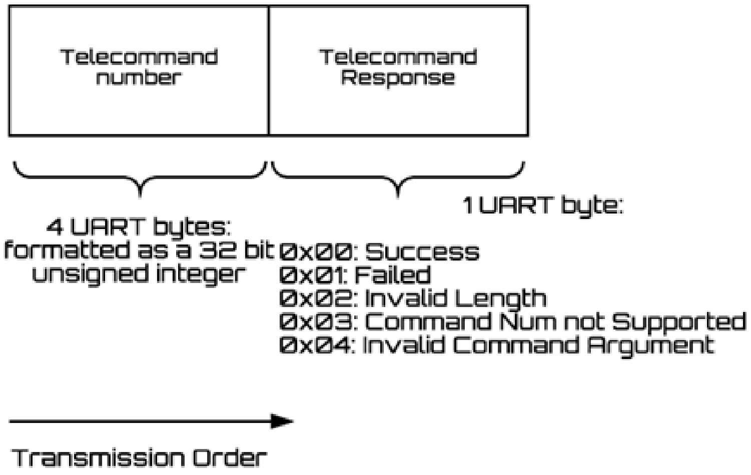


PLAT_COMMS_00055

Mercury GS shall interpret a frame where DataType is set to 0x02 as a telecommand response which has a data field formatted as shown below. (NOTE: this is the data field as part of the frame defined by PLAT_COMMS_00040)

Rationale: The telecommand response needs to include the telecommand number otherwise the program cannot associate commands with responses.

Release: Rel1

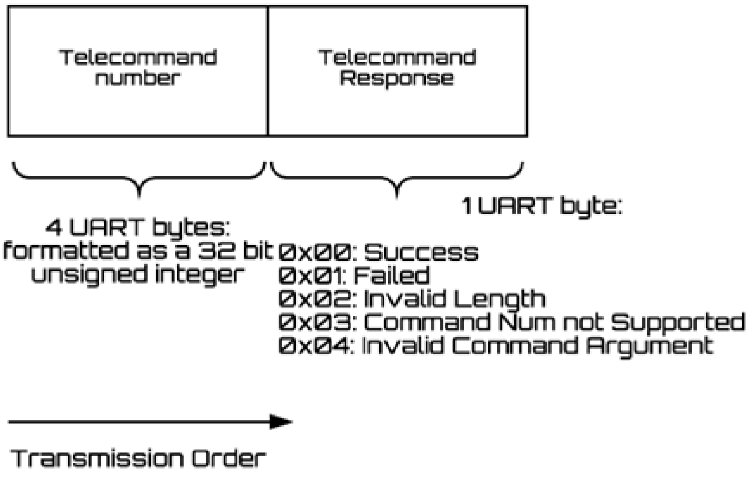

Telecommand number
Telecommand Response
4 UART bytes: formatted as a 32 bit unsigned integer
1 UART byte:
0x00: Success
0x01: Failed
0x02: Invalid Length
0x03: Command Num not Supported
0x04: Invalid Command Argument
Transmission Order

PLAT_COMMS_00056

The last received telecommand response shall be displayed on the GUI.

Rationale: The user needs to understand how the command has been interpreted by the target.

Release: Rel1


Telecommand number
Telecommand Response
4 UART bytes: formatted as a 32 bit unsigned integer
1 UART byte:
0x00: Success
0x01: Failed
0x02: Invalid Length
0x03: Command Num not Supported
0x04: Invalid Command Argument
Transmission Order

PLAT_COMMS_00060

Mercury GS shall incorporate a GUI dialogue where the user can enter a telecommand number and either:
a string (max 8 chars)
an integer (64 bit signed)
a floating point value (double precision)
as telecommand data.

Rationale: This gives the user the ability to specify the content of a telecommand to be transmitted.

Release: Rel1

PLAT_COMMS_00070

The GUI for Mercury GS shall support a checkbox to select the continuous transmission of the telecommand at the rate specified by a TC rate field in the GUI configuration.

Rationale: This gives us the ability to stress the platform computer with a high data demand.

Release: Rel1

PLAT_COMMS_00080

The GUI Mercury GS shall support a button to trigger the one-off transmission of the telecommand.

Rationale: This gives us the ability transmit a single ad-hoc command to the platform computer

Release: Rel1

PLAT_COMMS_00090

Mercury GS shall monitor for the reception of a telecommand acknowledgement that corresponds with telecommands transmitted and shall display the number of timeouts that have exceeded <TC/TLM timeout>

Rationale: This gives us the ability to stress test the platform computer identifying when data is lost.

Release: Rel1

PLAT_COMMS_00100

Mercury GS shall allow the user to alter the <TC/TLM timeout> via the GUI configuration options.

Rationale: This gives us the ability to adjust the timeout should the platform computer become sluggish

Release: Rel1

5.3 Telemetry

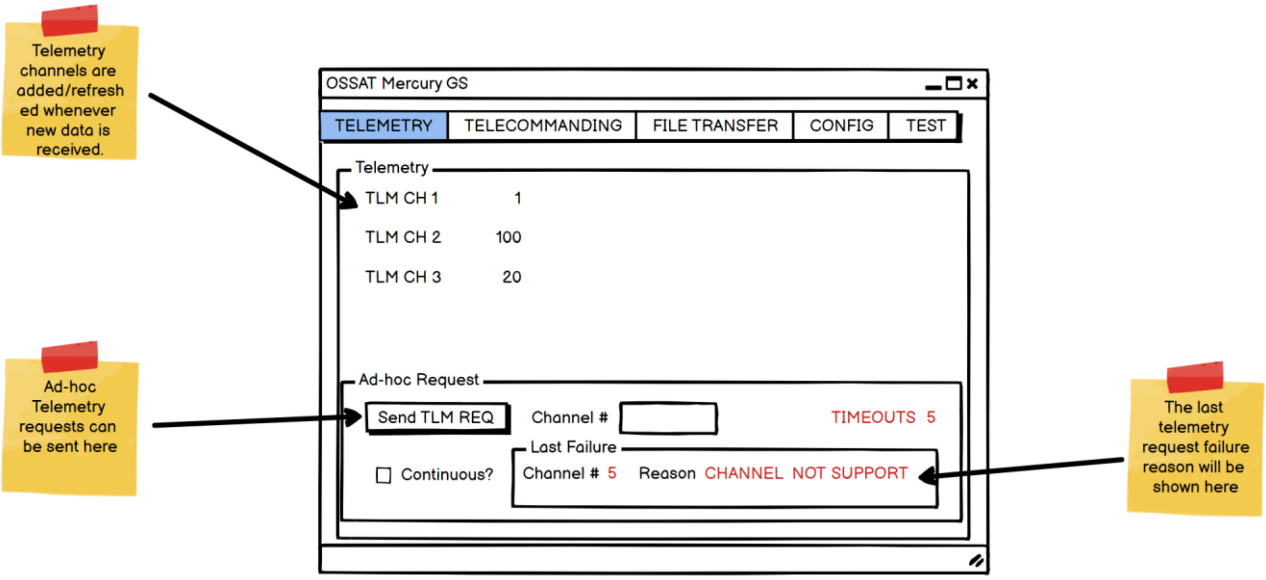
The following requirements refer to the display, requesting and storage of telemetry. Telemetry data will be broadcast from the target continuously. However, provision is given for the target to accept a request for a particular telemetry request. NOTE: telemetry requests are to be implemented as a 1 off transmission, automatic retries are not required.

PLAT_COMMS_00105

Mercury GS shall provide a GUI with respect to telemetry as shown below.

Rationale: This gives the programmer a better understanding of what is required.

Release: Rel1



OSSAT Mercury GS

TELEMETRY | TELECOMMANDING | FILE TRANSFER | CONFIG | TEST

Telemetry

TLM CH 1	1
TLM CH 2	100
TLM CH 3	20

Ad-hoc Request

Send TLM REQ   Channel #   TIMEOUTS 5

☐ Continuous?

Last Failure

Channel # 5   Reason CHANNEL NOT SUPPORT

Telemetry channels are added/refreshed whenever new data is received.

Ad-hoc Telemetry requests can be sent here

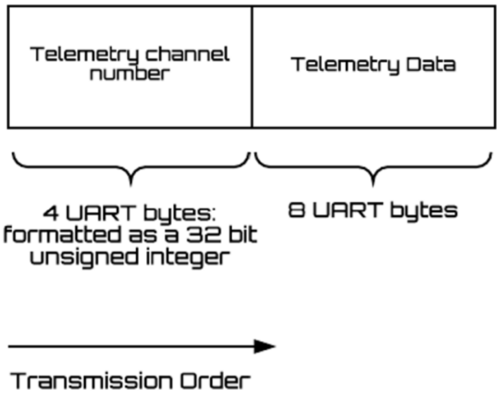
The last telemetry request failure reason will be shown here

PLAT_COMMS_00110

Mercury GS software shall display the latest received telemetry channel and data on screen as a 64 bit integer whenever a telemetry response (DataType 0x03) is received from [OSSAT Mercury GS] where the Data field is formatted as follows (NOTE: this is the data field as part of the frame defined by PLAT_COMMS_00040):

Rationale: This gives the platform software the ability to test the real time telemetry feed from the spacecraft. NOTE: all data is interpreted as double integer for Rel1, other requirements are relevant for Rel2 and beyond.

Release: Rel1



PLAT_COMMS_00120

Mercury SS shall interpret telemetry data by channel and display the data in accordance with these data types:
- Floating point number
- String

- Integer
- Enumeration

On the basis of a configuration file that can be selected in the GUI configuration.

Rationale: This gives greater flexibility on the way data is displayed to the user. NOTE: there is purposeful contradiction here with the previous requirement because this is a Release 2 function to make the tool more powerful.

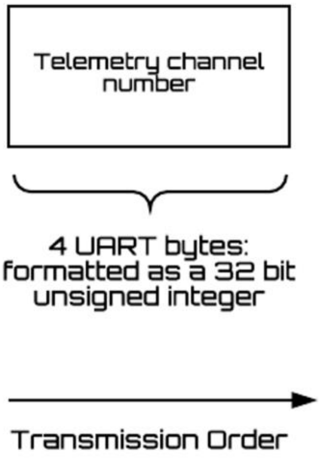
Release: Rel2.3

PLAT_COMMS_00130

Mercury GS shall support a dialogue to transmit a telemetry request (DataType 0x04) with data formatted into the data field as follows (NOTE: this is the data field as part of the frame defined by PLAT_COMMS_00040):

Rationale: This means telemetry can be requested ad-hoc. This is not needed imminently since telecommanding can be used to load the computer with requests.

Release: Rel1



PLAT_COMMS_00150

Following PLAT_COMMS_00130, Mercury GS shall support a button to transmit a single telemetry request

Rationale: This means telemetry can be requested ad-hoc. This is not needed imminently since telecommanding can be used to load the computer with requests.

Release: Rel1

PLAT_COMMS_00160

Mercury GS shall monitor for the reception of a telemetry response that corresponds with telemetry request transmitted and shall display the number of timeouts that have exceeded <TC/TLM timeout>

Rationale: This means telemetry can be requested ad-hoc. This is not needed imminently since telecommanding can be used to load the computer with requests.

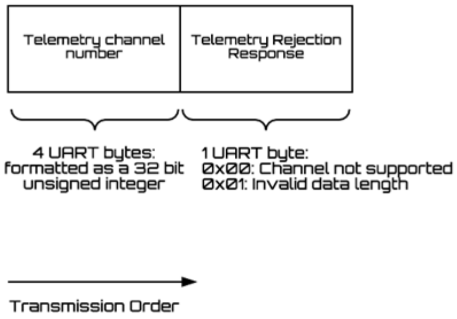
Release: Rel1

PLAT_COMMS_00161

Mercury GS shall monitor for the reception of a telemetry request rejection response (formatted as shown below) that corresponds with telemetry requests transmitted and shall display the number of timeouts that have exceeded <TC/TLM timeout>. (NOTE: the below diagram shows the data field as part of the frame defined by PLAT_COMMS_00040).

Rationale: This means telemetry can be requested ad-hoc. This is not needed imminently since telecommanding can be used to load the computer with requests.

Release: Rel2.1


Telemetry channel number
Telemetry Rejection Response
4 UART bytes: formatted as a 32 bit unsigned integer
1 UART byte: Ø×ØØ: Channel not supported Ø×Ø1: Invalid data length
Transmission Order

PLAT_COMMS_00165

Mercury GS shall display the relevant channel and failure reason on the GUI when it receives a telemetry request rejection frame (DataType 0x07) from [OSSAT Mercury SS] in response to a telemetry request, where the Data field is formatted as shown in PLAT_COMMS_00161.

Rationale: It really helps operators to know why a particular action failed.

Release: Rel2.1

PLAT_COMMS_00170

Mercury GS shall support a checkbox to select the continuous transmission of the telemetry requests at the rate specified by a TLM rate field in the GUI configuration.

Rationale: This allows scope for a telemetry loading test, similar to the telecommanding equivalent.

Release: Rel1

PLAT_COMMS_00175

Mercury GS shall optionally log all incoming telemetry against time into a file.

Rationale: This can then be used to visualise data through e.g. excel, Matlab. a standard data format is required that is widely supported (e.g. csv). The GUI options configuration options can include a filename and checkbox to enable/disable logging

Release: Rel3

PLAT_COMMS_00176

The GUI configuration options shall include a filename and checkbox to enable/disable logging.

Rationale: This can be used to control where the data is logged.

Release: Rel3

5.4 Time Handling

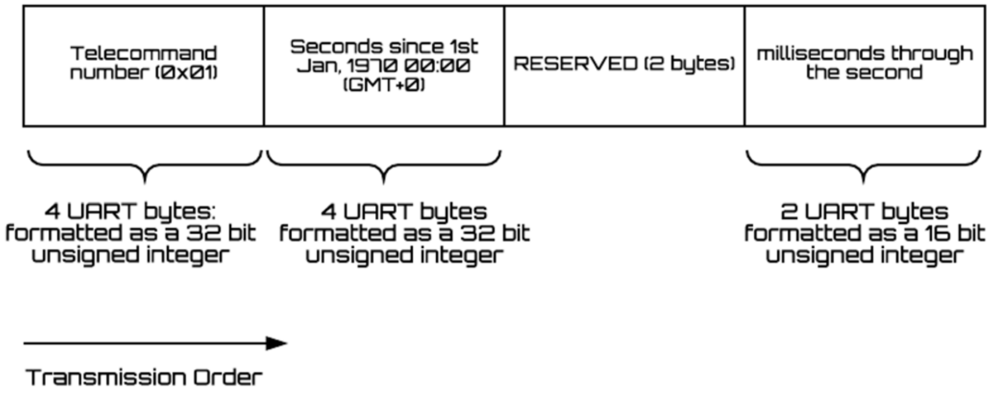
The system as a whole requires time synchronisation. The spacecraft uses either a GPS receiver or the ground as a time source. This section documents features that will allow users to test the spacecrafts ability to synchronise with a ground time source (see [OSSAT Plat sw requirements] section "Time Handling").

PLAT_COMMS_00180

Mercury GS shall support a specific telecommand to set the time on the platform computer in accordance with the time set on the PC running the software. The format of the command in the DataField being as follows(NOTE: this is the data field as part of the frame defined by PLAT_COMMS_00040):

Rationale: This can simulate the time sync between ground and spacecraft. This is a release 3 command because the Command Line Interface can be used in the short term to set the time.

Release: Rel2.2


Telecommand number (0x01)
Seconds since 1st Jan, 1970 00:00 (GMT+0)
RESERVED (2 bytes)
milliseconds through the second
4 UART bytes: formatted as a 32 bit unsigned integer
4 UART bytes formatted as a 32 bit unsigned integer
2 UART bytes formatted as a 16 bit unsigned integer
Transmission Order

PLAT_COMMS_00190

Mercury GS shall support a specific telecommand to set the time on the platform computer in accordance with a time set on some GUI fields (as shown in PLAT_COMMS_0048). The format of the command as specified in PLAT_COMMS_00180.

Rationale: This allows us to test the synchronisation from a known erroneous starting point.

Release: Rel2.2

5.5 File Handling

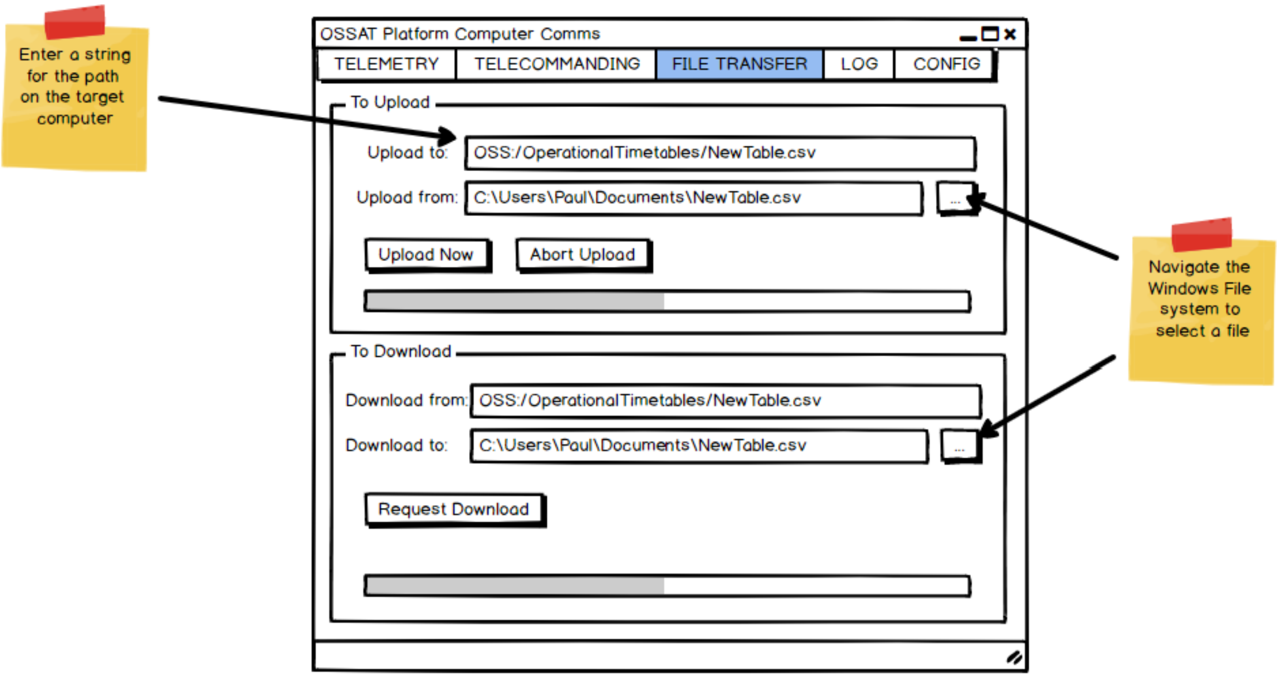
This section covers the handling of file upload/download. NOTE: these requirements are set as release 3 because the Command Line Interface can be used in the short term to perform file upload/download.

PLAT_COMMS_00205

Mercury GS shall incorporate a window similar to the following wireframe.

Rationale: This gives the developer an clear understanding of what is required.

Release: Rel1



Enter a string for the path on the target computer

OSSAT Platform Computer Comms

To Upload

Upload to: OSS:/OperationalTimetables/NewTable.csv

Upload from: C:\Users\Paul\Documents\NewTable.csv

To Download

Download from: OSS:/OperationalTimetables/NewTable.csv

Download to: C:\Users\Paul\Documents\NewTable.csv

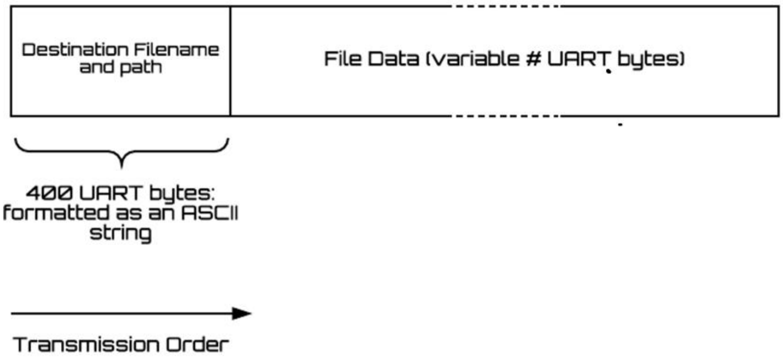
Navigate the Windows File system to select a file

PLAT_COMMS_00210

Mercury GS shall support a file upload request (DataType 0x05) that takes parameters from the GUI including the source file path and the destination file path and name. The format of the request in the Data field is as follows (NOTE: Destination Filename and Path is a NULL terminated ASCII string. This is the data field as part of the frame defined by PLAT_COMMS_00040):

Rationale: This allows us to upload files to specific destination directory.

Release: Rel3



Destination Filename and path	File Data (variable # UART bytes)

400 UART bytes: formatted as an ASCII string

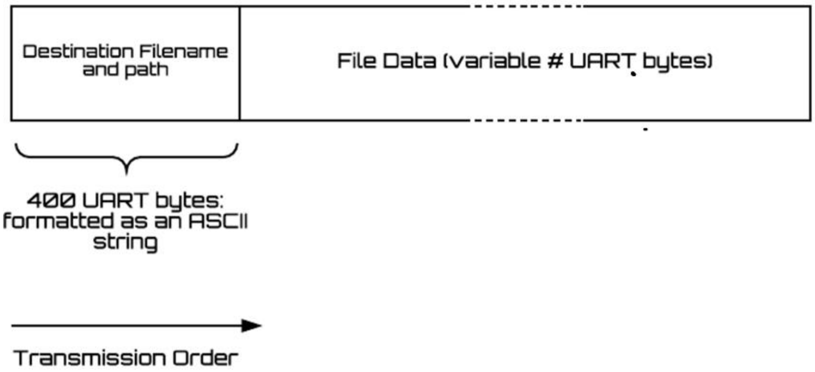
Transmission Order

PLAT_COMMS_00215

Mercury GS shall support an abort button to stop transmission of a file upload request.

Rationale: This is a simple measure to allow the user to stop an upload mid-upload if it was triggered erroneously.

Release: Rel3

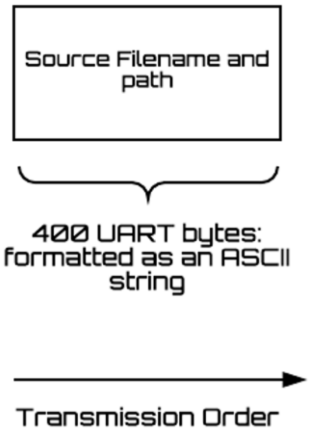

Destination Filename and path
File Data (variable # UART bytes)
400 UART bytes: formatted as an ASCII string
Transmission Order

PLAT_COMMS_00220

Mercury GS shall support a file download request (DataType 0x06) that takes parameters from the GUI including the source file path and the destination file path and name. The format of the request in the Data field is as follows(NOTE: Source filename and path is a NULL terminated ASCII string. This is the data field as part of the frame defined by PLAT_COMMS_00040):

Rationale: This allows us to download files to specific destination directory.

Release: Rel3

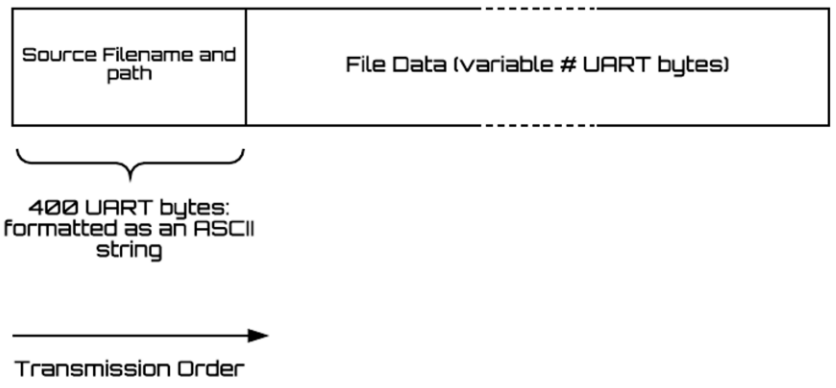


PLAT_COMMS_00230

Mercury GS shall commit data to a file on the PC following reception of a download file that is received in the format within the Data field as follows:

Rationale: This allows us to download files to specific destination directory.

Release: Rel3


Source Filename and path
File Data (variable # UART bytes)
400 UART bytes: formatted as an ASCII string
Transmission Order

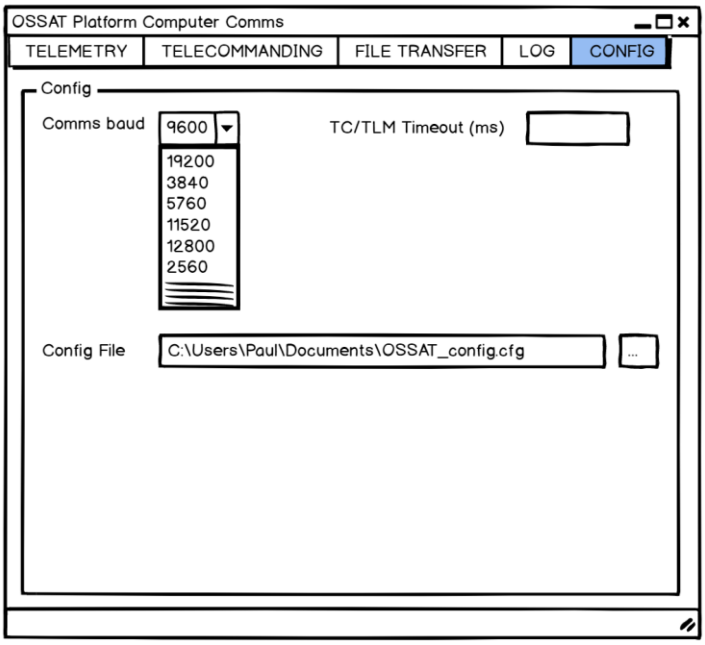
5.6 Configuration

PLAT_COMMS_00240

Mercury GS shall support a GUI as illustrated below in order to configure the application.

Rationale: There are a few GUI configuration options that need to be supported.

Release: Rel1



Comms baud 9600

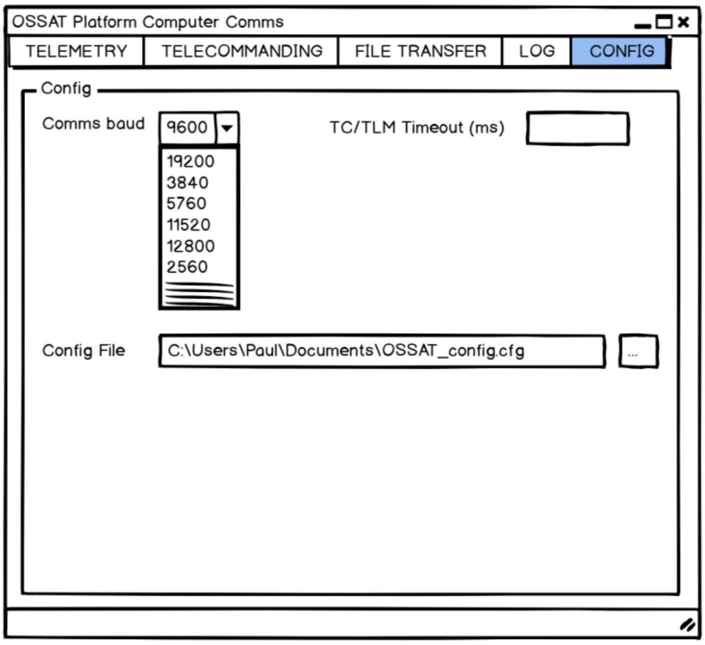
Config File C:\Users\Paul\Documents\OSSAT_config.cfg

PLAT_COMMS_00241

Mercury GS config tab shall support a drop down menu to change the com port that is used to talk to the spacecraft.

Rationale: The com port used will not always be the same on different systems.

Release: Rel1



Comms baud: 9600

19200
3840
5760
11520
12800
2560

TC/TLM Timeout (ms)

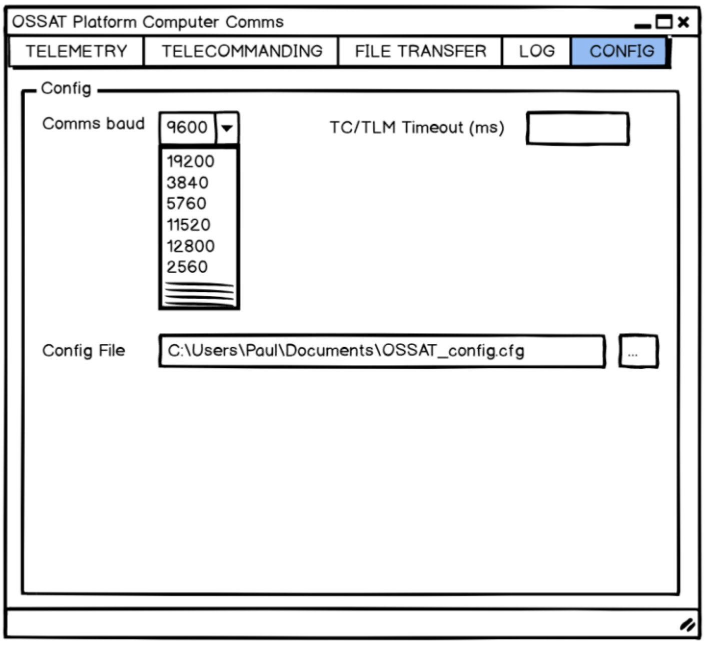
Config File: C:\Users\Paul\Documents\OSSAT_config.cfg

PLAT_COMMS_00242

Mercury GS config tab shall support a field to adjust the rate of telemetry and telecommand messages sent when continuous transmission is selected. This will be measured in messages per second.

Rationale: Adjusting the rate of continuous transmission is required for stress testing both Mercury GS and the spacecraft.

Release: Rel1



Comms baud: 9600

19200
3840
5760
11520
12800
2560

TC/TLM Timeout (ms)

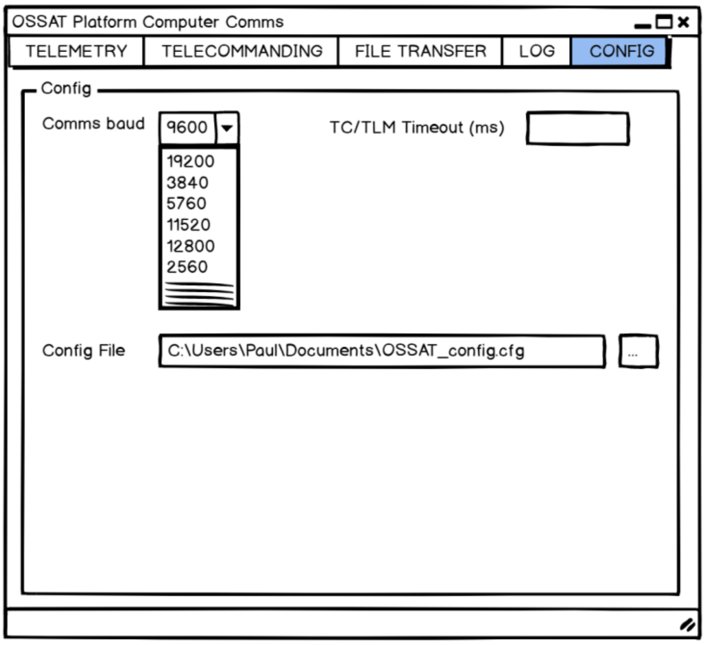
Config File: C:\Users\Paul\Documents\OSSAT_config.cfg

PLAT_COMMS_00243

Mercury GS config tab shall support a drop down menu to select either COM or RFM69 for low level transmisssion and reception.

Rationale: The integration of a radio module will require the ability to choose whether to communicate with the spacecraft via different methods.

Release: Rel2.1_RF



Comms baud: 9600

19200
3840
5760
11520
12800
2560

TC/TLM Timeout (ms)

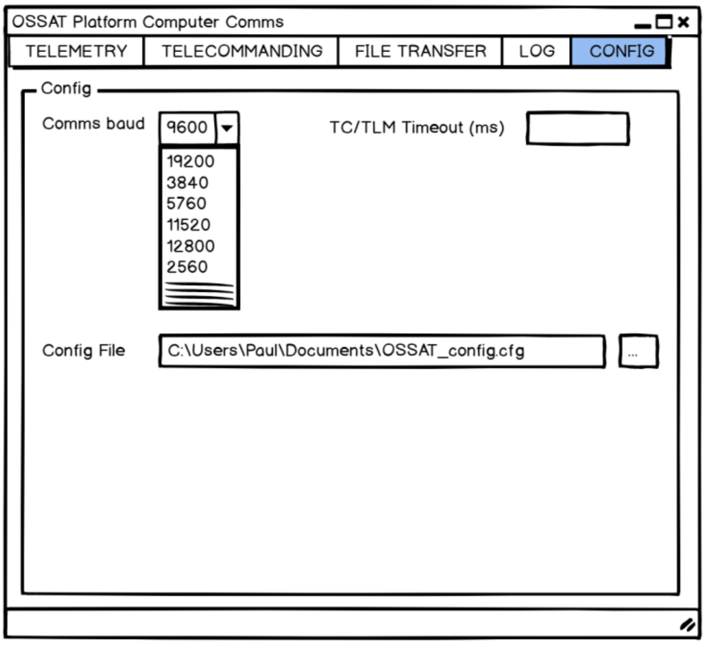
Config File: C:\Users\Paul\Documents\OSSAT_config.cfg

PLAT_COMMS_00244

If "RFM69" is selected in the drop down menu on the config tab, the baud rate and com port fields shall be disabled.

Rationale: When using a radio transceiver to communicate with the spacecraft, the baud rate and com port are irrelevant.

Release: Rel2.1_RF


OSSAT Platform Computer Comms

Comms baud: 9600

TC/TLM Timeout (ms):

Config File: C:\Users\Paul\Documents\OSSAT_config.cfg

5.7 Test Interface

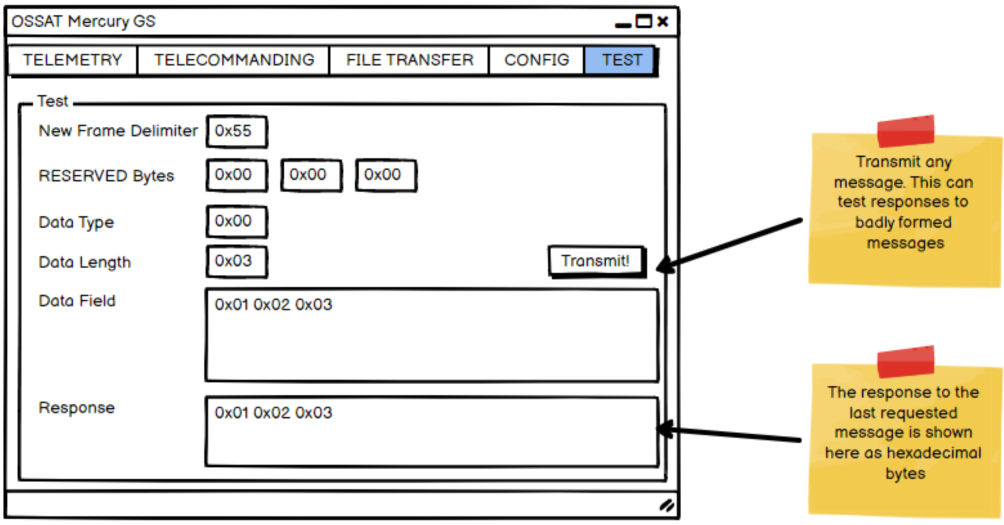
This section covers requirements relating to a "test" tab that can be used to send valid and invalid data to Mercury SS in order to test exception handling within Mercury SS.

PLAT_COMMS_00250

Mercury GS shall support a GUI as illustrated below in order to support the testing of invalid data to [OSSAT Mercury SS].

Rationale: In order to test that [OSSAT Mercury SS] deals with invalid data input correctly, this interface is required.

Release: Rel1



Test

New Frame Delimiter: 0x55

RESERVED Bytes: 0x00 0x00 0x00

Data Type: 0x00

Data Length: 0x03

Data Field: 0x01 0x02 0x03

Response: 0x01 0x02 0x03

Transmit any message. This can test responses to badly formed messages

The response to the last requested message is shown here as hexadecimal bytes

PLAT_COMMS_00260

When the transmit button is clicked, the Mercury GS software shall transmit the data as specified in the fields on the GUI page, allowing for discrepancies between the Data Length and number of bytes specified in the Data Field.

Rationale: This allows for testing of [OSSAT Mercury SS] input validation.

Release: Rel1

PLAT_COMMS_00270

Once the transmit button has been clicked, Mercury GS shall sample the incoming UART data from [OSSAT Mercury SS] for one second and then display the incoming data as hexadecimal bytes in the "Response" field.

Rationale: The test interface allows for the transmission of invalid data to [OSSAT Mercury SS] in order to test its validation code. In order to know how [OSSAT Mercury SS] is responding to the incoming data, Mercury GS cannot assume that data will be formatted in accordance with section 5.1. Therefore, Mercury GS needs to simply sample and display the data from [OSSAT Mercury SS] as raw bytes, the one second ensures that there is no assumption concerning the end of a frame.

Release: Rel1

5.8 Web Interface

This section provides requirements related to the Mercury GS Web, this includes requirements on the Mercury GS Software itself, the Mercury GS Web Middleware and the web page available to a collaborator (see section 4.1)

PLAT_COMMS_00280

The Mercury GS Collaborator web page shall include a means for users to login via an email address and password.

Rationale: This ensures some level of authentication.

Release: TBD

PLAT_COMMS_00290

The Mercury GS collaborator website shall include displays as set out in this document, including:
- Telemetry
- Telecommand
- File Transfer
- Configuration
- Test

Rationale: This will allow collaborators the ability to interface with the satellite similar to groundstation operators.

Release: TBD

PLAT_COMMS_00300

The Mercury GS Web Middleware shall support communications with remote browsers using an encrypted web API.

Rationale: This is to ensure a secure link.

Release: TBD

PLAT_COMMS_00310

The Mercury GS Web Middleware shall forward telecommands from the Mercury GS collaborator website to the Mercury GS software using the "Transmit Telecommand" web schema specified below:

Rationale: Telecommands need to transfer to the physical spacecraft from the web.

Release: Rel1.0_Web

Transmit Telecommand

The Mercury GS Web Middleware to Mercury GS Software Telecommand URL Schema:

Property	Type	Description
telecommandNumber	integer	A telecommand number which is allocated as e.g. set spacecraft mode.
telecommandDataType	string	This designates the type of the data in the "telecommandData" property. Can be "String", "Integer" or "Float"
telecommandData	string OR integer OR float	this can be a string of 8 characters, an integer or a floating point number.

PLAT_COMMS_00311

The Mercury GS Web Middleware shall forward a command to enable or disable the continuous transmission of a telecommand using the web schema specified below:

Rationale: Telecommands can be configured to transmit continuously.

Release: Rel1.0_Web

Enable/Disable Continuous Telecommand

Property	Type	Description
continuous	boolean	If set to true: this designates that Mercury GS software will continuously send the subsequently sent telecommands. If set to false, this designates that Mercury GS will stop sending continuously (if it is not currently sending continuously, there is no affect). The rate that the telecommand is sent is defined by the config command.

PLAT_COMMS_00312

The Mercury GS Web Middleware shall forward a command to set the rate of transmission of both continuously transmitted telecommands and telemetry requests using the web schema as specified below:

Rationale: Telecommands can be configured to transmit continuously, the rate needs to be set.

Release: Rel1.0_Web

Configuration Command

Property	Type	Description
continuousTransmissionRate	integer	Optional: this specifies the rate at which continuously transmitted telecommands and telemetry requests will be transmitted. Default is: 1 (Hz)

Note that this schema will be expanded upon following Rel1.0_Web.

PLAT_COMMS_00320

The Mercury GS Web Middleware shall forward telemetry requests from the Mercury GS collaborator website to the Mercury GS software

Rationale: Telemetry Requests need to transfer to the physical spacecraft from the web.

Release: Rel1.1_Web

www.opensourcesatellite.org

in linkedin.com/company/open-source-satellite

@SatelliteOpen

info@opensourcesatellite.org