Foresail-1p

Spacecraft Space/Ground Interface Control Document

PUBLIC DOCUMENT

Aalto University

School of Electrical Engineering

Authors

Name	Worked with parts
Petri Niemelä	Whole Document
Markus Hiltunen	Skylink
Baris Dinc	Radio Amateur Repeater
Topi Räty	Updates for FS1p
Klaus Kivirikko	Updates for FS1p

Table of Contents

1 General

1.1 Scope

This document describes the complete interface description between the Foresail-1p satellite and the ground segment. In context of this document, the ground segment term covers any ground based satellite receiving station capable of receiving at the 70 cm radio amateur band. This document has been written to be complete, without referring to the project's internal documentation and targeted for the radio amateur and satellite tracker community around the globe.

1.2 Reference documents

RD-01	FS1 UHF Skylink Protocol specification
RD-02	ECSS-E-ST-70-41C Packet Utilization Standard (PUS-C)
RD-03	AX.25 & APRS specification

1.3 Document-specific Abbreviations and Acronyms

APR	Amateur Packet Repeater
GMSK	Gaussian Minimum Shift Keying
HAM	Radio Amateur
IARU	International Amateur Radio Union
MAC	Medium Access Control
TDD	Time Division Duplex
UHF	Ultra High Frequency (generally 300 - 3000 MHz, 430 MHz in this document) Used to refer to UHF-band Telemetry, Tracking and Commanding subsystem.
BATT	Battery Subsytem
OBC	On-board Computer
EPS	Electrical Power System (consists PCDU and BATT)
PB	Plasma Brake
PCDU	Power Conditioning and Delivery Unit
PATE	Particle Telescope
ST	Telecommand/Telemetry Service Type following the PUS
SST	Telecommand/Telemetry Service Sub Type
TM(A,B)	Telemetry Service Type A and Sub Type B.

2 Foresail-1p Communication System

The Foresail-1p satellite is a 3U cubesat which operates at the 70 cm (436 - 438MHz) radio amateur band. The mission is a successor to Foresail-1 and utilizes the same communication scheme, but has an updated telemetry structure. The satellite operates at a 437.125 MHz frequency with +/- 10kHz of Doppler shift and covers a 20 kHz bandwidth. The frequency has been coordinated with the Finnish radio amateur community and the International Amateur Radio Union (IARU application), and has been accepted by the International Telecommunication Union (ITU). The same operating frequency is used for both satellite telecommanding and telemetry, and it also implements amateur radio repeater functionalities.

The satellite’s general transmission parameters are:

- Center frequency: 437.125 MHz with +/- 10kHz Doppler shift
- Bandwidth: 20 kHz
- Modulation: GMSK
- Data/Symbol rate: 4800, 9600 (nominal), 19200, or 38400 bauds
- Output power 1.3 Watts (32 dBm)
- Polarization: Circular polarization
    - Handiness of the circular polarization depends on the satellite attitude.
    - With any linear polarization the satellite can be received with 3dB attenuation and minimum fading.
- Framing: Skylink PHY (GOMSpace Mode 5 Compatible) [RD-01]
    - Preamble: 6 x 0xAA (5 ms @ 9600 bauds)
    - Sync word: 0x1A 0xCF 0xFC 0x1D
    - Length: 24-bit long Golay-24 coded field
        - Lower 8 bits codes the payload length. Rest of the bits are zeros.
    - Payload/RS codeword: max 223 bytes
    - RS Parity bytes: 32 bytes long Reed Solomon. CCSDS RS(255,223)
- Windowed Time Division Duplex (TDD)
    - Window lengths will vary based on transfer demand.
- The satellite can be expected to be active every 60 seconds on average broadcasting a multiframe beacon. Regularly transmitted beacon frames are:
    - OBC Housekeeping TM(3,2) on VC1
    - EPS Housekeeping TM(3,3) on VC1
    - ADCS Housekeeping TM(3,4) on VC1
    - UHF Housekeeping TM(3,5) on VC1
    - Deployment Housekeeping TM(3, 6) on VC1:
        - Transmitted only for a period after the launch.
    - Radio amateur packet repeater beacon messages on VC3.

2.1 Ground Segment

The primary ground station used for commanding the Foresail-1p satellite is located in Otaniemi, Espoo, Finland at Aalto University’s premises. The satellite is usually commanded on passes over Finland and the satellite’s two-way communication can be generally received all over Europe. The mission has been designed to support 3rd party ground stations and Foresail-1p also supports a downlink-only style file transfer mode whose transmissions can be heard over mission 3rd party ground stations. The feature is included to increase downlink potential via collaborating radioamateur networks.

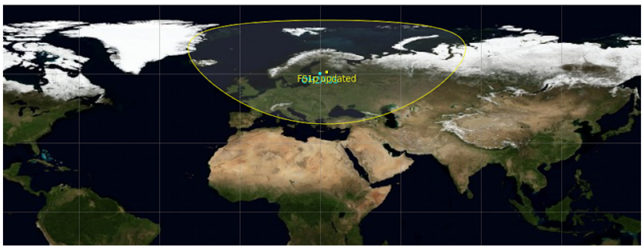

Figure 1: Coverage of the Otaniemi ground station to 500 km Low Earth Orbit.

The Foresail-1p mission control software can utilize incoming telemetry frames from various sources, such as SiDS databases. Received telemetry frames from the HAM radio community are also greatly appreciated.

2.2 Skylink Protocol

The Foresail-1p satellite uses the Skylink protocol for its communication. The protocol was designed for the mission and is designed to push forward from the AX.25-era. The detailed protocol specification can be found from RD-01 and the implementation is freely available at: https://github.com/aaltosatellite/skylink

Briefly describing, the Skylink protocol contains a physical layer specification which closely follows the widely adopted “industry standard”, which has evolved during the last years, with a transport layer protocol on top of it.

The transport layer protocol implements many features useful for implementing a small satellite commanding link. The general specifications are:

- Four logical virtual channels for mission specific purposes
- Windowed Time Division Duplexing (TDD)
- Reliable data transfer using automatic retransmission (ARQ)
- Uplink and downlink data authentication

The Skylink’s simplified frame format is illustrated in Figure 2.

Protocol Identifier 8-bits	Satellite Identifier 3-7 x 8-bits	Flags (inc. VC) 8-bits	Frame Sequence 16-bits	Extension Length 8-bits	Extension Header N x 8-bits	Payload N x 8-bits	Authentication 4 x 8-bits	FEC 32 x 8-bits

Figure 2: Simplified Skylink frame structure

- Protocol Identifier and identifier length (1 byte)
    - 5 bits version and 3 bits identifier length. Possible values 0x63, 0x64, 0x65, 0x66 and 0x67, where the second character is essentially the identity length.
- Satellite Identifier: (3-7 bytes)
    - From the satellite: 0x4F 0x48 0x32 0x46 0x31 0x53 (‘`OH2F1S`’ in ASCII)
- Flags: (8-bits)
    - Reserved 1 bits (MSB)
    - Sequence control (2 bits)
        - Unused.
    - Frame CRCed
        - If set, skylink will check the frame CRC.
        - Important to note that there is a bug onboard in the CRC decoding due to which frames with this flag shouldn’t be sent to the satellite. However frames with this flag will be sent by the satellite in VC3 (Repeater). There are no known issues in the encoding side however.
    - HAS_AUTHENTICATION
        - If set, the last 4 bytes in the frame are authentication code.
        - The satellite uplink and downlink require authentication for VC0 to VC2. The VC3 can be without authentication and the downlink is not authenticated.
    - ARQ on
        - Informs if the frame was sent while automatic retransmission was enabled.

- Virtual channel: (2-bits)
  - Virtual channel index to differentiate the logical channels.
  - The VC index also determines the protocol used inside the payload field!
- The virtual channels for Foresail-1p are listed in Table 1.

- Extension Header length (8-bits)
  - The length of the extension header field immediately followed this field.
  - Each frame will by default have at least a 5 byte MAC extension header.
- Frame Sequence counter (16-bits)
- Extension header: (N x 8-bits)
  - Header section for varying header and control data.
  - The content of the field can be ignored by the telemetry receivers.
  - The details for decoding the field can be found from RD-01.
- Payload: (N x 8-bits) (max 181 bytes)
  - The actual telemetry data.
- Authentication: (4 x 8-bits)
  - Field used to authenticate the source of the frame and only present if the HAS_AUTHENTICATION flag in the header is set.
  - Used only by the Foresail mission control software. The field can be ignored by other telemetry receivers.
- FEC: (32 x 8-bits)
  - Reed Solomon error correction code as in RS(255,223) defined by CCSDS. (TODO REF)

The Skylink protocol specification has been written to be generic and suitable for any mission. As a general description of the protocol and how the channels are used, please see the Table below.

Table 1: FORESAIL-1 virtual channels

Virtual Channel #	Name	Description
0	Commanding	• Used for satellite telecommanding and telemetry/acknowledgment reception. • Conveys CCSDS/ECSS PUS frames. • Can be configured for reliability. • Authenticated channel.
1	Mass data transfer	• Used for mass data transfer such as file transfer. • Conveys CCSDS/ECSS PUS frames. • Can be configured for reliability. • Authenticated channel
2	Service channel	• Can be configured for reliability. • Custom control protocol for the radio and electrical power system. • Authenticated channel.
3	HAM repeater	• Used for HAM repeater functionalities. • Conveys AX.25 UI-frames. • Unreliable transfer only.

		• Non-authenticated channel.

2.3 ECSS PUS Telemetry format

The satellite's main telemetry data is generated by the On-Board Computer (OBC) and conveyed inside the Skylink data transfer frames on virtual channels 0 and 1. The telemetry frame format follows the ECSS Packet Utilization Standard (PUS-C) with heavy tailoring [RD-02].

The simplified PUS frame format structure is illustrated in Figure 3. Standard PUS frame formats are used for service numbering and only service type 1: Telecommand verification follows exactly the standard. Each telemetry (and telecommand) frame type can be identified with service type and subtype numbers. This naming followed later in the document and for example TM(3, 2) where service type is 3 and service subtype is 2. The satellite always uses Application Process ID (APID) 820.

General PUS header:

Packet version number	Packet ID			Packet Sequence Control		Packet Data Length	Telemetry Data Field Header	Source Data
	packet type	secondary header flag	Application Process ID	Sequence flags	Packet sequence count			
3 bits	1 bit	1 bit	11 bits	2 bits	14 bits	16 bits	16 bits	variable
static 0x0B 0x34				TBD		0x00 001		

Telemetry frames:

Spare	TM Source Packet PUS version number	Spare	Service Type	Service Subtype	Telemetry Data
1 bit	3 bits	4 bits	8 bits	8 bits	variable
static 0x10					

Figure 3: Simplified PUS Telemetry frame format used for Foresail-1p satellite

3 Housekeeping Data Structures

The Foresail-1p housekeeping data is transmitted on PUS telemetry frames using service type 3. Housekeeping service subtypes are described in the following subsections.

Each housekeeping payload structure starts with a big endian 32-bit unsigned integer which conveys the timestamp when the data was collected as an UNIX timestamp (seconds since the UNIX epoch 00:00:00 1.1.1970 UTC). During passes over commanding ground stations, the same telemetry types are used to downlink historical housekeeping. Thus, the housekeeping frame timestamp might indicate a time in the past.

3.1 OBC Housekeeping

The OBC housekeeping data is transmitted using housekeeping service TM(3,2). The telemetry frame’s payload section has the following structure:

Table 2: OBC Housekeeping structure (38 bytes)

Pos	Type	Name	Description
0	UINT8	Redundancy Side	Currently running OBC side/configuration 0 = Side-A 1 = Side-B 2 = Side-A recovery 3 = Side-B recovery
1	UINT8	FDIR state	OBC SW FDIR state/mode
2	UINT16	System Watchdog Counter	Number of unacknowledged OBC boots. Also incremented by inactivity. FDIR triggered by this counter hitting high numbers without reset.
4	UINT8	FS Mounted	Is the OBC File System Mounted
5	UINT8	Software Revision	The running software revision number.
6	UINT32	Uptime	OBC uptime in seconds
10	UINT8	Heap Free	Heap free (0 = 0%, 255 = 100%)
11	UINT8	CPU Load	Estimate CPU load (0 = 0%, 255 = 100%)
12	UINT16	File System free space	Free space in the filesystem (multiple of 4kBytes).
14	UINT16	Arbiter uptime	Arbiter uptime in seconds. Overflows every 18.2 hours
16	UINT16	Arbiter age	Arbiter’s monotonic slow rate counter (incremented every 5 minutes)
18	UINT16	Arbiter bootcount	Arbiter bootcount
20	INT16	Arbiter temperature	Arbiter MCU temperature in desi Celsius degrees.
22	UINT8	Side A Bootcount	OBC Side-A total bootcount
23	UINT8	Side A Heartbeats	Last heartbeat count received from the OBC Side-A
24	UINT8	Side A Fail Counter	OBC Side-A failure counter
25	UINT8	Side A Fail Reason	OBC Side-A failure reason
26	UINT8	Side B Bootcount	OBC Side-B total bootcount
27	UINT8	Side B Heartbeats	Last heartbeat count received from the OBC Side-B
28	UINT8	Side B Fail Counter	OBC Side-B failure counter
29	UINT8	Side B Fail Reason	OBC Side-B failure reason
30	4*UINT16	Arbiter log	Binary formatted log of arbiter’s actions

3.2 EPS Housekeeping

The EPS housekeeping data is transmitted using the housekeeping service TM(3,3). The telemetry frame's payload section has the following structure:

Table 2: EPS (PCDU+BATT) Housekeeping structure (110 bytes)

Pos	Type	Name	Description
0	UINT32	Uptime	Power Conditioning and Distribution Unit (**PCDU)** MCU uptime in seconds.
4	UINT8	PCDU Boot count	PCDU MCU boot count
5	UINT8	BB Boot count	Battery Board (**BB)** MCU boot count
6	UINT8	APR Boot count	Power Regulator (**APR**) MCU boot count
7	UINT8	PDM Expected	Power distribution switch expected enable states: bit 0: PDM0 PATE Batt bit 1: PDM1 PB Batt bit 2: PDM2 PB 3.6V bit 3: PDM3 CAM 3.6V bit 4: PDM4 MAG 3.6V bit 5: PDM5 OBC 3.6V bit 6: PDM6 UHF 3.6V bit 7: PDM7 ADCS 3.6V
8	UINT8	PDM Faults	Power distribution switch fault states. The bit definitions are the same as for PDM Expected.
9	UINT8	Padding Byte	Padding byte required between the EPS OBC and telemetry database.
10	UINT16	EPS State	Electrical Power System state: { "value": 0, "mask": 1, "string": "BB OFF" }, { "value": 1, "mask": 1, "string": "BB ON" }, { "value": 0, "mask": 14, "string": "HEATER OFF" }, { "value": 2, "mask": 14, "string": "HEATER ON" }, { "value": 4, "mask": 14, "string": "HEATER FORCE_ON" }, { "value": 0, "mask": 112, "string": "BALANCER OFF" }, { "value": 16, "mask": 112, "string": "DISCHARGING LOWER CELLS" }, { "value": 32, "mask": 112, "string": "DISCHARGING UPPER CELLS" }, { "value": 0, "mask": 128, "string": "APR OFF" }, { "value": 128, "mask": 128, "string": "APR ON" }, { "value": 0, "mask": 768, "string": "APR X MPPT" }, { "value": 256, "mask": 768, "string": "APR X MANUAL" }, { "value": 512, "mask": 768, "string": "APR X MPPT" }, { "value": 0, "mask": 3072, "string": "APR Y MPPT" }, { "value": 1024, "mask": 3072, "string": "APR Y MANUAL" }, { "value": 2048, "mask": 3072, "string": "APR Y MPPT" }, { "value": 0, "mask": 12288, "string": "SCOPE IDLE" }, { "value": 4096, "mask": 12288, "string": "SCOPE SCAN" }, { "value": 8192, "mask": 12288, "string": "SCOPE TRACE" }, { "value": 0, "mask": 16384, "string": "SCOPE MEM RDY" }, { "value": 16384, "mask": 16384, "string": "SCOPE MEM BUSY" }
12	UINT16	Heater PWM On Uptime	Heater power on uptime. 5000 count = 100%
14	INT16	Current APR X	**X-axis** (panel X+ or X-) power generation current in milliAmps
16	INT16	Current APR Y	**Y-axis** (panel Y+ or Y-) power generation current in milliAmps
18	INT16	Voltage APR X	X-axis voltage in milliVolts
20	INT16	Voltage APR Y	Y-axis voltage in milliVolts
22	INT16	X Charger Max Current	X-axis maximum current in milliAmps
24	INT16	X Panel Max Voltage	X-axis maximum voltage in milliVolts
26	INT16	Y Charger Max Current	Y-axis maximum current in milliAmps
28	INT16	Y Panel Max Voltage	Y-axis maximum voltage in milliVolts

30	INT16	Battery Pack Voltage	Battery pack total voltage in milliVolts (nominal 6500 - 8200 mV)
32	INT16	Battery Pack Lower Cell Voltage	Battery pack lower cell pair voltage in milliVolts (nominal 3250 - 4100 mV)
34	INT16	Payload Buck Voltage	Output voltage of the buck converter for PB, CAM, MAG in milliVolts. Nominal 3700 mVmilliAmps.
36	INT16	OBC/ADCS Buck Voltage	Output voltage of the buck converter for OBC, ADCS in milliVolts. Nominal 3700 mV
38	INT16	UHF Buck Voltage	Output voltage of the buck converter for UHF in milliVolts. Nominal 3700 mV
40	INT16	PCDU MCU Temperature	PCDU MCU temperature in desiCelsius degrees
42	INT16	BB MCU Temperature	Battery Board MCU temperature in desiCelsius degrees
44	INT16	APR MCU Temperature	Power regulator MCU temperature in desiCelsius degrees
46	INT16	Battery Temperature	Battery pack temperature in desiCelsius degrees
48	INT16	Panel X- temperature	Panel temperature in desiCelsius degrees
50	INT16	Panel X+ temperature	Panel temperature in desiCelsius degrees
52	INT16	Panel Y- temperature	Panel temperature in desiCelsius degrees
54	INT16	Panel Y+ temperature	Panel temperature in desiCelsius degrees
56	INT16	Battery Current	Immediate battery charging current in milliAmps
58	INT16	Minimum Battery Current	Minimum battery charging current in milliAmps [1]
60	INT16	Maximum Battery Current	Maximum battery charging current in milliAmps
62	INT16	PATE Current	Immediate **PATE** (Particle Telescope) Battery voltage bus current consumption in milliAmperes.
64	INT16	PB Battery Current	Immediate **PB** (Plasma Brake) Battery voltage bus current consumption in milliAmperes.
66	INT16	PB 3.6V Current	Immediate PB 3.6V bus current consumption in milliAmps
68	INT16	CAM Current	Immediate **CAM** (Camera) 3.6V bus current consumption in milliAmps
70	INT16	MATTI Current	Immediate **MATTI** (Magnetometer) 3.6V bus current consumption in milliAmps
72	INT16	OBC Current	Immediate **OBC** (On-Board Computer) 3.6V bus current consumption in milliAmps
74	INT16	UHF Current	Immediate **UHF** (Ultra-High Frequency) Radio 3.6V bus current consumption in milliAmps
76	INT16	ADCS Current	Immediate **ADCS** (Attitude Determination and Control System) 3.6V bus current consumption in milliAmps
78	INT16	PATE Minimum Current	Minimum PATE Battery voltage bus current consumption in milliAmps [2]
80	INT16	PB Battery Minimum Current	Minimum PB Battery voltage bus current consumption in milliAmps
82	INT16	PB 3.6V Minimum Current	Minimum PB 3.6V bus current consumption in milliAmps
84	INT16	CAM Minimum Current	Minimum CAM 3.6V bus current consumption in milliAmps
86	INT16	MATTI Minimum Current	Minimum MATTI 3.6V bus current consumption in milliAmps
88	INT16	OBC Minimum Current	Minimum OBC 3.6V bus current consumption in milliAmps
90	INT16	UHF Minimum Current	Minimum UHF Radio 3.6V bus current consumption in milliAmps

[1] The battery pack minimum and maximum currents are from the last 60 seconds (TBC, configurable).
[2] The power distribution switch minimum and maximum currents are from the last 60 seconds (TBC, configurable).

92	INT16	ADCS Minimum Current	Minimum ADCS 3.6V bus current consumption in milliAmps
94	INT16	PATE Maximum Current	Maximum PATE battery voltage bus current consumption in milliAmps
96	INT16	PB Battery Maximum Current	Maximum Plasma Brake battery voltage bus current consumption in milliAmps
98	INT16	PB 3v6 Maximum Current	Maximum Plasma Brake 3.6V bus current consumption in milliAmps
100	INT16	Camera Maximum Current	Maximum Camera 3.6V bus current consumption in milliAmps
102	INT16	MATTI Magnetometer Maximum Current	Maximum Magnetometer 3.6V bus current consumption in milliAmps
104	INT16	OBC Maximum Current	Maximum OBC 3.6V bus current consumption in milliAmps
106	INT16	UHF Maximum Current	Maximum UHF Radio 3.6V bus current consumption in milliAmps
108	INT16	ADCS Maximum Current	Maximum ADCS 3.6V bus current consumption in milliAmps

3.3 ADCS Housekeeping (Structure updated, descriptions TBU)

The ADCS housekeeping data is transmitted using the housekeeping service TM(3,4). The telemetry frame’s payload section has the following structure:

Table 2: ADCS Housekeeping structure (79 bytes)

Pos	Type	Name	Description
0	UINT8	Determination State	0 = Off 1 = Triad 2 = FLAE 3 = Kalman 4 = Kalman Right Handed
1	UINT8	Control State	0 = Off 1 = Bdot 2 = Ruiter Control 3 = PD Control
2	FLOAT	MJD	ADCS systems current time as Modified Julian Date
6	3x FLOAT	Position vector	Orbit propagators output position vector in ECI frame. Units in kilometers.
18	3x FLOAT	Velocity vector	Orbit propagators output velocity vector in ECI frame. Units in kilometers per second.
30	UINT32	Attitude Quaternion Compressed	First 32 bits of compressed attitude quaternion. The whole structure of this compression is getting the three smallest elements, quantizing them in order to 12 bits using the limit values for smallest elements (+-1/sqrt(2)). The final element can be then calculated using the 4 last bits of which 2 is the index of the largest (in x, y, z, w) and 1 is the sign of the largest. This is done to save space in the onboard database and significantly lower the size of ADCS housekeeping. Should give multiple decimals of accuracy to what is actually used for calculations on board.
34	UINT8	Attitude Quaternion Compressed Overflow	Last 8 bits of previous
35	3x INT16	Estimated Angular Rate	Satellite angular rates (Pitch, Yaw, Roll) as milliradians per second.

41	3x INT16	Estimated Mag Bias	In nanoTeslas.
47	3x INT16	Estimated Gyro Bias	In milliradians per second.
53	INT16	Attitude Variance	
55	INT16	Angular Velocity Variance	
57	INT16	Mag Bias Variance	
59	INT16	Gyro Bias Variance	
61	3x INT16	Magnetometer Measurement	
67	3x INT16	Gyroscope Measurement	
73	3x INT16	Sun Vector Measurement	

3.4 UHF Housekeeping

The UHF housekeeping data is transmitted using the housekeeping service TM(3,5). The telemetry frame's payload section has the following structure:

Table 2: UHF Housekeeping structure (56 bytes)

Pos	Type	Name	Description
0	UINT32	Uptime	MCU uptime in seconds
4	UINT16	Bootcount	Number of boots
6	UINT8	FDIR counter	Counter which determines if recovery operations should be run.
7	UINT8	Watchdog Reset Count	
8	UINT8	MBE Count	Multi Bit Error fixed in memories
9	UINT8	Bus Sync Errors	Bus sync word errors
10	UINT8	Bus Len Errors	Bus length byte errors
11	UINT8	Bus CRC Errors	Bus packet CRC errors
12	UINT8	Bus Receive Timeouts	
13	UINT32	Total TX frames	Total number of transmitted frames
17	UINT32	Total RX frames	Total number of received frames
21	UINT32	Total TX HAM frames	Total number of transmitted HAM frames
25	UINT32	Total RX HAM frames	Total number of received HAM frames
29	UINT16	Hardware Side A bootcount	
31	UINT16	Hardware Side B bootcount	
33	UINT8	Side	Current hardware redundancy side: 0 = Side-A, 1 = Side-B
34	UINT8	RX symbol rate	Rx symbolrate 0 = GMSK 4800 (unused), 1 = GMSK 9600, 2 = GMSK 19200, 3 = GMSK 38400
35	UINT8	TX symbol rate	Tx symbolrate 0 = GMSK 4800 (unused), 1 = GMSK 9600, 2 = GMSK 19200, 3 = GMSK 38400
36	INT32	My window length	Satellite transmission window length in ms.
40	INT32	Peer window length	Assumed ground transmission window length in ms.
44	INT16	MCU Temperature	Temperature of the Radio MCU in desi Celsius degrees.
46	INT16	PA Temperature	Temperature of the active side Power Amplifier in desi Celsius degrees.

48	INT16	Background RSSI	RSSI of the background noise. (Rolling Average) Calibration: rssi = value - 111dBm
50	INT16	Background MAX RSSI	RSSI of the background noise. (Maximum value from rolling average (Same calibration as on previous field)
52	INT16	Last RSSI	RSSI (Relative Signal Strength Indicator) of last received frame.
54	INT16	Last frequency offset	Frequency offset of last received frame . offset = value * 19.07 Hz

3.5 File Transfer

The Foresail-1p satellite mission supports reliable data transfer based on files. The files are transferred by the file transfer system in blocks of 160 bytes which are transmitted as individual frames. These frames can be received during over passes over Finland/Europe or at other times when a downlink-only file transfer has been scheduled.

The file transfer service has the Service Type code 6 and has the following downlink telemetry service types:

- TM(6,8) Downlink Init Report
- TM(6,13) Downlink Transmit

TM(6,8) Downlink Init Report
The downlink init report is a response frame to the downlink init TC or it can be broadcasted when a file dump has been autonomously initialized. Init reports contain crucial information such as its filename, file size and CRC-checksum. The frame payload field has the following structure:

Table: TM(6,8) Downlink Init Report structure

Transfer Index	File size	CRC32	Filename
uint8_t	uint32	uint32	N x char

TM(6,13) Downlink Transmit
The transfer indexes are automatically coordinated by the Foresail-1p mission control software. The standard file block size is 160 bytes. All the blocks except the last one are this length. The frame payload field has the following structure:

Table: TM(6,13) Downlink Transmit structure

Transfer Index	Block Index	Block Data
uint8	uint16	N x uint8

4 Radio Amateur Repeater

The Foresail-1p Amateur Packet Repeater is implemented on the virtual channel number #3 of the Skylink Protocol. This is used for forwarding encapsulated AX.25 / APRS (Automatic Position Reporting System) messages defined by the [APRS Protocol Reference V1.0]

4.1 Functional Overview

Encapsulated APRS Packets transmitted by Hamradio stations will be forwarded back to Earth by the satellite. These repeated packets include the satellite callsign and can be forwarded for example to APRS.fi by gateway stations operated by HamRadio enthusiasts. Otherwise the repeater can be also used as a general AX.25 repeater without any structure in the information field. No validation or checking is performed for the data inside the AX.25 information field on board.

4.2 Operational description

The Amateur Packet Repeater will function as follows:

- Amateur ground stations can send packets using the Skylink protocol with APRS messages encapsulated as a payload to the targeted Virtual Channel #3
- Any packet arriving on the Skylink Virtual Channel #3 of the satellite link, starting with a known valid AX.25 header (see below), will be modified using the satellite callsign as the repeating station and will be re-transmitted back to the Earth.
- These AX.25 packets will not be forwarded to the satellite internal bus in any situation but repeated during the next available free TDD window on Virtual Channel #3. Exact response time will vary and depends on the Skylink protocol configuration.
- After successful APRS repeating the satellite stores the repeated frame to its non-volatile memory (up to 8 frames) and re-transmits frames from memory.
- The default re-transmit repetition count and repetition interval is 4 times every 4 minutes. These configurations may change during the mission.
- For each forwarded incoming or outgoing APRS packet a counter is increased. These counters are included in UHF subsystems in housekeeping telemetry.

4.2 APRS Packet Encapsulation

The payload of the Skylink Protocol on Virtual Channel #3 can include an AX.25 frame APRS message described below. The maximum length of an APRS frame (included header, data etc.) is **128 bytes.**

The satellite will add itself to the "Digipeater Addresses" as a digipeating (repeating) station and forward the packet back with a small latency.

The AX.25 Frame All APRS transmissions use AX.25 UI-frames, with 9 fields of data:

	AX.25 UI-FRAME FORMAT								
	Flag	***Destination Address***	***Source Address***	***Digipeater Addresses (0-8)***	***Control Field (UI)***	***Protocol ID***	***INFORMATION FIELD***	***FCS***	***Flag***
Bytes:	1	7	7	0–56	1	1	1–256	2	1

The Field Descriptions are as follows:

Note: This is how the frame is supposed to be. There are some unpatchable bugs onboard, which result in the frame not being a standard AX.25 frame. These are listed on the next page.

- Start flag: Static 0x7E
- Destination Address - This field should contain an APRS destination callsign or APRS destination
  - Accepted destinations are: OH2F1S, APRS, ALL*, BEACON, CQ*
- Source Address - This field should contain the callsign and SSID of the transmitting station
- Digipeater Addresses - Up to 8 digipeater callsigns may be included in this field, the satellite adds its own callsign to this list. If the list already has 8 repeaters then the first one will be discarded and the satellite callsign will be added as the last repeater
- Control Field - This field must always be set to **0x03** (UI-frame)
- Protocol ID - This field must always be set to **0xF0** (no layer 3 protocol)
- Information Field - This field contains APRS data. The first character of this field should be the APRS Data Type Identifier that specifies the nature of the data that follows, which might be;
  - Location information,
  - Weather information,
  - Telemetry,
  - Messages and bulletins,
  - Queries,
  - Query responses
  - Content of the field is not validated.
- Frame Check Sequence: X.25 CRC-16 checksum
- Eng flag: Static 0x7E

Foresail1p bug information from skylink README:
Important to note about the repeater frames sent to Foresail1p. These will be fixed for Aalto-3 and other future projects: The radio board code can not be updated and has a few errors in repeater frame parsing.

1. There is no HLDC start or end flag (0x7E) that is standard to AX.25 frames. The frames sent to the satellite will also not include this.
2. The CRC (FCS) is MSB instead of LSB.
3. The CRC sent back will only have the correct second byte of the CRC and the first one will be from the original frame.
4. Sending skylink frame CRC field which is default on the repeater channel of the satellite will result in the repeater frame not being accepted. However the frame that is sent back will include this CRC which can still be used to check validity of frame. This CRC will be the 4 final bytes of the entire frame after Golay, RS, Scrambling and will cover the whole skylink frame. Details (used polynomial etc.) can be found in src/crc.c.

Special Thanks to Daniel (DL7NDR) for helping us debug this.

A generic APRS Information Field can be seen as;

	Generic APRS Information Field			
	Data Type ID	APRS Data	APRS Data Extension	Comment
Bytes:	1	n	7	n

A basic packet identification on the satellite will be done by checking that the:

- Control Field is 0x03
- Protocol ID field is 0xF0
- Checking the FCS (X.25 CRC-16)

Note: Frames not complying to this format will be ignored!

Any HDLC related features (bit stuffing etc.) are not applied and shall not be applied for the payload.

4.3 APRS Packet Encapsulation Examples

Uplink:

	Destination address: ALL
	Source address: OH2AGS-0
0x30	Control Field: UI-Frame
0xF0	Protocol ID: No layer 3 protocol implemented
…	APRS data or any other message
0x?? 0x??	FCS

Downlink:

	Destination address: ALL
	Source address: OH2AGS-0
	Repeater address : OH2F1S-11

0x30	Control Field
0xF0	Protocol ID
…	APRS data or any other message
0x?? 0x??	FCS with only the second byte correct. See notes above.

Broadcast:

	Destination address: BEACON
	Source address: OH2F1S-11
0x30	Control Field
0xF0	Protocol ID
…	APRS data or any other message
0x?? 0x??	FCS with only the second byte correct. See notes above.

Appendix A: List of used Telemetry service types

Complete list of all telemetry service type and subtype codes can be found from Telemetry receiver softwares source code in JSON-format.

Service Type	Name	Description
1	Telecommand Verification	Telecommand verification (acknowledgements)
2	Ping	Ping service
3	Housekeeping	Housekeeping service (real-time and historical)
4	Events	On-board Event service
5	Time	Time management service
6	File Transfer	File Transfer service
7	File System	File System service
8	Configuration System	On board Configuration System
9	Bus	Bus commanding service
10	FDIR	Failure Detection, Isolation and Recovery service
11	OBC	On-Board Computer commanding service
128	Scheduler	On-Board telecommand scheduler service
129	EPS	Electrical Power System commanding service
130	UHF	UHF TT&C commanding service
131	ADCS	Attitude Determination and Control System (ADCS) commanding
132	ADCS Sensors	ADCS sensor commanding service
133	ADCS Recorder	ADCS data recorder service
134	Compressor	Data compressing and decompressing service
135	PATE	Particle Telescope commanding service
136	Plasma Brake	Plasma Brake commanding service
137	MATTI	Magnetometer commanding service
138	Camera	Camera commanding service
139	ADCS Magnetorquer	Magnetorquer commanding service
140	Sequencer	Telecommand sequencer service

Appendix B: Example Frames

OBC housekeeping frame

```
66 4F 48 32 46 31 53 09 C6 7F 05 44 00 FA 00 FA 0B 34 0B 34 00 2D 10 03 02 69 29 A3 6C
00 00 02 00 00 1F 43 3E 00 00 89 00 00 00 51 3E 19 66 A7 00 EE 00 CA 00 03 01 AA 11 00
05 46 04 46 24 E4 05 E4 25 C2 D0 AE F9
```

EPS housekeeping frame

```
66 4F 48 32 46 31 53 09 C6 85 08 44 00 FA 00 F6 25 17 EF 0B 34 0B 34 00 75 10 03 03 69
29 A3 6D 44 3E 00 00 27 00 00 60 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 EC 0D BE 0D C4 0D 38 01 00 00 00 00 00 00 F4 00 F7 00 F7 00 F4
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 59 00 34 00 00 00 00 00 00 00 00 00
00 00 00 00 57 00 34 00 00 00 00 00 01 00 00 00 00 00 00 00 59 00 7B 02 00 00 0D A2 81
6E
```

UHF Housekeeping Frame

```
66 4F 48 32 46 31 53 09 C6 81 05 44 00 FA 00 04 0B 34 0B 34 00 3F 10 03 04 69 29 A3 6C
DC 3D 00 00 B8 00 02 00 00 00 00 00 00 27 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 1F C4 06 00 01 01 FA 00 00 00 FA 00 00 00 04 01 02 01 23 00 68 00 00 00 00 00 3D 2D
60 F6
```

ADCS Housekeeping Frame

```
66 4F 48 32 46 31 53 09 C6 80 05 44 00 FA 00 8E 0B 34 0B 34 00 56 10 03 05 69 29 A3 6C
00 00 90 4F 6E 47 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 80 00 08 80 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F4 01 09 00 64
00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F3 01 BC 6C
```

Deployment Housekeeping Frame:

```
66 4F 48 32 46 31 53 09 C6 A3 05 44 00 FA 00 F5 0B 34 0B 34 00 11 10 03 06 69 4B 09 3E
00 00 04 00 0F 00 00 00 00 00 EF 54 26 58
```

Nominal Event (RID: 506, Timestamp: 2026-01-26T13:31:18, Info: b'\x96\xff\xff\xff' (-106))

```
66 4F 48 32 46 31 53 08 3A DB 05 44 00 FA 00 F1 0B 34 0B 34 00 0D 10 04 04 69 77 6C A6
01 FA 96 FF FF FF C3 63 34 37
```

Execution Completion success acknowledgement TM(1,7)

```
66 4F 48 32 46 31 53 08 F3 72 05 44 00 FA 00 F2 0B 34 0B 34 00 09 10 01 07 1B 34 DF 88
00 00 66 AA 0F C0
```

Ham Repeater Frame (Message: “Hello from Satlab!”)

```
66 4F 48 32 46 31 53 13 00 0D 05 44 00 FA 00 FA 9E 90 64 8C 62 A6 60 9E 90 64 82 8E A6
60 9E 90 64 8C 62 A6 F7 03 F0 48 65 6C 6C 6F 20 66 72 6F 6D 20 53 61 74 6C 61 62 21 6D
53 B2 B0 FF D4
```