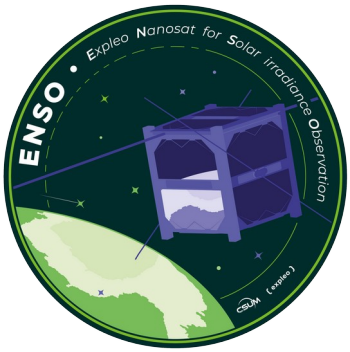
DECODING ENSO MISSION AND HAMRADIO APPLICATION



1 INTRODUCTION

This document describes the new satellite of the University Space Center of Montpellier (CSUM): **ENSO** (Expleo Nanosat for Solar-irradiance Observations). It shows the satellite characteristics, its mission and describes how to decode beacons received from the satellites. Also, this document provides information about the RadioHam Application running on board the satellite and how members of the Amateur Radio community can use it.

1.1 The Mission

ENSO is the result of a collaboration between Expleo France and the Centre Spatial Universitaire de Montpellier. Its main goal is to study Earth's ionosphere by generating and emitting High Frequency (HF) signals able to probe this layer of the atmosphere for environmental research. The mission also aims at taking picture of Earth's surface by means of a compact camera installed on the satellite's payload.

The reception on the ground of the HF beacon by the South African National Space Agency (SANSA) will allow them to better characterize the ionosphere and calibrate the ground equipment used for reception. Other Ground Segments of the SuperDARN newtwork, as well as CPUT's own Ground Segment, may also be used within the project. The mission shall last for one year, extendable to 2 years.

1.2 Launch Information

The satellite will be launched on board Transporter 9 (SpaceX) at the beginning of November at around 520 km of altitude at a 97 degrees inclination. More information about the launch will be provided soon.

2 HAM RADIO APPLICATION

The HF ENSO beacon is an opportunity for the radio amateur community to explore and characterize the propagation channels in the ionosphere (reflection and propagation) on white areas where there is no radiofrequency coverage (oceans, deserts, poles, etc.). And of course, at the same time, to calibrate their own HF ground station.

The HAM RADIO Community will be able to activate the HF beacon upon request to the mission operators. The system in charge of coordinate the different requests will be made available before the launch date. A more detailed document and a set of tools will then be provided to the community one week after the launch, provided the commissioning phase is nominal.

3 DECODING

3.1 PHYSICAL LAYER

PARAMETER	VALUE
HAM RADIO CALLSIGN	FX6FRC
DOWNLINK FREQUENCY	436.5 MHz
UPLINK FREQUENCY	435.5 MHz
MAX BANDWIDTH	20 kHz
MODULATION*	GMSK
DATARATE (UP/DOWNLINK) *	2400 bps
PROTOCOL	AX25
BEACON INTERVAL (Default)	29 seconds
EXPECTED TIME FOR FIRST BEACON	TBD

** The platform allows the change of up and downlink modulation scheme during the mission. The following configuration are also possible. The HAMRadio community will be informed in case the RF link parameters are changed:*
- AFSK 1k2 bps – AX25 protocol.
- GMSK 9k6 bps – AX25 protocol.

3.2 BEACON PACKET

The satellite communication systems is based on the AX.25 protocol. The packet structure is shown in the tables below.

3.2.A AX25 PACKET

AX25 PACKET					
PREAMBLE	FLAG	AX25_HEADER	AX25_INFO	FCS (CRC)	FLAG
32 Bytes	1 Byte	16 Bytes	142 Bytes	2 Bytes	1 Bytes
0x55	0x7E			CRC-16/ X.25	0x7E

3.2.B AX25 HEADER

AX25 HEADER					
ADDRESS				CONTROL	PID
DESTINATION*	DESTINATION SSID	SOURCE*	SOURCE SSID		
6 Bytes	1 Byte	6 Byte	1 Byte	1 Byte	1 Byte
0x8C689694B040	0xE0	SAT CALLSIGN	0xE1	0x03	0xF0

* In AX25 protocol, the destination and source callsigns are equal to the equivalent ASCII characters in hex shifted one bit to the left. Ex: F4KJX = 0x8C689694B040

3.2.C AX25 INFO/DATA

AX25 DATA							
Length	Frame Type	TS	OBDH	EPS	TTC	PAYLOAD	HAM MESSAGE
1 Byte	1 Byte	4 Bytes	16 Bytes	32 Bytes	16 Bytes	48 Bytes	24 Bytes
0xEA	0x10	UNIX Timestamp*					

** Sent with LSByte (Less Significant Byte) first. Ref: https://www.unixtimestamp.com/*

3.2.C.1 OBDH DATA

Field	Description	Length (bytes)	Value
OBDH Timestamp	Time at the OBDH system	4	Unix Timestamp, LSB
Temperature	Last recorded temperature by the OBDH	2	Signed Integer, Big Endian Unit: °C
Satellite Mode	Mode of the Satellite	1	0x00 -> STANDBY 0x01 -> DEPLOY 0x02 -> COMMISSIONNING 0x03 -> COMM_PL 0x04 -> MISSION 0x05 -> LOW_P_MISSION 0x06 -> TRANSMIT 0x07 -> SURVIVAL 0x08 -> SILENT
OBDH Mode	Mode of the OBDH subsystem	1	0x11 -> STANDBY 0x22 -> DEPLOY 0x33 -> COMMISSIONNING 0x44 -> COMM_PL 0x55 -> MISSION 0x66 -> LOW_POWER_MISSION 0x77 -> SILENT 0xFF -> POR
Bytes to transmit	Number of data bytes stored in memory to be downlinked	4	Unsigned Integer Big Endian
Number of Resets	Number of OBDH resets	2	Unsigned Integer Big Endian
Number of Errors	Number of errors that occurred in the OBDH	2	Unsigned Integer Big Endian

3.2.C.2 EPS DATA

Field	Description	Length (bytes)	Value

EPS Mode	Mode of the EPS subsystem	1	0x00 -> IDLE 0x11 -> SURVIVAL 0x22 -> STANDBY 0x33 -> DEPLOY 0x44 -> COMMISSIONNING 0x55 -> MISSION 0x66 -> LOW_POWER_MISSION 0x77 -> SILENT
Battery Voltage	Last Battery Voltage	1	=int(**byte**)*20 Unit: mV
Battery Temperature	Last measured battery temperature	1	Signed Int Unit: °C
Min Battery Voltage	Minimum Battery Voltage measured since reboot	1	=int(**byte**)*20 Unit: mV
Max Battery Voltage	Maximum Battery Voltage measured since reboot	1	=int(**byte**)*20 Unit: mV
Avg Battery Voltage	Average Battery Voltage measured since reboot	1	=int(**byte**)*20 Unit: mV
Avg Charge Current	Average Charge Current measured since reboot	1	=int(**byte**)*12 Unit: mA
Max Charge Current	Maximum Charge Current measured since reboot	1	=int(**byte**)*12 Unit: mA
Z- Face Temperature	Temperature measured at the –Z face of the satellite	1	Signed int Unit: °C
OBDH Current	Current consumption of the OBDH	1	Unsigned Int Unit: mA
EPS Current	Current consumption of the EPS	1	Unsigned Int Unit: mA
TTC µC Current	Current consumption of the TTC Microcontroller	1	Unsigned Int Unit: mA
TTC PA Current	Current consumption of the Power Amplifier of the TTC	1	=int(**byte**)*5 Unit: mA
TTC PA Current MAX	MAX Current consumption of the Power Amplifier of the TTC	1	=int(**byte**)*5 Unit: mA
PAYLOAD Current	Current consumption of the Payload	1	=int(**byte**) Unit: mA
Charge Current	Total charge current of the battery	1	=int(**byte**) Unit: mA
Temperature X+	Temperature of the X+ face	1	Temperature = (signed int8) byte
Temperature X-	Temperature of the X- face	1	Temperature = (signed int8) byte
Temperature Y+	Temperature of the Y+ face	1	Temperature = (signed int8) byte
Temperature Y-	Temperature of the Y- face	1	Temperature = (signed int8) byte
Temperature Z+	Temperature of the Z+ face	1	Temperature = (signed int8) byte
OBDH Voltage	Supply Voltage of the OBDH	1	Voltage [mV] = (Byte * 10) + 4000
TTC PA Voltage	Supply Voltage of the Power amplifier	1	Voltage [mV] = (Byte * 10) + 4000
Payload Voltage	Supply Voltage of the Payload	1	Voltage [mV] = (Byte * 10) + 4000
MOS1 Voltage	Voltage of the MOSFET 1 (Passivation)	1	Voltage [mV] = (Byte + 2200) * 0.805

MOS2 Voltage	Voltage of the MOSFET 2 (Passivation)	1	Voltage [mV] = (Byte + 2200) * 0.805
MOS3 Voltage	Voltage of the MOSFET 3 (Passivation)	1	Voltage [mV] = (Byte + 2200) * 0.805
Reference Voltage	Passivation Reference Voltage	2	Voltage [mV] = (Byte) * 0.805
Temperature 5V REG	Temperature of 5V regulator	1	Temperature = (signed int8) byte
Temperature 6V REG	Temperature of 6V regulator	1	Temperature = (signed int8) byte
TTC MCU Voltage	Supply Voltage of the TTC microcontroller	1	Voltage [mV] = (Byte * 10) + 4000

3.2.C.3 TTC DATA

Field	Description	Length (bytes)	Value
TTC Mode	Mode of the TTC subsystem	1	0x01 -> IDLE 0x11 -> BEACON 0x22 -> COMMISSIONNING 0x44 -> SILENT
Number of TTC Resets	Number of resets of the TTC subsystem	2	Unsigned Int
Last reset cause	Cause of last TTC reset	1	0x11 -> POR (Power supply reset) 0x22 -> WDTTO (Watchdog) 0x33 -> OSC (Oscillator Error) 0x44 -> HW (Reset Pin) 0x55 -> DEBUG (Debugger Reset) 0x77 -> RI (Software Reset)
Number of Received valid packets	Number of received packets with valid CRC since reset	2	Unsigned Int
Number of transmitted packets	Number of transmitted packets since reset	2	Unsigned Int
Measured Transmission Power	Output RF power measured by TTC	2	Unsigned Int ADC counts (max: 4048)
Last Error Code	Cause of last error in the TTC	1	0x00 -> NULL 0x11 -> RADIO_HW_ERROR 0x22 -> TX_QUEUE_FULL 0x33 -> RX_QUEUE_FULL 0x44 -> TX_BUS_QUEUE_FULL 0x55 -> RX_BUS_QUEUE_FULL 0x66 -> OBC_TEMP_HW_ERROR 0x77 -> OBC_TEMP_H_LIMIT_ERROR 0x88 -> OBC_TEMP_L_LIMIT_ERROR 0x99 -> PA_TEMP_HW_ERROR 0xAA -> PA_TEMP_H_LIMIT_ERROR 0xBB -> PA_TEMP_L_LIMIT_ERROR 0xCC -> OBDH_NACK 0xDD -> PF_RESET_REQ 0xD1 -> TTC_RESET_REQ

			0xEE -> RADIO_TASK_TIMEOUT 0xFF -> RADIO_UNQUEUE 0x01 -> OBDH_STATUS_REQ 0x02 -> OBDH_BDR_REQ 0xA1 -> FRAM_ID_ERROR 0xA2 -> FRAM_HW_ERROR 0xA3 -> FRAM_READ_ERROR 0xA4 -> FRAM_WRITE_ERROR 0xA5 -> EVENT_QUEUE_READ_ERROR
Power Configuration	Output Power Configuration	1	Unsigned Int Max = 120
Power Amplifier Temperature	Last Measured temperature of the Power Amplifier	1	Signed Int Unit: °C
RSSI of last received packet	Received Signal Strength Indicator of last received packet	1	-1 * Int(byte) Unit: dBm
Frequency Deviation of last received packet	Frequency Deviation of last received packet with valid CRC	1	17 * SignedInt(byte) Unit: Hz
Beacon period	Time interval between beacon transmissions	1	Int(byte) Unit: seconds

3.2.C.4 MESSAGE

Field	Length (bytes)	Value
Message	24	ASCII Character

3.3 TOOLS FOR RECEIVING AND DECODING BEACONS

3.3.A CSUM GITHUB PAGE

CSUM's Github page provides a set of tools to decode signals from the satellites and send the packets directly to CSUM servers. More information can be found in the documentation page of the GitHub repository.

3.3.B CSUM DATA DIFFUSION PLATFORM

CSUM's Data Diffusion Platform (DDP) provide users direct access to radio packet decoded from our satellites. Amateur Radio members who push packets to CSUM's servers can see their packets decoded in real-time by accessing the DDP Public Page of ENSO (available soon). In case you want to build your own decoder and send packets to CSUM's server, you can use the information below:

Parameter	Value
Protocol	Simple Downlink Share Convention (SiDS)
Endpoint	https://ddp.csum.umontpellier.fr/store_beacon
Method	HTTP POST
Data Structure (Minimum)	**source**: HAM Callsign of the receiving station **frame**: AX.25 received frame in hex (no Flag, no CRC)

	Example:
	{
	'source': 'F4KJX ',
	'frame':
	'8c689694b040e08cb06c8ca482e103f0ea10c7524a5c71524a5cb501023378b002004200af0044b355a8bbb30000810e083c1f0000002210001100000800ef0e00642200001d5104ff078000800040000504ff07e000c0001e000000000000000000000000000000000000000000000000000000030000524f42555354412d315520465836465241000'
	}

4 CONTACT INFORMATION

In case you have questions about the information in this document or related to this mission in general, do not hesitate to contact us through the following email address:

csum-radioham-contact@umontpellier.fr

Good luck to all!

73's