



FOSSASAT-1 Communication Guide



Table of Contents

1 Introduction.....	3
2 Decoder Setup	4
2.1 Hardware.....	4
2.2 Software	4
3 Communication Protocol.....	5
3.1 Terminology.....	5
3.2 Frame Structure.....	5
3.3 Function IDs.....	5
3.4 Examples.....	7
4 Uploading To The Ground Station Database	8
Introduction.....	8
Browsing.....	8
Registering.....	8
Logging Transmissions.....	9
5 References.....	11

Please contact Fossa Systems directly using the following form if you have any issues:

<https://fossa.systems/contact/>

1 Introduction

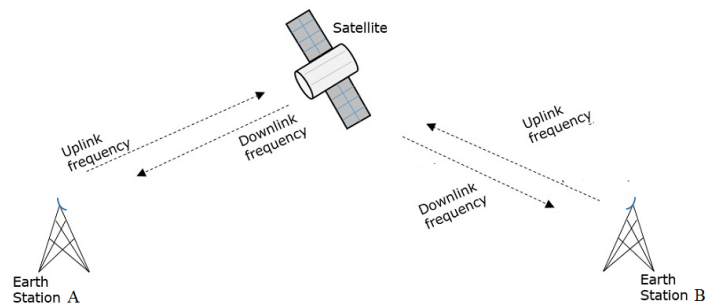
Before attempting to carry out any transmissions intended for FOSSASAT-1, please check its status here: <http://groundstationdatabase.com/index.php>

We **urge** users to only decode telemetry and not send commands until Fossa Systems has evaluated correct operation of the satellite.

The purpose of this document is to explain the communication system used by FOSSASAT-1. First, decoder setup is explained in chapter 2, both in terms of recommended hardware and software. Chapter 3 explains the communication protocol employed by the satellite. Finally, chapter 4 shows how to upload any received frames into a global database.

The main communication channel used by the satellite is its LoRa radio, with the following configuration:

- Carrier frequency 436.7 MHz
- Bandwidth 125 kHz
- Spreading factor 11
- Coding rate 4/8
- Sync word 0x0F0F



In general, exchanges with the satellite can occur on either automatic, or manual basis. Automatic frames are sent by the satellite based on fixed transmission period, and consist of system information, such as battery voltage, temperatures, solar cell voltages, etc. (see RESP_SYSTEM_INFO frame description in chapter 3.3). These frames are sent via LoRa modulation each 30 - 180 seconds. In addition, system frame is also transmitted via RTTY every 3-4 minutes with the following configuration:

- Carrier frequency 436.7 MHz
- Frequency shift 182 Hz
- Baudrate 45 baud
- Encoding ITA2 (5-bit, aka "Baudot code")
- Single stop bit, no parity

In RTTY mode, system information is sent as ASCII-encoded hexadecimal numbers, so that every 2 hexadecimal numbers give a single byte. The format of LoRa and RTTY system info frame is identical.

Manual exchanges with the satellite are initiated by the ground station and are replied to by the satellite. Because the satellite uses multiple different modulations and contains a single radio module, only single communication channel is active at any given point: the satellite listens for incoming LoRa transmissions for 5 seconds out of any given 9-second period.

It should be noted that the satellite contains safety precautions that may prevent it from responding under certain conditions. In addition, some on-board actions have higher priority than responding to incoming frames. Because of this, it cannot be guaranteed that every frame sent will induce a response.

2 Decoder Setup

The following is the recommended setup for the ground station. However, all the hardware and software below can be changed to suit the user's needs. The only mandatory component of the system is the LoRa radio module - though the exact type, i.e. SX127x, SX126x or RFM9x is completely up to the user as well.

2.1 Hardware

FOSSASAT-1 frames are transmitted using LoRa modulation. To receive them, it is recommended to use dedicated SX126x LoRa radio, together with an appropriate power source. It is also possible to receive FOSSASAT-1 frames using SX127x series of LoRa radios, however, SX126x is recommended as that is the satellite on-board radio, offers improved sensitivity and allows integration with TCXO (temperature-compensated crystal oscillator).

To control the LoRa radio, an appropriate microcontroller must also be used. Due to the nature of used communication libraries, any of the following microcontroller families may be used – provided it is running on the Arduino platform:

- AVR ATmega (Arduino Uno is recommended)
- ESP8266 and ESP32
- STM32
- SAMD

SPI is used as the hardware interface between the LoRa radio and the microcontroller. In addition, DIO0 and DIO1 pins must be connected. On SX126x, the BUSY pin must also be connected to the MCU.

2.2 Software

To receive and decode FOSSASAT-1 frames, Arduino IDE 1.8.9 (or higher) is required. The software may be downloaded from <https://www.arduino.cc/en/Main/Software>.

In addition to Arduino IDE, the following libraries must also be installed. For a guide on installing Arduino libraries, see [2].

- RadioLib 1.2.2 (or higher) - <https://github.com/jgromes/RadioLib>
- FOSSA-Comms 1.0.0 - <https://github.com/Bambofy/FOSSA-Comms>

RadioLib is used as the driver for LoRa radio modules, whereas FOSSA-Comms is the implementation of FOSSASAT communication protocol described in the following chapter. Both libraries contain ready-to-use examples on both LoRa radio control (in RadioLib), as well as sending and receiving various FOSSASAT frames (in FOSSA-Comms). These examples include:

- RadioLib/SX126x/SX126x_Receive – basic reception of LoRa packets
- RadioLib/SX126x/ X126x_Receive_Interrupt – non-blocking reception of LoRa packets
- FOSSA-Comms/GetPacketInfo – reading data about the last received packet
- FOSSA-Comms/GetSystemInfo – reading satellite system info
- FOSSA-Comms/PingPong – ping-pong exchange between the ground station and the satellite
- FOSSA-Comms/RepeatMessage – retransmission of arbitrary message
- FOSSA-Comms/RepeatMessageCustom – retransmission of arbitrary message using custom LoRa configuration

3 Communication Protocol

FOSSASAT-1 implements simple frame-based communication protocol using LoRa modem. This protocol allows ground stations to perform basic actions with the satellite, for example fetch current status, send ping/pong exchanges or request retransmission of a message.

3.1 Terminology

- Packet = Physical layer minimum data unit, contains preamble, sync word, header, CRC and payload. See [1] for details.
- Frame = FOSSASAT Communication Protocol minimum data unit. Contains callsign, function ID and optional data. Payload of physical layer Packet.
- Field = Logical section of frame. For example, callsign and function ID are fields.
- Section = Collection of fields.
- Callsign = String of printable ASCII characters, sent as the first field of each frame. Set to "FOSSASAT-1" by default.
- FCP = FOSSASAT Communication Protocol.
- Command = uplink message (from ground station to satellite)
- Response = downlink message (from satellite to ground station)

3.2 Frame Structure

Each FCP frame consist of the following fields:

<callsign><function ID>(optional data length)(optional data)

Fields callsign and function ID are mandatory. Fields optional data length and optional data are not mandatory and may not be present in some frames. Due to the limitations of physical layer packets, total frame length shall be less than or equal to 255 bytes.

3.3 Function IDs

Function ID is the single-byte field following callsign and determines the action satellite should take. In general, function IDs are separated by communication direction (commands vs. responses). Function ID of every response has a value that is by 0x10 larger than that of associated command. The response is sent immediately, where applicable.

Following is a list of function IDs currently implemented in FOSSASAT-1.

Function ID type	Function ID value	Function ID alias	Description
Public commands	0x00	CMD_PING	PING command. Satellite shall respond with RESP_PONG.
	0x01	CMD_RETRANSMIT	Repeater command. Satellite shall respond with RESP_REPEATED_MESSAGE containing the same data that was received as the optional data of the command frame (maximum length is 64 bytes).
	0x02	CMD_RETRANSMIT_CUSTOM	Command to retransmit message with custom configuration (1).
	0x03	CMD_TRANSMIT_SYSTEM_INFO	Command to send system info.
	0x04	CMD_GET_LAST_PACKET_INFO	Command to get information about last received packet.

Function ID type	Function ID value	Function ID alias	Description
Public responses	0x10	RESP_PONG	Response to CMD_PING.
	0x11	RESP_REPEATED_MESSAGE	Response to CMD_RETRANSMIT.
	0x12	RESP_REPEATED_MESSAGE_CUSTOM	Response to CMD_RETRANSMIT_CUSTOM.
	0x13	RESP_SYSTEM_INFO	Response to CMD_TRANSMIT_SYSTEM_INFO. (2)
	0x14	RESP_LAST_PACKET_INFO	Response to CMD_GET_LAST_PACKET_INFO. (3)

- (1) Configuration is sent in the optional data field. Variables larger than a single byte are sent LSB first:

$optionalDataLen = 7 + N$

$optionalData[0]$ = bandwidth value (0x00 for 7.8 kHz, 0x07 for 125 kHz)

$optionalData[1]$ = spreading factor value (0x00 for SF5, 0x07 for SF12)

$optionalData[2]$ = coding rate (0x05 to 0x08, inclusive)

$optionalData[3 - 4]$ = preamble length in symbols (0x0000 to 0xFFFF; LSB first)

$optionalData[5]$ = CRC enabled (0x01) or disabled (0x00)

$optionalData[6]$ = output power in dBm (signed 8-bit integer; -17 to 22)

$optionalData[7 - N]$ = message to be repeated (64 bytes or less)

- (2) System info is sent in the optional data field. Variables larger than a single byte are sent LSB first:

$optionalDataLen = 15$

$optionalData[0]$ = battery charging voltage * 20 mV (unsigned 8-bit integer)

$optionalData[1 - 2]$ = battery charging current * 10 μ A (signed 16-bit integer; LSB first)

$optionalData[3]$ = battery voltage * 20 mV (unsigned 8-bit integer)

$optionalData[4]$ = solar cell A voltage * 20 mV (unsigned 8-bit integer)

$optionalData[5]$ = solar cell B voltage * 20 mV (unsigned 8-bit integer)

$optionalData[6]$ = solar cell C voltage * 20 mV (unsigned 8-bit integer)

$optionalData[7 - 8]$ = battery temperature * 0.01 $^{\circ}$ C (signed 16-bit integer; LSB first)

$optionalData[9 - 10]$ = board temperature * 0.01 $^{\circ}$ C (signed 16-bit integer; LSB first)

$optionalData[11]$ = MCU temperature * 1 $^{\circ}$ C (signed 8-bit integer)

$optionalData[12 - 13]$ = reset counter (unsigned 16-bit integer; LSB first)

$optionalData[14]$ = power control configuration

- (3) Packet information is sent in the optional data field:

$optionalDataLen = 2$

$optionalData[0]$ = SNR * 4 dB

$optionalData[1]$ = RSSI * -2 dBm

3.4 Examples

Following are examples of valid exchanges between ground station and satellite, assuming callsign "FOSSASAT-1". Quote marks not a part of the frame. Byte values representing unprintable ASCII characters are represented by their hexadecimal value in angle brackets. Variables are shown in curly brackets, with the respective size in bytes. Note that this list is by no means exhaustive and only serves to illustrate all sections above.

1. Ping-Pong exchange:

From ground station: "FOSSASAT-1<0x00>"
From satellite: "FOSSASAT-1<0x10>"

2. Repeater exchange:

From ground station: "FOSSASAT-1<0x01><0x0C>Hello World!"
From satellite: "FOSSASAT-1<0x11><0x0C>Hello World!"

3. Repeater exchange with custom configuration:

From ground station:
"FOSSASAT-1<0x02><0x15><0x07><0x0C><0x06><0x20><0x01><0x0F><0x0A>I'm a message!"
From satellite: "FOSSASAT-1<0x0E>I'm a message!" (transmitted with bandwidth 62.5 kHz, spreading factor 12, coding rate 4/7, preamble length 32, with CRC and output power 16 dBm)

4. System info exchange:

From ground station: "FOSSASAT-1<0x03>"
From satellite:
"FOSSASAT-1<0x13><0x0E>{batteryChargingVoltage, 1B}{batteryChargingCurrent, 2B}
{batteryVoltage, 1B}{solarCellAVoltage, 1B}{solarCellBVoltage, 1B}{solarCellCVoltage, 1B}
{batteryTemperature, 2B}{boardTemperature, 2B}{mcuTemperature, 1B}{resetCounter, 2B}
{powerConfig, 1B}"

4 Uploading To The Ground Station Database

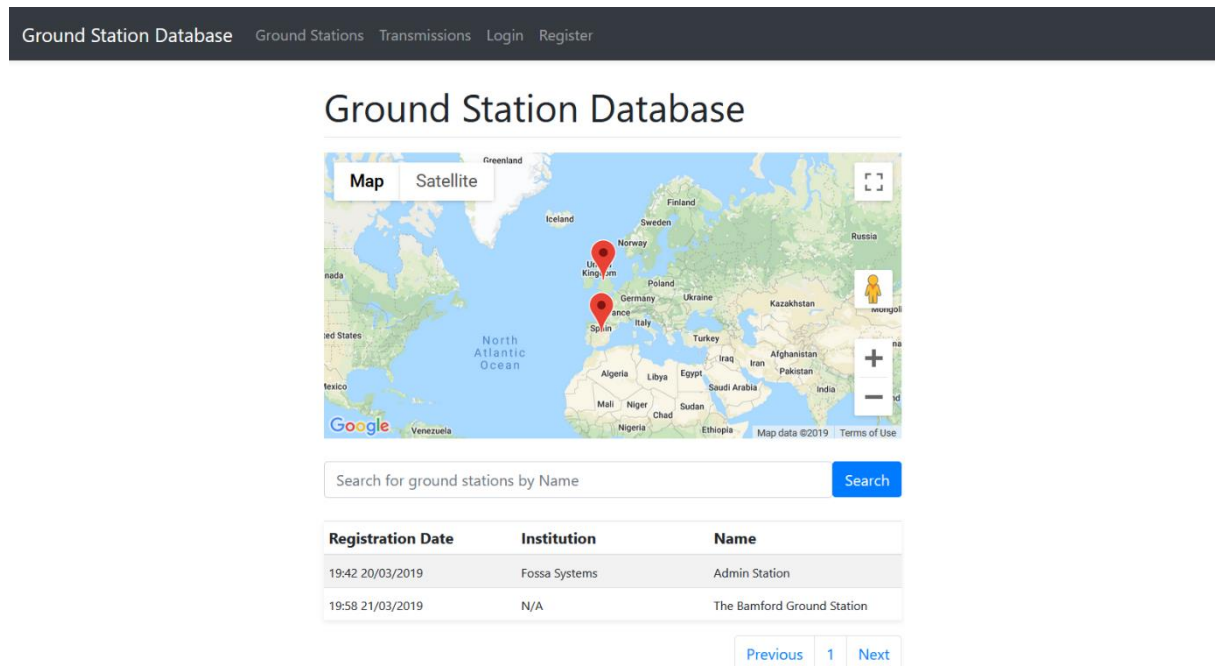
Introduction

The ground station database is a website created by Fossa Systems to help you share your ground station information and received to other radio enthusiasts.

Browsing

All ground stations that are registered appear on the page:

<http://groundstationdatabase.com/database.php>



Registration Date	Institution	Name
19:42 20/03/2019	Fossa Systems	Admin Station
19:58 21/03/2019	N/A	The Bamford Ground Station

Registering

To register your ground station please use the web form found at:

<http://groundstationdatabase.com/registerStation.php>

You will define two things when you register, the ground station you are operating and your own personal information. Your personal information is used to log into this site.

Logging Transmissions

ations Transmissions **Log Transmission** Logout

Transmissions Database

Callsign

Date	Name
20:30 21/03/2019	Admin Station
20:31 21/03/2019	Admin Station
13:13 22/03/2019	The Bamford Ground Station

Click here to enter the transmission logging page.

Log Transmission

Information

Ground Station Name
The Bamford Ground Station

Device Name
SX1278 & Arduino Shield

User Email
rbamfordz@gmail.com

Message

Observations

This is the actual message recieved by your ground station. E.g. FOSSASAT-110;

This is where you can place any observations you note at the time.

5 References

- [1] SX1268 datasheet. *Semtech*. Rev. 1.0 March 2018, online. Available from https://www.semtech.com/uploads/documents/DS_SX1268_V1.0.pdf
- [2] Installing Additional Arduino Libraries. *Arduino*. September 2019, online. Available from <https://www.arduino.cc/en/guide/libraries>