IoT Terminal

UHF modem

Datasheet

info@alen.space
+34 623 186 080
www.alen.space

info@alen.space
+34 623 186 080
www.alen.space

info@alen.space
+34 623 186 080
www.alen.space

1 Changelog

Table 1 - Changelog

Date	Revision	Author	Description
20/07/2020	1.0	ANV	First revision
03/08/2020	1.1	ANV	- Include information about Arduino version - Add electrical characteristics
11/08/2020	1.2	ANV	Update absolute maximum ratings
18/12/2020	1.3	ANV	Update images and ratings
12/08/2021	1.4	ANV	Font change Remove old arduino form factor Update external interfaces Improve API documentation and some bugs fixed Add RF characteristics section Update Physical characteristics section

2 Overview

The IoT terminal is a radio communication device for transmitting and receiving short messages under both Humsat 1.1 and Humsat 2.0 protocols. The IoT terminal is a simple device with a UART interface (3V3) for control and communication and an RF connector to be attached to a matched antenna for radio communications (SMA). The UART interface is the mean of communication on ground, which allows the terminal to communicate with any device, while the RF interface is used to establish a radio link with the satellite(s) to send/received data. In the front panel, 4 LEDs indicate the user the current status of the terminal : ON/OFF, TX, RX and Status.

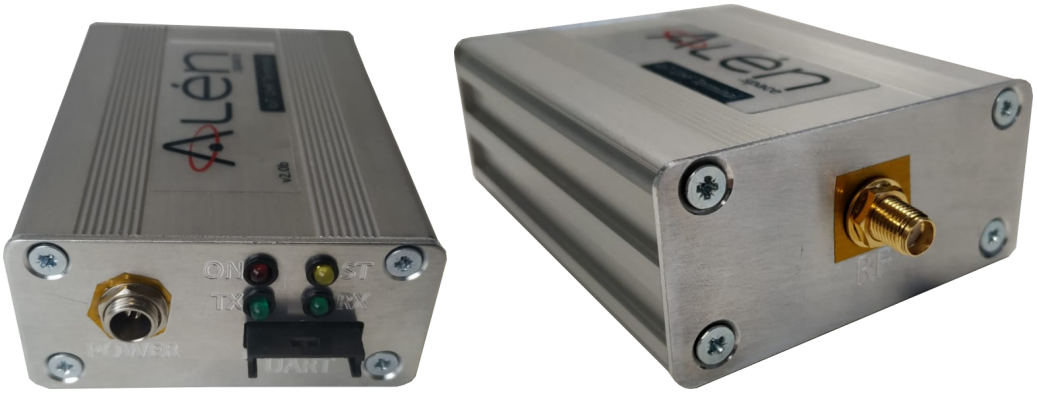


Figure 2.1. IoT terminal version

2.1 Highlighted Features

- 712RA Switchcraft connector: power supply
- UART 2.54 mm pitch:
  - Communication up to 9600 bps
- RF interface:
  - RF Data Rate: 1200bps, 2400bps or 4800bps
  - TX power: 30dBm
  - Sensitivity: -124.5 dBm (GMSK 1200 bps)
  - Modulation: GMSK
  - SMA connector
- 4 x front panel LEDs for status

3 Block diagram

The IoT terminal will act as a slave in a master-slave communication paradigm. According to this, the master will always be the initiator of the communication by sending a series of predefined **commands**. A **response** is always sent by the IoT terminal to the node (master).

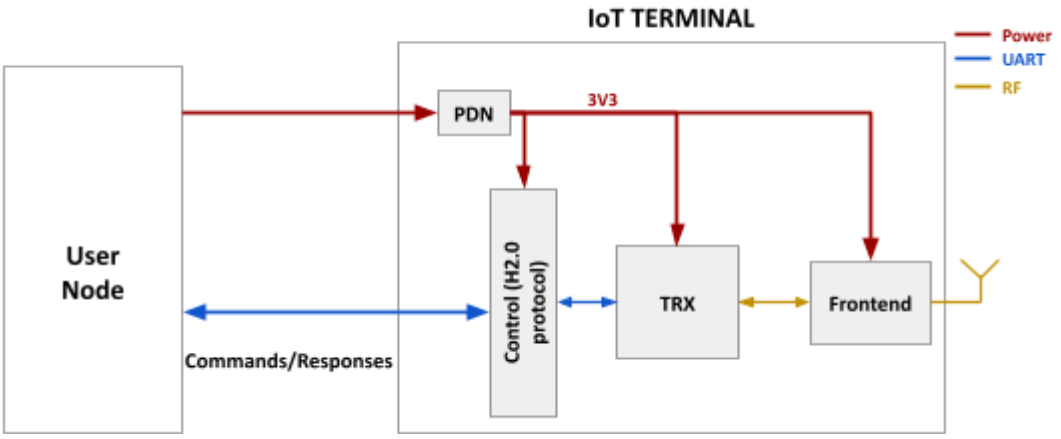


Figure 3.1 IoT terminal architecture

3.1 Power Distribution network (PDN)

The power distribution network is in charge of both converting the input voltage and delivering the power needs of the different parts from the input power connector.

3.2 Control

The control unit is in charge of interacting with the user and controlling the transceiver. It receives high-level commands from the user and translates them into a set of internal commands to control the transceiver.

3.3 Transceiver

The main purpose of the transceiver is to implement the physical layer of the Humsat protocol. In transmission, it is in charge of modulating and upconverting the signal to the desired frequency channel. In reception, the transceiver downconverts and demodulates the signal received and extracts the data to be sent to the user.

3.4 Frontend

The RF frontend connects the antenna with the transceiver. A power amplifier boosts the signal to the desired output power while the LNAs accommodate the received signal for demodulation. An antenna switch is included to change from transmission mode to reception mode.

4 Communication

The IoT terminal implements different layers for proper communication. On top, the **Terminal to User Interface** (TUI) manages the user commands, **Uart Master-Slave** protocol (UMS) coordinates the communication between master and slave and **KISS** protocol defines the packet structure.

Figure 4.1 UARTCommunication layers

- Bit rate : 9600 bps
- Parity : None
- Stop bits : One
- ByteSize : 8

4.1 TUI - Terminal to User Interface

This protocol is designed for the communication between the user (master) and the terminal (slave). The user interacts with the terminal by sending commands and receiving responses from the terminal. It is through these high-level commands where the user can operate the terminal, send and receive data. The RF link is established automatically by the terminal and is transparent to the user.

4.2 UMS

The Uart Master-Slave protocol is designed to coordinate the communication between the master (user) and the slave (terminal). Since the master is always the initiator of the communications, it always needs to wait for a slave response before sending the next command.

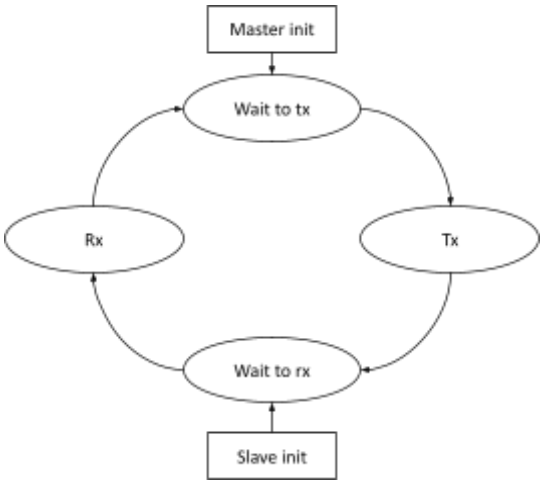


Figure 4.2. UMS diagram

4.3 KISS

The node will transmit/receive commands/responses using the **KISS protocol**. Its only purpose is to delimit the frames. The KISS frame is formed by the TUI commands/responses (payload) and special characters that are appended at start and end (KISS FEND char). In addition, a CRC32[1] is performed to the payload. Table X shows the KISS special characters, including escape ones. For more information, check the following link : https://en.wikipedia.org/wiki/KISS_(TNC)

[1] CRC32 is computed as a result of the data + TUI command code.

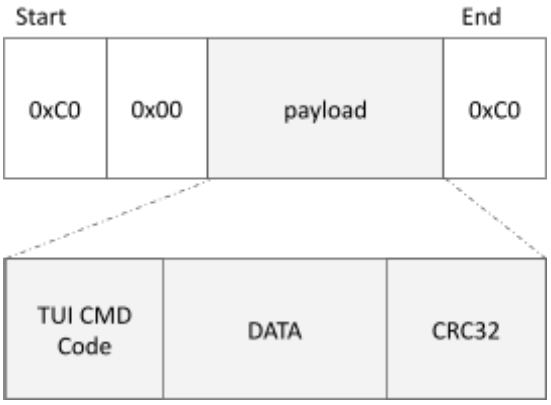


Figure 4.3 KISS frame structure

Table 2 - KISS special characters

Abbreviation	Description	Hex value
FEND	Frame END	0xC0
FESC	Frame Escape	0xDB
TFEND	Transposed Frame END	0xDC
TFESC	Transposed Frame Escape	0xDD

5 Operational modes

The IoT terminal has two different operational modes : STAND BY and COMMUNICATION.

5.1 Stand By

- Default mode. The IoT terminal enters this mode after power ON
- Any change in configuration ('set' commands) can only be made in this operational mode
- Once the IoT terminal is properly configured, the user is allowed to change to "Communications" mode by sending the StartOperation command

5.2 Communications

- This is the mode in which the IoT terminal can receive and send messages through the RF interface
- The user cannot change any configuration in this mode.
- Enter this mode with StartOperation from Stand By mode
- Once in Communications mode, reception and transmission are enabled with the configuration provided
- Use the Set Transmission Packet command to send a packet. The terminal will send it to the satellite once synchronized.
- Use Get Received Packet command to read packets received from the satellite

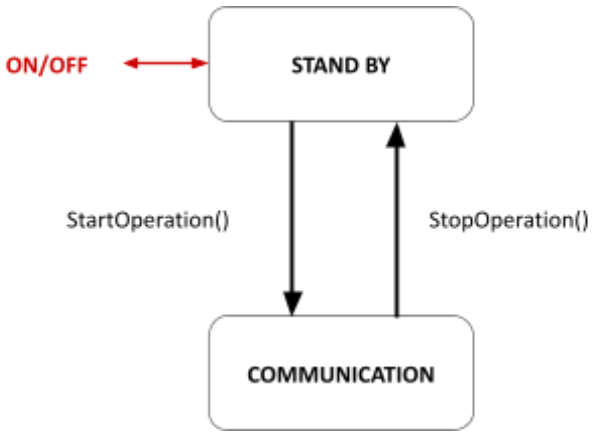


Figure 5.1. Operational modes

6 External interfaces

6.1 Front panel

The front panel includes the 712RA power connector, the UART interface and 4 Leds

Figure 6.1: Front panel

6.1.1 Switchcraft 712RA connector

The power connector is a 712RA from switchcraft that allows the terminal to be powered according to recommended electrical specifications (center pin is VCC)

Figure 6.2: Power connector.

6.1.2 UART connector

P2- TE-Connectivity 280379-2	
Pin	**Description**
1	Ground
2	NC
3	NC
4	RX (Terminal receive)[1]
5	TX (Terminal transmit)[1]
6	NC

Note 1: UART signal level must be 3V3, otherwise the IoT terminal can be damaged

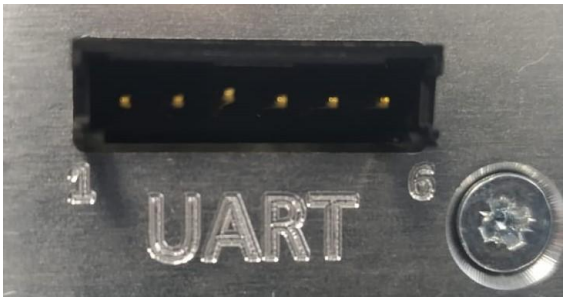


Figure 6.3: Front panel

6.1.3 LEDs description

- ON/OFF : A red LED is ON when the IoT terminal is powered up
- TX : Green LED toggles when transmitting a message (RF interface)
- RX : Green LED is ON when the IoT terminal is in communications mode
- Status: Yellow LED is ON when the IoT terminal status has changed.

6.2 Rear panel

The rear panel has a SMA connector (RF port).

Warning! Always place an antenna or a 50 ohm load (>1W) when using the terminal, otherwise it may be damaged

Figure 6.4 Rear panel

7 Absolute maximum ratings

Table 5 - Absolute maximum ratings

Symbol	Description	Min	Typ	Max	Unit
Vin	Supply voltage	4.5	5	5.25	V
Pin	Supply power draw	-		3.8	W
Imax	Maximum supply current	0.55	0.75	1	A
Tº	Operating temperature range	-30	-	+85	ºC

8 Electrical characteristics

Table 6 - Electrical characteristics

Symbol	Description	Min	Typ	Max	Unit
Pstby	Power consumption, STAND BY	-	235	260	mW
Prx	Power consumption, RX mode	-	360	390	mW
Ptx	Power consumption, TX mode	-	3350	3800	mW

9 RF characteristics

Table 7 - RF characteristics

Symbol	Description	Min	Typ	Max	Unit
TX_freq[1]	Transmission frequency range	395	-	410	MHz
RX_bw1[1]	Reception bandwidth	395	-	410	MHz
TX_freq2[1]	Transmission frequency	430	-	440	MHz
RX_bw2[1]	Reception bandwidth	430	-	440	MHz
Pout	RF output power	-	30	30.5	dBm
RX_sns[2]	Sensitivity	-	-124.5	-	dBm

Note 1: Frequency band must be selected when ordered. Ask for information if other frequency ranges are needed.

Note 2: Reception sensitivity of the transceiver with Reed Solomon and GMSK 1200 bps

10 Physical characteristics

Table 7- Physical characteristics

Description	Value	Unit
Total mass	140	g

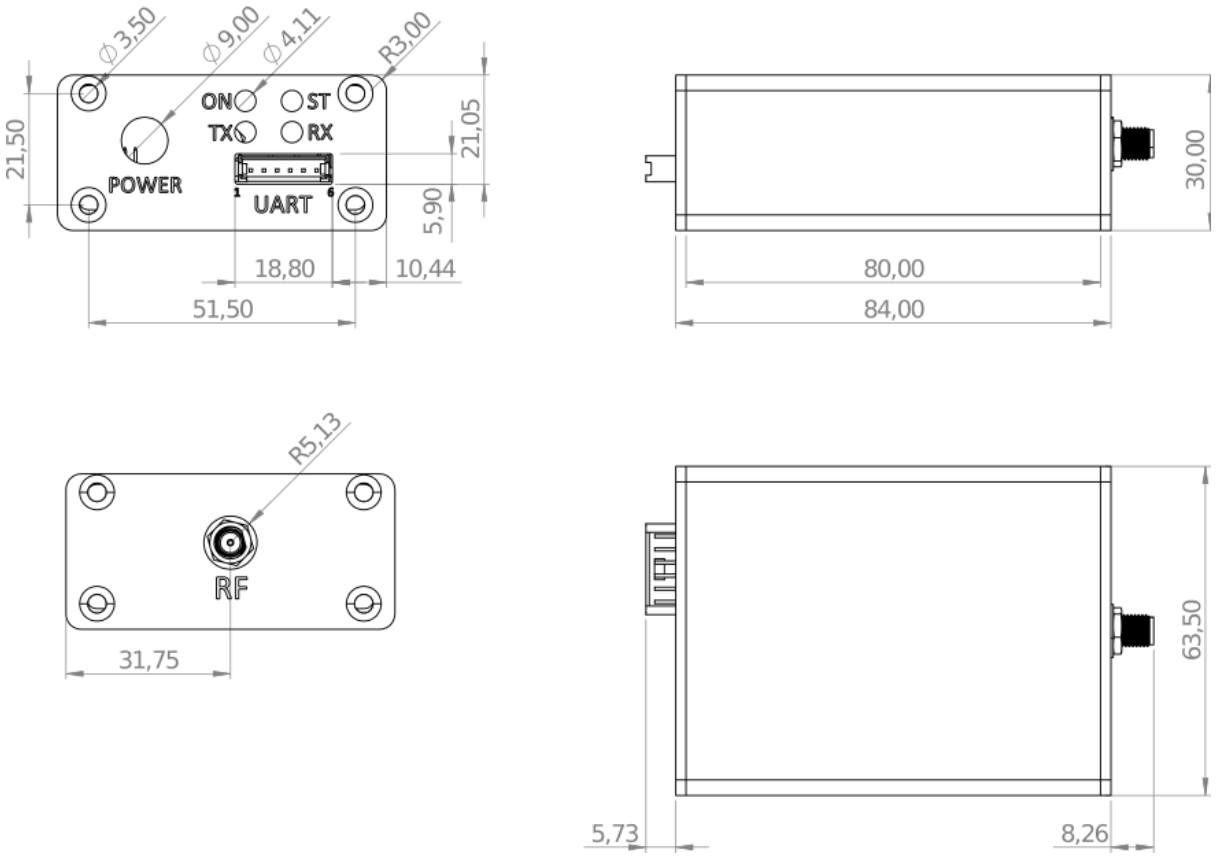


Figure 9.1 Physical dimensions (mm)

.

11 Annex A: API

11.1 TUI Master commands

Command	Code	Data	Response
Start Operation	0x10	-	OK/NOK
Stop Operation	0x11	-	OK/NOK
Set Transmission packet	0x20	TX Packet	OK/NOK
Get Received packet	0x30	-	Received packet
Set User Configuration	0x40	User configuration	OK/NOK
Get User Configuration	0x50	-	User configuration
Set Group List	0x80	Group list	OK/NOK
Get Group List	0x90	-	Group list
Get Statistics	0xA0	-	Statistics
Get Status	0xB0	-	Status

11.2 TUI Slave responses

Command	Code	Data	From command
Received packet	0x31	Packet length and packet	Get Received packet
User Configuration	0x51	Current configuration	Get User configuration
Group List	0x91	Group list	Get group list
Statistics	0xA1	Statistics	Get Statistics
Status	0xB2	Status	Get Status
OK	0x01	-	Others
NOK	0x0F	Error code	Others

11.3 Master data

TX Packet			
Parameter	**Description**	**Value**	**Length**
Destination ID	Storage and forward	0x**3**0000000 (hex)	4B
	Data relay to group	0x**2**YYYYYYY (hex)[1]	4B
	Data relay to terminal	0x**1**YYYYYYY (hex)[1]	4B
Data length	Length of the data to be sent	length(USER_DATA)	1B
TX packet	User data to be transmitted	USER_DATA	Max = 192B
Total Length			**197 bytes**

[1]Where "Y" needs to be substituted with the destination ID

User Configuration			
Parameter	**Description**	**Value**	**Length**
My area	ID of your working area on Earth (check payload configuration)	0xYYYY	2B
TDMA Slot	Assigned TDMA slot (check payload configuration)	0xYYYY Aloha mode: 0xFFFF	2B
Terminal priority	Priority assigned to the terminal (check payload configuration)	0xYY Priority levels	1B
Terminal type ID	ID of terminal	0x1YYYYYYY[1]	4B
Reserved	N/A	0x00	1B
Reserved	N/A	0x00	1B
Reserved	N/A	0x00	1B
Total Length			**12B**

[1]Where "Y" needs to be substituted with own ID

Group list			
Parameter	**Description**	**Value**	**Length**
Number of groups	Number of groups to be stored	min = 0x01, max = 0x05	1B
Group 1	ID of the group	0x2YYYYYYY[1]	4B
Group 2	ID of the group	0x2YYYYYYY[1]	4B
Group 3	ID of the group	0x2YYYYYYY[1]	4B
Group 4	ID of the group	0x2YYYYYYY[1]	4B
Group 5	ID of the group	0x2YYYYYYY[1]	4B
Total Length			**21B**

[1]Where "Y" needs to be substituted with the destination ID

11.4 Slave Responses

OK/NOK Responses			
Command	**Source**	**Code**	**Data**
OK	Slave	0x01	N/A
NOK	Slave	0x0F	Error code[1]

[1] See TUI NOK error code lists

RX Packet				
Command	**Source**	**Data**		
		Parameter	**Description**	**Length**
Received packet	Slave	Source ID	Packet source ID	4B
		Destination ID	Packet destination ID	4B
		Packet Length	Length of data (next field)	1B
		Data	Data received	Max = 192B

Statistics					
Command	**Source**	**Data**			
		Parameter	**Description**	**Type**	**Length**
Statistics	Slave	state	RF state : 0 (STBY), 1 (TX), 2 (RX)	byte	1B
		error	Error code	byte	1B
		rx_pkts_buff	Received packets in buffer	byte	1B
		total_rx_pkts	Total rx packets in this session	uint	2B
		total_tx_pkts	Total tx packets in this session	uint	2B
		total_rx_pkts_last	Total rx packets in last RX state	uint	2B

		rx_error_pkts_last	Number of error packets rx in last reception	uint	2B
		freq_dev_last_rx	Maximum frequency deviation in last reception (Hz)	int	4B
		max_pwr_last_rx	Maximum power received in last reception (dBm)	int	2B

Status					
Command	**Source**	**Data**			
		Parameter	**Description**	**Type**	**Length**
Status	Slave	Current Operational Mode	0 : Stand by 1 : Communication	char	1B
		Proto Error	Proto Error code[1]	char	1B
		Modem Error	Modem Error code[1]	char	1B
		Modem NOK	Modem NOK code[1]	char	1B
		Packet received	0 : No packet received 1 : Packet received	char	1B
		Last TX packet status	0 : Packet not tx yet 1 : Packet tx	char	1B

[1] See Status error code lists

11.5 TUI NOK codes

TUI NOK Codes		
HEX code	**Layer**	**Description**
0x00	TUI	No ERROR
0xD8	TUI	Error command length
0xD7	TUI	Error command code
0xD6	TUI	Error command CRC

0xD4	TUI	Error TX packet length
0xD3	TUI	Error RX packer length

11.6 Status error code list

PROTO error Codes		
HEX code	**Layer**	**Description**
0x00	COMCON	No ERROR
0xFF	KISS	Error no data
0xFE	KISS	Error not empty
0xFD	UMS	Error state
0xFC	COMCON	Error response length received
0xFB	COMCON	Error response code received
0xFA	COMCON	Error response CRC received
0xF8	COMCON	Error Transmission packet length
0xF7	COMCON	Error Parameters
0xEC	RF_IF	Error wrong command
0xEB	RF_IF	Error wrong RF state
0xEA	RF_IF	Error modems state
0xE9	RF_IF	Error response packet
0xE8	RF_IF	Error wrong modem
0xE7	RF_IF	Transmission OFF
0xE6	RF_IF	Modem NOK received
0xE5	RF_IF	Error unknown
0xE2	PROTO	Error mode

0xE0	PROTO	Error state
0xDE	PROTO	No packet received
0xF3	FRAME	Header fail
0xEF	FRAME	FEC option wrong
0xEE	FRAME	FEC CRC fail
0xED	FRAME	FEC RS fail

MODEM Error Codes	
HEX code	**Description**
0x00	No ERROR
0x01	UMS State
0x02	Received packet in wrong state
0x04	AXRADIO No chip found
0x08	AXRADIO Ranging error
0x10	Radio state

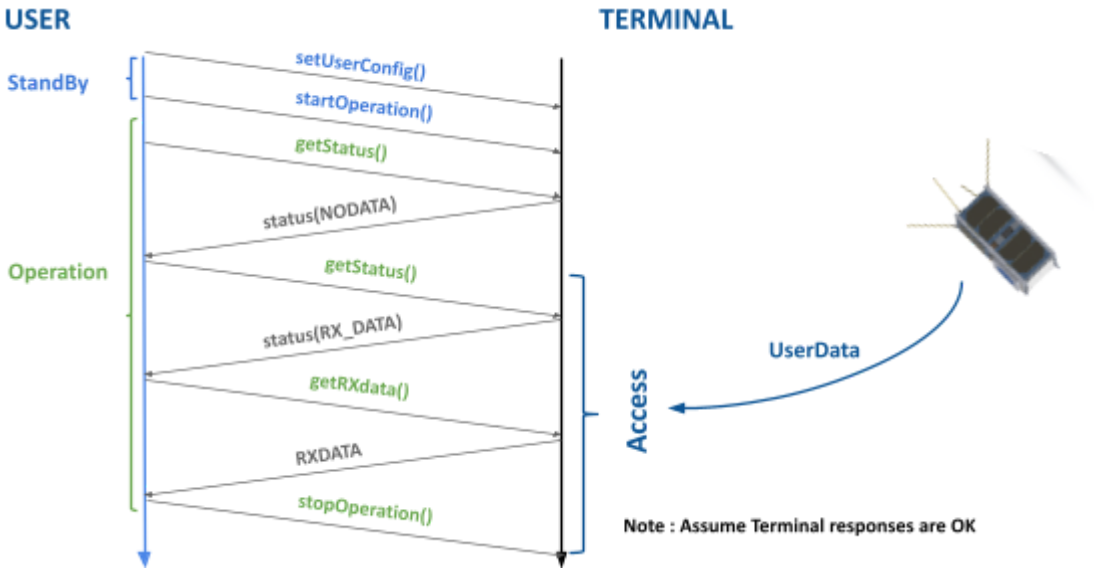
info@alen.space
+34 623 186 080
www.alen.space

MODEM NOK Codes	
HEX code	**Description**
0x01	AXRADIO - Not supported
0x02	AXRADIO - Busy
0x04	AXRADIO - Invalid
0x05	AXRADIO - No chip
0x06	AXRADIO - Ranging error
0x07	Error command CRC
0x08	Error command length
0x09	Unknown command
0x0A	Error parameters
0x0B	Operation not valid in this state

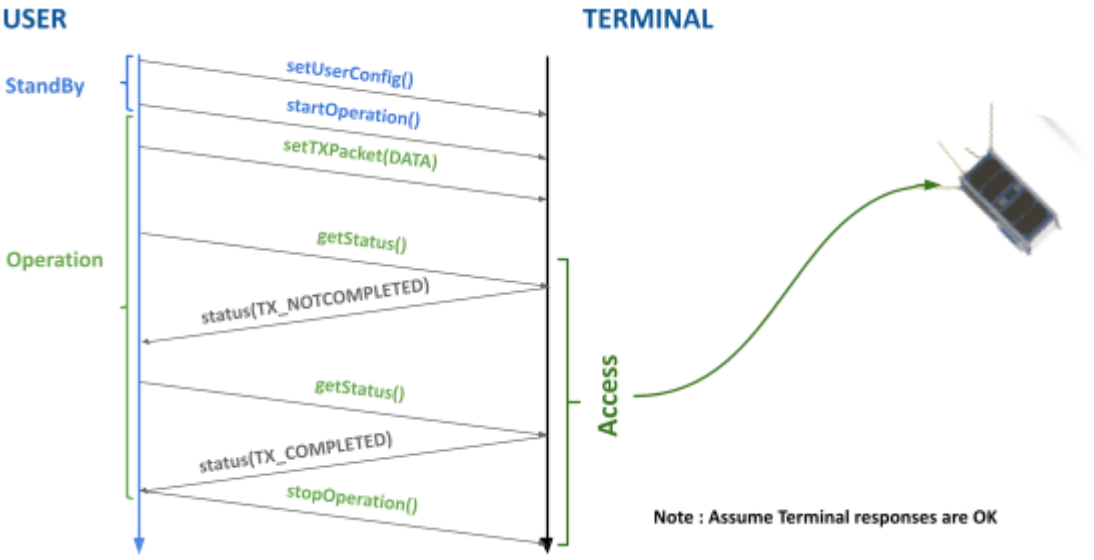
info@alen.space
+34 623 186 080
www.alen.space

12 Annex B : Scenarios examples

12.1 RX scenario



12.2 TX scenario



info@alen.space
+34 623 186 080
www.alen.space