RE001 Requirements for the Module codecs_lib.cobs

Conventions

Requirements listed in this document are constructed according to the following structure:

Requirement ID: REQ-UVW-XYZ

Title: Title / name of the requirement

Description: Descriprion / definition of the requirement

Verification Method: I / A / T / D

The requirement ID starts with the fixed prefix 'REQ'. The prefix is followed by 3 letters abbreviation (in here 'UVW'), which defines the requiement type - e.g. 'FUN' for a functional and capability requirement, 'AWM' for an alarm, warnings and operator messages, etc. The last part of the ID is a 3-digits *hexadecimal* number (0..9|A..F), with the first digit identifing the module, the second digit identifing a class / function, and the last digit - the requirement ordering number for this object. E.g. 'REQ-FUN-112'. Each requirement type has its own counter, thus 'REQ-FUN-112' and 'REQ-AWN-112' requirements are different entities, but they refer to the same object (class or function) within the same module.

The verification method for a requirement is given by a single letter according to the table below:

Term	Definition
Inspection (I)	Control or visual verification
Analysis (A)	Verification based upon analytical evidences
Test (T)	Verification of quantitative characteristics with quantitative measurement
Demonstration (D)	Verification of operational characteristics without quantitative measurement

Functional and capability requirements

Requirement ID: REQ-FUN-100

Title: COBS implementation

Description: The module must provide a proper implementation of the COBS algorithm for encoding and decoding without concern for the package delimiters.

Verification Method: A

Requirement ID: REQ-FUN-101

Title: COBS interface

Description: The module should provide two functions or class methods: one for encoding of the data and one for decoding of the data

Verification Method: T

Requirement ID: REQ-FUN-110

Title: Data encoding

Description: The function / method for encoding of the data should accept an arbitrary byte string (type **bytes**) or byte array (type **bytearray**) object as its argument, which can contain any number of the zero characters ('\x00'), and return another byte string, which is the COBS encoded data using srict definition of the alforithm without any delimiters added, i.e. it will not contain any zero characters.

Verification Method: T

Requirement ID: REQ-FUN-120

Title: Data decoding

Description: The function / method for decoding of the data should accept an arbitrary byte string (type **bytes**) or byte array (type **bytearray**) object as its argument, remove all leading and tailing zero characters ('\x00'), and return another byte string, which is the COBS decoded data using strict definition of the algorithm without any delimiters added. Note that a zero character inside the string to be decoded (not in the leading / tailing position) is not allowed as a violation of the encoding algorithm.

Verification Method: T

Alarms, warnings and operator messages

Requirement ID: REQ-AWM-100

Title: Improper input type for encoding or decoding raises an exception

Description: The passed argument is neither a byte string nor a byte array. **TypeError** exception or its sub-class must be raised.

Verification Method: T

Requirement ID: REQ-AWM-120

Title: Improper input value for decoding raises an exception

Description: A zero character inside the string to be decoded (not in the leading / tailing position) is not allowed as a violation of the encoding algorithm. **ValueError** exception or its sub-class must be raised.

Verification Method: T