NMEA Namespace Reference

Functions

bool	**isWellFormedSentence** (string NMEAString) This function determines whether the parameter is a well-formed **NMEA** sentence A **NMEA** sentence contains the following contents: . More...
bool	**hasValidChecksum** (string NMEAString) This function verifies whether a sentence has a valid checksum. To be valid, the checksum value should equal the XOR reduction of the character codes of all characters between the '$' and the '*' (exclusive). . More...
SentenceData	**extractSentenceData** (string NMEAString) This function extracts the sentence format and the field contents from a **NMEA** sentence. The '$GP' and the checksum are ignored. . More...
GPS::Position	**positionFromSentenceData** (SentenceData sentenceData) This function computes a Position from **NMEA** Sentence Data. Currently only supports the GLL, GGA and RMC sentence formats. . More...
Route	**routeFromLog** (std::istream &ss) This function reads a stream of **NMEA** sentences (one sentence per line), and constructs a route, ignoring any lines that do not contain valid sentences. A sentence is valid if: . More...

Detailed Description

This application has been created to read and validate **NMEA** sentences
It does this by extracting the data from the sentence and ensuring the format and checksum are suitable.

Function Documentation

◆ extractSentenceData()

bool NMEA::extractSentenceData (string NMEAString)

This function extracts the sentence format and the field contents from a **NMEA** sentence.
The '$GP' and the checksum are ignored.
.

Pre-condition

The parameter is a well-formed **NMEA** sentence.

Parameters

NMEAString

Returns

The sentence data is returned

◆ hasValidChecksum()

bool NMEA::hasValidChecksum (string NMEAString)

This function verifies whether a sentence has a valid checksum.
To be valid, the checksum value should equal the XOR reduction of the character codes of all characters between the '$' and the '*' (exclusive).
.

Pre-condition

The parameter is a well-formed **NMEA** sentence.

Parameters

NMEAString

Returns

'true' if the checksum calculated matches the checksum provided

◆ isWellFormedSentence()

bool NMEA::isWellFormedSentence (string NMEAString)

This function determines whether the parameter is a well-formed **NMEA** sentence

A **NMEA** sentence contains the following contents:
.

- The prefix "$GP"
- Followed by a sequence of three (English) alphabet characters identifying
- The sentence format
- Followed by a sequence of comma-separated data fields
- Followed by a '*' character \
- Followed by a two-character hexadecimal checksum
- The '$' and '*' characters are reserved, and may not appear in the data fields

For ill-formed sentences, this function returns false (it does not throw an exception or terminate the program)

Parameters

NMEAString

Returns

'true' if all the criteria is met

◆ positionFromSentenceData()

GPS::Position NMEA::positionFromSentenceData (SentenceData sentenceData)

This function computes a Position from **NMEA** Sentence Data.
Currently only supports the GLL, GGA and RMC sentence formats.
.

Exceptions

Throws an invalid argument exception for unsupported sentence formats, or if the necessary data fields are missing or contain invalid data.

Parameters

SentenceData sentenceData

Returns

The position data from the **NMEA** sentence data

◆ routeFromLog()

Route NMEA::routeFromLog (std::istream & ss)

This function reads a stream of **NMEA** sentences (one sentence per line), and constructs a route, ignoring any lines that do not contain valid sentences.
A sentence is valid if:
.

- It is a well-formed **NMEA** sentence
- The checksum is valid
- The sentence format is supported (currently GLL, GGA and RMC)
- The necessary data fields are present and contain valid data

Parameters

std::istream &ss

Returns

The valid route positions in a vector