NMEA Library

Generated by Doxygen 1.8.4

Tue Jul 15 2014 14:41:33

Contents

Chapter 1

NMEA

Arduino library for decoding and parsing NMEA GPS messages

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 NMEA Class Reference

Public Member Functions

- NMEA ()
- NMEA (Stream &dev)
- void begin ()
- void process ()
- double getLatitude ()
- double getLongitude ()
- double getBearing (bool mag=false)
- double getSpeed (bool knots=false)
- double getAltitude ()
- char getAltitudeUnits ()
- double getEllipsoidHeight ()
- char getEllipsoidHeightUnits ()
- uint8_t getSatellites ()
- uint8_t getDay ()
- uint8_t getMonth ()
- uint16_t getYear ()
- uint8_t getHour ()
- uint8_t getMinute ()
- uint8_t getSecond ()
- bool isLocked ()
- bool isUpdated ()

3.1.1 Constructor & Destructor Documentation

3.1.1.1 NMEA::NMEA ( )

Default constructor

This constructor defaults the NMEA parser to using the Serial object.

3.1.1.2 NMEA::NMEA (Stream & *dev*)

Device specific constructor

This allows you to pass a specific serial object to the parser. It is your responsibility to ensure that the serial object is configured for 38400 baud (or the baud of your GPS module) and has had "begin" called on it.

3.1.2 Member Function Documentation

3.1.2.1 void NMEA::begin ( )

Initialize the processor

Pre-configures any required variables. Calling this before any processing is done is required.

3.1.2.2 double NMEA::getAltitude ( )

getAltitude()

Returns the current height above sea level. The units are not defined, but can be obtained with the getAltutude-Units() function.

3.1.2.3 char NMEA::getAltitudeUnits ( )

getAltitudeUnits()

Returns the units used for the height above sea level. Usually 'M' for meters.

3.1.2.4 double NMEA::getBearing (bool *mag* = `false`)

getBearing(magnetic)

Returns the current calculated bearing or heading. If true is passed as a parameter it returns the bearing to magnetic north. If false is passed, or no parameter is used, it returns the bearing to true north.

3.1.2.5 uint8_t NMEA::getDay ( )

getDay()

Returns the current day of the month (1 ... 31).

3.1.2.6 double NMEA::getEllipsoidHeight ( )

getEllipsoidHeight()

Returns the height above a WGS84 ellipsoid. The units are not defined, but can be obtained with the getEllipsoid-HeightUnits() function.

3.1.2.7 char NMEA::getEllipsoidHeightUnits ( )

getEllipsoidHeightUnits()

Returns the units used for the height above a WGS84 ellipsoid. Usually 'M' for meters.

3.1.2.8 uint8_t NMEA::getHour ( )

getHour()

Returns the current hour of the day (0 ... 23).

3.1.2.9 double NMEA::getLatitude ( )

getLatitude()

Returns the current latitude in degrees

3.1.2.10 double NMEA::getLongitude ( )

getLongitude()

Returns the current longitude in degrees

3.1.2.11 uint8_t NMEA::getMinute ( )

getMinute()

Returns the current minutes (0 ... 59).

3.1.2.12 uint8_t NMEA::getMonth ( )

getMonth()

Returns the current month number (1 ... 12).

3.1.2.13 uint8_t NMEA::getSatellites ( )

getSatellites()

Returns the number of currently locked satellites.

3.1.2.14 uint8_t NMEA::getSecond ( )

getSecond()

Returns the current seconds (0 ... 59).

3.1.2.15 double NMEA::getSpeed (bool *knots* = `false`)

getSpeed(knots)

Returns the current calculated speed. If true is passed as a parameter it returns the speed in Knots. If false is passed, or no parameter is used, it returns the speed in kilometers per hour.

3.1.2.16 uint16_t NMEA::getYear ( )

getYear()

Returns the current year (2000 ... 2099).

3.1.2.17 bool NMEA::isLocked ( )

isLocked()

Returns true if the receiver is locked on, false otherwise.

3.1.2.18 bool NMEA::isUpdated ( )

isUpdated()

Returns true if the processor has received and processed a new valid message. Resets the updated flag internally.

3.1.2.19 void NMEA::process ( )

Process incoming messages

This function is the main heart of the NMEA processor. It receives characters from the serial device, identifies the frame wrapper characters, and stores the frame data into a buffer. When the frame is complete it calls the relevant decoder function to handle the data.

This function must be called frequently to process the data.

The documentation for this class was generated from the following files:

- NMEA.h
- NMEA.cpp

Index