NMEA 0183

NMEA 0183 is a combined electrical and data specification for communication between marine electronics such as echo sounder, sonars, anemometer, gyrocompass, autopilot, GPS receivers and many other types of instruments. It has been defined by, and is controlled by, the National Marine Electronics Association. It replaces the earlier **NMEA 0180** and **NMEA 0182** standards.[1] In marine applications, it is slowly being phased out in favor of the newer NMEA 2000 standard.

The electrical standard that is used is EIA-422, although most hardware with NMEA-0183 outputs are also able to drive a single EIA-232 port. Although the standard calls for isolated inputs and outputs, there are various series of hardware that do not adhere to this requirement.

The NMEA 0183 standard uses a simple ASCII, serial communications protocol that defines how data are transmitted in a "sentence" from one "talker" to multiple "listeners" at a time. Through the use of intermediate expanders, a talker can have a unidirectional conversation with a nearly unlimited number of listeners, and using multiplexers, multiple sensors can talk to a single computer port.

At the application layer, the standard also defines the contents of each sentence (message) type, so that all listeners can parse messages accurately.

Contents

Serial configuration (data link layer)

Typical Baud rate	4800
Data bits	8
Parity	None
Stop bits	1
Handshake	None

There is a variation of the standard called NMEA-0183HS that specifies a baud rate of 38,400. This is in general use by AIS devices.

Message structure

- All transmitted data are printable ASCII characters between 0x20 (space) to 0x7e (~)
- Data characters are all the above characters less the reserved characters (See next line)

- Reserved characters are used by NMEA0183 for the following uses:

ASCII	Hex	Dec	Use
<CR>	0x0d	13	Carriage return
<LF>	0x0a	10	Line feed, end delimiter
!	0x21	33	Start of encapsulation sentence delimiter
$	0x24	36	Start delimiter
*	0x2a	42	Checksum delimiter
,	0x2c	44	Field delimiter
\	0x5c	92	TAG block delimiter
^	0x5e	94	Code delimiter for HEX representation of ISO/IEC 8859-1 (ASCII) characters
~	0x7e	126	Reserved

- Messages have a maximum length of 82 characters, including the $ or ! starting character and the ending <LF>
- The start character for each message can be either a $ (For conventional field delimited messages) or ! (for messages that have special encapsulation in them)
- The next five characters identify the talker (two characters) and the type of message (three characters).
- All data fields that follow are comma-delimited.
- Where data is unavailable, the corresponding field remains blank (it contains no character before the next delimiter - see Sample file section below).
- The first character that immediately follows the last data field character is an asterisk, but it is only included if a checksum is supplied.
- The asterisk is immediately followed by a checksum represented as a two-digit hexadecimal number. The checksum is the bitwise exclusive OR of ASCII codes of all characters between the $ and *. According to the official specification, the checksum is optional for most data sentences, but is compulsory for RMA, RMB, and RMC (among others).
- <CR><LF> ends the message.

As an example, a waypoint arrival alarm has the form:

    *$GPAAM,A,A,0.10,N,WPTNME*32*

Another example for AIS messages is:

    !AIVDM,1,1,,A,14eG;o@034o8sd<L9i:a;WF>062D,0*7D

The new standard, NMEA 2000, accommodates several *talkers* at a higher baud rate, without using a central hub, or round-robin packet buffering.

The NMEA standard is proprietary and sells for at least US$250 (except for members of the NMEA) as of November 2017.[2] However, much of it has been reverse-engineered from public sources.[3][4]

Vendor extensions

Most GPS manufacturers include special messages in addition to the standard NMEA set in their products for maintenance and diagnostics purposes. Extended messages begin with "$P". These extended messages are not standardized.

Software compatibility

NMEA 0183 is supported by various navigation and mapping software. Notable applications include:

- DeLorme Street Atlas
- ESRI
- Google Earth

- Google Maps Mobile Edition[5]
- gpsd - Unix GPS Daemon
- JOSM - OpenStreetMap Map Editor
- MapKing
- Microsoft MapPoint
- Microsoft Streets & Trips
- NetStumbler
- OpenCPN - Open source navigation software
- Rand McNally StreetFinder

Sample file

A sample file produced by a Tripmate 850 GPS logger. This file was produced in Leixlip, County Kildare, Ireland. The record lasts two seconds.

```
$GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,1.03,61.7,M,55.2,M,,*76
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,19,13,28,070,17,26,23,252,,04,14,186,14*79
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092750.000,A,5321.6802,N,00630.3372,W,0.02,31.66,280511,,,A*43
$GPGGA,092751.000,5321.6802,N,00630.3371,W,1,8,1.03,61.7,M,55.3,M,,*75
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,16,13,28,070,17,26,23,252,,04,14,186,15*77
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092751.000,A,5321.6802,N,00630.3371,W,0.06,31.66,280511,,,A*45
```

Note some blank fields, for example:

- **GSV** records, which describe satellites 'visible', lack the SNR (signal–to–noise ratio) field for satellite 16 and all data for satellite 36.
- **GSA** record, which lists satellites used for determining a fix (position) and gives a DOP of the fix, contains 12 fields for satellites' numbers, but only 8 satellites were taken into account—so 4 fields remain blank.

C implementation of checksum generation

The checksum at the end of each sentence is the XOR of all of the bytes in the sentence, excluding the initial dollar sign. The following C code generates a checksum for the string entered as "mystring" and prints it to the output stream. In the example, a sentence from the sample file is used.

```c
#include <stdio.h>

int checksum(const char *s) {
    int c = 0;

    while(*s)
        c ^= *s++;

    return c;
}

int main()
{
    char mystring[] = "GPRMC,092751.000,A,5321.6802,N,00630.3371,W,0.06,31.66,280511,,,A" ;

    printf("String: %s \nChecksum: 0x%02X \n", mystring, checksum(mystring));

    return 0;
}
```

Status

NMEA 0183 continued to be maintained separately: V4.10 was published in early May 2012, and an erratum noted on 12 May 2012.[6] It is not clear whether there is any active development. The latest NMEA announcement on the subject is older than the V4.10 standard.[7]

See also

- GPS Exchange Format
- TransducerML
- IEEE 1451
- IEC 61162
- NMEA 2000
- NMEA OneNet

References

1. Peter Bennett (Sep 15, 1997). "The NMEA FAQ" (https://web.archive.org/web/20140215150802/http://www.kh-gps.de/nmea.faq). Archived from the original (http://www.kh-gps.de/nmea.faq) on 2014-02-15. Retrieved 2013-10-21.
2. "Publications and Standards from the National Marine Electronics Association (NMEA) / NMEA 0183" (http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp). NMEA. November 2008. Retrieved 2013-10-21.
3. Raymond, Eric S. (January 2016). "NMEA Revealed" (http://www.catb.org/gpsd/NMEA.html)
4. Dale DePriest. "NMEA data" (http://www.gpsinformation.org/dale/nmea.htm). Retrieved 2013-10-21.
5. "Google Maps for Android" (http://www.google.com/mobile/default/maps.html). Google. Retrieved 2013-10-21.
6. "ERRATA # 0183 120512" (http://www.nmea.org/Assets/0183120512%20spo_errata.pdf) (PDF). NMEA. Retrieved 2013-10-21.
7. "NMEA Standards Update March 2012" (http://www.nmea.org/Assets/20120315%20nmea%20standards%20update%20march%202012.pdf) (PDF). NMEA. March 2012. Retrieved 2013-10-21.

External links

- National Marine Electronics Association
- NMEA's website about NMEA 0183
- NMEA data information
- Sirf-Chipset NMEA data reference manual
- NMEA Specifications at APRS Info
- MTK proprietary NMEA extensions
- NMEA 0183 Protocol Converter from Excalibur Systems Inc.
- Free software libraries for parsing or building NMEA sentences

  - C library (parsing and building NMEA)
  - Java library (decoding and encoding of NMEA)
  - Java/Eclipse plug-in (parsing NMEA)
  - C# library (parsing NMEA)l
  - Ruby gem (parsing NMEA and AIS)
  - C#/Java library (parsing and building NMEA)