

**Application Note:
GSM2000AN009**

**Enfora Mobile Tracker Output
Message Header Decoding**

Revision 1.02

Confidential and Proprietary Information - © 2003 Enfora, Inc.
Do not duplicate without express permission from Enfora, Inc.

**Enfora, Inc.
www.enfora.com**

Objective:

The intent of this document is to give the user a basic understanding of the message header transmitted with GPS messages over the air.

Supported devices:

- GSM2203 MT-G
- GSM2208 MT-G
- GSM2218 MT-GL

Note: Some features may not be available on some hardware or firmware revisions. Please consult the applicable hardware documentation and firmware release notes.

References:

- GSM2208QS001 - MT-G Quick Start Guide
- GSM2208UG001 - MT-G User Manual
- GSM2218QS001 - MT-GL Quick Start Guide
- GSM2218UG001 - MT-GL User Manual
- GSM2000AT001 - MT-G AT Command Set
- GSM0107AT001 - Enabler-IIG AT Command Set

Overview:

A message header is appended at the beginning of any message generated by the Enfora® MT devices before being transmitted to a remote host. The Enfora® MT products send an Over The Air (OTA) message via one of the three methods:

1. UDP Message
2. UDP Message with ACK
3. UDP Message Broadcasts

The above mentioned message transmitting options are configured as part of the \$EVENT command – parameter 3 (event category). Parameter 4 of the command specifies a message number that can be appended to the message to provide a user-defined number to uniquely identify each output message. Parameter 5 is the bit-wise OR of the individual bits. These bits are defined below.

Bit	Equivalent Hex Value	Description
0	0x00000001	ASCII or Binary format
1	0x00000002	User Specified Number
2	0x00000004	Modem ID
3	0x00000008	GPIO
4	0x00000010	A/D 1
5	0x00000020	A/D 2
6	0x00000040	Store GPS Messages
7	0x00000080	Input Event

8	0x00000100	Binary Date or NMEA GGA message
9	0x00000200	Binary Status or NMEA GLL message
10	0x00000400	Binary Latitude or NMEA GSA message
11	0x00000800	Binary Longitude or NMEA GSV message
12	0x00001000	Binary Speed or NMEA RMC message
13	0x00002000	Binary Heading or NMEA VTG message
14	0x00004000	Binary Time
15	0x00008000	Binary Altitude
16	0x00010000	Binary Number of Satellites
17	0x00020000	Binary Disable OTA messages in low pwr mode
18	0x00040000	Send message with SMS as backup OTA method
19	0x00080000	Send Last know GPS data if GPS is Invalid
20	0x00100000	GPS Odometer
21	0x00200000	RTC Time
22 – 31	-	Reserved

Table 1. Output Bit Settings

Bit 0 - ASCII or Binary format

This bit instructs the Enfora® MT-G to send data as an ASCII string or binary format. If 0, then Parameter 5 bits 8-13, listed in Table 2 above, specify GPS NMEA data. If 1, then Parameter 5 bits 8-16 specify GPS binary data.

Bit Value: 0 – ASCII
 1 – Binary

Bit 1 - User Specified Number

This bit, when enabled, adds the user-specified number (Parameter 4) at the beginning of the message transmitted OTA. The length and format of message number depends on the *ASCII or Binary* bit setting. If the user requests this data in ASCII format, an 11 digit ASCII number will be sent. Otherwise, A four-byte equivalent of the ASCII value is transmitted in hex format.

Total data length: 11 – Bytes (ASCII) – The data field of 10 bytes will be followed by a space.
 4 – Bytes (Binary)

Bit 2 - Modem ID

This bit, when enabled, adds the modem ID set by \$MDMID command to the message transmitted OTA. The length of this field is not affected by the *ASCII or Binary* bit setting. A 22-byte modem ID is appended to the messages transmitted OTA. The format of this field will be a single space followed by 20 bytes of MDMID followed by another single space.

Bit 3 - GPIO

This bit, when enabled, adds the GPIO mask and data value to the message transmitted OTA. The length and format of this data field depends on the *ASCII or Binary* bit setting.

Total data length: 6 – Bytes (ASCII) – First 2 bytes is the data mask as defined by AT\$IOCFG followed by a comma followed by 2 bytes of current GPIO data followed by a space.

2 – Bytes (Binary) – First byte is the data mask as defined by AT\$IOCFG followed by a byte of current GPIO data.

Bit 4 - A/D 1

This bit, when enabled, adds the current value of A/D channel 1 to the message header. The A/D data is returned in millivolts with 0 as the minimum and 1750 as the maximum value. The A/D uses a 10-bit DAC.

Total data length: 5 – Bytes (ASCII) – First four bytes of data followed by a single space

2 – bytes (Binary – big-endian format)

Bit 5 - A/D 2

This bit, when enabled, adds the current value of A/D channel 2 to the message header. The A/D data is returned in millivolts with 0 as the minimum and 1,750 as the maximum value. The A/D uses a 10-bit DAC.

Total data length: 5 – Bytes (ASCII) – First four bytes of data followed by a single space

2 – bytes (Binary – big-endian format)

Bit 6 - Store GPS Messages

This bit, when enabled, stores the GPS messages in a buffer if the device is not able to transmit GPS messages OTA. Up to 500 messages will be stored in a FIFO manner if this bit is enabled. The oldest message will be discarded and replaced with the new one when the 500 message buffer is filled

Bit 7 - Input Event

This bit, when enabled, adds the input event number that caused the output event. This field can also be looked upon as the reason why the message is being sent. The length and format of this data field depend on the *ASCII or Binary* bit setting.

Total data length: 3 – Bytes (ASCII) – First 2 bytes are input event number followed by a single space

1 – Byte (Binary)

Bit 8 - Binary Date or NMEA GGA message

This bit, when enabled, adds the current date in binary format or a GGA NMEA message in ASCII format to the message transmitted OTA. A GGA NMEA message is added if the *ASCII or Binary* bit is set to 0. Otherwise, date value in binary format is added.

Total data length: up to 80 – Bytes (ASCII)
3 – bytes (Binary Date – big-endian format)

Bit 9 - Binary Status or NMEA GLL message

This bit, when enabled, adds the current message status in binary format or the GLL NMEA message in ASCII format to the message transmitted OTA. A GLL NMEA message is added if the *ASCII or Binary* bit is set to 0. Otherwise, message status value in binary format is added.

Total data length: up to 80 – Bytes (ASCII)
 1 – byte (Binary Status – big-endian format)

Bit 10 - Binary Latitude or NMEA GSA message

This bit, when enabled, adds the current latitude in binary format or the GSA NMEA message to the message transmitted OTA. A GSA NMEA message is added if the *ASCII or Binary* bit is set to 0. Otherwise, latitude value in binary format is added.

Total data length: up to 80 – Bytes (ASCII)
 3 – bytes (Binary Latitude – big-endian format)

Bit 11 - Binary Longitude or NMEA GSV message

This bit, when enabled, adds the current longitude in binary format or the GSV NMEA message to the message transmitted OTA. A GSV NMEA message is added if the *ASCII or Binary* bit is set to 0. Otherwise, longitude value in binary format is added.

Total data length: up to 240 – Bytes (ASCII)
 4 – bytes (Binary Longitude – big-endian format)

Bit 12 - Binary Speed or NMEA RMC message

This bit, when enabled, adds the current speed in binary format or the RMC NMEA message to the message transmitted OTA. A RMC NMEA message is added if the *ASCII or Binary* bit is set to 0. Otherwise, speed value in binary format is added.

Total data length: up to 80 – Bytes (ASCII)
 2 – bytes (Binary Speed – big-endian format)

Bit 13 - Binary Heading or NMEA VTG message

This bit, when enabled, adds the current heading in binary format or the VTG NMEA message to the message transmitted OTA. A VTG NMEA message is added if the *ASCII or Binary* bit is set to 0. Otherwise, heading value in binary format is added.

Total data length: up to 80 – Bytes (ASCII)
 2 – bytes (Binary Heading – big-endian format)

Bit 14 - Binary Time

This bit, when enabled, adds the current time in binary format if the *ASCII or Binary* bit value is set to 1. This bit does not have any effect on the message if the *ASCII or Binary* bit is set to 0.

Total data length: 0 – Bytes (ASCII)
 3 – bytes (Binary Time – big-endian format)

Bit 15 - Binary Altitude

This bit, when enabled, adds the current altitude value in binary format if the *ASCII or Binary* bit value is set to 1. This bit does not have any effect on the message if the *ASCII or Binary* bit is set to 0.

Total data length: 0 – Bytes (ASCII)
 3 – bytes (Binary Altitude– big-endian format)

Bit 16 - Binary Number of Satellites

This bit, when enabled, adds the current number of GPS satellites in view in binary format if the *ASCII or Binary* bit value is set to 1. This bit does not have any effect on the message if the *ASCII or Binary* bit is set to 0.

Total data length: 0 – Bytes (ASCII)
 1 – bytes (Binary Number of Satellites – big-endian format)

Bit 17 - Binary Disable OTA Messages in Low Pwr Mode

This bit, when enabled, disables sending of OTA messages when the MT is in Low Power Mode in binary format if the *ASCII or Binary* bit value is set to 1. This bit does not have any effect on the message if the *ASCII or Binary* bit is set to 0.

Bit Value: 0 – Send OTA message when the modem is in low power mode
 1 – Do not send OTA message when the modem is in low power mode

Bit 18 - Send Message with SMS as backup OTA method

This bit, when enabled, sends OTA messages via SMS if the unit is unable to send via GPRS.

Bit Value: 0 – Don't send OTA messages via SMS if unable to send via GPRS
 1 – Send OTA messages via SMS if unable to send via GPRS

Bit 19 - Send Last Know GPS data if current GPS status is Invalid.

This, bit when enabled, will fill all GPS data with the last known GPS data if the current data is invalid. It will also change to the status field or bit to a “9” to indicate the message contains last know GPS data.

Bit Value: 0 – Do not use last know GPS data if the current GPS data is invalid
 1 – Use last known GPS data if the current GPS data is invalid.

Bit 20 - Add value given by AT\$ODOMETER command.

This, bit when enabled, adds the value given by the AT\$ODOMETER command.

Total data length: 11 – Bytes (ASCII) with a preceding space
4 – bytes Binary

Bit 21 - Send RTC Time.

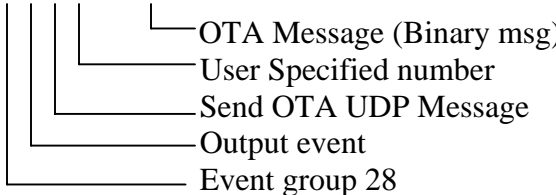
This bit, when enabled, appends the RTC time to the message. It contains: Year, Month, Day, Hour, Minute, and Seconds information.

Total data length: 13 – Bytes (ASCII) with a preceding space in the format of YYMMDDhhmmss
6– bytes Binary in the format of YMDhms

Example

A) The following example creates a timer event to transmit a GPS message in binary format with all bits enabled.

AT\$EVTIM1=60 (send message every 60 seconds)
AT\$EVENT=28,1,12,1,1 (configure a Timer1 event and start the timer)
AT\$EVENT=28,3,40,9,131071 (configure an UDP Msg output event)



OTA Message (Binary msg)
User Specified number
Send OTA UDP Message
Output event
Event group 28

The output message is created by decoding of value:

131071 (see table 1 above for bit-field decoding).

131071 (decimal) = 0x0001FFFF (Hex)

Binary data enabled

User Specified Number to be added (9 – this value comes from

AT\$EVENT=1,3,40,9,131071)

Modem ID to be added

(this value come from AT\$MDMID)

GPIO data to be added

(this value comes from AT\$IOCFG and

AT\$IOGPA)

Binary A/D 1 to be added

(this value comes from AT\$IOAD1)

Binary A/D 2 to be added

(this value comes from AT\$IOAD2)

Input event to be added

(12 – this value comes from

AT\$EVENT=1,1,12,1,1)

Binary date to be added

Binary status to be added

Binary latitude to be added

Binary longitude to be added

Binary speed to be added
 Binary heading to be added
 Binary time to be added
 Binary altitude to be added
 Binary number of satellite to be added

The GPS output data will look as follows:

Data type	Original value in ASCII	Converted values in ASCII format	Value in binary format
Date	31-03-01	310301	0x04BC1D
Status	1	1	0x01
Latitude	37 23.247(North)	3723247 (deg * 100000)	0x38CFEF
Longitude	121 58.341(West)	-12158341 (deg * 100000)	0xFF467A7B
Speed	1.2	12 (speed * 10)	0x000C
Heading	309.6	3096 (heading * 10)	0x0C18
Time	11:33:42	113342	0x01BABE
Altitude	234	234	0x0000EA
GPIO	9F,05 (mask,value)	40709	0x9F05
A/D 1	0	0	0x0000
A/D 2	1075	1075	0x0433

The GPS output message received at the remote host will look as follows:

NOTE: Data in the below example is displayed in HEX format.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
IP Header data															
IP Header															

Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23	Byte 24	Byte 25	Byte 26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31
				UDP Header data								00	05	02	00
IP Header (continued...)				UDP Header								UDP-API Header			

Byte 32	Byte 33	Byte 34	Byte 35	Byte 36	Byte 37	Byte 38	Byte 39	Byte 40	Byte 41	Byte 42	Byte 43	Byte 44	Byte 45	Byte 46	Byte 47
00	00	00	09	20	20	20	20	20	20	20	20	45	6E	66	6F
User Specified Number				Modem ID (Enfora MT-G 1)											

Byte 48	Byte 49	Byte 50	Byte 51	Byte 52	Byte 53	Byte 54	Byte 55	Byte 56	Byte 57	Byte 58	Byte 59	Byte 60	Byte 61	Byte 62	Byte 63
72	61	20	4D	54	2D	47	20	31	20	9F	05	00	00	04	33
Modem ID (continued...)										Mask	Value	A/D 1		A/D 2	
										Gpio					

Byte 64	Byte 65	Byte 66	Byte 67	Byte 68	Byte 69	Byte 70	Byte 71	Byte 72	Byte 73	Byte 74	Byte 75	Byte 76	Byte 77	Byte 78	Byte 79
0C	04	BC	1D	01	38	CF	EF	FF	46	7A	7B	00	98	0C	18

Input Event	Date	Status	Latitude	Longitude	Speed	Heading
-------------	------	--------	----------	-----------	-------	---------

Byte 80	Byte 81	Byte 82	Byte 83	Byte 84	Byte 85	Byte 86
01	BA	BE	00	00	EA	05
Time		Altitude			Number of Satellites	

The following is an example of converting the little-endian binary data string after it has been converted to a decimal value.

Step 1. Received data converted to decimal: 29 188 4
Step 2. Convert decimal to Hex: 1D BC 04
Step 3. Reverse order: 04 BC 1D
Step 4. Convert to decimal: 310301

B) The following example creates a timer event to transmit a NMEA GGA GPS message in ASCII format.

AT\$EVTIM1=60 (send message every 60 seconds)
AT\$EVENT=29,1,12,1,1 (configure a Timer1 event and start the timer)
AT\$EVENT=29,3,40,123,398 (configure an UDP Msg output event)

OTA Message (Binary msg)
User Specified number
Send OTA UDP Message
Output event
Event group 29

The output message is created by decoding of value: 398 (see table 2 above for bit-field decoding).

398 (decimal) = 0x0000018E (Hex)
ASCII data enabled
User Specified Number to be added (123 – this value comes from AT\$EVENT=1,3,40,123,398)
Modem ID to be added (this value come from AT\$MDMID)
GPIO data to be added (this value comes from AT\$IOCFG and AT\$IOGPA)
Input event to be added (12 – this value comes from AT\$EVENT=1,1,12,1,1)
ASCII GGA message to be added

The GPS output message transmitted OTA will look as follows:

NOTE: Data in the below example is displayed in HEX format.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
IP Header data															
IP Header															

Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23	Byte 24	Byte 25	Byte 26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31
				UDP Header data								00	04	02	00
IP Header (contd...)				UDP Header								UDP-API Header			

Byte 32	Byte 33	Byte 34	Byte 35	Byte 36	Byte 37	Byte 38	Byte 39	Byte 40	Byte 41	Byte 42	Byte 43	Byte 44	Byte 45	Byte 46	Byte 47
20	20	20	20	20	20	20	31	32	33	20	20	20	20	20	20
User Specified Number (123)											Modem ID				

Byte 48	Byte 49	Byte 50	Byte 51	Byte 52	Byte 53	Byte 54	Byte 55	Byte 56	Byte 57	Byte 58	Byte 59	Byte 60	Byte 61	Byte 62	Byte 63
20	20	20	45	6E	66	6F	72	61	20	4D	54	2D	47	20	31
Modem ID continued... (Enfora MT-G 1)															

Byte 64	Byte 65	Byte 66	Byte 67	Byte 68	Byte 69	Byte 70	Byte 71	Byte 72	Byte 73	Byte 74	Byte 75	Byte 76	Byte 77	Byte 78	...
20	39	46	2C	30	43	20	31	32	20	24	47	50	47	47	...
Modem ID continued	Mask		comma	Data		space	Input Event Number				ASCII NMEA GGA message (\$GPGGA,.....)				
	GPIO														

NOTE: The ASCII GGA message always starts with “\$GPGGA” and ends with “*<CR><LF>” and could be up to 80 characters in length.

Revision History

Date	Rev	Author	Description
6/5/03	1.00	Matt Glover	Initial Release.
11/20/03	1.01	C Patel	Defined bit-18 in the bit field
08/16/05	1.02	C French	Changed all references from Spider to Enfora. Defined Bits 19-21 in the bit field. Minor format change.