AT Command Doc
GSM2228AT001

Enfora® Mini-MT
AT Command Set Reference

Release 1.03

Enfora, Inc.
http://www.enfora.com

Document Title:	Enfora GSM/GPRS Mini-MT AT Command Set Reference
Version:	1.03
Date:	12/5/2007
Status:	Released
Document Control ID:	GSM2228AT001

General

All efforts have been made to ensure the accuracy of material provided in this document at the time of release. However, the items described in this document are subject to continuous development and improvement. All specifications are subject to change without notice and do not represent a commitment on the part of Enfora Inc. Enfora Inc. will not be responsible for any loss or damages incurred related to the use of information contained in this document.

This product is not intended for use in life support appliances, devices or systems where a malfunction of the product can reasonably be expected to result in personal injury. Enfora Inc. customers using, integrating, and/or selling this product for use in such applications do so at their own risk and agree to fully indemnify Enfora Inc. for any damages resulting from illegal use or resale.

Copyright

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Enfora Inc.

Enfora may have patents, patent applications, trademarks, copyrights or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Enfora, the furnishing of this document does not give you any license to these patents, trademarks, copyrights or other intellectual property.

Enabler is a registered trademark of Enfora Inc. in the United States.

Table of Contents

1 Introduction 1

1.1 Document Scope 1

1.2 Platform Reference and Use 1

1.3 Command Syntax 1

1.4 Revision History 2

2 AT Command Format 3

3 Mini-MT AT Commands 4

3.1 Battery Level 4

3.2 Call Delay 6

3.3 Dispatch Notification 7

3.4 Dispatch Phone Number 8

3.5 Emergency Phone Number 9

3.6 EVENT 11

3.7 Geo-Fencing 21

3.8 Geofence Debounce 23

3.9 GPS Clear 25

3.10 GPS Local Subscription 26

3.11 GPS Quality Filters 29

3.12 GPS Receiver Operation Mode 31

3.13 GPS Read 33

3.14 Interval Wakeup Timer 36

3.15 Keybeep Sound 38

3.16 Key Function Disable 40

3.17 LTO Download 42

3.18 Message Log Clear 44

3.19 Message Log Enable 45

3.20 Message Log Read Data 47

3.21 Message Send 50

3.22 Modem Wake Duration 52

3.23 Motion Transition Count 55

3.24 Motion Wake Enable 57

3.25 Odometer 60

3.26 Ring Indicator 62

3.27 RTC Wakeup Timer 63

3.28 Vibration Command 66

3.29 Writing GPS NV RAM 67

4 Appendix A – Default AT Values 68

5 Appendix B – AT Commands over SMS 71

1 Introduction

1.1 Document Scope

This documentation pertains to the AT Command Set to be used in conjunction with the Enfora® Mobile Tracker.

1.2 Platform Reference and Use

The Enfora® Mobile Tracker will be referred to using various terms, to include: MS (Mobile Station), TA (Terminal Adapter), DCE (Data Communication Equipment), or ME (Mobile Equipment).

The Enfora® Mobile Tracker can be controlled via the use of a DTE (Data Terminal Equipment) platform by issuing the AT commands via a serial interface.

1.3 Command Syntax

The attention or "AT" prefix is required prior to entering any command. All commands require a carriage return or <CR> following the entry of the desired command. All command responses are encapsulated by a carriage return and line feed or <CR><LF>. The ASCII display of these characters is suppressed with only the modem response being presented.

The following examples demonstrate the potential usage of AT commands presented:

Type	Example	Description
Command Format Query	AT$GXXX=?	When entered will return the command format and value ranges.
Command Read	AT$GXXX?	When entered will return the current value assigned to the command.
Command Write	AT$GXXX=<value>,<value>,…	When entered will set the command to specified value(s).
Command Execution	AT$GXXX	When entered will execute the specified command.

1.4 Revision History

Date	Rev	Author	Description
1/09/07	Initial Release	Diane O'Neil	Initial Release
2/08/07	1.01	Diane O'Neil	Edited wake conditions in $WAKEENBL Added Bit23 to Parm 2 of $EVENT table Added Input Trigger 64 to $EVENT Added $LTODL
4/19/07	1.02	Diane O'Neil	Added note to $ODOMETER
12/5/07	1.03	Diane O'Neil	Added $KEYSND Added more data to $WAKEINTVL Changed parameter of input events 62 & 63 Edited Input Event 49 Added Input Events 66,67,68,69,70 Added $VIBNOW Added $GPSFLASH Added $GOPMD Edited $BATTLVL Edited $MSGLOGEN Added $GPSCLR Added $GFDBNC Added Input Event 65, edited Output Event 45, added Output Events 54 – 58 Added Input Event 72, edited Bit 24 Added $KEYFNC Edited $GPSQUAL

2 AT Command Format

The following is the format in which all commands will be presented.

xx.xx (Command Number) Atx(Command) Xxxxx(Command Description)

Command Function	(Description of the command function)
Command Functional Group	(Functional group identification)
Command Format Query **Response**	ATx=? **ATx: (parameter1 name 1 – 15), (parameter2 name 1-10),…**
Write Format **Response**	ATx=**<value>,<value>[,<optional value>],…** **OK** or **ERROR**
Read Format **Response**	ATx? **<value>,<value>,…**
Execution Format **Response**	ATx **OK**, **ERROR**, or **<value>**
Parameter Values	**<value1>,<value2>** ATx: (1-15),(1-10)
Reference	(Applicable standard reference)
Standard Scope	Mandatory or Optional
Enfora Implementation Scope	Full, Partial, or Not Supported
Notes	(Additional command notes)

Please note that, where applicable, the <value> responses provided for the READ and EXECUTION formats are modem default values. All efforts will be made by Enfora, Inc. to keep these values current in the documentation but will not be responsible for any differences that may occur as a result subsequent software builds and version enhancements.

3 Mini-MT AT Commands

3.1 Battery Level

$BATTLVL	**Battery Level**
Command Function	This command allows the user to view the approximate remaining battery level percentage.
Command Functional Group	Enfora Specific
Command Format Query	N/A
Response	N/A
Write Format	N/A
Response	N/A
Read Format	AT$BATTLVL?
Response	$BATTLVL: **<percentage>** OK
Execution Format	N/A
Response	N/A
Parameter Values	
<percentage>	Approximate percentage of battery life remaining.
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	The user must wait one (1) minute before issuing the AT$BATTLVL? Command to check battery status when charging the battery. When the modem wakes up, the initial readings related to the battery level are not a reliable indicator of the actual battery level. So for the first 50 seconds after the modem wakes up, the modem

3.1 $BATTLVL Battery Level (continued)

collects battery level readings to establish an initial battery level percentage. If $BATTLVL is queried before this initial percentage is established, $BATTLVL will return 50.

After the initial battery level percentage is established, readings are taken once a minute to adjust the percentage, if necessary. Even though the percentage is filtered, the oscillations in readings can cause the percentage to temporarily ‘bounce’ between two values for a few minutes (20%, 19%, 20%, 19%, etc). Since this could result in multiple low battery messages via the event engine for the same low threshold crossing, the algorithm used for calculating the percentage while the battery is discharging forces an increment of at least 2% before it will report a different value.

To send a low battery message, you would typically set up the following input event (using event group 20 and 15% for the low battery threshold in this case):

```
AT$EVENT=20,0,59,0,15
```

3.2 Call Delay

$DLYCALL	**Call Delay**
Command Function	This command provides a delay between the detection of the Push-To-Call (PTC) button press and the actual initiation of the call to the dispatch number. The delay allows the event engine time to perform tasks such as sending GPS data via UDP while the modem is still GPRS registered.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$DLYCALL=? $DLYCALL=(0-5) OK
Write Format **Response**	AT$DLYCALL=**<seconds>** OK
Read Format **Response**	AT$DLYCALL? $DLYCALL: **<seconds>** OK
Execution Format **Response**	N/A N/A
Parameter Values	
<seconds>	Number of seconds to delay between detection of PTC button press and initiation of call.
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.3 Dispatch Notification

$DSPNOTIF	**Dispatch Notification**
Command Function	This command is used to control the 'Dispatch Notification' LED. It is used primarily via the AT command over SMS function to notify the user that the user should call the dispatch number.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$DSPNOTIF=? $DSPNOTIF:(0-1) OK
Write Format **Response**	AT$DSPNOTIF=**<state>** OK
Read Format **Response**	AT$DSPNOTIF? $DSPNOTIF:**<state>** OK
Execution Format **Response**	N/A N/A
Parameter Values	
<state>	1 = causes dispatch notification LED to flash 0 = caused dispatch notification LED to stop flashing
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.4 Dispatch Phone Number

$DSPATCH	**Dispatch Phone Number**
Command Function	This command allows the user to query/set the phone number used when the CALL button is pressed
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$DSPATCH=? $DSPATCH:”Number” OK
Write Format **Response**	AT$DSPATCH=”**<dispatch number>**” OK
Read Format **Response**	AT$DSPATCH? $DSPATCH:”**<dispatch number>**” OK
Execution Format **Response**	N/A N/A
Parameter Values	
<dispatch number>	Phone number used when CALL button is pressed
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	N/A

3.5 Emergency Phone Number

$EMERNUM	**Emergency Phone Number**
Command Function	This command allows the user to query/set the phone number used when the CALL button and USER-DEFINED button are pressed in the sequence defined for emergency service calls. To make an emergency service call, press and hold down the CALL and USER-DEFINED buttons at the same time. When the Power-On, Dispatch Notification/GPS Lock, and Low Battery LEDs start flashing in unison, the emergency service call logic is armed. Without releasing the CALL button, release and re-press the USER-DEFINED button to make the emergency service call.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$EMERNUM=? $EMERNUM: “Number” OK
Write Format	AT$EMERNUM=**”<emergency number>”**
Response	OK
Read Format **Response**	AT$EMERNUM? $EMERNUM:”**<emergency number>**” OK
Execution Format **Response**	N/A N/A
Parameter Values	
<emergency number>	Phone number used for emergency assistance

3.5 $EMERNUM	**Emergency Phone Number (continued)**
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	N/A

3.6 EVENT

$EVENT — **User Defined Input/Output**

Command Function

This command allows the user to customize the modem's input and output capabilities. Any combination of input events can be monitored to trigger any combination of output events.

Command Functional Group

Enfora Specific

Command Format Query
Response

AT$EVENT=?
$EVENT: (0-99),(0-3),(0-255),(-2147483647 - 2147483647),(-2147483647 - 2147483647)

Write Format
Response

AT$EVENT=<**event group**>,<**event type**>,<**event category**>,<**parm1**>,<**parm2**>
OK

Read Format
Response

```
AT$EVENT?
$EVENT:   evgp evtyp evcat     p1       p2
            1A    0    27       1        1
            1B    3    22       0        0
OK
```

Execution Format
Response

N/A
N/A

3.6 $EVENT User Defined Input/Output (continued)

Parameter Values

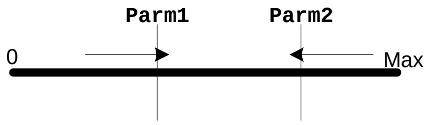
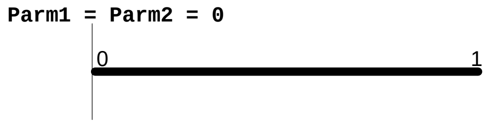
<event group>

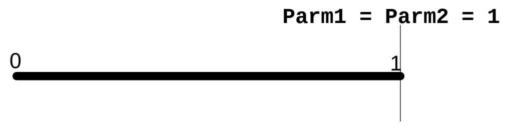
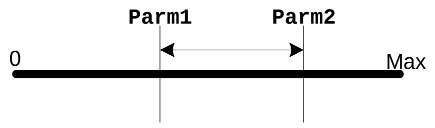
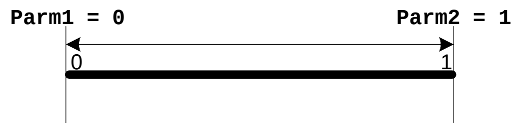
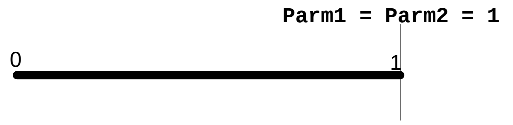
This parameter defines the group number of a group of events and the order they are executed. Events are grouped together to control execution sequence. A group number has to have at least one input event and one output event. Multiple input events within a group number would be treated as a logical **AND** condition. Multiple output events within a group number would be executed individually in a sequential manner.

Valid values for group number are: 1 thru 99.

<event type>

This parameter defines the type of event: Input or Output. An Input event can be defined as: Transition, Occurrence, or Input. The output event is executed when input event conditions are met.

Value	Type of event	Description
0	***Transition Trigger***	A transition Trigger is defined as an input condition, defined by **<event category>**, whose value was previously **<parm1>** or less is now greater than **<parm1>** and less than **<parm2>** **or** was greater or equal to **<parm2>** is now less than **<parm2>** but greater than **<parm1>**. The output event would be executed when an input **<event category>** requirements are satisfied or transition to the value set by **<parm1>** and **<parm2>** when they are equal. **<parm1>** should be the min value and **<parm2>** should be the max value. Example 1:  **Figure 1.** An output event will be executed when the value of an input event exceeds **<Parm1>** (previously it was **<Parm1**> or less) or decreases to a value less than **<Parm2>** (previously it was <**Parm2>** or greater). Example 2:  **Figure 2.** An output event will be executed when the value of an input event is 0 (previously it was

		anything else but 0) and **<Parm1>** along with **<Parm2>** is set to 0. Example 3:  **Figure 3.** An output event will be executed when the value of an input event is 1 (previously it was anything else but 1) and **<Parm1>** along with **<Parm2>** is set to 1.
1	***Occurrence Trigger***	An Occurrence Trigger is defined as an input condition, defined by **<event category>**, whose current value is greater than or equal to **<parm1>** and less than or equal to **<parm2>**. The output event would be executed when an input **<event category>** requirements are satisfied or transition to the value set by **<parm1>** and **<parm2>** when they are equal. **<parm1>** should be the min value and **<parm2>** should be the max value Example 4:  **Figure 4.** An output event will be executed when the current value of an input event is between **<Parm1>** and **<Parm2>** including boundary conditions. Example 5:  **Figure 5.** An output event will be executed when the value of the input event changes from 0 to 1 or vice-versa. Example 6:  **Figure 6.** An output event will be executed when the value of the input event is 1 and **<Parm1>** along with **<Parm2>** is set to 1.
2	***Input Trigger***	An Input Trigger is defined as an input condition, defined by **<event category>**, that should be

		used as a logical **AND** condition to another input condition defined as Transition Trigger or an Occurrence Trigger. An Output event is not triggered when Input Trigger condition is valid. The input event, defined as Input Trigger, is valid when within the event range defined by **`<parm1>`** and **`<parm2>`** or when **`<parm1>`** and **`<parm2>`** are equal.
3	***Output***	An Output event is executed when all input event conditions (defined as Transition Trigger, Occurrence Trigger, or Input Trigger) for that particular **`<event group>`** are met.

<event category> This parameter defines the actual Input or Output Event number and their valid range for **<parm1>** and **<parm2>**.

The below table defines the values for **<event category>**, **<parm1>** and **<parm2>** parameter for input events defined as a ***Transition Trigger***, ***Occurrence Trigger***, or ***Input Trigger***.

	event category	**Parm1**	**Parm2**	**Description**
	0	N/A	N/A	Reserved
	1	N/A	N/A	Reserved
	2	N/A	N/A	Reserved
	3	N/A	N/A	Reserved
	4	N/A	N/A	Reserved
	5	N/A	N/A	Reserved
	6	N/A	N/A	Reserved
	7	N/A	N/A	Reserved
	8	1	1	Modem power up indication
	9	0 to 5	0 to 5	Modem GSM registration (see AT+CREG command description for GSM registration status information)
	10	0 to 8	0 to 8	Modem GPRS registration (see AT%CGREG command description for GPRS registration status information)
	11	0 or 1	0 or 1	Receipt of IP address. 0 = No IP address 1 = Valid IP address obtained
	12	1	1	Timer 1 (set by AT$EVTIM1)
	13	1	1	Timer 2 (set by AT$EVTIM2)
	14	1	1	Timer 3 (set by AT$EVTIM3)
	15	1	1	Timer 4 (set by AT$EVTIM4)
	16	0 to 1000000	1000000	GPS Distance (unit of measurement is: meters
	17	0 to 250	250	Maximum Velocity (unit of measurement is: Knots)
	18	N/A	N/A	Reserved
	19	N/A	N/A	Reserved
	20			Reserved

	21	0 or 1	0 or 1	Geo Fence #1. See AT$GEOFNC command for details on setting a circular geo-fence 0 = Leaving Geofence area 1 = Entering Geofence area
	22	0 or 1	0 or 1	Geo Fence #2
	23	0 or 1	0 or 1	Geo Fence #3
	24	0 or 1	0 or 1	Geo Fence #4
	25	0 or 1	0 or 1	Geo Fence #5
	26	N/A	N/A	Reserved
	27	0 or 1	0 or 1	GPS Status 0 = Invalid GPS data 1 = Valid GPS data
	28	N/A	N/A	Reserved
	29	0 to 1000000	1000000	Invalid GPS data for a period of time (unit of measurement is: increment of GPS reporting interval defined in $GOPMD command)
	30	0 to 1000000	1000000	Unit staying Idle in one place (unit of measurement is: increment of GPS reporting interval defined in $GOPMD command)
	31	0 or 1	0 or 1	Geo Fence #6. See AT$GEOFNC command for details on setting a circular geo-fence 0 = Leaving Geofence area 1 = Entering Geofence area
	32	0 or 1	0 or 1	Geo Fence #7
	33	0 or 1	0 or 1	Geo Fence #8
	34	0 or 1	0 or 1	Geo Fence #9
	35	0 or 1	0 or 1	Geo Fence #10
	36	0 or 1	0 or 1	Geo Fence #11
	37	0 or 1	0 or 1	Geo Fence #12
	38	0 or 1	0 or 1	Geo Fence #13
	39	0 or 1	0 or 1	Geo Fence #14
	40	0 or 1	0 or 1	Geo Fence #15
	41	0 or 1	0 or 1	Geo Fence #16
	42	0 or 1	0 or 1	Geo Fence #17
	43	0 or 1	0 or 1	Geo Fence #18
	44	0 or 1	0 or 1	Geo Fence #19
	45	0 or 1	0 or 1	Geo Fence #20
	46	0 or 1	0 or 1	Geo Fence #21
	47	0 or 1	0 or 1	Geo Fence #22
	48	0 or 1	0 or 1	Geo Fence #23
	49	0 or 1	0 or 1	Geo Fence #24
	50	0 or 1	0 or 1	Geo Fence #25
Will only work on occurrence trigger, not transitions	51	0	0	**Input Event Counter. This event will occur when a counter reaches the maximum number of a selected Input event count.
	52	0 or 1	0 or 1	New SMS indication. 0 = SMS message read from SIM 1 = New SMS message received
	53	0 to -1	0 to –1	Current Input Event Counter count that can be used as an AND condition with other input events
	54	0-1	0-1	Geofence Exist
	55	N/A	N/A	Reserved (Do Not Use)

	56	N/A	N/A	Reserved (Do Not Use)
	57	0-2147483647	0-2147483647	Messages to be sent Over-The-Air exist
	58	0 – 4	0 – 4	Keypress Event 0 = Set Geofence key pressed 1 = User Defined key pressed 2 = Push To Call key pressed 3 = Volume Down key pressed 4 = Volume Up key pressed
	59	0-100	0-100	Battery Level Event Approximate percentage of battery life left (0-100% - see $BATTLVL)
	60	0-2147483647	0-2147483647	Number of unsent messages
	61	0-100	0-100	Memory full percentage
	62	0-1	0-1	Motion Status 1 = moving 0 = stopped
	63	0-1	0-1	Power Source 1 = External power 0 = Battery power
	64	1 – 3	1 – 3	1 = LTO download started 2 = LTO download completed successfully 3 = LTO download failed
	65	1 to 5	1 to 5	Receipt of Incoming Call with Call Identifier matching one the numbers configured via the **$EVCID** command. <Parm1> and <Parm2> correspond to range **$EVCID** entries which will generate the input event.
	66	1	1	Timer 5 (set by AT$EVTIM5)
	67	1	1	Timer 6 (set by AT$EVTIM6)
	68	1	1	Timer 7 (set by AT$EVTIM7)
	69	1	1	Timer 8 (set by AT$EVTIM8)
	70	0-4000000000	0-4000000000	Current $ODOMETER value
	71	N/A	N/A	Reserved
	72	0 – 1	0 – 1	** MiniMT Only – Keypress Event 0 = A GPS overspeed interval has ended 1 = A GPS overspeed interval has begun

The below table defines the values for **<event category>**, **<parm1>** and **<parm2>** parameter for output events defined as ***Output***.

event category	Parm1	Parm2	Description
0	N/A	N/A	Reserved
1	N/A	N/A	Reserved
2	N/A	N/A	Reserved
3	N/A	N/A	Reserved
4	N/A	N/A	Reserved
5	N/A	N/A	Reserved

6	N/A	N/A	Reserved
7	N/A	N/A	Reserved
8	N/A	N/A	Reserved
9	N/A	N/A	Reserved
10	N/A	N/A	Reserved
11	N/A	N/A	Reserved
12	N/A	N/A	Reserved
13	N/A	N/A	Reserved
14	N/A	N/A	Reserved
15	N/A	N/A	Reserved
16	N/A	N/A	Reserved
17	N/A	N/A	Reserved
18	N/A	N/A	Reserved
19	N/A	N/A	Reserved
20	N/A	N/A	Reserved
21	N/A	N/A	Reserved
22	N/A	N/A	Reserved
23	N/A	N/A	Reserved
24	N/A	N/A	Reserved
25	N/A	N/A	Reserved
26	N/A	N/A	Reserved
27	N/A	N/A	Reserved
28	N/A	N/A	Reserved
29	N/A	N/A	Reserved
30	N/A	N/A	Reserved
31	N/A	N/A	Reserved
32	N/A	N/A	Reserved
33	N/A	N/A	Reserved
34	N/A	N/A	Reserved
35	N/A	N/A	Reserved
36	N/A	N/A	Reserved
37	N/A	N/A	Reserved
38	N/A	N/A	Reserved
39	N/A	N/A	Reserved
40	0-2147483647	See **Bit-Field Table** below	Generate and transmit one UDP Message to first IP address listed in **$FRIEND** command and port number listed in **$UDPAPI** command based on **Parm1** and **Parm2** values
41			Generate and transmit a UDP message with Acknowledge. This message is controlled by **$ACKTM** command for number of retries sent. This message has to be acknowledged to avoid sending of retries.
42			Generate and transmit one UDP Message to all IP address listed in **$FRIEND** command and port number listed in **$UDPAPI** command based on **Parm1** and **Parm2** values

43	1 – 4	0	Resets the timer (Timer #1 – Timer #4) specified by **Parm1** to the time (in seconds) specified by **Parm2**. **Parm2**, when set to 0, resets the timer to the time last set by **$EVTIMx** command. A value other than 0 would set the timer to expire at the new specified interval. A timer can only be disabled by setting **$EVTIMx** command to 0.
44	1 – 15	0	Execute AT command stored at index number of the **$STOATEV** command. **Parm1** identifies the index number.
45	0-2147483647	See **Bit-Field Table** below	Sends data over SMS to All SMS destination addresses configured via **$SMSDA** command. (For select **$SMSDA** entries, see event categories 54-58)
46	N/A	N/A	Reserved
47	0	0 to –1	Input Event Counter
48	0	0	Reset Event Counter to zero
49	1 – 25	0 - 1000000	Set geo-fence specified by parm1 to current latitude & longitude with radius specified by parm2
50	0 – 57	0 to –1	Emulate AT$EVTEST command via event engine. Parm1 is the input event number while Parm2 is the value to emulate for the input event
51	N/A	N/A	Reserved
52	0 to –1	See Bit-Field Table below	Generate and transmit one TCP/IP Message to IP address & port number listed by $FRIEND command based on Parm1 and Parm2 values
53	N/A	N/A	Reserved
54	0-2147483647	See **Bit-Field Table** below	Sends data over SMS to the first indexed SMS destination address configured via **$SMSDA** command
55	0-2147483647	See **Bit-Field Table** below	Sends data over SMS to the first indexed SMS destination address configured via **$SMSDA** command
56	0-2147483647	See **Bit-Field Table** below	Sends data over SMS to the first indexed SMS destination address configured via **$SMSDA** command
57	0-2147483647	See **Bit-Field Table** below	Sends data over SMS to the first indexed SMS destination address configured via **$SMSDA** command
58	0-2147483647	See **Bit-Field Table** below	Sends data over SMS to the first indexed SMS destination address configured via **$SMSDA** command

Bit-Field Table

Parm2 value is obtained as a result of selecting individual bit-fields from the table below.

Parm2	
Bit 0:	1 = send all data generated as a result of this table in Binary format 0 = send all data generated as a result of this table in ASCII format
Bit 1:	1 = add **parm1** data to UDP message (4 – bytes in Binary format, 11 – bytes of data in ASCII format) 0 = do not add **parm1** data to outbound UDP message
Bit 2:	1 = add **$MDMID** value (22 – bytes of ASCII data – irrespective of Bit– 0 setting) 0 = do not add **$MDMID** value
Bit 3:	N/A
Bit 4:	N/A
Bit 5:	N/A
Bit 6:	N/A
Bit 7:	1 = add input **<event category>** number (1 – byte in binary format, 3 – bytes in ASCII format) 0 = do not add input **<event category>** number
Bit 8:	1 = add GPS data (3 – bytes of **Date** information in Binary format **or** up to 80 – bytes of **$GPGGA** NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 9:	1 = add GPS data (1 – bytes of **Status** information in Binary format **or** up to 80 – bytes of **$GPGLL** NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 10:	1 = add GPS data (3 – bytes of **Latitude** information in Binary format **or** up to 80 – bytes of **$GPGSA** NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 11:	1 = add GPS data (4 – bytes of **Longitude** information in Binary format **or** up to two 80 – bytes of **$GPGSV** NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 12:	1 = add GPS data (2 – bytes of **Velocity** information in Binary format **or** up to 80 – bytes of **$GPRMC** NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 13:	1 = add GPS data (2 – bytes of **Heading** information in Binary format **or** up to 80 – bytes of **$GPVTG** NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 14:	1 = add GPS data (3 – bytes of **Time** information in Binary format **or** 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 15:	1 = add GPS data (3 – bytes of **Altitude** information in Binary format **or** 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data

Bit 16: 1 = add GPS data (1 – byte of **Number Of Satellites In View** information in Binary format **or** 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 17: N/A
Bit 18: 1 = send this OTA message via SMS when GPRS services is not available 0 = send this OTA message via GPRS only
Bit 19: 1 = send Last Valid GPS data if current data is invalid 0 = send current GPS data – valid or invalid
Bits 20: 1 = add Odometer reading (4 – bytes of **Odometer** information in Binary format **or** 11 – bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bits 21: 1 = add RTC time (6 – bytes of **RTC** time in Binary format **or** 13 – bytes if Bit-0 is set to 0) 0 = do not add RTC time with GPS data
Bits 22: 1 = Replace/append modem ID field with 10-byte modem ID (including one leading and one ending space character) if bit-0 is set to 0. Replace/append it with 8-bytes long modem ID value if bit-0 is set to 1 (no leading or ending space characters in binary mode). (**NOTE**: bit-22 setting overrides bit-2 setting) 0 = sent the modem ID as defined by Bit-2
Bit 23: 1 = Add battery level (1-byte in Binary format or 3-bytes if Bit-0 is set to 0) 0 = Do not add battery level
Bit 24: 1 = add GPS overspeed data (6 – bytes of **Odometer** information in Binary format **or** 6 to 18 – bytes if Bit-0 is set to 0). Binary format: xxyyzz: xx is speed specified by AT$GPSOSI, yy is the maximum speed incurred during the interval, zz is the duration in seconds of the interval: ASCII format: “ x y z”: space delineated, length of each field varies with its value 0 = do not add this particular field of GPS data

Reference N/A

Standard Scope Optional

Enfora Implementation Scope Full

Notes Maximum of 150 events (input and output).

3.7 Geo-Fencing

$GEOFNC	**Geo fencing a circular area**
Command Function	This command allows a user to send a GPS message when the device moves in or out of a geographical area.
Command Functional Group	Enfora Specific
Command Format Query Response	AT$GEOFNC=? $GEOFNC: (1 – 25),(0 - 100000),(-90 - +90),(-180 - +180) OK
Write Format Response	AT$GEOFNC=<***fenceNum***>,<***radius***>,<***latitude***>,<***longitude***> OK
Read Format Response	AT$GEOFNC? $GEOFNC: <***fenceNum***>,<***radius***>,<***latitude***>,<***longitude***> OK
Execution Format Response	N/A N/A
Parameter Values	
<***fenceNum***>	Defines the fence number
<***radius***>	Defines radius of the circle from given Latitude and Longitude coordinates
<***latitude***>	Defines the latitude for the center point of a circle
<***longitude***>	Defines the longitude for the center point of a circle
Reference	N/A
Standard Scope	Optional

3.7 $GEOFNC Geo fencing a circular area (continued)

Enfora Implementation Scope Full

Notes

An AT$EVENT command has to be set to send a GPS message to the remote host when entering or exiting the fenced area. See the MT-G Users Manual for example.

3.8 Geofence Debounce

$GFDBNC	**Set Geofence Debounce count**
Command Function	This command allows the user to set the # of consecutive geofence positions required to trigger an ‘inside geofence’ or ‘outside geofence’ event.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$GFDBNC=? $GFDBNC:(0-250, 0-250) OK
Write Format **Response**	AT$GFDBNC=**<out_cnt>,<in_cnt>** OK
Read Format **Response**	AT$GFDBNC? $GFDBNC: <out_cnt>, <in_cnt> OK
Execution Format **Response**	N/A
Parameter Values	
<out_cnt>	consecutive GPS position reports outside a geofence required to trigger ‘0’ condition for geofence input event (see $EVENT)
<in_cnt>	consecutive GPS position reports inside a geofence required to trigger ‘1’ condition for geofence input event (see $EVENT)
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.8 $GFDBNC Set geofence debounce count (continued)

Notes	The GPS reporting interval varies depending on the product. For the Mini-MT, the interval is typically set via the 3rd parameter applied to the $GOPMD command.

3.9 GPS Clear

$GPSCLR **Clear GPS Positioning Information**

Command Function This command allows the user to clear the selected GPS positioning information.

Command Functional Group Enfora Specific

Command Format Query Response

```
AT$GPSCLR=?
$GPSCLR: (1-31)
OK
```

Write Format Response N/A

Read Format Response N/A

Execution Format Response AT$GPSCLR=<**mask**>

Parameter Values

<***mask***> Bit mask specifying which GPS information is to be cleared. Each bit specifies the GPS information that is to be cleared as enumerated in the table below.

Bit Value	GPS Info
16	Last known position

To select multiple items to clear, add the bit values of each item to be cleared. To clear LTO, Ephemeris and Almanac, the mask value is 7 (1 + 2 + 4).

Notes If GPS is positioning when this command is issued, an error will be returned and the command will not be executed.

The only Bit Value that applies to the Mini-MT is 16.

3.10 GPS Local Subscription

$GPSLCL	**Configure sending of GPS message to the Serial Port**
Command Function	This command allows the user to configure sending of GPS data on the USB port.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$GPSLCL=? $GPSLCL: (0 – 4),(0-63) OK
Write Format **Response**	AT$GPSLCL=<***option***>,<nmeaMsgs> OK or ERROR
Read Format	AT$GPSLCL?
	$GPSLCL: <***option***>,<nmeaMsgs>
Response	OK
Execution Format **Response**	N/A N/A

3.10 $GPSLCL — Configure sending of GPS message to the Serial Port (continued)

Parameter Values

<***option***>

0 – Disable sending of GPS data to the local USB port when the device is in AT command mode (***Default***)
1 – Enable sending of GPS NMEA ASCII data to the local USB port when the device is in AT command mode
2 – Enable sending of GPS NMEA ASCII data to the local USB port. This option has to be sent by the user in DUN mode. Data sent as a result of this option will always contain a UDP/IP header. Data will be sent to the IP address and port number set by $UDPAPI command. This option has no effect on the operation of the modem when entered via the AT command mode.
3 – Reserved
4 – Reserved

<***nmeaMsgs***>

This field is the bit-wise OR of the type of messages desired. The user has following message options to select from. Maximum value for <***nmeaMsgs***> in this case would be 3F

3.10 $GPSLCL Configure sending of GPS message to the Serial Port (continued)

User Selectable Bits	Type of NMEA Message
0x01	GGA
0x02	GLL
0x04	GSA
0x08	GSV
0x10	RMC
0x20	VTG

Reference N/A

Standard Scope Optional

Enfora Implementation Scope Full

Notes N/A

3.11 GPS Quality Filters

$GPSQUAL	**GPS Quality Filters**
Command Function	This command allows the user to set/query the filter values used to determine when to interpret GPS data as valid.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$GPSQUAL=? $GPSQUAL:(0-1), (0-255) OK
	Write Format AT$GPSQUAL=**”<fix type>,<HDOP level>”**
Response	OK
Read Format **Response**	AT$GPSQUAL? $GPSQUAL:**<fix type>,<HDOP level>**
Execution Format **Response**	N/A
Parameter Values	
<fix type>	0 (default) = consider GPS data valid if $GPGSA fix is either 2D GPS fix (2) or (3D) Differential GPS fix (3). 1 = consider GPS data valid only if $GPGSA fix is (3D) Differential GPS fix (3).
<HDOP level>	0 (default) = do not use HDOP value from $GPGSA sentence when determining whether GPS is valid 1-255 = consider GPS data valid only if HDOP value from $GPGSA sentence is less than or equal to indicated this HDOP limit.

3.11 $GPSQUAL GPS Quality Filters (continued)

Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

Notes

3.12 GPS Receiver Operation Mode

$GOPMD — **GPS receiver operation mode**

Command Function
This command allows a user to set the operation mode for the GPS receiver.

Command Functional Group
Enfora Specific

Command Format Query
AT$GOPMD=?
$GOPMD: (0-3),(1-3),(0,2-7200)
OK

Write Format Response
AT$GOPMD=<***option***>,<***fixMode***>,<***reportInterval***>
OK

Read Format Response
AT$GOPMD?
$GOPMD:
=<***option***>,<***fixMode***>,<***reportInterval***>
OK

Execution Format Response
N/A

Parameter Values

<***option***>
0 – Turn GPS receiver Off
1 – Autonomous
2 – Mobile Based
3 – Enhanced Autonomous

<***fixMode***>
1 – One-Time Fix
2 – Low Power Navigation
3 – Timed Interval

<***reportInterval***>
0 – One-Time Fix or Native Mode
2 – 7200 Time in seconds at which the NMEA GPS data will be generated in Timed Interval mode
NOTE: A value of 0 when fixMode = Timed Interval Mode is the same as turning the GPS receiver Off.

3.12 $GOPMD GPS Receiver Operation Mode (continued)

<***platform_mode***>	**0** – Will be reset to 1 **1** – Tells the Global Locate library that it is running on a slow processor, so it can make appropriate internal adjustments **2** – Tells the Global
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	N/A

3.13 GPS Read

$GPSRD	**Read Current GPS ASCII Data**
Command Function	This command allows a user to read current NMEA format GPS data.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$GPSRD=? $GPSRD: [(0-3F),(0-63)], (0-1) OK
Write Format **Response**	N/A N/A
Read Format **Response**	AT$GPSRD=<***nmeaMsgs***>,<***decimal***> “$GPG…………….” OK
Execution Format **Response**	N/A N/A
Parameter Values	The output NMEA sentence depends on whether the <***nmeaMsgs***> parameter is entered in Hex or Decimal format. By default, the <***decimal***> parameter is not required and <***nmeaMsgs***> parameter has to be entered as HEX value without the preceding “0x” characters as outlined in Hex Format table below.
<***nmeaMsgs***>	This field is the sum of the type of NMEA messages desired. A user has the following message options to select from. Maximum value for <***nmeaMsgs***> in this case would be 3F in Hex format or 63 in decimal format.

3.13 $GPSRD Read current GPS ASCII data (continued)

Hex Format

User Selectable	Type of NMEA Message
0x01	GGA
0x02	GLL
0x04	GSA
0x08	GSV
0x10	RMC
0x20	VTG

Decimal Format

User Selectable	Type of NMEA Message
1	GGA
2	GLL
4	GSA
8	GSV
16	RMC
32	VTG

<***decimal***>

1 = <***nmeaMsg***> value has to be sum of User Selectable values from decimal table format
0 = select values out of hex table format

Reference N/A

Standard Scope Optional

Enfora Implementation Scope Full

Notes N/A

3.13 $GPSRD Read current GPS ASCII data (continued)

Example

To read RMC NMEA sentence in HEX format, one would send:
AT$GPSRD=10.

To read RMC NMEA sentence in decimal format, one would send
AT$GPSRD=16,1

To read All (GGA, GLL, GSA, GSV, RMC, VTG) NMEA sentences in hex format, send AT$GPSRD=3f
(3f = 0x01 + 0x02 + 0x04 + 0x08 + 0x10 + 0x20)

To read All NMEA sentences in decimal format, send AT$GPSRD=63,1
(63 = 1 + 2 + 4 + 8 + 16 + 32)

3.14 Interval Wakeup Timer

$WAKEINTVL	**Interval Wakeup Timer**
Command Function	This command allows the user to set/query a timer used to periodically wake up the modem at a user-defined interval (in minutes).
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$WAKEINTVL=? $WAKEINTVL:(0-65535) OK
Write Format **Response**	AT$WAKEINTVL=**<interval>** OK
Read Format **Response**	AT$WAKEINTVL? $WAKEINTVL:**<interval>**
Execution Format **Response**	N/A
Parameter Values	
<interval>	Number of minutes between each attempt to wake up the modem (if modem is already awake, no action is taken). If value is zero, the wake interval feature is disabled.
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	This command relays the <interval> value to the MSP430. If the value is non-zero, the MSP430 starts a counter, counting down from the interval value. When the count reaches zero, the microprocessor wakes the modem if the modem is inactive (asleep). Regardless

3.14 $WAKEINTVL Interval Wakeup Timer (continued)

of whether it needs to wake the modem or not, the microprocessor resets it count and starts the cycle all over again.

The interval value is also stored by the modem in non-volatile memory. So if the battery is removed (therefore causing the microprocessor to lose its memory of the interval value), the modem relays the interval value to the MSP430 at power-up (when the battery is installed) and the cycle starts again at this point.

This command is completely independent of the $wakeenbl and $wakertc commands. Although it would normally be used instead of the $wakeenbl and $wakertc commands, you can use this command in conjunction with the other commands.

For example, let's suppose you wanted the modem to be awake whenever the device was in motion, but you also wanted the modem to wake up for at least five minutes once an hour even if there was no motion. You could use the following commands:

(Wake up any time there is motion)
at$wakeenbl=4
(Stay awake for at least five minutes)
at$waketime=300
(Must detect no motion for 120 consecutive seconds before declaring unit stopped)
at$mottrans=120 (default)
(Check modem every 60 minutes and wake it up, if asleep)
at$wakeintvl=60

3.15 Keybeep Sound

$KEYSND	**Keybeep Sound Setting**
Command Function	This command allows the user to enable/disable the internal keybeep sounds that are played when a button is pressed. This allows the user to set up custom keybeep sounds using the event engine.
Command Functional Group	Enfora Specific
Command Format Query	AT$KEYSND=?
Response	$KEYSND:(0-1) OK
Write Format	AT$KEYSND=**<setting>**
Response	OK
Read Format	AT$KEYSND?
Response	$KEYSND:**<setting>**
Execution Format	N/A
Response	
Parameter Values	
<setting>	0 – play keybeep sounds hard-coded in the software (default) 1 – disable internal keybeep sounds
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	

3.15 $KEYSND — Keybeep Sound Setting (continued)

Example

Use the following commands to set up custom keybeep sounds using the at+sttone command:

(Disable default keybeep sounds)
at$keysnd=1

(Generic keybeep sound)
at$stoatev=3,at+stone=1,16,100

(Volume Up button event)
at$event=95,1,58,4,4
at$event=95,3,44,3,0

(Volume Down button event)
at$event=96,1,58,3,3
at$event=96,3,44,3,0

(Push-To-Call button event)
at$event=97,1,58,2,2
at$event=97,3,44,3,0

(User-Defined button event)
at$event=98,1,58,1,1
at$event=98,3,44,3,0

(Geofence button event – no GPS lock)
at$stoatev=2,at+sttone=1,18,100
at$event=99,1,58,0,0
at$event=99,2,27,0,0
at$event=99,3,44,2,0

(Geofence button event – GPS lock)
(at$event=1,1,58,0,0 is factory default)
(at$event=1,49,3,1,805 is factory default)
at$stoatev=1,at+sttone=1,17,100
at$event=1,2,27,1,1
at$event=1,3,44,1,0

3.16 Key Function Disable

$KEYFNC `	**Key Function Disable**
Command Function	The $KEYFNC command enables the user to disable the hard-coded actions of the Mini-MT buttons (initiate phone call via PTC button, increase/decrease volume via +/- buttons, and play special tones for geo-fnc button depending on whether GPS data is currently valid). The buttons can still be used via the event engine (see EVENT) when the default hard-coded actions are disabled (for example, to execute a user-defined action and play a customized tone).
Command Functional Group	Enfora Specific
Command Format Query Response	AT$KEYFNC=? $KEYFNC: (0-1) OK
Write Format Response	AT$KEYFNC=<status> OK
Read Format Response	AT$KEYFNC? $KEYFNC:<status> OK
<status>	**0** = hard-coded key functions enabled (default) **1** = hard-coded key functions disabled
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.16 $KEYFNC Key Function Disable (continued)

Notes	When $KEYFNC=1, pressing PTC button will still wake the modem depending on the $WAKEENBL setting (see Motion Wake Enable). The default $WAKEENBL value of 20 wakes modem on motion or PTC button press. Also note that button sounds are disabled via the $KEYSND command (see Keybeep Sound).

3.17 LTO Download

$LTODL

Initiate and query the status of an LTO download and LTO data

Command Function

This command allows the user to initiate an LTO download, and query the status of an LTO download and of the LTO data. An LTO download cannot be initiated if a download is already in progress, or in the absence of a GPRS registration.

Command Functional Group

Enfora Specific

Command Format Query Response

AT$LTODL=?
$LTODL
OK

Write Format Response

N/A
OK

Read Format Response

AT$LTODL?
$LTODL: <***lto_valid***>, <***dnld_sts***>, <***failures***>, <***dnld_year***>, <***dnld_month***>, <***dnld_day***>, <***dnld_hour***>, <***dnld_min***>, <***dnld_sec***>,
OK

Execution Format Response

AT$LTODL
OK

Parameter Values

<***lto_valid***>

0 – Current LTO data is not valid
1 – Current LTO data is valid

<***dnld_sts***>

0 – LTO download is not in progress
1 – LTO download is in progress

<***failures***>

Number of LTO download failures since last successful download

3.17 $LTODL — LTO Download (continued)

<***dnld_year***>	Year of completion of last LTO download
<***dnld_month***>	Month of completion of last LTO download
<***dnld_day***>	Day of completion of last LTO download
<***dnld_hour***>	Hour of completion of last LTO download
<***dnld_min***>	Minute of completion of last LTO download
<***dnld_sec***>	Second of completion of last LTO download
Notes	If the download completes when the modem does not know the time, the time reported will be the age of the LTO data in hours, minutes, seconds.

Example:

Events that will display when an LTO download starts, completes or fails.

```
at$event=8,1,64,1,1
at$event=8,3,44,8,0
at$stoatev=8,at$msgsnd=0,"LTO started"
at$event=9,1,64,2,2
at$event=9,3,44,9,0
at$stoatev=9,at$msgsnd=0,"LTO complete"
at$event=10,1,64,3,3
at$event=10,3,44,10,0
at$stoatev=10,at$msgsnd=0,"LTO failed"
```

3.18 Message Log Clear

$MSGLOGCL	**Message Log Clear**
Command Function	The $MSGLOGCL command erases the log file.
Command Functional Group	Enfora Specific
Command Format Query **Response**	N/A N/A
Write Format **Response**	N/A N/A
Read Format **Response**	N/A N/A
Execution Format **Response**	AT$MSGLOGCL OK
Parameter Values	N/A
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	N/A

3.19 Message Log Enable

$MSGLOGEN	**Message Log Enable**
Command Function	The $MSGLOGEN command has been created to enable or disable saving GPS data generated via the event engine in modem’s memory
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$MSGLOGEN=? $MSGLOGEN: (0-1) OK
Write Format **Response**	AT$MSLOGEN=<**setting>** OK
Read Format **Response**	AT$MSGLOGEN? $MSGLOGEN: **<setting>** OK
Execution Format **Response**	N/A N/A
Parameter Values	
<setting> =	**0 – 1** (possible valid values) **0** = Disable message logging (default). Event data is sent to the remote server upon occurrence. **1** = Enable message logging. Event data has to be read via AT$MSGLOGRD command or when AT$MSGLOGEN=0 is sent.
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.19 $MSGLOGEN	**Message Log Enable (continued)**
Notes	If AT$MSGLOGEN command was enabled and any unsent messages exist in memory, then the unsent data will be sent to the remote server when data logging is disabled.

3.20 Message Log Read Data

$MSGLOGRD	**Message Log Read Data**
Command Function	The $MSGLOGRD command has been created to read data from memory.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$MSGLOGRD=? $MSGLOGRD: (0-2),(0-x),(0-y) OK
Write Format **Response**	N/A N/A
Read Format **Response**	AT$MSGLOGRD? $MSGLOGRD: **<queue>,<number of messages>,<starting index>** OK
Execution Format **Response**	N/A N/A
Parameter Values	
<queue> =	0 – 2 (possible valid values). 0 = event data that was configured to be sent to a remote server via GPRS only 1 = event data that was configured to be sent to a remote server via GPRS primarily but also use SMS as backup method if GPRS is not available 2 = event data that was configured to be sent to a remote server via SMS only

3.20 $MSGLOGRD Message Log Read Data (continued)

<number of messages> = x

x = total number of messages one desires to read from the memory. A user can choose to read 1 message in which case x = 1 or read all messages in which case x = 65535.

<starting index> = y

y = starting index number of messages that are stored in the memory.

NOTE: y cannot be greater than maximum number of stored messages.

Reference N/A

Standard Scope Optional

Enfora Implementation Scope Full

Notes

AT$MSGLOGRD? command returns 8 values. The first two values correspond to data stored for the GPRS queue. The next two values correspond to data stored for SMS AS BACKUP queue, the next two values correspond to data stored for SMS queue, and the last two values correspond to data stored for the TCP API queue.

- Each value is comma (,) delimited.
- The first value of any queue represents “Total Number of Unread Messages”. This value can be used as the <number of messages> field while reading messages
- The second value of any queue represents: “Total Number of Messages Stored for that Queue”. Subtract the “Total Number of Unread Messages” from the “Total

3.20 $MSGLOGRD Message Log Read Data (continued)

Number of Messages Stored for that Queue” and use that as the <starting index> of where to read data from in the memory.

3.21 Message Send

$MSGSND	**Message Send**
Command Function	The $MSGSND command has been created to allow sending of data from one mode to another.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$MSGSND=? $MSGSND: (0-4),("ASCII DATA") OK
Write Format **Response**	AT$MSGSND=**<destination>,<"data">** OK
Read Format **Response**	N/A N/A
Execution Format **Response**	N/A N/A
Parameter Values	
<destination> =	0 – 4 (possible Valid Values) 0 = <"data"> is sent out the serial port 1 = <"data"> is sent to all SMS addresses listed in AT$SMSDA command. 2 = <"data"> is sent via GPRS to first IP address, configured as server, in AT$FRIEND command and port number defined by AT$UDPAPI command 3 = <"data"> is sent via GPRS to IP address and Port number listed in the AT$PADDST command 4 = <"data"> is sent via TCP API
<"data"> =	a maximum of 99 bytes ASCII characters

3.21 $MSGSND	**Message Send (continued)**
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	N/A

3.22 Modem Wake Duration

$WAKETIME

Control time that modem is in active state

Command Function

This command allows the user to set/query the amount of time that the modem will remain in the active state before going into Mini-MT “sleep” mode. When the Mini-MT is in sleep mode, the modem is completely shut down to conserve power. The MSP430 is responsible for re-awakening the modem at the appropriate time based on user’s configuration settings.

Once a second, the modem executes the code to determine if it is time to sleep based on the following algorithm:

1. Internal waketime count is incremented.

2. If there is an active call or ringing state, code will exit and modem will not initiate sleep command.

3. If device is currently in moving state and $wakeenbl is set to wake modem on moving state, code will exit and modem will not initiate sleep command.

4. If device is currently in stopped state and $wakeenbl is set to wake modem on stopped state, code will exit and modem will not initiate sleep command.

5. If external (USB) power is currently applied, flag will be set, code will exit and modem will not initiate sleep command. If external power is removed and flag is set, modem will reset internal waketime count to 0.

3.22 $WAKETIME Control time that modem is in active state (continued)

6. If $waketime value = 0, code will exit and modem will not initiate sleep command.

7. If internal waketime count less than $waketime value, code will exit and modem will not initiate sleep command.

8. If waketime count greater than $waketime value and all checks above have passed, modem will initiate sleep command. This is the equivalent of executing the at$off command which allows the modem to shut down gracefully (de-register from network, save current status to flash, update MSP430 with current date/time, etc.)

Command Functional Group
Enfora Specific

Command Format Query
Response
AT$WAKETIME=?
$WAKETIME:(0-4294967295)
OK

Write Format
Response
AT$WAKETIME=**”<waketime>”**
OK

Read Format
Response
N/A

Execution Format
Response
N/A

Parameter Values

<waketime>
Time in seconds that modem will be active before shutting down to conserve power. If <waketime>is 0 (default), modem will stay active indefinitely.

Reference
N/A

3.22 $WAKETIME — Control time that modem is in active state (continued)

Standard Scope Optional

Enforа Implementation Scope Full

Notes N/A

Example

To set up the modem to stay awake while device is in moving state and sleep while device is in stopped state, use the following commands:

```
AT$WAKETIME=60
AT$MOTTRANS=120
AT$WAKEENBL=4
```

Modem will wake up on motion ($WAKEENBL=4) and stay in motion for at least 120 seconds ($MOTTRANS=120). Since $waketime count (60) expires before the $mottrans, code will not initiate sleep while motion state is still moving. When motion state transitions to moving, $waketime algorithm will immediately be able to initiate sleep command.

3.23 Motion Transition Count

$MOTTRANS	**Motion Transition Count**
Command Function	This command allows the user to set/query the motion transition count used to declare an intermediate period between the “moving” and “stopped” states. For example, this might be used to inhibit the immediate transition from “moving” to “stopped” when a vehicle is waiting at a red light. If transition count = 0, then the device can transition from “moving” to “stopped” immediately upon crossing to $MOTSTOP threshold.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$MOTTRANS=? $MOTTRANS:(0-65535) OK
Write Format **Response**	AT$MOTTRANS=**<transition count>** OK
Read Format **Response**	AT$MOTTRANS? $MOTTRANS:**<transition count>** OK
Execution Format **Response**	N/A N/A
Parameter Values	
<transition count>	Number of seconds to remain in moving state as long as filtered motion count is below motion stop threshold before declaring the “stopped” state. If transition count = 0, state will change directly to “stopped” from “moving” as soon as motion stop threshold is crossed.

3.23 $MOTTRANS Motion Transition Count (continued)

Reference N/A

Standard Scope Optional

Enfora Implementation Scope Full

Notes

The Mini-MT motion sensor generates interrupts based on movement of a ball-bearing in the sensor device. Each interrupt increments a motion count. Once a second the MSP430 takes the motion count, clips it if necessary, and filters it to produce a filtered motion count.

The filtered motion count is computed as follows:

Filtered motion count =
((coef A * previous motion filter count) +
(coef B * clipped current motion count)) /
65536

The filtered motion count is compared to the motion start threshold (if in the stopped state) or motion stop threshold (if in the moving state) to determine if a state change is required. These state changes can be responsible for waking the modem if configured by the user ($WAKEENBL).

3.24 Motion Wake Enable

$WAKEENBL	**Motion Wake Enable**
Command Function	This command allows the user to set/query the optional conditions used by the MSP430 to wake the modem. The modem will always be activated by a Power-On Reset, application of external (USB) power, and a press of the Push-To-Call (PTC) button when the modem is asleep. The optional wake conditions controlled via this command include motion state transitions and motion timer expirations.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$WAKEENBL=? $WAKEENBL:(0-15) OK
Write Format **Response**	AT$WAKEENBL=**<wake conditions>** OK
Read Format **Response**	AT$WAKEENBL? $WAKEENBL:**<wake conditions>** OK
Execution Format **Response**	N/A N/A
Parameter Values	
<wake conditions>	**1** – transition from “stopped” to “moving” state **2** – transition from “moving” to “stopped” state **4** – current state is “moving” (no transition required) **8** – current state is “stopped” (no transition required)

3.24 $WAKEENBL — Motion Wake Enable (continued)

Multiple wake conditions can be specified by adding these values. For example, AT$WAKEENBL=12 would enable MSP430 to wake modem for transition to “stopped” or “moving” state (4 +8 = 12).

Reference N/A

Standard Scope Optional

Enfora Implementation Scope Full

Notes The Mini-MT motion sensor generates interrupts based on movement of a ball-bearing in the sensor device. Each interrupt increments a motion count. Once a second the MSP 430 takes the motion count, clips it if necessary, and filters it to produce a filtered motion count.

The filtered motion count is computed as follows:

Filtered motion count =
((coef A * previous motion filter count) +
(coef B * clipped current motion count)) /
65536

3.24 $WAKEENBL

Motion Wake Enable (continued)

The filtered motion count is compared to the motion start threshold (if in the stopped state) or motion stop threshold (if in the moving state) to determine if a state change is required. These state changes can be responsible for waking the modem if configured by the user ($WAKEENBL).

Multiple wake conditions can be specified by adding these values. For example, AT$WAKEENBL=3 would enable MSP430 to wake modem for transition to "stopped" or "moving" state (1 + 2 = 3). However, normally only one of the bits will be used.

3.25 Odometer

$ODOMETER	**MT Trip Odometer**
Command Function	The $ODOMETER command records how far the vehicle has traveled in one trip. The user can reset the odometer at the beginning of a new trip.
Command Functional Group	Enfora Specific
Command Format Query	AT$ODOMETER=?
Response	$ODOMETER: (0-4000000000) OK
Write Format	AT$ODOMETER=1234 (where 1234 is distance in meters)
Response	OK
Read Format	AT$ODOMETER?
Response	$ODOMETER xxxx (xxxx=distance traveled in meters) OK
Execution Format	N/A
Response	N/A
Parameter Values	N/A
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.25 $ODOMETER

Odometer (continued)

Notes

The user shall be able to set a seed value for the Virtual Odometer (including a value of 0 but not higher than the maximum value of 4000000000)

The AT&F command shall not reset the seed value to 0.

The Virtual Odometer reading would be a 4-byte value starting from 0 to 4000000000 (maximum of approximately 2500000 miles before it rolls over to 0)

The unit for Virtual Odometer shall be in METERS.

The Virtual Odometer history shall be updated every second

The Virtual Odometer history shall be saved once a minute in modem's memory. This value shall be retained through an internal or external reset and can be read upon the next power up or during run time mode. The delta distance traveled between the minute marks could be lost due to an unexpected external or non-modem originated reset. However, the total distance traveled till the prior minute would still be preserved.

The AT+ODOMETER feature will only work if the Mini-MT is receiving power at all times

3.26 Ring Indicator

$RINGIND	**Ring Indicator**
Command Function	This command configures the Mini-MT to either ring, vibrate, or remain silent when a call comes in.
Command Functional Group	Enfora Specific
Command Format Query	AT$RINGIND = ?
Response	$RINGIND: (0-2) OK
Write Format	AT$RINGIND = <option>
Response	OK
Read Format	AT$RINGIND?
Response	RINGIND: <option>
Execution Format	N/A
Response	N/A
Parameter Values	
<option>	0 = ring 1 = vibrate 2 = off (neither ring nor vibrate)
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	N/A

3.27 RTC Wakeup Timer

$WAKERTC	**RTC Wakeup Timer**
Command Function	This command allows the user to set/query a wakeup timer that is initially set based on RTC inputs.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$WAKERTC=? $WAKERTC:(0,,6),(0..99),(1..12),(1..31),(0..23),(0..59),(0..59) OK
Write Format	AT$WAKERTC=**<rtc_wkday>,<rtc_year>,<rtc_month>,<rtc_day>,<rtc_hour>,<rtc_min>,<rtc_sec>**
Response	OK
Read Format **Response**	AT$WAKERTC? $WAKERTC: **<mins_left>** OK
Execution Format **Response**	N/A N/A
Parameter Values	Parameters are positional dependent, any parameter may be omitted with the use of the comma (‘,’) as a place holder on command line. If a parameter is omitted then the current ($RTCTIME) value in the hardware is used.
<rtc_wkday>	Current week day matching time day being set. The week day values range from 0..6, where: 0 –> Sunday, 1 -> Monday, 2-> Tuesday, 3 -> Wednesday, 4 -> Thursday, 5 -> Friday, and 6 -> Saturday

3.27 $WAKERTC	RTC Wakeup Timer (continued)
<rtc_year>	The year on which the time is being set to. The RTC supports years 2000-2099. The data is entered as a two digit value 0..99.
<rtc_month>	The month on which the time is being set to. Values range from 1..12.
<rtc_hour>	The hour on which the time is being set to. Values range from 0 to 33.
<rtc_min>	The minute on which the time is being set to. Values range from 0..59.
<rtc_sec>	The second on which the time is being set to. Values range from 0..59.
<mins_left>	The number of minutes remaining before the $WAKERTC timer will expire.
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.27 $WAKERTC RTC Wakeup Timer (continued)

Notes

When setting the $WAKERTC timer, the RTC inputs are compared against the current ($RTCTIME) hardware values to determine the number of minutes remaining until the target time arrives. (If modem is already awake when target time arrives, no action is taken.) Therefore, it is important that the $RTCTIME values have been set properly before executing this command (this normally happened upon initial GPS acquisition each time the modem is activated). This calculated value is loaded into the MSP430, which starts counting down until the number of minutes has expired. The $WAKERTC inputs are not stored in persistent memory, so if the power is removed (battery and USB connected both pulled), then the $WAKERTC settings will be lost and will have to be re-entered.

3.28 Vibration Command

$VIBNOW	**Exercise Vibration Motor**
Command Function	This command allows the user to exercise the vibration motor for a user-defined number of seconds.
Command Functional Group	Enfora Specific
Command Format Query **Response**	AT$VIBNOW=? $VIBNOW:(1-255) OK
Write Format **Response**	AT$VIBNOW=**<secs>** OK
Read Format	N/A
Execution Format **Response**	N/A
Parameter Values	
<secs>	number of seconds the vibration motor will be active.
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full
Notes	

3.29 Writing GPS NV RAM

$GPSFLASH

Specify and query the time interval of the flushing of GPS NVRAM to the FFS

Command Function

This command allows the user to specify the interval, in minutes, at which the modem will stop positioning, write GPS' NVRAM to the FFS and then resume positioning.

Command Functional Group

Enfora Specific

Command Format Query Response

AT$GPSFLASH=?
$GPSFLASH: (0-1440)
OK

Write Format Response

AT$GPSFLASH=(0-1440)
OK

Read Format Response

AT$GPSFLASH?
$GPSFLASH: <***interval***>, <***remaining***>
OK

Execution Format Response

AT$GPSFLASH
ERROR

Parameter Values

<***interval***>

Interval in minutes at which the modem will stop positioning, write GPS' NVRAM to the FFS and then resume positioning.

<***remaining***>

Seconds remaining in the current interval. It will be this many seconds until Whistler stops positioning, writes GPS' NVRAM to the FFS and then resumes positioning.

Notes

If the interval is set to 0, then the Whistler will never stop positioning to write NVRAM.

4 Appendix A – Default AT Values

$BATTLVL	**Battery Level**
Default Value:	N/A
Default Value Meaning:	N/A

$DLYCALL	**Delay Call**
Default Value:	0
Default Value Meaning:	No user-defined delay between pressing the PTC button and placing a call.

$DSPATCH	**Dispatch Notification Phone Number**
Default Value:	12345678123456789
Default Value Meaning:	This is the number the Mini-MT will call when the PTC button is pressed.

$DSPNOTIF	**Dispatch Notification**
Default Value:	0
Default Value Meaning:	Dispatch notification is not active.

$EMERNUM	**Emergency Number**
Default Value:	411
Default Value Meaning:	This is the emergency number the Mini-MT will call when the emergency call sequence is performed.

$EVENT	**User-defined Input/Output**
Default Value:	Set Geofence at current location
Default Value Meaning:	Event group 1 configured for half-mile geofence when pressing geofence button.

$GEOFNC	**Geo fencing a circle area**
Default Value:	1,0,0,0 – (default for push button) 2,0,0,0 3,0,0,0 4,0,0,0 5,0,0,0 6,0,0,0 7,0,0,0 8,0,0,0 9,0,0,0 10,0,0,0 11,0,0,0 12,0,0,0 13,0,0,0

	14,0,0,0
	15,0,0,0
	16,0,0,0
	17,0,0,0
	18,0,0,0
	19,0,0,0
	20,0,0,0
	21,0,0,0
	22,0,0,0
	23,0,0,0
	24,0,0,0
	25,0,0,0
Default Value Meaning:	feature disabled
$GPSLCL	**GPS Local Subscription**
Default Value:	0
Default Value Meaning:	feature disabled
$GPSRD	**Read current GPS NMEA data**
Default Value:	N/A
Default Value Meaning:	N/A
$MOTTRANS	**Motion Transition Count**
Default Value:	120
Default Value Meaning:	N/A
$MSGLOGCL	**Message Log Clear**
Default Value:	N/A
Default Value Meaning:	N/A
$MSGLOGEN	**Message Log Enable**
Default Value:	0
Default Value Meaning:	Message log is enabled
$MSGLOGRD	**Message Log Read Data**
Default Value:	N/A
Default Value Meaning:	N/A
$MSGSND	**Message Send**
Default Value:	N/A
Default Value Meaning:	N/A
$MSPCNT	**MSP430 Raw Motion Count**
Default Value:	N/A
Default Value Meaning:	N/A

$MSPFLASH	**MSP430 Firmware Update**
Default Value:	N/A
Default Value Meaning:	N/A
$MSPVER	**MSP Version**
Default Value:	N/A
Default Value Meaning:	N/A
$ODOMETER	**Trip Odometer**
Default Value:	N/A
Default Value Meaning:	N/A
$RINDIND	**Ring Indicator**
Default Value:	0
Default Value Meaning:	The Mini-MT is configured for audible ring for incoming calls
$WAKEENBL	**Motion Wake Enable**
Default Value:	0
Default Value Meaning:	
$WAKEINTVL	**Interval Wakeup Timer**
Default Value:	0
Default Value Meaning:	Mini-MT will not be scheduled to wakeup at an interval.
$WAKERTC	**RTC Wakeup Timer**
Default Value:	
Default Value Meaning:	The Mini-MT is not configured to wake up based on future date/time.
$WAKETIME	**Modem Wake Duration**
Default Value:	0
Default Value Meaning:	Mini-MT will always be awake

5 Appendix B – AT Commands over SMS

The Mini-MT supports the following commands over SMS:

$MSPIO
$MOTSTART
$MOTSTOP
$MOTCLIP
$MOTCOEF
$WAKEENBL
$MOTTRANS
$WAKEINTVL
$WAKERTC
$MMTTEST
$DSPNOTIF
$DLYCALL
$MSPFLASH
$DSPATCH
$EMERNUM
$WAKETIME
$RINGIND
$HWVER