



GSM2228PB001MAN

Enfora® Mini-MT
AT Command Set Reference
Release 1.00

Confidential and Proprietary Information - © 2007 Enfora, Inc.
Do not duplicate without express permission from Enfora, Inc.

Enfora, Inc. 661 E. 18th Street Plano Texas 75074
www.enfora.com

Document Title:	Enfora GSM/GPRS Mini-MT AT Command Set Reference
Version:	1.00
Date:	1/09/2007
Status:	Released
Document Control ID:	GSM2228PB001MAN

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.

©2007 Enfora Inc. All rights reserved.

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	5
3.3	Dispatch Notification	6
3.4	Dispatch Phone Number.....	7
3.5	Emergency Phone Number	8
3.6	EVENT.....	10
3.7	Geo-Fencing.....	19
3.8	GPS Local Subscription.....	21
3.9	GPS Read	24
3.10	Interval Wakeup Timer.....	27
3.11	Message Log Clear	28
3.12	Message Log Enable	29
3.13	Message Log Read Data	31
3.14	Message Send.....	34
3.15	Modem Wake Duration.....	36
3.16	Motion Clip Threshold.....	38
3.17	Motion Filter Coefficients	40
3.18	Motion Start Threshold	42
3.19	Motion State	44
3.20	Motion Stop Threshold.....	45
3.21	Motion Transition Count.....	47
3.22	Motion Wake Enable	49
3.23	MSP430 Firmware Update	52
3.24	MSP430 Firmware Version.....	54
3.25	MSP430 Raw Motion Count	55
3.26	Odometer.....	57
3.27	Ring Indicator	59
3.28	RTC Wakeup Timer	60
4	Appendix A – Default AT Values	63
5	Appendix B – AT Commands over SMS	67

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 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

\$BATTTLVL	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 Response	N/A N/A
Write Format Response	N/A N/A
Read Format Response	AT\$BATTTLVL? \$BATTTLVL: <percentage> OK
Execution Format Response	N/A N/A
Parameter Values	
<percentage>	Approximate percentage of battery life remaining.
Reference	N/A
Standard Scope	Optional
Enfora Implementation Scope	Full

3.2 Call Delay

\$DLYCALL

Command Function

Call Delay

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?
\$MOTCNT: <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

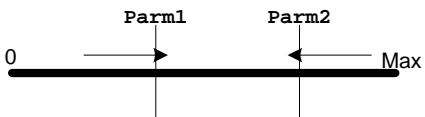

<event group>

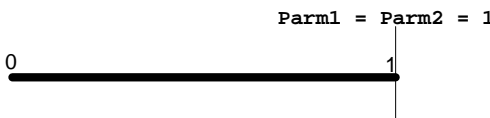
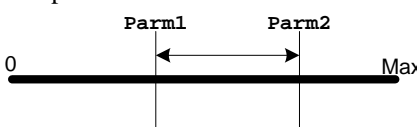
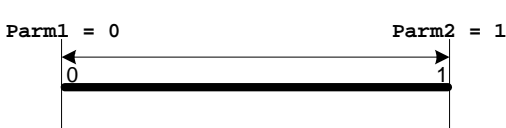
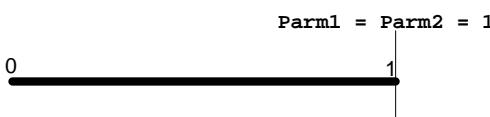
This parameter defines the group number of a group of events and the order they are executed. Events are grouped together to control execution sequence. A group number has to have at least one input event and one output event. Multiple input events within a group number would be treated as a logical **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	<i>Transition Trigger</i>	<p>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>.</p> <p>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.</p> <p>Example 1:</p>  <p>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).</p> <p>Example 2:</p> <p>Parm1 = Parm2 = 0</p>  <p>Figure 2. An output event will be executed when the value of an input event is 0 (previously it was</p>

		<p>anything else but 0) and <Parm1> along with <Parm2> is set to 0.</p> <p>Example 3:</p>  <p>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.</p>
1	Occurrence Trigger	<p>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</p> <p>Example 4:</p>  <p>Figure 4. An output event will be executed when the current value of an input event is between <Parm1> and <Parm2> including boundary conditions.</p> <p>Example 5:</p>  <p>Figure 5. An output event will be executed when the value of the input event changes from 0 to 1 or vice-versa.</p> <p>Example 6:</p>  <p>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.</p>
2	Input Trigger	<p>An Input Trigger is defined as an input condition, defined by <event category>, that should be</p>

		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	<i>Output</i>	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: seconds)
	30	0 to 1000000	1000000	Unit staying Idle in one place (unit of measurement is: seconds)
	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	1-25	0-100000	Set Geo Fence (must have GPS lock to have any effect). Parm 1 is GeoFence index number. Parm2 is radius of circle from current Latitude and Longitude coordinates. (unit of measurement is meters)
	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%)
	60	0-2147483647	0-2147483647	Number of unsent messages
	61	0-100	0-100	Memory full percentage
	62	0-1	0	Motion Status 1 = moving 0 = stopped
	63	0-1	0	Power Source 1 = External power 0 = Battery power

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- 2147483 647	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- 2147483 647	See Bit- Field Table below	Sends data over SMS. SMS destination address has to be configured via \$SMSDA command
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

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

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=< <i>fenceNum</i> >,< <i>radius</i> >,< <i>latitude</i> >,< <i>longitude</i> > OK
Read Format Response	AT\$GEOFNC? \$GEOFNC: < <i>fenceNum</i> >,< <i>radius</i> >,< <i>latitude</i> >,< <i>longitude</i> > OK
Execution Format Response	N/A N/A
Parameter Values	
< <i>fenceNum</i> >	Defines the fence number
< <i>radius</i> >	Defines radius of the circle from given Latitude and Longitude coordinates
< <i>latitude</i> >	Defines the latitude for the center point of a circle
< <i>longitude</i> >	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 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=< <i>option</i> >,<nmeaMsgs> OK or ERROR
Read Format Response	AT\$GPSLCL? \$GPSLCL: < <i>option</i> >,<nmeaMsgs> OK
Execution Format Response	N/A N/A

3.8 \$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

3.8 \$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

3.9 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

AT\$GPSRD=<*nmeaMsgs*>,<*decimal*>
"\$GPG....."

Response

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.

3.9 \$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

3.9 \$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)

3.10 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>
OK

Execution Format Response

N/A
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).

Reference

N/A

Standard Scope

Optional

Enfora Implementation Scope

Full

Notes

N/A

3.11 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

3.12 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\$MSGLOGEN=<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\$MSGLOGEN command or when AT\$MSGLOGEN=0 is sent.

Reference

N/A

Standard Scope

Optional

Enfora Implementation Scope

Full

3.12 \$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.

3.13 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

3.13 \$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

3.13 \$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.

3.14 Message Send

\$MSGSEND

Message Send

Command Function

The \$MSGSEND command has been created to allow sending of data from one mode to another.

Command Functional Group

Enfora Specific

Command Format Query Response

AT\$MSGSEND=?
\$MSGSEND: (0-4),("ASCII DATA")
OK

Write Format Response

AT\$MSGSEND=<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

3.14 \$MSGSEND

Message Send (continued)

Reference

N/A

Standard Scope

Optional

Enfora Implementation Scope

Full

Notes

3.15 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.

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

AT\$WAKETIME?
\$WAKETIME:<waketime>
OK

Execution Format Response

N/A
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

Standard Scope

Optional

3.15 \$WAKETIME

**Control time that modem is in active
state
(continued)**

Enfora Implementation Scope

Full

Notes

N/A

3.16 Motion Clip Threshold

\$MOTCLIP

Motion Clip Threshold

Command Function

This command allows the user to set/query the motion clip threshold used in the computation to declare motion states.

Command Functional Group

Enfora Specific

Command Format Query Response

AT\$MOTCLIP=?
\$MOTCLIP:(0-65535)
OK

Write Format Response

AT\$MOTCLIP=<clip threshold>
OK

Read Format Response

AT\$MOTCLIP?
\$MOTCLIP:<clip threshold>
OK

Execution Format Response

N/A
N/A

Parameter Values

<clip threshold>

Threshold used to clip once-a-second motion interrupt counts prior to performing computations to determine motion state. Default = 0 which keeps Mini-MT in “stopped” state indefinitely.

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

3.16 \$MOTCLIP

Motion Clip Threshold (continued)

motion count (see \$MOTCNT), clips it if necessary (see \$MOTCLIP), and filters it (see \$MOTCOEF) to produce a filtered motion count.

The filtered motion count is computed as follows:

$$\begin{aligned} \text{Filtered motion count} = & \\ & ((\text{coef A} * \text{previous motion filter} \\ & \quad \text{count}) + \\ & \quad (\text{coef B} * \text{clipped current motion} \\ & \quad \text{count})) / \\ & 65536 \end{aligned}$$

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 (see \$MOTSTAT and \$MOTTRANS). These state changes can be responsible for waking the modem if configured by the user (\$WAKEENBL).

3.17 Motion Filter Coefficients

\$MOTCOEF	Motion Filter Coefficients
Command Function	This command allows the user to set/query the filter coefficients used to declare motion states.
Command Functional Group	Enfora Specific
Command Format Query Response	AT\$MOTCOEF=? \$MOTCOEF:(0-65535), (0-65535) OK
Write Format Response	AT\$MOTCOEF="<coef A>,<coef B>" OK
Read Format Response	AT\$MOTCOEF? \$MOTCOEF:<coef A>,<coef B> OK
Execution Format Response	N/A
Parameter Values	
<coef A>	Coefficient that assigns weight to be given to previous motion filter counts.
<coef B>	Coefficient that assigns weight to be given to current motion count (motion interrupts in the last second).
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

3.17 \$MOTCOEF

Motion Filter Coefficients (continued)

interrupt increments a motion count. Once a second the MSP430 takes the motion count (see \$MOTCNT), clips it if necessary (see \$MOTCLIP), and filters it (see \$MOTCOEF) to produce a filtered motion count.

The filtered motion count is computed as follows:

$$\begin{aligned} \text{Filtered motion count} = & \\ & (\text{coef A} * \text{previous motion filter} \\ & \quad \text{count}) + \\ & (\text{coef B} * \text{clipped current motion} \\ & \quad \text{count}) / \\ & 65536 \end{aligned}$$

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 (see \$MOTSTAT and \$MOTTRANS). These state changes can be responsible for waking the modem if configured by the user (\$WAKEENBL).

3.18 Motion Start Threshold

\$MOTSTART

Motion Start Threshold

Command Function

This command allows the user to set/query the threshold used to declare that Mini-MT is in the “moving” state.

Command Functional Group

Enfora Specific

Command Format Query Response

AT\$MOTSTART=?
\$MOTSTART:(0-65535)
OK

Write Format

AT\$MOTSTART="**<motion start
threshold>**"

Response

OK

Read Format Response

AT\$MOTSTART?
\$MOTSTART:**<motion start
threshold>**
OK

Execution Format Response

N/A
N/A

Parameter Values

<motion start threshold>

Threshold count used in MSP430 to declare Mini-MT is in “moving” state. See Notes.

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 (see \$MOTCNT), clips it if

3.18 \$MOTSTART

Motion Start Threshold (continued)

necessary (see \$MOTCLIP), and filters it (see \$MOTCOEF) to produce a filtered motion count.

The filtered motion count is computed as follows:

$$\begin{aligned} \text{Filtered motion count} = & \\ & ((\text{coef A} * \text{previous motion filter} \\ & \quad \text{count}) + \\ & \quad (\text{coef B} * \text{clipped current motion} \\ & \quad \text{count})) / \\ & 65536 \end{aligned}$$

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 (see \$MOTSTAT and \$MOTTRANS). These state changes can be responsible for waking the modem if configured by the user (\$WAKEENBL).

3.19 Motion State

\$MOTSTAT

Motion State

Command Function

This command allows the user to query the motion state.

Command Functional Group

Enfora Specific

Command Format Query Response

N/A
N/A

Write Format Response

N/A
N/A

Read Format Response

AT\$MOTSTAT?
\$MOTSTAT:<state>
OK

Execution Format Response

N/A
N/A

Parameter Values

<state>

0 “stopped” state
1 “moving” state

Reference

N/A

Standard Scope

Optional

Enfora Implementation Scope

Full

Notes

The Mini-MT maintains two motion states: “moving,” and “stopped.” Each state has its own timer that is reset when the state is entered, when the modem is deactivated, or when its timer expires. The modem can be re-activated by any motion state transition or expiration or expiration of any of the timers, based on user configuration.

3.20 Motion Stop Threshold

\$MOTSTOP

Motion Stop Threshold

Command Function

This command allows the user to set/query the threshold used to declare that Mini-MT is in the “stopped” or “transition” state.

Command Functional Group

Enfora Specific

Command Format Query Response

AT\$MOTSTOP=?
\$MOTSTOP:(0-65535)
OK

Write Format

AT\$MOTSTOP=”<motion stop
threshold>”

Response

OK

Read Format Response

AT\$MOTSTOP?
\$MOTSTOP:<motion stop
threshold>
OK

Execution Format Response

N/A
N/A

Parameter Values

<motion stop threshold>

Threshold count used in MSP430 to declare Mini-MT is in “stopped or transition” state.
See Notes.

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

3.20 \$MOTSTOP

Motion Stop Threshold (continued)

motion count (see \$MOTCNT), clips it if necessary (see \$MOTCLIP), and filters it (see \$MOTCOEF) to produce a filtered motion count.

The filtered motion count is computed as follows:

$$\begin{aligned} \text{Filtered motion count} = & \\ & ((\text{coef A} * \text{previous motion filter} \\ & \quad \text{count}) + \\ & (\text{coef B} * \text{clipped current motion} \\ & \quad \text{count})) / \\ & 65536 \end{aligned}$$

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 (see \$MOTSTAT and \$MOTTRANS). These state changes can be responsible for waking the modem if configured by the user (\$WAKEENBL).

3.21 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.

3.21 \$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 (see \$MOTCNT), clips it if necessary (see \$MOTCLIP), and filters it (see \$MOTCOEF) to produce a filtered motion count.

The filtered motion count is computed as follows:

$$\begin{aligned} \text{Filtered motion count} = & \\ & (\text{coef A} * \text{previous motion filter} \\ & \text{count}) + \\ & (\text{coef B} * \text{clipped current motion} \\ & \text{count}) / \\ & 65536 \end{aligned}$$

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 (see \$MOTSTAT and \$MOTTRANS). These state changes can be responsible for waking the modem if configured by the user (\$WAKEENBL).

3.22 Motion Wake Enable

\$WAKEENBL

Command Function

Motion Wake Enable

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 – reserved (do not use)
- 2 – reserved (do not use)
- 4 – declaration of “moving” state
- 8 – declaration of “stopped” state

3.22 \$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 (see \$MOTCNT), clips it if necessary (see \$MOTCLP), and filters it (see \$MOTCOEF) to produce a filtered motion count.

The filtered motion count is computed as follows:

$$\begin{aligned} \text{Filtered motion count} = & \\ & ((\text{coef A} * \text{previous motion filter} \\ & \text{count}) + \\ & (\text{coef B} * \text{clipped current motion} \\ & \text{count})) / \\ & 65536 \end{aligned}$$

3.22 \$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 (see \$MOTSTAT and \$MOTTRANS). These state changes can be responsible for waking the modem if configured by the user (\$WAKEENBL).

3.23 MSP430 Firmware Update

\$MSPFLASH

MSP430 Firmware Update

Command Function

This command allows the user to re-program the contents of the MSP430 processor with the image embedded in the modem. This allows the MSP430 to be field-upgradeable with a new modem release.

Command Functional Group

Enfora Specific

Command Format Query Response

AT\$MSPFLASH=?
\$MSPFLASH: ("password")
OK

Write Format

AT\$MSPFLASH=("password")

Response

setting up for BSL baud rate
Invoking the BSL entry sequence
sending RX password
sending RX data blocks
success
resetting MSP430
setting debug port back to 1200 baud
Srec download complete.

OK
AT-Command Interpreter ready

OK

Read Format Response

N/A
N/A

Execution Format Response

N/A
N/A

Parameter Values

Reference

N/A

3.23 \$MSPFLASH

Upload Code into MSP430 (continued)

Standard Scope

Optional

Enfora Implementation Scope

Full

Notes

When download is complete, modem resets the MSP430. When the MSP430 comes back up, it resets the modem.

The password is “flash” – all lower case.

3.24 MSP430 Firmware Version

\$MSPVER

MSP430 Firmware Version

Command Function

This command allows the user to query the MSP430 firmware version.

Command Functional Group

Enfora Specific

Command Format Query Response

N/A

Write Format Response

N/A

Read Format Response

AT\$MSPVER?
\$MSPVER: <ver>

Execution Format Response

N/A

Parameter Values

<ver>

MSP430 firmware version

Reference

N/A

Standard Scope

Optional

Enfora Implementation Scope

Full

3.25 MSP430 Raw Motion Count

\$MOTCNT

MSP430 Raw Motion Count

Command Function

This command allows the user to query the MSP430 interrupt count used in its motion sensor calculations. Periodic execution of this command can provide a crude histogram of the motion count which can then be used to tune the motion sensor configuration values.

Command Functional Group

Enfora Specific

Command Format Query Response

N/A

Write Format Response

N/A

Read Format Response

AT\$MOTCNT?
\$MOTCNT: <count>

Execution Format Response

N/A

Parameter Values

<count>

Number of motion interrupts during most recent one-second period that motion calculations were performed.

<filtered count>

Count value after filtering by the algorithm listed in the \$WAKEENBL command. It is the filtered count which is compared to the start/stop thresholds to determine whether the device should be declared moving or stopped.

Reference

N/A

Standard Scope

Optional

Enfora Implementation Scope

Full

3.25 \$MOTCNT

MSP430 Raw Motion Count (continued)

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 (see \$MOTCNT), clips it if necessary (see \$MOTCLP), and filters it (see \$MOTCOEF) to produce a filtered motion count.

The filtered motion count is computed as follows:

$$\begin{aligned} \text{Filtered motion count} = & \\ & ((\text{coef A} * \text{previous motion filter} \\ & \quad \text{count}) + \\ & (\text{coef B} * \text{clipped current motion} \\ & \quad \text{count})) / \\ & 65536 \end{aligned}$$

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 (see \$MOTSTAT and \$MOTTRANS). These state changes can be responsible for waking the modem if configured by the user (\$WAKEENBL).

3.26 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 Response

AT\$ODOMETER=?
\$ODOMETER: (0-4000000000)
OK

Write Format

AT\$ODOMETER=1234 (where 1234 is distance in meters)

Response

OK

Read Format Response

AT\$ODOMETER?
\$ODOMETER xxxx (xxxx=distance traveled in meters)
OK

Execution Format Response

N/A
N/A

Parameter Values

N/A

Reference

N/A

Standard Scope

Optional

Enfora Implementation Scope

Full

3.26 \$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.

3.27 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 Response

AT\$RINGIND = ?
\$RINGIND: (0-2)
OK

Write Format Response

AT\$RINGIND = <option>
OK

Read Format Response

AT\$RINGIND?
RINGIND: <option>

Execution Format Response

N/A
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

3.28 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>  
OK
```

Response

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

3.28 \$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

3.28 \$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.

4 Appendix A – Default AT Values

\$BATTlvl Default Value: Default Value Meaning:	Battery Level N/A N/A
\$DLycall Default Value: Default Value Meaning:	Delay Call 0 No user-defined delay between pressing the PTC button and placing a call.
\$DSPATCH Default Value: Default Value Meaning:	Dispatch Notification Phone Number 12345678123456789 This is the number the Mini-MT will call when the PTC button is pressed.
\$DSPNOTIF Default Value: Default Value Meaning:	Dispatch Notification 0 Dispatch notification is not active.
\$EMERNUM Default Value: Default Value Meaning:	Emergency Number 411 This is the emergency number the Mini-MT will call when the emergency call sequence is performed.
\$EVENT Default Value: Default Value Meaning:	User-defined Input/Output Set Geofence at current location Event group 1 configured for half-mile geofence when pressing geofence button.
\$GEOFNC Default Value:	Geo fencing a circle area 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

Default Value:

Default Value Meaning:

GPS Local Subscription

0

feature disabled

\$GPSRD

Default Value:

Default Value Meaning:

Read current GPS NMEA data

N/A

N/A

\$MMTTEST

Default Value:

Default Value Meaning:

Debug/Test Command

N/A

N/A

\$MOTCLIP

Default Value:

Default Value Meaning:

Motion Clip Threshold

65535

No clipping

\$MOTCOEF

Default Value:

Default Value Meaning:

Motion Filter Coefficients

500,50000

Old reading to new readings 1/10,000 ratio

\$MOTSTART

Default Value:

Default Value Meaning:

Motion Start Threshold

100

N/A

\$MOTSTAT

Default Value:

Default Value Meaning:

Motion Status

N/A

N/A

\$MOTSTOP

Default Value:

Default Value Meaning:

Motion Stop Threshold

50

N/A

\$MOTTRANS Default Value: Default Value Meaning:	Motion Transition Count 120 N/A
\$MSGLOGCL Default Value: Default Value Meaning:	Message Log Clear N/A N/A
\$MSGLOGEN Default Value: Default Value Meaning:	Message Log Enable 0 Message log is enabled
\$MSGLOGRD Default Value: Default Value Meaning:	Message Log Read Data N/A N/A
\$MSGSEND Default Value: Default Value Meaning:	Message Send N/A N/A
\$MSPCNT Default Value: Default Value Meaning:	MSP430 Raw Motion Count N/A N/A
\$MSPFLASH Default Value: Default Value Meaning:	MSP430 Firmware Update N/A N/A
\$MSPIO Default Value: Default Value Meaning:	MSP Input/Output Debug N/A N/A
\$MSPVER Default Value: Default Value Meaning:	MSP Version N/A N/A
\$ODOMETER Default Value: Default Value Meaning:	Trip Odometer N/A N/A
\$RINDIND Default Value: Default Value Meaning:	Ring Indicator 0 The Mini-MT is configured for audible ring for incoming calls

\$WAKEENBL

Default Value:

Default Value Meaning:

Motion Wake Enable

0

\$WAKEINTVL

Default Value:

Default Value Meaning:

Interval Wakeup Timer

0

Mini-MT will not be scheduled to wakeup at an interval.

\$WAKERTC

Default Value:

Default Value Meaning:

RTC Wakeup Timer

The Mini-MT is not configured to wake up based on future date/time.

\$WAKETIME

Default Value:

Default Value Meaning:

Modem Wake Duration

0

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