

**Application Note  
GSM0000AN015**

**Event Monitor and Reporting  
Overview**  
**Revision 1.01**

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

**Objective:** The intent of this document is to provide appropriate configuration settings for the Enabler @-G modem when the event monitor and/or reporting architecture is used. The Enabler platform is able to monitor events such as system powerup, network registration, receipt of IP address, and input/output line activity.

**Overview:** The basic architecture of the event processor is, when an input event occurs an output event is generated. The input events can be physical input lines, certain network activity, and/or a powerup event. The events can be based on occurrence, transition, or activity. Output events can generate activity on a physical output line and/or generate UDP-based messages that can be delivered over the air upon an associated event trigger. The message field(s) can be generated in binary or ASCII format and can be configured to append unique information or a range of pre-defined message options. Please refer to Appendix A of this overview. It contains the AT Command structure and syntax for the event and associated commands.

The following examples are provided as a general reference for the most common event and reporting scenarios. Each example will provide an objective and detail the configuration and testing of the solution.

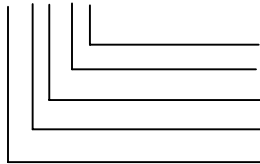
**Scenario #1**

Objective: Create an input event that will generate an output event.

AT\$EVENT parameters:

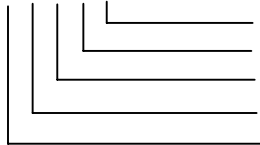
- 1a). Define, as a part of the first event group, a setting that monitors a state transition on pin 1 when it goes high (value of 1)
- 1b). Define, as a part of the first event group, a setting that transitions I/O (output) pin #2 state

AT\$EVENT=1,0,0,1,1



Ending range of 1 (high)  
 Starting range of 1 (high)  
 Activity on I/O line #1 based on range  
 Input transition event  
 Event group 1

AT\$EVENT=1,3,25,0,0



Ignored  
 Ignored  
 Transition I/O line signal on GPIO #2  
 Output event  
 Event group 1

Query the EVENT table:

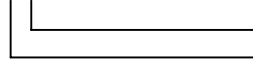
AT\$EVENT?

The table should reflect the following:

\$EVENT:	evgp	evtyp	evcat	p1	p2
1A	0	0	1	1	
1B	3	25	0	0	

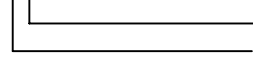
AT\$EVTEST (to test this example):

AT\$EVTEST=0,0



Create a low input signal  
Event category 0 (Input line 1)

AT\$EVTEST=0,1



Create a high input signal  
Event category 0 (Input line 1)

Results:

GPIO pin #2 (output) should transition state each time the AT\$EVTEST sequence above is issued.

## Scenario #2

Objective: To take the example provided in Scenario #1 and add a timer that will trigger the input event on a timed cycle.

Non-\$EVENT parameters:

1). Define event timer period.

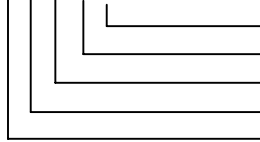
AT\$EVENT parameters:

1c). Define, as a part of the first event group, a setting that will use event timer #1 to generate the input event.

AT\$EVTIM1=60

Set input timer #1 to 60 seconds

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



Ending range of 1 (high)  
Starting range of 1 (high)  
Activate event timer 1  
Input event  
Event group 1

Query the EVENT table:

AT\$EVENT?

The table should reflect the following:

\$EVENT:	evgp	evtyp	evcat	p1	p2
1A	0	0	1	1	
1B	1	12	1	1	
1C	3	25	0	0	

Results:

GPIO pin #2 (output) should transition state every 60 seconds.

### Scenario #3

Objective: To take the example provided in Scenario #1, delete the event timer, and add new output event that will generate a message that will be sent to a server address.

Non-\$EVENT parameters:

- 1). Delete the timer event created in the last example.
- 2). Define IP address(es) to receive UDP output messages.

AT\$EVENT parameters:

- c). Define, as a part of the first event group, a setting that will create an output message containing the \$MDMID and the GPIO pin information.

AT\$EVDEL=1B

Delete the timer event in the last example

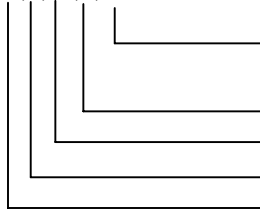
Query the EVENT table:

AT\$EVENT?

The table should reflect the following:

```
$EVENT: evgp evtyp evcat p1 p2
        1A  0    0    1  1
        1B  3   25    0  0
```

AT\$EVENT=1,3,41,1,12



Output message in ASCII format containing \$MDMID and the GPIO line activity indicator  
No user-defined message  
Output message w/Acknowledgement  
Output event  
Event group 1

Notes: The IP address(es) defined in the AT\$FRIEND will receive the output messages. Please refer to Appendix A and the AT Command documentation for command options and details.

Query the EVENT table:

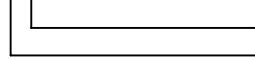
AT\$EVENT?

The table should reflect the following:

```
$EVENT: evgp evtyp evcat p1 p2
        1A  0    0    1  1
        1B  3   25    0  0
        1C  3   41    1  12
```

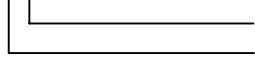
AT\$EVTEST (to test this example):

AT\$EVTEST=0,0



Create a low input signal  
Event category 0 (Input line 1)

AT\$EVTEST=0,1



Create a high input signal  
Event category 0 (Input line 1)

Results:

GPIO pin #2 (output) should transition state each time the \$EVENT sequence above is issued and the IP address(es) defined in the AT\$FRIEND command will receive UDP packets containing the \$MDMID and GPIO line activity indicator.

## Scenario #4

Objective: To delete all events created for event group 1 and define events that will toggle a GPIO line at a specified flash rate until a network event occurs then make the line go high based on another network event. In this example, we will flash an output line at ¼ second intervals until GSM network registration is achieved. These events will only support home network registration and will not indicate a loss of registration. The following scenario will provide roaming and registration loss support.

Non-\$EVENT parameters:

- 1). Delete all previously defined events.

AT\$EVENT parameters:

- 1a). Define, as a part of the first event group, a setting that will monitor the +CREG values that reflect the modem attempting to register (values 2 through 4)
- 1b). Define, as a part of the first event group, a setting that will toggle a GPIO line at a specified rate until a network event occurs.
- 2a). Define, as a part of the second event group, a setting that will monitor another +CREG value of 1 for home network registration.
- 2b). Define, as a part of the second event group, a setting that will toggle a GPIO line high based on successful registration.

AT\$EVDL=1 Delete all events in the last example

Query the EVENT table:

AT\$EVENT?

The table should reflect the following:

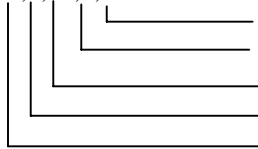
\$EVENT: evgp evtyp evcat p1 p2

AT\$EVENT=1,0,9,2,4



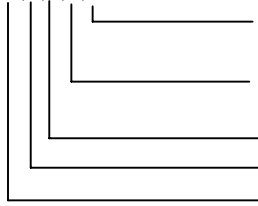
+CREG ending value of 4  
+CREG starting value of 2  
Monitor +CREG value  
Input transition event  
Event group 1

AT\$EVENT=1,3,33,1,0



Flash forever  
Flash at ¼ second intervals  
Flash GPIO line #2  
Output event  
Event group 1

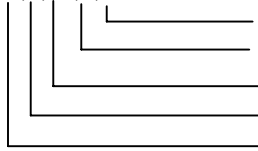
AT\$EVENT=2,0,9,1,1



+CREG ending value of 1 – Registered on home network  
 +CREG starting value of 1 – Registered on home network

Monitor +CREG value  
 Input transition event  
 Event group 2

AT\$EVENT=2,3,17,0,0



Ignored  
 Ignored  
 Set GPIO line #2 high  
 Output event  
 Event group 2

Query the EVENT table:

AT\$EVENT?

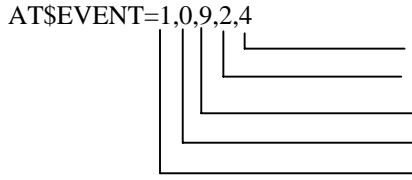
The table should reflect the following:

\$EVENT:	evgp	evtyp	evcat	p1	p2
1A	0	9	2	4	
1B	3	33	1	0	
2A	0	9	1	1	
2B	3	17	0	0	

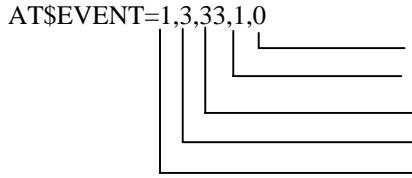
Results:

GPIO pin #2 (output) should flash at ¼ second intervals until GSM registration on home network.

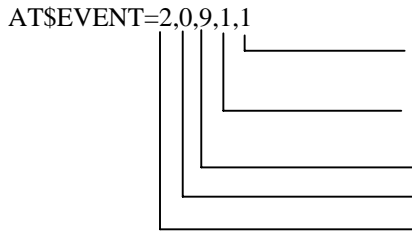




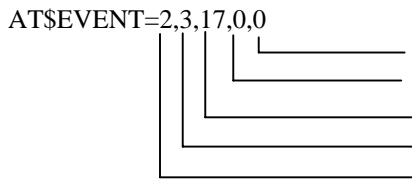
+CREG ending value of 4  
 +CREG starting value of 2  
 Monitor +CREG value  
 Input transition event  
 Event group 1



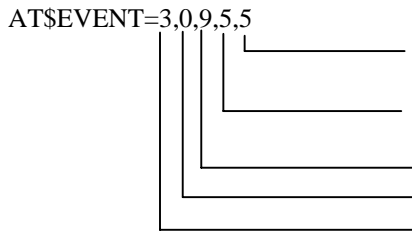
Flash forever  
 Flash at 1/4 second intervals  
 Flash GPIO line #2  
 Output event  
 Event group 1



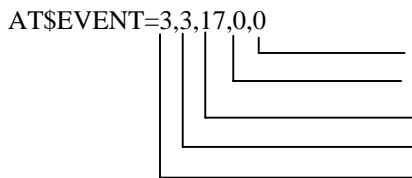
+CREG ending value of 1 – Registered on home network  
 +CREG starting value of 1 – Registered on home network  
 Monitor +CREG value  
 Input transition event  
 Event group 2



Ignored  
 Ignored  
 Set GPIO line #2 high  
 Output event  
 Event group 2



+CREG ending value of 1 – Registered on roaming network  
 +CREG starting value of 1 – Registered on roaming network  
 Monitor +CREG value  
 Input transition event  
 Event group 3



Ignored  
 Ignored  
 Set GPIO line #2 high  
 Output event  
 Event group 3

Query the EVENT table:

AT\$EVENT?

The table should reflect the following:

\$EVENT:	evgp	evtyp	evcat	p1	p2
1A	0	9	2	4	
1B	3	33	1	0	
2A	0	9	1	1	
2B	3	17	0	0	
3A	0	9	5	5	
3B	3	17	0	0	

Results:

GPIO pin #2 (output) should flash at ¼ second intervals until GSM registration on home or roaming networks. Once registered, GPIO pin #2 will go high. If registration status is lost, the I/O pin will flash.

## Appendix A

<b>\$EVENT</b>	User Defined Input/Output
<b>Command Function</b>	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.
<b>Command Functional Group</b>	Enfora Specific
<b>Command Format Query Response</b>	AT\$EVENT=? \$EVENT: (0-99),(0-3),(0-255), (-2147483647 - 2147483647)
<b>Write Format Response</b>	AT\$EVENT=<event group>,<event type>,<event category>,<parm1>,<parm2> OK
<b>Read Format Response</b>	AT\$EVENT? \$EVENT: evgp evtyp evcat p1 p2 1A 0 9 2 4 1B 3 33 1 0 2A 0 9 5 5 2B 3 17 0 0 3A 0 9 0 0 3B 3 9 0 0 4A 0 9 1 1 4B 3 17 0 0
<b>Execution Format Response</b>	N/A

## \$EVENT

## User Defined Input/Output (continued)

### Parameter Values

<event group>

The event group. This parameter defines all of the events and the order they are executed. Events are grouped together to control execution sequence. If all input event group entries or an entry in the group are within range and at least 1 trigger input is valid, all event group outputs are activated. An event group should have at least 1 trigger input event, and at least 1 output event.

<event type>

Event type

Type of event	Description	Value
transition trigger	Trigger when the requested event has transitioned into the event range. Valid when within the event range.	0
occurrence trigger	Trigger anytime the event occurs and is in the valid event range. Valid when within the event range.	1
input	Valid when within the event range.	2
output	Event is created when all inputs are valid & a trigger is present.	3

<event category>

Event categories

**Input State Event** - Event based on GPIO pin activity when defined as an input. 0 = LOW 1 = HIGH

Value	Description	Valid range
0	GPIO1 – General Purpose Input/Output #1	0,1
1	GPIO2 – General Purpose Input/Output #2	0,1
2	GPIO3 – General Purpose Input/Output #3	0,1
3	GPIO4 – General Purpose Input/Output #4	0,1
4	GPIO5 – General Purpose Input/Output #5	0,1
5	GPIO6 – General Purpose Input/Output #6	0,1
6	GPIO7 – General Purpose Input/Output #7	0,1
7	GPIO8 – General Purpose Input/Output #8	0,1

### Miscellaneous Input State Event

52	0 or 1	0 or 1	New SMS message Event. This event is set to a 1 on the arrival of a new SMS message and clears when any SMS read command is issued
----	--------	--------	--

**\$EVENT****User Defined Input/Output (continued)****Modem Status Events** - Events based on modem status.

<b>Value</b>	<b>Description</b>	<b>Valid range</b>
8	PWRUP – Modem power up	1
9	PLUS_CREG – Modem GSM registration. See AT+CREG command for range definitions.	0 to 5
10	PLUS_CGREG – Modem GPRS registration. See AT+CGREG command for range definitions.	0 to 5
11	NETWORK_IP – Receipt of IP address from network provider.	0,1  0 = No IP 1 = Available IP received from network

**Input Timer Events** - Event timer used to provide input signal based on timer value. See AT\$EVTIME command to set pulse rate.

<b>Value</b>	<b>Description</b>	<b>Valid range</b>
12	EVTIM1 – Input event timer #1	1
13	EVTIM2 – Input event timer #2	1
14	EVTIM3 – Input event timer #3	1
15	EVTIM4 – Input event timer #4	1

**SEVENT****User Defined Input/Output (continued)**

**Output State Event** - Events based on GPIO pin activity when defined as an output.

<b>Value</b>	<b>Description</b>
0	GPIO1 input – Changes GPIO line #1 to an input
1	GPIO2 input – Changes GPIO line #2 to an input
2	GPIO3 input – Changes GPIO line #3 to an input
3	GPIO4 input – Changes GPIO line #4 to an input
4	GPIO5 input – Changes GPIO line #5 to an input
5	GPIO6 input – Changes GPIO line #6 to an input
6	GPIO7 input – Changes GPIO line #7 to an input
7	GPIO8 input – Changes GPIO line #8 to an input
8	GPIO1 low – Generate a low signal on GPIO line #1
9	GPIO2 low – Generate a low signal on GPIO line #2
10	GPIO3 low – Generate a low signal on GPIO line #3
11	GPIO4 low – Generate a low signal on GPIO line #4
12	GPIO5 low – Generate a low signal on GPIO line #5
13	GPIO6 low – Generate a low signal on GPIO line #6
14	GPIO7 low – Generate a low signal on GPIO line #7
15	GPIO8 low – Generate a low signal on GPIO line #8
16	GPIO1 high – Generate a high signal on GPIO line #1
17	GPIO2 high – Generate a high signal on GPIO line #2
18	GPIO3 high – Generate a high signal on GPIO line #3
19	GPIO4 high – Generate a high signal on GPIO line #4
20	GPIO5 high – Generate a high signal on GPIO line #5
21	GPIO6 high – Generate a high signal on GPIO line #6
22	GPIO7 high – Generate a high signal on GPIO line #7
23	GPIO8 high – Generate a high signal on GPIO line #8

**Output Transition Events** - Events based on GPIO pin activity to transition an output line to the opposite state.

24	GPIO1 toggle – Transition line signal on GPIO line #1
25	GPIO2 toggle – Transition line signal on GPIO line #2
26	GPIO3 toggle – Transition line signal on GPIO line #3
27	GPIO4 toggle – Transition line signal on GPIO line #4
28	GPIO5 toggle – Transition line signal on GPIO line #5
29	GPIO6 toggle – Transition line signal on GPIO line #6
30	GPIO7 toggle – Transition line signal on GPIO line #7
31	GPIO8 toggle – Transition line signal on GPIO line #8

**\$EVENT****User Defined Input/Output (continued)****Output Flash Events** - Events based on GPIO pin activity to flash an output line.

32	GPIO1 flash – Flash line signal on GPIO line #1 based on parm1 and parm2 values
33	GPIO2 toggle – Flash line signal on GPIO line #2 based on parm1 and parm2 values
34	GPIO3 toggle – Flash line signal on GPIO line #3 based on parm1 and parm2 values
35	GPIO4 toggle – Flash line signal on GPIO line #4 based on parm1 and parm2 values
36	GPIO5 toggle – Flash line signal on GPIO line #5 based on parm1 and parm2 values
37	GPIO6 toggle – Flash line signal on GPIO line #6 based on parm1 and parm2 values
38	GPIO7 toggle – Flash line signal on GPIO line #7 based on parm1 and parm2 values
39	GPIO8 toggle – Flash line signal on GPIO line #8 based on parm1 and parm2 values

**Output Message Events** - Events that generate a UDP message based on GPIO pin activity.

40	Udp Message – Generate and transmit one UDP message based on parm1 and parm2 values.
41	Udp Message w/ Acknowledge - Generate and transmit a UDP message based on parm1 and parm2 values. This message is controlled by the \$ACKTM and \$WAKEUP parameters. This message can also be acknowledged thus canceling the \$ACKTM and \$EVTIM parameters.

**\$EVENT****User Defined Input/Output (continued)**

&lt;parm1&gt;

parameter values

<b>Event Type</b>	<b>Event Category</b>	<b>Result</b>
Input, transition, occurrence	Input State Event	Defines the valid start range
Input, transition, occurrence	Modem Status Event	Defines the valid start range
Input, transition, occurrence	Input Timer Event	Defines the valid start range
Output, transition, occurrence	Output State Event	Parm1 is ignored
Output	Output Transition Event	Parm1 is ignored
Output	Output Flash Event	parm1 is flash toggle rate in 1/4 second intervals. 0 = no flash.
Output	Output Message Event	parm1 is a special user message identification. The 32-bit number entered can be used to dynamically generate intelligent messages. It will be the first 4 bytes of data following the API header in the output message. An example of using this field would be to designate the bytes as unique identifiers. Bytes 1 and 2 could identify the device, byte 3 could define the message format, and byte 4 could provide the event group that triggered the message.

**\$EVENT****User Defined Input/Output (continued)**

&lt;parm2&gt;

parameter values

<b>Event Type</b>	<b>Event Category</b>	<b>Result</b>
Input, transition, occurrence	Input State Event	Defines the valid end range
Input, transition, occurrence	Modem Status Event	Defines the valid end range
Input, transition, occurrence	Input Timer Event	Defines the valid end range
Output	Output State Event	Parm2 is ignored
Output	Output Transition Event	Parm2 is ignored
Output	Output Flash Event	Parm2 is the number of flash toggles before returns to pre-flash state. 0 = toggle forever.
Output	Output Message Event	Parm2 defines the output UDP message format. <b>Bit 1:</b> 1 = send all numeric data in binary format 0 = ASCII format <b>Bit 2:</b> add PARM1 to generated UDP message (4 bytes binary, 11 bytes ASCII) <b>Bit 3:</b> add \$MDMID value to generated UDP message (22 bytes ASCII) <b>Bit 4:</b> add GPIO data and direction to generated UDP message (2 bytes binary, 6 bytes ASCII) <b>Bit 5-32</b> TBD

**Reference****Standard Scope**                      Optional**Enfora Implementation Scope**      Full

## Revision History

Date	Rev	Author	Description
2/28/03	1.00	Matt Glover	Initial Release.
3/30/06	1.01	Diane O'Neil	Added Input State 52