

Функция «Обработка событий» в GSM/GPRS-модулях фирмы Enfora

В модулях Enabler предусмотрены широкие возможности по обработке событий. Модуль может быть запрограммирован на отслеживание внешних событий и выполнение определенного рода ответных действий.

Виктор Алексеев, к. ф.-м. н.
info@telemetry.spb.ru

Введение

В качестве входных могут быть использованы следующие группы событий:

- состояние вводов/выводов (8 вводов/выводов);
- питание (включено или выключено);
- регистрация GSM/GPRS (процесс регистрации, работа в домашней сети, роуминг, нет регистрации);
- сетевые IP-события (нет IP, первый IP, новый IP, нет IP при переключении, действующий IP при переключении);
- входящее SMS-сообщение;
- часы реального времени;
- таймер событий;
- счетчик событий;
- GPS расстояние;
- GPS максимальная скорость;
- GPS регион (5 заданных точек с фиксированными координатами центра и радиусом);
- GPS спутник (действующий, недействующий);
- АЦП (2 аналоговых входа).

В качестве реакции модуля (выходных событий) рассматривается тот же набор, что и для входных событий — сообщения UDP и SMS, состояние I/O, сетевые события и др.

Программирование входных и выходных событий с помощью AT\$EVENT

Новые пользовательские AT-команды, предназначенные для обработки событий, задаются специальной командой AT\$EVENT [1].

Алгоритм команды охватывает практически все стандартные ситуации при работе с приложениями M2M.

Структура команды выглядит следующим образом:

```
AT$EVENT=<event group>,  
<event Response type>,<event category>,  
<parm1>,<parm2>
```

Первый параметр — **Event group** — определяет группу событий. Все события могут быть дополнительно разбиты по отдельным группам, которые будут обрабатываться поочередно. Данный параметр определяет

очередность обработки заданных событий. Таким образом, можно задать циклическую обработку нескольких событий. В случае, если происходят все заданные в группе события или, по крайней мере, явным оказывается одно из заданных входных событий, то активируются все заданные в группе выходные события. В каждой группе должно быть задано хотя бы одно входное и одно выходное событие. Номер группы может принимать значения от 0 до 99. Поскольку первые четыре группы обычно в модемах отведены под заводские настройки, то на практике рекомендуется использовать значения от 5 до 99. Кроме единичных событий можно задать также и множественные, последовательные события — такие как «Состояние вводов/выводов» и «Сетевые IP-события» и т. д. Множественные события внутри одной группы должны рассматриваться как логическое условие «и».

Event Response type — этот параметр определяет тип события: «Входное» (Input) или «Выходное» (Output). Этот параметр может принимать значения от 0 до 3.

Третий параметр **Event category** описывает конкретные входные и выходные события. Параметр может принимать значения от 0 до 255. Часть из этих значений задействована, а часть зарезервирована.

Parm1 и **parm2** — эти параметры задают диапазон входных событий и тип выходных событий. Их нельзя рассматривать в отрыве от категории событий. Например, для входных событий GPIO значения parm1, 2, равные 0 или 1, соответствуют высокому или низкому уровням. Для АЦП значения parm1, 2 будут означать уровни входного сигнала. Для выходных событий значения parm1, 2 определяют такие параметры, как частота мигания светодиодов, содержание сообщений UDP и SMS, смену времени таймера, сохраненные AT-команды. Parm1 и parm2 могут принимать значения от 0 до 2147483647.

«Входные события» разделены на три основных класса: **Transition trigger**, **Occurrence trigger**, **Input trigger**. Такая градация позволяет определить «истинность» и «ложность» входного события и исключить возможность случайного появления выходного события.

«Выходное событие» — **Output** — возникает при «истинном» входном событии.

Transition trigger — это тип входных событий, соответствующий значению «0» в параметре `<event Response type>`. Данное входное событие инициируется при переходе из одного состояния в другое. Иными словами, это событие считается «истинным» и запускает выходное событие только в том случае, если оно в процессе изменений попадает в заранее заданную область. Рис. 1 иллюстрирует такое событие.

Возможны три варианта, когда **Transition trigger** становится «истинным». В одном случае параметры в процессе изменений попадают внутрь заданной области (рис. 1).

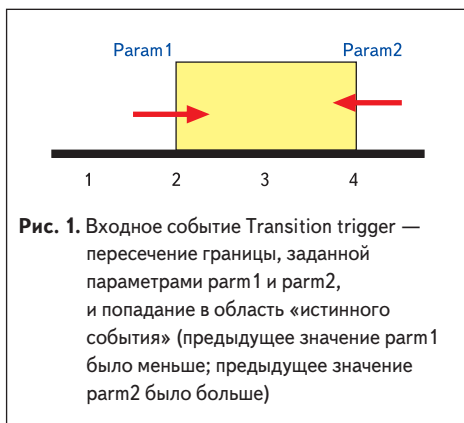


Рис. 1. Входное событие **Transition trigger** — пересечение границы, заданной параметрами `parm1` и `parm2`, и попадание в область «истинного события» (предыдущее значение `parm1` было меньше; предыдущее значение `parm2` было больше)

Во втором случае оба параметра последовательно становятся равными нулю (рис. 2).

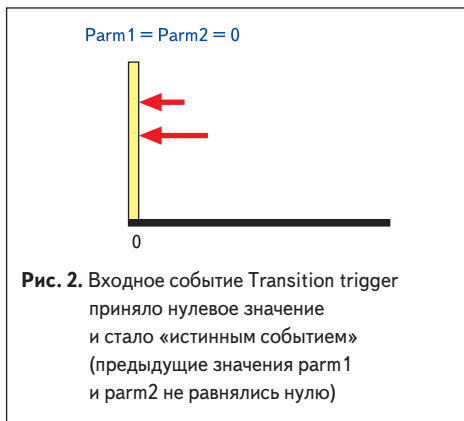


Рис. 2. Входное событие **Transition trigger** приняло нулевое значение и стало «истинным событием» (предыдущие значения `parm1` и `parm2` не равнялись нулю)

В третьем случае оба параметра последовательно становятся равными единице (рис. 3).

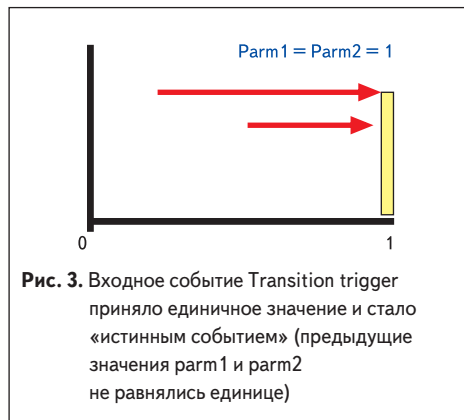


Рис. 3. Входное событие **Transition trigger** приняло единичное значение и стало «истинным событием» (предыдущие значения `parm1` и `parm2` не равнялись единице)

Для примера можно рассмотреть случай потери регистрации в сети GSM. Состояние связи можно запросить командой:

```
AT+CREG?
```

Ответ на эту команду может быть 0, 1, 2, 3, 4, 5:

- 0 — нет регистрации, модем не ищет оператора;
- 2 — нет регистрации, но модем ищет оператора;
- 1 — зарегистрирован в домашней сети;
- 5 — зарегистрирован в роуминге;
- 3 — регистрация прервана;
- 4 — неизвестное состояние.

Предположим, что ответ на команду `AT+CREG` постоянно меняется в диапазоне от 0 до 5. Потеря регистрации в сети GSM будет трактоваться как «истинное» событие, когда ответ попадет в диапазон значений от 2 до 4 и останется в этой области (рис. 1). При этом будет запущено выходное событие.

Occurrence trigger — тип входного события, соответствующий значению «1» в `event Response type`. Данное событие уже находится внутри заданной области к моменту рассмотрения. В случае с примером потери регистрации в сети событие будет «истинным», если к моменту использования его значение находится в диапазоне от 2 до 4 и не выходит за эти границы.

Также, по определению, событие «**Occurrence trigger**» запустит выходное событие, когда его значения находятся в интервале от 0 до 1.

Input trigger — входное событие, у которого значение `event Response type` равно «2». Это событие рассматривается как логическое условие «И» по отношению к другим входным событиям.

Output — выходное событие, параметр `event Response type` которого задается равным «3». В данном случае, в результате любого «истинного» входного события запускается устойчивое выходное событие, которое может быть изменено только при перезапуске модуля или под воздействием внешнего управляющего сигнала. Например, в результате некоторого «истинного» входного события состояние GPIO1 изменилось с низкого на высокое. Если через какое-то время это входное событие станет «ложным», состояние GPIO1 все равно останется высоким.

Примеры **Event category**

Если в `event category` задано число 1, то это соответствует описанию состояния ввода-вывода GPIO2. Параметры `parm1` и `parm2` могут принимать здесь значения только 0 или 1, означающие низкий или высокий логические уровни.

Категория № 9 описывает рассмотренный выше случай потери регистрации в сети GSM.

Категория № 18 соответствует состоянию АЦП. Параметры `parm1` и `parm2` находятся в диапазоне от 0 до 1760.

Категория № 44 заставляет модем выполнить в качестве выходного события любую заранее

заданную AT-команду, которая прописывается в контексте команды:

```
AT$STOATEV = <1-25>, <AT command >
```

Если, например, записать эту команду в виде:

```
AT$STOATEV=1,AT+CMSS=3,
```

то это означает, что в качестве выходного события выбрана команда отсылки SMS, которое записано в ячейке № 3. Параметр 1 соответствует порядковому номеру команды и задается с помощью `parm1`. Всего в новой версии ПО можно сохранять до 25 команд.

Подробно все реализованные в ПО Enfora варианты событий рассмотрены в последней версии GSM0308PB001MAN.

Следует обратить внимание на то, что в модулях Enfora предусмотрено 255 наименований различных категорий событий, которые в совокупности с комбинациями из пяти параметров в команде `AT$EVENT` предоставляют пользователю возможность легко написать приложения для большинства приложений M2M.

Пример работы с **Event Processing**

В качестве примера работы с **Event Processing** рассмотрим случай, когда нужно, чтобы модем GSM2238 посылал UDP-сообщение на центральный сервер через определенный интервал времени [2].

1. Выбор группы входного события `<event group>`. Проверяем, какие группы заняты под заводские настройки. Команда:

```
AT$EVENT?
```

Показывает, что под заводские настройки занято 11 групп. Поэтому выбираем группу № 12.

2. Выбор типа входного события `<event Response type>`. Нам нужно, чтобы по окончании заданного времени работы таймера запустилось выходное событие. Наилучшим образом это описывается с помощью входного события типа **Occurrence trigger**. Следовательно, выбираем значение `<event Response type>`, равное 1.

3. Выбор категории входного события `<event category>`. Из постановки задачи понятно, что нужно искать событие, связанное с таймером. В таблице категорий **Input State Event** (AT Command Document, GSM0308PB001MAN) находим, что это категории с 12 по 15 и с 66 по 69. Теперь нужно определить, какие таймеры заняты под заводские настройки. С помощью команды:

```
AT$EVTIM?
```

определяем, что занят только таймер № 1, а остальные свободны. Поэтому мы возьмем следующий таймер № 2. В таблице категорий в столбце `Description` выбираем № 13 — входное событие, связанное с таймером № 2.

4. Выбор параметров входного события <parm1>, <parm2>. Из таблицы видно, что для категории № 13 оба параметра parm1 и parm2 равны 1.

5. Описание входного события. С учетом п. п. 1–4 описываем входное событие:

```
AT$EVENT=12,1,13,1,1
```

6. Параметры таймера. Необходимый интервал времени задается командой:

```
AT$EVTIM2=Z,
```

где Z — время в секундах от 0 до 604800. Выбираем, например, 600 с.

7. Выбор группы выходного события <event group>. Поскольку у нас простой случай и нет последовательности событий, то для выходного события выбираем ту же самую группу № 12, что и для входного события.

8. Выбор типа входного события <event Response type>. Выходное событие однозначно определяется значением <event Response type>, равным 3.

9. Выбор категории входного события <event category>. Нам нужно, чтобы по окончании работы таймера было послано UDP-сообщение. По таблице категорий **Output Message Events** (AT Command Document, GSM0308PB001MAN) находим, что лучше всего подходит категория № 40. Эта категория описывается следующим образом: «Создание и отправка UDP-сообщения с помощью выбора значений параметров parm1 и parm2.

В данном случае UDP-сообщение посылается по IP-адресам, заданным командой \$FRIEND и на номера портов, перечисленных в \$UDPAPI.

10. Выбор параметров выходного события <parm1>, <parm2>. Из таблицы категорий «Bit-Field Table» (AT Command Document, GSM0308PB001MAN) определяем значение parm1 равным 8, а parm2 равным 4350. В данном случае parm2 определяет OTA Message (ASCII RMC NMEA msg).

Значение parm1 здесь может быть любым, например 11111 или 12345. Оно задано исключительно для идентификации.

Значение parm 2 (4350) — это десятичная форма записи двоичного кода 100001111110.

Этот код расшифровывается справа налево следующим образом:

- 0 = отослать все полученные данные в виде таблицы в формате ASCII;
- 1 = добавить значение parm1 в UDP-сообщение (у нас в примере это 8);
- 1 = добавить значение \$MDMID;
- 1 = не важно для MiniMT;
- 1 = не важно для MiniMT;
- 1 = не важно для MiniMT;
- 1 = не важно для MiniMT;
- 1 = добавить категорию входного события;
- 0 = не добавлять GPS-данные (3 байта информационных данных в двоичном коде или до 80 байтов в \$GPGGA NMEA-сообщении);

- 0 = не добавлять GPS-данные (1 байт информации о статусе в двоичном коде или до 80 байтов в \$GPGGA NMEA-сообщении);

- 0 = не добавлять GPS-данные (3 байта информации о широте в двоичном коде или до 80 байтов в \$GPGGA NMEA-сообщении);

- 0 = не добавлять GPS-данные (3 байта информации о долготе в двоичном коде или до 80 байтов в \$GPGGA NMEA-сообщении);

- 0 = не добавлять GPS-данные (3 байта информации о скорости в двоичном коде или до 80 байтов в \$GPGGA NMEA-сообщении).

Подробно эта процедура описана в “Mini-MT AT command, с. 332” [3].

11. Описание выходного события. На основании п. п. 7—10 описываем выходное событие в виде:

```
AT$EVENT=12,3,40,8,4350
```

12. Запись команд в память модема. С помощью программы HyperTerminal записываем команды, полученные выше в пунктах 5, 6 и 11 в память модема. Сохраняем эти команды с помощью AT&W. В результате модем будет посылать на сервер каждые 10 минут UDP-сообщение с координатами в формате GPS RMC NMEA.

Приведенный пример показывает, что при наличии небольшого опыта обычный пользователь, не знакомый с программированием, может писать свои собственные приложения для модулей Enfora.

Прикладная программа Enfora Event Tools

Для решения простых задач по программированию пользовательских IO можно использовать утилиту Enfora Event Tools.

Основные возможности программы Event Tools:

- программирование обработки различных событий на пользовательских вводах/выходах модемов Enfora;

- запись заданных событий в память модуля;
- чтение заданных событий из памяти;
- запись заданных событий в файл;
- чтение заданных событий из файла;
- редактирование списка событий.

Работа с программой осуществляется в диалоговом режиме. Окно «Входящие события» показано на рис. 4.

В этом окне можно выбрать необходимые события, однако следует учитывать, что программа Event Tools имеет ограниченный набор входных событий по сравнению с \$EVENT. В частности, не поддерживаются группы входящих SMS-сообщений, группа часов реального времени и группа счетчика событий. Рассмотрим случай использования модемов GSM1208 и GSM1308 для контроля доступа в помещение по GSM/GPRS-каналу.

Модем GSM1208 имеет программируемые пользовательские входы/выходы: GPIO1, GPIO3, GPIO5, GPIO6, а также ADC1, ADC2, DAC, GND.

Пользовательские входы/выходы могут быть запрограммированы на срабатывание герконовых датчиков, датчиков движения, аудио-датчиков, датчиков контроля температуры и т. д. При поступлении сигнала тревоги от внешних датчиков или отклонения измеряемых параметров от заданных устройство GSM1208 посылает пользователю SMS или UDP-сообщение через сеть GSM/GPRS. При необходимости удаленного включения или выключения дополнительного оборудования на объекте (сирена, прожектор и т. д.) пользователь со своего сотового телефона посылает управляющее SMS-сообщение, получив которое, модем дает команду управляющему блоку на включение или выключение приборов.

Пусть к контактам № 1 (GPIO1) и № 8 (GND) подключен обычный герконовый датчик, у которого есть два положения «замкнуто» и «разомкнуто».

Идея заключается в том, чтобы модем посылал SMS с тревожным сообщением на заданные номера мобильных телефонов при размыкании контактов датчика.

Если в качестве входящего события выбрано состояние GPIO1, то в окне, показанном

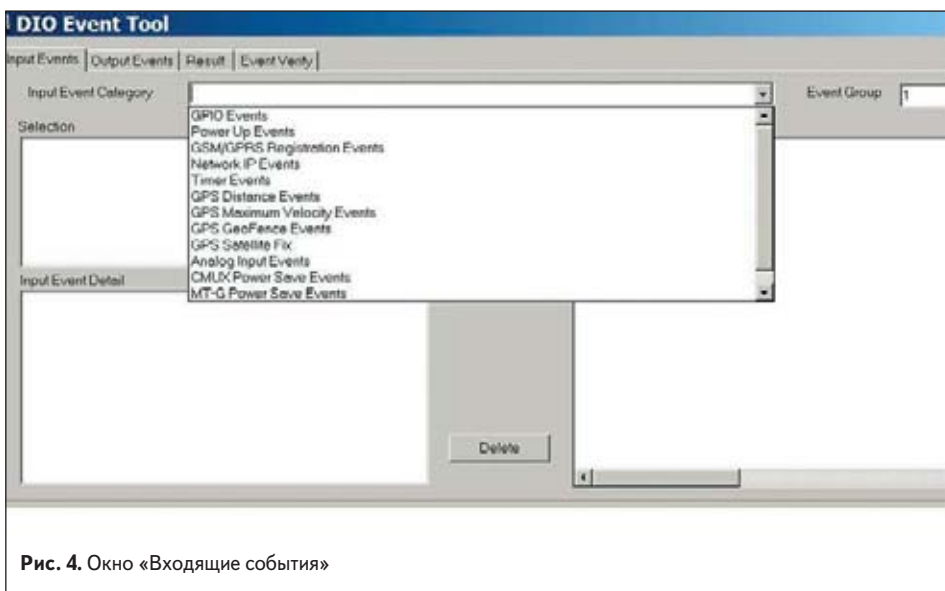


Рис. 4. Окно «Входящие события»

на рис. 3, будут предложены варианты состояний:

- инверсия — высокое/низкое значение;
- инверсия — низкое/высокое значение;
- переходное состояние;
- низкое значение без изменения;
- высокое значение без изменения.

Здесь можно выбрать следующие значения:

- Состояние вводов/выводов (8 вводов-выводов с высоким или низким значением, инверсия состояния, без изменения).
- UDP API-сообщение (номер UDP API-сообщения, ID-модуля, номер GPIO, аналоговые сигналы 1 и 2, сохранить сообщение; триггерное событие, GPS-дата и время, координаты, количество спутников).
- UDP-подтверждение (номер сообщения, ID-модуля, номер GPIO, аналоговые сигналы 1 и 2, сохранить сообщение; триггерное событие, GPS-дата и время, координаты, количество спутников).
- UDP broadcast сообщение (номер сообщения, ID-модуля, номер GPIO, аналоговые сигналы 1 и 2, сохранить сообщение; триггерное событие, GPS дата и время, координаты, количество спутников).
- Сброс таймера событий (4 таймера).
- Выполняемая AT-команда (пять различных AT-команд, которые выполняются при конкретном событии).

Если в качестве выходного события выбрано Stored AT-commands, то в открывшемся окне будет предложено выбрать:

- Stored AT-command 1;
- Stored AT-command 2;
- Stored AT-command 3;
- Stored AT-command 5;
- Stored AT-command 6.

Эта опция заставляет модем выполнить любую заданную AT-команду. Всего в программе Event Tools можно выбирать до пятнадцати AT-команд, соответствующих конкретному входящему событию.

В нашем примере нужно выбрать команду AT+CMSS=1, при выполнении которой на заданный номер отправляется SMS-сообщение, хранящееся в ячейке № 1.

После ввода описаний событий, при нажатии на Result, в окне Output Event Summary появляется набор AT-команд, соответствующих обрабатываемым событиям. Эти команды затем записываются в память модуля и в контрольный файл.

Для записи в окне Modem COM Port нужно ввести номер COM-порта (в нашем случае это COM 1), затем поставить галочку Save Events in Modem's Non-Volatile Storage и нажать кнопку Write to Modem.

Записанные команды можно также прочитать из памяти и при необходимости отредактировать. Теперь можно задать параметры тревожного SMS-сообщения и ввести любой текст.

Другим примером наиболее часто используемого приложения Event Processing может служить функционал, обеспечивающий автоматическое подключение к сети GSM и GPRS, постоянный автоматический контроль состояния подключения и автоматическое восстановление подключения к GSM и GPRS в случае потери соединения. В случае, когда модем обнаруживает отсутствие связи в сети GSM/GPRS, он через заданный интервал времени начнет автоматическую перезагрузку. При этом происходит полная перезагрузка программного обеспечения терминала и соответственно автоматическое, корректное завершение работы всех текущих процессов. После этого терминал самостоятельно возобновляет работу.

Модули Enfora имеют мощный IP-ориентированный программный блок и позволяют обойтись базовыми настройками для реализации многих стандартных задач.

При работе с динамическим адресом возникает ряд сложностей, когда требуется передать информацию по IP-сети GPRS-устройству, адрес которого заранее неизвестен. При этом адрес устройства, видимый во внешней IP-сети, может меняться уже после подключения к сети GPRS. С помощью Event Processing модули Enfora могут быть запрограммированы так, чтобы автоматически посылать на один

или несколько серверов сообщения, которые очевидным образом содержат актуальный IP-адрес устройства. Внутри такого сообщения содержится специальная информация, позволяющая идентифицировать конкретного отправителя.

Например, можно посылать уведомление о получении модемом нового IP-адреса по заранее заданному списку серверов. При этом одновременно определяется список IP-адресов, с которых можно осуществлять удаленное управление модулем.

Используя в качестве входного события UDP-сообщения, можно управлять модулем по PPP-соединению. Такое управление возможно даже в том случае, когда модем не зарегистрирован в сети GPRS и ему не назначен внешний динамический IP-адрес. Сопряжение с устройствами, не поддерживающими IP-протокол, реализуется с помощью команды AT\$HOSTIF=<type>.

Заключение

Enfora постоянно совершенствует фирменное программное обеспечение. В последней версии для SW078 добавлены перечисленные ниже новые Event команды:

- AT\$EVTIMQRY — запрос статуса таймера событий;
- новое событие, описывающее отправку данных ASCII на последовательный порт;
- AT\$STOATEV — количество команд увеличено с 15 до 25;
- Added AT command to query — запрос на количество использованных групп событий;
- AT\$EVTIM5, 6, 7, 8 — новые таймеры. ■

Литература

1. AT Command Document, GSM0308PB001MAN, Enfora Enabler III GSM/GPRS/EDGE Radio Modem AT Command Set Reference Ver. 1.01.
2. Event Processing Commands 2008.
3. Mini-MT AT Command Set Reference. Event Processing Commands.
4. GSM0000PB003MAN, Enfora Event Tools, Rev. 1.00, User Manual.