Application Note
Using AT commands to control TCP/IP stack on SEM GSM modules

Sony Ericsson

Fourth edition (August 2004)

Sony Ericsson

Contents

1 Introduction

In order to supply a TCP or UDP interface to users via the system AT interface a new set of commands has been defined for use in this context. These commands are a simplified set of interface functions that will allow the TCP\UDP transport mechanisms to be used with the minimum of pre-configuration and error handling.

Certain restrictions are imposed on their use but are intended to make the initial implementation a simple and intuitive experience, as well as providing a reliable but flexible method of creating a robust mechanism for handling of packet data.

When using GPRS with the SEM modules some units have an embedded TCP/IP stack others do not, theses are listed below:-

Units without the TCP/IP stack	- GM29
	- GM41
	- GM47/GM48
Units with the TCP/IP stack	- GM47r5/GM48r5
	- GM29r5
	- GR47/GR48
	- GT47/GT48

2 Overview Of The TCP\UDP AT Commands

The TCP/IP features provided by the AT commands for the GR47 are intended to provide a subset of the features normally available at the socket level when using a conventional TCP/IP stack, with some simplifications and customisations based on the specific features of the GR47.

The new commands allow the application writer to create and destroy UDP and TCP sockets, to control underlying GPRS PDP contexts, to transfer data to and from the module, and to interrogate IP status information about the active link.

The socket interface is provided by a series of AT commands outlined below.

AT Command	Functionality
AT*E2IPA	Activate IP session using stored PDP context
AT*E2IPO	Initiate a new IP connection to the module(UDP/TCP)
AT*E2IPC	Close a currently open IP Connection
AT*E2IPI	Reports the current IP status of the module
AT*E2IPRH	Returns a string denoting an IP host URL of an active session.
AT*E2IPE	Returns an error code from the last active IP session to assist in debug of error causes.
AT*E2IPS	Modify the way in which data received by the module is handled with respect to transmitting it across TCP/UDP
AT*E2IPL	Allows the unit to listen for IP traffic on a specific port number i.e. port 80 (HTML) and then start a socket session with anyone that requests a connection.

For more detailed information on the TCP/IP AT commands and their parameters, please refer to the AT commands manual for the GR47/GR48 product or your nearest M2M distributor.

2.1 Restrictions

Below known restrictions of the stack are listed.

- TCP/UDP connections can only be made over GPRS using PDP context information assumed to be previously defined by using the AT command AT+CGDCONT= n,”IP”,”xxxx”
- UDP packets greater > 1536 bytes in size will be truncated. If a single UDP packet received is larger than this, then only the first 1536 bytes will be displayed\forwarded.
- The service pin should only be used for debug purposes due to the loading effect of the extra level of information on the channel.
- When using software versions R5B003 or earlier the application must implement the DTR line to switch in and out of on line command and data modes.
- Software R5B009 or newer are able to use the escape sequence (+++at) when turning it on through the at*e2ips command.

2.2 Modes Of operation.

Whilst using the AT based TCP/IP commands the serial port has, as normal, two modes of operation, data mode and on-line command mode.

- The user must assume that, once an IP connection has been established, anything received over the serial port when in normal data mode for TCP/IP is destined to be sent via the currently active socket. Equally, anything received by the host over the serial link is assumed to be the data received via the currently active socket.
- The use of the DTR line to switch into and out of on line command mode by the Host allows, at any point during the data communications, to enter on line command mode, modify and interrogate IP (or any other) settings and then change back DTR to revert to on line data mode by issuing the ATO command.
- When in on line command mode it is possible to read SMS and answer circuit switched calls during a GPRS session (while using the internal TCP stack or an external one).

If an unsolicited status change occurs - such as a socket closure or timeout error – the active data session will be terminated, and will revert to offline command mode, using a NO CARRIER response. The error causing this closure will be stored and can be analysed using the Error reporting AT command.

3 Use of the commands

3.1 Mobile Originated GPRS session

In a normal application it is envisaged that the TCP/IP functions will be used as follows:

1. If not already performed, define a PDP context for application use by sending AT command AT+CGDCONT=n,”IP”,”xxxx”
2. If the server that you are trying to connect to through the APN requires a username and/or password then use the following at*enad command
   AT*ENAD=x,”GPRS x”,”username”,”password”,1,0
3. Perform an IP Activate specifying the PDP Context ID. This will attach the module to the APN and allocate the module an IP Address. Connections can now be opened / closed.
4. Perform an IP Connect specifying the type of connection (TCP or UDP), the remote IP address, and the remote port. For a TCP session an active connection will be established with the remote. For a UDP session the IP Connect call defines where subsequent data will be sent. After a successful call to IP Connect the module will be in data mode ready to communicate over the socket
5. Send transmit data to the remote by writing data in sequence to the serial port.
6. Receive data from the remote by reading the contents of the serial port input buffers
7. When the user wishes to issue commands, for instance to check the session status, de-assert the DTR line to go to online command mode.
8. Issue AT commands to check module status.
9. Reassert DTR and issue AT command ATO to drop back to data mode.
10. Perform more data transfer over the socket by reading and writing the serial port.
11. When the user wishes to close the session, de-assert the DTR line.
12. Close the session by using IP Close. The session will also close if the session is closed remotely, or an unrecoverable error occurs on the IP Connection.
13. Finish

3.2 Disconnection and retransmission

TCP provides reliable transmission through the use of re transmission of un acknowledged data. When the unit decides that the connection has gone it will return a NO CARRIER response and enter command mode. The module behaves in the following manner to determine when this should happen.

1. If the link at some point drops out i.e. server crashes, RF coverage is lost, etc, the unit will start re transmissions.
2. The first retransmission will occur 6 seconds after initial transmission, the second 13 seconds later, the third 26 seconds, the fourth 56 seconds later, from then on they will occur every 64 seconds.
3. The NO CARRIER response is returned after the appropriate number of retransmissions has been reached as determined by the AT*E2IPS command, the maximum number is 8 or never time out if it is set to 0.
4. After this, even though the application will have received the NO CARRIER response the lower layers of the software will still try to re transmit the data that is currently in its buffer if the unit is able to reconnect to the network i.e. comes back into RF coverage. This does not affect the application re connecting to another IP address i.e. a backup server or reconnecting to the same address.

If at any point a disconnect message is received the module will return NO CARRIER and go into command mode.

Obviously this does not apply to UDP as it is connectionless and unreliable.

3.3 IP Server Listen

The server listen function allows the unit to start a session with the network i.e. have an IP address assigned to it, not connect to anything but wait for the unit to receive an incoming request for communication over IP.

1. Start a session with the network APN as previously described.
2. Run the listen command (AT*E2IPL) which enables the unit to wait in on line data mode for information addressed to it on a specific port. This can be set to run indefinitely or to time out (max 255 seconds).
3. Once it receives a request for a connection, the IP address requesting the connection will be passed back to the application and the unit will start to negotiate a socket connection.
4. If the application does not want to connect to the IP address it will need to go into on line command mode and close the socket. If it does want to continue communication a CONNECT will be returned and data can be sent/received as normal on line data mode.
5. When the session is terminated from the other end, the unit will return DISCONNECT to the application. If the application wants to end the

session it needs to go into on line command mode and terminate the socket using the appropriate command.

6. Finish.

3.4 SMS/Call handling

The unit can handle a circuit switched call during a TCP/IP session.

1. Start a session with the network APN as previously described.
2. During the session a call is received, the only indications of this will be that the unit will stop receiving data and the RI line will start toggling as per a normal call. The RI line can also be setup to pulse on reception of an SMS using the AT*E2SMSRI command.
3. To handle the call you either need to wait for it to time out and go to voice mail at which point the GPRS session will resume. The call can also be either hung up or answered by entering on line command mode (toggling DTR) and using the multiparty commands, these are shown in the examples in the following section.
4. When in command mode SMS's can be read normally,
5. Finish.

Note : The serial port can only handle one data connection for each AT session active at any one time. As a result the unit is unable to answer GSM circuit switched data calls.

4 AT command examples

4.1 MO GPRS session

An example of the AT command sequence for carrying out the operation as described in section 3.1 is as follows:

-> Setup PDP Context (in this example T-mobile's APN in UK)

AT+CGDCONT = 1,"IP","general.t-mobile.uk"

OK

-> If required enter the username and password for the AT*ENAD command where (x) is the same as the PDP context number
AT*ENAD=1,"GPRS 1","username","password",x,0 (also see note below)

OK

-> Activate IP

AT*E2IPA=1,1

OK

-> Now we are IP active we can read the IP status of the module.

AT*E2IPI=0

*E2IPI: 10.123.12.234

OK

-> We can also resolve a host address.

AT*E2IPRH = "www.google.co.uk"

*E2IPRH: 129.59.217.99

OK

-> We can open a TCP connection to google's HTTP Port (80)

AT*E2IPO = 1,"129.59.217.99",80

CONNECT

-> We are now in data mode, DCD is active and data sent now will go to Port 80 on Google's server, and any reply information will be received to the AT port.

-> We can now switch back to online command mode by de-asserting the DTR line –

OK

-> Now back in online command mode (DCD is still active)

-> Find out the primary DNS server address.

AT*E2IPI=1

*E2IPI: 129.1.13.100

OK

-> Check the Connection is still active…

AT*E2IPO?

*E2IPO: 1

OK

-> We can go back to online mode..

ATO

CONNECT

-> If the remote host closes the connection we revert to off-line command mode..

NO CARRIER

-> Now DCD is de-asserted (inactive). At any time the socket/connection is dropped the DCD pin will go inactive. We can either start a new connection, or deactivate the IP session..

AT*E2IPA=0,1

OK

-> IP session now inactive.

Finish Sequence.

4.1.1 Usernames and passwords with @

Some APN's/servers use usernames and passwords in the format of E mail addresses i.e. sony@ericsson.com, under these circumstances the modules are not able to perform the translation from the GSM alphabet to ASCII which most servers require these parameters to be sent in, as a result the @ character is not interpreted correctly.

A work around for this is to change the character set of the module to UTF-8 as shown below and then enter the enad command with the application using ASCII.

AT+CSCS="UTF-8"

OK

AT*ENAD=1,"GPRS 1","username@SEM.com","pw@SEM.com",x,0

OK

4.1.2 On line command mode

There are currently 2 methods for entering command mode, firstly as was shown in the previous example DTR can be toggled. The second method is to use the escape sequence of +++AT. This is, by default, is turned off, the unit needs to be instructed to use it through the AT*E2IPS command as shown below.

```
at*e2ips?
*E2IPS: 2,8,2,1020,0

OK

at*e2ips=2,8,2,1020,1
OK
```

4.2 IP server listen

4.2.1 TCP connection

An example of the IP server listen functionality is shown below.

at*e2ipa=1,1

OK

-> GPRS session started as normal

at*e2ipl=0,1,5001,0

-> Set the unit to start listening to port 5001 for TCP traffic

*E2IPL: Listening on IP Port: 5001

*E2IPL: IP Port: 35338 IP Address: 10.132.42.214

CONNECT

-> Unit is now able to send and receive data as normal with the socket that has been setup.

abcdefghijklmnopqrstuvwxyz

NO CARRIER

OK

-> The socket has now been terminated from the end that made the connection if another listen session is required it must be initiated by the application.

4.2.2 UDP connection

The operation of UDP listen is similar to TCP listen other than because there is no negotiation or request for transmission it needs to be predetermined what port the unit is going to be transmitting on prior to any data being sent/received.

The port the unit will transmit on will be that which the source port of the first packet is set to.

Because of the connectionless nature of UDP the unit will transmit blindly even if it is receiving destination unreachable messages from the network. It is highly recommended that some form of response from the server it is connecting to is used.

4.3 Call/SMS handling during a session

An example of a unit handling a circuit switched call is shown below.

-> The RI line is set to pulse for 1 second on reception of an SMS

at*e2smsri=1000

OK

-> A Normal session is started as previously described

at*e2ipa=1,1
OK

-> Connection made to a web server

at*e2ipo=1,"216.239.41.99",80
CONNECT

HTTP/1.0 200 OK

Cache-Control: private

Content-Type: text/html

Set-Cookie:
PREF=ID=3d9e22dd3b8e2f08:LD=en:TM=1070905485:LM=1070905485:
S=pIjdfuq3RsFLtkZU; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/;
domain=.google.co.uk

Server: GWS/2.1

Content-Length: 3092

Date: Mon, 08 Dec 2003 17:44:45 GMT

Connection: keep-alive

-> During the session the incoming data stops and the RI line pulses as per a normal incoming call (or pulses for an SMS). The unit then needs to enter on line command mode

OK

RING

-> In command mode the unit will receive the normal unsolicited responses and is then able to answer the call.

ata
OK

-> When the call is finished the application must terminate it using the multi party command below, if an ordinary ATH command is used it will hang up all calls , including the GPRS session.

at+chld=1
OK

-> The unit is now able to go back into on line command mode and will receive any data that has been stored in the buffer or that is incoming.

ato

CONNECT

HTTP/1.0 200 OK

Cache-Control: private

Content-Type: text/html

Set-Cookie: PREF=ID=329014d0584d6cef:LD=en:TM=1070905525:LM=1070905525:S=wvN_S-o5-jA593ya; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.co.uk

Server: GWS/2.1

Content-Length: 3092

Date: Mon, 08 Dec 2003 17:45:25 GMT

Connection: keep-alive

OK

at*e2ipa=0,1

-> The session can now be closed down in an orderly fashion.

OK

-> Finish

5 References

A reasonable understanding of TCP/IP would be of benefit when implementing these AT commands and would make implementation an easier proposition. However most of the official documentation is available as RFC's from http://www.ietf.org and they can provide essential technical background for anyone requiring a deeper understanding of TCP/IP. A small sample of the more pertinent RFC's and their titles are outlined below but many more are available.

1. RFC 791 Internet protocol
2. RFC 792 Transmission Control Protocol
3. RFC 394 File Transfer Protocol
4. RFC 1180 TCP/IP Protocol

6 Software vs document revisions

The table below defines the

Document revision	Applicable software version
R1A	R4A021
R1B	R4A021
R1C	R5B009
R1D	R6Axxx