Application Note
Using AT commands to control TCP/IP stack on GR47/GR48

Sony Ericsson

First edition (June 2003)

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.

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

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

Only TCP Client Functions supported. No capability for TCP server functions such as listen and accept and the provision for only one socket (either UDP or TCP) to be supported on initial implementation.

- 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.

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.

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

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. 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.
3. 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
4. Send transmit data to the remote by writing data in sequence to the serial port.
5. Receive data from the remote by reading the contents of the serial port input buffers
6. 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.
7. Issue AT commands to check module status.
8. Reassert DTR and issue AT command ATO to drop back to data mode.
9. Perform more data transfer over the socket by reading and writing the serial port.
10. When the user wishes to close the session, de-assert the DTR line.
11. 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.
12. Finish

3.1 AT command example

An example of the AT command sequence for carrying out the above is as follows:

-> Setup PDP Context (in this example T-mobile's APN in UK)
AT+CGDCONT = 1,"IP","general.t-mobile.uk"
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 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