



iMODEM: COMMAND REFERENCE MANUAL

**Supported Commands, and Definitions
and References for
the following iMODEM Products:**

ANALOG iMODEM Models

CH2166

CH2166A

GPRS/RF iMODEM Models

CH2168

CH2168A

GENERAL DISCLAIMER

All efforts have been made to insure the accuracy of the material presented herein at the time of release. However, items described and/or contained in this document are subject to continuous improvement and change without notice and do not represent a commitment on the part of Cermetek Microelectronics, Inc. Cermetek Microelectronics, Inc will not be responsible for any loss or damages incurred related to the use of the information in this document.

Products described herein are *not* intended for use in life support appliances, devices, or systems where a malfunction of the product, device or system can reasonably be expected to result in personal injury or death. Any Cermetek Microelectronics, Inc customers using, integrating, and/or selling products described herein for use in such applications, devices or systems do so at their own risk and agree to fully indemnify and hold harmless Cermetek Microelectronics, Inc for any damages resulting from illegal or unauthorized use or resale of the products described herein.

DOCUMENT CONTENTS

1. Introduction
 - 1.1 Document Scope
 - 1.2 Modes of Operation
 - 1.3 Analog iMODEM Internet Connectivity
 - 1.4 GPRS/RF iMODEM Internet Connectivity
 - 1.5 Profiles
 - 1.6 API Command Syntax
 - 1.7 Sessions
 - 1.8 Concurrent Operations and Sessions
2. API Characteristic Commands
 - 2.1 G_SET_ECHO Set iMODEM Echoing.
 - 2.2 G_SET_ERROR_MODE Set verbose iMODEM Responses.
 - 2.3 G_SET_LT_MODE Set Line Termination Characters.
3. Mode Control Commands
 - 3.1 @IMENTER Enter Modem Mode.
 - 3.2 @IMEXIT Return to Internet Mode.
4. Session Management Commands
 - 4.1 G_GET_SESS_ID Allocate a Session ID.
 - 4.2 G_FREE_SESS_ID De-allocate a Session ID.
5. E-Mail Transmission Commands (SMTP)
 - 5.1 ETX_CONNECT Enable IP/SMTP Connectivity.
 - 5.2 ETX_DISCONNECT Terminate IP/SMTP Connectivity.
 - 5.3 ETX_SEND_FILE Send a Stored E-Mail.
 - 5.4 ETX_START_DATA Start a Streaming E-Mail Session.
 - 5.5 ETX_END_DATA End a Streaming E-Mail Session.
 - 5.6 ETX_SEND_DATA Send a Line of Data.
6. E-Mail Storage Commands
 - 6.1 Storage Operation Commands
 - 6.1.1 ES_OPEN_FILE Open an E-Mail File.
 - 6.1.2 ES_OPEN_ATTACH Open an Attachment File.
 - 6.1.3 ES_CLOSE_FILE Close a File.
 - 6.1.4 ES_WRITE_DATA Write a Line of Data.
 - 6.1.5 ES_READ_DATA Read a Line of Data.
 - 6.2 Storage Management Commands
 - 6.2.1 ES_LIST_FILE List E-Mail Files.
 - 6.2.2 ES_LIST_ATTACH List attachment Files.
 - 6.2.3 ES_DELETE_FILE Delete an E-Mail File.
 - 6.2.4 ES_DELETE_ATTACH Delete an Attachment File.

DOCUMENT CONTENTS (continued)

7. ISOCKET API Commands

7.1	IS_IPCONNECT	Establish IP Connectivity.
7.2	IS_IPRELEASE	Terminate IP Connectivity.
7.3	IS_SOCKET	Allocate a Socket.
7.4	IS_BIND	Bind to an address.
7.5	IS_CONNECT	Make a TCP Connection.
7.6	IS_CLOSE	Release a Socket and TCP Connection
7.7	IS_RECV	Receive data.
7.8	IS_SEND	Send data.
7.9	IS_GETHOSTBYNAME	Perform DNS resolution
7.10	IS_GETHOSTNAME	Get the local host name
7.11	IS_GETSOCKNAME	Get local host address data.

8. E-Mail Retrieval Commands (POP3)

8.1	ERX_CONNECT	Establish IP/POP3 Connectivity.
8.2	ERX_DISCONNECT	Terminate IP/POP3 Connectivity.
8.3	ERX_MAIL_MSG	Retrieve a message.
8.4	ERX_MSG_DATA	Receive a line of data.
8.5	ERX_LIST_MSGS	List all messages.

9. Configuration Commands

9.1	LINK_CONFIG	Configure the Link Profile.
9.2	ETX_CONFIG	Configure the SMTP Profile.
9.3	ERX_CONFIG	Configure the POP3 Profile.

10. Utility Commands

10.1	@SHOW	Display Profile Settings.
10.2	IM_SET_MODEM	Set device attributes.

11. Analog iMODEM Specific Commands

11.1 Analog iMODEM Proprietary Commands

11.1.1	@SEND	Send stored Data
--------	-------	------------------

11.2 Single Line Analog iMODEM Modem Commands

11.3 Standard Analog iMODEM Modem Commands

12. GPRS/RF iMODEM Specific Commands

12.1 GPRS/RF iMODEM - Modem Mode Commands

GPRS/GSM General, Network Service, & Diagnostics

12.1.1	AT+CSQ	Signal Quality and Bit Error Rate
12.1.2	AT+CREG	Network Registration Info.
12.1.3	AT+CGREG	GPRS Network Registration Status
12.1.4	AT+COPS	Operator Selection
12.1.5	AT%CGREG	GPRS Extended Registration State
12.1.6	AT+CIMI	Request IMSI

DOCUMENT CONTENTS (continued)

GSM Call Control

- 12.1.7 AT+CEER Extended Error Reporting
- 12.1.8 AT+CPOL Preferred Operator List
- 12.1.9 AT+CLCC List Current Calls

GSM Modem Equipment Errors

- 12.1.10 AT+CMEE Report Mobile Equipment Errors

GPRS/GSM Standard Commands

- 12.1.11 AT+CGPADDR Show PDP Address
- 12.1.12 AT\$LOCIP Display Local Modem to Host IP & DNS
- 12.1.13 AT\$NETIP Display Network Assigned IP & DNS
- 12.1.14 AT\$GATEWAY Gateway IP
- 12.2 Standard GPRS/RF iMODEM Modem Commands

13. HWX – Hardware Event Signaling Interface

- 13.1 HWX Hardware
- 13.2 HWX API Command Overview
- 13.3 Trigger Condition Configuration Commands
 - 13.3.1 ADC_TRIGGER_CONFIG: Configure ADC Input Line Trigger Condition
 - 13.3.2 STATE_TRIGGER_CONFIG: Configure State Input Line Trigger Conditions
- 13.4 Low Level ADC Configuration Command
- 13.5 Response Configuration Commands
 - 13.5.1 HWX_RESPONSE_CONFIG: Configure the iMODEM Trigger Condition Response
 - 13.5.2 TOUT_CONFIG: Configure the TRIGGER OUTPUT
- 13.6 Device Control API Commands
 - 13.6.1 HWX_START: Start operation of an HWX device.
 - 13.6.2 HWX_AUTO_CONFIG: Configure an HWX device to automatically start.
 - 13.6.3 HWX_STOP: Stop operation of a specific HWX input device.
 - 13.6.4 HWX_SHUTDOWN: Stop operation of all HWX input devices.
 - 13.6.5 HWX_TRIGGER: Actuate the OUTPUT TRIGGER.
- 13.7 User Notification API Messages
 - 13.7.1 EVENT: Report a Trigger Condition to the iMODEM host.
 - 13.7.2 OK Acknowledgement.
 - 13.7.3 FAULT: Report an HWX device problem.

14. Miscellaneous Test Commands

- 14.1 @QTM Send Stored E-Mail.
- 14.2 @QRX Retrieve an E-Mail Message.

DOCUMENT CONTENTS (continued)

Appendix A - iMODEM API Error Codes

Appendix B - Release Causes for Extended Error Reporting (+CEER)

Appendix C - GPRS/GSM ERROR CODES

Appendix D - References

1. INTRODUCTION

1.1 Document Scope

This document describes the API Commands for the Analog and GPRS/RF iMODEM products.

IMPORTANT NOTE

The Analog iMODEM models use V.92 modems for data transmission. The GPRS/RF iMODEM models are Class B GPRS devices intended for use on transparent GPRS Networks.

The Analog iMODEM products use telephone lines as a physical transmission media, the PPP protocol and a PSTN (Public Switched Telephone Network) to reach the Internet. An ISP POP (Point of Presence) is the Analog iMODEM's entry point to the Internet. The Analog iMODEM can also function as an analog modem over PSTN telephone lines.

The GPRS/RF iMODEM products use GSM RF as a physical transmission media, GPRS protocols and a PLMN (Public Land Mobile Network) to reach the Internet. A GGSN (Gateway GPRS Serving Node) Gateway is the GPRS/RF iMODEM's entry point to the Internet. However, the GPRS/RF iMODEM uses RF only for direct communications with a Base Station connected to cellular tower equipment. The Base Station is the entry point into the GPRS PLMN network.

IMPORTANT NOTE

For the purposes of simplicity, the term GPRS Network is used throughout this document to refer to a GPRS PLMN system.

1.2 Modes of Operation.

iMODEM family members can operate in either Internet or Modem mode and Internet mode is the default mode. Data is conveyed between the iMODEM and Internet hosts with the TCP/IP family of protocols in Internet mode.

The Analog iMODEM functions as a standard V.92 analog modem in Modem mode. The GPRS/RF iMODEM can use standard GPRS/GSM AT commands to perform configuration and diagnostic tests in Modem mode.

All iMODEM API commands described by this document except the IM_EXIT, and those in sections 11 and 12, unless specifically noted, are Internet mode commands.

1.3 Analog iMODEM Internet Connectivity

The Analog iMODEM communicates with an ISP POP over a PSTN network with V.92 analog modem protocols. Data, in the form of IP datagrams, is transmitted between the iMODEM and ISP POP with the PPP protocol. The ISP POP has Internet connectivity, and conveys IP datagrams from the iMODEM across the Internet to an Internet host and vice versa.

IMPORTANT NOTE

The term POP (Point Of Presence) refers to the set of systems which perform authentication, and form the PSTN to Internet interface for dialup connections. A POP always includes analog modem banks but the exact distribution of the systems and the specific functionality is ISP dependent, and follows the RADIUS model per RFC2133. Typically, the POP contains modem banks and is the front end of a NAS (Network Access Server). The NAS is the boundary between the PSTN and the Internet. Actual authentication is performed via transactions between the NAS and an authentication server.

IP connectivity must be established before the iMODEM can make TCP connections and communicate with other Internet hosts. The following things must occur to establish IP connectivity.

- A dialup connection must be established between the iMODEM and ISP POP.
- The iMODEM may need to authenticate with the ISP that operates the ISP POP. This can occur before or during PPP operation.
- A PPP connection must be established between the iMODEM and ISP POP.
- An IP address must be allocated to the iMODEM.

Figure 1.3A illustrates how the iMODEM is connected to the Internet, and Figure 1.3B illustrates what protocols are used.

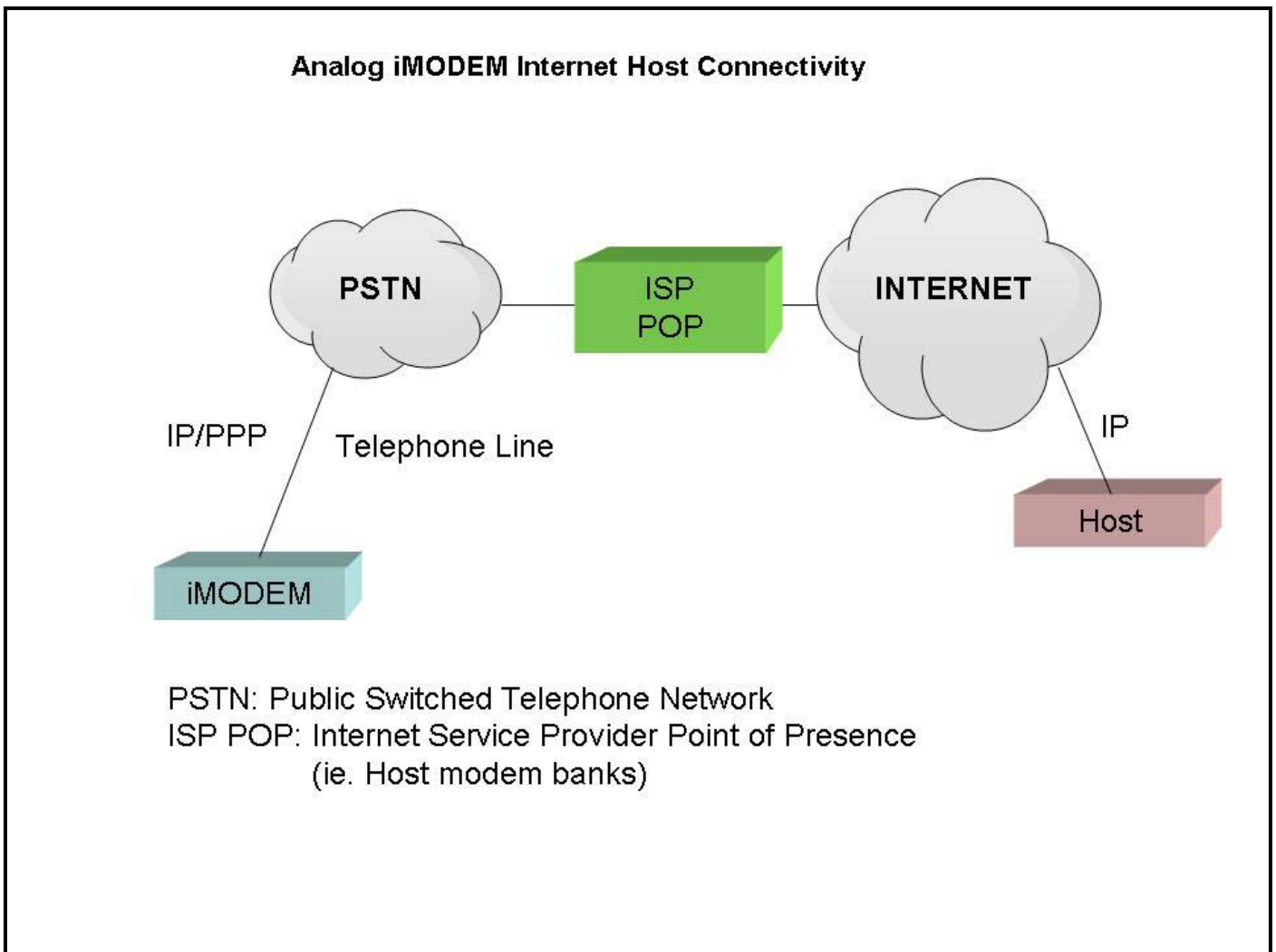


Figure 1.3A

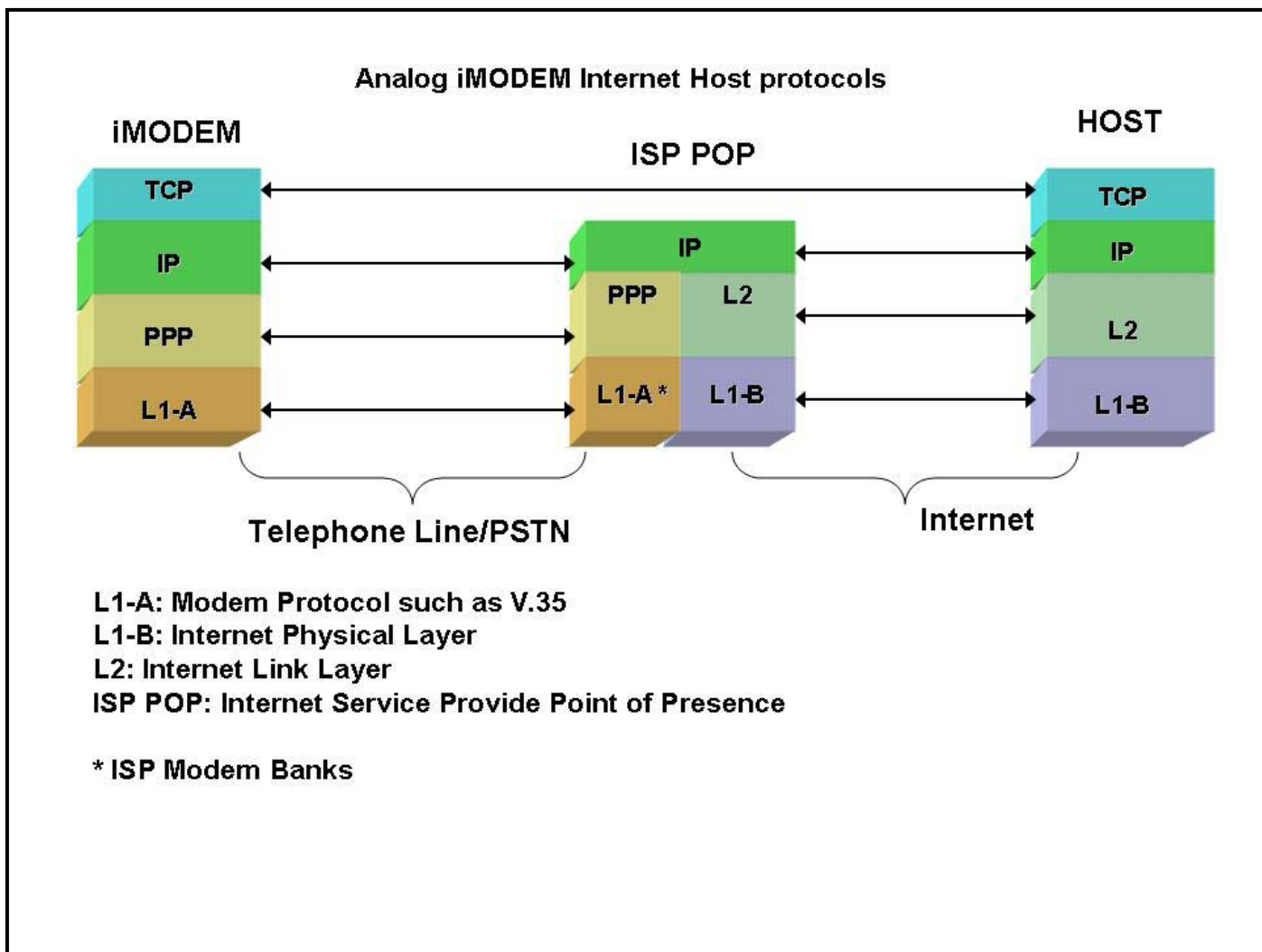


Figure 1.3B

1.4 GPRS/RF iMODEM Internet Connectivity

The GPRS/RF iMODEM communicates with a Base Station connected to cellular tower equipment over the air with GSM RF. Data, in the form of IP datagrams, is transmitted directly between the iMODEM and Base Station in LLC frames with the RLC protocol. The Base Station conveys the LLC frames from the iMODEM, across the GPRS Network to an Internet Gateway, and vice versa. IP datagrams exchanged between the Gateway and an Internet host are transmitted across the Internet with whatever transmission media and link layer are native to the Internet segments they traverse.

IP connectivity must be established before the iMODEM can make TCP connections and communicate with other Internet hosts. The following two groups of operations must occur to establish IP connectivity.

The first group occurs during GPRS network registration with the following operations performed during that process. Many ETSI standards call the process network attachment. Registration automatically occurs shortly after good RF signal quality is achieved.

- Establishing RF connectivity with a Base Station.
- Acquiring permission to use the network.
- Conveying the iMODEM location and capabilities to the network.
- Updating network registries

The second group occurs during PDP (Packet Data Protocol) attachment and activation and is triggered by an explicit iMODEM attempt to establish IP connectivity. The following operations are performed during that process:

- Making an internal iMODEM Pseudo-PPP connection.
- Allocation of the network resources needed for IP packet transfer.
- Establishing connectivity with an Internet Gateway.
- Assigning an IP address to the iMODEM.

IMPORTANT NOTES

The Pseudo-PPP connection is internal to the iMODEM and is essentially an interface. The iMODEM is divided into TE (Terminal Equipment) and ME (Mobile Equipment) parts. The TE contains the user APIs and a TCP/IP stack, and exchanges data with the ME through the Pseudo-PPP connection. The ME contains all the RF components and the LLC layer.

Figure 1.4A illustrates how the iMODEM is connected to the Internet, and Figure 1.4B illustrates the protocols utilized. Figure 1.4C describes the internal organization of the iMODEM.

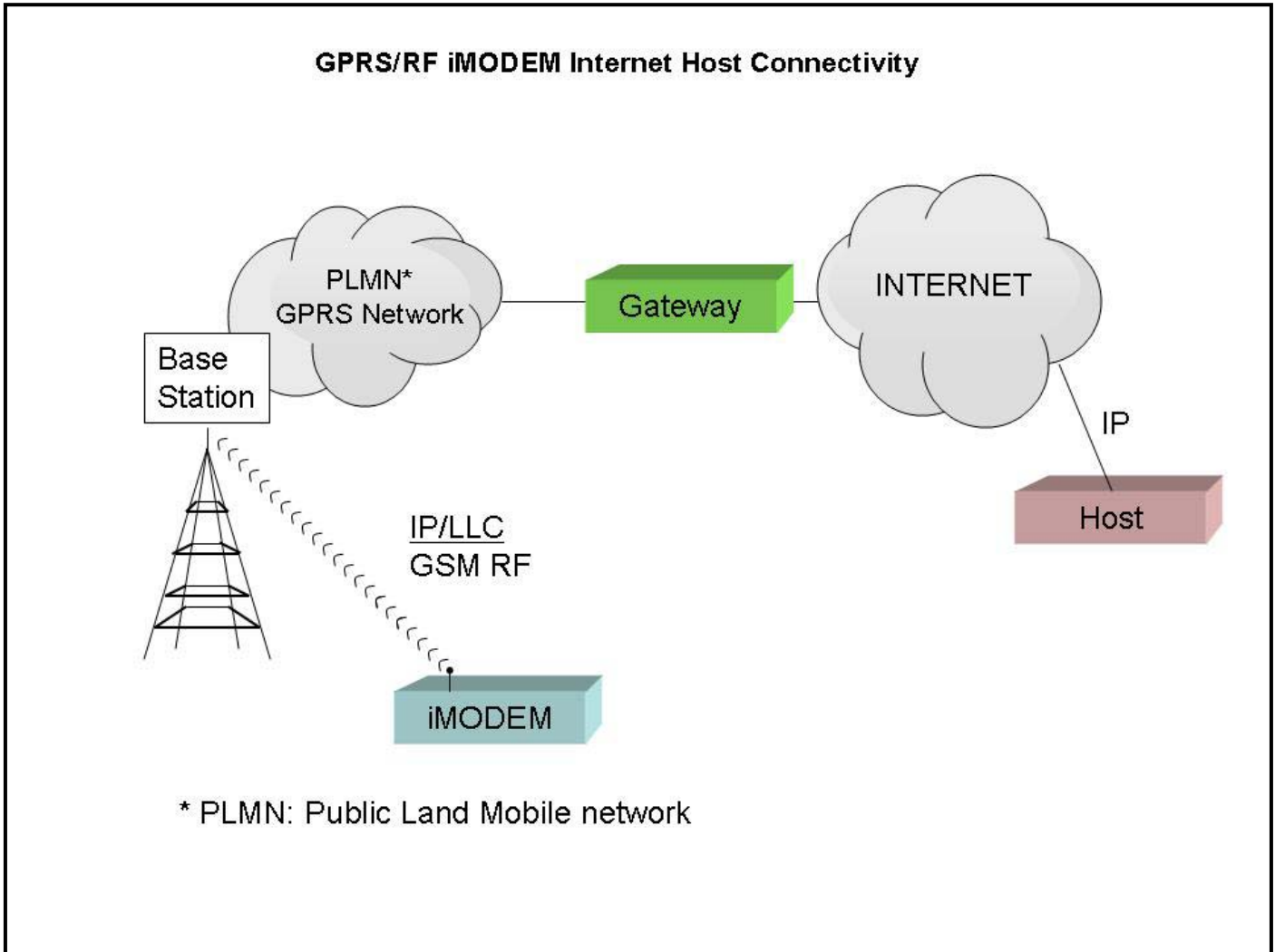


Figure 1.4A

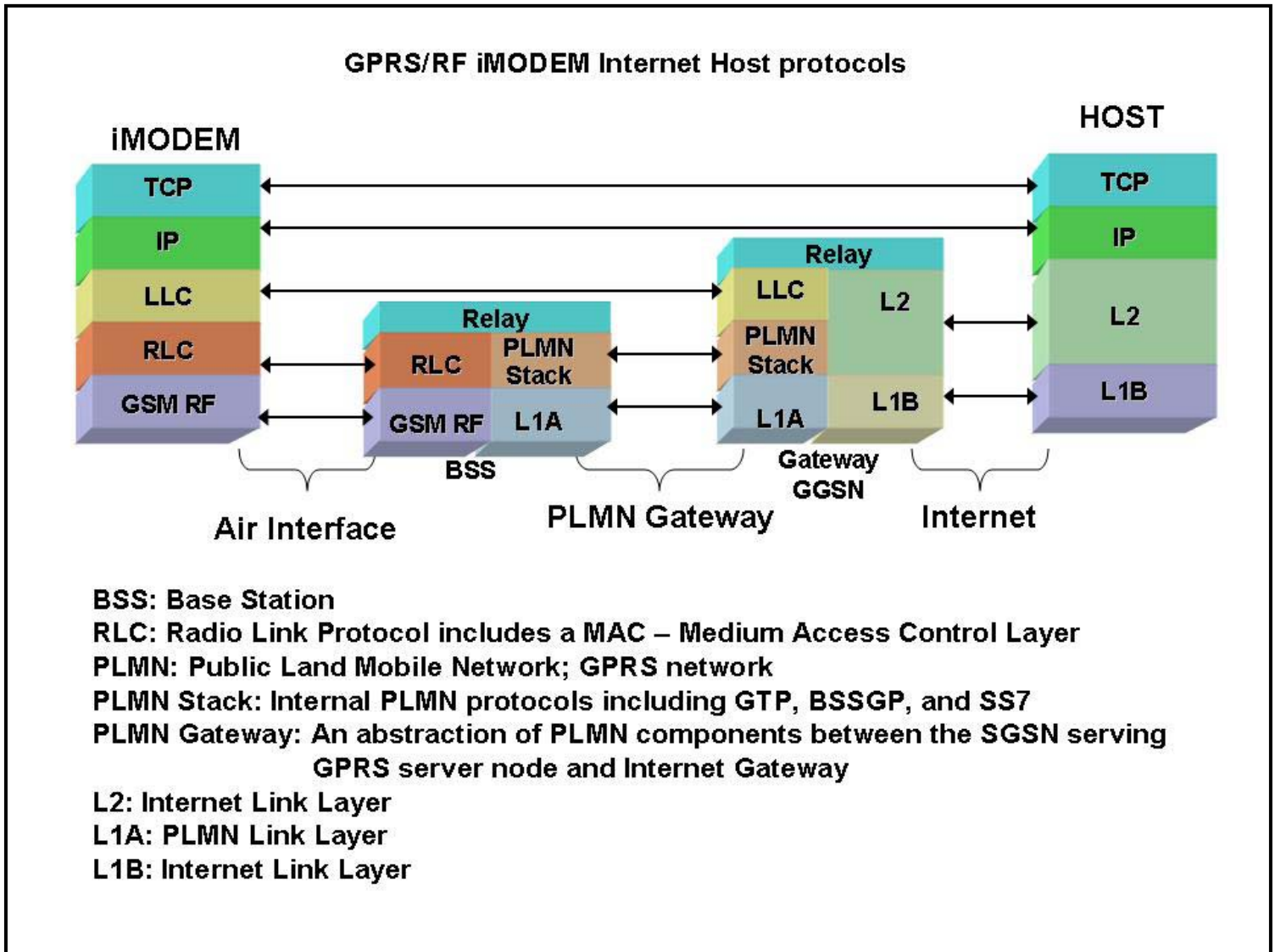


Figure 1.4B

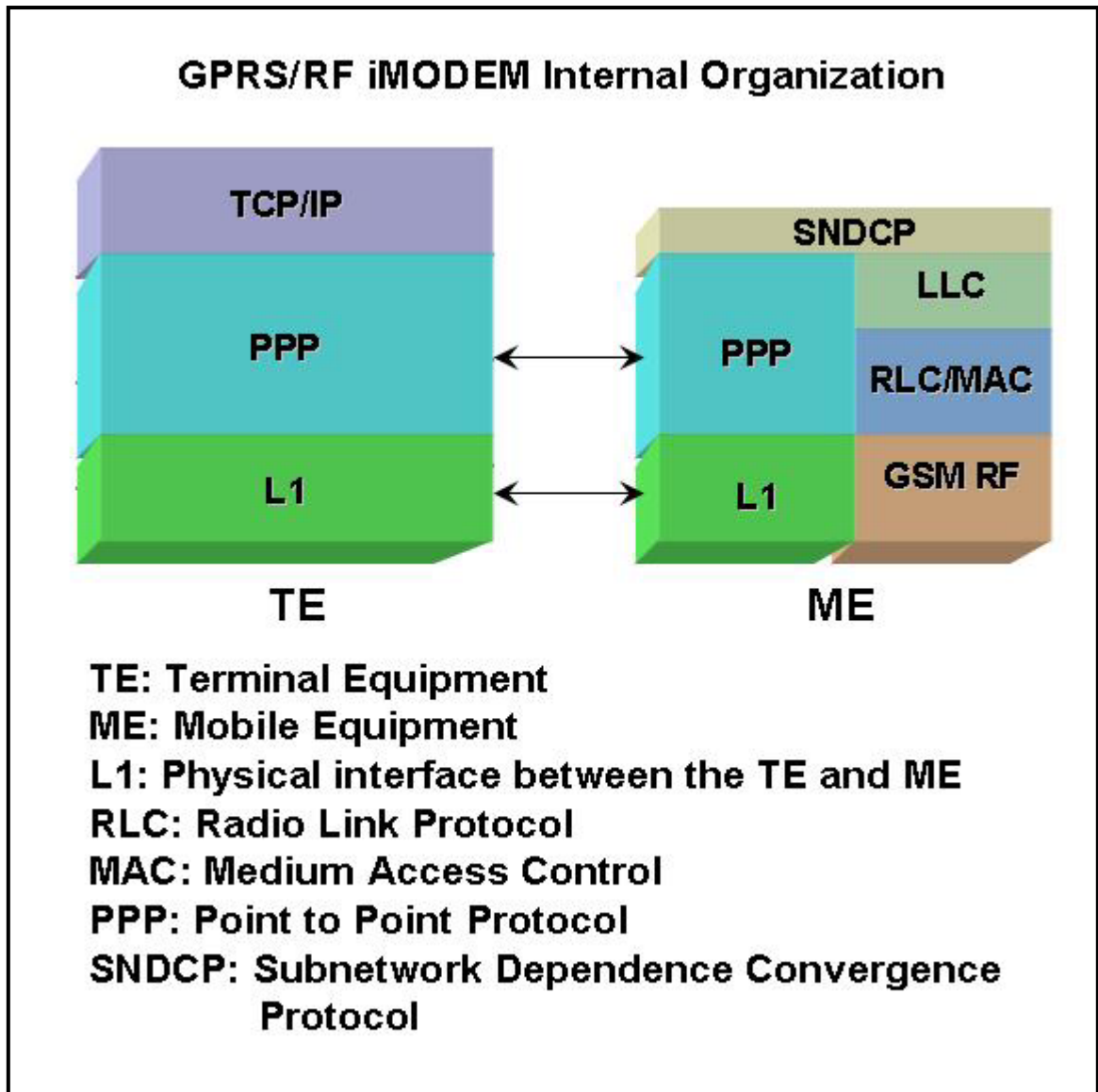


Figure 1.4C

1.5 Profiles

iMODEM configuration data is stored in on board profiles, and there are two basic types of profiles:

- Application Profiles contain the configuration data for a standard iMODEM user application.
- Link Profiles contain configuration data for the network service provider, the transmission media and related low level protocols.

Figure 1.5A illustrates how the iMODEM profiles are organized.

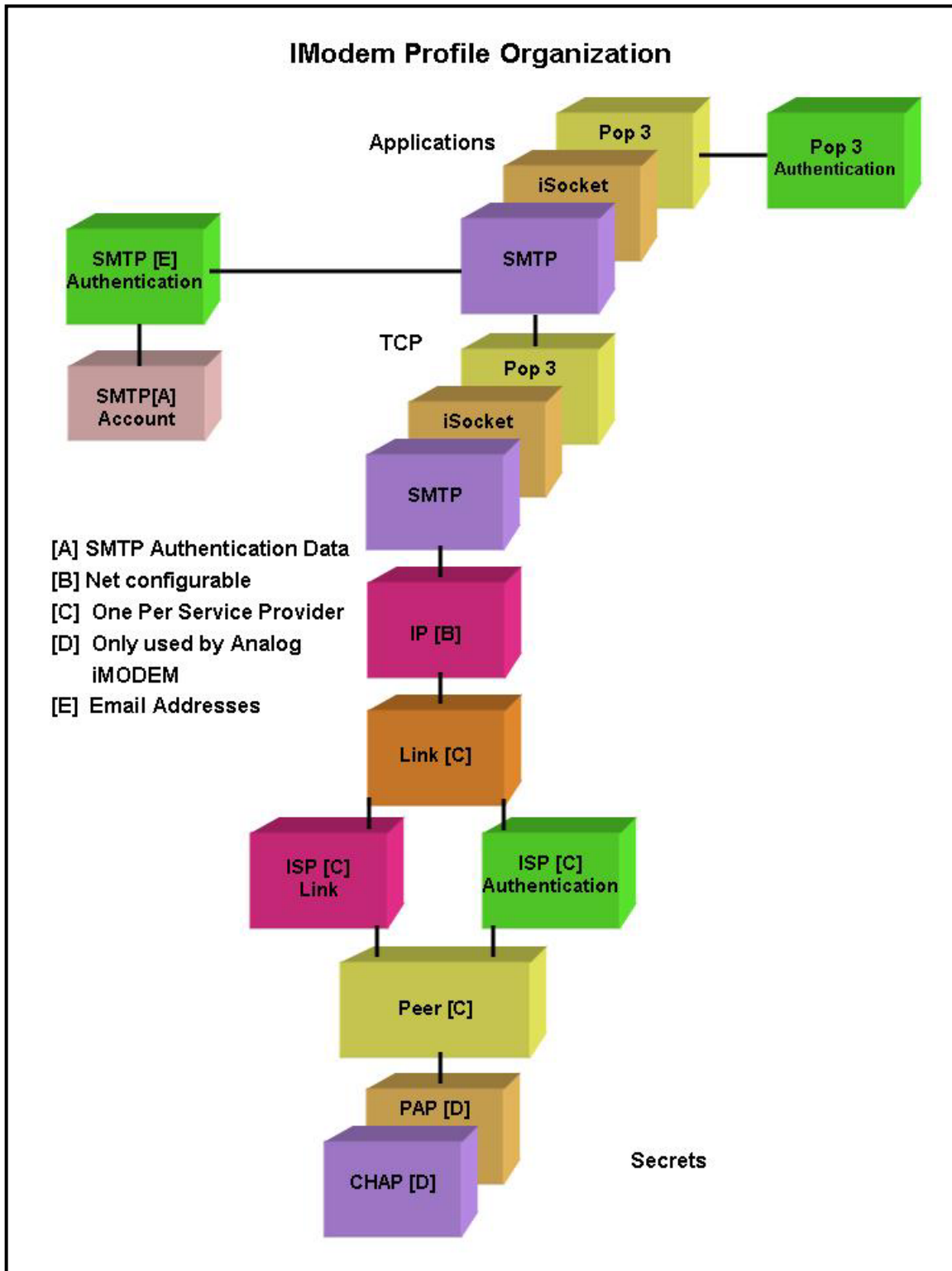


Figure 1.5A

The standard iMODEM Application Profiles are sometimes known as "Profile Sets" and are as follows:

- SMTP E-Mail Transmission - The SMTP Profile
- POP3 E-Mail Retrieval - The POP3 Profile
- ISOCKET API - The ISOCKET Profile

The Analog iMODEM and GPRS/RF iMODEM link profiles are quite different and their basic functionality is as follows:

- Analog iMODEM Link Profile: Configuration related to ISP authentication, the PPP protocol, and analog modem dialup operations.
- GPRS/RF iMODEM Link Profile: Configuration related to GPRS Network access, resource allocation, and the Pseudo-PPP protocol used to control LLC protocol and GSM/RF operations.

A single Link Profile and a set of standard iMODEM Application Profiles form a "Provider Profile".

IMPORTANT NOTE

Generally, the iModem is configured for a single ISP or GPRS Network provider. Consequently, a single Link Profile is normally shared by all Provider Profiles.

1.6 API Command Syntax.

All Internet mode API commands use the syntax covered in this section. The Modem mode command syntax is described in the sections that cover those commands.

There are two categories of API commands: General, and Session. A Session command includes a "Session ID" which identifies the instance of the standard iMODEM application to which it applies.

IMPORTANT NOTE

Variable components of API commands and responses are represented herein by enclosing them with angle brackets '<' and '>'. This is done for clarity only and angle bracket characters are not part of the command or response.

The syntax for General Commands is as follows:

A. General Command with no parameters:

<Command_Name><LT>

B. General Command with parameters:

<Command_Name>: <Par-1> <Par-2> ... <Par-N><LT>

C. Session command with no parameters.

<Command_Name>[<SID>]<LT>

D. Session command with parameters.

<Command_Name>[<SID>]<LT>: <Par-1> <Par-2> ... <Par-N><LT>

IMPORTANT NOTE

<CR> and <LF> denote the carriage return and the line feed characters respectively. <LT> denotes a line termination sequence. By default, <LT> consists of a single <CR> but, depending on how the iMODEM is configured, it can also be a single <LF>, or a <CR><LF> pair.

The components of commands are as follows:

- Command_Name: A string of alphabetic, '@', and '_' characters without spaces or tabs forming the name of an API command.
- Par: A command parameter which can have the following two forms:

<Par_Name>

or

<Par_Name>=<Par_Value>

- Par_Name is the parameter name, and Par_Value is the parameter value. Parameters are separated by spaces and tabs and there must be no space around the '=' character. Par_Value may be a quoted string as in the following case:

SUBJECT="Message of The Day"

Any printable ASCII characters, except for the following, are legal in a quoted string:

Quote Character: ""
Backslash: \'

- SID: A Session ID which is a number greater or equal to 0, in ASCII format,

Example 1.6A: General Command with no Parameters

```
IM_SET_MODEM
```

Example 1.6B: General Command with Parameters.

```
@SHOW: PROFILE=smtp.Imi LEVEL=ALL
```

Example 1.6C: Session Command with no Parameters

```
IS_IPCONNECT[1]
```

Example 1.6D: Session Command with Parameters

```
ETX_SEND_FILE[0]: PROFILE=smtp.Imi FILE=test
```

The iMODEM initially responds to an API command with the message "POSTED", and then "OK" or "ERROR" depending on the command outcome. POSTED indicates that the iMODEM has begun executing the command, OK indicates the command succeeded, and ERROR indicates it failed.

General Command responses have the following forms:

IMPORTANT NOTE

<LT> is <CR><LF> by default for iMODEM responses.

A. Positive General Command Response

```
POSTED: <Message> <LT>
OK:<Message><LT>
```

B. Positive Session Command Response

```
POSTED[<SID>]:<Message><LT>
OK[<SID>]: <Message><LT>
```

C. Negative General Command Response

```
POSTED:<Message><LT>
ERROR <Code>: <Message><LT>
```

D. Negative Session Command Response

```
POSTED[<SID>]: <Message><LT>
ERROR[<SID>]: <Code>: <Message><LT>
```

Message: An optional brief message describing the command outcome.

Code: A numeric error code in ASCII for error code in ASCII format.

Example 1.6E: Positive General Command Response.

POSTED: Operation Started
 OK: Configuration OK

Example 1.6F: Negative Session Command Response.

POSTED[0]: Operation Started
 ERROR[0] 38: TCP Connection Failure

IMPORTANT NOTE

Appendix A contains a complete listing of iModem API error codes.

IMPORTANT NOTE

This document makes extensive use of Arrowgrams to illustrate the flow of iMODEM commands and responses between the user and iMODEM, and they typically have two columns.

The left hand column, labeled **USER**, contains commands issued by the iMODEM user or host.

The right hand column, labeled **iMODEM**, contains responses and data generated by the iMODEM.

Arrows between the right and left columns indicate the direction in which iMODEM API commands and responses flow. API commands and responses are implicitly terminated by a carriage return <CR>.

The '\ ' character at the end of either command line or response indicates continuation of that command or response on the next line. This facilitates column alignment and enhances readability.

The string "... " after an iMODEM response represents a descriptive message appended by the iMODEM. The iMODEM is configured to append such messages by default.

The example below is a typical Arrowgram.

Example 1.6G: Typical Arrowgram

<u>USER</u>		<u>iMODEM</u>
G_SET_ERROR_MODE: VERBOSE	==>	
	<==	POSTED: Operation Started
	<==	OK: Response mode reset.

The POSTED message indicates the iMODEM has begun executing the G_SET_ERROR_MODE command, and the OK message indicates the command completed successfully.

1.7 Sessions

A session is an instance of a standard iMODEM application, and multiple sessions can execute concurrently. It also allows the host CPU to perform other tasks during slow operations such as TCP connection establishment.

Each session is uniquely identified by a "Session ID" which is an ASCII format number greater or equal to 0. The Session ID is allocated with the G_GET_SESS_ID API command and de-allocated with the G_FREE_SESS_ID API command. A Session ID can be allocated for the following standard iMODEM applications:

IMPORTANT NOTE
Sessions only occur in Internet Mode

- SMTP E-Mail Transmission.
- E-Mail Storage.
- ISOCKET API.
- POP3 E-Mail Retrieval.
- HWX Hardware Event Signaling Interface.

The Session ID is included in iMODEM commands and responses so they can be associated with the correct session. The following example illustrates how the Session ID is used.

Example 1.7A: Use of a Session ID

<u>USER</u>		<u>iMODEM</u>
ETX_SEND_FILE[2]: FILE=test	==> <== <==	POSTED[2]: Operation Started OK[2]: E-Mail successfully sent

1.8 Concurrent Operations and Sessions

Allowing iMODEM sessions to execute concurrently has the following advantages for the iMODEM user or host:

- Multiple TCP/IP connections can easily be managed.
- The host CPU can perform other tasks during slow iMODEM operations, such as TCP connection establishment.
- The host can handle sporadic data and events quickly.

The iMODEM is in Multi-mode when 1 or more Session IDs are allocated, and supports concurrent operations. Otherwise, the iMODEM is in Uni-mode. Uni-mode which is the default. Both Multi-mode and Uni-mode are submodes of Internet mode.

The Session ID is what allows multiple iMODEM sessions to execute concurrently. It allows the iMODEM user to correlate API commands and responses because they have the same Session ID. However, this ability also imposes the following restrictions:

- The iMODEM user or host can not issue General API Commands during Multi-mode (i.e., Commands without a Session ID.)
- After issuing an API command, the user or host must wait for a POSTED response before issuing another API command or sending any data to the iMODEM.
- After issuing a session API command, the user or host must wait for an OK or ERROR response before issuing another API command with the same Session ID.

IMPORTANT NOTE

There is one exception to the above rules: The G_GET_SESS_ID command can be issued during concurrent execution as long as no other API command is executing.

The following example illustrates a simple case to two iMODEM sessions executing concurrently. It assumes Session ID 0 is allocated to Session A, and Session ID 1 is allocated to Session B.

Example 1.8A:: Two Concurrent iMODEM Applications. It is assumed the host has one CPU.

<u>USER</u>	==>	<u>iMODEM</u>
Session A starts data transmission		
ETX_SEND_DATA[0]:Data_A	==>	
Session A is notified the transmission has started. The host CPU must wait for POSTED before sending any data to the iMODEM.		
	<==	POSTED[0]: Operation Started
The host CPU can perform any task at this point Session B starts data transmission		
ETX_SEND_DATA[1]:Data_B	==>	
Session A is notified that transmission is complete		
	<==	OK[0]: Data sent
Session B is notified that transmission is started		
	<==	POSTED[1]: Operation Started
Session B is notified that transmission is complete		
	<==	OK[1]: Data Sent

After issuing a General API Command, the user or host can not issue another command to, or send data to, the iMODEM until the command completes. This restriction is quite different from the case of a Session API Commands. However, the host CPU can perform other tasks after receiving a POSTED response. The comments in the following example emphasize the general API command restrictions.

Example 1.8B: General API Command Restrictions

<u>USER</u>	==>	<u>iMODEM</u>
G_SET_ERROR_MODE: VERBOSE	==>	
The host or user must not send any data to the iMODEM during this period.		
	<==	POSTED: Operation Started
The host CPU can now perform other tasks However, it can not issue more API commands.		
	<==	OK: Response mode reset.
The host or user can now issue another API command		

2. API CHARACTERISTIC COMMANDS

These API commands are used to configure the behavior of the iMODEM API, and they are unique because they apply to the iMODEM as a device rather than a Profile. Also, the parameters they configure revert to their default values when the iMODEM loses power. In brief, they are as follows:

<u>COMMAND</u>	<u>ABBREVIATED FORM</u>
G_SET_ECHO: Set iMODEM Echoing.	@GSECHO
G_SET_ERROR_MODE: Set verbose iMODEM responses.	@GSLT
G_SET_LT_MODE: Set Line Termination Characters.	@GSERR

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. Appendix A lists the error codes appended to the ERROR message. The syntax and use of the commands are described below.

2.1 G_SET_ECHO: This command configures how the iMODEM echoes host input.

SYNTAX

G_SET_ECHO: <Echo_Mode>

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

PARAMETERS

Echo_Mode: ON and OFF are the valid values and take effect after the command completes. The iMODEM will echo host input if the value is ON and will not if the value is OFF. The default value is ON.

Example 2.1A: Assume Echoing is Currently ON.

<u>USER</u>		<u>iMODEM</u>
G_SET_ECHO: OFF	==>	
	<==	POSTED: Operation Started
	<==	OK: Echo mode reset.

2.2 G_SET_ERROR_MODE: This command configures the iMODEM to append descriptive messages to API command responses.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

G_SET_ERROR_MODE: <Response_Mode>

PARAMETERS

Response_Mode: BRIEF and VERBOSE are the valid values and take effect after the command completes. If the value is VERBOSE the iMODEM appends descriptive messages to API Command responses. Otherwise, when the value is BRIEF, it does not. The default value is VERBOSE.

Typical examples of messages the iMODEM appends to responses in VERBOSE mode are as follows:

POSTED: Operation Started
 OK: Data Sent
 ERROR 11: No free channels.

In BRIEF mode the same responses would appear as follows:

POSTED:
 OK:
 ERROR:

Example 2.2C: Assume the iMODEM is in BRIEF mode.

<u>USER</u>		<u>iMODEM</u>
G_SET_ERROR_MODE: VERBOSE	==>	
	<==	POSTED:
	<==	OK: Response mode reset.

2.3 G_SET_LT_MODE: Configures the line termination sequence expected and generated by the iModem. At least one of the parameters must be specified.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

G_SET_LT_MODE: INPUT=<Input_Lt> OUTPUT=<Output_Lt>

PARAMETERS

Input_Lt: The termination sequence appended to lines of input sent from the host to the iMODEM. The four possible value are as follows:

CR: Carriage Return--hex 0D.
 LF: Line Feed--hex 0A.
 CR_OR_LF: Either a Carriage Return or Line Feed. (i.e. The default.)
 CRLF: A Carriage Return followed by a Line Feed.

Output_Lt: The termination sequence appended to lines of output sent from the iMODEM to the host. The same four values as for Input_Lt are possible. However, CRLF is the default value, and works with most manual input programs such as HyperTerminal and ProComm.

Example 2.3A: Reset the defaults.

<u>USER</u>		<u>iMODEM</u>
G_SET_LT_MODE: INPUT=\ CR_OR_LF OUTPUT=CRLF	==>	
	<==	POSTED: Operation Started
	<==	OK: Line termination \ characteristics set.

3 MODE CONTROL COMMANDS

These API commands are used to switch the iMODEM between Internet and Modem mode.

COMMAND

- @IMENTER: Enter Modem Mode
- @IMEXIT: Return to Internet Mode

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

3.1 @IMENTER: Switches the iMODEM from Internet to Modem command mode.

USAGE - This command can not be issued while other API commands are executing. It can only be used in Internet mode.

SYNTAX

@IMENTER

PARAMETERS

None

Example 3.1A: Enter Modem Mode

<u>USER</u>		<u>iMODEM</u>
@IMENTER	==>	
	<==	POSTED: Operation Started
	<==	OK: Modem command mode \
		started

3.2 @IMEXIT: Switches the iMODEM from Modem command to Internet mode.

USAGE - This command can only be issued from Modem command mode. If the Analog iMODEM is not in command mode, it must be put in command mode with the appropriate AT commands.

SYNTAX

@IMEXIT

PARAMETERS

None

Example 3.2A: Exit Modem Mode to Internet Mode

<u>USER</u>		<u>iMODEM</u>
@IMEXIT	==>	
	<==	POSTED: Operation Started
	<==	OK: Exited modem mode.

4. SESSION MANAGEMENT COMMANDS

These commands allocate and de-allocate iMODEM Session IDs.

<u>COMMAND</u>	<u>ABBREVIATED FORM</u>
G_GET_SESS_ID: Allocate a Session ID.	@GIDGET
G_FREE_SESS_ID: Deallocate a Session ID.	@GIDFREE

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. Appendix A lists the error codes appended to the ERROR message. The syntax and use of the commands are described below.

4.1 G_GET_SESS_ID: Allocate a Session ID.

Successful completion of this command indicates the iMODEM has begun Multi-mode if no other Session IDs are allocated. It also begins an iMODEM application session instance.

USAGE - This command can only be issued if no other iMODEM API command is executing. It is unique because it may be issued after Multi-mode has begun.

SYNTAX

G_GET_SESS_ID: <App_Type>

PARAMETERS

App_Type: This parameter is mandatory and specifies a standard iMODEM application. The valid values are as follows:

TX_MAIL: SMTP E-Mail Transmission.
 STORE_MAIL: E-Mail message storage.
 RX_MAIL: POP3 E-Mail Retrieval.
 ISOCKET: ISOCKET API.
 HWX: Hardware Event Signaling Interface

Example 4.1A: Allocate a Session ID.

<u>USER</u>		<u>iMODEM</u>
G_GET_SESS_ID: TX_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: E-Mail Transmission \
		Session ID allocated.

0 is the newly allocated Session ID.

4.2 G_FREE_SESS_ID[<SID>]: De-allocate a Session ID.

If this command completes successfully and no other Session IDs are allocated, it terminates iMODEM Multi-mode. The application also ends an iMODEM application session instance.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

G_FREE_SESS_ID[<SID>]

PARAMETERS

None

Example 4.2A: De-allocate a Session ID.

<u>USER</u>		<u>iMODEM</u>
G_FREE_SESS_ID[1]	==>	
	<==	POSTED[1]: Operation Started
	<==	OK[1]: E-Mail Transmission \
		Session ID freed.

5 E-MAIL TRANSMISSION COMMANDS (SMTP)

These commands are used to send E-Mail messages via an SMTP server, and can only be used within an iMODEM session.

<u>COMMAND</u>	<u>ABBREVIATED FORM</u>
ETX_CONNECT: Enable IP/SMTP Connectivity.	@ETXCONN
ETX_DISCONNECT: Terminate IP/SMTP Connectivity.	@ETXDISC
ETX_SEND_FILE: Send Stored E-Mail.	@ETXFILE
ETX_START_DATA: Start a Streaming E-Mail Session.	@ETXDSTART
ETX_END_DATA: End a Streaming E-Mail Session.	@ETXDEND
ETX_SEND_DATA: Send a Line of Data.	@ETXDATA

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. Appendix A lists the error codes appended to the ERROR message. The syntax and use of the commands are described below.

5.1 ETX_CONNECT: Enable IP/SMTP connectivity

This command establishes IP connectivity. It enables the E-Mail transmission session instance to support E-Mail transactions. It does not actually establish SMTP server connectivity. After successful completion of the command, multiple SMTP E-Mail transactions can be performed within the same transmission session.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

Analog iMODEM Syntax

```
ETX_CONNECT[<SID>]: PROFILE=<Prof_Name>
SERVER=<Smtp_Server> PORT=<Tcp_Port>
AUTH_METHOD=<Smtp_Auth_Proto> SERVLOG=<Smtp_Auth_Data>
DIALUP=<Link_Pars> PHONE=<Phone_No>
OPENLOG=<Isp_Auth_Data> LINKLOG=<PPP_Auth_Data>
```

GPRS/RF iMODEM Syntax

```
ETX_CONNECT[<SID>]: PROFILE=<Prof_Name>
SERVER=<Smtp_Server> PORT=<Tcp_Port>
AUTH_METHOD=<Smtp_Auth_Proto> SERVLOG=<Smtp_Auth_Data>
DIALUP=<Link_Pars> APN=<APN_Name>
```

PARAMETERS

Prof_Name: Optional - The name of an SMTP profile such smtp.att. If this parameter is not specified, default SMTP parameters are used.

Smtplib_Server: Optional - An SMTP server name or IP address. It may have the following two formats:

- **Text Format:** SERVER=TEXT-<Server_Name> where Server_Name is the name of server such as smtp.earthlink.net.
- **IP4 Format:** SERVER=IPV4-<ip_address> where ip_address is the IPV4 format server Internet address such as 66.117.140.246.

Tcp_Port: Optional - A TCP port number.

Smtplib_Auth_Proto: Optional. Many SMTP servers require authentication. The parameter specifies what SMTP authentication protocol will be used with such servers. The valid values are as follows:

- **CRAM-MD5:** Authentication data is encrypted, and exchanged via the challenge mechanism described by RFCs 4422 and 1321.
- **PLAIN:** Authentication data is sent as base 64 encrypted strings. [A]
- **LOGIN:** Authentication data is sent as unencrypted ASCII strings. [A]
- **ANY:** The iMODEM and SMTP server automatically negotiate the best protocol which is recommended.
- **NONE:** The iMODEM does not perform authentication.

[A]: Not generally considered secure.

Smtplib_Auth_Data: This optional parameter is only used if SMTP authentication is selected as described above. It has the following format: <ID>^<Secret>. ID is the SMTP Login ID, and Secret is the SMTP secret or password.

Link_Pars: This optional parameter must be set to TRUE if any of the following parameters are used: PHONE, APN, OPENLOG, LINKLOG. Otherwise it must be omitted or set to FALSE.

Phone_No: This parameter is Analog iMODEM specific and optional. It is an ISP local access telephone number.

APN_Name: This parameter is GPRS/RF iMODEM specific and optional. It is a GPRS Network APN--Access Point Name.

Isp_Auth_Data: This parameter is Analog iMODEM specific and optional. It may be used if Open ISP authentication is selected, in which case authentication occurs before PPP protocol operations begin. It has the following format: <ID>^<Password>. ID is the ISP Login ID and, Password is the ISP password.

PPP_Auth_Data: This parameter is Analog iMODEM specific and optional. It may be used only if PAP or CHAP authentication are selected, in which case authentication occurs during PPP protocol negotiations. It has the following format: <ID>^<Password>. ID is the Login ID known as the local system name in the CHAP and PAP standards. Password is a password known as the secret in the CHAP and PAP standards. The significance of ID and Password are detailed by RFCs 1334 and 1994.

Example 5.1A: Enable SMTP/IP Connectivity.

<u>USER</u>	<u>iMODEM</u>
ETX_CONNECT[0]: PROFILE=smtp.att \	==>
SERVER=TEXT-smtp.pop.att \	
AUTH_METHOD=ANY \	
AUTH_DATA=me^my_password	

```

<==          POSTED[0]: Operation Started
<==          OK[0]: E-Mail transmission \
              service ready.

```

5.2 ETX_DISCONNECT: Terminate IP/SMTP Connectivity.

If there is no other session with IP connectivity, this command terminates IP connectivity. It terminates the E-Mail transmission session instance support for E-Mail transactions.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
ETX_DISCONNECT[<SID>]
```

PARAMETERS

None

Example 5.2A: Terminate IP/SMTP Connectivity.

<u>USER</u>		<u>iMODEM</u>
ETX_DISCONNECT[0]	<pre> ==> <== <== </pre>	<pre> POSTED[0]: Operation Started OK[0]: E-Mail transmission \ service finished. </pre>

5.3 ETX_SEND_FILE: Send a stored E-Mail message

Support of E-Mail transactions must be enabled with the ETX_CONNECT command in order to use this command. This command makes and releases a TCP connection to an SMTP server.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```

ETX_SEND_FILE[<SID>]: PROFILE=<Prof_Name>
INTERNAL FILE=<File_Name> ATTACH=<Attach_Name>
FROM=<Source_Add> TO=<Dest_Add_List>
SUBJECT=<Subject_Msg>

```

PARAMETERS

Prof_Name: Optional - The name of an SMTP profile such smtp.att. If this parameter is not specified, default SMTP parameters are used.

INTERNAL: This is a keyword parameter and is optional. If it is used, it indicates, the stored E-Mail message has internal headers, and the FROM, TO, and SUBJECT parameters can not be used.

File_Name: Mandatory - The name of a file containing the E-Mail message.

Attach_Name: Optional - The name of a file containing the E-Mail attachment.

Source_Add: Optional - The source E-Mail address. The E-Mail address in the effective SMTP profile is used if this parameter is omitted.

Dest_Add_List: Optional - A list of at least 1 destination E-Mail addresses. The E-Mail addresses in the effective SMTP profile are used if this parameter is omitted. Addresses are separated by the ',' character without spaces and the format is as follows:

<Add-1>,<Add-2>, ...<Add-N>

Add is an E-Mail address and the following examples illustrate its use.

Example: Single E-Mail Address.

TO=dick_tracy@spy.com

Example: Multiple E-Mail Addresses.

TO=john@doe.net,jane@doe.net,boss@top.com

Subject_Msg: Optional - The E-Mail subject. A sentence describing the E-Mail message subject. It must be enclosed in quotes. For example: SUBJECT="This is the subject".

Example 5.3C: Send an E-Mail Message

<u>USER</u>		<u>iMODEM</u>
ETX_SEND_FILE[0]: FILE=test INTERNAL	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: E-Mail successfully sent

5.4 ETX_START_DATA: Start a streaming E-Mail session

Support of E-Mail transactions must be enabled with the ETX_CONNECT command in order to use this command. This command makes a TCP connection to an SMTP server. It also sends the E-Mail headers to the server.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
ETX_START_DATA[<SID>]: PROFILE=<Prof_Name>
FROM=<Source_Add> TO=<Dest_Add_List>
SUBJECT=<Subject_Msg>
```

PARAMETERS

Prof_Name: Optional - The name of an SMTP profile such smtp.att. If this parameter is not specified, default SMTP parameters are used.

Source_Add: Optional - The source E-Mail address. The E-Mail address in the effective SMTP profile is used if this parameter is omitted.

Dest_Add_List: Optional - A list of at least 1 destination E-Mail addresses. The E-Mail addresses in the effective SMTP profile are used if this parameter is omitted. Addresses are separated by the ',' character without spaces and the format is as follows:

<Add-1>,<Add-2>, ...<Add-N>

Add is an E-Mail address and the following examples illustrate its use.

Example: Single E-Mail Address.

Line: A line of ASCII Data.

Example 5.6A: Send a line of E-Mail data.

<u>USER</u>		<u>iMODEM</u>
ETX_SEND_DATA[0]:Dear Jane and Spot,	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Data Sent

6. E-MAIL STORAGE COMMANDS

The commands in this section are for storing, reviewing, and managing E-Mail messages and attachments in files on the iMODEM. They can only be used within an iMODEM session. The files contain lines of ASCII data which is almost completely text.

6.1 Storage Operation Commands

These commands are for storing and reviewing E-Mail messages and attachments.

<u>COMMAND</u>	<u>ABBREVIATED FORM</u>
ES_OPEN_FILE: Open an E-Mail File.	@ESOPEN
ES_OPEN_ATTACH: Open an Attachment File.	@ESATTOPEN
ES_CLOSE_FILE: Close a File.	@ESCLOSE
ES_WRITE_DATA: Write a Line of Data	@ESWDATA
ES_READ_DATA: Read a Line of Data.	@ESRDATA

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

6.1.1 ES_OPEN_FILE: Open an E-Mail message File

This command can be used to open an existing message file for review, or to open and create a new message file. New files can include E-Mail headers or only the E-Mail message body.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
ES_OPEN_FILE[<SID>]: FILE=<File_Name> CREATE VOLATILE
FROM=<Source_Add> TO=<Dest_Add_List>
SUBJECT=<Subject_Msg>
MAX=<Max_Read_Size>
```

PARAMETERS

File_Name: Mandatory - The name of the file where the E-Mail message will be stored, or from which an existing message will be read.

CREATE: This is a keyword parameter. If it is specified, a new file is created. Otherwise, File_Name must correspond to an existing file which will be opened only for reading, and the FROM, TO, and SUBJECT parameters can not be used.

CREATE: This is a keyword parameter. If it is specified, a new file is created. Otherwise, File_Name must correspond to an existing file which will be opened only for reading.

VOLATILE: This is a keyword parameter which can only be used if CREATE is specified. If VOLATILE is specified, a new file will be stored in volatile memory. Otherwise, the new file will be stored in static memory. Files in volatile memory are lost when the iMODEM loses power and those in static memory are preserved.

Max_Read_Size: This parameter is mandatory for files opened for reading. It specifies the maximum size line which can be read with a single ES_READ_FILE command in bytes. It protects the iMODEM user from buffer overflows.

Example 6.1D: Create an attachment in volatile memory.

<u>USER</u>		<u>iMODEM</u>
ES_OPEN_ATTACH[0]: FILE=attachment1 \	==>	
CREATE VOLATILE	<==	POSTED[0]: Operation Started
	<==	OK[0]: E-Mail file open

6.1.3 ES_CLOSE_FILE: Close an E-Mail file

This command is used to close an E-Mail message or attachment file.

SYNTAX

ES_CLOSE_FILE[<SID>]

PARAMETERS

None

Example 6.1E:

<u>USER</u>		<u>iMODEM</u>
ES_CLOSE_FILE[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: E-Mail file closed

6.1.4 ES_WRITE_DATA: Write data to an E-Mail file

This command is used to write a line of ASCII data to a message or attachment file. The file must have been opened with the CREATE option.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

ES_WRITE_DATA[<SID>]:<Line>

PARAMETERS

Line: A line of ASCII data.

Example 6.1F:

<u>USER</u>		<u>iMODEM</u>
-------------	--	---------------


```

ES_WRITE_DATA[0]:Hi Dick:          ==>
                                   <== POSTED[0]: Operation Started
                                   <== OK[0]: Data Stored
    
```

6.1.5 ES_READ_DATA: Read data from an E-Mail file

This command is used to read a line of ASCII data from a message or attachment file. The file must have been opened without the CREATE option.

The iMODEM OK response to this command is followed by the number of bytes in the line--not counting carriage returns etc, and the line of ASCII data. The syntax of that response is as follows:

OK[<SID>] <N_Bytes>:<Line>

N_Bytes is the number of bytes in the line, and Line is the line Of ASCII data.

The iMODEM responds to the command as follows if all the data in the file has been read.

END[<SID>]: E-Mail message retrieved.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

ES_READ_DATA[<SID>]

PARAMETERS

None

Example 6.1G:

<u>USER</u>		<u>iMODEM</u>
ES_READ_DATA[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0] 17:This is an iModem

6.2 Storage Management Commands

These commands are for managing and reviewing E-Mail messages and attachments in files on the iMODEM. They include commands for deleting unneeded files.

<u>Command</u>	<u>Abbreviated Form</u>
ES_LIST_FILE: List E-Mail Files.	@ESLIST
ES_LIST_ATTACH: List attachment Files.	@ESATTLIST
ES_DELETE_FILE: Delete an E-Mail File.	@ESDELETE
ES_DELETE_ATTACH: Delete an Attachment File.	@ESATTDELETE

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

6.2.1 ES_LIST_FILE: List E-Mail message files on the iMODEM

This command can be used to list a single or all message files, and only files in a particular kind or any kind of iMODEM memory. Files can be listed in brief or long format.

The iMODEM response to this command is different than the standard API command response and is as follows:

```
POSTED[<SID>]: <Message>
START[<SID>]:
L[<SID>]: <File_Desc-1>
L[<SID>]: <File_Desc-2>
....
L[<SID>]: <File_Desc-N>
END[<SID>]:
```

File_Desc is the file description and may have the following formats:

Brief Format

File_Name> <Memory_Type>

Long Format

<File_Name> <Memory_Type> <Size> <Date_Info>

File_Name is the file name, Memory_Type is "ST" for static memory and "VO" volatile for volatile memory, Size it the file size in bytes, and Date_Info is the creation date etc.

The following examples illustrate brief and long format listings.

Example: Brief format listing of a file in static memory.

```
L[0]: test ST
```

Example: Long format listing of a file in volatile memory.

```
L[0]: check VO 127 Mon Mar 9 16:02:02 2007
```

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
ES_LIST_FILE[<SID>]: FILE=<File_Name>
FORMAT=<Format_Spec> TYPE=<Memory_Type>
```

PARAMETERS

File_Name: Optional - The name of a specific message file.

Format_Spec: Optional - The format with which files are listed. LONG is the only valid value, and if it is specified, the listing will included extended information such as the date of creation. Otherwise, if this parameter is omitted, the files will be listed in brief format.

Memory_Type: Optional - The type of files to be listed. The values VOLATILE and STATIC, restrict the list to files in volatile and static memory respectively. Otherwise, if this parameter is omitted, all file types are listed.

Example 6.2C: List all files in short format.

<u>USER</u>		<u>iMODEM</u>
ES_LIST_FILE[0]:	==>	
	<==	POSTED[0]: Operation started.
	<==	START[0]:
	<==	L[0]: file1 VO

```
<== L[0]: file2 VO  
<== L[0]: file3 ST  
<== END[0]:
```

6.2.2 ES_LIST_ATTACH: List E-Mail attachment files on the iMODEM

This command can be used to list a single or all attachment files, and only files in a particular kind or any kind of iMODEM memory. Files can be listed in brief or long format.

The iMODEM response to this command is different than the standard API command response and is the same as for the ES_LIST_FILE command, above.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
ES_LIST_ATTACH<SID>]: FILE=<File_Name>
FORMAT=<Format_Spec> TYPE=<Memory_Type>
```

PARAMETERS

File_Name: Optional - The name of a specific message file.

Format_Spec: Optional - The format with which files are listed. LONG is the only valid value, and if it is specified, the listing will include extended information such as the date of creation. Otherwise, if this parameter is omitted, the files will be listed in brief format.

Memory_Type: Optional - The type of files to be listed. The values VOLATILE and STATIC, restrict the list to files in volatile and static memory respectively. Otherwise, if this parameter is omitted, all file types are listed.

Example 6.2D: List all attachments in long format.

<u>USER</u>		<u>iMODEM</u>
ES_LIST_ATTACH[0]: FORMAT=LONG	==>	
	<==	POSTED[0]: Operation started.
	<==	START[0]:
	<==	L[0]: sample VO 256 \
	<==	Tue Jan 1 12:00:00 2007
	<==	L[0]: attach1 VO 44 \
	<==	Wed Jan 2 12:00:00 2007
	<==	L[0]: expense ST 127 \
	<==	Thu Jan 3 12:00:00 2007
	<==	END[0]:

6.2.3 ES_DELETE_FILE: Delete an E-Mail message file from the iMODEM

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
ES_DELETE_FILE[<SID>]: FILE=<File_Name> TYPE=<Memory_Type>
```

PARAMETERS

File_Name: Mandatory - The file name.

Memory_Type: Mandatory - The type of memory where the file is stored. The valid values are as follows:

VOLATILE: Volatile memory.

STATIC: Static memory.

ANY: Either volatile or static memory.

Example 6.2E: Delete a message file.

<u>USER</u>		<u>iMODEM</u>
ES_DELETE_FILE[2]: FILE= \ time TYPE=ANY	==>	
	<==	POSTED[2]: Operation Started
	<==	OK[2]: E-Mail file deleted

6.2.4 ES_DELETE_ATTACH: Delete an E-Mail attachment file from the iMODEM

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

ES_DELETE_ATTACH<SID>: FILE=<File_Name> TYPE=<Memory_Type>

PARAMETERS

File_Name: Mandatory - The file name.

Memory_Type: Mandatory - The type of memory where the file is stored. The valid values are as follows:

VOLATILE: Volatile memory.

STATIC: Static memory.

ANY: Either volatile or static memory.

Example 6.2F: Delete an attachment in static memory.

<u>USER</u>		<u>iMODEM</u>
ES_DELETE_ATTACH[2]: FILE= \ at1 TYPE=STATIC	==>	
	<==	POSTED[2]: Operation Started
	<==	OK[2]: E-Mail file deleted

7. ISOCKET API COMMANDS

These commands provide the same functionality as the analogous BSD sockets calls, and can only be used within a session.

Command

IS_IPCONNECT: Establish IP Connectivity.
 IS_IPRELEASE: Terminate IP Connectivity.
 IS_SOCKET: Allocate a Socket.
 IS_BIND: Bind to an address.
 IS_CONNECT: Make a TCP Connection.
 IS_CLOSE: Release a Socket and TCP Connection
 IS_RECV: Receive data.
 IS_SEND: Send data.
 IS_GETHOSTBYNAME: Perform DNS resolution
 IS_GETHOSTNAME: Get the local host name.
 IS_GETSOCKNAME: Get local host address data.

Abbreviated Form

@ISIPCONN
 @ISIPREL
 @ISOCKET
 @ISBIND
 @ISCONN
 @ISCLOSE
 @ISRXD
 @ISTXD
 @ISIPADD
 @ISHOST
 @ISHOSTIP

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

Figure 7A illustrates the protocols used during an ISOCKET session.

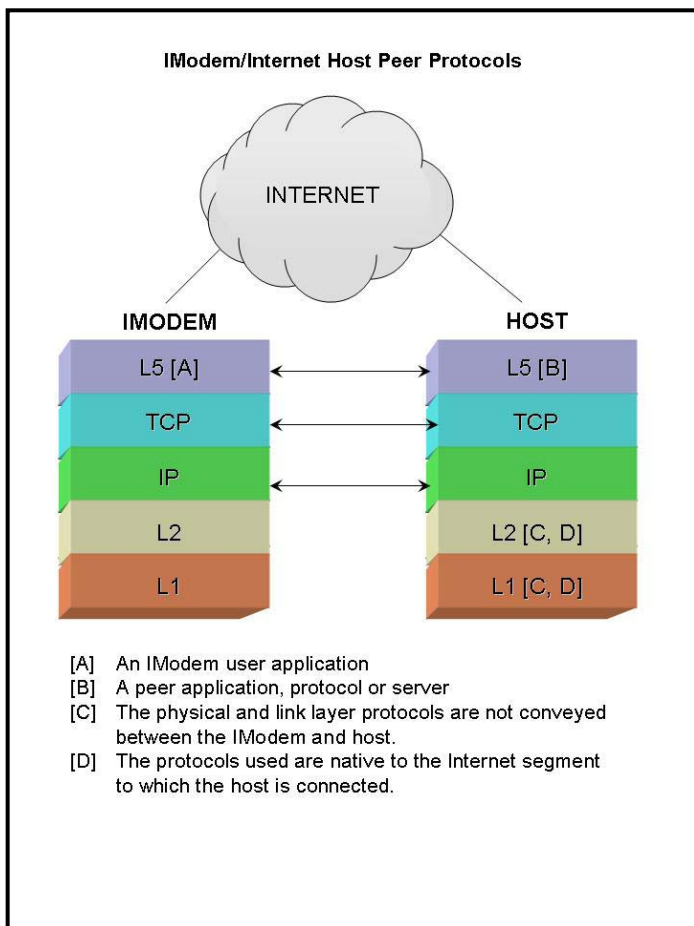


Figure 7A

7.1 IS_IPCONNECT: Establish IP Connectivity

This command establishes IP connectivity if no other session has already done so.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

Analog iMODEM Syntax

```
IS_IPCONNECT[<SID>]: PROFILE=<Prof_Name>
PHONE=<Phone_No>
OPENLOG=<Isp_Auth_Data> LINKLOG=<PPP_Auth_Data>
```

GPRS/RF iMODEM Syntax

```
IS_IPCONNECT[<SID>]: PROFILE=<Prof_Name>
```

PARAMETERS

Prof_Name: Optional - The name of a Link Profile such link.lmi. If this parameter is not specified, default Link Profile parameters are used.

Phone_No: This parameter is Analog iMODEM specific and optional. It is an ISP local access telephone number.

Isp_Auth_Data: This parameter is Analog iMODEM specific and optional. It may be used if Open ISP authentication is selected, in which case authentication occurs before PPP protocol operations begin. It has the following format: <ID>^<Password>. ID is the ISP Login ID and, Password is the ISP password.

PPP_Auth_Data: This parameter is Analog iMODEM specific and optional. It may be used only if PAP or CHAP authentication are selected, in which case authentication occurs during PPP protocol negotiations. It has the following format: <ID>^<Password>. ID is the Login ID known as the local system name in the CHAP and PAP standards. Password is a password known as the secret in the CHAP and PAP standards. The significance of ID and Password are detailed by RFCs 1334 and 1994.

Example 7.1A: Establish IP connectivity.

<u>USER</u>		<u>iMODEM</u>
IS_IPCONNECT[0]: PROFILE=link.att	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: IP Link is up

7.2 IS_IPRELEASE: Terminate IP Connectivity

This command terminates IP connectivity if no other session has IP connectivity. It can only be used in a session with IP connectivity.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
IS_IPRELEASE[<SID>]
```

PARAMETERS

None

Example 7.2A: Terminate IP connectivity.

<u>USER</u>		<u>iMODEM</u>
IS_IPRELEASE[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: IP Link is down

7.3 IS_SOCKET: Allocate a Socket

This command allocates a socket and associates it with an ISOCKET session.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

IS_SOCKET[<SID>]: RXMAX=<Rx_Max_Ms> TXMAX=<Tx_Max_Ms>

PARAMETERS

Rx_Max_Ms: Mandatory - The maximum period, in milliseconds, for an attempt to receive data with IS_RXDATA command to complete.

Tx_Max_Ms: Mandatory - The maximum period, in milliseconds, for an attempt to transmit data with IS_TXDATA command to complete.

Example 7.3A: Allocate a socket.

<u>USER</u>		<u>iMODEM</u>
IS_SOCKET[0]: rxmax=3000 txmax=3000	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Socket Allocated

7.4 IS_BIND: Bind to an address.

This command binds a local IP address and port number to a socket. It can only be used in a session that has allocated a socket with an IS_SOCKET command.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

IS_BIND[<SID>]: PORT=<Tcp_Port> ADDRESS=<IP_Address>

PARAMETERS

Tcp_Port: A TCP port number.

IP_Address: An IPV4 format numeric address.

Example 7.4A:

<u>USER</u>		<u>iMODEM</u>
IS_BIND[1]: PORT=32 ADD=66.23.100.18	==>	
	<==	POSTED[1]: Operation started.
	<==	OK[1]: Local data bound.

7.5 IS_CONNECT: Make a TCP Connection

This command makes a TCP connection to another host. It can only be used in a session that has allocated a socket with the IS_SOCKET command and has IP connectivity.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
IS_CONNECT[<SID>]: PROFILE=<Prof_Name>
SERVER=<Host> PORT=<Tcp_Port_No>
```

PARAMETERS

Prof_Name: Optional - The name of a Link Profile such link.lmi. If this parameter is omitted, default Link Profile parameters are used.

Host: Optional - The name or address of another Internet host. It may have the following two formats:

- Text Format: SERVER=TEXT-<Host> where Host is the name of a host such as cermetek.com.
- IP4 Format: SERVER=IPV4-<ip_address> where ip_address is the IPV4 format host Internet address such as 66.117.140.246.

Tcp_Port: Optional - A TCP port number.

Example 7.5A: Connect to an Internet Host

<u>USER</u>		<u>iMODEM</u>
IS_CONNECT[0]: SERVER=TEXT \ -time-a.nist.gov port=13	==> <== <==	POSTED[0]: Operation ... OK[0]: TCP socket connected

7.6 IS_CLOSE: Release a Socket and TCP Connection

This command terminates a TCP connection and releases a socket. It can only be used in a session with TCP and IP connectivity.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
IS_CLOSE[<SID>]
```

PARAMETERS

None

Example 7.6A:

<u>USER</u>		<u>iMODEM</u>
IS_CLOSE[0]	==> <== <==	POSTED[0]: Operation Started OK[0]: TCP socket released

7.7 IS_RECV: Receive data

This command is used to receive data over a TCP connection. It can only be used in a session with TCP and IP connectivity.

The iMODEM delivers received data in the following format:

```
OK[<SID>]: <N_Bytes>:<Data>
```

N_Bytes: The number of bytes received.

Data: N_Bytes of binary data.

A timeout will occur if no data is received with the maximum time configured with IS_SOCKET command RXMAX parameter. In that case the iMODEM will respond with the following error: message:

```
ERROR[<SID>] 87 : Socket Data operation timeout
```

This is not necessarily an error and often indicates no data has arrived from the peer host.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

```
IS_RECV[<SID>]: MAX=<Max_Bytes>
```

PARAMETERS

Max_Bytes: The maximum number of bytes to receive.

Example 7.7A: Receive a maximum of 256 bytes.

<u>USER</u>		<u>iMODEM</u>
IS_RECV[1]: max=256	==>	
	<==	POSTED[1]: Operation Started
	<==	OK[1] 6:abcdef

7.8 IS_SEND: Send data

This command is used to send data over a TCP connection. It can only be used in a session with TCP and IP connectivity.

The iMODEM responds with a message of the following format if data is successfully sent.

OK[<SID>]: <N_Bytes>

N_Bytes is the number of data bytes actually sent.

A timeout will occur if no data can not be transmitted in the time configured with IS_SOCKET command TXMAX parameter. In that case the iMODEM will respond with the following error: message:

ERROR[<SID>] 87 : Socket Data operation timeout

This is not necessarily an error and may occur due to network congestion or a resource shortage.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

IS_SEND[<SID>] <N_Bytes>:<Data>

PARAMETERS

N_Bytes: The number of bytes of data to send.

Data: N_Bytes of binary data.

Example 7.8A: Send 9 bytes of data.

<u>USER</u>		<u>iMODEM</u>
IS_SEND[0]9:junk.junk	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: 9

7.9 IS_GETHOSTBYNAME: Perform DNS resolution

This command queries the Internet for the IP address of a host. It can only be used in a session with IP connectivity.

The iMODEM responds with a message of the following format if this command succeeds:

OK[<SID>]: <IP_Address>

IP_Address is an IPV4 format address such 64.211.19.28.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

IS_GETHOSTBYTNAME[<SID>]: NAME=<Host_Name>

PARAMETERS

Host_Name: The symbolic name of an Internet host.

Example 7.9A: Query for a host's IP address.

<u>USER</u>		<u>iMODEM</u>
IS_GetHostByName[2]: NAME= \	==>	
cermetek.com	<==	POSTED[2]: Operation Started
	<==	OK[2]: 64.227.162.69

7.10 IS_GETHOSTNAME: Get the local host name

This command gets the symbolic name of the iMODEM. If it succeeds, the iMODEM response is of the following format:

OK[<SID>]: <Host_Name>

Host_Name is the symbolic name of the iMODEM.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

IS_GETHOSTNAME[<SID>]

Example 7.10A: Get the local iMODEM name.

<u>USER</u>		<u>iMODEM</u>
IS_GETHOSTNAME[0]	==>	
	<==	POSTED[0]: Operation started
	<==	OK[0]: node101@imodem.net

7.11 IS_GETSOCKNAME: Get local host address data

This command gets the local IP address and TCP port number associated with a socket. It can only be used in a session that has allocated a socket with the IS_SOCKET command. If it succeeds, the iMODEM response has the following format:

OK[<SID>]: <Tcp_Port>, <IP_Address>

TCP_Port is the TCP port number and IP_Address is an IPV4 format address.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

IS_GETSOCKNAME[<SID>]

PARAMETERS

None

Example 7.11A: Get local IP address information.

<u>USER</u>		<u>iMODEM</u>
IS_GET_SOCKETNAME[1]	==> <== <==	POSTED[1]: Operation Started. OK[1]: 56, 65.127.99.1

8. E-MAIL RETRIEVAL COMMANDS (POP3)

These commands are used to retrieve E-Mail messages from a POP3 server, and can only be used within an iMODEM session.

<u>COMMAND</u>	<u>ABBREVIATED FORM</u>
ERX_CONNECT: Establish IP/POP3 Connectivity.	@ERXCONN
ERX_DISCONNECT: Terminate IP/POP3 Connectivity.	@ERXDISC
ERX_MAIL_MSG: Retrieve a message.	@ERXMAIL
ERX_MSG_DATA: Receive a line of data.	@ERXDATA
ERX_LIST_MSGS: List all messages.	@ERXLIST

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

8.1 ERX_CONNECT: Establish IP/POP3 Connectivity

This command establishes a TCP connection with a POP3 server, and establishes IP connectivity if it is not already established. It also authenticates with a POP3 server and performs the appropriate handshake. The command starts a POP3 session.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

Analog iMODEM Syntax

```
ERX_CONNECT[<SID>]: PROFILE=<Prof_Name>
SERVER=<Pop3_Server> PORT=<Tcp_Port>
SERVLOG=<Pop3_Auth_Data>
DIALUP=<Link_Pars> PHONE=<Phone_No>
OPENLOG=<Isp_Auth_Data> LINKLOG=<PPP_Auth_Data>
```

GPRS/RF iMODEM Syntax

```
ERX_CONNECT[<SID>]: PROFILE=<Prof_Name>
SERVER=<Pop3_Server> PORT=<Tcp_Port>
SERVLOG=<Pop3_Auth_Data>
DIALUP=<Link_Pars> APN=<APN_Name>
```

PARAMETERS

Prof_Name: Optional - The name of an POP3 profile such pop.att. If this parameter is not specified, default POP3 parameters are used.

Pop3_Server: Optional - A POP3 server name or IP address. It may have the following two formats:

- Text Format: SERVER=TEXT-<Server_Name> where Server_Name is the name of server such as pop.earthlink.net.
- IP4 Format: SERVER=IPV4-<ip_address> where ip_address is the IPV4 format server Internet address such as 66.117.140.246.

Tcp_Port: Optional - A TCP port number.

Pop3_Auth_Data: Optional - The POP3 server authentication data. It has the following format: <ID>^<Password>. ID is the POP3 Login ID, and Password is the POP3 password.

Link_Pars: This optional parameter must be set to TRUE if any of the following parameters are used: PHONE, APN, OPENLOG, LINKLOG. Otherwise it must be omitted or set to FALSE.

Phone_No: This parameter is Analog iMODEM specific and optional. It is an ISP local access telephone number.

APN_Name: This parameter is GPRS/RF iMODEM specific and optional. It is a GPRS Network APN--Access Point Name.

Isp_Auth_Data: This parameter is Analog iMODEM specific and optional. It may be used if Open ISP authentication is selected, in which case authentication occurs before PPP protocol operations begin. It has the following format: <ID>^<Password>. ID is the ISP Login ID and, Password is the ISP password.

PPP_Auth_Data: This parameter is Analog iMODEM specific and optional. It may be used only if PAP or CHAP authentication are selected, in which case authentication occurs during PPP protocol negotiations. It has the following format: <ID>^<Password>. ID is the Login ID known as the local system name in the CHAP and PAP standards. Password is a password known as the secret in the CHAP and PAP standards. The significance of ID and Password are detailed by RFCs 1334 and 1994.

Example 8.1A: Connect to a POP3 Server.

<u>USER</u>		<u>iMODEM</u>
ERX_CONNECT[0]: PROFILE= \	==>	
pop3.earthlink	<==	POSTED[0]: Operation Started
	<==	OK[0]: Connected to E-Mail \
		Retrieval Server

8.2 ERX_DISCONNECT: Terminate IP/POP3 Connectivity

This command terminates a TCP connection with a POP3 server, and terminates IP connectivity if it is not established by another session. It also terminates a POP3 session.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

ERX_DISCONNECT[<SID>]

PARAMETERS

None

Example 8.2A: Disconnect from a POP3 server.

<u>USER</u>		<u>iMODEM</u>
ERX_DISCONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Disconnected from E-Mail \ Retrieval Server

8.3 ERX_MAIL_MSG: Retrieve a message

This command begins retrieval of a specific message from a POP3 server. If it succeeds, then the iMODEM user can start receiving lines of E-Mail data.

The command may fail if there is no error but the specified message is not stored on the POP3 server. In that case the iMODEM response will be as follows:

ERROR 28: Bad message number.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

ERX_MAIL_MSG[<SID>]: MSG=<Msg_No> MAX=<Max_Bytes>

PARAMETERS

Msg_No: Mandatory - The number of a message stored on the POP server which must be greater than 0.

Max_Bytes: Mandatory - The maximum line size--in bytes, that can be received with a single ERX_MSG_DATA API command. This protects the user from buffer overflows.

Example 8.3A: Begin retrieval of a message from a POP3 server.

<u>USER</u>		<u>iMODEM</u>
ERX_MAIL_MSG[0]: MSG=1 MAX=256	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Ready to retrieve E-Mail Data

8.4 ERX_MSG_DATA: Receive a line of data

The command is used to receive a line of data from a POP3 server. The iMODEM OK response to this command is followed by the number of bytes in the line--not counting carriage returns etc, and the line of ASCII data. The syntax of that response is as follows:

OK[<SID>] <N_Bytes>:<Line>

N_Bytes is the number of bytes in the line, and Line is the line of ASCII data.

The iMODEM responds as follows when the complete E-Mail message has been received.

END[0]: E-Mail Message retrieved.

IMPORTANT NOTE

The ERX_MAIL_MSG command must have succeeded in the same session as this command in order to use this command.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

ERX_MSG_DATA[<SID>]

PARAMETERS

None

Example 8.4A:

<u>USER</u>		<u>iMODEM</u>
ERX_MSG_DATA[0]	==>	
	<==	
	<==	POSTED[0]: Operation Started OK[0] 35:Return-Path: \ <prtest@earthlink.net>

8.5 ERX_LIST_MSGS: List all messages

This command lists all messages on a POP3 server. Messages can be listed in brief or long format.

The iMODEM response to this command is different than the standard API command response and is as follows:

```
POSTED[<SID>]: <Message>
START[<SID>]:
L[<SID>]: <Msg_Desc-1>
L[<SID>]: <Msg_Desc-2>
....
L[<SID>]: <Msg_Desc-N>
END[<SID>]:
```

Msg_Desc is the message description and has the following format:

<Msg_No> <Msg_Size>

Msg_no: The message number.

Msg_Size: The size of the message in bytes.

The following example illustrates the message listing.

Example

```
L[0]: 8 1315
```

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID--SID.

SYNTAX

ERX_LIST_MSGS[<SID>]:

PARAMETERS

None

Example 8.5B: List all messages.

<u>USER</u>		<u>iMODEM</u>
ERX_LIST_MSGS[0]:	==>	
	<==	POSTED[0]: Operation started
	<==	START[0]:
	<==	L[0]: 1 128
	<==	L[0]: 2 47
	<==	L[0]: 3 300
	<==	END[0]:

9. CONFIGURATION COMMANDS

These commands are used to configure Profile parameters that commonly require reconfiguration. They are used when the iMODEM is not in Multi-mode.

<u>COMMAND</u>	<u>ABBREVIATED FORM</u>
LINK_CONFIG: Configure the Link Profile.	@LINKC
ETX_CONFIG: Configure the SMTP Profile.	@ETXC
ERX_CONFIG: Configure the POP3 Profile.	@ERXC

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

9.1 LINK_CONFIG: Configure the Link Profile

This command is used to reconfigure Link Profile parameters.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

Analog iMODEM Syntax

```
LINK_CONFIG: PROFILE=<Prof_Name> LEVEL=<Prof_Level>
PHONE=<Phone_No>
RAW_ID=<Open_Login_Id> RAW_PWD=<Open_Login_Password>
PPP_AUTH_METHOD=<PPP_Auth_Proto>
PPP_ID=<PPP_Login_ID> PPP_SECRET=<PPP_Secret>
```

GPRS/RF iMODEM Syntax

```
LINK_CONFIG: PROFILE=<Prof_Name> LEVEL=<Prof_Level>
APN=<APN_Name>
```

PARAMETERS

Prof_Name: Mandatory - The name of a Link Profile, or an Application Profile which references a Link Profile.

Prof_Level: Mandatory - The kind of profile referenced by Prof_Name. The value LINK indicates it references a Link Profile--such as link.kore, and the value APP indicates it references an Application Profile--such as smtp.att.

Phone_No: This parameter is Analog iMODEM specific and optional. It is an ISP local access telephone number.

APN_Name: This parameter is GPRS/RF iMODEM specific and optional. It is a GPRS Network APN--Access Point Name.

Open_Login_Id: This parameter is Analog iMODEM specific and optional. It is the unencrypted ASCII Login ID sent to the ISP during Open authentication. Open authentication occurs between the time a telephone connection is established and PPP protocol operations begin.

Open_Login_Password: This parameter is Analog iMODEM specific and optional. It is the unencrypted ASCII password sent to the ISP during Open authentication. Open authentication occurs between the time a telephone connection is established and PPP protocol operations begin.

PPP_Auth_Proto: This parameter is Analog iMODEM specific and optional. It configures the authentication protocol used by the iMODEM and ISP during PPP protocol negotiations. The valid values are as follows:

CHAP: Use the CHAP protocol.

PAP: Use the PAP protocol.

NONE: Do not perform authentication during PPP negotiations.

PPP_Login_ID: This parameter is Analog iMODEM specific and optional. It is the CHAP or PAP Login ID. For PAP this is the User ID expected by the ISP, and for CHAP this is your "name" per RFCs 1334 and 1994. It can only be used if PAP or CHAP authentication is selected by the appropriate PPP_Auth_Proto setting.

PPP_Secret: This parameter is Analog iMODEM specific and optional. It is the CHAP or PAP Secret. For PAP this normally the password expected by the ISP, and for CHAP this is the "secret" per RFCs 1334 & 1994. It can only be used if PAP or CHAP authentication is selected by the appropriate PPP_Auth_Proto setting.

Example 9.1A: Configure the APN of the Link Profile referenced by smtp.earthlink

<u>USER</u>		<u>iMODEM</u>
LINK_CONFIG: level=APP PROFILE \ =smtp.earthlink APN=c1.korem2m.com	==>	
	<==	POSTED: Operation ...
	<==	OK: Configuration OK

9.2 ETX_CONFIG: Configure the SMTP Profile

This command is used to reconfigure SMTP profile parameters.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

```
ETX_CONFIG: PROFILE=<Prof_Name>
SERVER=<Smtplib_Server>
FROM=<Source_Add> TO=<Dest_Add_List>
AUTH_METHOD=<Smtplib_Auth_Proto>
LOGIN_ID=<Login_ID> PASSWORD=<Password>
```

PARAMETERS

Prof_Name: Mandatory - The name of an SMTP Profile.

Smtplib_Server: Optional - An SMTP server name or IP address. It may have the following two formats:

- Text Format: SERVER=TEXT-<Server_Name> where Server_Name is the name of server such as smtp.earthlink.net.
- IP4 Format: SERVER=IPV4-<ip_address> where ip_address is the IPV4 format server Internet address such as 66.117.140.246.

Source_Add: Optional - The source E-Mail address.

Dest_Add_List: Optional - A list of at least 1 destination E-Mail addresses. Addresses are separated by the ',' character without spaces and the format is as follows:

<Add-1>,<Add-2>, ...<Add-N>

Add is an E-Mail address and the following examples illustrate its use.

Example: Single E-Mail Address.

TO=dick_tracy@spy.com

Example: Multiple E-Mail Addresses.

TO=john@doe.net,jane@doe.net,boss@top.com

Smtplib_Proto: Optional. Many SMTP servers require authentication. The parameter specifies what SMTP authentication protocol will be used with such servers. The valid values are as follows:

- CRAM-MD5: Authentication data is encrypted, and exchanged via the challenge mechanism described by RFCs 4422 and 1321.
- PLAIN: Authentication data is sent as base 64 encrypted strings. [A]
- LOGIN: Authentication data is sent as unencrypted ASCII strings. [A]
- ANY: The iMODEM and SMTP server automatically negotiate the best protocol which is recommended.
- NONE The iMODEM does not perform authentication.

[A]: Not generally considered secure.

Login_Id: Optional - The SMTP Login ID which is only used if Smtplib_Method is not set to NONE.

Password: Optional - The SMTP password which is only used if Smtplib_Method is not set to NONE.

Example 9.2C: Configure SMTP Profile E-Mail addresses.

<u>USER</u>		<u>iMODEM</u>
ETX_CONFIG: PROFILE=smtplib.cerz \	==>	
SERVER=TEXT-smtplib.main.cz\		
FROM=dick@best.net		
TO=jane@att.net,spot@dog.net		
	<==	POSTED: Operation Started
	<==	OK: Configuration OK

9.3 ERX_CONFIG: Configure the POP3 Profile

This command is used to reconfigure the POP3 Profile parameters.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

```
ERX_CONFIG: PROFILE=<Prof_Name>
SERVER=<Pop3_Server>
LOGIN_ID=<Login_ID> PASSWORD=<Password>
```

PARAMETERS

Prof_Name: Mandatory - A POP3 Profile name.

Pop3_Server: Optional - An POP3 server name or IP address. It may have the following two formats:

- Text Format: SERVER=TEXT-<Server_Name> where Server_Name is the name of server such as pop.earthlink.net.
- IP4 Format: SERVER=IPV4-<ip_address> where ip_address is the IPV4 format server Internet address such as 66.117.140.246.

Login_Id: Optional - The POP3 Login ID.

Password: Optional - The POP3 password.

Example: Configure POP3 parameters.

Example 9.3A: Configure POP3 parameters.

<u>USER</u>		<u>iMODEM</u>
ERX_CONFIG: PROFILE=pop3.cerz \ SERVER=TEXT-pop3.main.cz \ ID=dick PWD=tracy's_secret	==>	
	<==	POSTED: Operation Started
	<==	OK: Configuration OK

10. UTILITY COMMANDS

These commands make using the iMODEM convenient and are used when the iMODEM is not in Multi-mode.

<u>COMMAND</u>	<u>ABBREVIATED FORM</u>
@SHOW: Display Profile Settings. IM_SET_MODEM: Set device attributes.	@IMSET

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

10.1 @SHOW: Display Profile Settings

This command displays profile parameter values.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

@SHOW: PROFILE=<Prof_Name> TYPE=<Prof_Type> LEVEL=<Prof_Level>

PARAMETERS

Prof_Name: The name of a standard Application Profile. If this parameter is specified, and the TYPE parameter must be omitted.

Prof_Type: The type of a standard Application Profile. The valid values are TX_MAIL for an SMTP Profile, ISOCKET for an ISOCKET Profile, and RX_MAIL for a POP3 Profile. If this parameter is specified, the parameters of the appropriate Application Profile are displayed, and the PROFILE parameter must be omitted.

Prof_Level: Optional - Specified what level of information is displayed. The valid values are as follows:

ALL: Display all profile information.
TOP: Display application specific information.
TCP: Display TCP/IP protocol information.
LINK: Display Link Profile information.

IMPORTANT NOTES

The default is ALL.

Example 10.1A: Display all SMTP Profile parameters.

<u>USER</u>		<u>iMODEM</u>
@SHOW: PROFILE=smtp.via LEVEL=ALL	==>	
	<==	***** \

		AUTH PROFILE = auth.smtp.via
		TCP PROFILE = tcp.smtp.via
		IP PROFILE = ip.dialup_default
		LINK PROFILE = link.via

SERVER = smtp.viawest.net
TCP PORT = 587

SMTP Specific Data

.....
SRC ADD = johnd@viawest.net
DEST[0] = johnd@viawest.net
DEST[1] = paulr@rcom-software.com
DEST[2] = hroskos@cermetek.com
.....

SMTP Authentication Data

.....
SMTP Account Profile = acct.smtp.via
AUTHENTICATION METHOD = ANY
LOGIN ID = johnd
PASSWORD = my_secret
DOMAIN = viawest.net

LINK Profile Data

MEDIA = PSTN
ISP LINK PROFILE = isp_link.via
ISP ACCOUNT PROFILE = \
isp_acct.via
.....

ISP Account Profile Data

.....
PPP AUTH METHOD = PAP
LOCAL NAME/LOGIN ID = johnd
SECRET/PASSWORD = my_secret
REMOTE AUTHENTICATOR = *
.....

ISP Link Profile Data

.....
PEER FILE = isp.via
PHONE = 1-408-847-4099
SPEED = 38400
MAX DIAL TIME = 180
CONNECT RESPONSE = \
CONNECT
ISP GREETING = Sign-on:
LOGIN PROMPT =
PASSWORD PROMPT =
PPP CONNECT DELAY = -1
PPP MRU = -1
PPP MTU = 1500
LCP ECHO INTERVAL = 30
LCP ECHO FAIL MAX = 4

PPP MAGIC NO = 1
PPP ASYNC MAP = 00000000
PPP IPV4 NET MASK = FFFFFFFF
***** \

.
OK: Display Complete

10.2 IM_SET_MODEM: Set device attributes.

This command is used to configure iMODEM device attributes. The configuration is preserved when the iMODEM loses power.

IMPORTANT NOTES

All parameters are optional and if the command is entered without arguments, their values plus the iMODEM serial number are displayed.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

Analog iMODEM Syntax

```
IM_SET_MODEM: USER=<Modem_ID> SPEED=<UI_Speed>
AUTO=<Startup_Mode> PRESERVE=<Save_Config>
LOGIN=<Login_ID> PASSWORD=<Password>
```

GPRS/RF iMODEM Syntax

```
IM_SET_MODEM: USER=<Modem_ID> SPEED=<UI_Speed>
```

PARAMETERS

Modem_ID: Optional - The iMODEM device user ID.

UI_Speed: Optional - The baud rate of iMODEM serial port through which the host connects. The default is rate 19200 baud and the iMODEM must be re-powered before a new baud rate becomes effective.

Startup_Mode: Optional - A Analog iMODEM specific parameter. Configures the Analog iMODEM mode upon iMODEM power-up. If the value is TRUE, the Analog iMODEM comes up in Modem mode. Otherwise, if the value is FALSE, it comes up in Internet mode. Internet mode is the default.

Save_Config: Optional - A Analog iMODEM specific parameter. Configures whether analog modem parameters--such as the S0 ring indication parameter, are reinitialized when the Analog iMODEM transitions from Internet to Modem mode. If the value is TRUE, the Analog iMODEM will not reinitialize the parameters. Otherwise, if the value is FALSE, they will be reinitialized. Reinitializing the modem parameters is the default.

Login_ID: Optional - A Analog iMODEM specific parameter. The incoming caller Login ID. If this parameter is set, the remote caller must supply a Login ID after making a modem connection with the Analog iMODEM. A value of 0 means no Login ID is required and is the default.

Password: Optional - A Analog iMODEM specific parameter. The incoming caller Password. If this parameter is set, the remote caller must supply a Password after making a modem connection with the Analog iMODEM. A value of 0 means no Password is required and is the default.

Example 10.2A:

<u>USER</u>		<u>iMODEM</u>
IM_SET_MODEM: USER=userA1	==>	
	<==	POSTED: Operation Started
	<==	OK: Configuration OK

11. ANALOG iMODEM SPECIFIC COMMANDS

11.1 Analog iMODEM Proprietary Commands

The user may find the following proprietary commands useful.

11.1.1 @SEND: Send stored Data.

The @SEND command is used to send a file to a remote system in Modem mode. It may be invoked locally in command mode or remotely in data mode.

SYNTAX

```
@SEND -V -S -M <FileName>
```

The -V and -S flags are optional. -V means send a file in volatile memory and -S means send a file in static memory--Flash. The FileName may be up to 64 contiguous ASCII characters and may not contain white space, '*' or '/' characters.

The -M flag is optional. It causes the line termination character to be mapped to <CR><LF> when a line of data is transmitted to a remote system. It should only be with ASCII files.

When initiated locally the iModem generates an OK if the transmission is successful and ERROR otherwise. Also, the iModem transitions to data mode during file transmission and remains in that mode after transmission is complete.

Sending a file to a remote system.

The Analog iMODEM must be in MODEM command mode in order to do this. The Analog iMODEM remains in MODEM data mode after file transmission is complete.

Example 11.1A:

LOCAL SYSTEM		LOCAL MODEM		REMOTE SYSTEM
atd 339-2348	==>			
			<==	CONNECT 38400
Joe	==>		<==	Login:
Blow	==>		<==	Password:
+++	==>		<==	Welcome REMOTE-A1
@send test	==>	OK		
			==>	Test Data
		<==		OK

Remote Transactions in MODEM Mode.**Example 11.1B:** Typical Login

LOCAL SYSTEM		LOCAL MODEM		REMOTE SYSTEM
			<==	atd 339-2348
		CONNECT 57600	==>	
Login:	==>		<==	Joe
Password:	==>		<==	Blow
Options:	==>		<==	
Welcome - iMODEM READY	==>		==>	

Example 11.1C: Retrieving a file.

LOCAL SYSTEM		LOCAL MODEM		REMOTE SYSTEM
			<==	atd 339-2348
		CONNECT 57600	==>	
Login:	==>		<==	Joe
Password:	==>		<==	Blow
Options:	==>		<==	@send fname
Welcome - iMODEM READY	==>		==>	
START: fname Size = 6	==>		==>	
Hello	==>		==>	
<EOF>	==>		==>	
END: fname Size = 6 Sent = 6	==>		==>	

11.2 Single Line Analog iMODEM Modem Commands

The following Analog iMODEM commands (and associated parameters) MUST be executed singly (i.e., only one command followed by a <CR> per line). All other supported AT commands can be concatenated.

<u>COMMAND</u>	<u>FUNCTION/DESCRIPTION</u>
Single Line Execution Commands	
ATA	Go off-hook and attempt to answer call
ATD	Dial modifier
ATO	Go back on-line
ATH	Initiate a hang-up sequence
ATV	Result Code display control
ATE	Echo control

<u>COMMAND</u>	<u>FUNCTION/DESCRIPTION</u>
Single Line Execution Commands (continued)	
ATS0	Select/Display/Set S Register 0
ATS2	Select/Display/Set t S Register 2
ATS3	Select/Display/Set S Register 3
ATS4	Select/Display/Set S Register 4
ATS5	Select/Display/Set S Register 5
ATS7	Select/Display/Set S Register 7
ATS10	Select/Display/Set S Register 10
ATS12	Select/Display/Set S Register 12

IMPORTANT NOTE

The Analog AT commands listed in Section 11.3 must be executed singly (i.e., one command followed by the carriage return <CR>) per line.

11.3 Standard Analog iMODEM Modem Commands

The Analog iMODEM product family supports the Hayes compatible modem commands defined and specified in TIA/EIA-602-A, ITU-T V.250 and TIA/EIA-602-A. For brevity, the supported commands are listed below in alphanumeric order. For a more complete discussion of these commands, refer to Cermetek specification # 615-0001, AT Commands and S-Registers.

<u>COMMAND</u>	<u>FUNCTION/DESCRIPTION</u>
Basic Commands	
A/	Re-execute command
A	Go off-hook and attempt to answer call
B0	Select CCITT connection at 300 or 1200bps
B1	Sect Bell connection at 300 or 1200bps
Dn	Dial modifier
P	Pulse Dial
T	Touch Tone Dial
W	Wait for Dial Tone
;	Return to Command State
@	Wait for Quiet Answer Command
!	Flash Hook
,	Pause
0-9/ABCD	Dial Digits/Characters & wait for credit card dial tone
^	Toggles calling tone
L	Redial last number
*,#	Star digit – tone dialing
S=n	Dial the number and store in Directory, n= 0 to 3
E0	Disable command echo

E1	Enable command echo
H, H0	Initiate a hang-up sequence
H1	If on-hook, go off-hook and enter command mode
L0	Set low speaker volume
L1	Set low speaker volume
L2	Set medium speaker volume
L3	Set high speaker volume
M0	Speaker off
M1	Speaker on during hand shaking and speaker off while receiving carrier
M2	Speaker on during hand shaking and while receiving carrier
M3	Speaker off during dialing and receiving carrier and speaker on during answering
O0	Go on-line
O1	Go on-line and initiate a retrain sequence
Sn	Select S Register n
Sn=m	Write value m to S Register n
Sn?	Read the value of S Register n
?	Returns Last addressed S Register value
V0	Short Form Result Code Option
V1	Long Form Result Code Option
W0	Reports DTE Speed
W1	Reports DCE Speed and Error Correction
W2	Reports DCE Speed Only
X1, X2, X3, X4	Result Code Reporting Options
+++	Escape Code Sequence
Z0, Z1	The modem performs soft reset. Restore stored profile 0 or 1 after reset

Enhanced Commands

&C0	Force DCD active regardless of the carrier state
&C1	DCD to follow the carrier state
&D0	Modem ignores DTR
&D1	Modem assumes command state

COMMAND**FUNCTION/DESCRIPTION****Enhanced Commands (Continued)**

&D2	Modem hangs up, assumes command state and disables auto-answer
&D3	Modem assumes initialization state
&F0	Restore Factory configuration 0
&F1	Restore Factory configuration 1
&G0, 1, 2	Guard Tone Options
&K0	Disable DTE/DCE flow control
&K1,2	Not Supported
&K3	Enable RTS/CTS DTE/DCE flow control
&K4	Enable XON/XOFF DTE/DCE flow control
&K5	Enable transparent XON/XOFF flow control
&P0, 1, 2, 3	Make to Break Ratio Options
&Q0	Select direct asynchronous mode
&Q1,2,3,4	Not Supported
&Q5	Modem negotiates an error corrected link
&R0	CTS tracks RTS
&R1	CTS always on
&S0	DSR always on
&S1	DSR active after answer tone detected and inactive after loss of carrier
&T0	Terminate Loopback Test
&T1	Initiate Analog Loopback Test
\Bn	Send break of n x 100 msec (n=0 to 9)
\N0	Select normal speed buffered mode
\N1	Select direct mode
\N2	Select reliable mode
\N3	Select auto reliable mode
\N4	Force LAPM mode
\N5	Force MNP mode
\V0	Connect messages are controlled by the command settings X, W, and S95
\V1	Connect messages are displayed in the single line format
Modulation Commands	
+MS	Select modulation
+MCR=n	Carrier protocol n reporting
+MR=n	Select extended Modulation Reporting of +MCR and +MRR to DTE
+MRR=m, n	Report Transmit m and Receive n DCE rates to DTE

V.92 Commands

+PCW	Enable/Disable Call Waiting Detection
+PCM	Enable/Disable PCM Upstream
+PIG	PCM Upstream Ignore
+PMH	Enable/Disable MOH
+PMHF	MOH Hook Flash
+PMHT	MOH Timer
+PMHR	Initiate MOH
+PQC	Enable/Disable Quick Connect
+PSS	Use Short Startup Sequence
+QCPC	Force Full Startup Sequence
+QCPS	Enable/Disable Quick Connect Profile Save

Caller ID Commands

+VCID	Select data presentation format n (0-2) for incoming calls.
+VRID	Select data presentation format n (0-1) for most recently received incoming call.

<u>COMMAND</u>	<u>FUNCTION/DESCRIPTION</u>
Error Correcting Commands	
%C0	Disable data compression.
%C1	Enable MNP5 data compression.
%C2	Enable V.42bis data compression
%C3	Enable both V.42bis and MNP 5 compression
Stored Profile Commands	
&V	Display current configuration
&W0	Store the active profile in NVRAM profile 0
&W1	Store the active profile in NVRAM profile 1
&Y0	Recall stored profile 0 upon power up
&Y1	Recall stored profile 1 upon power up
&Zn=x	Store dial string x (31 digits max) to location n (0-3)
MNP10 Commands	
-K0	Disable MNP 10 extended services
-K1	Disable MNP 10 extended services
-K2	Disable MNP 10 extended services
FAX Commands	
+FCLASS=n	Service Class
+F<cmd>?	Report active configuration for specified command cmd
+F<cmd>=?	Report capability for specified command cmd
+FAA=n	Data/Fax Auto Answer
+FAE=<value>	Auto Answer
+FLO	Flow Control
+FAR=<off/on>	Adaptive reception control
+FCL=<time>	Carrier loss timeout
+FDD=<value>	Double escape character replacement control
+FIT=<time>,<action>	DTE inactivity timeout
+FPR=<rate>	Fixed DTE Rate
+FMI?	Report Manufacturer ID
+FMM?	Report Model ID
+FMR?	Report Revisions ID
+FRM=n	Receive Data with n protocol
+FRS=<time>	Wait for silence
+FTM=n	Transmit Data with n protocol
+FRH=n	Receive Data with HDLC Framing with n protocol
+FTH=n	Transmit Data with HDLC Framing with n protocol
+FTS=<time>	Wait for silence
+FTS=n	Stop Transmission and Wait
Fast Connect Commands	
\$F0	Normal connection (Default)
\$F1,3,4	Not Supported
\$F2	V.22 Fast Connect
Identification Commands	
I0	Reports product code
I3	Reports firmware Identification code

12. GPRS/RF iMODEM SPECIFIC COMMANDS

The AT commands in this section are for use of the GPRS/RF iModem family of products in Modem Mode. All commands are detailed in the GSM and GPRS standards and are only briefly described below.

IMPORTANT NOTE

The term ME is equivalent to the GPRS/RF iMODEM in this section.

The commands in this section are very useful for isolating GPRS/GSM network problems, and verifying the GPRS/RF iMODEM configuration. They are detailed in the GSM 07.07 standard (ETSI TS 100 916) and the format used to describe the commands is similar to that of the standard.

12.1 GPRS/RF iMODEM: Important Modem Mode Commands.

GPRS/GSM General, Network Service, & Diagnostics

12.1.1 AT+CSQ: Signal Quality and Bit Error Rate

Command Function

Execution command returns received signal strength indication <rssi> and channel bit error rate <ber> from the ME.

COMMAND FUNCTIONAL GROUP

Command Format Query
Response +CSQ:

PHONE CONTROL

AT+CSQ=?
(2-31,99),(99)
OK

Execution Format
Response

AT+CSQ
+CSQ: <rssi>, <ber>
OK

Parameter Values:

<rssi>

0 -113 dBm or less
1 -111 dBm
2-30 -109... -53 dBm
31 -51 dBm or greater
99 not known or not detectable

<ber> (in percent)

0-7 as RXQUAL values in the table in GSM 05.08 [20] subclause 8.2.4
99 not known or not detectable

Reference
Standard Scope

GSM Ref. 07.07 Chapter 8.5
Optional

NOTES

N/A

12.1.2 AT+CREG: Network Registration Information.

Command Function

Write command controls the presentation of an unsolicited result code +CREG: <stat>.

Read command returns the status of result code, which shows whether the network has currently indicated the registration of the ME.

COMMAND FUNCTIONAL GROUP

Command Format Query
Response

Write Format
Response
Read Format
Response

Parameter Values:

<n>

<stat>

<lac>

<ci>

Reference
Standard Scope

NOTES

12.1.3 AT+CGREG: GPRS Network Registration Status.

Command Function

COMMAND FUNCTIONAL GROUP

Command Format Query
Response

Write Format
Response
Read Format
Response

Parameters Values:**NETWORK INFORMATION**

AT+CREG=?
+CREG: (0,2)
OK
AT+CREG=[<n>]
OK
AT+CREG?
+CREG: <n>,<stat>[,<lac>,<ci>]
OK

0 disable network registration
unsolicited result code
1 enable network registration
unsolicited result code +CREG: <stat>
2 enable network registration and
location information unsolicited result
code +CREG: <stat>[,<lac>,<ci>]

0 not registered, ME is not currently
searching a new operator to
register to
1 registered, home network
2 not registered, but ME is currently
searching a new operator to
register to
3 registration denied
4 unknown
5 registered, roaming

string type; two-byte location area code
in hexadecimal format (e.g. "00C3"
equals 195 in decimal)
string type; two-byte cell ID in
hexadecimal format

GSM Ref. 07.07 Chapter 7.2
Optional

N/A

Controls the presentation of an unsolicited
result code +CGREG.

GPRS COMMANDS

AT+CGREG=?
+CGREG: (0,2)
OK
AT+CGREG=1
OK
AT+CGREG?
+CREG: <n>,<stat>[,<lac>,<ci>]
OK

<n> 0 disable network registration
 unsolicited result code
 1 enable network registration
 unsolicited result code +CGREG:
 <stat>
 2 enable network registration and
 location information unsolicited result
 code +CGREG: <stat>[,<lac>,<ci>]

<stat> 0 not registered, ME is not currently
 searching a new operator to register to
 1 registered, home network
 2 not registered, but ME is currently
 searching a new operator to register to
 3 registration denied
 4 unknown
 5 registered, roaming

<lac> string type; two-byte location area code
 in hexadecimal format (e.g. "00C3"
 equals 195 in decimal)

<ci> string type; two-byte cell ID in
 hexadecimal format

Reference GSM Ref. 07.07 Chapter 10.1.13
 Standard Scope Optional

NOTES If parameter <n> is omitted the
 command does nothing.

12.1.4 AT+COPS: Operator Selection

Command Function

Read command returns the current mode
 and the currently selected operator. If no
 operator is selected, <format> and <oper>
 are omitted.

Test command returns a list of quadruplets,
 each representing an operator present in the
 network. Quadruplet consists of an integer
 indicating the availability of the operator
 <stat>, long and short alphanumeric format
 of the name of the operator, and numeric
 format representation of the operator. Any
 of the formats may be unavailable and will
 then be an empty field (,). The list of
 operators comes in the following order:
 Home network, networks referenced in SIM,
 and other networks.

COMMAND FUNCTIONAL GROUP

Command Format Query
 Response

GPRS COMMANDS

AT+COPS=?
 +COPS: (2, " ", " ", "31022"),
 (3, " ", " ", "310380")
 OK

Read Format

AT+COPS?

Response

+COPS: 0
OK**Parameter Values:**

<mode>

0 automatic (<oper> field is ignored)
 1 manual (<oper> field shall be present)
 2 deregister from network
 3 set only <format> (for read command +COPS?), do not attempt registration/deregistration (<oper> field is ignored); this value is not applicable in read command response
 4 manual/automatic (<oper> field shall be present); if manual selection fails, automatic mode (<mode=0) is entered .

<stat>

0 Unknown
 1 Available
 2 Current
 3 Forbidden

Reference
Standard ScopeGSM Ref. 07.07 Chapter 7.3
Optional

NOTES

Example:

To read operator information:

```
AT+COPS=?
+COPS: (2,"Voicestream","Vstream","31022")
```

12.1.5 AT%CGREG: GPRS Extended Registration State

Command Function

This command reports extended information about GPRS registration state. %CGREG behaves exactly as +CGREG does. In addition %CGREG supports three states +CGREG does not support.

COMMAND FUNCTIONAL GROUPCommand Format Query
ResponseWrite Format
Response
Read Format
Response**GPRS COMMANDS**

```
AT%CGREG=?
%CGREG: (0,2)
OK
AT%CGREG=<mode>
OK
AT%CGREG?
%CGREG: <n>,<stat>[,<lac>,<ci>,<act>]
OK
```

Parameter Values:

<mode>

enable or disable extended GPRS registration state reporting
 0 do not report registration state
 1 do report registration state

2 enable network registration and location information unsolicited result code +CGREG: <stat>[,<lac>,<ci>]
 3 enable network registration, location information, and activated/deactivated PDP context unsolicited result code +CGREG: <stat>[,<lac>,<ci>,<act>].

<state>

- 0 not registered
- 1 registered to home network
- 2 not yet registered, but searching for network to register to
- 3 registration denied
- 4 unknown state
- 5 registered to foreign network (roaming)
- 6 limited service (cell might be overloaded)
- 7 GSM call active
- 8 no cell available
- 9 next attempt to update MS

<lac>

string type; two-byte location area code in hexadecimal format (e.g. "00C3" equals 195 in decimal)

<ci>

string type; two-byte cell ID in hexadecimal format

<act>

- 0 deactivated
- 1 activated

Reference
 Standard Scope
 NOTES

N/A
 N/A
 N/A

12.1.6 AT+CIMI: Request IMSI

Command Function

This command is used to obtain the International Mobile Subscriber Identity (IMSI) value assigned to the SIM.

COMMAND FUNCTIONAL GROUP

Command Format Query
 Response
 Execution Format
 Response

EQUIPMENT INFORMATION

AT+CIMI=?
 OK
 AT+CIMI
 310260101xxxxx
 OK

Parameter Values

N/A

Reference
 Standard Scope

GSM Ref. 07.07 Chapter 5.6
 Optional

NOTES

Return value is manufacturer specific. The TA returns the International Mobile Subscriber Identity (IMSI).

GSM Call Control

12.1.7 AT+CEER: Extended Error Reporting

Command Function

This command is used to control the display of extended result codes for last unsuccessful call setup, in-call modification, last call release, last short message, or last GPRS session.

COMMAND FUNCTIONAL GROUP

Command Format Query
Response
Execution Format
Response

CALL CONTROL GROUP

AT+CEER=?
OK
AT+CEER
+CEER: <DEFBY>, <ORIGSIDE>,
<ORIGIN_ENTITY>, <VALUE>
OK

Parameter Values:

<DEFBY> (defined by)

0 - Standard
1 - iMODEM

<ORIGSIDE>(originating side)

0 - Network
1 - MS

<ORIGIN_ENTITY>:

0 - SIM
1 - ACI
2 - RLP
3 - RR
4 - MM
5 - CC
6 - SS
7 - SMSCP
8 - SMSRP
9 - SMSTP
10 - GMM
11 - SM
12 - FAD
13 - T30
14 - GRR
15 - PPP
16 - LLC
17 - SNDCP
18 - PKTIO
19 - PSI

<VALUE>

See AT+CEER Table in Appendix B

Reference
Standard Scope

GSM Ref. 07.07 Chapter 6.10,
Optional

NOTES

N/A

12.1.8 AT+CPOL: Preferred Operator List

Command Function

This command is used to list and edit the SIM preferred list of networks.

COMMAND FUNCTIONAL GROUP

Command Format Query
Response

Write Format
Response
Read Format
Response

Parameter Values:

<indexn>:

<format>:

<opern>:

Reference
Standard Scope

NOTES

12.1.9 AT+CLCC: List Current Calls

Command Function

COMMAND FUNCTIONAL GROUP

Command Format Query
Response
Execution Format
Response

NETWORK

AT+CPOL=?
+CPOL: (1-30), (0-2)
OK
AT CPOL=[<index>][, <format>[, <oper>]]
OK
AT+CPOL?
+CPOL: <index1>,<format>,<oper1>...
<index10>,<format>,<oper10>
OK

integer type; the order number of operator in the SIM preferred operator list

0 long format alphanumeric <oper>
1 short format alphanumeric <oper>
2 numeric <oper>

string type; <format> indicates if the format is alphanumeric or numeric (see +COPS)

GSM Ref. 07.07 Chapter 7.18
Optional

This command is used to edit the SIM preferred list of networks. Execute command writes an entry in the SIM list of preferred operators (EFPLMNsel). If <index> is given but <oper> is left out, entry is deleted. If <oper> is given but <index> is left out, <oper> is put in the next free location. If only <format> is given, the format of the <oper> in the read command is changed.

Returns list of current calls of ME. If Command succeeds but no calls are available, no information response is sent to TE.

CALL CONTROL

AT+CLCC=?
OK
AT+CLCC
[+CLCC: <id1>,<dir>,<stat>,<mode>,<mpty>[, <number>,<type>[, <alpha>]]
[<CR><LF>+CLCC: <id2>,<dir>,<stat>,<mode>,<mpty>[, <number>,<type>[, <alpha>]]
[...]]
OK

Parameter Values:

<idx>	integer type; call identification number as described in GSM 02.30 [19] subclause 4.5.5.1; this number can be used in +CHLD command operations
<dir>	0 mobile originated (MO) call 1 mobile terminated (MT) call
<stat>	(state of the call): 0 active 1 held 2 dialing (MO call) 3 alerting (MO call) 4 incoming (MT call) 5 waiting (MT call)
<mode>	(bearer/teleservice): 0 voice 1 data 2 fax 3 voice followed by data, voice mode 4 alternating voice/data, voice mode 5 alternating voice/fax, voice mode 6 voice followed by data, data mode 7 alternating voice/data, data mode 8 alternating voice/fax, fax mode 9 unknown
<mpty>	0 call is not one of multiparty (conference) call parties 1 call is one of multiparty (conference) call parties
<number>	string type phone number in format specified by <type>
<type>	type of address octet in integer format (refer GSM 04.08 [8] subclause 10.5.4.7)
<alpha>	string type alphanumeric representation of <number> corresponding to the entry found in phonebook; used character set should be the one selected with command Select TE Character Set +CSCS
Reference Standard Scope	GSM Ref. 07.07 Chapter 7.17 Optional
NOTES	N/A

GSM Modem Equipment Errors

12.1.10 AT+CMEE: Report Mobile Equipment Errors

Command Function

Set command disables or enables the use of result code +CME ERROR: <err> as an indication of an error relating to the functionality of the ME. When enabled, ME related errors cause +CME ERROR: <err> final result codes to be returned, instead of the default ERROR final result code. ERROR is returned normally when error is related to syntax, invalid parameters, or TA functionality.

COMMAND FUNCTIONAL GROUP

Command Format Query
Response

Write Format
Response
Read Format
Response

Parameter Values:

<n>

Reference
Standard Scope

NOTES

RESPONSE CONTROL

AT+CMEE=?
+CMEE: (0-2)
OK

AT+CMEE=<n>
OK
AT+CMEE?
+CMEE: 0
OK

0 Disable +CME ERROR
1 Enable +CME result code and
username values
2 Enable +CME result code and ME
verbose values

GSM Ref. 07.07 Chapter 9.1
Mandatory

See Appendix B for error code
descriptions.

GPRS/GSM Standard Commands

12.1.11 AT+CGPADDR: Show PDP Address

Command Function

The execution command returns a list of PDP addresses for the specified context identifiers.

COMMAND FUNCTIONAL GROUP

Command Format Query
Response

Execution Format
Response

Parameter Values:

<cid>

Reference
Standard Scope

NOTES

GPRS COMMANDS

```
AT+CGPADDR=?
+CGPADDR: (1)
OK
AT+CGPADDR=<cid>
+CGPADDR: 1
OK
```

numeric value of PDP context activation

GSM Ref. 07.07 Chapter 10.1.7
Optional

N/A

12.1.12 AT\$LOCIP: Display Local Modem to Host IP & DNS

Command Function

This command allows the user to query the modem's locally assigned IP.

COMMAND FUNCTIONAL GROUP

Read Format
Response

Parameter Values:

<IP>

<DNS1>
<DNS2>

Reference
Standard Scope

NOTES

PROPRIETARY

```
AT$LOCIP?
<"IP">,<"DNS1">,<"DNS2">
```

local host to modem IP

local host to modem DNS1
local host to modem DNS2

N/A
Optional

N/A

12.1.13 AT\$NETIP: Display Network Assigned IP & DNS

Command Function

This command allows the user to query the modem's network assigned IP.

COMMAND FUNCTIONAL GROUP

Read Format
Response

Parameter Values:

<IP>

PROPRIETARY

```
AT$NETIP?
<"IP">,<"DNS1">,<"DNS2">
```

network assigned IP

<DNS1> network assigned DNS1
 <DNS2> network assigned DNS2

Reference N/A
 Standard Scope Optional

NOTES N/A

12.1.14 AT\$GATEWAY: Gateway IP.

Command Function

This command allows the user to select a gateway IP. Windows CE 3.0 devices and some Linux platforms require a gateway address. Default value "0.0.0.0" indicates that no gateway IP will be requested from the host. A non-zero value will cause the modem to request the indicated gateway IP from the host.

COMMAND FUNCTIONAL GROUP

Command Format Query
 Response

PROPRIETARY

AT\$GATEWAY=?
 \$GATEWAY: ("<IP>")
 OK
 AT\$GATEWAY?
 \$GATEWAY: "<IP >"

Read Format
 Response

Parameter Values:

<IP> gateway IP address.

Standard Scope Optional

NOTES N/A

12.2 Standard GPRS/RF iMODEM Modem Commands.

The following modem commands are Hayes compatible and work as described in TIA/EIA 602-A and ITU-T V.250.

AT	Attention
ATE	Command Echo Mode
AT&V	Display Current Profile
AT+GCAP	Request Overall Capabilities for TA

13. HWX - HARDWARE EVENT SIGNALING INTERFACE

HWX is a hardware interface which facilitates the exchange of information between the iMODEM and the host hardware environment, and is specific to the CH2166A and CH2168A GPRS/iMODEM models.

HWX is used to signal events and convey voltage levels occurring in the host hardware environment to the iMODEM. It is also used to signal events detected by the iMODEM to the host hardware environment.

HWX includes API commands for configuring and controlling the operation of the hardware interface. Those API commands are exchanged over the serial port in the same fashion as all other iMODEM API commands.

iMODEM detection of a host hardware environment event or level it has been programmed to detect is known as a "trigger condition". Both the criteria that constitutes a trigger condition and the iMODEM response to a particular trigger condition are configurable with the HWX API commands.

13.1 HWX Hardware.

The functionality of the HWX hardware inputs and outputs is as follows:

INPUT TRIGGER - Notifies the iMODEM of an event in the host hardware environment when it is actuated. It is a digital input which is active low and it is actuated by pulling it low for about 20 MS. Note, it should not be held low for more than 20 MS.

STATE1 INPUT - The host hardware environment sets the state of this line according to the application requirements. It is a binary state digital input line and is active low. The iMODEM input value is 1 (TRUE) when the line is driven low by the host hardware environment and it is 0 (FALSE) when it is held high in the host hardware environment. A state must be maintained a minimum of 10mS in order to be detected.

STATE2 INPUT - The host hardware environment sets the state of this line according to the application requirements. It is a binary state digital input line and is active low. The iMODEM input value is 1 (TRUE) when the line is driven low by the host hardware environment and it is 0 (FALSE) when it is held high in the host hardware environment. A state must be maintained a minimum of 10mS in order to be detected.

ADC INPUT -The host hardware environment sets the level of this line according to the application requirements. It is analog DC input whose value ranges for 0 to 3.3 volts.

OUTPUT TRIGGER - Generally, the iMODEM actuates this line to notify the host hardware environment that a trigger condition has been detected by the iMODEM. The exact trigger condition that actuates this line must be configured with the HWX API commands. It is active low and the iMODEM actuates it by pulling it low for 70mS. The actuation period can be configure with the TOUT_CONFIG API command.

The exact pins used for HWX functions are different for the CH2166A and CH2168A iMODEM models.

CH2166A function to pin correspondence.

<u>Function</u>	<u>Label</u>	<u>Pin Number</u>
INPUT TRIGGER	IN1	7
STATE1 INPUT	IN2	9
STATE2 INPUT	IN3	12
ADC INPUT	AD1	18
OUTPUT TRIGGER	SENT	8
GROUND	GND	6

CH2168A function to pin correspondence.

<u>Function</u>	<u>Label</u>	<u>Pin Number</u>
INPUT TRIGGER	IN1	1
STATE1 INPUT	IN2	2
STATE2 INPUT	IN3	3
ADC INPUT	AD1	5
OUTPUT TRIGGER	SENT	4
GROUND	GND	6

13.2 HWX API Command Overview.

There are 5 categories of HWX API commands and messages, and the relevant sections are as follows:

13.3 - Trigger Condition Configuration Commands (i.e. These apply only to the ADC and state input lines.)
 Commands: ADC_TRIGGER_CONFIG, STATE_TRIGGER_CONFIG.

13.4 - Low Level ADC Configuration Command: ADC_DEVICE_CONFIG

13.5 - Response Configuration Commands: HWX_RESPONSE_CONFIG, TOUT_CONFIG

13.6 - Device Control API Commands: HWX_START, HWX_AUTO_CONFIG, HWX_STOP, HWX_SHUTDOWN, HWX_TRIGGER

13.7 - User Notification API Messages: EVENT, OK, FAULT

IMPORTANT NOTE

The commands and messages in sections 13.6 and 13.7 require that a session ID has been obtained with the G_GET_SESSION_ID API command for the type HWX. Also, only one HWX session ID can be allocated.

Example 13.2A:

<u>USER</u>	<u>IMODEM</u>
G_GET_SESSION_ID: HWX ==>	
<==	POSTED: Operation Started
<==	OK[0]: HWX Event ID Allocated

When a trigger event is detected, several iMODEM responses are possible. The standard ones are as follows:

- Report the event to the host via the serial port.
- Actuate the TRIGGER OUTPUT.
- Send an E-Mail message.

The iMODEM can be configured to generate any combination of the responses above. If more than one response is configured, they occur in the following order.

- 1) E-Mail is sent.
- 2) The TRIGGER OUTPUT is actuated.
- 3) The event is reported.

In addition, the iMODEM can be configured to generate a custom response. This requires on board iMODEM code support, and the following routines can be customized for that purpose:

```
BOOL u_trigger_event(int channel_no, int rsp_code)
```

This routine handles a trigger condition generated by the INPUT TRIGGER.

BOOL u_adc_event(int channel_no, int rsp_code, int sample_value)
This routine handles a trigger condition generated by the ADC INPUT.

BOOL u_state1_event(int channel_no, int rsp_code, int state)
This routine handles a trigger condition generated by the STATE1 INPUT,
or the STATE1 and STATE2 INPUTs in case of Dual Monitoring described
below.

BOOL u_state2_event(int channel_no, int rsp_code, int state)
This routine handles a trigger condition generated by the STATE2 INPUT.

These routines must return TRUE if they are used and FALSE otherwise. It is the responsibility of the custom code designer to define the actions taken by these routines.

The following routines--described in sections below, are available to customize iMODEM trigger condition detection:

int u_adc_calculate(int previous , int current)
BOOL u_adc_evaluate(int last_sample, int current_sample)
BOOL u_state1_evaluate(int last_state, int current_state)
BOOL u_state2_evaluate(int last_state, int current_state)

In addition the following routines can be used to customize internal low level iMODEM responses to a trigger condition:

BOOL u_trigger_response(int rsp_code)
BOOL u_adc_response(int value , int rsp_code)
BOOL u_state1_response(int state, int rsp_code)
BOOL u_state2_response(int state, int rsp_code)

13.3 Trigger Condition Configuration Commands.

COMMANDS

ADC_TRIGGER_CONFIG
STATE_TRIGGER_CONFIG

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

13.3.1 ADC_TRIGGER_CONFIG: Configure ADC Input Line Trigger Conditions

This command configures what ADC sample values constitute a trigger condition. Selecting the appropriate values requires that the step size, as described in section 13.4, is known.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode. This command does not start trigger condition detection, or configure the iMODEM response to a trigger condition.

SYNTAX

ADC_TRIGGER_CONFIG: TYPE=<Mode> INTERVAL=<Time> SUM=<SValue>
RANGE=<RDef> LOWER=<Lower_Bound> UPPER=<Upper_Bound>
REFINE=<RType> CALC

PARAMETERS

TYPE: (Mandatory) The trigger detection mode. The following Mode values are valid:

SAMPLE - Each sample is a trigger condition. All other parameters except **CALC** are illegal in this mode.

SUM - A set of **N** samples is a trigger condition. The value of **N** must be specified with the **SUM** parameter.

INTERVAL - When a period of **N** MS has elapsed, it is a trigger condition. The value of **N** must be specified with the **INTERVAL** parameter.

LIMIT - When a sample value **X** exceeds **N**, it is a trigger condition. Exclusively the **LOWER** or **UPPER** parameter must specify **N**. If the **LOWER** parameter is specified the comparison $X < N$, is used to detect a trigger condition, and if the **UPPER** parameter is specified the comparison $X > N$ is used to detect a trigger condition. In addition, the **REFINE** parameter can be used to refine how the comparison is performed. Whether a raw or transformed sample value is used in the comparison depends on the whether the **CALC** parameter, described below, is specified.

RANGE - Depending on the configuration, it is a trigger condition if a sample **X** is in a range, or a sample **X** is outside of a range. The **RANGE** parameter specifies whether the trigger condition criteria is that **X** is in or outside the range and is mandatory. The **LOWER** and **UPPER** parameters are also mandatory. If the **REFINE** parameter is specified, the same "refinement" applies to both the **UPPER** and **LOWER** parameters.

CUSTOM - Custom on board iMODEM code is used detect a trigger condition. It is the responsibility of the custom code designer, to define what sample values constitute a trigger condition and how to evaluate them. The routine `u_adc_evaluate()` is used to detect trigger conditions. It returns **TRUE** when a trigger condition is detected and **FALSE** otherwise, and has the following syntax:

BOOL `u_adc_evaluate(int last_sample, int current_sample)`

The variables `current_sample` and `last_sample` are the most recent and second most recent ADC samples respectively.

INTERVAL: Time is the period in MS that constitutes a trigger condition. This parameter is only valid in **INTERVAL** mode.

SUM: **Svalue** is the number of contiguous samples that constitute a trigger condition. This parameter is only valid in **SUM** mode.

RANGE: **Rtype** specifies the range criteria used to detect a trigger condition. Its value determines what test is used to evaluate sample value **X**, and **INTERIOR** and **EXTERIOR** are the valid values, and they are used as follows:

INTERIOR - **X** is tested to determine if it is in a specific range, **X** is in the range if it satisfies the following relationship:
 $a < X < b$ where **a** and **b** are respectively the lower and upper bound of the range. (i.e. The range is a open interval.)

EXTERIOR - **X** is tested to determine if it is outside a specific range. **X** is outside the range if it does not satisfy the following relationship: $a < X < b$ where **a** and **b** are respectively the lower and upper bound of the range. (i.e. The range is an open interval.)

This parameter is only valid in **RANGE** mode, and if it is specified the **LOWER** and **UPPER** parameters are mandatory.

LOWER: **Lower_Bound** is the value used to determine if a sample $X < \text{Lower_Bound}$. The **REFINE** parameter can be used to refine how the comparison is performed.

UPPER: **Upper_Bound** is the value used to determine if a sample $X > \text{Upper_Bound}$. The **REFINE** parameter can be used to refine how the comparison is performed.

REFINE: Refines how lower and upper bound comparisons are performed. The valid values and syntax follow. This parameter is only valid in **RANGE** and **LIMIT** mode and requires that the **UPPER** or **LOWER** parameter or both are specified.

CONTIG - A set of **N** contiguous samples that exceeds a specific limit is a trigger condition. The syntax is as follows:

REFINE=CONTIG,<N>

N is the number of contiguous samples.

AVERAGE - A set of N contiguous samples that have an average value which exceeds a specific limit is a trigger condition. The average value is calculated relative to the beginning of ADC operation, and is the sum of samples acquired divided by the number of samples acquired since that point. The syntax is as follows:

```
REFINE=AVERAGE,<N>
```

N is the number of contiguous samples.

SUBSET - A subset of M samples out of a set of N contiguous samples that exceeds a specific limit is a trigger condition. The syntax is as follows:

```
REFINE=SUBSET,<N>,<M>
```

N is the number of contiguous samples.

M is the number of samples in the subset.

CALC: Transform raw ADC samples before they are evaluated. The custom on board iMODEM routine `u_adc_calculate()` performs the transformation. It is the responsibility of the custom code designer to define what transformation is performed and how it is done. The syntax is as follows:

```
int u_adc_calculate(int previous, int current)
```

The routine returns the transformed value. The parameter `current` is the most recent raw ADC sample acquired, and `previous` is the value returned by the last call to the routine.

Typical Examples 13.3A:

<u>USER</u>	<u>iMODEM</u>
ADC_TRIGGER_CONFIG: type=INTERVAL \ interval=5000	==> <== POSTED: Operation Started <== OK: HWX Operation OK
ADC_TRIGGER_CONFIG: type=SAMPLE	==> <== POSTED: Operation Started <== OK: HWX Operation OK
ADC_TRIGGER_CONFIG: type=LIMIT \ LOWER=200 \ REFINE=SUBSET,7,5	==> <== POSTED: Operation Started <== OK: HWX Operation OK
ADC_TRIGGER_CONFIG: type=RANGE \ RANGE=EXTERIOR \ LOWER=100 UPPER=900	==> <== POSTED: Operation Started <== OK: HWX Operation OK

13.3.2. STATE_TRIGGER_CONFIG: Configure State Input Line Trigger Condition

This command configures how the digital state input lines are monitored and what criteria constitutes a trigger condition. It can be used to configure a single line to be monitored independently, or a pair of lines to be monitored jointly. (i.e. Dual Monitoring)

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in

Multi-mode. This command does not start trigger condition detection, or configure the iMODEM response to a trigger condition.

SYNTAX

```
STATE_TRIGGER_CONFIG: LINE=<N> TYPE=<Mode> INTERVAL=<Time>
                    S1=<State1> S2=<State2>
```

PARAMETERS

LINE: (Mandatory) Specifies the lines being configured, and the following value are valid for N:

1 - Monitor line 1 independently. The device name STATE1 refers to his line with respect to other device specific HWX API commands.

2 - Monitor line 2 independently. The device name STATE2 refers to this line with respect to other device specific HWX API commands.

3 - Monitor lines 1 and 2 jointly. The device name STATE1 refers to both lines with respect to other device specific HWX API commands.

TYPE: (Mandatory) The monitoring mode. The following Mode values are valid:

MONO_STATE - A transition to a specific state, or state pair, specified by S1 or S2, described below, is a trigger condition. For Dual Monitoring, parameters S1 and S2 are mandatory. S1 is mandatory if LINE has the value 1, and S2 is mandatory if LINE has the value 2.

BI_STATE - Any state transition is a trigger condition.

MONO_TIMED - (Mono-Interval Mode) When a specific state , or state pair, specified by S1 or S2, described below, is maintained for the duration specified by the INTERVAL parameter—described below, it is a trigger condition. For dual monitoring, parameters S1 and S2 are mandatory. S1 is mandatory if LINE has the value 1, and S2 is mandatory if LINE has the value 2.

BI_TIMED - (Bi-Interval Mode) When the same state, or state pair, is maintained for the duration specified by the INTERVAL parameter— described below, it is a trigger condition.

CUSTOM - Custom on board iMODEM code is used to monitor a state, or state pair, and to detect a trigger condition. It is the responsibility of the custom code designer to define what criteria constitute a trigger condition and how to detect that criteria. The custom code is implemented by the routines described below. They all return TRUE if a trigger condition is detected and FALSE otherwise. The routines and the state inputs to which they apply are as follows:

IMPORTANT NOTE

The S1, S2, and INTERVAL parameters are illegal in this case.

1 - BOOL u_state1_evaluate(int last_state, int current_state)

The variables last_state and current_state contain the line 1 value prior to the last call to the routine, and the current value. Variable values of TRUE and FALSE correspond to line values of 1 and 0 respectively.

2 - BOOL u_state2_evaluate(int last_state, int current_state) The variables last_state and current_state contain the line 2 value prior to the last call to the routine, and the current value. Variable values of TRUE and FALSE correspond to line values of 1 and 0 respectively.

3 - BOOL u_state1_evaluate(int last_state, int current_state) The variables last_state and current_state contain both the line 1 and 2 value prior to the last call to the routine, and the current value. Bit 0 is the value of line 1, and bit 1 is the value of line 2, and the bit value is equivalent to the line value.

INTERVAL: This parameter is mandatory in BI_TIMED and MONO_TIMED monitoring mode. It is the interval for which a state, or state pair, must be maintained for a trigger condition to occur in those modes. Time is expressed in milliseconds and must be

greater than 0. Durations of less than 200 MS are not reliable, and there is variable latency between the time of trigger condition detection and response.

S1: This parameter is mandatory in MONO_STATE and MONO_TIMED mode for the configuration of line 1 and for Dual Monitoring. State1 is the line 1 state monitored in order to detect a trigger condition. HIGH and LOW are the valid values, and correspond to a line value of 1 and 0 respectively. (i.e. It is illegal if LINE has the value 2.)

S2: This parameter is mandatory in MONO_STATE and MONO_TIMED mode for the configuration of line 2 and for Dual Monitoring. State2 is the line 2 state monitored in order to detect a trigger condition. HIGH and LOW are the valid values, and correspond to a line value of 1 and 0 respectively. (i.e. It is illegal if LINE has the value 1.)

Typical Examples 13.3B:

<u>USER</u>		<u>iMODEM</u>
STATE_TRIGGER_CONFIG: line=1 \ type=MONO_STATE \ s1=HIGH	==>	
	<==	POSTED: Operation Started
	<==	OK: HWX Operation OK
STATE_TRIGGER_CONFIG: line=2 \ type=MONO_INTERVAL \ s2=HIGH interval=5000	==>	
	<==	POSTED: Operation Started
	<==	OK: HWX Operation OK
STATE_TRIGGER_CONFIG: line=3 \ type=MONO_STATE \ s1=HIGH s2=LOW	==>	
	<==	POSTED: Operation Started
	<==	OK: HWX Operation OK

13.4 Low Level ADC Configuration Command.

COMMAND

ADC_DEVICE_CONFIG

The iMODEM will respond with an ERROR rather than OK message if this command fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of this command is described below.

13.4.1 ADC_DEVICE_CONFIG: Configure ADC Operation Parameters

This command configures low level ADC operating parameters.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode. This command does not start trigger condition detection, or configure the iMODEM response to a trigger condition.

SYNTAX

ADC_DEVICE_CONFIG: RATE=<SCode> QSIZE=<NQelem> STEPS=<NSteps>

PARAMETERS

RATE: (Mandatory) The sampling rate specified the number of samples per second acquired by the ADC in samples per second. The following codes are valid Scode values and their significance is as follows:

<u>CODE</u>	<u>RATE</u>	<u>CODE</u>	<u>RATE</u>
70	133.5	83	8.5
80	66.8	84	4.18
81	33.4	85	2.8
82	16.7		

QSIZE: The maximum number of samples that can be stored without being processed. The NQelem value must be a power of 2 greater or equal to the sampling rate. Generally, a value which at least twice the sampling rate is sufficient. However, a larger value may be necessary if there is latency in processing the samples. If the QSIZE parameter is omitted, the smallest power of 2 which is greater than twice the sampling rate will be used as a default.

STEPS: NSteps is the number of discreet quantum steps into which a continuous DC input voltage from 0 - 3.3 Volts is divided. The value must be between 2 and 4096, and 4096 is the default value. Each step corresponds, roughly, to an equal increment of voltage.

Typical Example:

<u>USER</u>		<u>iMODEM</u>
ADC_DEVICE_CONFIG: rate=84 \	==>	
qsize=4096 steps=1000		
	<==	POSTED: Operation Started
	<==	OK: HWX Operation OK

13.5 Response Configuration Commands.

COMMANDS

HWX_RESPONSE_CONFIG
TOUT_CONFIG

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. (i.e. Appendix A lists the error codes appended to the ERROR message.) The syntax and use of the commands are described below.

13.5.1 HWX_RESPONSE_CONFIG: Configure the iMODEM Trigger Condition Response

This command configures how the iMODEM responds to a trigger condition on a per device basis.

This command configures how the iMODEM responds to a trigger condition on a per device basis.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode. This command does not start trigger condition detection.

SYNTAX

```
HWX_RESPONSE_CONFIG: DEVICE=<Dname> RSP=<Rtype> AUX1=<A1type> AUX2=<A2type>
                  FLOW_VALUE=<Vtype> PAUSE=<Time> PROFILE=<Pname>
                  FILE=<Fname> SUBJECT=<Sub> N=<Ntrys> INTERNAL
```

PARAMETERS

DEVICE: (Mandatory) Specifies the HWX device being configured.
The following Dname values are valid:

ADC - The A to D Converter with a 0 - 3.3 volt analog input line.

TRIGGER - The digital trigger input line.

STATE1 - The binary state digital state1 input line. [1]

STATE2 - The binary state digital state2 input line.

NOTE 1: This device can apply to both state input lines as detailed in LINE parameter description of the STATE_TRIGGER_CONFIG command.

RSP: (Mandatory) The main response to a trigger condition. The following Rtype values are valid:

MAIL - Send E-Mail. In this case the FILE parameter is mandatory.

REPORT - Send an EVENT API message to the iMODEM host via the serial port.

ACTUATE - Actuate the digital trigger output line.

CUSTOM - A custom on board iMODEM response is performed by one of the following routines:

```
BOOL u_trigger_event(int channel_no, int rsp_code)
```

```
BOOL u_adc_event(int channel_no, int rsp_code, int sample_value)
```

```
BOOL u_state1_event(int channel_no, int rsp_code, int state)
```

```
BOOL u_state2_event(int channel_no, int rsp_code, int state)
```

AUX1: The primary auxiliary trigger condition response. Only the values REPORT and ACTUATE are valid A1type values.

AUX2: The secondary auxiliary trigger condition response. Only the values REPORT and ACTUATE are valid A2type values.

FLOW: EVENT message flow control. The host must acknowledge EVENT API messages with an OK when this parameter is specified. This parameter is applicable only if REPORT is one of the specified responses.

VALUE: Specifies the kind of value to be reported in an EVENT message. The following Vtype values are valid and the parameter is applicable only if REPORT is one of the specified responses and the device is the ADC INPUT. The default—SAMPLE—is used if this parameter is omitted.

SAMPLE: The ADC sample value—raw or transformed depending on the CALC parameter of the ADC_TRIGGER_CONFIG API.

SUM: The number of ADC samples acquired since ADC operation started.

AVERAGE: The average ADC sample value since ADC operation started. The average is defined as the sum of ADC sample values divided by the number of samples acquired. The sample values used in the calculation --raw or transformed—depend on whether the CALC parameter is specified with the ADC_TRIGGER_CONFIG API.

PAUSE: The amount of time to suspend operation of the device after that device detected a trigger condition. The value is specified in MS. The device is reinitialized after the suspension is complete.

PROFILE: The name of the SMTP Profile used to derive parameters for E-Mail transmission. This parameter is only applicable if the FILE parameter is specified. If FILE is specified and this parameter is omitted, the default SMTP profile is used.

FILE: The name of a file, containing an E-Mail message, to send. This parameter is mandatory if RSP is MAIL.

SUBJECT: The subject of an E-Mail message. This parameter is applicable only if the FILE parameter is specified.

N: The maximum number of E-Mail message transmission attempts. Transmission is attempted only once if this parameter is omitted. This parameter is applicable only if the FILE parameter is specified.

INTERNAL: Indicates the file containing E-Mail has internal headers. This parameter is applicable only if the FILE parameter is specified.

Typical Examples 13.5A:**USER****iMODEM**

```

HWX_RESPONSE_CONFIG: device=ADC \      ==>
    rsp=REPORT pause=30000
                                         <==
                                         <==
                                         POSTED: Operation Started
                                         OK: HWX Operation OK

HWX_RESPONSE_CONFIG: device=TRIGGER \   ==>
    rsp=MAIL aux1=ACTUATE \
    file=test
                                         <==
                                         <==
                                         POSTED: Operation Started
                                         OK: HWX Operation OK

HWX_RESPONSE_CONFIG: device=STATE1 \    ==>
    rsp=REPORT \
    aux1=ACTUATE FLOW
                                         <==
                                         <==
                                         POSTED: Operation Started
                                         OK: HWX Operation OK

```

13.5.2 TOUT_CONFIG: Configure the TRIGGER OUTPUT

This command configures the Trigger Output attributes.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode. This command does not start trigger condition detection.

SYNTAX

TOUT_CONFIG: T=<Duration>

PARAMETERS

T: (Mandatory) Duration is the period, in milliseconds, for which the Trigger Output is pulled low when it is actuated. The default period is 70 MS.

Example 13.5B:**USER****iMODEM**

```

TOUT_CONFIG: T=100      ==>
                        <==
                        <==
                        POSTED: Operation Started
                        OK: HWX Operation OK

```

13.6 Device Control API Commands.**COMMANDS**

```

HWX_START
HWX_AUTO_CONFIG
HWX_STOP
HWX_SHUTDOWN
HWX_TRIGGER

```

The iMODEM will respond with an ERROR rather than OK message if one of the commands fails. Appendix A lists the error codes appended to the ERROR message. The syntax and use of the commands are described below.

13.6.1 HWX_START: Start operation of an HWX device

This API command starts operation of a specific HWX input device. The device can detect trigger conditions and the iMODEM can respond to them after this command is issued. The device trigger response must be configured with the of the API command in section 13.5 before this command can be used. The digital input state lines and ADC input line, trigger condition detection mode must be configured with the appropriate API commands in section 13.3 before this command can be used. Finally, the ADC operational parameters must be configured with the API command in section B before this command can be used.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID—SID.

SYNTAX

HWX_START[<SID>]: DEVICE=<DName>

PARAMETERS

DEVICE: (Mandatory) The name of the device being started.
The following Dname values are valid:

ADC - The A to D Converter with a 0 - 3.3 volt analog input line.

TRIGGER - The digital trigger input line.

STATE1 - The binary state digital state1 input line. [1]

STATE2 - The binary state digital state2 input line.

NOTE 1: This device can apply to both state input lines as detailed in LINE parameter description of the STATE_TRIGGER_CONFIG command.

Example 13.6A:

USER

```
HWX_START[0]: device=ADC      ==>
                               <==
                               <==
```

iMODEM

```
POSTED[0]: Operation Started
OK[0]: HWX Operation OK
```

13.6.2 HWX_AUTO_CONFIG: Configure a HWX device to automatically begin operation.

This API command configures an HWX device to automatically begin operation upon iMODEM power up. Upon successful power up, the HWX devices will function the same as if they were started with the HWX_START command (see HWX_START command description above for details). In addition, the configuration requirements that apply to the HWX_START command also apply to the HWX_AUTO_CONFIG command.

The HWX_AUTO_CONFIG command does not take effect until after the iMODEM is successfully rebooted. After rebooting, the iMODEM will be in Multi-mode and a connection ID of 0 is automatically allocated for HWX operations. HWX operations can be terminated by issuing the HWX_SHUTDOWN or HWX_STOP commands in the normal way. The iMODEM can then be returned to Uni-Mode with the G_FREE_SESS_ID API command.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-Mode

SYNTAX

HWX_AUTO_CONFIG: DEVICE=<DName> AUTO=<TF>

PARAMETERS

DEVICE: (Mandatory). The name of the device being configured. The following Dname values are valid:

ADC - The A to D Converter with a 0 - 3.3 volt analog input line.

TRIGGER - The digital trigger input line.

STATE1 - The binary state digital state1 input line. [1]

STATE2 - The binary state digital state2 input line.

NOTE 1: This device can apply to both state input lines as detailed in LINE parameter description of the STATE_TRIGGER_CONFIG command

TF: A value of TRUE configures the device to automatically begin operation. A value of FALSE configures the device to not automatically begin operation.

Example 13.6B: Typical Example

USER

```
HWX_AUTO_CONFIG: device=ADC
                  Auto=TRUE  ==>
                              <==
                              <==
```

iMODEM

```
POSTED: Operation Started
OK: HWX Operation OK
```

13.6.3 HWX_STOP: Stop operation of specific HWX input device

This API command terminates operation of a device started with the HWX_START API command.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID—SID.

SYNTAX

```
HWX_STOP[<SID>]: DEVICE=<DName>
```

PARAMETERS

DEVICE: (Mandatory) The name of the device being started. The following Dname values are valid:

ADC - The A to D Converter with a 0 - 3.3 volt analog input line.

TRIGGER - The digital trigger input line.

STATE1 - The binary state digital state1 input line. [1]

STATE2 - The binary state digital state2 input line.

NOTE 1: This device can apply to both state input lines as detailed in LINE parameter description of the STATE_TRIGGER_CONFIG command.

Example 13.6C:

USER

```
HWX_STOP[0]: device=STATE2  ==>
                              <==
                              <==
```

iMODEM

```
POSTED[0]: Operation Started
OK[0]: HWX Operation OK
```

13.6.4 HWX_SHUTDOWN: Stop operation of all HWX input devices

This API command stops operation of all HWX devices started with HWX_START API command.

This API command stops operation of all HWX devices started with HWX_START API command.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID—SID.

SYNTAX

HWX_SHUTDOWN[<SID>]

PARAMETERS

None.

Example 13.6D:

USER

```
HWX_SHUTDOWN[0]      ==>
                      <==
                      <==
```

iMODEM

```
POSTED[0]: Operation Started
OK[0]: HWX Operation OK
```

13.6.5 HWX_TRIGGER: Actuate the OUTPUT TRIGGER

This API command actuates the output trigger.

USAGE - This command can only be issued if no other iMODEM API command with the same Session ID is executing. It requires a previously allocated Session ID—SID.

SYNTAX

HWX_TRIGGER[<SID>]

PARAMETERS

None.

Example 13.6E:

USER

```
HWX_TRIGGER[0]      ==>
                      <==
                      <==
```

iMODEM

```
POSTED[0]: Operation Started
OK[0]: HWX Operation OK
```

13.7 User Notification API Messages

MESSAGE OR COMMAND

```
EVENT
OK
FAULT
```

13.7.1 EVENT: Report a Trigger Condition to the iMODEM host

This API message is sent to the host by the iMODEM via the serial port when a trigger condition is detected by the iMODEM. It is only sent if the device that generated the trigger condition is configured to send it with the “REPORT” response of the HWX_RESPONSE_CONFIG API command.

This message is sent to the host asynchronously unless the associated HWX device is configured to use flow control with the FLOW parameter of HWX_RESPONSE_CONFIG API command. In that case, the host must acknowledge the message with an OK before the iMODEM will send another EVENT message. Messages are delivered to the host in the order in which the corresponding trigger conditions occur. However, if flow control is used, the oldest message will be discarded if another trigger event occurs and delivery of more than 64 messages is pending.

USAGE - This message uses a previously allocated Session ID--SID.

SYNTAX

EVENT[<SID>]:<Device_ID>,<Value>

PARAMETERS

Device_ID: The code corresponding to the device that generated the trigger condition. It may have the following values:

- 1 - The ADC analog input line.
- 2 - The trigger input line.
- 3 - Digital state line 1. [1]
- 4 - Digital state line 2.

NOTE 1: This device can apply to both state input lines as detailed in LINE parameter description of the STATE_TRIGGER_CONFIG command.

Value: This parameter is device and dependent and its significance for each HWX device is as follows:

INPUT TRIGGER - Not used.

STATE1 INPUT - The value of digital state input line 1. [2]

STATE2 INPUT - The value of digital state input line 2.

ADC INPUT - The value reported depends on the VALUE parameter of the HWX_RESPONSE_CONFIG API command.

NOTE 2: The value of both state input lines is used if the LINE parameter is set to 3 as detailed in the STATE_TRIGGER_CONFIG command description. It is a digit corresponding to a two bit, and the lower order bit is the state1 line value, and the high order bit is the state2 line value.

Example 13.7A:

<u>USER</u>	<u>iMODEM</u>
<==	EVENT[0]:1,2008

13.7.2 OK Acknowledgment

This API message is used by the iMODEM host to acknowledge an EVENT message from the iMODEM when flow control is configured.

USAGE - This message uses a previously allocated Session ID—SID. The iMODEM does not send a POSTED message or any other response, if the OK is valid.

SYNTAX

OK[<SID>]: DEVICE=<Dname>

PARAMETERS

DEVICE: (Mandatory) Specifies the HWX device that generated the EVENT message. The following Dname values are valid:

ADC - The A to D Converter with a 0 - 3.3 volt analog input line.

TRIGGER - The digital trigger input line.

STATE1 - The binary state digital state1 input line. [1]

STATE2 - The binary state digital state2 input line.

NOTE 1: This device can apply to both state input lines as detailed in LINE parameter description of the STATE_TRIGGER_CONFIG command.

Example 13.7B:

<u>USER</u>		<u>iMODEM</u>
	<==	EVENT[0]:1,2008
OK[0]: device=ADC	==>	

13.7.3 FAULT: Report an HWX device problem

This API message is sent to the iMODEM host via the serial port to report a problem with an HWX device by the iMODEM. It is sent asynchronously and requires no acknowledgement.

USAGE - This message uses a previously allocated Session ID--SID.

SYNTAX

FAULT[<SID>]:<Device_ID>,<Code>

PARAMETERS

Device_ID: The code corresponding to the device that generated the message. It may have the following values:

- 1 - The ADC analog input line.
- 2 - The trigger input line.
- 3 - Digital state line 1. [1]
- 4 - Digital state line 2.

NOTE 1: This device can apply to both state input lines as detailed in LINE parameter description of the STATE_TRIGGER_CONFIG command.

Code: An error code corresponding to the device problem.

Example 13.7C:

USER

<==

iMODEM

FAULT[0]:3,46

14. MISCELLANEOUS TEST COMMANDS

The commands in this section are useful for verifying iMODEM functionality. However, they are not intended for automated use.

14.1 @QTM: Send Stored E-Mail

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

```
@QTM: PROFILE=<Prof_Name> PHONE=<Phone_No>
INTERNAL FILE=<File_Name>
FROM=<Source_Add> TO=<Dest_Add_List>
SUBJECT=<Subject_Msg>
```

PARAMETERS

Prof_Name: Optional - The name of an SMTP profile such smtp.att. If this parameter is not specified, default SMTP parameters are used.

Phone_No: Optional Analog iMODEM specific parameter. This is the local access number for the ISP.

INTERNAL: This is keyword parameter and is optional. If it is used, it indicates, the stored E-Mail message has internal headers, and the FROM, TO, and SUBJECT parameters can not be used.

File_Name: Mandatory - The name of a file containing the E-Mail message.

Source_Add: Optional - The source E-Mail address. The E-Mail address in the effective SMTP profile is used if this parameter is omitted.

Dest_Add_List: Optional - A list of at least 1 destination E-Mail addresses. The E-Mail addresses in the effective SMTP profile are used if this parameter is omitted. Addresses are separated by the ',' character without spaces and the format is as follows:

```
<Add-1>,<Add-2>, ...<Add-N>
```

Add is an E-Mail address and the following examples illustrate its use.

Example: Single E-Mail Address.

```
TO=dick_tracy@spy.com
```

Example: Multiple E-Mail Addresses.

```
TO=john@doe.net,jane@doe.net,boss@top.com
```

Subject_Msg: Optional - The E-Mail subject. A sentence describing the E-Mail message subject. It must be enclosed in quotes. For example: SUBJECT="This is the subject".

Example 14.1C: Send stored E-Mail

<u>USER</u>		<u>iMODEM</u>
@QTM: FILE=test	==>	
	<==	POSTED: Operation Started

<==

OK: E-Mail successfully sent

14.2 @QRX: Retrieve an E-Mail Message**14.3**

The iMODEM response to this command is different than the standard API command response and is as follows:

```
POSTED: <Message>
OK: <Message>
<Line-1>
<Line-2>
....
<Line-N>
<CTL-D>
END: <Message>
```

Line is a line of ASCII data.

CTL-D: The control D character--hex 4.

USAGE - This command can not be issued while other API commands are executing or the iMODEM is in Multi-mode.

SYNTAX

@QRX: PROFILE=<Prof_Name> MSG=<Msg_No> PHONE=<Phone_No>

PARAMETERS

Prof_Name: Optional - The name of an POP3 profile such pop.att. If this parameter is not specified, default POP3 parameters are used.

Msg_No: Mandatory - The number of a message stored on the POP server which must be greater than 0.

Phone_No: Optional Analog iMODEM specific parameter. This is the local access number for the ISP.

Example 14.2A: Attempt to retrieve message 1 from a POP3 server.

<u>USER</u>		<u>iMODEM</u>
@QRX: MSG=1	==>	
	<==	POSTED: Operation Started
	<==	OK: Connected to E-Mail \
		Retrieval Server.

Now start receiving the E-Mail message**First, the initial header lines**

<==	X-Envelope-To: \
	<paulr@rcom-software.com>
<==	Return-Path: \
	<paulr@ rcom-software.com>

Next, several more header lines**Then, the final header lines**

<==	MIME-Version: 1.0
<==	To: paulr@rcom-software.com
<==	Subject: Test Message
<==	Content-Type: text/plain; character ...
<==	Content-Transfer-Encoding: 7Bit

```
<==          Date: Fri, 09, Dec 2008 \
              15.32.56 -0500
<==          X-ELNK-AV: 0
<==          X-Scanned-By: MIMEDefang 2.37
<==          X-UIDL: GLW"!djm"08L!YgY!
<==          Status: U
```

Now, the message body

```
<==          This is an iModem
<==          CH2168 test message
<==          from Cermetek.
```

**Next, an invisible EOT--hex 4 or <CTL>D--is received
which marks the end of the message body**

```
<==
```

**Finally, the END message from the iMODEM indicates
that E-Mail reception is complete and succeeded**

```
<==          END: E-Mail Message retrieved.
```

APPENDIX A

iMODEM API ERROR CODES

Page 1 of 3

<u>Error Code</u>	<u>Verbose Message</u>
0	Command not valid in this transfer mode
1	Previous command not complete
2	Invalid channel specified
3	Command does not exist.
4	Incomplete command: Must supply more data or abort.
5	Input line too long
6	Internal reception error
7	No operation in progress on this channel
8	Corrupt header format
9	Timeout: Line not completed fast enough.
10	Bad Parameters.
11	No free channels.
12	Invalid channel type.
13	Invalid state for requested command
14	SMTP E-mail transmission failed.
15	Error during file operation.
16	Error during SMTP quit.
17	Error during SMTP message transmission
18	Can not create storage file.
19	Can not open/configure modem.
20	BUG: Modem already in use.
21	Can't open/configure modem for input
22	Corrupt Message Header Format
23	Invalid Application/API type.
24	Not Implemented.
25	Can't connect to POP3 server
26	POP3 Server connection release error.
27	POP3 Server connection timeout.
28	Bad message number.
29	Other applications started.
30	The file already exists.
31	Bad Application Profile.
32	Bad Internet Server.
33	Bad Link Profile
34	Can't add ISP Gateway Route
35	Dialup Connection to another ISP
36	IP/Network Connection Down
37	Socket Error
38	TCP Connection Failure
39	Serious ! System File not accessible
40	Dialup configuration problem.
41	Chat Script failure
42	Chat Script timed out
43	Busy Signal
44	No Carrier Detected
45	Active Voice Connection
46	No Dial Tone
47	Modem not responding. (Dead)

APPENDIX A

iMODEM API ERROR CODES

Page 2 of 3

<u>Error Code</u>	<u>Verbose Message</u>
48	Dialing Failed - No CONNECT
49	No ISP Login prompt
50	Invalid ISP Login ID response
51	ISP Password Authentication Failure
52	PPP Can Not Execute
53	Unexpected PPP Termination
54	PPP Protocol Problem
55	IP Did Not Start
56	Bad Login ID or Password
57	No messages found
58	Delete failed - Bad range ?
59	Bad argument format.
60	Bad POP3 List command response.
61	Invalid IPC message
62	Illegal POP3 state
63	OS operation failed
64	Connection refused by Server
65	Bad Dialup IP/PPP state
66	No memory
67	No authentication data
68	TLS Error
69	Non-specific SMTP protocol error
70	SMTP rejected mail
71	Bad authentication profile
72	Invalid message in this state.
73	Dialup IP Link Establishment in progress
74	Bad Syntax
75	Bad argument value.
76	Inconsistent dialup options.
77	Inconsistent E-Mail addresses
78	No destination address
79	Bad E-Mail header file
80	Bad E-Mail file
81	Fatal software error: Please Report
82	Illegal operation on existing file.
83	Bad writeable file parameter.
84	File not found.
85	IP Link is down
86	Socket Already Allocated
87	Socket Data operation timeout
88	Remote TCP disconnection
89	No socket allocated
90	Bad Profile Type
91	Peer File Problem.
92	Link Account Profile Configuration
93	Link ISP Profile Configuration
94	Secrets File Configuration
95	POP3 Stat Command Failure
96	Error spawning Modem Mode process.

APPENDIX A
iMODEM API ERROR CODES
Page 3 of 3

Error Code

97
98
99
100
101
102

Verbose Message

Bad exit from Modem Mode.
Bad attachment file.
Attachment format error.
Bad RF signal quality
No network registration
Retry - Corrupted

APPENDIX B
RELEASE CAUSES FOR EXTENDED ERROR REPORTING (+CEER)
 Page 1 of 2

<u>Error Number</u>	<u>Error Description</u>
-1,255	no error
1	unassigned number
3	no route to destination
6	channel unacceptable
8	operator determined barring
16	normal call clearing
17	user busy
18	no user responding
19	user alerting
21	call rejected
22	number changed
26	non selected user clearing
27	destination out of order
28	invalid number format
29	facility rejected
30	response to status enquiry"
31	normal
34	no channel available
38	network out of order
41	temporary failure
42	switching equipment congestion
43	access information discarded
44	requested channel unavailable
47	resources unavailable
49	quality of service unavailable
50	requested facility unsubscribed
55	incoming calls barred within CUG
57	bearer capability not authorized
58	bearer capability not available
63	service not available
65	bearer service not implemented
68	ACM reached ACM maximum
69	facility not implemented
70	only restricted bearer cap. avail.
79	service not implemented
81	invalid TI
87	no member of CUG
88	incompatible destination
91	invalid transit network selection
95	incorrect message
96	invalid mandatory information
97	message type not implemented
98	message type incompatible
99	info element not implemented
100	conditional info element error
101	message incompatible
102	recovery on time expiry
101	unsuccessful GPRS attach
102	unsuccessful PDP context activation

APPENDIX B
RELEASE CAUSES FOR EXTENDED ERROR REPORTING (+CEER)
Page 2 of 2

<u>Error Number</u>	<u>Error Description</u>
103	GPRS detach
104	GPRS PDP context deactivation
128	NoService
202	timer 303 expiry
203	establishment failure
210	no error
211	operation failed
212	timeout
213	bearer service not compatible

APPENDIX C

GPRS/GSM ERROR CODES

Page 1 of 3

Modem Numeric Response

Modem Verbose Response

GENERAL ERROR CODES

0	phone failure
1	no connection to phone
2	phone-adaptor link reserved
3	operation not allowed
4	operation not supported
5	PH-SIM PIN required
6	PH-FSIM PIN required
7	PH-FSIM PUK required
10	SIM not inserted
11	SIM PIN required
12	SIM PUK required
13	SIM failure
14	SIM busy
15	SIM wrong
16	incorrect password
17	SIM PIN2 required
18	SIM PUK2 required
20	memory full
21	invalid index
22	not found
23	memory
24	text string too long
25	invalid characters in text string
26	dial string too long
27	invalid characters in dial string
30	no network service
31	network timeout
32	network not allowed - emergency calls only
40	network personalization PIN required
41	network personalization PUK required
42	network subset personalization PIN required
43	network subset personalization PUK required
44	service provider personalization PIN required
44	service provider personalization PIN required
45	service provider personalization PUK required
46	corporate personalization PIN required
47	corporate personalization PUK required
100	unknown

GPRS ERROR CODES

25 (19)	LLC or SMDCP error
26 (1a)	Insufficient resources
27 (1b)	Unknown or missing access point name
28 (1c)	Unknown PDP address or PDP type

APPENDIX C
GPRS/GSM ERROR CODES
Page 2 of 2

Modem Numeric Response

29 (1d)
30 (1e)
31 (1f)
32 (20)
33 (21)
34 (22)
35 (23)
36 (24)
37 (25)
38 (26)
39 (27)
40 (28)
103
106
107
111
112
113
132
133
134
148
149
150

Modem Verbose Response

User authentication failed
Activation reject by GGSN
Activation rejected, unspecified
Service option not supported
Requested service option not subscribed
Service option temporarily out of order
NSAPI already used
Regular PDP context deactivation
QoS not accepted
Network Failure
Reactivation requested
Feature not supported
Illegal MS
Illegal ME
GPRS services not allowed
PLMN not allowed
Location area not allowed
Roaming not allowed in this location area
service option not supported
requested service option not subscribed
service option temporarily out of order
unspecified GPRS error
PDP authentication failure
invalid mobile class

APPENDIX D REFERENCES

Page 1 of 2

CERMETEK Documents

612-0404	Application Note # 404: Analog iMODEM Network Architecture Guide
612-0411	Application Note # 411: GPRS/RF iMODEM Network Architecture Guide
612-0405	Application Note # 405: Analog iMODEM Configuration Manual
612-0406	Application Note # 406: GPRS/RF iMODEM Configuration Manual
612-0407	Application Note # 407: Analog iMODEM ISP/PSTN Configuration Guide
612-0413	Application Note # 413: GPRS/RF iMODEM GPRS Network Configuration Guide
612-0409	Application Note # 409: iMODEM Profile Download Tool Guide
612-0401	Application Note # 401: iMODEM Sending and Creating E-Mail
612-0400	Application Note # 400: MODEM Using the ISOCKET API
612-0410	Application Note # 410: iMODEM Retrieving E-Mail
613-0004	iMODEM: Command Reference Manual
652-0257	CH2168 iMODEM Demonstration Instructions
652-0258	CH2168 iMODEM Quick Start Guide
652-0268	CH2168A GPRS/RF iMODEM Quick Start Guide

INTERNET RFCs

RFC 793: Transmission Control Protocol(TCP)
RFC 791: Internet Protocol(IP)
RFC 1334: PPP Authentication Protocols
RFC 1661: The Point to Point Protocol(PPP)
RFC 1662: PPP in HDLC like Framing
RFC 2821: Simple Mail Transfer Protocol(SMTP)
RFC 4422: Simple Authentication and Security Layer.(SASL)
RFC 4954: SMTP Service Extension for Authentication
RFC 1939: Post Office Protocol - Version 3
RFC 1994: PPP Challenge Handshake Authentication Protocol (CHAP)
RFC 1700: Assigned Numbers (i.e. IANA TCP/UDP port numbers.)
RFC 3232: RFC 1700 is Replaced by an On-line Database

3GPP/ETSI Standards

GSM 02.60: (ETSI TS 101 113) Digital cellular communications systems (Phase 2+); General Packet Radio Service(GPRS); Service Description; Stage 1

GSM 03.60: (ETSI TS 101 344) Digital cellular communications systems (Phase 2+); General Packet Radio Service(GPRS); Service Description; Stage 2

GSM 03.64: (ETSI TS 101 350) Digital cellular communications systems (Phase 2+); Overall description of the GPRS radio interface; Stage 2

GSM 04.08: (ETSI EN 300 940) Digital cellular communications systems(Phase 2+); Mobile radio Interface layer 3 specifications.

APPENDIX D REFERENCES

Page 2 of 2

GSM 04.60: (ETSI EN 301 349) Digital cellular communications systems (Phase 2+); General Packet Radio Service(GPRS); Mobile Station(MS) - Base Station System(BSS) interface; Radio Link Control/Medium Access Control(RLC/MAC) protocol.

GSM 04.64: (ETSI TS 101 351) Digital cellular communications systems (Phase 2+); General Packet Radio Service(GPRS); Mobile Station - Serving GPRS Support Node(MS-SGSN) Logical Link Control(LLC) layer specification.

GSM 07.07: (ETSI TS 100 916) AT command set for GSM Mobile Equipment (ME)

OTHER Standards

TIA/EIA-602-A: Data Transmission Systems and Equipment -
Serial Asynchronous Automatic Dialing and Control

ITU-T V.250: Serial asynchronous dialing and control.

ITU-T V Series: Modem Transmission Protocol Standards.

Cermetek reserves the right to make changes in specifications at any time and without notice. The information furnished by Cermetek in this publication is believed to be accurate and reliable. However, Cermetek assumes no responsibility for its use, or for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Cermetek.

Cermetek provides the user with a FCC (USA) and IC (Canada) Approved or Approvable device. However, Cermetek does not have influence over nor knowledge of the specific user application environment. Therefore, for FCC and IC Approved or Approvable devices, the user assumes all risk for maintaining compliance to registration.

Printed in U.S.A