

iChip Email Client

Theory of Operation

Version 1.01

Pub. No.: 19-1600-00

International:
Connect One Ltd.
20 Atir Yeda Street, Kfar Saba 44643, Israel
Tel: +972-9-766-0456 Fax: +972-9-766-0461
info@connectone.com www.connectone.com

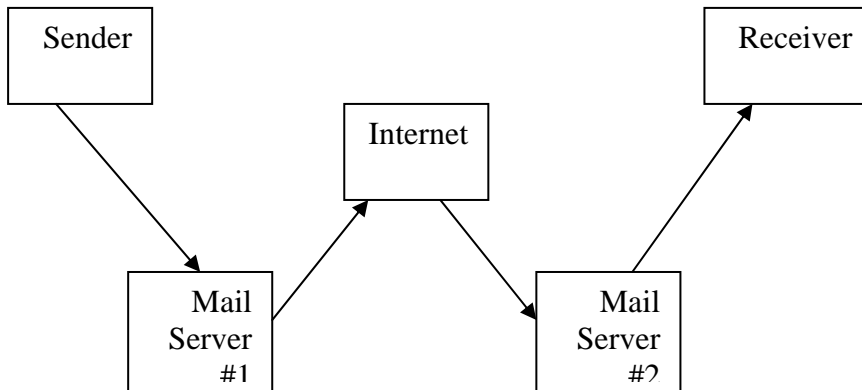
USA:
Connect One Semiconductors, Inc.
560 S. Winchester Blvd., Suite 500, San Jose, CA 95128
Tel: 408-572-5675 Fax: 408-572-5601
info@connectone.com www.connectone.com

Connect One
The Device Networking Authority

Introduction

E-mail (electronic mail) is the exchange of computer-stored messages by telecommunication. E-mail messages are usually encoded in ASCII text. However, you can also send non-text files, such as graphic images and sound files, as attachments sent in binary streams. E-mail was one of the first uses of the Internet and is still the most popular use. A large percentage of the total traffic over the Internet is E-mail.

A very simple diagram would show the process of sending and receiving E-mail as follows:



The sender gets online and logs into an Internet Service Provider, sending the E-mail to the Mail Server #1. From this point the process goes offline – the delivery of the E-mail depends on both Mail Servers, but most of all on the time in which the Receiver will download the E-mail message from his mailbox.

An E-mail message consists of header fields and, optionally, a body. The body is simply a sequence of lines containing ASCII characters. It is separated from the headers by a null line (i.e., a line with nothing preceding the CRLF).

Email header lines (To, From, Subject, Date etc.)
Empty line (containing only CRLF)
Email Body

Sending an E-mail

ASCII E-mail

The command syntax for sending an ASCII text E-mail is

AT+iEMA:<text lines>

<text lines> = Plain-text Email body. The Email body contains <CR/LF> terminated ASCII character strings. <text lines> must be terminated by a '.' (Dot character) in the 1st column of an otherwise empty line.

<text lines> ::= {<ASCII text line><CRLF> ...}<CRLF>.<CRLF>

When sending an ASCII text E-mail, iChip internally collects the E-mail contents, then sends them all at once, upon receiving the

<CRLF>.<CRLF> terminating string. Maximum size of <text lines> is limited to 18K, provided that no additional system resources are in use.

Once iChip receives the command's terminating string of <CRLF>.<CRLF>, it responds with I/OK. This means the command was received correctly, and iChip is now going to perform the necessary actions needed in order to send the E-mail.

Each E-mail contains a message header and body. iChip will construct the E-mail header using the following parameters:

Header Field	iChip Parameters	
Subject	SBJ	Contents of the Email subject field.
To	TOA[n]	Email Addressee <n>:01..50
	TO	Addressee Description/Name in Email header.
From	REA	Return Email Address
	FRM	Sender Description/Name in Email header.
Cc	CCn	Alternate Addressee <n>:1..4
Date		When the time has been retrieved from a Timer Server.

iChip will respond with I/DONE (or I/ONLINE) when the E-mail has been successfully sent to the SMTP server, or with I/ERROR if iChip was unable to send the mail. An I/DONE response is not a confirmation of the E-mail arriving at recipients' mailbox. It is a confirmation that the SMTP server correctly received the E-mail. It may take some time before the mail reaches its destination, depending on the Internet Service Providers (both the sender's and the receiver's), the SMTP server, and the recipients POP3 server. Some SMTP servers may be configured to send a message sent confirmation E-mail when the E-mail has been successfully delivered.

Binary E-mail

When you wish to send non-text files, or files larger than 18K, they should be sent as attachments. This is done using the MIME E-mail format. MIME (Multi-Purpose Internet Mail Extensions) is an extension of the original Internet E-mail protocol, that lets people use the protocol to exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kinds, as well as the ASCII text handled in the original protocol, the Simple Mail Transport Protocol (SMTP). Since the SMTP protocol requires that E-mail messages will contain only ASCII characters, a method of encoding binary data is required. iChip uses the base64 encoding system.

Binary email messages are sent via iChip using one or more AT+iEMB commands. The message format is limited to message header, an optional body of text and a single attachment. An AT+iEMB command sequence must be terminated by the AT+iE* command, indicating the end of the binary Email message.

MIME message format

The following fields are added by iChip Family to the main message header:

X-Mailer: iChip <software version>

Message-ID: <Unique #>@iChip

Mime-Version: 1.0

Content-Type: multipart/mixed; boundary="CONE-iChip-<software version>"

If the host application includes a text body for the message, it will also contain the following lines in its header:

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

X-iCoverpage: Email

When no textual body contents are included – this section will be omitted.

The binary attachment section will follow, beginning with a MIME attachment header containing the following fields:

*Content-Type: <User defined media type>/<User defined media subtype>;
name=<User defined attachment filename>*

Content-Transfer-Encoding: base64

Where,

<media type> := "text" / "image" / "audio" / "video" / "application"

<media subtype> := <A publicly – defined extension token.>

*<filename> := <User defined name (including extension)> or
<Unique filename>*

- *<media type>* defaults to "application" when otherwise not defined.
- *<media subtype>* defaults to "octet-stream" when otherwise not defined.

Following the header, a base64-encoded data stream will include the entire binary data transferred to the iChip from the host.

How to send a MIME E-mail

When several consecutive AT+iEMB commands are used, the host must issue the commands with an inter-command delay, which does not violate the SMTP server's timeout constraints. Otherwise, the SMTP server will timeout and abort the session. Average SMTP servers allow for delays in the range of 30 to 120 seconds. Additional AT+i commands may be interlaced within a sequence of AT+iEMB commands, except for The following AT+i commands: AT+iEMA, AT+iRML, AT+iRMH, AT+iRMM, AT+iRFU, AT+iRLNK, AT+iBDRA and AT+iSNMD.

The iChip Family does not limit the size of the binary attachment, however, ISP's might have limitations. An Internet connection will be initiated immediately after the first AT+iEMB command, while the rest of the command is received. Once the connection to the SMTP server has been

established, the iChip Family will act as a pipeline, receiving binary info from the host, encoding it, and transmitting it to the Internet on the fly. Following the AT+iE* command, the email will be terminated and the Internet connection closed.

Flow Control

When issuing an AT+iEMB command to generate a binary Email, the host transfers a binary data stream to the iChip Family. At times, this stream may be very large. Once the iChip Family establishes a connection, it acts as a pipeline, transferring data received from the host to the Internet. However, the data rates at the host and Internet ends are not always balanced. This happens for several reasons:

1. While the iChip Family is logging on to the Internet and establishing a connection, the host proceeds to send its data stream to the iChip Family. During this time the iChip Family receives data from the host, but cannot send it out.
2. When sending MIME attachments the iChip Family encodes the binary data, using base64. This inflates binary data roughly in the order of 30%. Thus, more data needs to be transmitted than is received from the host.
3. When using a TCP/IP socket, the iChip Family may need to retransmit packets.
4. When a faster baud rate is used between host ↔ iChip than the baud rate used for the iChip ↔ modem communication.

The amount of buffer space available in the iChip Family to accommodate for this imbalance is limited. Therefore, a flow control scheme is required to regulate host ↔ iChip Family communications. The FLW iChip Family parameter is set to reflect the preferred flow control mode.

Software flow control

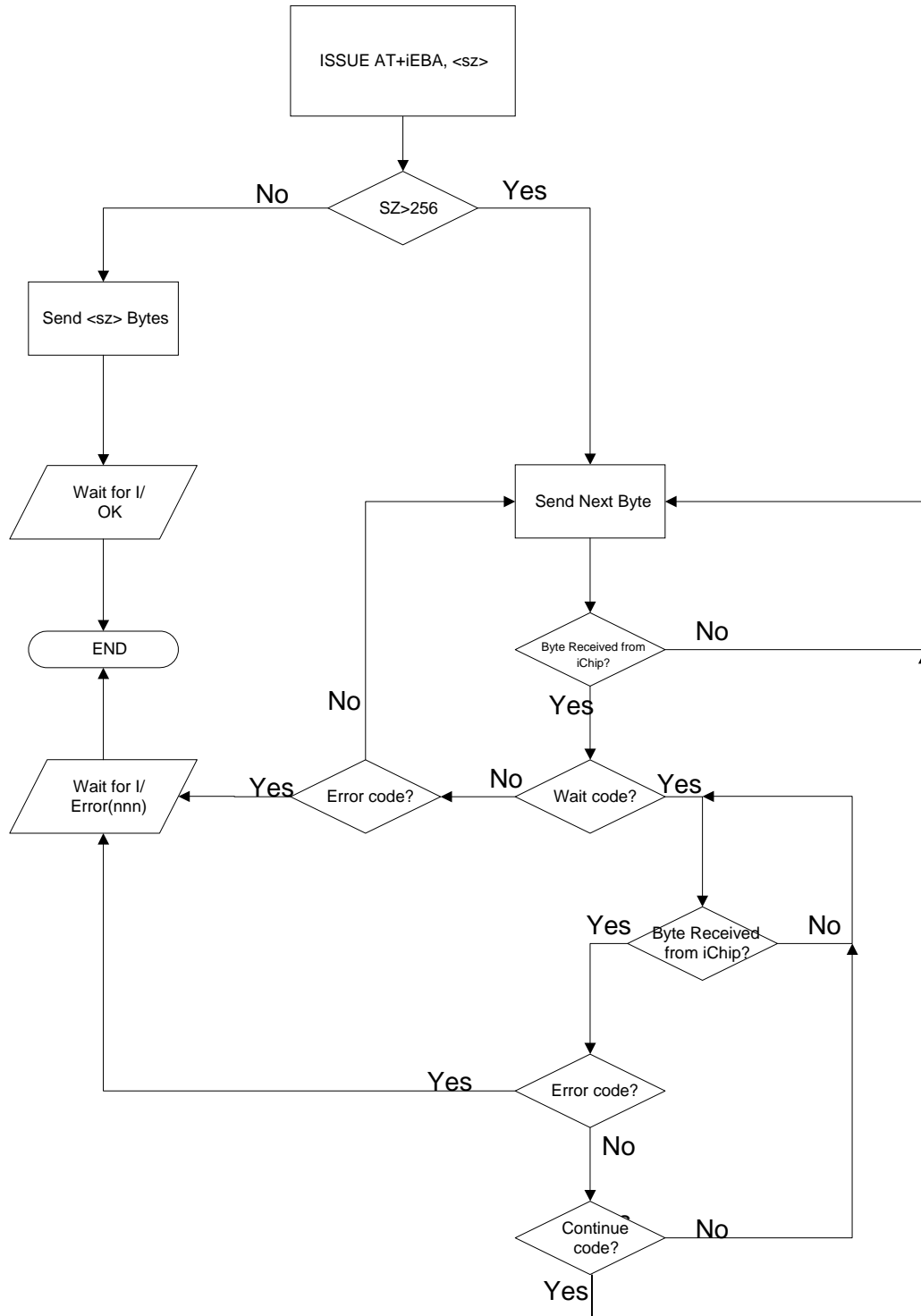
The software driven, flow control protocol, is defined as follows:

1. While the host is transferring the binary stream, following the +iEMB prefix, the iChip Family will issue a 'WAIT' control character when it needs to pause the host. The host application is required to monitor its serial receive line and pause the transmission when a 'WAIT' control character is received.
2. To resume the host transmission the iChip Family will issue a 'CONTINUE' control character. The host is required to monitor its receive line after being paused in anticipation of this control character. Once received the host may continue to transfer the data stream.
3. If an error occurs during the Internet session, while the host is transferring the data stream (or while paused), the iChip Family will issue an 'ERROR' control character, if some error occurred. Immediately after issuing this control character, the iChip Family will abort the Internet session and issue an 'I/ERROR (error number)' string. The host must cease transmitting the data stream when the 'ERROR' control character is received.

The control characters are defined as:

Control	ASCII Dec	ASCII Hex	Mnemonic
WAIT	22	0x16	SYN
CONTINUE	24	0x18	CAN
ERROR	5	0x5	ENQ

Software Flow Control Diagram in Binary Email Send



Hardware flow control

As an alternative to the software flow control method, which requires some software attention on behalf of the host, the iChip Family offers a hardware flow control mode.

This mode is selected by setting the iChip Family's FLW parameter Bit 0, using the AT+iFLW command. Note that to set FLW Bit 0, the ~CTSH signal needs to be LOW (Enabled), otherwise iChip will return I/ERROR(063). This convention safe-guards the iChip Family from lockup, which may arise if FLW Bit 0 is set while the ~CTSH signal is constantly HIGH.

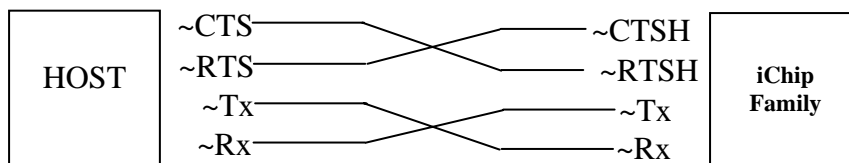
For hardware flow control to operate properly, the ~CTS and ~RTS signals between the host and iChip Family UARTS must be inter-connected.

The iChip family ~CTSH and ~RTSH may be shorted to circumvent H/W flow control.

Under this mode the iChip Family will assume that the host transmission may be paused by de-asserting the ~CTS signal. The host must adhere to this convention. Most UARTs support hardware flow control, however if this is not the case, the iChip Family's ~CTS signal should be monitored by the host software on a general-purpose I/O.

The host may also pause the iChip by de-asserting its ~CTS signal. However, this will force the iChip to 'stall' the mail server. This may only be done for limited periods of time. Otherwise, the POP3 server will timeout and abort the session. Average POP3 servers allow for delays in the range of 30 to 120 seconds, after which the mail server will terminate the session.

If a transmission error occurs during processing of a send command, the iChip shall accept all remaining characters pertaining to the current command (as specified by the <sz> parameter) before returning the relevant I/ERROR response.



Minimum Hardware Flow Control Connections

iChip parameter setup

a. Basic setup to establish a connection

1. iModem

Parameter	Type	Description
ISPn	Phone #	ISP's access phone number. <n>:1..2
ATH	Byte	Use CHAP(2), PAP(1) or Script(0) authentication.
USRN	String	ISP Connection User name
PWD	String	ISP Connection Password
RDL	Byte	Number of Redial tries.
RTO	Byte	Timeout before redialing [seconds]

2. iLan

Parameter	Type	Description
MACA	String	MAC address assigned to the iChip LAN
DIP	Default IP address	Default IP address stored in the iChip Family's nonvolatile memory.
IPA	IP address	IP address assigned to the iChip LAN.
IPG	IP address	IP gateway address assigned to the iChip LAN.
SNET	IP address	Subnet address assigned to the iChip LAN
HSTN	String	iChip LAN's Network Host Name. Will be registered in DNS through DHCP Server.

b. E-mail send parameters

1. General parameters

Parameter	Type	Description
DNSn	IP address	Domain Name Server IP address. <n>:1..2
SMTP	String	SMTP server name.
SBJ	String	Contents of the Email subject field.
TOA [n]	String	Email Addressee.
TO	String	Addressee Description/Name in Email header.
REA	String	Return Email Address
FRM	String	Sender Description/Name in Email header.
CCn	String	Alternate Addressee (CC: field) <n>:1..4

2. MIME parameters

Parameter	Type	Description
-----------	------	-------------

BDY	Text lines	Textual body contents for MIME encapsulated Email messages.
MT	Byte	Media Type: 0: Text; 1: Image ; 2: Audio ; 3: Video ; 4: application
MST	String	Media Subtype String. For a list see Appendix A.
FN	String	Attachment File Name (inc. extension). If a file name is not defined, the iChip will generate a unique filename.

Receiving E-mail

iChip offers 3 different commands to assist when receiving E-mail.

AT+iRML (Receive Mail List) and AT+iRMH (Receive Mail Header) provide information about E-mails waiting in the mailbox, while AT+iRMM (Receive Mail Message) receives them.

Retrieve Mail List (AT+iRML)

iChip responds with a list of qualifying Email message descriptors, separated by <CR/LF>. An Email message descriptor is composed of 5 <TAB> separated fields: <i><TAB><sz><TAB><date><TAB><subject string><TAB><type/subtype><CR/LF>

Where,

- <i> - Email message index in mailbox
- <sz> - Email message size in bytes
- <date> - Email message date (for the date field format refer to RFC822).
- <subject string> - Email message subject string (limited to 128 bytes).
- <type/subtype> - MIME Content-Type. The literal 'NONE' is used for non-MIME Email messages.

The message index <i> is provided by the mail server, and may be used to retrieve a specific mail message from the mailbox, using the command "AT+iRMM:i".

Retrieve Mail Headers (RMH)

To AT+iRMH iChip responds with all header lines of all qualifying E-mail messages. Header lines are returned "as-is". A line containing solely a '.' (period) in column 1 acts as a separator between the header lines of each E-mail. Header field syntax is described in RFC822 and RFC2045.

Retrieve Mail Message(s) (RMM)

iChip will retrieve the mail contents in one of two modes:

1. All Email contents shall be returned as-is. The Email's headers followed by the Email's body will be retrieved. MIME encapsulated Email messages shall be retrieved without BASE64 decoding.

It is assumed that when the 'XFH' (transfer Email headers) parameter is set to 'YES', the host processor attends to all Email field parsing and contents decoding.

2. Only the E-mail body (contents) shall be retrieved. If the Email message contains a MIME encapsulated attachment encoded in BASE64, the iChip shall perform the decoding and transfer pure binary data to the host. Binary attachments encoded in a scheme other than BASE64 shall be returned as-is. This mode will be used when 'XFH' parameter is set to 'NO'. At this stage messages encapsulated within messages are not supported – their contents will be transferred to the host 'as-is' and not be parsed. In this mode iChip will add lines describing the type of data to be sent to the host, using an information line beginning with 'I/PART', of the following format:

```
I/PART – <media type><TAB><media subtype><TAB><filename><TAB> <encoding method><CR/LF>
```

A line of this format will be sent before each part of the E-mail, describing the part about to follow.

For plain text E-mails (without MIME attachments):

```
<media type> = text
<media subtype> = plain
<filename> = '' (empty)
<encoding method> = quoted-printable
```

For E-mails containing MIME attachments these fields will be filled according to the E-mail contents.

At the end of each E-mail part, iChip will send I/EOP.

At the end of each E-mail message, iChip will send 'I/EOM'.

Required parameters

Parameter	Type	Description
LVS	Byte	Leave on Server: 1(YES – messages will not be deleted from mailbox). 0(NO – the messages will be deleted from the mailbox after being successfully downloaded)
DNSn[p]	IP address	Domain Name Server IP address. <n>:1..2
POP3[p]	String	POP3 server name
MBX	String	Mailbox User name
MPWD	String	Mailbox Password

Filters

iChip offers a powerful tool when working with E-mail – the filter string (FLS). The filter is an ASCII string which qualifies an Email message to be listed or retrieved by the iChip. This string must exist in the Email header for the message to qualify. If the string does not exist, the message will be ignored.

This enables a user to:

- Only look at and download E-mail messages relevant for a specific system (for instance, ignore spam mail).

- b. Use a single mailbox for several systems – each can define a filter for it's relevant messages.

Email error management

58	<i>Error when trying to establish SMTP</i>
59	<i>Error when trying to establish POP3</i>
60	<i>Single session body for MIME exceeds the maximum allowed</i>
61	<i>Internal memory failure</i>
76	<i>Unable to locate POP3 server</i>
77	<i>POP3 server timed out</i>
78	<i>Access denied to POP3 server</i>
79	<i>POP3 failed</i>
80	<i>No suitable message in mailbox</i>
81	<i>Unable to locate SMTP server</i>
82	<i>SMTP server timed out</i>
83	<i>SMTP failed</i>
105	<i>No POP3 defined</i>
106	<i>No MBX (mailbox) defined</i>
107	<i>No MPWD (mailbox password) defined</i>
108	<i>No TOA (addressee) defined</i>
109	<i>No REA (return address) defined</i>
110	<i>No SMTP defined</i>

Possible error codes to be returned in response to:

1. AT+iEMA / AT+iEMB:

81: SMTP server not found. Possible causes: The SMTP server name is a symbolic name but the iChip DNS settings are bad or the defined DNS servers are down and the SMTP server IP address cannot be resolved.

82: Timeout when connecting to SMTP server. iChip attempted to but was not able to open a socket connection to the SMTP server. Possible causes: The SMTP server name/IP address is incorrect, such a server does not exists, iChip requires the use of a gateway to connect to the SMTP server, but the gateway settings are incorrect or the gateway server is currently down.

83: Failed to send data on the SMTP socket. The server closed the socket or connection between iChip and the SMTP server was lost.

2. AT+iRMM / AT+iRML / AT+iRMH:

76: POP3 server not found. Possible causes: The POP3 server name is a symbolic name but the iChip DNS settings are bad or the defined DNS servers are down and the POP3 server IP address cannot be resolved.

77: Timeout when connecting to POP3 server, or while waiting for a reply. iChip attempted to but was not able to open a socket connection to the POP3 server. Possible causes: The POP3 server name/IP address is incorrect, such a server does not exists, iChip requires the use of a gateway to connect to the

POP3 server, but the gateway settings are incorrect or the gateway server is currently down.

79: Failed to send data on the POP3 socket. The server closed the socket or connection between iChip and the POP3 server was lost.

Examples

1. *Send an ASCII E-mail:*

- Step 1: Configure iChips email send related parameters:
SMTP, DNS, TOA, TO, REA, FRM
- Step 2: Send the AT+iEMA: command to iChip.
- Step 3: Send the text lines to iChip. Finish with
<CRLF>.<CRLF>. iChip should respond with
I/OK.
- Step 4: Wait for iChips' I/DONE (or I/ONLINE) response
to indicate the session is complete.

2. *Send an E-mail with a binary attachment:*

- Step 1: Configure iChips' email send related parameters:
SMTP, DNS, TOA, TO, REA, FRM
- Step 2: Configure binary send related parameters:
FN, MT, MST, BDY
- Step 3: Send the AT+iEMB: command to iChip.
- Step 4: Start sending the binary data. If one of the flow
control methods is used (software or hardware) the
data may all be sent in a single command
(AT+iEMB:25,788,<25,788 bytes of data>). When
no flow control method has been implemented, the
binary data must be sent in packets no larger than
256 bytes (AT+iEMB:128,<128 bytes of data>).
- Step 5: Wait for iChip's I/OK response to indicate all bytes
of data have been
- Step 6: Repeat Step 4 until no more data to send
- Step 7: Terminate the session with AT+iE*.

3. *Receive an ASCII E-mail:*

- When XFH is set to 0 (E-mail headers not to be retrieved) –
- Step 1: Configure iChips' E-mail retrieve related
parameters:
POP3, MBX, MPWD
 - Step 2: Send the AT+iRMM command to iChip. Wait for
the I/OK response.
 - Step 3: Expect "I/PART - text plain quoted-printable".
 - Step 4: Collect the data arriving into a file. While no I/EOP
message within the data.
 - Step 5: If I/EOM is received to indicate retrieval of the E-
mail message is complete, go to Step 3 to expect
another E-mail.

Step 6: I/DONE or I/ONLINE will indicate the session is complete.

When XFH is set to 1 (E-mail to be retrieved 'as is') –

Step 1: Configure iChips' E-mail retrieve related parameters:

POP3, MBX, MPWD

Step 2: Send the AT+iRMM command to iChip. Wait for the I/OK response.

Step 3: Expect I/RCV – Indicating an E-mail will now be retrieved.

Step 4: Collect the E-mail contents, while no "I/EOM – End of Message" has been received.

Step 5: If I/EOM is received to indicate retrieval of the E-mail message is complete, go to Step 3 to expect another E-mail.

Step 6: I/DONE or I/ONLINE will indicate the session is complete.

4. *Receive an E-mail with a binary attachment:*

When XFH is set to 0 (E-mail headers not to be retrieved) –

Step 1: Configure iChips' E-mail retrieve related parameters:

POP3, MBX, MPWD

Step 2: Send the AT+iRMM command to iChip. Wait for the I/OK response.

Step 3: Expect I/PART – the line containing information about the attachment about to be retrieved. For example:

```
"I/PART application      octet-stream  west.jpg
base64"
```

Step 4: Collect the binary data arriving into a file. While no I/EOP message within the data.

Step 5: If I/EOM is received to indicate retrieval of the E-mail message is complete, go to Step 3 to expect another E-mail.

Step 6: I/DONE or I/ONLINE will indicate the session is complete.

When XFH is set to 1 (E-mail to be retrieved 'as is') –

Step 1: Configure iChips' E-mail retrieve related parameters:

POP3, MBX, MPWD

Step 2: Send the AT+iRMM command to iChip. Wait for the I/OK response.

Step 3: Expect I/RCV – Indicating an E-mail will now be retrieved.

- Step 4: Collect the E-mail contents, while no “I/EOM – End of Message” has been received.
- Step 5: If I/EOM is received to indicate retrieval of the E-mail message is complete, go to Step 3 to expect another E-mail.
- Step 6: I/DONE or I/ONLINE will indicate the session is complete.