Application Note #146:

CH2056 V.80 Support Details Using V.80 for Synchronous Data Communication

SUMMARY

The CH2056 modem supports either a serial or parallel asynchronous host interface. This means that data sent to the modem must include a start bit, 8 data bits (sometimes made up of 7 data bits plus a single parity bit), and 1 or more stop bits. While this nicely supports most communications applications it can present a problem if one wishes to communicate with a remote mainframe computer that only supports a synchronous data protocol such as HDLC or BiSync. To deal with this issue the Synchronous Access Mode portion of the ITU V.80 specification can be employed to allow a host computer to send synchronous information across a modem's asynchronous port.

The V.80 specification was initially created to support applications like Video Conferencing which use HDLC formatted data to communicate compressed voice, video, and control data. Outside of this application however, V.80 can be used by a host computer as a bridge over the modem's asynchronous port…. so it can send and receive synchronous data.

This note is intended to provide the communications engineer with an overview of how V.80 can be used to support HDLC based synchronous data communications. It is not intended to provide a full description of the V.80 specification. With this in mind, it is recommended that the interested reader also acquire a copy of the ITU V.80 specification (see other resources on page 11).

USING V.80'S SYNCHRONOUS ACCESS MODE

V.80's Synchronous Access Mode allow the host computer to communicate synchronous data across the modem's Asynchronous port by defining many in-band control and status sequences. These in-band sequences allow the host computer to communicate special commands to the modem such as "Start an HDLC frame", "Send an Abort sequence," or "End an HDLC frame." These commands allow the host computer to control the HDLC synchronous protocol across the modem's asynchronous port.

The V.80 embedded commands or sequences are made up of an initial "alert" character that indicates a V.80 sequence will follow. This character is represented by the symbol **<EM>** and is the actually the ASCII Control-Y or hex 0x19 character. The character that follows **<EM>** indicates the nature of the command. For example, once a modem connection has been established the host computer can send the modem an asynchronous **<EM><Flag>** sequence (**<Flag>** is an ASCII 0xB1 code) which the V.80 enabled modem will realize is a command to enter a synchronous HDLC framed communication mode. The modem will interpret this sequence but will not send the **<EM><Flag>** on to the remote mode. If it is desired to send the "alert" character as data (and not as a V.80 alert character) then it should be sent twice in succession (i.e. **<EM><EM>**).

Likewise if the V.80 modem wants to communicate information about the synchronous connection to the host ***computer is precedes a status message*** with the same "alert" character. For example, if **<EM><MARK>** (**<MARK>** is an ASCII 0xB0 code) is received by the host, it indicates that the modem has detected an ABORT (all ones) sequence in HDLC framed mode. If the host receives **<EM><EM>**, it should translate this as the reception of a single character of ASCII value 0x19 and not as an embedded V.80 status message.

As indicated above, these sequences communicate vital information about the synchronous link. Later in the discussion, a more complete table of these command and status characters will be presented…. but the reader is advised to consult the V.80 specification for a complete listing.

NEED FOR FLOW CONTROL

Another V.80 example shown immediately below illustrates the data sent to the host computer when a short synchronous HDLC frame is received:

<EM><FLAG><01><02><03><04><05><EM><FLAG>

In this example, and HDLC frame received consists of ASCII hex characters 0x01 to 0x05. In this example frame, the first data character received is 0x01. This represents 8 bits of data from the telephone line where the LSB bit is a one and the 7 MSB bits are zero. For the modem to send this frame to the host computer it must add some asynchronous overhead bits to it (start and stop bits) to allow it to make it across the Asynchronous modem port. This means that the modem-to-host serial data rate must always a little faster than the modem-to-line rate. Precisely speaking… it must be at least 20% faster since for every synchronous 8 bits it receives it must send 10 bits to the host (8 data bits plus 1 start and 1 stop bit). If a modem connection rate of 33.6Kbps is achieved, it is recommended that the Serial communication rate used by the host to communicate to the modem be 57.6K bps or higher. This allows the modem to communicate the data received to the host faster than it is received and therefore avoids any received data bottle necks.

While this allows the modem to not worry about received data bottle necks it now potentially becomes a problem for the host computer. What if the host cannot receive the data continuously at the higher speed? The modem has a flow control buffer built in for this very reason. It allows the host computer to manage the flow of both received and transmitted data so that bottlenecks can be addressed and peak data rates can be managed.

Selecting Flow Control.

The CH2056 includes both In-Band or software flow control (sometimes called XON/XOFF) and Hardware (or RTS/CTS) flow control mechanisms selected by either the **AT&Kn** or **AT+IFC=n** commands. The default selection for the CH2056 and the most practical from of flow control for full duplex communications is Hardware Flow Control. More information on these modem AT Commands can be found in the Cermetek publication AT Commands and S Registers.

When Hardware flow control is selected the modem will only send received data to the host when the modem RTS control line in ON. If the host turns RTS OFF for a prolonged period the modem potentially will end up with a case where its flow control buffer is exceeded and data loss can occur. This is typically called an Over-run condition.

When the host wants to transmit data it must only do so when the modem indicates it can accept data into the transmit flow control buffer. To assure this the host must only send data to the modem when the modem has the CTS control line asserted ON.

When transmitting the modem by default will indicate to stop sending when it has 255 characters buffered up, and it will indicate to resume sending transmit characters when the number of buffered characters reaches 64 characters. The **AT+ITF** modem command can be used to alter these flow control buffer thresholds but the default settings are usually adequate for most applications. For additional information the reader should consult Cermetek publication AT Commands and S Registers.

TRANSPARENT OR FRAMED SUB-MODES

V.80's Synchronous Access Mode supports either Transparent or Framed Sub-Modes. Transparent operation allows the host computer to fully define the synchronous line protocol used. Anything the computer sends to the modem will be transmitted synchronously to the line and vise-versa. In the Framed mode however, the modem provides additional support for HDLC framed communication. It will generate idle flags, and optionally send and/or check received CRC (16 or 32 bit) error-check codes. It will also generate and detect abort codes and provide Zero insertion and deletion into the data stream to assure proper frame formatting. Since it is the intent of this note to show how HDLC formatted synchronous communications is supported only the Framed Sub Mode will be discussed further.

SETTING UP A SYNCHRONOUS CONNECTION

The **AT+ES** and **AT+ESA** commands are given to the modem before establishing a data to enable and configure V.80 synchronous access mode.

To get a full description of these commands the reader is encouraged to consult the V.80 specification but the following summarizes the definition of each command and how they are used for typical Synchronous Applications.

Enabling V.80 Synchronous Access Mode --- AT+ES Command Description.

Synchronous Access Mode is enabled by the **AT+ES** command. To support either transparent or Framed Sub (HDLC) mode the following AT command should be issued before a connection:

AT+ES=6,,8<cr>

This command advises the modem to enter V.80 Synchronous Access Mode after a connection is established…. either in answer and in originate mode. Notice that the middle parameter that controls error control fallback selection is not specified. This is because in Synchronous Access Mode the host computer is responsible for error handling since it is managing the HDLC frames.

The full definition of the **AT+ES** command is listed immediately below:

Parameter

+ES=[<orig_rqst>[,<orig_fbk>[,<ans_fbk>]]]

Description

This extended-format compound parameter is used to control the manner of operation of the V.42 protocol in the DCE (if present). It accepts three numeric sub-parameters:

- **<orig_rqst>**, which specifies the initial requested mode of operation when the DCE is operating as the originator.
- **<orig_fbk>**, which specifies the acceptable fallback mode of operation when the DCE is operating as the originator.
- **<ans_fbk>**, which specifies the acceptable fallback mode of operation when the DCE is operating as the answerer.

Defined Values

TABLE 1: Error Control Operation Sub-parameters.

<orig_rqst>	**Description**
0	Direct mode
1	Initiate call with Buffered mode only
2	Initiate V.42 without Detection Phase. If V.8 is in use, this is a request to disable V.42 Detection Phase
3	Initiate V.42 with Detection Phase
4	Initiate Alternative Protocol
5	Initiate Synchronous Mode when connection is completed, immediately after the entire CONNECT result code is delivered. V.24 circuits 113 and 115 are activated when Data State is entered.
6	Initiate Synchronous Access Mode when connection is completed, and Data State is entered.
7	Initiate Frame Tunneling Mode when connection is completed, and Data State is entered.
<orig_fbk>	**Description**
0	Error control optional (either LAPM or Alternative acceptable); if error control not established, maintain DTE-DCE data rate and use V.14 buffered mode with flow control during non-error-control operation
1	Error control optional (either LAPM or Alternative acceptable); if error control not established, change DTE-DCE data rate to match line rate and use Direct mode
2	Error control required (either LAPM or Alternative acceptable); if error control not established, disconnect
3	Error control required (only LAPM acceptable); if error control not established, disconnect
4	Error control required (only Alternative protocol acceptable); if error control not established, disconnect

TABLE 1: Error Control Operation Sub-parameters (continuation).

<ans_fbk>	Description
0	Direct mode
1	Error control disabled, use Buffered mode
2	Error control optional (either LAPM or Alternative acceptable); if error control not established, maintain DTE-DCE data rate and use local buffering and flow control during non-error-control operation
3	Error control optional (either LAPM or Alternative acceptable); if error control not established, change DTE-DCE data rate to match line rate and use Direct mode
4	Error control required (either LAPM or Alternative acceptable); if error control not established, disconnect
5	Error control required (only LAPM acceptable); if error control not established, disconnect
6	Error control required (only Alternative protocol acceptable); if error control not established, disconnect
7	Initiate Synchronous Mode when connection is completed, immediately after the entire CONNECT result code is delivered. V.24 circuits 113 and 115 are activated when Data State is entered.
8	Initiate Synchronous Access Mode when connection is completed, and Data State is entered.
9	Initiate Frame Tunneling Mode when connection is completed, and Data State is entered.

Configuring V.80 Synchronous Access Mode --- AT+ESA Command Description.

The **AT+ESA** command is used to further configure the operation of V.80 Synchronous Access Mode once it is enabled by the **AT+ES** command.

A full description of the **AT+ESA** command is listed below but the command that is typically used is as follows:

AT+ESA = 0,0,0,,X,0,255,<cr>

Where X defines what type of CRC processing is desired. If X is 0 then no transmit CRC bits are added to the message and CRC bits are not checked on reception. If X= 1, or 2 then the modem will respectively either transmit a 16 or 32 bit CRC polynomial an the end of the HDLC data message and will check for a proper CRC polynomial on reception.

AT+ESA Command Parameters.

+ESA=[<trans_idle>[,<framed_idle>[,<framed_un_ov>[,<hd_auto>[,<crc_type>[,<nrzi_en>[,<syn1> [,<syn2>]]]]]]]

Description

This extended-format compound parameter is used to control the manner of operation of the Synchronous Access Mode in the DCE (if present). It accepts six numeric sub-parameters:

- **<trans_idle>**, which specifies the bit sequence transmitted by the DCE when a transmit data buffer underrun condition occurs, while operating in Transparent sub-Mode.
- **<framed_idle>**, which specifies the bit sequence transmitted by the DCE when a transmit data buffer underrun condition occurs immediately after a flag, while operating in Framed sub-Mode.
- **<framed_un_ov>**, which specifies the actions undertaken by the DCE when a transmit data buffer underrun or overrun condition occurs immediately after a non-flag octet, while operating in Framed sub-Mode.
- **<hd_auto>**, which specifies whether or not, in V.34 half duplex operation, additional procedures besides those specified in § 12/V.34 shall be performed by the DCE when switching from primary channel to secondary channel operation, and vice versa.

- **<crc_type>**, which specifies the CRC polynomial used while operating in Framed sub-Mode.
- **<nrzi_en>**, which specifies if Non Return to Zero Inverted (NRZI) encoding is to be used by the DCE for transmit and receive data.
- **<syn1>, <syn2>**, which specifies the octet value(s) to be used while performing character-oriented framing.

Defined Values

TABLE 2: Synchronous Access Mode Operation Sub-parameters.

<trans_idle>	**Description**
0	In Transparent sub-Mode, DCE transmits 8 bit SYN sequence on idle. DCE receiver does not hunt for synchronization sequence
1	In Transparent sub-Mode, DCE transmits 8 bit SYN sequence on idle. DCE receiver hunts for 8 bit SYN sequence
2	In Transparent sub-Mode, DCE transmits 16 bit SYN sequence on idle. DCE receiver hunts for 16 bit SYN sequence
<framed_idle>	**Description**
0	In Framed sub-Mode, DCE transmits HDLC flags on idle
1	In Framed sub-Mode, DCE transmits marks (ones) on idle
<framed_un_ov>	description
0	In Framed sub-Mode, DCE transmits abort on underrun in middle of frame
1	In Framed sub-Mode, DCE transmits a flag on underrun in middle of frame, and notifies DTE of underrun or overrun.
<hd_auto>	**Description**
0	When switching between primary and secondary channel operation in V.34 half duplex, the DCE only executes those procedures defined in § 12/V.34/
1	When switching between primary and secondary channel operation in V.34 half duplex, the DCE executes additional procedures as described in § **Error! Reference source not found.** besides those defined in § 12/V.34
<crc_type>	**Description**
0	CRC generation and checking disabled
1	In Framed sub-Mode, the 16 bit CRC specified in § 8.1.1.6/V.42 is generated by the DCE in the transmit direction, and checked by the DCE in the receive direction
2	In Framed sub-Mode, the 32 bit CRC specified in § 8.1.1.6/V.42 is generated by the DCE in the transmit direction, and checked by the DCE in the receive direction
<nrzi_en>	**Description**
0	NZRI encoding and decoding disabled
1	NRZI encoding enabled in the DCE in the transmit direction, and NRZI decoding enabled in the DCE in the receive direction
<syn1>	**Description**
0-255	When **<trans_idle>=0**, specifies the 8 bit transmit idle sequence to be used by the DCE. When **<trans_idle>=1**, specifies the 8 bit synchronization sequence to be used by the DCE. When **<trans_idle>=2**, specifies first 8 bits of 16 bit synchronization sequence to be used by the DCE
<syn2>	**Description**
0-255	When **<trans_idle>=2**, specifies last 8 bits of 16 bit synchronization sequence to be used by the DCE

SETTING UP A TYPICAL HDLC SYNCHRONOUS CONNECTION

The following are typical AT command Sequences to set up an HDLC Synchronous Connection:

Originate Connection (*modem responses in italic*).

AT&F<cr> // initialize the modem to factory defaults
<cr><lf>OK<cr><lf> //modem responses with OK

AT+ES=6,,8<cr> // enable V.80 synchronous access mode
<cr><lf>OK<cr><lf> //modem responses with OK
AT+ESA=0,0,0,,1,0,255,<cr> // configure for 16 bit CRC generation and checking
<cr><lf>OK<cr><lf> //modem responses with OK

ATDT<number><cr> //Dial a connection
<cr><lf>CONNECT <DTE rate><cr><lf>
//connection established and
//Transparent Synchronous Access Mode entered

<EM><flag> //Enter HDLC mode
<EM><err> or <EM><flag> //modem receiver detects flags

Typical Answer Connection.

AT&F<cr> // initialize the modem to factory defaults
<cr><lf>OK<cr><lf> //modem responses with OK

AT+ES=6,,8<cr> // enable V.80 synchronous access mode
<cr><lf>OK<cr><lf> //modem responses with OK

AT+ESA=0,0,0,,1,0,255,<cr> // configure for 16 bit CRC generation and checking
<cr><lf>OK<cr><lf> //modem responses with OK

<cr><lf>RING<cr><lf> //wait for a incoming RING

ATA<cr> //
<cr><lf>CONNECT <DTE rate><cr><lf>
//connection established and
//Transparent Synchronous Access Mode entered

<EM><flag> //Enter HDLC mode
<EM><err> or <EM><flag> //modem receiver detects flags

TABLE OF V.80 SYNCHRONOUS ACCESS MODEM EMBEDDED COMMAND AND RESPONSES

The following Table lists all the embedded V.80 Synchronous Access modem sequences. The ones outlined in Italics below (**<EM><flag>**, **<EM><mark>**, and **<EM><err>**) are those that are typically used for sending and receiving HDLC frames. More details for each embedded sequence can be found in the V.80 specification.

TABLE 3: V.80 Synchronous Access Mode Sequences.

command/ indication pair symbol	hex codes	description, circuit 103	description, circuit 104	Transparent sub-mode	Framed sub-mode
		Character	**Character**		
<EM><t1>	5Ch	**Transparency**	**Transparency**	✓	✓
<EM><t2>	76h	transmit one	received one	✓	✓
<EM><t3>	A0h	19h pattern	19h pattern	✓	✓
<EM><t4>	A1h	transmit one	received one	✓	✓
<EM><t5>	5Dh	99h pattern	99h pattern	✓	✓
<EM><t6>	77h	transmit DC1	received DC1	✓	✓
<EM><t7>	A2h	transmit DC3	received DC3	✓	✓
<EM><t8>	A3h	transmit two	received two	✓	✓
<EM><t9>	A4h	19h patterns	19h patterns	✓	✓
<EM><t10>	A5h	transmit two	received two	✓	✓
<EM><t11>	A6h	99h patterns	99h patterns	✓	✓
<EM><t12>	A7h	transmit two	received two	✓	✓
<EM><t13>	A8h	DC1 patterns	DC1 patterns	✓	✓
<EM><t14>	A9h	transmit two	received two	✓	✓
<EM><t15>	AAh	DC3 patterns	DC3 patterns	✓	✓
<EM><t16>	ABh	transmit 19h, 99h	received 19h, 99h	✓	✓
<EM><t17>	ACh	transmit 19h, DC1	received 19h, DC1	✓	✓
<EM><t18>	ADh	transmit 19h, DC3	received 19h, DC3	✓	✓
<EM><t19>	AEh	transmit 99h, 19h	received 99h, 19h	✓	✓
<EM><t20>	AFh	transmit 99h, DC1	received 99h, DC1	✓	✓
		transmit 99h, DC3	received 99h, DC3		
		transmit DC1, 19h	received DC1, 19h		
		transmit DC1, 99h	received DC1, 99h		
		transmit DC1, DC3	received DC1, DC3		
		transmit DC3, 19h	received DC3, 19h		
		transmit DC3, 99h	received DC3, 99h		
		transmit DC3, DC1	received DC3, DC1		

TABLE 3: V.80 Synchronous Access Mode Sequences (continuation).

<EM><mark>	B0h	begin transparent sub-mode	HDLC Abort detected in Framed sub-Mode	✓	✓ (receive only)
<EM><flag>	B1h	Transmit a flag; enter Framed sub-Mode if currently in Transparent sub-Mode. If enabled, precede with FCS if this follows a non-flag octet sequence	Non-flag to flag transition detected. Preceding data was valid frame; FCS valid if CRC checking was enabled.		✓
<EM><err>	B2h	-transmit Abort-	Non-flag to flag transition detected. Preceding data was not a valid frame.		✓
<EM><hunt>	B3h	put receiver in hunt condition	-not applicable-	✓	✓
<EM><under> <EM><tover> <EM><rover> <EM><resume>	B4h B5h B6h B7h	-not applicable- -not applicable- -not applicable- resume after transmit	transmit data underrun underrun transmit data overrun	✓ ✓ ✓	✓ ✓ ✓ ✓
<EM><bnum>	B8h	underrun or overrun -not applicable-	receive data overrun -not applicable-	✓	✓
<EM><unum>	B9h	-not applicable-	the following octets, <octnum0><octnum1>, specifies the number of octets in the transmit data buffer. the following octets, <octnum0><octnum1>, specifies the number of discarded octets		✓

TABLE 3: V.80 Synchronous Access Mode Sequences (continuation).

<EM><eot> <EM><ecs> <EM><rrn> <EM><rtn> <EM><rate>	BAh BBh BCh BDh BEh	**duplex carrier control** terminate carrier, return to command state go to on-line command state Request rate reneg.(duplex) Request rate retrain (duplex) following octets, <tx><rx>, set max. tx and rx rates	**duplex carrier status** loss of carrier detected, return to command state -not applicable- indicate rate reneg.(duplex) indicate rate retrain (duplex) retrain/reneg. completed; following octets, <tx><rx>, indicate tx and rx rates	✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓
<EM><pri> <EM><ctl> <EM><rtnh> <EM><rtnc> <EM><rateh> <EM><eoth> <EM><ecs>	BCh BFh BDh C0h BEh BAh BBh	**V.34 HD carrier control** go to primary ch. operation go to control ch. operation initiate pri. channel retrain initiate ctl. channel retrain following octets, <maxp> <prefc>, set max. pri, rate and preferred ctl. ch. rate terminate carrier go to command state	**V.34 HD duplex carrier status** pri. ch. operation commenced; following octet, <prate>, indicates bit rate ctl. ch. operation commenced; following octets, <prate> <crate>, indicates bit rates indicate pri. channel retrain indicate ctl. channel retrain -not applicable- carrier termination detected -not applicable	✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓

HDLC DATA COMMUNICATION

After a connection has been establishing and the Synchronous Framed Sub Mode has been entered HDLC data transmission and reception can be started.

Next the transmission and reception of HDLC frames is described. Only typical frame operation is described. The reader is advised to consult V.80 for the processing of exception sequences caused by such events as Frame Aborts, and data under-run and over-runs.

Additionally, V.80 advises how the host computer is restrict the modems speed and initiate rate renegotiations and retrains. These sequences are beyond the scope of this note however.

HDLC Synchronous Transmission.

In HDLC communication the transmitted and received data is conditioned to assure that idle "flags" characters are not present in the framed data stream. To assure this, extra zeros are inserted in the transmit data when a continuous run sux 1's are found in the framed data and the inserted zeros are removed from the received data on the remote side. The modem performs this task, but for the modem to know when to do this it must first know when there is a start and end of a data frame.

The host computer signals the modem to initiate a frame with the **<EM><flag>** code, which signals the modem to transmit a flag. Additional flags may be explicitly specified by the host with additional **<EM><flag>** codes, or by an intentional transmit under run if the modem is configured for flag idle operation via the **AT+ESA** command. After one or more transmitted flags the modem will begin sending the HDLC data frame that it has received from the host computer. The host computer should end the transmission of the frame data by issuing another **<EM><flag>** command. If enabled by the **AT+ESA** command the modem will also append either a 16 or 32bit CRC polynomial on the end of the data message before a closing flag. The selection of 16 or 32 bit CRC polynomials is also made by the **AT+ESA** command.

An Example host computer transmission sequence is shown below that sends repeated frames (where the frames are 5 characters of an ascending count from 0x01 to 0x05):

<EM><flag> //optional if already in Framed Sub Mode
<01><02><03><04><05><EM><flag> //sends 01, 02, 03, 04, 05, **<CRC>**, then an ending flag
<01><02><03><04><05><EM><flag> /
<01><02><03><04><05><EM><flag>
<01><02><03><04><05><EM><flag>
<01><02><03><04><05><EM><flag>

During this example transmission the host must monitor the modem CTS line to ensure proper data flow control.

HDLC Synchronous Reception.

When Framed sub-Mode is initiated, the modem receiver will enter a "hunt" condition and search the received bit stream for HDLC flags. Until a valid flag character is detected, the modem will discard the received data and will not forward it to the host computer. Upon detecting a flag, the modem will send a **<EM><err>** code (where **<err>** is the ASCII hex code 0xB2) to the computer. Subsequent consecutive received flags are not forwarded to the host computer.

After a flag character is detected and the **<EM><err>** code has been forwarded to the computer, the modem will forward the first and subsequent non-flag characters to the DTE by removing the zero-inserted bits, appending start and stop asynchronous data bits, and EM-shielding certain embedded commands as defined in Table 3. Starting with the first non-flag character, the selected CRC polynomial is computed by the modem, if enabled by the **AT+ESA** command. If the non-flag octet sequence is terminated with a valid flag the modem will forward the **<EM><err>** code to the host computer if the enabled FCS was in error; otherwise, the modem will send the **<EM><flag>** code to the DTE. If the closing flag character is followed immediately by non-flag data, the flag will be considered the opening flag of the next frame.

An Example Host Computer Reception sequence is shown below (where the frames are the same frames transmitted above):

<EM><flag> or <EM><err> //indicates the modem is receiving initial flags
// if **<EM><err>** is received and CRC processing is
// enabled then the preceding frame was received in
// error.

// if **<EM><flag>** is received and CRC processing is
// enabled then the preceding frame was received
// without errors.

<01><02><03><04><05><EM><flag> //receives 01, 02, 03, 04, 05, **<CRC>**, then an ending flag, no errors

<01><02><03><04><05><EM><flag> //receives 01, 02, 03, 04, 05, **<CRC>**, then an ending flag, no errors
<01><02><03><04><04><EM><err> //receives a CRC error
<01><02><03><04><05><EM><flag> //receives 01, 02, 03, 04, 05, **<CRC>**, then an ending flag, no errors

While the host is receiving data from the modem is can momentarily interrupt the data from the modem by turning off the modem RTS line…but this is generally not recommended because data loss can occur if the data interruption is lengthy.

OTHER RESOURCES

The reader can purchase a copy of the ITU V.80 specification by visiting the ITU Electronic Bookstore at: http://www.itu.int/itudoc/itu-t/rec/v/v80.html

The CH2056 AT Command Reference Publication.

The AT Commands and S Registers publication is available from Cermetek via our web site at www.cermetek.com or by contacting our main office.

Printed in U.S.A

406 TASMAN DRIVE | SUNNYVALE CA 94089 | LOCAL: 408-752-5000 | TOLL FREE: 1-800-882-6271 | FAX: 408-752-5004
CERMETEK WEB SITE: http://www.cermetek.com | EMAIL: sales@cermetek.com
iModem Network WEB SITE: http://www.imodem.net/ or https://sunnyvale.imodem.net/