SIA Digital Communications Standard~~–~~ Receiver-to-Computer Interface Protocol ~~(Type 2) –~~ for Central Station Equipment Communications

SIA DC-07-2001.12~~04~~
(draft dated 10/14/2012)

Sponsor
Security Industry Association

Publication Order Number: xxxxx

FOREWORD

This standards document is published by the Security Industry Association (SIA) and was developed and adopted by a consensus of industry volunteers in accordance with SIA's standards development policies and procedures. It is intended to facilitate product compatibility and interchangeability, to reduce misunderstandings between manufacturers and purchasers, and to assist purchasers in obtaining the proper products to fulfill their particular needs.

The existence of this or any SIA standards document shall not prevent any SIA member or non-member from manufacturing, selling, or using products not conforming to this or any SIA standard. SIA standards are voluntary. SIA encourages the use of this document but will not take any action to ensure compliance with this or any other SIA Standard.

SIA assumes no responsibility for the use, application or misapplication of this document. Industry members using this document, particularly those having participated in its development and adoption, are considered by SIA to have waived any right they might otherwise have had to assert claims against SIA regarding the development process of this standard.

Although some SIA standards establish minimum performance requirements, they are intended neither to preclude additional product features or functions nor to act as a maximum performance limit. Any product the specifications of which meet the minimum requirements of a SIA standard shall be considered in compliance with that standard. Any product the specifications of which exceed the minimum requirements of a SIA standard shall also be considered in compliance with the standard, provided that such product specifications do not exceed any maximum requirements set by the standard. SIA standards are not intended to supersede any recommended procedures set by a manufacturer for its products.

SIA reserves the right to revise this document at any time. Because SIA policy requires that every standard be reviewed periodically and be either revised, reaffirmed, or withdrawn, users of this document are cautioned to obtain and use the most recent edition of this standard. Current information regarding the revision level or status of this or any other SIA standard may be obtained by contacting SIA.

Requests to modify this document are welcome at any time from any party, regardless of membership affiliation with SIA. Such requests, which must be in writing and sent to the address set forth below, must clearly identify the document and text subject to the proposed modification and should include a draft of proposed changes with supporting comments. Such requests will be considered in accordance with SIA's standards development policies and procedures.

Written requests for interpretations of a SIA standard will be considered in accordance with SIA's standards development policies and procedures. While it is the practice of SIA staff to process an interpretation request quickly, immediate responses may not be possible since it is often necessary for the appropriate standards subcommittee to review the request and develop an appropriate interpretation.

Requests to modify a standard, requests for interpretations of a standard, or any other comments are welcome and may be sent to:

Standards
Security Industry Association
8405 Colesville Road, Ste. 500
Silver Spring, MD 20910

~~**635 Slaters Lane, Suite 110**~~
~~**Alexandria, VA, 22314**~~

E-mail:
Standards@SIAOnline.org

ACKNOWLEDGEMENTS

Chairman of the SIA Standards Committee:
ADT ..... William N. Moody

Chairman of the SIA Computer Interface Standards Working Group:
Caddx Controls ..... John Jeffers

Contributing Members of the CIS Working Group:

Ademco ..... Rich Hinkson
Advanced Algorithms ..... Greg Spar
Bold Technologies ..... Kurt Emauelson
DS/Radionics ..... Rich Ader
Protection One ..... Noble Hetherington

This standard was approved by open industry vote on April 5, 2001

ADT ..... Dennis Yanek
DSC ..... David Clarke
DS/Radionics ..... Rich Ader
Interlogix ..... John Jeffers
SG Security Comm. ..... Stephan Frenette

Revision History

The following are changes made to this document, listed by revision.

APRIL 2001 BASELINE

Original Publication

October 2012

- non-substantive corrections to formatting
- update addresses
- eliminate "type 2" notation
- deleted undefined glossary term "packet frame"
- correct error in crcTable for calculation method 2

.

Table of Contents

Receiver-to-Computer Interface Protocol - ~~Type 2 –~~ for Central Station Equipment Communication

1 SCOPE & PURPOSE

1.1 Scope

This standard describes an interface format for communications between alarm signal receivers and automation computers. This standard is intended for use by equipment in security industry alarm monitoring centers, with possible uses in the areas of energy control and facilities monitoring and management.

This standard provides a common interface format for across-the-board compatibility of equipment, regardless of manufacturer, and provides for all the known communication needs between the computer and receiver.

This standard defines basic "codes" to identify commonly used dialer protocols used in alarm signal transmitters, as well as conditions in the central station equipment that require a technician or other manual attention.

Additions to these codes may be by application to SIA. Independent extensions to the codes will render a device non-compliant. Requests for additional codes, additional message fields, message interpretations or revisions to the standard,

should be submitted to SIA. The request will be distributed to the Subcommittee members for review and approval.

The standard is voluntary and self-enforcing. In the case of incompatibility, the problem should be resolved to the extent possible by manufacturer-to-manufacturer discussions. SIA's Intrusion ~~Digital Communications~~ Standards Subcommittee will act as an arbitration body if the problem cannot be otherwise resolved.

1.2 Purpose

This standard provides for the following objectives:

- Accommodate forwarding of messages received through standard security industry digital communications dialer protocols (SIA Format, SIA 2000, Ademco Contact ID) as well as all other common transmitter protocols
- Minimize the amount of processing required by the receiver (and allow the receivers to handle data from many transmitters)
- Minimize the transmission error rate
- Allow for a data message to have variable length and content

- Anticipate the future need for significant bi-directional data flow between the computer, receivers, and transmitters
- Be more predictable for automation software than the existing multiplicity of computer protocols and variations on standards
- Allow any reasonable implementation of SIA-CIS Type 1, to be easily differentiated from this protocol by automation software
- Allow adaptation to evolving lower level communications standards used in the computer industry

Various alternatives were considered in developing this standard, including modification of SIA's earlier Computer Interface Standard and variations on the currently available formats. Several of the formats reviewed had components of the structure required in a new format. However, none were applicable to the needs of the variety of systems currently in place.

There are currently several major computer- to-receiver interface formats on the market. The principal interfaces are listed alphabetically below. Copies of these proprietary interface formats are available from their manufacturers.

Table 1: Existing Computer Interface Formats

Acor CDR/P-250	Ademco 685
Applied Spectrum DWV-200	FBI CP220
ITI CS-4000	Morse SPC 5000
Morse V300	Osborne/Hoffman
Radionics D6000/D6500	Sescoa 3000
Silent Knight 9000	Silent Knight 9800
SIA-CIS Phase 1	SG MLR-2
SG-MLR-2000	VerSuS 90
And Many More	

These formats performed adequately in the service for which they were designed; however, the proliferation of proprietary formats has become a burden on the automation systems.

2 REFERENCE DOCUMENTS

2.1 Related Areas

SIA Standards:

- SIA DC-02 – Digital Communications Technical Report – Generic Overview of Security Industry Communicator Formats
- SIA DC-01 – Receiver-to-Computer Interface Protocol (Type 1)
- SIA DC-03 – "SIA Format" Protocol – for Alarm System Communications
- SIA DC-04 – SIA 2000 Protocol – for Alarm System Communications
- SIA DC-05 – Ademco Contact ID Protocol – for Alarm System Communications

Other Standards:

- TIA/EIA-232, Interface Between Data Terminal Equipment And Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange, Telecommunications Industry Association (Electronic Industries Alliance)
- IEEE 802.3 Ethernet, IEEE Standard for Information Technology, Institute for Electrical and Electronic Engineers
- Intel USB V1.1, Intel Corporation (www.intel.com)
- ANSI X3.4-1986 (R1997), Information Systems – Coded Character Sets – 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)

2.2 Other Supporting Resources

Internet RFC's - www.faqs.org/rfcs/

3 CONVENTIONS AND DEFINITIONS

3.1 Conventions

3.1.1 Units of Measurement

In accordance with SIA Policy, the units of measurements used throughout this publication are the units of the System International d' Unites (SI), commonly known as metric units. Equivalent English Units, enclosed in parenthesis, are also used in this publication. These equivalent English Units are approximate conversions and are provided for easy reference.

3.1.2 Tolerances

Unless otherwise specified, the tolerance for measurements specified within this standard shall be 10 percent (±10%).

3.1.3 Special Capitalization.

Alarm sequence events, alarm system commands and states, and digital communication codes transmitted by the control panel to the central station are capitalized within the text of this standard.

3.14 Nomenclature and Identification of Sections.

Sections within this standard are identified and referenced by the number preceding each section. Unless otherwise specified, references to a section refer to only that section and not to subsequent subsections within the section.

3.1.5 Binding Language

This standard uses the term "shall" to convey binding requirements.

The term "may" is used to convey features that are allowed but not required.

Terms such as "is", "are", "will", and others are used to convey statements of fact for advisory purposes only.

The annotation "NOTE:" also precedes advisory information.

Where this standard is silent on a feature, the feature is permitted so long as it is not in conflict with the requirements contained herein.

Appendices contain binding information.

3.2 Definitions

Account, The portion of a transmitted message which contains the information identifying a particular location; account number.

Acknowledgment, A signal sent from the RECEIVER to the TRANSMITTER indicating that the data has been received. A Positive Acknowledgment means data was received without any detected errors. A Negative Acknowledgment means data was received, but there were detected errors. Also used as a command to initiate transmission.

Area, A defined section of the protected system that can be armed and disarmed independently. Areas are numbered consecutively beginning with 1. A system must have at least one area*, area 1*.

ASCII, 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)

Automation, Software that takes the received data from the receiver. Typically this is a security software or fire dispatch software that is very interested in the state of the receiver and connected printers. The software runs on a Host Computer.

Baud Rate, A measurement of transmission speed. The number of signal elements per second based on the duration of the shortest element.

Bit, The smallest element of digital information. The value of a bit is either one or zero (true or false).

Block, A block of information consists of account or data information and includes the header and parity information.

Bypass, A zone state that ignores input changes regardless of the system arming state. A bypassed zone will NOT cause an alarm event.

Byte, A byte is a group of eight (8) bits. One alpha-numeric character can be represented by one byte.

Close, The act of arming a system.

Computer, Host Computer, The hardware embodiment of the automation software. Typically a industry standard server using Risc or x86 architecture which is redundant.

Data, That part of the transmitted data which refers to alarm point information or status of the sensors at a particular location.

Digital, Information in discrete or quantized form; not continuous.

Duplex Transmitter, One capable of receiving data from a central station.

DTE, Data Terminal Equipment

ETC, or Early To Close, is an event created by the arming of a system before a specified time.

ETO, or Early To Open, is an event created by the disarming of a system before a specified time.

FSK, Frequency Shift Keying, A signaling method for transmitting digital information which uses a discrete audio frequency for a logic one and a different frequency for a logic zero.

Full-Duplex, A mode of data transmission in which the data traffic in both directions can occur simultaneously.

Data Groups, A set of two or more blocks requiring only one acknowledgment, after the last block.

GMT, or Greenwich Mean Time, Is the current time in Greenwich, England. This is considered time zone 0. Other time zones are East or West of time zone 0. Time zones to the West are indicated by positive numbers, time zones to the East are indicated by negative numbers.

Half-Duplex, A mode of data transmission in which the data traffic travels in only one direction at a time, although the communication medium may allow full-duplex operation.

Handshake, A signal sent by the RECEIVER which indicates to the TRANSMITTER that a connection has been established.

Kiss Off, A term currently used in the industry for a positive acknowledgment.

LTC, or Late to Close, is an event created by the failure of a system to arm before a specified time.

LTO, or Late to Open, is an event created by the failure of a system to disarm before a specified time.

Mark Frequency, The discrete audio frequency of the FSK signal used as the information bit, and defined as a logic one (1).

Most Significant, The digit or bit which represents the highest value or weight.

Open, The act of disarming a system.

~~***Packet frame, TBD***~~

Parity Bit, A redundant bit added to a record to allow the RECEIVER to detect an odd number of bit errors in that record.

Parity Word, A record in which the data are redundant bits which allow the RECEIVER to detect an odd number of bit errors in each column of data bits in that block.

Receiver, The Digital Receiver located in the Central Station or Monitor Location.

Sender, The unit TRANSMITTER or RECEIVER currently in the process of transmitting information to the opposing unit.

Sounder, A device at the TRANSMITTER site used to signal an event such as fire. Sounders are number consecutively beginning with 1. A system is not required to have a sounder.

Recipient, The unit, TRANSMITTER or RECEIVER, currently in the process of receiving information from the opposing unit.

Reverse Channel, The transmission of data blocks in a direction opposite of the last block transfer.

Simplex Transmitter, One which is only capable of transmitting information to the central station RECEIVER.

Space Frequency, The discrete audio frequency of the FSK signal, which is the complement of the mark frequency, and defined as logic zero (0).

Subscriber, The person(s) at the TRANSMITTER site who operate and/or have access to the system.

Transmitter, The Digital Communicator located at the protected premise.

User, See Subscriber

USB, Universal Serial Bus

Warning, A device at the TRANSMITTER site used to alert the subscriber to an event such as power failure.

Weekly Minutes, Is the measure of time as minutes beginning with 0 for Sunday at 00:00, and ending with 10079 for Saturday at 23:59.

4 Mechanical and Electrical Layers

4.1 Mechanical Interface

The connection between the receiver and the computer will conform to the EIA standards for RS232 serial communications. The EIA standard for RS232 specifies connector style and pin locations. The connectors used are male 25 pin “DB25” style. The receiver and computer each have male connectors joined using a null modem cable having female connectors at both ends. Other acceptable connections are "AT" Standard DB-9 at RS-232 levels, RJ45 with 10BT and / or 100BT levels and Universal Serial Bus (USB) Type B receptacle on the receiver end.

4.2 Electrical Interface

4.2.1 RS-232 DB-25

The receiver and the computer will both act as DTE (Data Terminal Equipment) devices. The cable diagram demonstrates the connection. This is a "null modem" cable since it connects two DTE devices without using modems.

The EIA standard specifies that the data signals are marking (logic 1) when they are negative voltage, and spacing (logic 0) when positive. Control signals are ON when positive and OFF when negative. The standard further specifies that a positive voltage shall be greater than 3.0 volts, and a negative voltage shall be more negative than -3.0 volts. The signal is undefined in the region between +3.0 and -3.0 volts.

Table 2: Null Modem Cable

Receiver			Computer
1	**«–**	**Earth Ground**	**–» 1**
2	**–»**	**Transmit (to Computer)**	**–» 3**
3	**«–**	**Receive (from Computer)**	**«– 2**
Optional Signals (See Section 5.1.1)			
4, 5	**«–»**	**RTS /CTS To DCD**	«–»*8*
8	**«–»**	**DCD To RTS/CTS**	**«–»4, 5**
7	**«–**	**Signal Ground**	**–» 7**
6	**«–**	**Data Set Ready**	**«– 20**
20	**–»**	**Data Terminal Ready**	**–» 6**

4.2.2 RS-232 DB-9

Using EIA Standard RS-232 Signal levels a cable is used to connect both receiver and computer. The cable is commonly referred to as a laplink serial cable.

Table 3: Null 9 Pin Cable

Receiver			Computer
5	**«–**	**Earth Ground**	–» 5
3	**–»**	**Transmit (to Computer)**	**–» 2**
2	**«–**	**Receiver (from Computer)**	**«– 3**
Optional Signals (See Section 5.1.1)			
7, 8	**«–»**	**RTS /CTS To DCD**	**–» 1**
1	**«–**	**DCD To RTS/CTS**	«–» 7, 8
6	**«–**	**Data Set Ready**	**«– 4**
4	**–»**	**Data Terminal Ready**	**–» 6**

4.2.3 10BT/100BT Ethernet

Electrical levels for 10BT/100BT Ethernet are intended to be attached to the correct Hub via CAT 5 patch cords from both the receiver and the computer. (See IEEE 802.3 for construction and signal levels.

4.2.4 USB

In this configuration, the receiver is a "peripheral" and the computer is a "computer". According to the Universal Serial Bus USB standards the receiver is Downstream and has a type B receptacle. The computer is Upstream and has a type A receptacle. (Intel USB V1.1, Sept. 23 1998, Figure 6.2)

5 Transmission Layer

5.1 Signal Protocol

This standard defines bi-directional message flow between a single receiver and the computer, using data to acknowledge messages. This standard makes no requirements on the use of the handshaking signals provided by the RS232 standard, but allows common handshake techniques to be employed.

5.1.1 Signaling

It is anticipated that some computer systems will require the use of Data Set Ready and Data Terminal Ready to control the asynchronous inputs from the receiver units it services. If this method is used, both units should generally use Data Terminal Ready to signal a readiness to receive data, and an inactive DTR (received as Data Set Ready or DSR) should inhibit transmission.

As an alternative, asynchronous inputs can be controlled with XON and XOFF signals. An XOFF signal (ASCII 19) from one device to the other inhibits transmission. An XON (ASCII 17) signals a readiness to receive data.

In either case, a maximum time delay of 16 seconds is permitted by this standard. (See Section 5.4, Message Protocol for receiver and computer response to an expiration of the maximum time delay).

5.2 Physical Layer

5.2.1 RS-232

The signals will be transmitted asynchronously at 2400 BAUD as default (default shipment configuration), 10 total bits (1 start bit, 8 data bits, 1 stop bit, no parity). Binary ONE (1) shall be represented as a MARK or active signal. Binary ZERO (0) shall be represented as a SPACE or inactive signal, as provided by the EIA standards for RS232 serial communications. The receiver may be programmable from 300 baud to 115kbaud and shall be programmable from 300 to 9600 baud.

5.2.2 Ethernet

The signals will be transmitted according to RFC1122, 1123 for TCP/IP and will be at the appropriate rate for the network. Sockets will be opened on a user-selected port at a user selected IP address (Selection must be done at both receiver and Automation software). Please note that all receivers must be shipped defaulted to 0.0.0.0 to avoid crashing networks at startup. (0.0.0.0) is considered an invalid address.

5.2.3 USB

Signals will be transmitted according to USB standards~~V1.1~~. Accordingly, the receiver should emulate a networking device. However, unlike the USB standard (which defines the computer as the master), in this standard there is no master device.

6 Packet Protocol

6.1 Packet Flow

This protocol defines a full-duplex dialog. Messages flow between the receiver and computer. In most cases, the receiver transmits a message packet to the automation computer, and the computer answers with a response packet. The computer may also send data or commands to the receiver under certain circumstances. In all cases, the same packet structure is used.

To keep the protocol simple, the only initiating messages defined are Data Messages (forwarding alarm data or reporting other conditions), NULL Messages (a simple link test), and Data / Operation Requests (asking for data to be provided or an operation to be performed).

The only response messages defined are an ACK Response (successful acknowledgement), a NAK Response (a problem in the transmission), an RTN Response (data returned in response to a request or as a result of a requested operation), and a DUH (an indication that the request cannot be performed or understood).

Any of these messages may be sent by, either the receiver or the computer. Even though a particular device may not be capable of complying with a specific request, it must be able to recognize the message type and respond accordingly. (See Appendix A for more detailed descriptions of message types.)

Sequence Numbering for messages (including sub-sequence numbering for multi-packet transmissions) are preserved. That is, the sequence number of a response message is tied to the sequence number of the initiating message. Sequence numbering for events or messages is from 0001 to 9999. (Sequence number 0000 is reserved for Link Tests and NAK Responses.)

Although the normal communications mode will be “event reporting” from the receiver to the computer, occasionally the automation computer will need to send information to the receiver by initiating a message not tied to a prior message from the receiver. In this case, the computer uses its own unique sequence numbers for each packet.

A timestamp on a message is optional. If included it is placed at the end of the message and reflects when the message was queued for transmission by the sending device. The timestamp is in local time, as determined by the originator of the packet.

6.2 Routing

Messages sent by the receiver are always destined for the designated computer. This standard does not address multiple computer systems (unless the multi-processor system can operate using the sequence, receiver and line information is included in the standard format). Systems using TCP/IP can directly address computers, but this is not in the scope of this standard.

7 Message Packet Structure

The message format between the receiver and the computer provides two-way communication with sequenced message acknowledgment, and places a "wrapper" around the data, typically data received in transmitter formats.

When the receiver accepts a message from a transmitter, the receiver re-packages the message to conform to the structure of this standard.

The data characters are all in the ASCII printable range, decimal 32 to decimal 126. This allows the information being sent over the link between the receiver and the computer to be monitored on a printer or terminal.

7.1 Message Packet

Messages always begin with the flag character **<LF>**, followed by a Cyclic Redundancy Check **<CRC>** field, the message length, ID Token, sequence number (and segment number if applicable), receiver#, and line# strings. These items comprise the message header.

The beginning data delimiter character "**[**" follows the header. In most messages, the data is comprised of the **#** character, followed by the account number, then the field separator "**|**", and then the actual data. The actual data will vary and may have multiple data fields, depending on the native transmitter format in which it was received or the nature of information conveyed in certain special messages. The ending data delimiter "**]**" is used after the last data field. An optional Timestamp field may be included next, and the message ends with a carriage return **<CR>**.

Message packets from the receiver take the general form:

<LF> <CRC> <0LLL> <"ID"> <Sequence#**!**segment#|> <**R**Receiver#> <**L**Line#> […data…] <timestamp> <CR>

Descriptions of the fields that make up the packet are detailed below.

For the purpose of depicting attributes of this form and its various fields, the following conventions are used in this document to illustrate the form:

- The < and > characters and the spaces are not part of the actual form and are only provided to aid in depicting the form.
- The "**,**", **!**, **R**, and **L** characters (shown in boldface type) are literal characters used in the form as flags for their associated fields.
- The other characters shown above represent information or special characters used in the fields shown.
- Mandatory fields begin with an upper case letter, and optional fields begin with a lower case letter (though the data in the fields is not case sensitive)

7.1.1 <LF> Line Feed

This is a standard ASCII Line Feed character (ASCII 10).

7.1.2 <CRC> Cyclic Redundancy Check

This is the checksum associated with the message and used for error detection. The actual CRC field is a 4-byte field representing a 16 bit CRC value and presented in ACSII encoded hexadecimal notation, also known as Hex-ASCII. (For a detailed description of Cyclic Redundancy Check calculation, see Appendix B.)

7.1.3 <0LLL> Packet Length

The packet length is also presented in Hex-ASCII. Currently the field is limited to zero followed by 3 Hex Digits. In the future this field may be lengthened to 4 digits.

The packet length is the number of bytes in the packet beginning with the first quote character of the token ID up to, but not including, the <CR> character.

7.1.4 <"ID"> ID Token

The ID Token, with literal quotes before and after the ID, denotes the type of data that is being sent and allows differentiation between various embedded protocols. Tokens are assigned by SIA upon written request.

7.1.5 <Sequence#!segment#|>

The Sequence number is always present. Each device initiating a dialogue is responsible for assigning its own message sequence number. Since the automation computer typically communicates with multiple receivers, the computer shall internally manage the sequence numbers provided by the multiple receivers in their initiating messages.

The Sequence number is normally a 4-byte field, but it may be extended to multiple bytes. The four ASCII characters represent a 16-bit BCD value that increments with each unique message packet. The smallest allowable number is 0001 and the largest allowable number is 9999. Not allowable in this field are Hexadecimal characters. The value 9999 is succeeded by the value 0001. 0000 is NOT a valid sequence number unless it is in a NAK message (See Section 5.4.2.2, NAK Packets).

The sender increments its sequence number when:

- A valid Acknowledgment has been received.
- A communications fault has been detected and the message has been resolved by a local operator.

The sequence number is NOT incremented when:

- The message is being repeated due to a NAK, a failure to receive an ACK, or a failure to properly receive data in a response to a request.
- A valid continuation character has been transmitted (instead the segment is incremented.)

The segment number is used for long messages to number the segments of a multi-part message and it is necessary only on continued messages. It is a 2 to 5 byte field preceded by the "**!**" delimiter (ASCII 33 Decimal). It contains 1 to 4 ASCII numerals with preceding zeroes optional.

The delimiter "|" (ASCII 07) following the segment number is a continuation character. It denotes that more segments follow and is required in a continued message for all but the final segment. (See Section 5.3.4 Long Messages for additional details.)

7.1.6 <Rreceiver#>

The receiver number field is comprised of the literal character "**R**", followed by the actual number. This is anticipated to be 2 digits but in today's large central stations it could be more digits.

Valid characters are all hex numbers. The receiver number may NOT be all 0's for messages sent from the receiver to the computer.

7.1.7 <Lline#>

The line number field is comprised of the literal character "L" followed by the line number. This is anticipated to be two to 4 digits but may be 1 to 6 digits easily. Valid characters are all hex numbers. Since there is no card number field in the message form, each line in a receiver must have a unique number.

7.1.8 [...data...]

All data is in hex ASCII digits. Its format is dependent upon the ID Token of the message.

Where an account number is associated with a message, the actual data is preceded by the "**#**" character, the account number, and the field separator "**|**". This will be the case for most message types.

Because of their use in the message packet, data fields may not contain the special characters shown below.

Table 4: Reserved Characters

Character	**Meaning**
| - (ASCII 124)	Data Separator or Long Message Continuation
[- (ASCII 91)	Start of Text
] - (ASCII 93)	End of Text
LF - (ASCII 10)	Begin Message
CR - (ASCII 13)	End Message

7.1.9 <timestamp>

This field denotes the time the message was queued by the sender. The format is:

"_HH:MM:SS,MM-DD-YYYY"

It uses ASCII digits 0-9. The punctuation (colons, dashes, and comma) and the leading underscore character are required. No other special characters may be used. For fields with values less than two digits, the upper digits are to be padded with ASCII zeros (ASCII 48).

This field is optional. Examples in this standard typically do not show it in message forms and examples, but it may be used in any message type.

7.1.10 <CR>

This is a standard ASCII Carriage Return character (ASCII 13)

7.2 Long Messages

In certain instances, message data may not fit within a single message frame (i.e. when sending video information or large files). To accommodate this, the protocol allows a message to be extended across two or more frames, forming a "long message".

When long messages are used, the header is changed in the following manner. First, the sequence number field will end with an ASCII "!". This indicates the presence of a **segment number** field, containing from 1 to 4 ASCII digits (0-9), with leading zeros optional, and spaces not allowed. The segment number field will be added to the frames of multi-packet messages so that the receiving device can re-assemble the message, as well as detect missing or lost frames.

A second characteristic of long messages is the presence of the **continuation character**, an ASCII "|". When used, the | character appears immediately before the Receiver Number (i.e. "|R"). This indicates that this message will be continued in the next frame. Continuation characters are required in all frames of a long message, except the final frame.

Another item to note about long messages is that the sequence number, receiver number, and line number fields in the header portion of the frame must remain identical to the first frame for the message.

The form of a long message frame is:
<LF> <CRC> <0LLL> <"ID"> <Sequence#**!**segment#|> <**R**Receiver#> <**L**Line#> [...data...] <timestamp> <CR>

The rules for creation of segment numbers are:

- The segment number field is always allowed in any transmission, but is only REQUIRED for long messages, continued across multiple frames.
- Use of the segment number field is optional for one line messages, but is required in all frames of a multi-packet message.
- The format of the segment number field is defined to contain from 1 to 4 ASCII digits (0-9), with leading zeros optional, and spaces not allowed.
- The segment number field is placed after the sequence number. The continuation character (used in all but the last segment) appears immediately following the segment number.

An example of a long message is shown in Appendix C - Additional Examples.

7.3 Message Protocol

Under normal conditions, packet transfer is initiated by the receiver. The receiver formats a message packet and transmits it to the computer. The computer responds with a response packet, which may include data for the receiver or even for the transmitter.

Each message includes a unique sequence identification along with information about the origin of the message.

7.4 Initiating Packets

Initiating messages are those that are sent without being tied to a previous message.

Initiating messages shall be in one of the following forms:

1. **Data** - (mandatory) messages with forwarded data or data about the central station equipment status (See Section 7.4.1 and Appendix A for details on Data Message Types.)
2. **NULL** - (mandatory) a simple link test to ensure continued viability of the connection
3. **Data / Operation Request** - (optional) a request for data or for a particular operation (definition of these operations are outside the scope of this standard)

Data and NULL messages must be supported by all devices. Data / Operation Request messages are optional.

When the receiver sends an initiating message, the receiver and line number are those of the receiver. However, when the computer sends an initiating message, the receiver and line number are each zero.

7.4.1 Data Packet

Data Messages are mandatory messages. They are identified with ID Tokens as listed in Appendix A - Message ID Tokens.

7.4.2 NULL Packet (Link Test)

If a device simply wants to keep a link alive it may send a message with no data. This is accomplished by sending a Null Data packet where the ID Token is “NULL”, the Sequence# is all zeros, and the data portion of the packet is empty, with only the square brackets.

<LF> <CRC> <0LLL> <**"NULL"**> <**0000**> <**R**Receiver#> <**L**Line#> [] <CR>

NULL packets are mandatory messages.

7.4.3 Data / Operation Request Packet

Data / Operation Requests are optional messages. If used, they follow the same general form as other messages, but the ID Token and the data are peculiar to a given device. These parameters are beyond scope of this standard. It is the responsibility of device manufacturers to establish and advise of the tokens and data structures that will be used.

7.4.4 Response Packets

Each message sent between the receiver and computer requires a response of some sort. The response shall be in one of the following forms:

1. **ACK** - (mandatory) indicates that the data sent was received correctly
2. **NAK** - (mandatory) indicates that an error occurred
3. **RTN** - (optional) a packet containing requested data or the result of a requested operation
4. **DUH** - (mandatory) indicates that the device does not understand or support an initiating message that it received

ACK, NAK, and DUH messages must be supported by all devices. RTN messages are optional.

The appropriate response for a Data or NULL packet is an ACK or NAK packet. The appropriate response for a Data / Operation Request packet is an RTN or DUH packet.

A response packet is sent only once and does not require a corresponding ACK. For example, if the receiver requested the time from the computer, the computer would return the time without expecting an ACK. If an error occurred in the time packet, it would be up to the receiver to detect the error and request the time again.

ACK, RTN, and DUH packets mirror the receiver and line numbers of the message they are responding to. In a NAK message, the receiver and line numbers are each zero.

7.4.5 ACK Packet

All messages sent out by the receiver and the computer must be acknowledged by the opposite side. The minimum response packet contains only an acknowledgment of a message received from the receiver. This is accomplished by sending an "ACK Response" (ACK) packet. The ACK packet consists of the fields shown below:

<LF> <CRC> <0LLL> <**"ACK"**> <Sequence**!**#segment#|> <**R**Receiver#> <**L**Line#> [] <CR>

For an ACK packet, the <Sequence#**!**segment#|>, <**R**Receiver#>, and <**L**Line#> fields echo the data from the packet that is being acknowledged, but the data portion of the packet is empty, with only the square brackets.

7.4.6 NAK Packet

If a message was received with errors, it may not be possible to echo the fields as in an ACK. Therefore, the NAK message is used to respond, as shown below:

<LF> <CRC> <0LLL> <**"NAK"**> <**0000**> <**R0**> <**L0**> [] <CR>

Here the ID field contains the token "NAK". The sequence number is forced to 0000, with no segment number. The receiver number and line number are each set to zero, and the data portion of the packet is empty, with only the square brackets.

When a device receives this response, it should look into its buffer for the last (unacknowledged) message sent. Any re-send timeouts for the message shall be forced to expire, and the message shall be re-sent immediately.

7.4.7 RTN Packet

If a device requires data from its counterpart, it may issue a request for data. When data is returned, it must be tied to the data request.
This is accomplished by sending an "Return Data" (RTN) packet. The RTN packet consists of the fields described below:

<LF> <CRC> <0LLL> <**"RTN"**> <Sequence#**!**Segment#|> <**R**Receiver#> <**L**Line#> […data…] <CR>

For the RTN packet, the <Sequence#!segment#|>, <Rreceiver#>, and <Lline#> fields echo the data from the packet that is being reponded to. The data portion of the packet contains the returned data or result code.

7.4.8 DUH Packet

Because Data / Operation Request messages are peculiar to certain functional capabilities of equipment, it is possible that a device can receive a request that it does not understand or cannot handle. When a device is unable to comply with such a request, it shall respond with a DUH message as shown below:

<LF> <CRC> <0LLL> <**"DUH"**> <Sequence**!**#segment#|> <**R**Receiver#> <**L**Line#> [] <CR>

For an ACK packet, the <Sequence#**!**segment#|>, <**R**Receiver#>, and <**L**Line#> fields echo the data from the packet that is being responded to, but the data portion of the packet is empty, with only the square brackets.

7.4.9 Unrecognized Data Message

A receiver should attempt to forward all available data regarding a bad call from a transmitter. To provide for this, the internal data structure shall encompass all available data on the bad call. The Unrecognized Data message has the form shown below.

<LF> <CRC> <0LLL> <**"SIA-UD"**> <Sequence#**!**Segment#|> <**R**Receiver#> <**L**Line#> [//C …….//D ……..] <CR>

The ID Token for this message is "SIA-UD", and the remaining header fields are supplied by the receiver.

The internal structure follows common conventions used in most coding standards: The //C denotes the Caller ID (if available and patent rights allow) and the //D denotes that all following information was extracted and forwarded intact, but the receiver does not know how to deal with the "format".

The data is primarily made available for debugging in the central station, allowing for the possibility that a human reader may be able to see a pattern.

8 Assurance Layer

8.1 Message Integrity

Messages from the receiver, which contain unique sequence identification fields, are repeated until a response packet is received from the computer with a corresponding sequence identification or until a fault condition is detected. The repeat interval and number of times each message is repeated may be configured by the receiver setup, but they will be defaulted at 4 second intervals and 4 retries.

Detection of a fault in the computer link causes all messages that remain unacknowledged to be passed to the receiver operator for local resolution.

8.2 Fault Detection

Faults are of two types: **failure** and **trouble**

A communication **failure** exists when:

- No messages or ACK messages have been received on either side in 16 seconds with messages unacknowledged in the receiver.

A communication **trouble** exists when:

- Three consecutive response packets have been received and included framing errors or an incorrect checksum. At this point an error code such as G161_S33_Txxx with elaboration of error should be sent and also should be journaled/printed.

8.3 Fault Reaction

8.3.1 Communications Failure

The receiver shall create an internal condition code indicating a System Communications Failure (SIA-CSE code G161_S32) that includes a date and time stamp. This message and all other messages still unacknowledged at the time of the fault shall be delivered to the local operator, in accordance with UL requirements. Each message acknowledged locally by the receiver operator should create a new message for the computer to be delivered if the link is restored. This new message is a copy of the original, but with the time and the date of resolution added to the message data field.

It is understood that the capacity to store locally acknowledged messages for later delivery to the computer is limited. The standard imposes no minimum number of messages be stored. However, storage for at least eight such messages is strongly recommended. If messages awaiting delivery to the computer (after local acknowledgment) are destroyed due to capacity overrun, then the receiver shall include the condition code for Buffer Overflow / Data Loss (SIA-CSE code G161_S38) along with the associated text in the data field of the message created at the time the link failed.

8.3.2 Communication Trouble

When trouble in the link with the computer is detected, the receiver should generate a message with the condition code for Communications Failure (SIA-CSE code G161_S32) or Excessive Failures (SIA-CSE code G161-S33), as per the failure condition, along with the associated text describing the condition details. This message should be sent to the local receiver operator and the computer.

8.3.3 Link Integrity

Supervision of the receiver-to-computer link is performed by both devices, with each typically having it own interval of required communication. However, if either device fails to receive a link test or data from the other in 16 seconds (maximum), that device shall transmit a link test (NULL message) to other.

NOTE: This time can be shortened by setting options on the receiver and/or automation computer, but the recommended minimum time is 4 seconds. (Empirically, it has been found that a 4 second minimum is the shortest workable timeout .)

Then, if the test (NULL message) is not successfully acknowledged within 60 seconds (including the retries described in 6.1 Message Integrity), a fault shall be declared.

The maximum total time of lapsed communication to declare a fault shall not exceed 90 seconds, in accordance with UL requirements.

APPENDIX A - MESSAGE ID TOKENS

The general form of message ID Tokens is: *Manufacturer Abbreviation - Protocol Abbreviation* Tokens are not case sensitive at this time but are always printed and shown as upper case in this document. There is no limit to the number of characters, but they are typically a sufficient number of identifying characters to aid central station technicians in troubleshooting should that be necessary.

The following table shows all ID Tokens supported by this standard, including those for root messages, central station equipment condition messages, and forwarded Data messages received as proprietary manufacturer transmitter protocols. The ID Tokens listed (except the root messages) are expanded in the listings that follow the table. Data / Operation request messages are not listed, as they are peculiar to functional capabilities of various devices that are beyond the scope of this standard.

To facilitate additions of tokens as needed, transmitter protocol tokens are listed in alphabetical order.

Manufacturer	Protocol	Token	Name
this standard	Root	NULL	Link Test (Keep Link Alive)
this standard	Root	ACK	Acknowlege (successful)
this standard	Root	NAK	Negative Acknowlege (error)
this standard	Root	RTN	Return (Data or Result Code)
this standard	Root	DUH	Don't Understand / Can't Handle
this standard	Root	SIA-UD	Unrecognized Transmitter Data
this standard	Root	SIA-CSE	Central Station Equipment Status
generic	various	SIA-PUL	Generic Pulse Codes
ACR	SF	ACR-SF	Acron Super Fast
ADM	CID	ADM-CID	Ademco Contact ID
ADM	41E	ADM-41E	Ademco 4-1 Express
ADM	42E	ADM-42E	Ademco 4-2 Express
ADM	HS	ADM-HS	Ademco High Speed
DSC	43	DSC-43	DSC 4-3
FBI	SF	FBI-SF	FBI Super Fast
ITI	I	ITI-I	ITI Standard
SCN	S8	SCN-S8	Scancom 4-8-1, 5-8-1, 6-8-1
SCN	S16	SCN-S16	Scancom 4-16-1, 5-16-1, 6-16-1
SCN	S24	SCN-S24	Scancom 4-24-1, 5-24-1, 6-24-1
SCT			Scantronics Reserved

SES	SS	SES-SS	Sescoa Super Speed
SIA	DCS	SIA-DCS	SIA DCS
SIA	S2K	SIA-S2K	SIA 2000
SK	FSK1	SK-FSK1	Silent Knight FSK1
SK	FSK2	SK-FSK2	Silent Knight FSK2

Message Type SIA-UD (Unrecognized Data)

<LF><CRC><0LLL><"SIA-UD"><sequence#><Rreceiver#><Lline#>[//C############//D……]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-UD	Message Type: Unrecognized Data
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
//C########	Caller ID if Patent Rights allow
//Ddddddddd	All Unrecognized Data extracted

Message Type SIA-CSE (SIA Central Station Equipment messages)

<LF><CRC><0LLL><"SIA-CSE">[***Code***]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-CSE	Message Type – Central Station Equipment
Code	to contain code (and # if applicable) a mixture of ASCII Numerals and characters Formatted as below. Note that "_" is the separator to allow "." to be used in text. *Note: Code* is built to observe SIA 2000 Event code structures however is targeted at the central station equipment and has the following structure. Gxxx_Sxxx_Pxxx_Txxxxxxxx etc. each field is from 1 to n characters in length and must be separated by underscores "_" After the G and S fields. Subsequent fields may be cascaded at will.
Gxxx	General Event Code
Sxxx	Specific Event Code
Pxxx	Point ID for Computer Use - This may be associated with a text ID but a point ID allows for exact restore codes for event. As a note, P0 is typically used for a general issue for example all AC has failed or all communication is failed more specific codes are identified with _P1 and up. Remember the issue here is that the Automation can see a Trouble and restoral on the same “point” code and so does not have to guess.
Txxx	Event ID in free form text. This portion is meant for display to a human operator and so should be sensible. Note "_" is not allowed in a text message.
Uxxx	User ID on Receiver
Axxx	Account ID on Receiver or Caller ID etc.
Lxxx	Line Number (actual line)
Cxxx	Card Number (actual card)
Fxxx	File Handle - File name on file system must not contain "_"
Nxxx	Port Line Number - Identification of a "port" for live feed most likely a

extension # on Ethernet may be a socket number. It is expected at configuration time this will be resolved on automation and receiver by an option

The following tables describe most receiver trouble conditions and other events that occur in Central Station Equipment. These tables contain the same information, but for ease of reference, the first table is sorted by Description and the second table is sorted by CSE Event Code.

SIA-CSE Codes - Sorted by Description

Description	CSE Event Code	Notes
AC Fail	G161_S1_P0_Tx	Supply Point and Text Field
AC Restore	G162_S1_P0_Tx	Supply Point and Text Field
Audio Session Begin	G161_S40	
Audio Session End	G162_S40	
Audio to File Begin	G161_S44_Fxxx	Supply File Handle
Audio to File End	G162_S44_Fxxx	Supply File Handle
Audio to Port Begin	G161_S42_Nxxx	Supply port number
Audio to Port End	G162_S42_Nxxx	Supply port number
Battery Missing	G161_S2_Px_Tx	
Busy Seconds	G161_S14_Lx_Tx	Supply Line and Text Fields
Caller ID Information	G161_S35_Axxxx	Caller Id inserted in Account Structure
Communications Error	G161_S32	
Communications Error Restore	G162_S32	
Communications Failure	G161_S32_Lx_Px_Tx	
Communications Restore	G162_S32_Px_Tx	
Communications Trouble	G161_S32_Tx	
Computer Error / Trouble	G161_S50	Computer Trouble / Error Typically it may miss a heartbeat etc. You would follow with a _P0 and _T<disc> in this case
Computer Error / Trouble Restore	G162_S50	
CPU Software Version	G162_S8_Txxx	Supply in Text Field
Date Changed	G161_S9_Px_Tx	Supply Point and Text Field
Downloader Programming Denied	G161_S8	Supply Point and Text Field
Downloader Programming Failure	G161_S8	Supply Point and Text Field
Downloader Programming Started	G161_S8	Supply Point and Text Field
Downloader Programming Successful	G161_S8	Supply Point and Text Field
Excessive Data Lost	G161_S33_Txxx	Supply Text Field for Reference
Excessive Errors	G161_S33	Supply Point and Text Field
Extra Account Report	G161_S36	
Fail To Report	G161_S37_Ax_Tx	
Invalid Report	G161_S34	Add Line Point and Text as Required

Description	CSE Event Code	Notes
Lan Network Condition	G163_S10_Tx	LAN
Lan Network Failure	G163_S2	
Lan Network Restore	G164_S10_Tx	LAN
Line Activity Resumed	G162_S12_Lx	
Line Card Software Version	G161_S8_Lx_Tx	Supply Line and Text Field
Local Programming Denied	G161_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Ended	G162_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Failure	G161_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Initiated	G161_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Successful	G162_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Successful	G161_S5_Lx_Tx	
Log Off Operator	G162_S6_Ux_Tx	
Log On Operator	G161_S6_Ux_Tx	
Log Threshold	G161_S4_Px_Tx	Supply Point and Text Field
Log Threshold Restore	G162_S4	Supply Point and Text Field
Low Received Signal Strength	G161_S3_Lx_Px_Tx	
No Data Received	G161_S32_Px_Lx_Tx	Supply Point and Text Field
No Line Activity	G161_S12_Lx	
No Response To Handshake	G161_S31	
Paper In	G162_S21	
Paper Out	G161_S21	
Parameter Checksum Changed	G161_S8_Px_Lx_Tx	
Parameter Checksum Fail	G161_S8_Px_Lx_Tx	
Phone Line Restore	G162_S5_Lx_Tx	
Phone Line Trouble	G161_S5	
Power Supply Restore	G162_S1_Px_Tx	
Power Supply Trouble	G161_S1_Px_Tx	
Power Up	G161_S7	
Printer Restore / On Line	G162_S20_Px_Tx	
Printer Trouble / Off Line	G161_S20_Px_Tx	
Receiver Off Line	G161_S13_Tx	
Receiver On Line	G162_S13_Tx	
Receiver Restore	G162_S3_Lx_Px_Tx	
Receiver Trouble	G161_S3_Lx_Px_Tx	
Remote Programming Begin	G161_S8_Ux_Tx	Supply User Making Changes and Text
Remote Programming Denied	G161_S8_Ux_Tx	Supply User Making Changes and Tex
Remote Programming Failure	G161_S8_Ux_Tx	Supply User Making Changes and Text
Remote Programming Successful	G162_S8_Ux_Tx	Supply User Making Changes and Text
RF Interference	G161_S3_Lx_Px_Tx	
RF Interference Restore	G161_S3_Lx_Px_Tx	
Schedule Changed	G161_S8_Px_Tx	Supply Point and Text Field
Schedule Executed	G162_S8_Px_Tx	Supply Point and Text Field
Supplementary Text	G161_S60_Px_Tx	Supply Point and Text Field
System Battery Restore	G162_S2_Px_Tx	

Description	CSE Event Code	Notes
System Battery Trouble	G161_S2_Px_Tx	
Temperature High	G161_S11_Px_Tx	Supply Point and Text Field
Temperature Restore	G162_S11_Px_Tx	Supply Point and Text Field
Time Changed	G161_S10_Tx	Supply Point and Text Field?
Unknown Message	G161_S34	Add Line Point and Text as Required
UPS AC Failure	G161_S1_Px_Tx	Supply Point and Text Field
UPS AC Restore	G162_S1_Px_Tx	Supply Point and Text Field
UPS Battery Low	G161_S2_Px_Tx	Supply Point and Text Field
UPS Battery Restore	G161_S2_Px_Tx	Supply Point and Text Field
User Code Added	G162_S8_Ux_Tx	Supply User Making Changes and Text
User Code Changed	G161_S8_Ux_Tx	Supply User Making Changes and Text
User Code Deleted	G162_S8_Ux_Tx	Supply User Making Changes and Text
User Level Set	G162_S8_Ux_Tx	Supply User Making Changes and Text
Video Session Begin	G161_S41	
Video Session End	G162_S41	
Video to File Begin	G161_S45_Fxxx	Supply File Handle
Video to File End	G162_S45_Fxxx	Supply File Handle
Video to Port Begin	G161_S43_Nxxx	Supply port number
Video to Port End	G162_S43_Nxxx	Supply port number
Wan Network Condition	G163_S1_Tx	WAN
Wan Network Restore	G164_S1_Tx	WAN
Watchdog Reset	G161_S7_Px_Tx	

SIA-CSE Codes - Sorted by Event Code

Description	CSE Event Code	Notes
Time Changed	G161_S10_Tx	Supply Point and Text Field?
Temperature High	G161_S11_Px_Tx	Supply Point and Text Field
No Line Activity	G161_S12_Lx	
Receiver Off Line	G161_S13_Tx	
Busy Seconds	G161_S14_Lx_Tx	Supply Line and Text Fields
UPS Battery Low	G161_S2_Px_Tx	Supply Point and Text Field
UPS Battery Restore	G161_S2_Px_Tx	Supply Point and Text Field
Battery Missing	G161_S2_Px_Tx	
System Battery Trouble	G161_S2_Px_Tx	
Printer Trouble / Off Line	G161_S20_Px_Tx	
Paper Out	G161_S21	
RF Interference Restore	G161_S3_Lx_Px_Tx	
Low Received Signal Strength	G161_S3_Lx_Px_Tx	
RF Interference	G161_S3_Lx_Px_Tx	
Receiver Trouble	G161_S3_Lx_Px_Tx	
No Response To Handshake	G161_S31	
Communications Error	G161_S32	
Communications Failure	G161_S32_Lx_Px_Tx	
No Data Received	G161_S32_Px_Lx_Tx	Supply Point and Text Field
Communications Trouble	G161_S32_Tx	
Excessive Errors	G161_S33	Supply Point and Text Field
Excessive Data Lost	G161_S33_Txxx	Supply Text Field for Reference
Invalid Report	G161_S34	Add Line Point and Text as Required
Unknown Message	G161_S34	Add Line Point and Text as Required
Caller ID Information	G161_S35_Axxxx	Caller Id inserted in Account Structrure
Extra Account Report	G161_S36	
Fail To Report	G161_S37_Ax_Tx	
Log Threshold	G161_S4_Px_Tx	Supply Point and Text Field
Audio Session Begin	G161_S40	
Video Session Begin	G161_S41	
Audio to Port Begin	G161_S42_Nxxx	Supply port number
Video to Port Begin	G161_S43_Nxxx	Supply port number
Audio to File Begin	G161_S44_Fxxx	Supply File Handle
Video to File Begin	G161_S45_Fxxx	Supply File Handle
Phone Line Trouble	G161_S5	
Local Programming Successful	G161_S5_Lx_Tx	
Computer Error / Trouble	G161_S50	Computer Trouble / Error Typically it may miss a heartbeat etc_ You would follow with a _P0 and

Description	CSE Event Code	Notes
		_T<disc> in this case
Log On Operator	G161_S6_Ux_Tx	
Supplementary Text	G161_S60_Px_Tx	Supply Point and Text Field
Power Up	G161_S7	
Watchdog Reset	G161_S7_Px_Tx	
Downloader Programming Denied	G161_S8	Supply Point and Text Field
Downloader Programming Failure	G161_S8	Supply Point and Text Field
Downloader Programming Started	G161_S8	Supply Point and Text Field
Downloader Programming Successful	G161_S8	Supply Point and Text Field
Line Card Software Version	G161_S8_Lx_Tx	Supply Line and Text Field
Parameter Checksum Fail	G161_S8_Px_Lx_Tx	
Parameter Checksum Changed	G161_S8_Px_Lx_Tx	
Schedule Changed	G161_S8_Px_Tx	Supply Point and Text Field
User Code Changed	G161_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Initiated	G161_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Denied	G161_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Failure	G161_S8_Ux_Tx	Supply User Making Changes and Text
Remote Programming Begin	G161_S8_Ux_Tx	Supply User Making Changes and Text
Remote Programming Denied	G161_S8_Ux_Tx	Supply User Making Changes and Tex
Remote Programming Failure	G161_S8_Ux_Tx	Supply User Making Changes and Text
Date Changed	G161_S9_Px_Tx	Supply Point and Text Field
AC Restore	G162_S1_P0_Tx	Supply Point and Text Field
UPS AC Restore	G162_S1_Px_Tx	Supply Point and Text Field
Power Supply Restore	G162_S1_Px_Tx	
Temperature Restore	G162_S11_Px_Tx	Supply Point and Text Field
Line Activity Resumed	G162_S12_Lx	
Receiver On Line	G162_S13_Tx	
System Battery Restore	G162_S2_Px_Tx	
Printer Restore / On Line	G162_S20_Px_Tx	
Paper In	G162_S21	
Receiver Restore	G162_S3_Lx_Px_Tx	
Communications Error Restore	G162_S32	
Communications Restore	G162_S32_Px_Tx	
Log Threshold Restore	G162_S4	Supply Point and Text Field
Audio Session End	G162_S40	
Video Session End	G162_S41	
Audio to Port End	G162_S42_Nxxx	Supply port number
Video to Port End	G162_S43_Nxxx	Supply port number

Description	CSE Event Code	Notes
Audio to File End	G162_S44_Fxxx	Supply File Handle
Video to File End	G162_S45_Fxxx	Supply File Handle
Phone Line Restore	G162_S5_Lx_Tx	
Computer Error / Trouble Restore	G162_S50	
Log Off Operator	G162_S6_Ux_Tx	
Schedule Executed	G162_S8_Px_Tx	Supply Point and Text Field
CPU Software Version	G162_S8_Txxx	Supply in Text Field
User Code Deleted	G162_S8_Ux_Tx	Supply User Making Changes and Text
User Code Added	G162_S8_Ux_Tx	Supply User Making Changes and Text
User Level Set	G162_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Successful	G162_S8_Ux_Tx	Supply User Making Changes and Text
Local Programming Ended	G162_S8_Ux_Tx	Supply User Making Changes and Text
Remote Programming Successful	G162_S8_Ux_Tx	Supply User Making Changes and Text
Wan Network Condition	G163_S1_Tx	WAN
Lan Network Condition	G163_S10_Tx	LAN
Lan Network Failure	G163_S2	
Wan Network Restore	G164_S1_Tx	WAN
Lan Network Restore	G164_S10_Tx	LAN

Message Type SIA-PUL-1 (Generic Pulse Codes)

<LF><CRC><0LLL><"**SIA-PUL-1-*code***"><Sequence#><**R**Receiver#><**L**Line#>[**#**aaaald]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-PUL	Message Type – Generic Pulse Codes
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
Code	An optical code of the form:
Fxxxx	Where xxxx is the handshake frequency (Hz), either 1400 or 2300
Bxx	where xx is the Baud rate (pps), either 10, 14, 20, or 40
Sxxx	where xxx is the account / data split (bytes), either: 41 (for a 4/1 split) or 31 (for a 3/1 split). For data sent by a transmitter as "double round", only a single round is to be forwarded.
aaaa	Account number (variable: 3 or 4 digits)
d	where the data will be one characters as indicated in Sxx

Note: This message type applies to various pulse formats offered by various manufacturers. The Code is supplied for possible use by central station technicians and is typically not needed (and therefore ignored) by the automation computer.

Message Type SIA-PUL-2 (Generic Pulse Codes)

<LF><CRC><0LLL><"**SIA-PUL-2-*code***"><Sequence#><**R**Receiver#><**L**Line#>[**#**aaaaldd]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-PUL	Message Type – Generic Pulse Codes
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
Code	An optical code of the form:
Fxxxx	Where xxxx is the handshake frequency (Hz), either 1400 or 2300
Bxx	where xx is the Baud rate (pps), either 10, 14, 20, or 40
Sxxx	where xxx is the account / data split (bytes), either: 42 (for a 4/2 split) or 32 (for a 3/2 split), 41E (for 4/1 extended), 31E (for 3/1 extended). For data sent by a transmitter as "double round", only a single round is to be forwarded.
aaaa	Account number (variable: 3 or 4 digits)
dd	where the data will be one characters as indicated in Sxxx

Note: This message type applies to various pulse formats offered by various manufacturers. The Code is supplied for possible use by central station technicians and is typically not needed (and therefore ignored) by the automation computer.

Message Type SES-SS (Sescoa Super Speed)

<LF><CRC><0LLL><"SES-SS"><sequence#><Rreceiver#><Lline#>[#aaaa|IAAC]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SES-SS	Message Type – Sescoa Super Speed
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaa	The communicator's account number
I	Event code
AA	Two digit zone code or the first two digits user code
C	Space if zone report, or the last digit user code

Message Type ACR-SF (Acron Super Fast)

<LF><CRC><0LLL><"ACR-SF"><sequence#><Rreceiver#><Lline#>[#aaaa|CCCCCCCC]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
ACR-SF	Message Type: Acron Super Fast
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaa	The communicator's account number
CCCCCCCC	Channel 1-8

Message Type ADM-CID (Ademco Contact-ID)

<LF><CRC><0LLL><"ADM-CID"><sequence#><**R**receiver#><**L**line#>[**#**aaaaaa|QXYZsGGsCCC]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
ADM-CID	Message Type: Ademco Contact-ID
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaa	The communicator's account number
Q	Qualifier, 1=New event or opening, 3=New restore or closing, 6=Previous event
XYZ	Class code and event code
s	One space
GG	Group number
CCC	Zone codes or user ID

As an example, account 3456 sends a Perimeter Burglary alarm for Zone 5 <LF>????0028"ADM-CID"0001R01L02[#003456|1131s01s005]<CR>

Message Type ADM-41E (Ademco 4-1 Express)

<LF><CRC><0LLL><"ADM-41E"><sequence#><**R**receiver#><**L**line#>[**#**aaaaaaX]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
ADM-41E	Message Type: Ademco 4-1 Express
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaa	The communicator's account number
X	Zone number – or event code

As an example, account 3456 sends an event code 7
<LF>????001E"ADM-41E"0003R01L02[#003456|7]<CR>

Message Type ADM-42E (Ademco 4-2 Express)

<LF><CRC><0LLL><"ADM-42E"><sequence#><**R**receiver#><**L**line#>[**#**aaaaaaXY]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
ADM-42E	Message Type: Ademco 4-2 Express
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaa	The communicator's account number
XY	Event code and Zone number

As an example, account 3456 sends an event code 78
<LF>????001F"ADM-42E"0003R01L02[#003456|78]<CR>

Message Type ADM-HS (Ademco High Speed)

<LF><CRC><0LLL><"ADM-HS"><sequence#><**R**receiver#><**L**line#>[**#**aaaaaa|CCCCsCCCCsC]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
ADM-HS	Message Type: Ademco High Speed
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaa	The communicator's account number
CCCC	Channels 1-4
s	One space
CCCC	Channels 5-8
s	One space
C	Supervisory channel

As an example, account 3456 sends a Test Report
<LF>????0027"ADM-HS"0004R01L02[#003456|5555s5555s9]<CR>

Message Type DSC-43 (DSC 4-3)

<LF><CRC><0LLL><"DSC-43"><sequence#><Rreceiver#><Lline#>[#aaaa|XYY]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
DSC-43	Message Type: DSC/Sur-Gard 4-3
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaa	The communicator's account number
X	Event code number
YY	Zone number

Message Type SCN-S8 (Scancom 4-8-1, 5-8-1, 6-8-1)

<LF><CRC><0LLL><"SCN-S8"><sequence#><Rreceiver#><Lline#>[#aaaaaa|CCCCsCCCCsC]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SCN-S8	Message Type: Scancom 4-8-1, 5-8-1, 6-8-1
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaa	The communicator's account number
CCCC	Channels 1-4
s	One space
CCCC	Channels 5-8
s	One space
C	Supervisory Channel

Message Type SCN-S16 (Scancom 4-16-1, 5-16-1, 6-16-1)

<LF><CRC><0LLL><"SCN-S16"><sequence#><Rreceiver#><Lline#>[#aaaaaa|CCCCsCCCCsCCCCsCCCCsC]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SCN-S16	Message Type: Scancom 4-16-1, 5-16-1, 6-16-1
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaa	The communicator's account number
CCCC	Channels 1-4
s	One space
CCCC	Channels 5-8
s	One space
CCCC	Channels 9-12
s	One space
CCCC	Channels 13-16
s	One space
C	Supervisory Channel

Message Type SCN-S24 (Scancom 4-24-1, 5-24-1, 6-24-1)

<LF><CRC><0LLL><"SCN-S24"><sequence#><Rreceiver#><Lline#>[#aaaaaa|CCCCsCCCCsCCCCsCCCCsCCCCsCCCCsC]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SCN-S24	Message Type: Scancom 4-24-1, 5-24-1, 6-24-1
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaa	The communicator's account number
CCCC	Channels 1-4
s	One space
CCCC	Channels 5-8
s	One space
CCCC	Channels 9-12
s	One space
CCCC	Channels 13-16
s	One space
CCCC	Channels 17-20
s	One space
CCCC	Channels 21-24
s	One space
C	Supervisory Channel

Message Type FBI-SF (FBI Super Fast)

<LF><CRC><0LLL><"FBI-SF"><sequence#><Rreceiver#><Lline#>[#aaaa|TZZE]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
FBI-SF	Message Type: FBI Super Fast
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaa	The communicator's account number
T	Zone Type
ZZ	Zone number
E	Event Type

Message Type ITI-I (ITI)

<LF><CRC><0LLL><"ITI-I"><sequence#><Rreceiver#><Lline#>[#aaaaa|PGIZZEWN]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
ITI-I	Message Type: ITI
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaa	The communicator's account number
P	Panel Descriptor For ITI Panel ID
G	Group number
I	O/C user ID
ZZ	Zone number
E	Condition code
W	Protection level was
N	Protection level now

Message Type SCT-*TBD* (Scantronics reserved)

<LF><CRC><0LLL><"SCT-*TBD*"><sequence#><Rreceiver#><Lline#>[#aaaaaaa|cccc….cccc]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SCT-*TBD*	Message Type: Scantronics (reserved)
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaaa	The communicator's account number
cccc….cccc	ASCII-Hex representation of the received packet. (Variable length, complete packet, from header to checksum.)

Message Type SIA-S2K (SIA 2000)

<LF><CRC><0LLL><"SIA-S2K"><sequence#><Rreceiver#><Lline#>[#aaaaaaaaaa|cccc….cccc]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-S2K	Message Type: SIA 2000
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaaa	The communicator's account number which is 10 digits long
cccc….cccc	ASCII-Hex representation of the received SIA 2000 packet. (Variable length, complete packet, from header to checksum.)

Note: This is a non-extended message. See Long Messages section for discussion regarding very long / extended messages.

Message Type SIA-DCS (SIA DCS Preferred)

<LF><CRC><><"SIA-DCS"><sequence#><Rreceiver#><Lline#>[#aaaaaaa|cccc….cccc]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-DCS	Message Type: SIA DCS
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaaa	The communicator's account number
cccc….cccc	ASCII received SIA DCS packet. (Variable length, complete packet, from header to checksum.)

Message Type <HT>"horizontal tab ASCII 09" (SIA DCS Alternate)

<LF><CRC><0LLL><HT><sequence#><Rreceiver#><Lline#>[#aaaaaaa|cccc….cccc]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
HT	Message Type: SIA DCS
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaaa	The communicator's account number
cccc….cccc	ASCII received DCS packet. (Variable length, complete packet, from header to checksum.)

Message Type SK-FSK1 (Silent Knight 1)

<LF><CRC><0LLL><"SK-FSK1"><sequence#><Rreceiver#><Lline#>[#aaaaaaa|cccc_…cccc]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-FSK1	Message Type: Silent Knight 1
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaaa	The communicator's account number
cccc….cccc	ASCII representation of the received packet. (Variable length, complete packet, from header to checksum.)

Message Type SK-FSK2 (Silent Knight 2)

<LF><CRC><0LLL><"SK-FSK2"><sequence#><Rreceiver#><Lline#>[#aaaaaaa|cccc….cccc]<CR>

LF	Standard line feed character
CRC	Cyclical Redundancy Check number
0LLL	Length Field
SIA-FSK2	Message Type: Silent Knight 2
sequence#	The message sequence number
Rreceiver#	The Receiver number
Lline#	The line number
aaaaaaa	The communicator's account number
cccc….cccc	ASCII representation of the received packet. (Variable length, complete packet, from header to checksum.)

Appendix B - Packet Size and Packet Cyclic Redundancy Check Calculation

Packet Size

The packet size is available for additional protection. Implementors may choose to use this information in determining end of packet in addition to **<CR>** detection, however, size must be provided by the sender under all conditions. When the packet size is used to determine end of packet, scanning for **<CR>** should also be performed to avoid errors from bytes lost in the data stream.

Packet size is the number of bytes in the packet beginning with the first quote character of the Token ID up to, but not including, the **<CR>** character. The value is expressed in ACSII encoded hexadecimal notation.

For example, for the following message:

<LF>????00??"ADM-41E"0002|R01L02[#003456|7]<CR>

counting the characters up to but not including <CR>**]** yields 31 (1F Hex). Placed in the packet as hexadecimal encoded ASCII digits the result is:
<LF>????001F"ADM-41E"0002|R01L02[#003456|7]<CR>

Calculating Packet CRC

CRCs are based on treating bit strings as representations of polynomials with coefficients of 0 and 1 only. An n-bit message is regarded as the coefficient list for a polynomial with n terms, ranging from x ^ n-1 (high order bit) to x ^ 0 (low order bit).

For example, 110001 has 6 bits and thus represents a six term polynomial with coefficients 1, 1, 0, 0, 0, and 1: x ^ 5 + x ^ 4 + x ^ 0.

When the polynomial code method is employed, the sender and receiver must agree upon a generator polynomial in advance. Both the high order and low order bits of the generator must be 1. The basic idea is that the polynomial represented by the check summed message is divided by the generator. If there is a remainder, then there has been a transmission error.

A 16 bit generator polynomial which has been widely implemented in data transfer protocols (such as XMODEM CRC) is:
X ^ 16 + X ^ 15 + X ^ 2 + 1

This polynomial is called CRC-16. It catches all single and double errors, all errors with an odd number of bits, all burst errors of 16 bits or less, 99.997% (1/32768 chance of failure) of 17 bit burst errors, and 99.998% of 18 bit and longer burst errors (1/65536 chance of failure).

Polynomial arithmetic is done modulo 2, according to the rules of algebraic field theory, and therefore should be performed with simple exclusive-ORs. Floating point is not required.

For example the following message:

<LF>????001F"SIA-DCS"2034R99L88[#1234|NBA2]<CR>

yields a CRC value of 93FA Hex for the characters "SIA-DCS" through]. When added to the Cyclic Redundancy Check field the result would be:

<LF>93FA001F"SIA-DCS"2034R99L88[#1234|NBA2]<CR>

Calculation Routines
Two calculation routines are shown below, which each yield the same results. The second method of CRC calculation greatly speeds CRC processing, but does require a 512 byte table. The choice of method must depend upon the resources of the implementor.

Calculation Method 1
The following C Language program illustrates CRC calculation. It can be compiled as shown with many compilers to demonstrate the CRC process.

```
/* THESE INCLUDES FOR MICROSOFT C 5.1 */
#include "stdio.h"
#include "stdlib.h"

/* FORWARDS */
unsigned int calcCRC(unsigned CRC, int ch);

void main(void)
{
  unsigned int CRC;            /* 16 BIT CRC RESULT */
  int count, ch;
  char *ptr, str[1024];

  CRC = 0;
  count = 0;
  printf("Input string for CRC calculation (<CR> to end): ");
  ptr = gets(str);

  while (ch = *ptr++)
     {
        CRC = calcCRC(CRC, ch);   /* CALL CRC FUNCTION BELOW */
        printf("\nChar %c [%2.2x] CRC is %4.4x, %2.2x count",
                 (ch > 32) ? ch : '.', ch, CRC, ++count);
     }
}

unsigned int calcCRC(unsigned CRC, int ch)
{
  int i;
  unsigned char temp;
```

```
  temp = (unsigned char)ch;  /* TREAT LOCALLY AS UNSIGNED */
  for (i = 0; i < 8; i++)    /* DO 8 BITS */
    {
        temp ^= CRC & 1;              /* PROCESS LSB */
        CRC >>= 1;                    /* SHIFT RIGHT */
        if (temp & 1)
                CRC ^= 0xA001;        /* IF LSB SET,ADD FEEDBACK */
        temp >>= 1;                   /* GO TO NEXT BIT */
    }
 return CRC;
}
```

Calculation Method 2

Alternatively, the routine calcCRC could be replaced by the following, faster routine:

```
void calcCRC2(unsigned int CRC, int ch)
{
    static unsigned int crcTable[]= {

    /* DEFINE THE FIRST ORDER POLYTNOMIAL TABLE */
    0x0000,0xc0c1,0xc181,0x0140,0xc301,0x03c0,0x0280,0xc241,
    0xc601,0x06c0,0x0780,0xc741,0x0500,0xc5c1,0xc481,0x0440,
    0xcc01,0x0cc0,0x0d80,0xcd41,0x0f00,0xcfc1,0xce81,0x0e40,
    0x0a00,0xcac1,0xcb81,0x0b40,0xc901,0x09c0,0x0880,0xc841,
    0xd801,0x18c0,0x1980,0xd941,0x1b00,0xdbc1,0xda81,0x1a40,
    0x1e00,0xdec1,0xdf81,0x1f40,0xdd01,0x1dc0,0x1c80,0xdc41,
    0x1400,0xd4c1,0xd581,0x1540,0xd701,0x17c0,0x1680,0xd641,
    0xd201,0x12c0,0x1380,0xd341,0x1100,0xd1c1,0xd081,0x1040,
    0xf001,0x30c0,0x3180,0xf141,0x3300,0xf3c1,0xf281,0x3240,
    0x3600,0xf6c1,0xf781,0x3740,0xf501,0x35c0,0x3480,0xf441,
    0x3c00,0xfcc1,0xfd81,0x3d40,0xff01,0x3fc0,0x3e80,0xfe41,
    0xfa01,0x3ac0,0x3b80,0xfb41,0x3900,0xf9c1,0xf881,0x3840,
    0x2800,0xe8c1,0xe981,0x2940,0xeb01be01,0x2bc0,0x2a80,0xea41,
    0xee01,0x2ec0,0x2f80,0xef41,0x2d00,0xedc1,0xec81,0x2c40,
    0xe401,0x24c0,0x2580,0xe541,0x2700,0xe7c1,0xe681,0x2640,
    0x2200,0xe2c1,0xe381,0x2340,0xe101,0x21c0,0x2080,0xe041,
    0xa001,0x60c0,0x6180,0xa141,0x6300,0xa3c1,0xa281,0x6240,
    0x6600,0xa6c1,0xa781,0x6740,0xa501,0x65c0,0x6480,0xa441,
    0x6c00,0xacc1,0xad81,0x6d40,0xaf01,0x6fc0,0x6e80,0xae41,
    0xaa01,0x6ac0,0x6b80,0xab41,0x6900,0xa9c1,0xa881,0x6840,
    0x7800,0xb8c1,0xb981,0x7940,0xbb01,0x7bc0,0x7a80,0xba41,
    0xbe01,0x7ec0,0x7f80,0xbf41,0x7d00,0xbdc1,0xbc81,0x7c40,
    0xb401,0x74c0,0x7580,0xb541,0x7700,0xb7c1,0xb681,0x7640,
    0x7200,0xb2c1,0xb381,0x7340,0xb101,0x71c0,0x7080,0xb041,
    0x5000,0x90c1,0x9181,0x5140,0x9301,0x53c0,0x5280,0x9241,
    0x9601,0x56c0,0x5780,0x9741,0x5500,0x95c1,0x9481,0x5440,
    0x9c01,0x5cc0,0x5d80,0x9d41,0x5f00,0x9fc1,0x9e81,0x5e40,
    0x5a00,0x9ac1,0x9b81,0x5b40,0x9901,0x59c0,0x5880,0x9841,
    0x8801,0x48c0,0x4980,0x8941,0x4b00,0x8bc1,0x8a81,0x4a40,
    0x4e00,0x8ec1,0x8f81,0x4f40,0x8d01,0x4dc0,0x4c80,0x8c41,
    0x4400,0x84c1,0x8581,0x4540,0x8701,0x47c0,0x4680,0x8641,
```

```
0x8201,0x42c0,0x4380,0x8341,0x4100,0x81c1,0x8081,0x4040,
};

 unsigned char temp;

 temp = (unsigned char)ch;
 return (CRC >> 8) ^ (crcTable[temp ^ (CRC & 0xff)]);

}
```

Appendix C - Additional Examples

Long Messages

Generic Form:
<LF><CRC><0LLL><"ID"><seq#!seg#><|Rreceiver#><Lline#>[cccc.…cccc]<CR>

Example: SIA-2000 long message

The SIA-2000 example below is constructed so that <LF> and <CR> are understood. The segment number is shown as the value after the "!" character immediately following the sequence number. The final segment number shown has an additional leading zero. This is **NOT** required, but was added here to make the example more readable (by lining up the fields with above lines). Also note that the actual <CRC> value in the last line is represented in this example as ####.

####0061"SIA-S2K"0001!01|R01L01[0200000107900400008701000001020304050607080 90A0B0C0D0E0F1011121314151617]
####0061"SIA-S2K"0001!02|R01L01[18191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F303132333435363738393A3B]
####0061"SIA-S2K"0001!03|R01L01[3C3D3E3F404142434445464748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F]
####0061"SIA-S2K"0001!04|R01L01[606162636465666768696A6B6C6D6E6F707172737475767778797A7B7C7D7E7F80818283]
####0061"SIA-S2K"0001!05|R01L01[8485868788898A8B8C8D8E8F909192939495969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7]
####0061"SIA-S2K"0001!06|R01L01[A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACB]
####0061"SIA-S2K"0001!07|R01L01[CCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF]
####003D"SIA-S2K"0001!008R01L01[F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFFxxxx]

Security Industry Association
8405 Colesville Road, Ste. 500
Silver Spring, MD 20910
Phone: (703) 683-2075
~~**Fax: (703) 683-2469**~~

E-mail: Standards@SIAOnline.org