Extended Protocol Communications

Introduction

This document will describe the Extended Communications Protocol as implemented with InfoSight Marking System software. Extended Protocol is intended to provide secure communications with an intelligent host device. This document assumes familiarity with the ASCII character code as well as fundamental computer programming skills.

Disclaimer

This document is correct to the best knowledge of InfoSight Corporation which reserves the right to change the document's contents or the systems referred to at any time without prior notification.

Protocol Parameters

Extended Protocol can be implemented over legacy serial RS232 communications ports as well as more modern networking technology such as Ethernet. When communicating over a network, the Extended Protocol is typically implemented as a layer above TCP/IP.

When implemented via an RS232 serial connection, the port is opened with the following parameters:

Communications Parameters	
Baud Rate	Selectable 1200 - 19200
Data Bits	8
Parity	None
Start Bits	1
Stop Bits	1
Three Wire	RX, TX and GND

Serial implementations often use a form of handshaking to control the flow of data. Software handshaking (XON/XOFF) is implemented in some types of marking systems to control the flow of information. Hardware handshaking (DTR/DSR) is not implemented. No additional handshaking is used for network implementations beyond that provided by TCP/IP.

In legacy systems, communications with the marking system controller is accomplished via a Primary / Secondary arrangement with the Host being the Primary and the marker being the Secondary. The Secondary will only transmit in response to a message from the Primary.

Modern implementations of Extended Protocol are peer-to-peer and allow simultaneous bi-directional communications. In peer-to-peer implementations, the term Primary refers to the node that initiates a communications packet, while Secondary refers to the node that responds to that packet. Both nodes can simultaneously be Primary and Secondary for any given packet of data.

Note: All transmissions are in standard ASCII utilizing the following control characters:

Control Character Definitions			
Character	**Definition**	**Hex**	**Decimal**
SOH	Start of Header	0x01	1
STX	Start of Text	0x02	2
ETX	End of Text	0x03	3
CR	Carriage Return	0x0D	13
ACK	Acknowledge	0x06	6
NAK	Negative ACK	0x15	21
XOFF	Transmit Off	0x13	19
XON	Transmit On	0x11	17

Primary Data Format

SOH TYPE STX [DATA TEXT] ETX [BCC] CR

Where,
TYPE - A single printable ASCII character that defines the meaning and the contents of the message [DATA TEXT] field. Message types may be custom defined for certain applications as required. Standard message types are defined later.

[DATA TEXT] - An optional field which contains the actual data of the transmission. Some message types require no data since the "message" is conveyed by the TYPE character.

BCC - This is an optional field used to improve link reliability by providing fault detection. The BCC is computed by taking an eight bit addition of the TYPE and DATA TEXT characters and transmitting them as a three digit ASCII decimal number in the range 000 to 255. Refer to the example BCC computation later in this document.

Secondary Data Format (response)

The secondary will respond to the primary's transmission in one of two ways depending on whether errors were detected or not.

SOH TYPE ACK STX [DATA TEXT] ETX BCC CR

or

SOH TYPE NAK STX [DATA TEXT] ETX BCC CR

If no errors were detected in the reception of the packet, then the first response will be sent back to the primary. If any errors were detected (e.g. Parity, Framing, Overrun, BCC, Format, etc.) then the second (NAK) message will be sent. Note that the ACK message does not necessarily imply that the DATA TEXT field itself is correctly presented, just that no communications errors occurred.

The TYPE character will always be the same as the received TYPE.

The DATA TEXT field is optional and depends on the message TYPE. The BCC field will always be present in the response.

Retries

If the host does not receive a response from the I-Dent within three seconds, or it receives a NAK response, it should retransmit the entire packet. If, after three retries (four tries total), the host has not received a response, the host should declare the link to be "down".

Example BCC Computation

The following example is a typical transmission including the BCC.
To download the character string 'ABC123' to the currently assigned message buffer, send the following message.

SOH	1	STX	ABC123	ETX	141	CR

where '1' is the message TYPE and 141 is the BCC. The BCC is computed as follows (note all math shown in hexadecimal):

1) BCC = Message TYPE character + DATA TEXT characters.

	031H	1	- Message Type
	041H	A	\
	042H	B	|
	043H	C	| Message Text
	031H	1	|
	032H	2	|
+	033H	3	/
	18DH		

2) We are only interested in the lower eight bits of the sum, so we discard the first digit and keep the lower two. This results in a BCC of 8DH. Note that when performing the summation in an eight bit variable (e.g. unsigned char in 'C') that the most significant bits are automatically truncated. If the primary's programming language is incapable of doing eight bit addition, then the same result can be obtained by taking the MODULO 256 operation on a sixteen bit sum. The MODULO operation is division where the Remainder is kept and the Quotient is discarded.

3) Once the BCC value is obtained, it must be placed into the message packet after the ETX character. The BCC must be transmitted in its decimal ASCII form. The decimal equivalent of the hexadecimal value 8DH is 141 decimal. Converting the value 141 into three ASCII characters yields 031H, 034H and 031H. These three characters become the transmitted BCC.

The actual data transmitted (in hex) by the host for this message is:

001H	031H	002H	041H	042H	043H	031H
032H	033H	003H	031H	034H	031H	00DH

4) If the marker receives the message correctly, it will respond with the following message:

001H	031H	006H	002H	003H	030H	034H	039H	00DH

which equates to the following ASCII message:

SOH	1	ACK	STX	ETX	049	CR

Note that if the marker detected an error in receiving the message from the host it would respond with a NAK (015H) character in place of the ACK. The BCC would be unaffected by this since the ACK/NAK is not included in the BCC computation.

Conclusion

This concludes the description of the Extended Protocol for communications with InfoSight Corporation Marking System Software. Please refer all questions to the factory.

Extended Protocol Simulator Software for Windows is available for download.
Specifications are subject to change without notice.