# Crestron ISC symbol - Communications Format
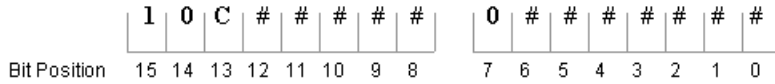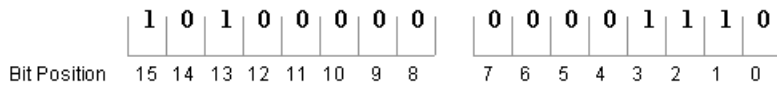
(Crestron HELP file)

**Digital data** is encoded in a 2-byte (16-bit) format, as follows:

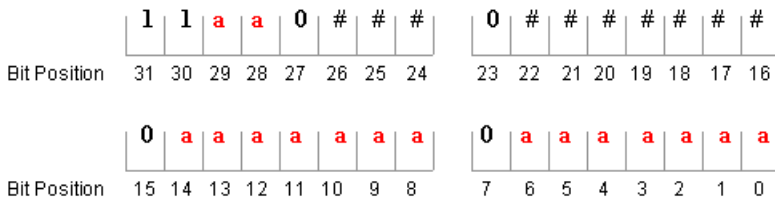| 1 | 0 | C | # | # | # | # | # | | 0 | # | # | # | # | # | # | # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bit Position

Where bit 15 (the most significant bit) is set to 1 and bit 14 is set to 0, denoting the start of new digital data. Bit 13 is set to the *complement* of the signal state. That is, if the signal is high, **C** equals 0; if the signal is low, **C** equals 1. Bit 7 (the high-order bit of byte 2) is set to 0. The 12-bit index of the signal (identified by # signs) is spread out over bytes 1 and 2, with leading zeros used as fills.

For example, consider a digital signal at index 14 (binary 0000 1110) that goes low. Following the format just described, the data would be sent as:
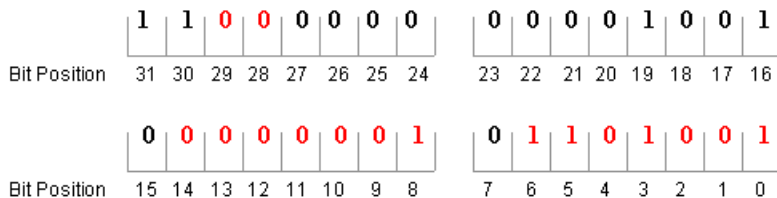
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bit Position

**Analog data** is encoded in a 4-byte (32-bit) format, as follows:

| 1 | 1 | a | a | 0 | # | # | # | | 0 | # | # | # | # | # | # | # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |

Bit Position

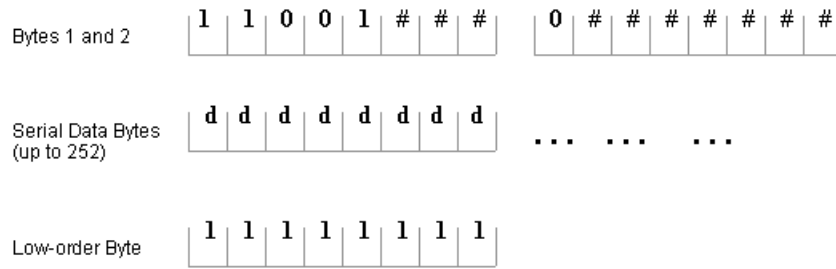| 0 | a | a | a | a | a | a | a | | 0 | a | a | a | a | a | a | a |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bit Position

Where bit 31 (the most significant bit) and bit 30 are both set to 1, denoting the start of new analog data. Bit 27 is set to 0, as are bits 23, 15, and 7 (the high-order bits of each byte). The 16-bit analog value of the signal (identified by red "a" letters) is spread out over bytes 1, 3 and 4, with leading zeros used as fills. The 10-bit index of the signal (identified by # signs) is spread out over bytes 1 and 2, with leading zeros used as fills.

For example, consider an analog signal at index 9 (binary 0000 1001) with a value of 233 (binary 1110 1001). Following the format just described, the data would be sent as:

| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |

Bit Position

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Bit Position

**Serial data**, unlike digital and analog data, which is encoded in a format of fixed length, serial data can be of variable length and is encoded as follows:

| Bytes 1 and 2 | 1 | 1 | 0 | 0 | 1 | # | # | # | | 0 | # | # | # | # | # | # | # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Serial Data Bytes (up to 252)

| d | d | d | d | d | d | d | d |
|---|---|---|---|---|---|---|---|

... ... ...

Low-order Byte

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

Where the five high-order bits are set to 1-1-0-0-1, denoting the start of new serial data. The high-order bit of byte 2 is set to 0. The 10-bit index of the signal (identified by # signs) is spread out over bytes 1 and 2, with leading zeros used as fills. This is followed by up to 252 bytes of serial data (identified by the "d" letters). Finally, the low-order byte is set to FFh (binary 1111 1111) denoting the end of the serial data.

10/2011