

# ***Jx Series User Guide***

## Contents

<b>Introduction</b>	
Jx Series Readers .....	5
Features .....	6
<b>Barcode Reading</b>	
Features .....	9
<b>Programming Commands</b>	
General Information .....	11
<b>Barcode</b>	
General .....	12
Code39 Options .....	14
Code39 Prefix .....	15
Code39 Suffix .....	15
Standard 2 of 5 Options .....	16
Standard 2 of 5 Prefix .....	17
Standard 2 of 5 Suffix .....	17
Industrial 2 of 5 Options .....	18
Industrial 2 of 5 Prefix .....	19
Industrial 2 of 5 Suffix .....	19
Interleaved 2 of 5 Options .....	20
Interleaved 2 of 5 Prefix .....	21
Interleaved 2 of 5 Suffix .....	21
Upca Options .....	22
Upca Prefix .....	23
Upca Suffix .....	23
Ean13 Options .....	24
Ean13 Prefix .....	25
Ean13 Suffix .....	25
Code128 Options .....	26
Code128 Prefix .....	27
Code128 Suffix .....	27
Codabar Options .....	28
Codabar Prefix .....	29
Codabar Suffix .....	29
Code 128 Options .....	30
Code 128 Prefix .....	31
Code 128 Suffix .....	31
<b>Barcode Masking</b>	
Mask # 1 Options .....	32
Mask # 2 Options .....	33

<b>Usid/Cac Barcode Programming</b>	
Usid 21 character Options.....	34
Usid 18 character / Cac Options .....	34
PDI Options .....	35
<b>Magstripe Programming</b>	
General Options .....	36
Track 1 Options .....	37
Track 2 Options .....	39
Track 3 Options .....	41
Track 1 Prefix.....	43
Track 1 Suffix.....	44
Track 2 Prefix.....	44
Track 2 Suffix.....	44
Track 3 Prefix.....	45
Track 3 Suffix.....	45
Track 1 Mask # 1.....	46
Track 1 Mask # 2.....	47
Track 2 Mask # 1.....	47
Track 2 Mask # 2.....	49
Track 3 Mask # 1.....	50
Track 3 Mask # 2.....	51
<b>Proximity/Iclass/Indala/PIV-II/Hid Long Programming</b>	
General Options .....	52
Format # 1 Setup.....	53
Format # 2 Setup.....	54
Format # 3 Setup.....	55
Format # 1 Mask.....	56
Format # 2 Mask.....	57
Format # 3 Mask.....	58
<b>Emulations (wiegand,aba,f2f,wand) Output Programming</b>	
Output Style .....	59
Wiegand Emulation Options .....	60
Magstripe Emulation Options.....	60
Wiegand Output for Barcodes.....	62
Wiegand Output for Magstripes .....	64
Wiegand Output for Rf cards, Format # 1 .....	66
Wiegand Output for Rf cards, Format # 2 .....	68
Wiegand Output for Rf cards, Format # 3 .....	70
Wiegand Output for Usid 21 character cards.....	72
Wiegand Output for Pdi numbers .....	74
Wiegand Output for Edi numbers .....	76
Wiegand Parity Overrides .....	78
<b>Speaker Programming</b>	
Speaker Setup .....	79
Speaker Duty Cycle .....	79

<b>Sense Inputs / Led Programming</b>	
Sense Input / Led Setup.....	81
Led Style Setup.....	82
General Led Settings.....	82
Led Good Read Settings .....	83
<b>Relay Programming</b>	
Relay Setup .....	84
<b>General Commands</b>	
Restart Reader .....	85
Transmit delay for multitech readers .....	85
Full Response for commands.....	85
Enable / Disable RF Processing.....	86
Process Cards meeting mask requirements only .....	86
<b>Serial Port Commands</b>	
General Setup.....	87
Multidrop Setup .....	87
Multidrop Transmit Delay .....	88
<b>External RS232 Input Commands</b>	
External Input Prefix.....	89
External Input Suffix.....	89
<b>Mifare Programming</b>	
General Read Setup.....	90
Mifare Block Read Setup.....	91
Mifare Key Setup.....	92
Mifare Power Level .....	92
<b>PIV-II Programming</b>	
Output Settings.....	93
<b>Serial Control Commands</b>	
Serial Commands .....	95

## JX Series Readers

Jx series readers are available to read any of the following types of cards:

- Barcode (1-D, Code39, I2/5, 2/5, Ind2/5, Codabar, Ean13,UPCA, Code128)
- Magnetic Stripe (Track1, Track2, Track3, ABA,ISO,IATA)
- Proximity (125Khz, Hid, Farpointe, Hid Long, Indala)
- Smart Card (13Mhz, Iclass, Mifare, PIV-II)

Readers are available to read one, two, or three card styles:

- Smart Slot Jx - reads barcode cards only
- Smart Mag Jx - reads magnetic stripe cards only
- Magbar Jx - reads barcode and magnetic stripe cards
- Smart Prox Jx - reads 125Khz prox or 13mhz smart cards, and/or barcode and magnetic stripe

Readers are available with the following interfaces:

- RS232
- RS485 (operates in 2-wire or 4-wire mode)
- Wiegand
- Magnetic stripe
- Usb
- Wand emulation
- F2f
- Tcp/ip (with our without POE)

## Features

### **Firmware Updating**

Jx readers can be flashed with new firmware, irregardless of the interface style. Readers which have a emulation style output (wiegand,aba, wand, f2f) can also operate in a serial mode, where commands can be sent, and the reader can be flashed with new firmware. Ibc offers a serial adapter for this purpose.

### **Enhanced Barcode Reading**

The barcode algorithms in the Jx series readers are more aggressive than those in the J series, allowing you to read difficult-to-read barcodes. Jx series readers can also read barcodes which are in close proximity to other writing on cards.

### **Magnetic Stripe Reading**

Jx readers can read magnetic stripe data from track1, track2, and/or track 3. Readers may be ordered to read one track, 2 tracks, or all three tracks. The readers support both hi-co and lo-co cards, and can decode data on any track using either the track 1 or track 2 (aba) character sets. Data from each track can be masked and concatenated together with data from other tracks prior to output or conversion to wiegand. Additional functionality such as prefix and suffix characters, and masking, allow tailoring of the data prior to output. Track 2 magstripes with both standard B start and C start can be read.

### **Programmable Output Styles for Readers with Emulation Outputs**

Emulation output readers (output code "G") can be programmed to output data in wiegand, track1, track2 (aba), wand emulation, f2f, alphanumeric wiegand, or wieaba.

### **Multiple Wiegand Output Formats**

Jx readers can output different wiegand and magstripe formats for barcode, magstripe, and prox/smart card data. All of these formats are programmable up to 192 bits.

### **Hid and Farpointe Card Reading**

SproxJx readers ordered with proximity input type "H" can read and process standard Hid cards and standard Farpointe cards. Card data can be decoded and re-encoded by the reader, or passed through undisturbed.

### **Iclass Card Reading**

Sproxjx readers ordered with input type "S" can read the Iclass serial number (CSN), both 32 and 64 bit from an Iclass card. Sproxj readers ordered with input type "I" can read the Hid Application Area "wiegand" data from Iclass cards.

## **Mifare Card Reading**

Sproxjx readers ordered with input type "S" can read the serial number (CSN), both 32 and 64 bit from Mifare cards. Sproxj readers ordered with input type "M" can read sector data from up to 3 sectors on a Mifare card.

## **Indala Card Reading**

Sproxjx readers ordered with input type "D" can read the proximity data from an Indala proximity card. The proximity card format must be specified on order and cannot be changed. Output formats however can be reprogrammed.

## **Hid Long Format Reading**

Sproxjx readers ordered with input type "L" can read the proximity data from an Hid "Long Format" proximity card. The long formats are card formats above 37 bits.

## **PIV-II Card Reading**

Sproxjx readers ordered with input type "P" can read the fasc-n data and expiration date data from a PIV-II formatted card. The reader can process any or all of the fields from the card and either pass thru the fasc-n data undisturbed, or reformat the data into a programmable wiegand, alphanumeric wiegand, or wieaba format.

## **USID/CAC Barcode Reading**

Jx readers with barcode can read and process the 1-D barcodes on Usid and Cac cards. Both the Pdi and Edi are supported, as well as the card instance identifier. 21 character and 18 character Usid cards are supported, and 18 character Cac cards are supported.

## **Programmable Output Timing**

The pulse width, data settle time, and timing between pulses is programmable for wiegand, alphanumeric wiegand, wieaba, and magnetic stripe emulation outputs. Adjustments to these timings are sometimes required for certain controllers.

## **Kronos Card Support**

Jx series readers support both code39 and I2/5 Kronos cards. The reader can autodiscriminate Kronos cards and process the Id number and optionally the facility code.

## **Sense Inputs**

Jx readers have two inputs which can be used as sense inputs for dps, rte, alarm, counters, or led control.

## **Led's**

Jx readers can contain up to 2 leds. The leds can be programmed to be controlled either internally or from external sources. Standard led configuration consists of 2 clear leds, one which turns red when activated and the other turns green. Bicolor led's will be supported in 2012.

**Relay**

Jx readers can be ordered with an internal relay. The relay can be triggered by a good read, a serial command, or from one of the two sense inputs. Standard relays are form-c, with a normally open, normally closed, and common line. Readers with POE can provide relay output power directly, or the relay can be left independent (option code "IR").

**Masking**

Each input (barcode, mag, prox, smart card) can be masked. Characters can be deleted or added, and substrings of the data can be selected, prior to output.

**Multidrop**

Readers with RS485 interface can be networked together onto one bus, using 2-wires or 4-wires. The 4-wire mode operates similar to an RS422 signal and can be used interchangeably.

**Tcp/IP**

Jx readers can be ordered with a tcp interface, and optionally, POE. If using POE, the power for an included internal relay can be taken directly from POE.

**Alphanumeric Display**

All Jx readers except readers which contain an rf input (prox, smart card, PIV) can be ordered with an 8 character alphanumeric display. Data can be shown on the display using serial commands.



## Barcode Reading

Jx series readers can read the following barcode symbologies:

- Code39
- Interleaved 2 of 5
- Standard 2 of 5
- Codabar
- Industrial 2 of 5
- Ean 13
- Upc A
- Code 128
- Code 93

### Read Direction

Barcodes can be read in either direction. A direction indicator ("f" for forward, "r" for reverse) can be pre-pended to the barcode data to show the direction the card was scanned.

### Masking

Masking (discussed later in the manual) can be used to select only portions of the barcode to process, and also add characters to the data when being processed. There are two possible masks which can be turned on. Each mask operates based on a barcode of a specific length.

### Check Digits

The reader can optionally check and verify check digits which may be included in the barcode for those symbologies that allow check digits. Note that Upca and Ean13 inherently use check digits and cannot be turned off. Check digit verification and transmission can be turned on or off for all of the other symbologies.

### Prefix and Suffix Characters

Up to 5 prefix characters and 5 suffix characters can be added to the barcode data. Prefix characters are especially useful to identify the data source on dual and tri-technology readers, when using readers with a serial output.

### Length Restrictions

The reader can be programmed to ignore barcodes which are not a specific length. You can enter in a different length requirement for each symbology, except Upca and Ean13.

### Turning Symbologies on and off

All symbologies are turned on by default. Each symbology can be turned on or off by command.

### **Codabar Clsi Conversion**

Codabar Clsi conversion can be turned on or off.

### **Usid/Cac Card Processing**

Jx readers can optionally process 18 and 21 character Usid cards, and 18 character Cac cards. For both cards, either the Pdi or Edi is extracted and converted to a base 10 number. The card instance identifier (on 18 character cards only) can also be appended to the data.

For Usid 21 character cards, you can elect to process (or not process) active cards, retired cards, reserve cards, sponsor cards, dependent cards, civilian cards, and reserve retired cards. The datatype taken from the 21 character card is the Pdi. For dependent cards, you can elect to take the dependent's Pdi, or the sponsor's Pdi.

For 18 character Usid and Cac cards, there are only two card type options - to include dependent cards, and include all other cards. The datatype for dependents can be the dependent's Edi, or the sponsor's Edi. The datatype for all other cards can be the Pdi, or Edi. You can also choose to append the card instance identifier to the output data.

### **Kronos Card Conversion**

Jx readers can autodiscriminate Kronos card formats if the Kronos Mode option is on. If on, then 17 digit code39 cards and 18 digit I2/5 cards are assumed to be in the Kronos format. There are two options - mode 1, using a 5 digit ID number, and mode 2, using a 9 digit ID number. The ID number is the 5 or 9 digits to the left of the facility code. For both modes, the site code is set to 0.

### **Alpha Delete**

One of the general barcode options is Alpha Delete. If on, then any alpha characters in the barcode are ignored. This is useful when reading a barcode that has an alpha, but the desired output is wiegand (wiegand cannot have non-numeric characters).

### **Read Normalization**

If there are two barcodes on the card in the same read path, or there is writing on the read path, then you can specify which one is the actual barcode to read. You specify the "smaller" or "larger" one - smaller or larger referring to the number of black "bars" the reader will see.

### **Good Read Beep**

You can turn on a good read beep to occur after reading a valid barcode. If you read a barcode which does not match length specifications, is not valid for a programmed mask, or cannot be decoded into a wiegand format because the data size is too large, then the reader will not beep.

### **Timeout**

The read timeout is programmable, from 1 to 9. The timeout is a relative amount of time after seeing the last black bar, that the reader then begins to decode the barcode data. The default timeout is 5. In some cases, adjusting the timeout can help to read barcodes on cards that have unusually wide white spaces, or writing very close to the barcode.

## Programming Commands

In the Jx series all programming commands start with an X, followed by a 3-digit command identifier and the command parameters. You can program the reader by sending the commands serially, or by encoding the commands into a magstripe or barcode and then swiping them through the reader. Readers which do not have a serial connection may still be programmed serially by using a programming adapter available through IBC. This includes readers with wiegand and aba outputs.

Serial commands must end with a carriage-return (hex 0d). Barcode and magstripe commands must start with a %U, then the command data. To program Track 2 only magstripe readers with a control card, encode the command on track 2 using the track 1 character set.

Note that command parameter values are loosely checked by the reader. Please verify the validity of all commands prior to entering them into the reader. Not all commands are verified.

The command descriptions in this manual contain spaces but these spaces are for readability only. Do not put the spaces in the commands.

## Barcode Programming Commands

Barcode programming consists of a number of commands. There is one General Setup command specifying global options for barcode processing. Each barcode symbology also has a general options command specific to that symbology, as well as commands to program prefixes and suffixes for each symbology. There are also a number of masking commands which can globally be used with barcode reading.

### General Barcode Options Command

X037 t n a m k b c s

t	= read timeout	1 thru 9, 5 is default
n	= normalize option	0=off, 1=small, 2=large
a	= alpha delete	0=disabled, 1=enabled
m	= masking	0=disabled, 1=enabled
k	= kronos mode	0=disabled, 1=mode 1, 2=mode 2
b	= good read beep	0=disabled, 1=enabled
c	= usic/cac option	0=disabled, 2=cac/usid only, * 1=accept cac/usid and all others

\*=command changed 10/30/09

### Notes

- Read timeout “t” is an adjustable amount of time before the reader times out when it believes it has seen the end of a barcode. It is best to leave this timeout at 5. Reduce the value to have the reader timeout quicker.
- If multiple barcodes are contained left to right on a card, or there is writing next to a barcode, set the “n” parameter to choose which one to read.
- Set “a” to a 1 to remove all alphas found in the barcode. This is used to mask out alphas in cases where they are not needed or not allowed, such as in a wiegand output reader.
- Set “m” to a 1 to turn barcode masking on. Masking is either turned on or off for all symbologies. It is not possible to mask only specific symbologies.
- Set “k” to 1 to enable Kronos mode 1, or set it to 2 to enable Kronos mode 2. Mode 1 creates a 2/5 string containing the first 2 digits of the site code followed by the last 5 digits of the id number. Mode 2 creates a 0/0 string containing a site code of 0 and the last 9 digits of the id. Code 39 and I2/5 formats are auto discriminated.
- Set “b” to a 1 to turn on the good read beep. Note that the good read beep will not beep if a barcode has been read but cannot be transformed into the proper output value. For example, if you read a 5 digit barcode “99999” and attempt to transmit that as an id number within a standard 26 bit wiegand format you will not get a good read beep since the number 99999 is too large for the programmed format.

- The “c” option is used to program the reading of code39 barcodes on the cac card and code39 barcodes on the usid card. if “c” is 0 then cac/usid cards are treated as ordinary cards and no special processing occurs. if “c” is a 2 then only cac/usid cards are read (18,21, or 24 character code39 cards). The card numbers are extracted and converted based on the “s” parameter. If “c” is a 1 then all cards are read normally except cards which contain cac/usid formats are automatically discriminated and processed based on the “s” parameter. Note that cac mode and kronos mode should not be turned on at the same time, as the two formats cannot always be autodiscriminated.

## Code39 General Setup

X001 e c t s d ll x p s

e	= enable code 39	0=disabled, 1=enabled
c	= check digit	0=disabled, 1=enabled
t	= transmit check digit	0=disabled, 1=enabled
s	= transmit start/stop	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= length	restrict length of barcode to ll
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read code39 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data.
- Note that the prefix can be up to 5 characters in length, and is prepended to the code39 barcode data after masking has taken place, if any.
- If c is set to a 1, then the code39 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If s is set to a 1, then the code 39 start/stop characters (\*) are sent along with the barcode data.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the code39 barcode read must match that length or it will be ignored. The length does not include start and stop characters, or the check digit. Length checking occurs prior to kronos or cac/usid checking, so specific kronos or cac/usid cards can be ignored based on the length parameter.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a code39 barcode. It is suggested that this parameter be left at the default of 2.

### Code39 Barcode Prefix

X016 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

#### notes

- Up to 5 prefix characters can be programmed for code39 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the code39 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

### Code39 Barcode Suffix

X017 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

#### notes

- Up to 5 suffix characters can be programmed for code39 barcodes. If the suffix option is turned on, then these suffix characters are appended to the code39 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Standard 2 of 5 General Setup

X002 e c t d ll x p s

e	= enable 2of5	0=disabled, 1=enabled
c	= check digit	0=disabled, 1=enabled
t	= transmit check digit	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= length	restrict length of barcode to ll
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read standard 2 of 5 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the 2 of 5 barcode data after masking has taken place, if any.
- If c is set to a 1, then the 2 of 5 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the 2 of 5 barcode read must match that length or it will be ignored.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a 2 of 5 barcode. It is suggested that this parameter be left at the default of 2.



## Standard 2 of 5 Barcode Prefix

X020 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for 2 of 5 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the 2 of 5 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix chracters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Standard 2 of 5 Barcode Suffix

X021 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for 2 of 5 barcodes. If the suffix option is turned on, then these suffix characters are appended to the 2 of 5 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065.
- A value of 000 means the end of the suffix. The reader will not process any suffix chracters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Industrial 2 of 5 General Setup

X003 e c t d ll x p s

e	= enable ind 2of5	0=disabled, 1=enabled
c	= check digit	0=disabled, 1=enabled
t	= transmit check digit	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= length	restrict length of barcode to ll
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read industrial 2 of 5 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the ind 2 of 5 barcode data after masking has taken place, if any.
- If c is set to a 1, then the ind 2 of 5 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the ind 2 of 5 barcode read must match that length or it will be ignored.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a ind2of5 barcode. It is suggested that this parameter be left at the default of 2.

## Industrial 2 of 5 Barcode Prefix

X030 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for ind2of5 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the ind2of5 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Industrial 2 of 5 Barcode Suffix

X031 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for ind2of5 barcodes. If the suffix option is turned on, then these suffix characters are appended to the ind2of5 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Interleaved 2 of 5 General Setup

X004 e c t d ll x p s

e	= enable I2of5	0=disabled, 1=enabled
c	= check digit	0=disabled, 1=enabled
t	= transmit check digit	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= length	restrict length of barcode to ll
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read I2of5 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the I2of5 barcode data after masking has taken place, if any.
- If c is set to a 1, then the I2of5 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the I2of5 barcode read must match that length or it will be ignored.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a I2of5 barcode. It is suggested that this parameter be left at the default of 2.

## Interleaved 2 of 5 Barcode Prefix

X018 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for I2of5 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the I2of5 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Interleaved 2 of 5 Barcode Suffix

X019 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for I2of5 barcodes. If the suffix option is turned on, then these suffix characters are appended to the I2of5 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## UPC-A General Setup

X005 e d x p s

e	= enable I2of5	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read UPC-A barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the UPC-A barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a UPC-A barcode. It is suggested that this parameter be left at the default of 2.

## UPC-A Barcode Prefix

X026 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for UPC-A barcodes. If the prefix option is turned on, then these prefix characters are prepended to the UPC-A barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## UPC-A Barcode Suffix

X027 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for UPC-A barcodes. If the suffix option is turned on, then these suffix characters are appended to the UPC-A barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## EAN-13 General Setup

X006 e d x p s

e	= enable EAN13	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read EAN13 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the EAN13 barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an “f” for forward scan, or “r” for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a EAN-13 barcode. It is suggested that this parameter be left at the default of 2.



## EAN13 Barcode Prefix

X028 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for EAN13 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the EAN13 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## EAN13 Barcode Suffix

X029 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for EAN13 barcodes. If the suffix option is turned on, then these suffix characters are appended to the EAN13 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Code128 General Setup

X007 e c t d ll x p s

e	= enable Code128	0=disabled, 1=enabled
c	= check digit	0=disabled, 1=enabled
t	= transmit check digit	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= restrict length	00=any, ll=restrict length
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read Code128 barcodes
- c parameter is set to a 1 if check digit validation is required
- t parameter is set to a 1 if check digit is to be transmitted
- Set ll to 00 to allow any length (default) or to a 2 digit length to restrict the length
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a Code 128 barcode. It is suggested that this parameter be left at the default of 2.

## Code128 Barcode Prefix

X024 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for Code128 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Code128 Barcode Suffix

X025 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for Code128 barcodes. If the suffix option is turned on, then these suffix characters are appended to the barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Codabar General Setup

X008 e c t s d ll x i p s

e	= enable Codabar	0=disabled, 1=enabled
c	= check digit	0=disabled, 1=enabled
t	= transmit check digit	0=disabled, 1=enabled
s	= transmit start/stop	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= restrict length	00=any, ll=restrict length
x	= extra bars	# extra bars allowed
i	= clsi	0=disabled, 1=enabled
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read Codabar barcodes
- c parameter is set to a 1 if check digit validation is required
- t parameter is set to a 1 if check digit is to be transmitted
- Set s to a 1 to transmit the start and stop characters
- Set i to a 1 to enable clsi interpretation
- Set ll to 00 to allow any length (default) or to a 2 digit length to restrict the length
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a Codabar barcode. It is suggested that this parameter be left at the default of 2.

## Codabar Barcode Prefix

X022 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii chracter, 000 denotes end
bbb= prefix character 2	3 digit ascii chracter, 000 denotes end
ccc= prefix character 3	3 digit ascii chracter, 000 denotes end
ddd= prefix character 4	3 digit ascii chracter, 000 denotes end
eee= prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for Codabar barcodes. If the prefix option is turned on, then these prefix characters are prepended to the barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Codabar Barcode Suffix

X023 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii chracter, 000 denotes end
bbb= suffix character 2	3 digit ascii chracter, 000 denotes end
ccc= suffix character 3	3 digit ascii chracter, 000 denotes end
ddd= suffix character 4	3 digit ascii chracter, 000 denotes end
eee= suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for Codabar barcodes. If the suffix option is turned on, then these suffix characters are appended to the barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Code 93 General Setup

X126 e d ll x p s

e	= enable Code 93	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= restrict length	00=any, ll=restrict length
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

### notes

- e parameter must be 1 in order to read Code 93 barcodes
- Set ll to 00 to allow any length (default) or to a 2 digit length to restrict the length
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a Code 93 barcode. It is suggested that this parameter be left at the default of 2.

## Code 93 Barcode Prefix

X127 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for Code 93 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Code 93 Barcode Suffix

X128 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for Code 93 barcodes. If the suffix option is turned on, then these suffix characters are appended to the barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

**Barcode Mask #1 Command - Part 1**

X033 ll p1 ll1 p2 ll2

ll	= length	length to match, 00 = any length
p1	= position 1	start position, 00 means ll1 is a constant
ll1	= length/constant 1	length,000=take remaining, or constant value
p2	= position 2	
ll2	= length/constant 2	

**notes**

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the barcode (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Barcode Mask #1 Command - Part 2**

X034 p3 ll3 p4 ll4 p5 ll5

p3	= position 1	parameters same as above
ll3	= length/constant 1	
p4	= position 2	
ll4	= length/constant 2	
p5	= position 2	
ll5	= length/constant 2	

**notes**

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the barcode (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.



## Barcode Mask #2 Command - Part 1

X035 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length, 000=take remaining, or constant value
p2 = position 2	
ll2 = length/constant 2	

### notes

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the barcode (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

## Barcode Mask #2 Command - Part 2

X036 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

### notes

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the barcode (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

## USID/CAC Barcode Programming

Jx readers can read and decode 24 character and 21 character Usid cards, 18 character Usid cards, and 18 character Cac cards. Note that Usid/Cac must be explicitly turned on in the barcode general options.

### Usid21 Processing Options

X104 1 2 3 4 5 6 7 8

1= Process active type A	0=no, 1=yes
2= Process retired type B	0=no, 1=yes
3= Process reserve Type C	0=no, 1=yes
4= Process sponsor Type D	0=no, 1=yes
5= Process dependent type E,F,G,H	0=no, 1=yes
6= Process civilian type K,L,M	0=no, 1=yes
7= Process retired reserve type I,J	0=no, 1=yes
8= Data type for dependents	0=dependent pdi, 1=sponsor pdi

#### notes

- To process a particular card type, the type must be explicitly turned on.
- The data type used for dependent cards can be the dependent's pdi, or the sponsor's pdi.

### Usid18/Cac Processing Options (firmware prior to May 20, 2011)

X105 1 2 3 4

1= Process all except dependents	0=no, 1=yes
2= Process dependents	0=no, 1=yes
3= datatype to process (non dependent)	0=pdi, 1=edi
4= datatype to process (dependent)	0=dependent edi, 1=sponsor's edi

#### notes

- To process a particular card type, the type must be explicitly turned on.
- The data type used for dependent cards can be the dependent's pdi, or the sponsor's pdi.

## Usid18/Cac Processing Options (firmware after May 20, 2011)

X105 1 2 3 4 5 6

1= Process all except dependents	0=no, 1=yes
2= Process dependents	0=no, 1=yes
3= datatype to process (non dependent)	0=pdi, 1=edi
4= datatype to process (dependent)	0=dependent edi, 1=sponsor's edi
5= append card instance identifier	0=no, 1=yes
6= transpose card instance identifier	0=no, 1=yes

### notes

- To process a particular card type, the type must be explicitly turned on.
- The data type used for dependent cards can be the dependent's edi, or the sponsor's edi.
- If parameter 5 is yes (append card instance), then the card instance (CI) is read and processed. There are two ways to process the CI. The first method decodes the CI as a 3-digit number, representing the decimal value of the ascii CI. For example, a CI of "A" is converted to a decimal 065. The second method assigns a decimal value to the CI based on it's position of the 36 possible CI values, i.e "0" = 0, "A" = 10, "Z" = 35. This value is decoded into a 2 digit number. To use the second method, set parameter 6 to "yes". The CI value will then be appended to the EDI or PDI number. For non-serial readers (wiegand, magstripe) the appropriate fields must be set up to extract the proper lengths for the PDI or EDI. The remaining characters will comprise of the CI, which should be put into the Issue Code field for subsequent transmission.
- For wiegand outputs, the extracted ID data is placed into the ID field.
- For wiegand output transmissions, we suggest using 34 bits, no parity, for the ID field. Transmit only the ID field. If the CI append is on, then 42 bits will be transmitted in total. Usid18/Cac Processing Options (firmware after May 20, 2011)

## PDI Output Options (firmware after June 26, 2011)

X120 1 2 3

1= PDI Length	0=9, 1=10
2= Append 000 for an Issue Code	0=no, 1=yes
3= Append 00 for an Issue Code	0=no, 1=yes

### notes

- The output PDI length can be either 9 or 10. Only 9 digits are needed, however 10 can be selected to provide compatibility in systems that are also reading EDI's.
- Either 00 or 000 can be appended to the PDI data, to provide compatibility with 18 character cards that process the CI. In this case, the 00 will be treated as the Issue Code.

## Magstripe Programming Commands

Magstripe programming consists of a number of commands. There is one General Setup command specifying global options for magstripe processing. Each Track also has a general options command specific to that track, as well as commands to program prefixes and suffixes for each track.

### General Magstripe Options Command

X045 1 2 3 0 sss

1 = tk1 required	0=no, 1=yes		
2 = tk2 required	0=no, 1=yes		
3 = tk3 required	0=no, 1=yes		
0 = concatenation order	1=1,2,3	2=1,3,2	3=2,1,3
	4=2,3,1	5=3,1,2	6=3,2,1
sss = seperator character	3 digit ascii char value, 000=none		

### notes

- Depending on the model of the reader, up to three tracks of magstripe data can be read from a card.
- The "required" parameters are set to a 1 if you want to ignore all reads which do not have a valid decode on those tracks which you specified were required.
- The concatenation order specifies the order that the decoded magstripe data is to be concatenated before processing. For example, if you read a card with tracks 1,2, and 3 and want them in that order, then you specify a concatenation order value of 1.
- If you are reading multiple tracks of data, you can place a separator between the different tracks. The separator is entered in as a 3 digit ascii decimal value, i.e 065 is an "A". Use a value of 000 to specify no separator character.
- \*\*\*\* Please Note - Track3 processing is not available at this time.

## Magstripe Track 1 settings - Part 1

X039 e c y s l d a m b p x

e	= track 1 enabled	0=no, 1=yes
c	= use track 1 character set	0=no, 1=yes
y	= c start	0=no, 1=yes
s	= include start/stop	0=no, 1=yes
l	= include lrc	0=no, 1=yes
d	= direction indicator	0=no, 1=yes
a	= alpha delete	0=no, 1=yes
m	= masking	0=off, 1=on
b	= good read beep	0=off, 1=on
p	= prefix	0=off, 1=on
s	= suffix	0=off, 1=on

### notes

- Setting “e” to a 1 enables the processing of track 1 data. The “enable” is not the same as the “required” setting in the general magstripe options command. The required option means that the track is required and must be properly decoded and valid. The enabled option means to process the track data after decoding. For example, you may wish to read and process data read on track 2 of a magstripe card, but only want to process the track 2 data if there is also valid data on track 1. To do this, you set both tracks 1 and 2 as required, but only enable track 2.
- The “c” option specified whether the track 1 character set is to be used when decoding the magstripe track.
- Setting “y” to a 1 tells the reader to only process the data if the data starts with a c start rather than the traditional b start. This applies only to the track 2 character set.
- If “s” is a 1, then the start and stop characters are included with the decoded data.
- If “l” is set to a 1 then the lrc character is included with the decoded data.
- If “d” is a 1, then a direction indicator (f for forward, r for reverse scan) is prepended to the decoded data.
- If “a” is a 1 then all alphas are deleted from the decoded data.
- If “m” is a 1 then the magstripe data is masked after it is read.
- If “b” is set to a 1 then the reader makes a beep with a proper decode.
- If “p” is set to a 1 then up to 5 programmable prefix characters can be prepended to the decoded data.
- If “s” is set to a 1 then up to 5 programmable suffix characters can be appended to the decoded data.

## Magstripe Track 1 settings - Part 2

X040 ll ss

ll = track 1 length

ll=length, 00=any length

ss = separator char translate

3 digit ascii value, 000=no translate

### notes

- If ll is set to 00, then any length track 1 magstripe may be read. If ll is set to a specific length, then only track 1 data with that length will be read.
- Set sss to the 3-digit decimal value of any ascii character that you want to use to replace any separator characters in the magstripe data.

**Magstripe Track 2 settings - Part 1**

X041 e c y s l d a m b p x

e	= track 2 enabled	0=no, 1=yes
c	= use track 1 character set	0=no, 1=yes
y	= c start	0=no, 1=yes
s	= include start/stop	0=no, 1=yes
l	= include lrc	0=no, 1=yes
d	= direction indicator	0=no, 1=yes
a	= alpha delete	0=no, 1=yes
m	= masking	0=off, 1=on
b	= good read beep	0=off, 1=on
p	= prefix	0=off, 1=on
s	= suffix	0=off, 1=on

**notes**

- Setting “e” to a 1 enables the processing of track 2 data. The “enable” is not the same as the “required” setting in the general magstripe options command. The required option means that the track is required and must be properly decoded and valid. The enabled option means to process the track data after decoding. For example, you may wish to read and process data read on track 2 of a magstripe card, but only want to process the track 2 data if there is also valid data on track 1. To do this, you set both tracks 1 and 2 as required, but only enable track 2.
- The “c” option specified whether the track 1 character set is to be used when decoding the magstripe track.
- Setting “y” to a 1 tells the reader to only process the data if the data starts with a c start rather than the traditional b start. This applies only to the track 2 character set.
- If “s” is a 1, then the start and stop characters are included with the decoded data.
- If “l” is set to a 1 then the lrc character is included with the decoded data.
- If “d” is a 1, then a direction indicator (f for forward, r for reverse scan) is prepended to the decoded data.
- If “a” is a 1 then all alphas are deleted from the decoded data.
- If “m” is a 1 then the magstripe data is masked after it is read.
- If “b” is set to a 1 then the reader makes a beep with a proper decode.
- If “p” is set to a 1 then up to 5 programmable prefix characters can be prepended to the decoded data.
- If “s” is set to a 1 then up to 5 programmable suffix characters can be appended to the decoded data.

## Magstripe Track 2 settings - Part 2

X042 ll ss

ll = track 2 length

ss = separator char translate

ll=length, 00=any length

3 digit ascii value, 000=no translate

### notes

- If ll is set to 00, then any length track 2 magstripe may be read. If ll is set to a specific length, then only track2 data with that length will be read.
- Set sss to the 3-digit decimal value of any ascii character that you want to use to replace any separator characters in the magstripe data.



**Magstripe Track 3 settings - Part 1 (RFU - not currently implemented)**

X043 e c y s l d a m b p x

e	= track 3 enabled	0=no, 1=yes
c	= use track 1 character set	0=no, 1=yes
y	= c start	0=no, 1=yes
s	= include start/stop	0=no, 1=yes
l	= include lrc	0=no, 1=yes
d	= direction indicator	0=no, 1=yes
a	= alpha delete	0=no, 1=yes
m	= masking	0=off, 1=on
b	= good read beep	0=off, 1=on
p	= prefix	0=off, 1=on
s	= suffix	0=off, 1=on

**notes**

- Setting “e” to a 1 enables the processing of track 3 data. The “enable” is not the same as the “required” setting in the general magstripe options command. The required option means that the track is required and must be properly decoded and valid. The enabled option means to process the track data after decoding. For example, you may wish to read and process data read on track 3 of a magstripe card, but only want to process the track 3 data if there is also valid data on track 1. To do this, you set both tracks 1 and 3 as required, but only enable track 3.
- The “c” option specified whether the track 1 character set is to be used when decoding the magstripe track.
- Setting “y” to a 1 tells the reader to only process the data if the data starts with a c start rather than the traditional b start. This applies only to the track 2 character set.
- If “s” is a 1, then the start and stop characters are included with the decoded data.
- If “l” is set to a 1 then the lrc character is included with the decoded data.
- If “d” is a 1, then a direction indicator (f for forward, r for reverse scan) is prepended to the decoded data.
- If “a” is a 1 then all alphas are deleted from the decoded data.
- If “m” is a 1 then the magstripe data is masked after it is read.
- If “b” is set to a 1 then the reader makes a beep with a proper decode.
- If “p” is set to a 1 then up to 5 programmable prefix characters can be prepended to the decoded data.
- If “s” is set to a 1 then up to 5 programmable suffix characters can be appended to the decoded data.

## Magstripe Track 3 settings - Part 2

X044 ll ss

ll = track 3 length

ss = separator char translate

ll=length, 00=any length

3 digit ascii value, 000=no translate

### notes

- If ll is set to 00, then any length track 3 magstripe may be read. If ll is set to a specific length, then only track1 data with that length will be read.
- Set sss to the 3-digit decimal value of any ascii character that you want to use to replace any separator characters in the magstripe data.
- Note that as of October 2009, track 3 has not been implemented yet.

## Track 1 Prefix

X055 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for Track 1 data. If the prefix option is turned on, then these prefix characters are prepended to the track 1 data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the track 1 data with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Track 1 Suffix

X058 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for Track 1. If the suffix option is turned on, then these suffix characters are appended to the track 1 data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the magstripe because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Track 2 Prefix

X056 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for Track 2 data. If the prefix option is turned on, then these prefix characters are prepended to the track 2 data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the track 2 data with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Track 2 Suffix

X059 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for Track 2. If the suffix option is turned on, then these suffix characters are appended to the track 2 data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the magstripe because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Track 3 Prefix

X057 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 prefix characters can be programmed for Track 3 data. If the prefix option is turned on, then these prefix characters are prepended to the track 1 data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix characters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the track 3 data with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

## Track 3 Suffix

X060 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

### notes

- Up to 5 suffix characters can be programmed for Track 3. If the suffix option is turned on, then these suffix characters are appended to the track 3 data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix characters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the magstripe because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

**Track 1 Mask #1 Command - Part 1**

X061 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length, 000 = take remaining, or constant value
p2 = position 2	
ll2 = length/constant 2	

**notes**

- There are 2 possible masks for track 1 data, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the track 1 data (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Track 1 Mask #1 Command - Part 2**

X062 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

**notes**

- There are 2 possible masks for track 1, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the track 1 data (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Track 1 Mask #2 Command - Part 1**

X063 ll p1 ll1 p2 ll2

ll = length

p1 = position 1

ll1 = length/constant 1

p2 = position 2

ll2 = length/constant 2

length to match, 00 = any length

start position, 00 means ll1 is a constant

length, 000 = take remaining, or constant value

**notes**

- There are 2 possible masks for track 1 data, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the track 1 data (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Track 1 Mask #2 Command - Part 2**

X064 p3 ll3 p4 ll4 p5 ll5

p3 = position 1

ll3 = length/constant 1

p4 = position 2

ll4 = length/constant 2

p5 = position 2

ll5 = length/constant 2

parameters same as above

**notes**

- There are 2 possible masks for track 1, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the track 1 data (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Track 2 Mask #1 Command - Part 1**

X065 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length, 000 = take remaining, or constant value
p2 = position 2	
ll2 = length/constant 2	

**notes**

- There are 2 possible masks for track 2 data, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the track 2 data (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Track 2 Mask #1 Command - Part 2**

X066 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

**notes**

- There are 2 possible masks for track 2, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the track 2 data (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.



**Track 2 Mask #2 Command - Part 1**

X067 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length, 000=take remaining, or constant value
p2 = position 2	
ll2 = length/constant 2	

**notes**

- There are 2 possible masks for track 2 data, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the track 2 data (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Track 2 Mask #2 Command - Part 2**

X068 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

**notes**

- There are 2 possible masks for track 2, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the track 2 data (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

### Track 3 Mask #1 Command - Part 1 (RFU, currently not implemented)

X069 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1 value	length,000=take remaining, or constant
p2 = position 2	
ll2 = length/constant 2	

#### notes

- There are 2 possible masks for track 3 data, which are executed if the length of the track 3 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 3 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the track 3 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

### Track 3 Mask #1 Command - Part 2 (RFU, currently not implemented)

X070 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

#### notes

- There are 2 possible masks for track 3, which are executed if the length of the track 3 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 3 mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the track 3 data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

## Track 3 Mask #2 Command - Part 1 (RFU, currently not implemented)

X071 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length, 000=take remaining, or constant value
p2 = position 2	
ll2 = length/constant 2	

### notes

- There are 2 possible masks for track 3 data, which are executed if the length of the track 3 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 3 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the track 3 data (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

## Track 3 Mask #2 Command - Part 2 (RFU, currently not implemented)

X072 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

### notes

- There are 2 possible masks for track 3, which are executed if the length of the track 3 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 3 mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the track 3 data (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

## Proximity/Iclass/Indala/PIV-II/Hid Long Input Commands

There are a number of commands to program the reading of proximity style rf cards. These commands apply to all rf cards read by the Jx series except Mifare cards. Mifare card reading and writing is covered in a separate section.

The reader can uniquely process up to three different card formats, including formatting and masking of the input data. In the "passthru" mode, there is no limit to the card formats accepted. The passthru mode is limited to 200 bits.

### Proximity Input General Setup Command

X073 n m b p s

n = input mode	1=wiegand 2=aba 3=wiegand and aba
m = masking	0=off, 1=on
b = good read beep	0=off, 1=on
p = parity check	0=off, 1=on (26&28 bit cards only)
s = passthru	0=off, 1=on

#### notes

- Set the "n" parameter to the type of card input you are reading.
- Iclass, PIV-II, Hid Long, Indala, and card serial number readers must have the "n" parameter set to a 1.
- Set "m" to a 1 if you want to perform masking on the decoded data. Does not apply to PIV-II reading.
- Set "b" to a 1 for a good read beep when the card is read.
- Set "p" to a 1 to turn on parity checking for 26 bit and 28 bit cards only.
- Set "s" to a 1 to pass thru proximity data undisturbed. s must be set to a 1 for PIV-II card processing.

## Prox Format # 1 Settings, Part 1

X074 bbb sss sl b bss bl y

bbb= total bits  
sss = site (facility) start  
sl = site length  
b = site is lsb 0=no, 1=yes  
bss = badge (id) start  
bl = badge length  
y = badge is lsb 0=no, 1=yes

### notes

- Set “bbb” to the total number of bits on the card for Format # 1. Set this to 00 if no site code is used.
- Set “sss” to the starting bit position of the site code. Note the first bit is considered bit # 1, not 0.
- Set “sl” to the number of bits included in the site field.
- Set “b” to a 1 if the bits are lsb (backwards) within the site code field.
- Set “bss” to the starting bit position of the id field.
- Set “bl” to the number of bits included in the id field.
- Set “y” to a 1 if the id field is lsb.

## Prox Format # 1 Settings, Part 2

X075 iss il b sc ic uc

iss = issue code start  
il = issue length  
b = issue code is lsb 0=no, 1=yes  
sc = # of site characters  
ic = # of id characters  
uc = # of issue characters

### notes

- Set “iss” to the starting bit position of the issue code. Set this to 00 if no issue code is used.
- Set “il” to the number of bits included in the issue code field.
- Set “b” to a 1 if the bits are lsb (backwards) within the issue code field.
- Set “sc” to the number of characters to encode the site into. Set to 00 to ignore the site.
- Set “ic” to the number of characters to encode the id into. Set to 00 to ignore the ID.
- Set “uc” to the number of characters to encode the issue code into. Set to 00 to ignore the issue code.

### Prox Format # 2 Settings, Part 1

X076 bbb sss sl b bss bl y

bbb= total bits

sss = site (facility) start

sl = site length

b = site is lsb 0=no, 1=yes

bss= badge (id) start

bl = badge length

y = badge is lsb 0=no, 1=yes

#### notes

- Set "bbb" to the total number of bits on the card for Format # 2. Set this to 00 if no site code is used.
- Set "sss" to the starting bit position of the site code. Note the first bit is considered bit # 1, not 0.
- Set "sl" to the number of bits included in the site field.
- Set "b" to a 1 if the bits are lsb (backwards) within the site code field.
- Set "bss" to the starting bit position of the id field.
- Set "bl" to the number of bits included in the id field.
- Set "y" to a 1 if the id field is lsb.

### Prox Format # 2 Settings, Part 2

X077 iss il b sc ic uc

iss = issue code start

il = issue length

b = issue code is lsb 0=no, 1=yes

sc = # of site characters

ic = # of id characters

uc = # of issue characters

#### notes

- Set "iss" to the starting bit position of the issue code. Set this to 00 if no issue code is used.
- Set "il" to the number of bits included in the issue code field.
- Set "b" to a 1 if the bits are lsb (backwards) within the issue code field.
- Set "sc" to the number of characters to encode the site into. Set to 00 to ignore the site.
- Set "ic" to the number of characters to encode the id into. Set to 00 to ignore the ID.
- Set "uc" to the number of characters to encode the issue code into.. Set to 00 to ignore the issue code.

## Prox Format # 3 Settings, Part 1

X078 bbb sss sl b bss bl y

bbb= total bits

sss = site (facility) start

sl = site length

b = site is lsb 0=no, 1=yes

bss= badge (id) start

bl = badge length

y = badge is lsb 0=no, 1=yes

### notes

- Set "bbb" to the total number of bits on the card for Format # 3. Set this to 00 if no site code is used.
- Set "sss" to the starting bit position of the site code. Note the first bit is considered bit # 1, not 0.
- Set "sl" to the number of bits included in the site field.
- Set "b" to a 1 if the bits are lsb (backwards) within the site code field.
- Set "bss" to the starting bit position of the id field.
- Set "bl" to the number of bits included in the id field.
- Set "y" to a 1 if the id field is lsb.

## Prox Format # 3 Settings, Part 2

X079 iss il b sc ic uc

iss = issue code start

il = issue length

b = issue code is lsb 0=no, 1=yes

sc = # of site characters

ic = # of id characters

uc = # of issue characters

### notes

- Set "iss" to the starting bit position of the issue code. Set this to 00 if no issue code is used.
- Set "il" to the number of bits included in the issue code field.
- Set "b" to a 1 if the bits are lsb (backwards) within the issue code field.
- Set "sc" to the number of characters to encode the site into. Set to 00 to ignore the site.
- Set "ic" to the number of characters to encode the id into. Set to 00 to ignore the ID.
- Set "uc" to the number of characters to encode the issue code into. Set to 00 to ignore the issue code.

## Prox Format # 1 Mask Command - Part 1

X080 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length, 000 = take remaining, or constant value
p2 = position 2	
ll2 = length/constant 2	

### notes

- There is a separate mask for each of the three programmable input formats.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the 3 data (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

## Prox Format # 1 Mask Command - Part 2

X081 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

### notes

- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the data (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.



## Prox Format # 2 Mask Command - Part 1

X082 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = position 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length, 000=take remaining, or constant value
p2 = position 2	
ll2 = length/constant 2	

### notes

- There is a separate mask for each of the three programmable input formats.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1, ll2 specify the number of characters to extract. If ll1, ll2 are 000 then the remainder of the 3 data (starting at the p1, p2 position) is extracted.
- If p1, p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

## Prox Format # 2 Mask Command - Part 2

X083 p3 ll3 p4 ll4 p5 ll5

p3 = position 1	parameters same as above
ll3 = length/constant 1	
p4 = position 2	
ll4 = length/constant 2	
p5 = position 2	
ll5 = length/constant 2	

### notes

- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p3, p4, and p5 specify the starting positions of the substrings, and ll3, ll4, and ll5 specify the number of characters to extract. If ll3, ll4, ll5 are 000 then the remainder of the data (starting at the p3, p4, or p5 position) is extracted.
- If p3, p4, or p5 is 00, then ll3, ll4, ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value. Prox Format # 1 Mask Command - Part 1

**Prox Format # 3 Mask Command - Part 2**

X084 ll p1 ll1 p2 ll2

ll = length

p1 = position 1

ll1 = length/constant 1

p2 = position 2

ll2 = length/constant 2

length to match, 00 = any length

start position, 00 means ll1 is a constant

length,000=take remaining, or constant value

**notes**

- There is a separate mask for each of the three programmable input formats.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the 3 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

**Prox Format # 3 Mask Command - Part 2**

X085 p3 ll3 p4 ll4 p5 ll5

p3 = position 1

ll3 = length/constant 1

p4 = position 2

ll4 = length/constant 2

p5 = position 2

ll5 = length/constant 2

parameters same as above

**notes**

- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii character which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value. Prox Format # 1 Mask Command - Part 1

## General Emulations Output Commands

### Emulation Output Style ( prior to version 1.5 )

X052 s

s = output type  
1=wiegand (including wieaba & alphawie)  
2=mag emulation (including f2f)  
3=wand emulation

#### notes

- Readers with output code "G" support wiegand, alphanumeric wiegand, wieaba, track2 emulation, track1 emulation, f2f, and wand emulation as outputs.
- All inputs (barcode, magstripe, rf) will be output using the same output style, except for rf card reading set in the passthru mode.

### Emulation Output Style (version 1.5+)

X052 s t u

s = output type for all rf cards  
1=wiegand (including wieaba & alphawie)  
2=mag emulation (including f2f)  
3=wand emulation

t = output type for barcode  
1=wiegand (including wieaba & alphawie)  
2=mag emulation (including f2f)  
3=wand emulation

u = output type for magstripe  
1=wiegand (including wieaba & alphawie)  
2=mag emulation (including f2f)  
3=wand emulation

#### notes

- Readers with output code "G" support wiegand, alphanumeric wiegand, wieaba, track2 emulation, track1 emulation, f2f, and wand emulation as outputs.
- Each input (barcode, magstripe, rf) can be programmed to output to a different interface (wiegand, aba, etc.). Note that if rf card reading is set to passthru, these settings are ignored for rf cards.

## Wiegand Emulation Output Control

X054 o b p a b c d e

o = output type	0=standard wiegand 1=wieaba 2=alphanumeric wiegand
b = pulse to pulse timing	time in ms between pulses (1-5)
p = pulse timing	1=20us, 2=50us, 3=100us
a = site override for barcodes	0=disable, 1=enable
b = site override for magstripe	0=disable, 1=enable
c = site override for prox format #1	0=disable, 1=enable
d = site override for prox format #2	0=disable, 1=enable
e = site override for prox format #3	0=disable, 1=enable

### notes

- The default pulse timing is 50us with 1ms between pulses

## Magstripe Emulation Output Control (changed 1/14)

X053 c d t f cc bc ss pp ee

c = cstart	0=off, 1=one
d = must be 0	
t = character set	1=track1, 2=track2
f = f2f	0=off, 1=on
cc = clock time	cc*100us, max=20 (2 ms)
bc = interval time	bc*100us, max=20 (2 ms)
ss = data settle time	0-20 (*1us) data settle time
pp = preceding 0 chars	0-70
ee = trailing 0 chars	0-70

### notes

- This command was changed on 1/22/14 to facilitate faster clock frequencies. Firmware prior to 1/22/14 will use a different command. Consult the manual for your version for the appropriate command.
- Set c to a 1 to use a c start sentinel rather than a b start sentinel. Applies to the track 2 character set only.

- t can be set to output the data as a track 2 (aba) magstripe, or track 1.
- Set f to a 1 to transmit f2f rather than clock and data.
- The interval time is the time from clock up to the next clock down. For a 50% duty cycle, set cc and bc to the same value.
- The data settle time is programmable from 0us to 20us. The default is 2. Use a value of 0 to have clock and data change at approximately the same time (Neuron mode).
- The default clock/interval settings are 2,2 (200us clock, 200us interval) resulting in a 50% duty rate.

## Wiegand Output Programming Commands

The reader can transmit different wiegand output formats for each of the styles of cards read. There is a separate output format for barcode, magstripe, for each of the three programmed proximity input formats, and for Usid/Cac cards.

Each format is programmed in using 3 commands.

### Wiegand Output Settings for Barcode - Part 1

X046 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

#### notes

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted to generate the field.

### Wiegand Output Settings for Barcode - Part 2

X047 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

#### notes

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.

### Wiegand Output Settings for Barcode - Part 3

X048 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

#### notes

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.

## Wiegand Output Settings for Magstripe - Part 1

X049 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

### notes

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted for generate the field.
- All tracks of the magstripe which are enabled are concatenated together and then masked (if applicable) prior to generating the wiegand output.

## Wiegand Output Settings for Magstripe - Part 2

X050 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

### notes

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.



**Wiegand Output Settings for Magstripe - Part 3**

X051 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

**notes**

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.

### Wiegand Output Settings for RF cards Format # 1 - Part 1

X086 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

#### notes

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted to generate the field.

### Wiegand Output Settings for RF cards Format # 1 - Part 2

X087 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

#### notes

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.

**Wiegand Output Settings for RF cards Format # 1 - Part 3**

X088 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

**notes**

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.

## Wiegand Output Settings for RF cards Format # 2 - Part 1

X089 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

### notes

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted to generate the field.

## Wiegand Output Settings for RF cards Format # 2 - Part 2

X090 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

### notes

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.

## Wiegand Output Settings for RF cards Format # 2 - Part 3

X091 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

### notes

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.

**Wiegand Output Settings for RF cards Format # 3 - Part 1**

X092 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

**notes**

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted for generate the field.

**Wiegand Output Settings for RF cards Format # 3 - Part 2**

X093 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

**notes**

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.

### Wiegand Output Settings for RF cards Format # 3 - Part 3

X094 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

#### notes

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.

## Wiegand Output Settings for Usid 21 character cards - Part 1

X111 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

### notes

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted to generate the field.
- With 21 character Usid cards, the only field extracted from the card is the PDI, which is placed into the ID field with a length of 9 characters. The site and issue fields should be 0, and the id start set to 01, id length set to 09.

## Wiegand Output Settings for Usid 21 character cards - Part 2

X112 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

### notes

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.
- The default for Usid 21 character cards is 0 site and issue bits, no parity, 30 bits for the id. It is recommended not to use the site or issue bits (without an override) when reading Usid 21 character cards.



## Wiegand Output Settings for Usid 21 character cards - Part 3

X113 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

### notes

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.
- The default format for Usid 21 character cards is field 1=id, fields 2 and 3 are 0.

## Wiegand Output Settings for PDI number in Usid and Cac 18 character cards - Part 1

X114 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

### notes

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted to generate the field.
- With 18 character Usid and Cac cards, if the PDI is being read, then the PDI is extracted first resulting in a 9-digit number. If the card instance field is also extracted, it is appended to the 9 digit PDI as a 3 digit number, resulting in a data string of 12 digits. The default extractions for pdi only are site 0, id start 1, len 9, and issue 0. The default extractions for pdi + card instance is site 0, id start 1, len 9, issue start 10, len 3,

## Wiegand Output Settings for PDI number in Usid and Cac 18 character cards - Part 2

X115 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

### notes

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.
- The default for 18 character Usid and Cac cards is 0 site bits, no parity, 30 bits for the id, and 8 bits for the issue code. The Issue Code field contains the card instance.

**Wiegand Output Settings for PDI number in Usid and Cac 18 character cards - Part 3**

X116 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

**notes**

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.
- The default format for PDI data from 18 character cards is field 1=id, field 2=Issue, and field 3=0.

## Wiegand Output Settings for EDI number in Usid and Cac 18 character cards - Part 1

X117 ss ll ii il uu ul

ss =	site start position	position in decimalized string
ll =	site length	
ii =	id start position	position in decimalized string
il =	id length	
uu =	issue start position	position in decimalized string
ul =	issue length	

### notes

- There are 3 parts to this command. All 3 need to be entered to properly program the wiegand output structure.
- The start positions and lengths refer to the character positions which are being extracted to generate the field.
- With 18 character Usid and Cac cards, if the EDI is being read, then the EDI is extracted first resulting in a 10-digit number. If the card instance field is also extracted, it is appended to the 10 digit EDI as a 3 digit number, resulting in a data string of 13 digits. The default extractions for edi only are site 0, id start 1, len 10, and issue 0. The default extractions for edi + card instance is site 0, id start 1, len 10, issue start 11, len 3,

## Wiegand Output Settings for EDI number in Usid and Cac 18 character cards - Part 2

X118 sb ib ub a b c lpp rpp

sb =	number of site bits	0 thru 64, 00=ignore site
ib =	number of id bits	0 thru 64, 00=ignore id
ub =	number of issue bits	0 thru 64, 00=ignore issue
a =	site is lsb	0=no, 1=yes
b =	id is lsb	0=no, 1=yes
c =	issue is lsb	0=no, 1=yes
lpp =	left parity count	000=no parity
rpp =	right parity count	000=no parity

### notes

- Set the number of site, id, and issue bits to the number of bits used for the output.
- Set a,b, or c to a 1 if the output field is supposed to be lsb.
- Set lpp or rpp to the number of bits used to calculate parity. Set them to 000 to not generate parity.
- The default for the EDI from 18 character Usid and Cac cards is 0 site bits, no parity, 34 bits for the id, and 8 bits for the issue code. The Issue Code field contains the card instance.

### Wiegand Output Settings for EDI number in Usid and Cac 18 character cards - Part 3

X119 a b c 0000 ss

a = field 1	0=none, 1=site, 2=id, 3=issue
b = field 2	0=none, 1=site, 2=id, 3=issue
c = field 3	0=none, 1=site, 2=id, 3=issue
0000=site override	site override in hex
ss = special format number	00-99

#### notes

- The specific wiegand fields can be output in any order. It is not necessary to output the wiegand data in the order of site, id, issue which is the norm.
- Field 1 is the first field to be transmitted, field 2 the second, and field 3 the third. Set any of them to 0 to ignore that field.
- To override the site code set the site override to a non-zero value.
- The special format number refers to preprogrammed formats which are loaded in the reader. You can use these formats rather than program the format, if the format you need is available as one of the "special" formats.
- The default format for EDI data from 18 character cards is field 1=id, field 2=Issue, and field 3=0.

**Standard Wiegand Parity Overrides**

X123 b m 1 2 3 left parity override

b = barcode	0=even, 1=odd
m = magstripe	0=even, 1=odd
1 = rf format # 1	0=even, 1=odd
2 = rf format # 2	0=even, 1=odd
3 = rf format # 3	0=even, 1=odd

X124 b m 1 2 3 right parity override

b = barcode	0=odd, 1=even
m = magstripe	0=odd, 1=even
1 = rf format # 1	0=odd, 1=even
2 = rf format # 2	0=odd, 1=even
3 = rf format # 3	0=odd, 1=even

**notes**

- Standard wiegand parity is even on the left side and odd on the right side. Use this command to change the calculated parities used on wiegand outputs.

## Speaker Programming Commands

### Speaker Setup Command (firmware prior to 09/04/09)

X013 s f gg

s = startup beep	0=disabled, 1=enabled
f = frequency	1=low, 2=medium, 3=high pitch
gg= good read beep time	10ms increments (10ms thru 990ms)

#### notes

- With this command there are only 3 different frequencies that can be used for the speaker.

### Speaker Setup Command (firmware after 09/03/09)

X095 s fff gg

s = startup beep	0=disabled, 1=enabled
fff = frequency	001-255, default is 19
gg= good read beep time	10ms increments (10ms thru 990ms)

#### notes

- With this command you can specify whatever frequency you like for the speaker. As the number gets higher, the frequency gets higher.
- Because the speaker is a pizzo, certain frequencies will appear to be louder than others. The loudest frequency setting is 19, which is the default.

### Speaker Duty Cycle (firmware after 12/28/09)

X102 oo ff

oo= on cycles	01-15
ff = off cycles	01-15

#### notes

- To conserve power, the speaker can run less than 100% duty cycle. The defaults are on=01, off=03, or 25% duty cycle. Note that changing the duty cycle will also affect the perceived frequency.

## **Invalid Card Beep**

X129 x

x = 0 off

x = 1 on

### **notes**

- If x is set to 1, an invalid card beep will be heard whenever a card is read which does not match the programmed parameters (wiegand formats, etc..)



## Sense Inputs and LED Programming Commands

Readers with sense inputs can use the sense input lines for led control, dps, rte, or as counters.

### Sense Inputs/External Led's Setup

X009 i P i P

i = type	0=disabled
	1=input controls left led
	2=input controls right led
	3=door position sense (manual and auto)
	4=request to exit
	5=counter
	6=dps with alarm condition
P = polarity	0=normal, 1=reverse

### notes

- The first set of "iP" parameters refers to sense input #1, the 2nd set refers to sense input # 2
- Set i to the desired operation for the sense input. Each sense input can control either the left led, right led, or be used as a door position sense which can be manually queried or automatically transmitted, a request to exit, or as a counter.
- P specified the polarity for the sense input. 0 is normal meaning high is inactive, low is active. Set P to a 1 to reverse the polarity.

### Led Style for left bicolor led

X010 x

x = style:

0=red/green

1=off/red

2=off/green

#### notes

- If the left led is a bicolor led, the led can operate in three different on/off color patterns rather than the traditional off/red pattern. If set to 0, then “off” lights the led red and “on” lights the led green. If set to 2, the “off” turns off the led but “on” turns the led green.

### General Led Settings

X011 s t b gg s t b gg

s = style

0=internal 1=external control

t = startup

0=off, 1=on, 2=red(bi) 3=green(bi)

(2&3 refer to led1 only)

b = startup blink

0=none 1=blink 2=fast blink

gg= good read time

#### notes

- The leds can be controlled internally or externally (using the sense input lines)
- For internally controlled leds only, the state of the led(s) on startup can be off or on. For the left led which may be a bicolor led, it can also be ON red or ON green.
- Internally controlled leds can also be set to blink slowly or fast on startup and remain in that state.
- gg specifies the number of seconds the led should go ON for when a “good read” occurs
- Startup blink mode has not been implemented as of June 2011.

## Led Good Read settings

X012 a b

a = action led 1

0=on 1=off 2=blink on 3=blink off

4=fblink on 5=fblink off

b = action led 2

0=on 1=off 2=blink on 3=blink off

4=fblink on 5=fblink off

### notes

- Either led 1 or led 2 can be turned on or blinked after a good read.
- The leds are turned on or blinked for the amount of time specified in the gg parameter of the X011 command.
- Note that as of 10/09 the fast blink settings have not been implemented.

## Relay Commands

### Relay Setup

X032 a n l tt

a = autotrigger	0=off, 1=on
n = startup position	0=off, 1=on
l = autolock on sense	0=off, 1=on
tt = autotrigger time	time in seconds, max=99

### notes

- If autotrigger is on, then the relay is turned on for the autotrigger time whenever a good read occurs
- The relay position, when the reader is powered up (or restarted) can be programmed to be on or off.
- If autolock on sense is turned on, then the relay is turned off whenever a door is opened and then closed, even if there is time remaining on the relay cycle. To use this option, one of the sense inputs must be programmed for a dps (door position switch) operation.

## General Programming Commands

Readers with sense inputs can use the sense input lines for led control, dps, rte, or as counters.

### Restart Command

X0151

no parameters

#### notes

- This command causes the reader to restart.

### Transmit time delay for multitech readers

X038 tt

tt = wait and delay time                      00-99 (\*10ms)

#### notes

- When using combination readers that may have proximity and magstripe or barcode on the same card, this command allows you to program a wait time that the reader will wait after transmitting prox, barcode, or mag data, before processing one of the other inputs. This is designed to put a delay of time between two successive outputs.

### Full Response for Commands

X107 t

t = full response                                      0=off, 1=on

#### notes

- Jx readers can emulate J series readers, where no response is returned when sending commands. If the full response setting is on, the the Jx reader will always respond with OK or NOK when a command is sent to it. Note that some commands are loosely checked by the reader, so the receipt of an OK does not in itself mean the command format was ok,. It merely means the reader received and processed the command.

**Enable/Disable RF processing**

X108 t

t = process rf cards 0=off, 1=on

**notes**

- Rf card reading can be turned off with this command. This is useful when installing dual or tri technology readers; however you do not want to process rf cards at this time.

**Process Cards Meeting Mask Requirements only**

X109 x y z

x = barcode	0=off, 1=on
y = mag track 1	0=off, 1=on
z = mag track 2	0=off, 1=on

**notes**

- If these options are on, then cards which are read, that do not match the length requirements for one of the programmed masks, are ignored.

## Serial Port Commands

### Serial Port General Setup

X014 b p ttt yyy1

b = baudrate	1=1200
	2=2400
	3=4800
	4=9600
	5=19200
	6=38400
	7=57600
	8=115200
p = parity	1=8 none
	2=7 even
	3=7 odd
ttt= terminator 1	001-127
yyy= terminator 2	001-127, 000=none

### notes

- The terminator character can be any ascii character from 001 thru 127. The terminator is entered in using the 3-digit decimal value of the ascii character, i.e. a carriage return is 013. You must have at least one terminator character.. If you are not using a second terminator character, set yyy to 000.
- Changing the terminator character(s) affects output only. Serial input is always terminated with a 0d hex.
- You cannot change the serial parameters for readers with tcp. The serial parameters are fixed for tcp units.
- Changes made to the serial port do not take effect until the reader is power cycled, or a restart command is issued.

### Multidrop (Protocol Mode) Setup

X096 xxx y

xxx= reader address	001-126
y = protocol mode on/off	0=off, 1=on

## notes

- The command to change the reader address in the Jx series readers is different than the command used in the J series. The J series command (null IA) will not work in the Jx series readers.
- For reader firmware prior to June 2011, multidrop features are available only with multidrop firmware.
- For reader firmware after June 2011, all serial readers can operate in protocol or non-protocol mode.
- Multidrop readers are shipped from the factory with address 000.

## Multidrop Transmit Delay (version 1.5+)

X125 dd

dd= delay in milliseconds

## notes

- Applies to protocol mode only. Set dd to the amount of time (in milliseconds) that you want the reader to wait after receiving a command, before sending a response. This is required for some RS422 or RS485 converters which are not able to process incoming data immediately after transmitting data. This applies to both 2-wire and 4-wire mode.



## External RS232 Input Commands

Jx readers can be special ordered with an external rs232 input. Data received on the rs232 input line is transmitted to the host undisturbed without any processing, other than the addition of prefix and/or suffix characters. The baud rate of external rs232 inputs is fixed at 9600,N,8,1. Not all firm-ware versions will support this feature. Readers must be ordered with special firmware to support this feature.

### External Input Prefix

X100 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii chracter, 000 denotes end
bbb=	prefix character 2	3 digit ascii chracter, 000 denotes end
ccc=	prefix character 3	3 digit ascii chracter, 000 denotes end
ddd=	prefix character 4	3 digit ascii chracter, 000 denotes end
eee=	prefix character 5	3 digit ascii chracter, 000 denotes end

#### notes

- Up to 5 prefix characters can be programmed.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the track 3 data with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

### External Input Suffix

X101 aaa bbb ccc ddd eee

aaa=	suffix character 1	3 digit ascii chracter, 000 denotes end
bbb=	suffix character 2	3 digit ascii chracter, 000 denotes end
ccc=	suffix character 3	3 digit ascii chracter, 000 denotes end
ddd=	suffix character 4	3 digit ascii chracter, 000 denotes end
eee=	suffix character 5	3 digit ascii chracter, 000 denotes end

#### notes

- Same as above
- The largest decimal value (ascii char) that can be used as a suffix is 126.

## Mifare Programming

Jx readers with the Mifare sector option can read either the csn from Mifare Classic cards or up to three blocks (48 bytes) of sector information. Both 32 bit and 64 bit csn is supported. Only 1K cards are supported. 4K Mifare cards are not currently supported, but will be in a future firmware upgrade.

Data which is read from Mifare cards is transmitted using hexadecimal characters. CSN's are in normal order, not reversed.

To read the csn, select the csn option in the X097 command. The csn will be returned as 8 hex characters or 16 hex characters depending on whether the csn is 32 bits or 64 bits.

To read block data from a card, you can specify up to 3 sector/block addresses to be read using the X098 command. For each sector/block address 16 bytes of data will be returned in hexadecimal form (16 bytes = 32 hex characters). You specify the sector number (0 thru 15), block number within the sector (0 thru 3) and finally the key (A or B) which is used to read the block. The data which is read is concatenated into one string before transmission. Data can be read from multiple sectors.

Keys are programmed using the X099 command. You specify the sector number (0 thru 15) and the key style (A or B) followed by 12 hexadecimal characters representing the 6 key bytes.

As of June 2011, only Mifare reading is supported. Mifare sector writing will be supported in later firmware versions.

### Mifare Read Setup

X097 x y

x = read type	0=no mifare read, 1=read csn, 2=read blocks
y = good read beep	0=disabled, 1=enabled

### notes

- You cannot read both the csn and block data. set the x parameter to specify csn or block reading.

## Mifare Block Read Setup

X098 ss bb k ss bb k ss bb k

ss = sector number

specify sector number 0 thru 15

use 16 to mark the end of the list

bb = block number

specify block number (0-3) within the sector

k = key number

specify A or B

### notes

- You can read up to 3 blocks of data. ss,bb, and k are repeated 3 times in the command for up to 3 blocks to be read. Specify a sector number of 16 to end data reading.
- Data which is read from the card is converted to hexadecimal characters prior to transmission.
- Data which is read from the card is concatenated into one string. If you program the reader to read 2 blocks (32 bytes) then you will receive 64 hexadecimal characters, the first 32 representing the first programmed block, and the second representing the second programmed block read.

### examples

Sector 1 Block 0 contains 000102030405060708090A0B0C0D0E0F hexadecimal

Sector 2 Block 0 contains A0A1A2A3A4A5A6A7A8A9AAABACADAEAF hexadecimal

The X098 command to read these two blocks, using key "A" is

X0980100A0200A16000

The data returned from the reader will be

"000102030405060708090A0B0C0D0E0FA0A1A2A3A4A5A6A7A8A9AAABACADAEAF"

## Mifare Key Setup

X099 ss k mmmmmmmmmmm

ss = sector number	specify sector number 0 thru 15
k = key number	specify A or B
m = key	12 hex chars (6 bytes) key data

### notes

- The Mifare keys are 6 bytes long. You specify the key data in hexadecimal.
- The proper Mifare key for any specific sector must be programmed in order to read data from the sector, even if the sector is using the default key.

example

Sector 1 is using Key A to read data, Key A = 010203040506 hexadecimal

To program this key:

X09901A010203040506

## Mifare Power Level

X106 ppp

ppp= power level	0-100
------------------	-------

notes

- The mifare power level does not always have to be at 100% to read a mifare card at reasonable distances. Turning the power level down reduces reader interference. The default value is 50% (050).

## PIV-II Programming

JX readers are pre-programmed at the factory to read either the 75 bit format, the 200 bit fasc-n data, or the 200 bit fasc-n data plus expiration date. For readers ordered with the 200 bit fasc-n plus expiration date, any combination of fields from the card can be read and output in any format you wish.

For readers pre-programmed with the 75 bit or 200 bit processing, the only available option is to passthru the data undisturbed.

Readers with firmware prior to 05/23/11 can process PIV-II data using only the 75 bit or 200 bit formats. Firmware after 05/23/11 can use any of the options below, including concatenating selected fields, if the reader was pre-programmed at the factory to read the complete fasc-n and expiration date.

### PIV-II Output Settings (firmware after 05/23/11 only)

X110 x 1234

x= output mode

0=passthru 75 bit  
or fasc-n undisturbed (200 bit mode)  
1=40 4-bit hex chars  
2=48 4-bit chars  
3=standard 75 bit output  
4=programmable using mask  
4 hex chars, see below

1234=mask

#### Outputs:

Passthru undisturbed:

Passes the 200 bit fasc-n as encoded on the card, or passes the 75 bit GSA output depending on the factory setting.

40 4-bit chars:

Transposes the fasc-n data to 160 bits total, each 4 bits representing a hex character, each hex character representing a character from the fasc-n.

48 4-bit chars:

Same as above, but appends the expiration date to make 192 bits total.

Standard 75 bit output:

Wiegand output in the standard 75 bit format.

Programmable mask:

Allows you to select which fields will be included in the output. The fields are concatenated together into one bit stream based on the selection in the 1234 mask. Note that the fields are concatenated in the order shown below.

## Mask:

The mask contains 4 hex characters (0-F). Each hex character represents 4 bits, which are 0 if the field is not included, or a 1 if the field is included.

Mask 1 '	Bit 3	agency code	14 bits
	Bit 2	system code	14 bits
	Bit 1	credential number	20 bits
	Bit 0	credential series	04 bits
Mask 2 '	Bit 3	individual credential issue	04 bits
	Bit 2	person identifier	34 bits
	Bit 1	organizational category	04 bits
	Bit 0	organizational identifier	14 bits
Mask 3 '	Bit 3	person/organization association category	04 bits
	Bit 2	expiration date	25 bits
	Bit 1	rfu	
	Bit 0	rfu	
Mask 4 '	Bit 3	rfu	
	Bit 2	rfu	
	Bit 1	rfu	
	Bit 0	rfu	

Example: Mask is 1440  
Output is Agency Code + Person Identifier + Expiration Date

## Serial Control Commands

The following commands are run-time commands which can be sent to the Jx reader. Most of these commands are the same as in the J series readers.

- !xx Turn Relay ON for xx seconds, maximum 15
- \* Turn Relay OFF immediately
- &xx Turn Beeper ON for xx\*10 milliseconds, maximum 490ms
- }xx Turn Red led (led 1) ON for xx seconds then turn off  
xx=00=leave on permanently
- {xx Turn Red led (led 1) OFF for xx seconds, then turn on  
xx=00=leave off permanently
- ]xx Turn Green led (led 2) ON for xx seconds, then turn off  
xx=00=leave on permanently
- [xx Turn Green led (led 2) OFF for xx seconds, then turn on  
xx=00=leave off permanently
- )xx Blink Red led (led 1) for xx seconds, then leave on
- (xx Blink Red led (led 1) for xx seconds, then leave off
- >xx Blink Green led (led 2) for xx seconds, then leave on
- <xx Blink Green led (led 2) for xx seconds, then leave off
- ? Poll command, for multidrop mode
- V Return version identification string
- Dxxxxxxx Display message xxxxxxxx on alphanumeric display
- % Return status of Sense Input 1 (returns ON or OFF)
- \$ Return status of Sense Input 2 (returns ON or OFF)
- Fx Rotate display, 0=normal(horizontal), 1=rotated(vertical)