

## Codeology i500/i100 Serial/Ethernet Interface

Issue Date 2 300610

Using the optional serial/Ethernet board, it is possible to control the ink jet printer either via a serial cable or via a standard Ethernet network. The commands are the same using either route into the ink jet printer. To help in program development, a PC program (iText) is available from Codeology which controls the ink jet via either serial or Ethernet connection.

The commands and string structures are listed in this document to help you develop the software to control the ink jet printer remotely.

### Terminology

Reference	Description
ACK	Character 06 sent by the ink jet to signify reception of a command
Command	Single character that defines the type of command string sent to the ink jet (variable character)
CR	Single byte that terminates each string to and from the ink jet (character 13)
Data	1 to n bytes of data between Command and CR sent to the ink jet, or prior to CR sent from the ink jet
Head	Each line of print will print from one print head, so each ink jet can have up to 6 heads. Sometimes these heads are combined into pairs and called Twinline heads, so each ink jet can run 6 single heads or 3 Twinline heads
LF	Character 10 sent within messages to signify an end of line during transmission and that following data is for the next print line
Line	Each message can have up to 6 lines of print
Message	A printing message sent to, from or within the ink jet. This consists of ascii characters that are printed by the print heads when the ink jet is triggered. There are up to 101 messages in the ink jet, from message 0 to message 100.
NAK	Character 21 sent by ink jet to signify failure of correctly formatted data
STX	Single byte that starts each string sent to the ink jet (character 02)

## General rules

### Serial commands

Serial commands to the ink jet all follow the same sequence.

#### Commands to the ink jet

Bytes received by ink jet (in order)

Bytes number	Name	Description
1	STX	Single byte, character 02
2	count	single byte, value (n data+2). Byte count of n data characters plus count byte plus CR terminator byte. Do <u>not</u> include STX as part of the count
3.. n+3	Data	String n characters with first character as command, the rest as data. Some commands are immediate and have no associated data
n+4	CR	CR terminator

#### Data sent from the ink jet

The ink jet will never send data unless it has been requested first. The first character will therefore always be an ACK to signify the requesting string was correctly received by the ink jet. There may be a delay between transmission of the ACK and the rest of the data.

Bytes transmitted by ink jet (in order)

Bytes number	Name	Description
1	ACK	Signifies received command string was correct
2.. n+2	Data	String n characters of data
n+3	CR	CR terminator

#### Notes

There is a timeout (approximately 0.5 seconds) in which to send a complete string to the ink jet. If a complete string is not received, reception is terminated and the ink jet sends a NAK (character 21). If the host continues sending after the timeout, the ink jet will be expecting to see an STX as the first character and will respond with a NAK for each received character until an STX and correct string is received.

If the first character is not an STX, the inkjet responds with a NAK and starts the receive process again.

The first character of the data string is the command letter, an ASCII character e.g. 'D'. Generally an uppercase character has data to go to the ink jet from the host, a lower case character means the ink jet will be expecting to send data back to the host.

When data is sent from the ink jet it is delimited with a CR.

Avoid sending data when the ink jet is printing. It will ignore data that starts during a print cycle. It is possible to corrupt the ink jet by sending parameters or message data while it is printing.

There may be pauses in transmission if the ink jet is printing, it will give priority to printing tasks which may interrupt the transmissions.

### **Printing Messages, End-of-line markers and End-of-download-line markers**

Each ink jet can contain up to 101 messages (there may be more but the upload and download limit should be left at 101).

Each printing messages consist of 6 lines of data, which are printed by print heads 1 to 6. Each line consists of 40 characters of ASCII alphanumeric characters.

Each message also has its own printing parameters. These are Speed, Dotsize, Forward Delay and Reverse Delay. These are always sent as part of an upload or download. It is possible just to alter the printing parameters of a message but leave the text alone. In this case the CR terminator is sent immediately after the parameters and no message text is sent.

#### End-of-line markers

The ink jet knows that it can print up to 40 characters per line. It will stop print before the 40<sup>th</sup> character if it finds a null anywhere in the line. The null is the ink jet internal end of line marker.

#### End-of-download-line markers

When downloading messages into the printer, the printer will accept up to 40 characters per line. It will also accept shorter lines. When it detects a LF it accepts that as an end-of-download line marker, data afterwards will be treated as for the next line.

#### How to use the two markers

To send data to the ink jet, a LF must delimit each line of data. If there are not 6 LF's in a message, the ink jet will respond with a NAK and you may corrupt the data in the ink jet memory.

If you wish to 'skip' a line and leave the existing data in the inkjet memory intact for a line, send the LF on its own.

If you wish to overwrite data on a line, send the new data then a null to tell the ink jet that it is the end of the printing data, then a LF to tell the ink jet it is the end of the receiving string as well.

If you wish to erase a line, send a null followed by a LF.

The nulls are important because downloaded data is written directly over the top of existing data in memory. If a new message is shorter than a previous one, the ink jet will not find a null from your new short message and will carry on printing the characters that are already in the memory until it either finds a null or reaches the 40<sup>th</sup> character.

In short the rules for **downloading** are

If sending less than 40 characters per line terminate each line with null and LF

If sending exactly 40 characters per line terminate each line with LF

Always send 6 lines of data (each string must contain 6 LF's)

If you do not want to alter a line of an existing message, send just the LF.

To erase a line, send null and LF

When **uploading** from the ink jet, the message is sent as a memory dump. This means you get the printing parameters and then  $40 \times 6 = 240$  characters of data with a final CR terminator. Before presenting the data to an end-user, you should search each line for the first null and stop displaying data at that point otherwise you may confuse the user.

Note if you are writing software for distribution, you should bear in mind that under certain circumstances you may find customers with special software that has more or less than 40 characters per line. In order to determine the actual number you should upload a message from the ink jet, count the printing characters received and divide by 6. This will give you the number of characters per line.

## Upper Case Command List

Command Letter	Action	Data string	Ink Jet response	Notes
A	Set time and date	A plus bcd bytes	ACK	Bytes are ordered Minutes, hours, day of week, date, month (00-12), year (00-99)
B	Set shift codes	ACK then data string of bcd bytes	ACK	Bytes are ordered hours, minutes, ASCII shift code character. Repeated 4 times for 4 all 4 shifts.
C	Clear boxcount	Just C	ACK	
D	Set repeat interval	D followed by three ASCII characters '000' to '999'	ACK	
G	Global parameters	G followed by dotsize, speed, forward delay and reverse delay. Each parameter is one byte 0-255	ACK	This sets the dotsize, speed and delays to ALL messages. <b>NOTE If you set the dotsize too low, the valves will not open and you will see no print.</b>
I	Enter purge mode <b>Warning: This turns off print and enables purging. Use with caution, this ejects ink when the photocell is covered with no warning to operators.</b>	I plus 6 bytes 00 or 255. 00 means no purge, 255 means purge. So I 255 255 0 0 0 0 will purge lines 1 and 2, but not 3,4,5 and 6.	ACK	ANY character received after this command turns purge mode off. Suggest sending a NAK, the inkjet will respond with a NAK. Any response from the ink jet means purge has been switched off.
K	Keyboard timer	K plus one byte 35-255.	ACK	Numbers smaller than 35 are ignored
L	Set language	L plus one byte, 00 for English, 01 for Spanish	ACK	
M	Set message	See separate section for detailed explanation	ACK	
O	Set options	O plus 1 byte 00-31 Set bit to 1 for: Bit 0 shaft encoder Bit 1 0 or O Bit 2 password on Bit 3 repeat print Bit 4 opto select Bits 5-7 don't care	ACK	<b>Note enabling the shaft encoder with no encoder fitted stops the print.</b>
P	Set picture or logo	P followed by one byte for logo number (0-9), one byte for logo length then two sets of data, top head first then second head	ACK	Contact Codeology for help with logo formats
R	Set parameters	R plus 1 byte 00-31 Set bit to 1 for: Bit 0 don't care Bit 1 direction (0=forward) Bit 2 orientation (0=normal) Bit 3 aspect1 (0,1) Bit 4 aspect2 (0,1) Bit 5-7 don't care	ACK	Both aspects normally set to zero. Can be 00,01,10,11 for aspect 0,1,2,or 3
S	Select message	S plus 1 byte 0-100 plus 3 ascii characters 000 to 100 (msb first)	ACK	Returns with NAK if message number too big
W	Wipe memory	W plus password of '322244'	ACK	<b>WARNING. Wipes entire printer memory of all data and resets to factory default setting.</b>

B,E,F,H,J,N,T,U,V X, Y, Z	No action		NAK	
------------------------------	-----------	--	-----	--

### Command M, set message

Data string format is

Character	Description
Message number	Message number one byte, 0-100. If too big then a NAK is returned and the rest of the data is ignored.
Dotsize	1 byte 0-255 (note less than 150 may not be enough to turn the valves on)
speed	1 byte 0-255 (usual range 15[very fast] to 55[very slow])
forward delay	1 byte 0-255
reverse delay	1 byte 0-255
<b>EITHER</b>	
CR	No data is to follow, you are only altering the Dotsize, Speed and Delays for message M
<b>OR</b>	
Data for line 1	1 to 40 characters followed by LF. Send null then LF for a blank line. ALWAYS make the last two characters a null followed by LF unless you are sending exactly 40 characters, when the null can be discarded
Data for line 2	ditto
Data for line 3	ditto
Data for line 4	ditto
Data for line 5	ditto
Data for line 6	ditto
CR	Terminator

### Example 1 Only change dotsize, speed and delay

Ink jet has in message 01

Dotsize 150  
Speed 55  
Forward Delay 25  
Reverse Delay 35

Line 1 BATCH 1234  
Line 2 SPECIAL OFFER  
Line 3 10 CENTS  
Line 4 (blank)  
Line 5 (blank)  
Line 6 (blank)

You wish to change the dotsize but not the speed or delays. If you send down:

STX 08 M 01 165 55 25 35 CR

You will not affect the printing message, but the parameters for message 1 will now be

Dotsize 165  
Speed 55  
Forward Delay 25  
Reverse Delay 35

### **Example 2 Change dotsize, speed and delay and part of the message**

Ink jet has in message 01

Dotsize 150  
Speed 55  
Forward Delay 25  
Reverse Delay 35

Line 1 BATCH 1234  
Line 2 SPECIAL OFFER  
Line 3 10 CENTS  
Line 4 (blank)  
Line 5 (blank)  
Line 6 (blank)

You wish to change the dotsize, speed, delays and the batch number To 9876. If you send down:

```
STX 25 M 01 165 65 45 75 BATCH 9876 00 LF LF LF LF LF LF CR
```

Note

1. the count byte has increased
2. you cannot send down just the batch number, you must send all the characters before it as well

This will affect only line 1 of message 1 and the parameters which will now be

Dotsize 165  
Speed 65  
Forward Delay 45  
Reverse Delay 75

Line 1 BATCH 9876  
Line 2 SPECIAL OFFER  
Line 3 10 CENTS  
Line 4 (blank)  
Line 5 (blank)  
Line 6 (blank)

**Example 3 Change dotsize, speed and delay and the entire message  
(note there are two deliberate errors shown in the message here)**

Ink jet has in message 01

Dotsize 150  
Speed 55  
Forward Delay 25  
Reverse Delay 35

Line 1 BATCH 1234  
Line 2 SPECIAL OFFER  
Line 3 10 CENTS  
Line 4 (blank)  
Line 5 (blank)  
Line 6 (blank)

You wish to change the dotsize, speed, delays and the message to BATCH 9876, NEW PRICE on line 2 and nothing on line 3. If you send down:

```
STX 34 M 01 165 65 45 75 'BATCH 9876' 00 LF 'NEW PRICE' LF LF LF LF LF CR
```

You will now affect line one and two of the printing message and the parameters for message 1 which will now be

Dotsize 165  
Speed 65  
Forward Delay 45  
Reverse Delay 75

Line 1 BATCH 9876  
Line 2 NEW PRICEFFER  
Line 3 5 CENTS  
Line 4 (blank)  
Line 5 (blank)  
Line 6 (blank)

You can see that line 2 has  
NEW PRICEFFER

When what you wanted was  
NEW PRICE

And line 3 still has 5 CENTS when you wanted it blank.

This has occurred because the null 00 character is missing from after the text for line 2, and you need another null character as the first character of line 3 to tell the ink jet to blank that line.

Change the data to

STX 36 M 01 165 65 45 75 'BATCH 9876' 00 LF 'NEW PRICE' 00 LF 00 LF LF LF LF CR

And now you will get

Line 1 BATCH 9876

Line 2 NEW PRICE

Line 3 (blank)

Line 4 (blank)

Line 5 (blank)

Line 6 (blank)

**Lower Case Command List** (When data is sent from the ink jet it is terminated with a CR)

Command Letter	Action	Data string	Ink Jet response	Notes
a	Get time and date	a	ACK then data string of bcd bytes	Bytes are ordered Minutes, hours, day of week, date, month (00-12), year (00-99)
b	Get shift codes	b	ACK then data string of bcd bytes	Bytes are ordered hours, minutes, ASCII shift code character. Repeated 4 times for all 4 shifts.
c	Get boxcount	c	ACK then 17 ASCII characters	8 ASCII digits for box count, comma, 8 ASCII digits for hidden boxcount. All leading zeros shown
d	Get repeat interval	d	ACK then 3 ASCII characters '000' to '999'	
i	End purge mode	i	ACK	Although ANY character received turns purge mode off, this is the controlled way of stopping it.
k	Keyboard timer	k.	ACK then 1 byte value 35-255	keyboard timer value
l	Get language	l	ACK then 1 byte value 00-255	0=English 1=Spanish
m	Get message	See separate section for detailed explanation		
o	Get options		ACK then 1 byte value 00-255	Bit 0 shaft encoder (1=on) Bit 1 0 or O (1=O) Bit 2 password (1=on) Bit 3 repeat print (1=on) Bit 4 opto select (1=on) Bits 5-7 undefined
p	Get picture (reserved for future use)			
r	Get parameters	r plus	ACK then 1 byte value 00-255	Bit 0 undefined Bit 1 direction (0=forward) Bit 2 orientation (0=normal) Bit 3 aspect1 (0,1) Bit 4 aspect2 (0,1) Bit 5-7 undefined
s	Send message number selected for print	s	ACK then 3 ASCII characters '000' to '999'	
v	Get version	v	ACK then ASCII characters representing ink jet software version	Gets software version, variable length, ends with CR.
x	Read opto inputs	x	ACK then 1 byte, top bit undefined	
b,e,f,g h,j,n,q,t u,,w,y,z	No action		NAK	

## Command m, get message

Data string format is

Character	Description
Message number	Message number one byte, 0-100.
Dotsize	1 byte 0-255
speed	1 byte 0-255
forward delay	1 byte 0-255
reverse delay	1 byte 0-255
No of heads	1 byte 0-n, usually 6
Characters per line	1 byte 0-n, usually 40
Data for line 1	40 characters, no terminator
Data for line 2	ditto
Data for line 3	ditto
Data for line 4	ditto
Data for line 5	ditto
Data for line 6	ditto
CR	Terminator

Note that the host software should strip out any characters after the nulls as they could confuse the user.

Referring to the Command M example 3 where the message sent down was:

Line 1 BATCH 9876  
Line 2 NEW PRICE  
Line 3 (blank)  
Line 4 (blank)  
Line 5 (blank)  
Line 6 (blank)

What will be returned as data for lines 1-6 will be:

BATCH 9876 00 (then 30 random characters)  
NEW PRICE 00 FER 00 (then 27 random characters)  
00 CENTS 00 (then 33 random characters)  
00 (then 39 random characters)  
00 (then 39 random characters)  
00 (then 39 random characters)

## I100 and i500 Serial Interface Hardware

Both the i100 and the i500 use RXD, TXD, CTS and RTS interfacing. Settings are 9600 baud, no parity, 8 data bits, 1 stop bit.

The connector is a 9-pin d-type plug. The interface was designed to plug into a pc with a direct pin-to-pin male to male cable.

Other hand shake lines have pull-up resistors available to complete PC handshake lines if required. These lines and the jumper setting to enable/disable pull ups are shown below.

The pin-outs and jumper settings are:

Pin	Name	Jumper	1-2	open
1	(output) DCD to PC.	2	10k pull up to +12	n.c.
2	(output) TXD to PC			
3	(input) RXD from PC			
4	n.c.			
5	0v			
6	(output) DSR to PC.	3	10k pull up to +12	n.c.
7	(input) CTS from PC			
8	(output) RTS to PC			
9	(output) RI to PC	1	direct to +12	n.c.

By default jumper 1 is set to provide +12v power supply for hand controller, all other jumpers are open.

RTS drops when the i100 is printing and should be monitored by any transmitting device.