the team to trust ▪ ▪ ▪

Network interface manual ▪ ▪ ▪

9410
9450

9450

Contents

Contents

Contents

Protocol - Precisions/Example 91

Parallel link 107

Contents

General

General

This manual concerns the serial and parallel links on the 9410 and 9450 printers. See the user manual for details of ordinary use and programming.
A 9410 or 9450 printer is capable of managing various protocols for various types of external links (depending on its configuration).

■ Serial links - Introduction

The printer features an asynchronous serial interface enabling connection in V24/RS232C mode or RS422 "voltage level" mode.
Serial links are connected to the industrial interface board.

REMINDER: *Transmission on RS422 links offers very good immunity to electrical and electromagnetic interference. It is therefore particularly recommended for links over long distances or in environments with significant interference.*

The RS232C/V24 or RS422 standard describes the signals available during dialog between a DTE (Data Terminal Equipment) and a DCE (Data Communication Equipment).

NOTE: *The printer should be considered as a DTE.*

■ Compatibility

This document relates only to the native 9410-9450 protocol.
The printer supports the 9030 protocol.
94x0-compatible 9030 commands are given in the "List of identifiers" section. See the specifications for the 9030 protocol for a detailed description of compatible commands.

A 9030 twin jet message can be converted in the following cases:
- if the data is on jet 1, the content is empty on jet 2.
- if the data is on jet 2, the content is empty on jet 1. In this case the content on jet 2 is transferred to the equivalent positions on jet 1.
- if data is present on jets 1 and 2, only the data on jet 1 is processed.

■ Liste des algorithmes standards

Se reporter au Manuel utilisateur et au chapitre "Performances d'impression" pour connaître les détails des algorithmes disponibles.

General

■ Connection

The serial link is connected to the industrial interface board (**1**) as follows:
– Shut down the printer by pressing [icon].
– Disconnect the printer from the mains.
– Remove the 9 screws (**3**) then withdraw the rear panel (**2**). The Industrial interface board (**1**) is on the rear of the removable panel.

– Pass the shielded connection cable through one of the cable clamps (**4**) on the removable panel.
– Connect the wires to the corresponding terminals on connector J5 (**5**) depending on the type of link used.

IMPORTANT: The shield of the connection cable used must be connected to the edge of the metal cable clamp on the printer. The same type of connection must be used on the computer.

View of the industrial interface board

J5

■ Wiring diagrams

The cable is wired between terminal block J5 on the industrial interface board and the terminals on the computer or PC.

TXD-1: V24/RS232C data (series 1)

TXD-2: V24/RS232C data (series 2)

TXD-2+ and TXD-2-: RS422 data (series 2)

(Equivalent to TD: Transmit Data)

Data transmit line

RXD-1: V24/RS232C data (series 1)

RXD-2: V24/RS232C data (series 2)

RXD-2+ and RXD-2-: RS422 data (series 2)

(Equivalent to RD: Receive Data)

Data receive line

GND: Electrical ground (V24/RS232C only)

General

☐ **Wiring diagrams for voltage transmission**

■ RS422 link

R+	TXD-2+	P017
		1
R-	TXD-2-	2
T+	RXD-2+	P0123
		1
T-	RXD-2-	2

■ V24/RS232C link

Dsub 9 (Dsub 25)   Dsub 9

3 (2)	**TXD**	**TXD-1 / 2**	3
2 (3)	**RXD**	**RXD-1 / 2**	2
5 (7)	**GND**	**GND**	5
7 (4)	**RTS**		
8 (5)	**CTS**		
6 (6)	**DSR**		
1 (8)	**DCD**		
4 (20)	**DTR**		
(1)			

■ Transmission format and speed

The transmission speed and format for the serial link are configured in the following menu:

Data is coded in hexadecimal unless otherwise indicated.

Multibyte data must be transmitted with the high order byte first. All bytes must be transmitted with the low order bit first.

Transmission speed (in baud):
9600
19200
38400
57600
115200

Parity choice (1 or 2 stop bit):
None
Even
Odd

■ Electrical specifications

The electrical specifications correspond to the V24/RS232C standard or the RS422 standard.

General principle

markem·imaje
A DOVER COMPANY

General principle

■ General principle of dialog

❑ Data sent from computer to printer

COMPUTER   PRINTER

ENQ (1 byte)	➡

⬅ | ACK (1 byte) |

| Ident. (1 byte) | Length (2 bytes) | Data (0 to n bytes) | Checksum (1 byte) | ➡ |

⬅ | ACK (1 byte) |

NOTE: The transmission request (ENQ – ACK) is optional.

❑ Data requested by computer from printer

COMPUTER   PRINTER

| ENQ (1 byte) | ➡ |

⬅ | ACK (1 byte) |

| Ident. (1 byte) | Length (2 bytes) | Data (0 to n bytes) | Checksum (1 byte) | ➡ |

⬅ | ACK (1 byte) | Ident. (1 byte) | Length (2 bytes) | Data (0 to n bytes) | Checksum (1 byte) |

| ACK (1 byte) | ➡ |

NOTE: The transmission request (ENQ – ACK) is optional.

General principle

❑ **Transmission protocol**

⟶	
ENQ (05h)	Transmission request If the printer's response is not ACK, the computer must repeat ENQ.
⟵	
ACK (06h)	Response: transmission request accepted or resumption of control confirmed.
⟵	
NACK (15h)	Response: transmission request rejected or resumption of control not confirmed.
⟵	
ENQ (05h)	Response: transmission request rejected as the printer has data to send to the computer. The computer must switch to receive mode. If the computer's response is not ACK, the printer sends ENQ again.

☐ Terminology

■ Identifier (1 hexadecimal byte)

Specific to each command (see tables on following pages).

■ Length (2 hexadecimal bytes)

The length is a hexadecimal value representing the number of bytes present after the two length bytes and not including the check byte (Checksum).
In general, the maximum value is 2044 bytes or 07h FCh.
For transmission of a message for printing, the maximum value is 4092 bytes or 0Fh FCh.

Note: The check byte is not checked by the printer if b7 of the first length byte is set to 1. In this case the data in the frame received is not checked.

■ Data (0 to n bytes)

Zero bytes for a general request from the computer to the printer.
n bytes representing the instructions needed to define a function.

■ Checksum (1 hexadecimal byte)

This corresponds to an exclusive OR of all preceding bytes (identifier, length and data bytes).

General principle

■ Dialog control and faults

☐ Receiving

The printer can receive commands from the computer at all times.

☐ Sending

The printer's maximum response time, not counting line transfer times, is 5 ms.

☐ Faults

■ "External communication TIME OUT" fault

During dialog initialization:
After the printer sends an ACK or ENQ, if the computer does not respond within two seconds, the printer stops the dialog, sends a NACK and reports an "External communication TIME OUT" fault.

During dialog:
– If the time between sending two bytes by the computer is greater than two seconds, the printer stops the dialog, sends a NACK and reports an "External communication TIME OUT" fault.
– At the end of transmission by the printer, if the computer does not respond within two seconds, the printer stops the dialog, sends a NACK and reports an "External communication TIME OUT" fault.

■ "V24 DATA" fault

At the end of dialog, if the printer responds NACK, the checksum is Invalid and communication must restart from the beginning with ENQ. After three failures or if the command identifier is unknown, the printer reports an "External data received incorrect" fault.

markem·imaje
A DOVER COMPANY

List of identifiers

markem·imaje
A DOVER COMPANY

List of identifiers

■ Commands 9410-9450

❏ Transmissions

COMMAND SUMMARY	COMMAND IDENT.	RESPONSE IDENT.	PAGE
Sending Multitop value	2E		27
Sending an acknowledge faults	3Ch		27
Send a job selected according to its position	5Ah		28
Send a time-out value for external communication	6Ch		28
Send a job to the library	9Bh	C5h	29
Send a 9410-9450 job to the library/Custom rank	9Dh	C5h	31
Sending advanced settings information	26h		33
Stop/Start the printer	30h		33
Maintenance	31h		34
Print acknowledgement request from computer	41h	E7h	34
Sending data printed by the computer "Mark & Read" (9450 option)	41h		35
Negative Print acknowledgement request from computer	41h	E1h	36
Send a job and a Custom font file to directory store	50h	50h	36
Update the printing data in the print unit	50h	50h	38
Send a job for printing	94h		38
Initialize counters	97h		39
Select a job by number	98h		40
Select a job by his name	A0h	A0h	41
Stop/Start the jet or stop the printer	C6h		42
Delete a job	C7h	C5h	43
Send an autodating table	DFh		44
Send external variables	E8h		46
Non-double printing	E9h		47
Sending a complete job for printing	EEh		47
Send a 9410-9450 job for printing	EFh		48
Promotionnal coding	6Dh		48

List of identifiers

☐   Requests

COMMAND SUMMARY	COMMAND IDENT.	RESPONSE IDENT.	PAGE
Request printer status	23h	23h	57
Request warnings/faults (ink circuit, head, printing)	24h	24h	59
Request advanced settings information	27h	27h	61
Request jet status	32h	32h	62
Request printing status	32h	32h	63
Request for information of message in library	47h	47h	64
Request information concerning AUTO/ CUSTOM mode library	47h	47h	65
Request of the value of the total print counter	56h	56h	67
Request for a job by number in the store	93h	C3h	68
Request current value of counters	95h	96h	69
Request for types of Character generators available in the printer	A4h	A5h	70
Request for current value of the autodating	A9h	ABh	72
Request software versions	B1h	B2h	75
Request for autodating parameters	BBh	BCh	76
Request Shift Code 4 parameters (custom Shift code)	BBh	BCh	80
Request for current printing counter parameters	BEh	BEh	81
Request cartridge Tag general information	BFh	BFh	82
Request for general parameters by default	CBh	CDh	83
Request for history of the thirty faults	CFh	D0h	84
Request warnings/faults (number and codes)	DAh	D2h	85
Request active job number	DBh	91h	86
Request for au auto dacting table	DEh	DFh	87
Request for the languages of the month to be printed	E0h	C4h	89

■ Commands 9030 compatible with 9410-9450.

❑ Transmissions

COMMAND SUMMARY	COMMAND IDENT.	RESPONSE IDENT.	PAGE
Sending a job to the library	9Bh	C5h	29
Sending of external variables	E8h		46
Initialization of the autodating	C8h		
Sending of the print acknowledgement by the printer	CEh		
Sending request of the print acknowledgement by the computer	D8h		
Sending an autodating table	DFh		44
Sending a job for printing	E3h		
Acknowledgement of faults	E6h		
Sending the autodating parameters	ECh		

❑ Requests

COMMAND SUMMARY	COMMAND IDENT.	RESPONSE IDENT.	PAGE
Request for a free job number in the Store	A2h	A3h	
Request for types of character generators available in the printer	A4h	A5h	70
Request printer status	A6h	E4h	
Request for the current value of the autodating elements	A9h	ABh	72
Request for functions	ADh	B0h	
Request for software	B1h	B2h	75
Request for autodating parameters	BBh	BCh	76
Request for general parameters by default	CBh	CDh	83
Request for the history of the last thirty faults	CFh	D0h	84
Request for the autodating	D6h	9Ch	
Request of warnings/faults	DAh	D2h	85
Request for an autodating table	DEh	DFh	87
Request for the languages of the month to be printed	E0h	C4h	89

Protocol - Send to the printer

Protocol - Send to the printer

This section and the following cover all commands (transmissions and requests), for dialog with the printer.
All data is stored in big endian mode (high order byte first). Data is defined in hexadecimal by default, unless stated otherwise.

IMPORTANT: For clarity, transmission protocol elements (ENQ, ACK and the checksum byte) are not given. See the "General principle of dialog" section for more details.

◻ Sending Multitop value

COMPUTER   PRINTER

                                                        Request ➡

Identifier (1 byte)	2E
Length (2 bytes)	00h, 01h
Data: - Multitop value : number of repetition of the printer message (01h to FFh) (1 byte)	

NOTE :   The triggering mode "OBJECT" is automatically selected. This value affects the print engine and not the parameters of the corresponding message in the message store.
A value of 1 causes one print.
The interval between repetitions is defined by the "Repetitive Interval", depending on the selected unit (millimeter or Htrame).
If multitop function is disabled when printing depends on the trigger mode (Unique, Repetitive).

◻ Sending an acknowledge faults

COMPUTER   PRINTER

                                                        Request ➡

Identifier (1 byte)	3Ch
Length (2 bytes)	00h, 00h

Protocol - Send to the printer

❑ Send a job selected according to its position

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	5Ah
Length (2 bytes)	00h, 03h
Data: - Reserved - Position of the library (2 bytes)	00h 00h, 01h to 00h, FF (1-255) (message present in the library)

NOTE: The job must be present in the library beforehand.
The library can contain up to 255 messages with an ID between 1 and 999.
This command can only be used as in CUSTOM rank library mode.

❑ Send a time-out value for external communication

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	6Ch
Length (2 bytes)	00h, 01h
Data: - Value in seconds: 2 to 255 (02h to FFh) (1 byte)	

NOTE: Changing the timeout can be made at any time, and remains effective as
long as the printer is powered.
The printer returns a character NACK if the acknowledgment of the data's
response by the controller is not received during this delay.

☐ Send a job to the library

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)		9Bh
	Length (2 bytes)	xxh, xxh
	Data (max. 4 KB) : - Job parameters and contents in 9410-9450 format as follows: header, parameters*, content lines and end tag. - Type of writing (Optional) (1 byte)**	binary

*NOTE: The printer should contain fonts and algorithms used by this data message.
This command accepts the data message does not contain print parameters
(Type = 01h), they will be replaced by the settings preference.*

** IMPORTANT: If this byte is not set the printer chosen this value based of
existence or not of the file in the Store.

NOTE: For more details, see the job structure section.

⬅ Response

C5h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Report (1 byte)

Protocol - Send to the printer

■ Detail of data bytes

Request (type of writing) - Optional

00h   Creation of a job

01h   Replacement of an existing job

Response (report)

00h   Write successful, job replaced

01h   Write successful, job created

02h   Write failed (library full)

05h   Write failed (ID already exist inside another message)

08h   Undefined error

09h   Existing job in the Store (for creation)

0Bh   Maximum capacity of the Store reached (9410 = 99 / 9450 = 999)

0Ch   Message ID reserved for production message identifier. (9410 = 99 / 9450 = 999)

0Dh   Message used by a production, Cable or Standard Version

0Eh   No Library available

0Fh   Unable to remove active message

11h   Library mode not suitable (Auto Rank mode only)

13h   Font and/or algorithms missing in the store

❏ Send a 9410-9450 job to the library/Custom rank

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)		9Dh
	Length (2 bytes)	xxh, xxh
	Data : - Mode of transfert (1 byte) 0 = <u>insert</u> message to the specified position, the following messages are shifted in the followings positions. 1 = <u>replace</u> message at the specified position. - Position (2 bytes) - Message position, from 1 to N (number of present mesages in the library) ■ Insert mode: ◦ beginning of the list: position = 1 ◦ end of list: position = number of items +1 ■ replace mode: 1 ≤ position ≤ number of items - Job parameters and contents in 9410-9450 format as follows: header, parameters*, content lines and end tag. - Job parameters and contents in 9410-9450 format as follows: header, parameters, content lines and end tag.	Binary

NOTE: Data must be aligned on a multiple of 4 bytes.

⬅ **Response**

C5h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Report (1 byte)

■ Detail of data bytes

00h   Successful writing with message replacement

01h   Writing with replacement of succesful message

02h   Message not recorded (library full)

Protocol - Send to the printer

05h	Message not recorded (ID already exist inside another message)
09h	Existing job in the Store (for creation)
0B	Maximum capacity of the Store reached (9410 = 99 / 9450 = 999)
0D	Message used by a production, Cable or Standard Version.
0E	No Library available.
0F	Unable to remove active message
10h	Position out of range for Insert mode
11h	Library mode not suitable (Custom rank mode only)
13h	Font and/or algorithms missing in the store

Note 1: *The printer should contain fonts and algorithms used by this data message. This command accepts the data message does not contain print parameters (Type = 0x01), they will be replaced by the settings preference.*

Note 2: *For Auto or Custom mode library, if the file representing the message exists in the store, the report will be returned "Replace OK" even for the two modes Insert/Replace. This report is conditioned by the state of the message store.*

Note 3: *Behaviour during the activation of a message by the parallel port:*
- If a message is updated in the library and if it has the same name and same ID as the active message, whatever the library position, then it will be updated for printing. The evolution of the message content will be taken into account because the received message is sent to the printing unit.
- If a message is updated in the library at the position used by the active message with a new name and ID, it will be not updated for printing. A new activation by the parallel port will be necessary, because only the library is sent to the printing unit.

❏ Sending advanced settings information

Request ➡

Identifier (1 byte)			26h
	Length (2 bytes)		00h, 02h
		Data: - Action to be performed (1 byte)	xxh

■ Detail of data byte

This command defines the behavior after sending a job to the printer does not define all parameters.

00h	Missing parameters are replaced by those active in printing (same printing context as the previous job)
FFh	Missing parameters are replaced with the default settings (preferences settings)

❏ Stop/Start the printer

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)			30h
	Length (2 bytes)		00h, 01h
		Data: - Action to be performed (1 byte)	xxh

■ Detail of data byte

00h	Stop the printer (start shutdown)
01h	(Booked)
FFh	Start the printer.

Protocol - Send to the printer

☐ **Maintenance**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	31h
Length (2 bytes)	00h, 01h
Data: - Action to be performed (1 byte)	xxh

■ Detail of data byte

00h	Shut down the jet
01h	Start the jet
02h	Refresh
03h	Stability check
04h	Introduce additive
05h	Unclog nozzle
08h	Abort printing
09h	Suspend printing
0Bh	Start the jet with rinsing
0Ch	Stop the jet with rinsing
0Dh	Start break-off point adjustment

☐ **Print acknowledgement request from computer**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	41h
Length (2 bytes)	00h, 01h
Data: - Type (1 byte)	01h

This transmission should be sent once before a production start, and remains effective as long as the printer is powered. The printer sends a print acknowledgement after the TRIGG filtering of the trigger cell or just after printing is manually triggered.

← Response

E7h	Identifier (1 byte)

The acknowledgment only uses one byte (no length bytes). This byte is sent after the TRIGG filtering of the trigger cell or just after printing if manually triggered.

NOTE: If a conflict occurs during dialog, the printer has priority.

☐ Sending data printed by the computer "Mark & Read" (9450 option)

COMPUTER   **PRINTER**

Request ▶

Identifier (1 byte)	41h
Length (2 bytes)	00h, 01h
Data: - Type (1 byte): field contents (TAG) printed is sent during each printing.	10h

NOTE :   This command is exclusive with other print acknowledgement.

Protocol - Send to the printer

□ **Negative Print acknowledgement request from computer**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	41h
Length (2 bytes)	00h, 01h
Data: - Type (1 byte)	04h

Negative printing acknowledgement when printing conditions are not acquired at the start of printing (Jet off, Start running, Fault, ...).

⬅ Response

E1h	Identifier (1 byte)

□ **Send a Message and a Custom font file to directory store**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	50h
Length (2 bytes)	xxh, xxh
Data: Operation Data: File Header - Total length of the file (4 bytes) - Checksum (4 bytes) - Type and Version Number (2 bytes) - Message or Generator name (20 bytes) - Message or Generator number (2 bytes) - Summary (32 bytes) Data: Data's File - Parameters number	01h : header part 02h : data part (one or more data blocks)

■ Detail of data bytes

01h   Header part

02h   Data part (one or more data blocks)

Response

50h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Report (1 byte)

■ Detail of data bytes

Response

00h   Write successful, job created or replaced

01h   Sequence not respected: 1st = header, followig = data's

02h   Type unknown

03h   Version of the message file not supported

04h   Operation different of 01h or 02h

05h   Data size sent > Total length of the file specified in the header

06h   File message (same name) al ready exists with another ID

07h   Other message file, different name, using the same ID

08h   The message involved is used in production or in library

09h   Received data size is smaller than data size in the header. Operation of sending header aborted.

10h   Algorithm, used by the message, not present in the store

11h   Font, used by the message, not present in the store

0Ah   Number of font ≥ 1000

0Bh   File font (same name) already exists with anothe ID

0Ch   Other font file, different name, users the same ID

0Dh   Font used in message production

*NOTE: Message file extension: *.mim*
*Font file extension: *.miF*

Protocol - Send to the printer

❑ **Update the printing data in printing**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	50h
Length (2 bytes)	00h, 01h
Data: -Update printing data, for font and the active message production (1 byte)	FFh

⬅ Response

50h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Report (1 byte)

■ Detail of data bytes

Response

04h   Operation different of FFh

0Eh   Time out of transfer of printing data to print unit

0Fh   Transfert operation of the printing deita is in progress

❑ **Start printing (Send a Dtop for printing)**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	94h
Length (2 bytes)	00h, 00h

☐ Initialize counters

COMPUTER   PRINTER

Request →

Identifier (1 byte)	97h
Length (2 bytes)	00h, 02h
Data: - Counter numbers (2 bytes)	bit field

■ Detail of data

b15	b14	b13	b12	b11	b10	b9	b8

not used = 0

= 1 for initialization of counter 15, otherwise = 0

= 1 for initialization of counter 14, otherwise = 0

= 1 for initialization of counter 13, otherwise = 0

= 1 for initialization of counter 9, otherwise = 0

= 1 for initialization of counter 10, otherwise = 0

= 1 for initialization of counter 11, otherwise = 0

= 1 for initialization of counter 12, otherwise = 0

b7	b6	b5	b4	b3	b2	b1	b0

= 1 for initialization of counter 8, otherwise = 0

= 1 for initialization of counter 7, otherwise = 0

= 1 for initialization of counter 6, otherwise = 0

= 1 for initialization of counter 5, otherwise = 0

= 1 for initialization of counter 1, otherwise = 0

= 1 for initialization of counter 2, otherwise = 0

= 1 for initialization of counter 3, otherwise = 0

= 1 for initialization of counter 4, otherwise = 0

Protocol - Send to the printer

□ **Select a job by number**

This transmission selects a job library according to its position and send printing.

COMPUTER   **PRINTER**

Request →

Identifier (1 byte)	98h
Length (2 bytes)	00h, 02h
Data (2 bytes): - Job number (001 to 999)	from 00h, 01h to 03h, E7h

NOTE: *- The message must be present in the library beforehand. The library can contain up to 255 messages with an ID between 1 and 999.*
- This command can be used in AUTO or CUSTOM rank library mode. (even if a message is placed on multiple position in custom mode).

☐ Select a job by his name

This command selects a library job after his name and send printing.

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	A0h
Length (2 bytes)	01h to 14 h
Data (1 to 20 bytes): - Significant variable length string terminated or not by "10" character	ASCII

⬅ **Response**

A0h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Report (1 byte)

■ Detail of data bytes

00h	Message selection successful
01h	Message not present in the store
02h	No active production*
03h	Message not usable due to a faulty content
04h	Inconsistent production (distance A or B or C too small)*
05h	Algorithm, used by the message, not present in the store
06h	Font, used by the message, not present in the store

NOTE: *The message must be present in the message store beforehand and then be transfered ot the print unit as the current message.*
All the print data must be present: algorithms and fonts.

* Concerning the printer model 9232E

Protocol - Send to the printer

❑ Stop/Start the jet or Stop the printer

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)		C6h
	Length (2 bytes)	00h, 01h
		Data: - Jet or printer status (1 byte)

(Data value: xxh)

■ Detail of data byte

00h   Stop jet

01h   Start jet

08h   Stop printer

❏ Delete a job

Request ➡

Identifier (1 byte)	C7h
Length (2 bytes)	00h, 02h
Data (2 bytes): - Job number (001 to 999)	from 00h, 01h to 03h, E7h

⬅ **Response**

C5h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Report (1 byte)

■ Detail of data byte

06h   Job deleted

07h   Job not deleted (Job does not exist or job active)

Protocol - Send to the printer

☐ **Sending an autodating table**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	DFh
Length (2 bytes)	xxh, xxh
Data (4 ko max.) : - Type of table (1 byte) - Table (n bytes)	binary

■ Detail of data byte

Type of table

00h	hour table
01h	minute table
02h	days of the week table
03h	days of the year table
04h	days of the month table
05h	weeks table
06h	months of the year table
07h	years table
08h	shift code 2 table (compatibility 9030)
09h	Not used
11h	First day of the week
12h	Suppress zero digit before figure of day/month for Date & Date1-6.
13h	Shift code 4, custom
0Ah	shift code 3 table (compatibility 9030)
0Bh	Not used

0Dh	shift code 1, alpha –I –O
0Eh	shift code 1, numerical
0Fh	shift code 1, alpha

Table

Hour table	24 X 3 characters ASCII
Minute table	60 X 3 characters ASCII
Days of the week table	7 X 3 characters ASCII
Days of the year table	366X 3 characters ASCII
Days of the month table	31 X 3 characters ASCII
Weeks table	53 X 3 characters ASCII
Months of the year table	12 X 3 characters ASCII
Years table	10 X 3 characters ASCII
Shift code 2 table (compatibility 9030)	366 X 3 characters ASCII
Shift code 3 table (compatibility 9030)	7 x 24 x 3 characters ASCII
Shift code 1, alpha –I –O	24 X 1 characters ASCII
Shift code 1, numerical	100 X 2 characters ASCII
Shift code 1, alpha	26 X 1 v ASCII
Shift code 4, custom	7 x 24 x 3 characters ASCII*
First day of the week	00 to 06
Suppress zero digit before figure of day/month for Date & Date1-6.	0h: disable FFh: enable

*NOTE: each item in the table must be described in three digits ; the useful characters
should be left - justified with 1 or 2 digits of padding at 0x00.
Example: 41 00 00 = "A"
        41 42 00 = "AB"
7: days of the week
24: hours of the day
3: characters ASCII

Protocol - Send to the printer

☐ **Transmit external variables**

Request ➡

Identifier (1 byte)	E8h
Length (2 bytes)	xxh, xxh
External variable data	X bytes

NOTE: For more details, see the job structure section.

■ Detail of data

Variable 1 number (1 to n)	(1 byte)
Variable 1 length	(2 bytes)
Variable 1 contents	(X bytes)
…	
Variable n number (1 to n)	(1 byte)
Variable n length	(2 bytes)
Variable n contents	(X bytes)

NOTE: A job may contain up to 10 variables.

❑ Non-double printing

Request ➡

Identifier (1 byte)	E9h
Length (2 bytes)	00h, 01h
Data: - Type (1 byte)	xxh

This transmission is sent once before starting production. Non-double printing may be disabled without stopping the printer.

■ Detail of data byte

00h   Disable non-double printing

01h   Enable non-double printing

NOTE: *The role of this function is to prevent a job being printed twice in succession without its contents being modified*

❑ Sending a complete job for printing

Request ➡

Identifier (1 byte)	EEh
Length (2 bytes)	xxh, xxh
Data: - Header including the version of the job data - Number of parameters and list of parameters - Job to print (lines definition).	X octets

Protocol - Send to the printer

❑ Send a job for printing

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	EFh
Length (2 bytes)	xxh, xxh
Data: - Number of parameters and list of parameters - Job to be printed (definition of lines).	(X bytes)

The job is only sent to the printer's buffer memory.
It is not stored in the list of jobs.

NOTE: For more details, see the job structure section.

❑ Promotionnal coding

The main purpose of the queue management system implemented in 9232 is for promotional coding → printing unique codes on each product. So the idea is to send to the printer a list of data (can be complete messages or only external variables) and each data will be used once for one printout..

A new V24 command 0x6D has been added to the existing one in order to:
- Enable/Disable the data queue management
- Reset the queue
- Read data queue status
- Send data list into the data queue
- Send application options
- Get the max items number to transfer

The printer will always answer with another 0x6D command with reporting and data status. The particularity with this answer is that the printer will not wait for any acknowledgement.

When the queue has been populated with data, at each new trig the printer takes the first element of the queue, computes the message with the new value, prints it and removes this element from the queue.

■ 0x6D command in detail

– **Command format**

COMPUTER   **PRINTER**

Request →

Identifier (1 byte)	6Dh
Length (2 bytes)	xxh, xxh
Application type (1 byte)	
Sub-command (1 byte)	
Command definition (n bytes)	binary
Checksum (1 byte)	xxh

Protocol - Send to the printer

   – **Enable/Disable the data queue**

COMPUTER   **PRINTER**

Request →

Identifier (1 byte)	6Dh
Length (2 bytes)	00h, 03h
Application type (1 byte)	00h
Sub-command (1 byte)	01h
Action : 0 x 00 = Disable 0 x 01 = Enable	xxh
Checksum (1 byte)	xxh

   – **Read Status**

COMPUTER   **PRINTER**

Request →

Identifier (1 byte)	6Dh
Length (2 bytes)	00h, 02h
Application type (1 byte)	00h
Sub-command (1 byte)	03h
Checksum (1 byte)	xxh

– Reset data queue

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	6Dh
Length (2 bytes)	00h, 02h
Application type (1 byte)	00h
Sub-command (1 byte)	02h
Checksum (1 byte)	xxh

– Send data list to the queue

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	6Dh
Length (2 bytes)	xxh, xxh
Application type (1 byte)	00h
Sub-command (1 byte)	0Ah
Number of external variables (2 bytes). Number of external variables per group (1 byte) Data list of variables (N bytes) - Variable number (1 byte) - Variable length (2 bytes) - Variable content (X bytes)	binary
Checksum (1 byte)	xxh

Protocol - Send to the printer

– **Answer format**

Response

6Dh	Identifier (1 byte)
00h, 0Ah	Length (2 bytes)
xxh xxh	Command report (2 bytes)
Binary	Status (8 bytes)
xxh	Checksum (1 byte)

Command report definition

0 x 0000	Unknown Command
0 x 0001	Command processed
0 x 0002	State already asked (for enable/disable)
0 x 0003	Data processinf function nonexistant
0 x 0004	Not validated data queue
0 x 00FF	Data queue full (command not processed)

Status definition

Data queue status	1 byte
Printer status	1 byte
Remaining item in the queue	2 bytes
Remaining memory available in the queue in number of bytes	4 bytes

- Data queue status values

0 x 00	Data queue in use
0 x 01	Job in place without external variable
0 x 02	Queue empty, no data to print

- Printer status values

Bit 7	Printer not ready
Bit 6	-
Bit 5	-
Bit 4	Consumable empty
Bit 3	Consumable low
Bit 2	Warning except consumable
Bit 1	Fault on printer
Bit 0	-

markem·imaje
A DOVER COMPANY

■ Exemple

– **Diagramme de séquence**

– **Message Exchange**

⬜	Length
🟪	Application type + Sub command
🟧	Total number of external variables
🟦	Number of external variables per bloc. A bloc is one data queue element.
🟩	Variable number
⬛	Variable data length
🟪	Variable content
🟦	Command report
🟨	Status

Protocol - Send to the printer

   – **Printout**

```
000000000000H
000000000000G
 000000000000F
 000000000000E
 000000000000D
  000000000000C
  000000000000B
  000000000000A
```

■ Note

When disabling the queue management with the appropriate 6Dh command, we should send also the E9h command (E9 00 01 00) to disable the "No double printing" feature.

Protocol - Request to the printer

markem·imaje

A DOVER COMPANY

Protocol - Request to the printer

☐ Request printer status

COMPUTER   PRINTER

Request ➡

Identifier (2 bytes)	23h
Length (2 bytes)	00h, 00h

⬅ **Response**

23h	Identifier (2 bytes)
00h, 54h	Length (2 bytes)
xxh	Data: - Printer status (n bytes)

■ Detail of data

		Size in bytes	Details
Speed of motor (Rpm/min)		(2 bytes)	Rpm/min
Pressure (par pas de 10 mbar)		(2 bytes)	In step of 10 mbar
Viscosity (Seconde/10)		(2 bytes)	Second / 10
Tank Ink level (mm)		(2 bytes)	mm
Solvent added (cc)		(2 bytes)	CC
Electronic temperature (°C)		(2 bytes)	°C (value x313/10 000)
Ink temperature (°C)		(2 bytes)	°C
Head temperature (°C)		(2 bytes)	°C (In steps of 0,5 °C)
Ink circuit solenoid valve state	Bit field*	(2 bytes)	
Head solenoid valve state	Bit field*	(2 bytes)	
Ink task phase		(2 bytes)	MI use/Reserve MI
Ink task sub-phase		(2 bytes)	MI use
Tube task phase		(2 bytes)	MI use
Tube task sub-phase		(2 bytes)	MI use
Motor task phase		(2 bytes)	MI use
Jet speed		(2 bytes)	MI use
Number of printing phase		(2 bytes)	MI use
Solvent added		(2 bytes)	CC (Volume)
Viscosity target		(2 bytes)	In steps of 0,1 s

detail see page 48

Protocol - Request to the printer

	Size in bytes	Details
Offset sensor level	(2 bytes)	
Beginning measurement time	(2 bytes)	MI use
Break of time	(2 bytes)	MI use
Current tube level	(2 bytes)	mm
Piezo target	(2 bytes)	MI use
CKMax	(2 bytes)	MI use
CKMin	(2 bytes)	MI use
Recup Level	(2 bytes)	
THT target	(2 bytes)	Volt = (value/51 -1)x1500
Printing state	(2 bytes)	0xff: manual printing 0x0: normal printing
Total additive consumption	(4 bytes)	mm
Tank additive quantity	(4 bytes)	mm
Time left Ink	(2 bytes)	1/10 Hours
Viscosity management state	(2 bytes)	0 = automatic 1 = inhibited
Average ink consumption	(4 bytes)	cc/1/10h
Pressure target	(2 bytes)	
Speed motor target	(2 bytes)	
Ink autonomy display	(2 bytes)	1 = a bar is displayed
Additive cartridge	(2 bytes)	1 = the additive cartridge is not present
Additive pump pressure (Ptr2)	(2 bytes)	2 Parameter supplied during production machine, (1/10 bars)

■ Detail of Bit field

	Ink circuit solenoid valve state	Head solenoid valve state
Bit 15	Not used	Not used
Bit 14	Not used	Elv Recup
Bit 13	Not used	Elv Solvent
Bit 12	(Drop counter)	Elv Purge
Bit 11	Peltier cell	Elv Pressure
Bit 10	PTR3 (drain pump transfer)	Not used
Bit 9	PTR2 (solvent pump transfer)	Not used
Bit 8	PTR1 (ink pump transfer)	Not used
Bit 7	Elv 8	Not used
Bit 6	Elv 7	Not used
Bit 5	Elv 6	Not used

Bit 4	Elv 5 (head)	Not used
Bit 3	Elv 4 (solvent)	Not used
Bit 2	Elv (ink level)	Not used
Bit 1	Elv (return pressure)	Not used
Bit 0	Elv (ink)	Not used

◻ Request warnings/faults (ink circuit, head, printing)

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	24h
Length (2 bytes)	00h, 00h

⬅ Response

24h	Identifier (1 byte)
00h, 0Ch	Length (2 bytes)
bit field	Data: - Ink circuit fault (2 bytes) - Print head faults (2 bytes) - Printing faults (2 bytes) - Ink circuit warnings (2 bytes) - Print head warnings (2 bytes) - Printing warnings (2 bytes)

Protocol - Request to the printer

■ Detail of data

	Faults			Alarms		
	Ink circuit	Print head	Printing board	Ink circuit	Print head	Printing board
Bit 15						
Bit 14						
Bit 13	Level sensor out of specification					
Bit 12	Draining fault					
Bit 11	Ink circuit electrovalve supply failure					
Bit 10	Electronic fault					
Bit 9	Transfert pump 3 blocked	Phase detection fault (drop)				
Bit 8	Transfert pump 2 blocked	Recovery fault				
Bit 7	Transfert pump 1 blocked	Head cover missing				
Bit 6	Fan fault	ELV PSU failure		Ink temperature too high		
Bit 5	Additive tank empty	EHV PSU failure		Viscosity measurement incorrect		
Bit 4	Ink tank empty	HT PSU failure	Job not valid	Motor speed fault		
Bit 3	Viscosity too high	Piezo PSU failure	Hijri calendar	Pressure fault		Printing speed too low
Bit 2	Ink level too high	Communication error between FPGA and print head	No message to print	Ink level measurement timeout		No job available in "non-double" mode
Bit 1	Viscosity measurement timeout		Algorithm not available	Ink cartridge empty	Jet position warning	Flash memory write timeout
Bit 0	Measurement tube emptying timeout		Font not available	Additive cartridge empty	Communication error between FPGA and print head	Printing speed too high

markem·imaje
A DOVER COMPANY

☐ Request advanced settings information

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	27h
Length (2 bytes)	00h, 01h

⬅ **Response**

27h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: "Undefined Job Settings" option

■ Detail of data byte

This command behavior after sending a job to the printer does not define all parameters.

00h	Missing parameters are replaced by those active in printing (same printing context as the previous job)
FFh	Missing parameters are replaced with the default settings (preferences settings)

Protocol - Request to the printer

☐ **Request jet status**

PRINTER

Request ➡

Identifier (1 byte)	32h
Length (2 bytes)	00h, 00h

⬅ **Response**

32h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Jet status (1 byte)

■ Detail of data byte

00h	Jet stopped
01h	Jet starting
02h	Jet in refresh
03h	Jet in stability check
04h	Jet in introduce additive
05h	Jet in unclog nozzle
06h	Jet in adjustment to gutter
07h	Jet running
09h	Unblock gutter
0Bh	Jet starting in rinsing phase
0Ch	Jet stopped in rinsing phase
0Dh	Break off point adjustment

◻ Request printing status

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	32h
Length (2 bytes)	00h, 01h

⬅ **Response**

32h	Identifier (1 byte)
00h, 01h	Length (2 bytes)
xxh	Data: - Printing status

■ Detail of data byte

00h	Printing in pause
01h	Operational printing
02h	Printer not ready to print

Protocol - Request to the printer

❑ **Request for information of message in library**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	47h
Length (2 bytes)	00h, 02h

Data:
- Position in library (2 bytes)
  ● in AUTO rank mode the position is equal to the message number (ID)
  ● in CUSTOM rak mode the position corresponds to the position in the list of messages
	xxh, xxh
	00h, 01h to N
	(max 255)

⬅ **Response**

47h	Indentifier (1 byte)
xxh, xxh	Length (2 bytes)
xxh xxh	Data : - Message ID (2 bytes) - Message name (1 to 20 bytes)
00h	- Delimiter (1 octet)

■ Detail of data byte

Message ID
– if the message exist in library : Message number
– Position en dehors de la bibliothèque : = 00h

Message Name
– Only if the message exist on the library : Message name

Delimiter
– Only if the message exist on the library:
  00h : ending with the delimiter 00h

❑ Request information concerning AUTO/CUSTOM mode library

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	47h
Length (2 bytes)	00h, 03h

Data:
- Position in library (2 bytes)
  ● in AUTO rank mode the position is equal to the message number (ID)
  ● in CUSTOM rak mode the position corresponds to the position in the list of messages
- Type d'information bibliothèque
  ● general information
  ● list of message
	xxh, xxh
	00h, 01h to N
	(max 255)
	
	00h
	01h

⬅ **Response (Type 00h)**

47h	Indentifier (1 byte)
00h, 06h to 00h, 01Ah	Length (2 bytes)
00h, 01h Number of items in the list of library xxh xxh 00h	Data: - Library mode (1 byte) - Library size (2 bytes) - Message number (ID) (2 bytes) - Message name (1 to 20 bytes) - Delimiter (1 byte)

Protocol - Request to the printer

■ Detail of data byte

Library mode

00h   AUTO rank mode

01h   CUSTOM rank mode

Message ID
– If the message exists on the library: message number
– Position outside the library: 00h

Message name
– Only if the message exist on the library: message name (max. 20 characters)

Delimiter
– Only if the message exist on the library:
  00h : ending with the delimiter 00h

Response (type 01h)

47h	Indentifier (1 byte)
n bytes	Length (2 bytes)
n x n° message	Data: - List of messages (2 bytes) - Message name (1 to 20 bytes)

■ Detail of data byte

List of messages
– Classement AUTO : numéro du message (position = ID)
– Classement Personnalisé : position du message (1 ... n)

Message name
– Maximum 20 characters
– 00h: ending with the delimiter 00h

markem·imaje
A DOVER COMPANY

◻ Request of the value of the total print counter

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	56h
Length (2 bytes)	00h, 00h

⬅ **Response**

56h	Indentifier (1 byte)
00h, 04h	Length (2 bytes)
xxh	Data : - Printing counter value (4 bytes)

Protocol - Request to the printer

❑ **Request for a job by number in the store**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	93h
Length (2 bytes)	00h, 02h
Data: - job number (2 bytes)	de 00h, 01h to 03h, E7h (1 to 999)

⬅ **Response**

C3h	Identifier (1 byte)
Equal to the information in the header of the job = "Total length of the file" on a 16 bits word	Length (2 bytes)
n bytes	Data : - Job data: see chapter "Job format specification" for structure of the job.

NOTE :   *If the requested message does not exist, the command returns a Length of 0 and no Message data.*

markem·imaje
A DOVER COMPANY

❑ Request current value of counters

PRINTER

Request ➡

Identifier (1 byte)	95h
Length (2 bytes)	00h, 02h
Data: - List of counters (2 bytes)	bit field

■ Detail of data

High-order byte

b15	b14	b13	b12	b11	b10	b9	b8

batch count of current job = 0

= counter 15

= counter 14

= counter 13

= counter 9

= counter 10

= counter 11

= counter 12

Low-order byte

b7	b6	b5	b4	b3	b2	b1	b0

= counter 8

= counter 7

= counter 6

= counter 5

= counter 1

= counter 2

= counter 3

= counter 4

⬅ **Response**

96h	Identifier (1 byte)
xxh, xxh	Length (2 bytes)
ASCII (X bytes)	- For counters 1 to 15: 9 characters (0 to 9)
Binary	- Batch counter value (unsigned long size 4 bytes)

Protocol - Request to the printer

❑ **Request for types of Character generators available in the printer**

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	A4h
Length (2 bytes)	00h, 03h
Data: - Action to be performed (3 bytes)	FFh, 00h, 00h
(compatible fonts from 9040, number >= 1000/0x03E8)	FEh, 00h, 00h

⬅ **Response**

A5h	Identifier (1 byte)
xxh, xxh	Length (2 bytes)
xxh, xxh xxh, ... xxh, ...	Data: - Number of generator (2 bytes) - Description of the first generator (31 bytes) - Description of the end generator (31 bytes)

■ Detail of data byte

Type of table

Title of the generator ASCII (20 ASCII characters)

Reference number (2 bytes)

Number of drops per frame (1 byte)

Number of frames per character (2 bytes)

Number of characters (2 bytes)

Number of byte of the full frame (1 byte)

Language type: (1 bytes)

00h = logo

01h = Latin

02h = Arabic

03h = Chinese

04h = Hebrew

05h = Greek

06h = Pynin

07h = Japanese

08h = Korean

09h = Thai

0Ah = Cyrillic

0Bh = Hijri

0Ch = Latin Chimney

FEh = font is hidden on UI

FFh = reserved for extension.

Mode : fixed font = 0, proportional = 1 (1 byte)

Number of range (1 byte)

Protocol - Request to the printer

◻ **Request for current value of the autodating**

COMPUTER   **PRINTER**

Request ➡

	A9h
Identifier (1 byte)	A9h

	00h, 04h
Length (2 bytes)	00h, 04h

	Binary

Data (1 byte):
- timer type (Binary)
  00 = autodating
  01 = extended autodating
	Binary

- Current version (3 ASCII characters)	30h, 30h, 30h

⬅ **Response**

ABh	Indentifier (1 byte)
xxh, xxh	Length (2 bytes)
	Data: - According to the type required

markem·imaje
A DOVER COMPANY

■ Detail of data

Autodating

Seconds	from 00 to 59	(2 ASCII characters)
Minutes	from 00 to 59	(2 ASCII characters)
Hours	from 00 to 23	(2 ASCII characters)
Mode 12/24 H	AM/PM/SPACE	(2 ASCII characters)
Day of month	from 1 to 31	(2 ASCII characters)
Day of year	from 001 to 366	(3 ASCII characters)
Week of year	from 01 to 53	(2 ASCII characters)
Month of year	from 01 to 12	(2 ASCII characters)
Month of year in letters (language 1)	JAN to DEC	(3 ASCII characters)
Year on 2 digits	from 00 to 99	(2 ASCII characters)
Postday of month	from 01 to 31	(2 ASCII characters)
Postday of year	from 001 to 366	(3 ASCII characters)
Postweek of year	from 01 to 53	(2 ASCII characters)
Postmonth of year	from 01 to 12	(2 ASCII characters)
Postmonth of year in letters (language 1)	JAN to DEC	(3 ASCII characters)
Postyear	from 00 to 99	(2 ASCII characters)
Shift code 1 in letter	A to Z	(1 ASCII character)
Shift code 1 in number	from 01 to 99	(2 ASCII characters)
Shift code 1 in letter	A to Z -O -I	(1 ASCII character)
Day of week	from 1 to 7	(1 ASCII character)
Postday of year modulo programmable	from 1 to 999	(3 ASCII characters)
Delimiter ":"	3Ah	(1 ASCII character)
Delimiter "/"	2Fh	(1 ASCII character)
Delimiter "."	2Eh	(1 ASCII character)
Delimiter " "	20h	(1 ASCII character)
Month of year in letters (language 2)	JAN to DEC	(3 ASCII characters)
Postmonth of year in letters (language 2)	JAN to DEC	(3 ASCII characters)
Delimiter "("	28h	(1 ASCII character)
Delimiter ")"	29h	(1 ASCII character)
Delimiter "-"	2Dh	(1 ASCII character)
Century on 2 digits	"20"	(2 ASCII characters)

Extended autodating (For each postdate (2 to 6)):

□ **POSTDATE 2**

Postday of month	01 to 31	(2 ASCII characters)
Postday of year	001 to 366	(3 ASCII characters)
Postday of year modulo programmable		(3 ASCII characters)
Postweek of year	01 to 52	(2 ASCII characters)
Postmonth of year	01 to 12	(2 ASCII characters)
Postyear	00 to 99	(2 ASCII characters)
Postday of week in letters	MON to SUN	(3 ASCII characters)
Postmonth of year in letters	JAN to DEC	(3 ASCII characters)

□ **POSTDATE 3, 4, 5 and 6**

Postday of month	01 to 31	(2 ASCII characters)
Postday of year	001 to 366	(3 ASCII characters)
Postday of year modulo programmable		(3 ASCII characters)
Postweek of year	01 to 52	(2 ASCII characters)
Postmonth of year	01 to 12	(2 ASCII characters)
Postyear	00 to 99	(2 ASCII characters)
Postday of week in letters	MON to SUN	(3 ASCII characters)
Postmonth of year in letters	JAN to DEC	(3 ASCII characters)

Hours table		(3 ASCII characters)
Minutes table		(3 ASCII characters)
Month of year table		(3 ASCII characters)
Day of week table		(3 ASCII characters)
Day of month table		(3 ASCII characters)
Day of year table		(3 ASCII characters)
Week of year table		(3 ASCII characters)
Years table		(3 ASCII characters)
Shift code 2 table		(3 ASCII characters)
Shift code 3 table		(3 ASCII characters)
Delimiter ":"	3Ah	(1 ASCII character)
Delimiter "/"	2Fh	(1 ASCII character)
Delimiter "."	2Eh	(1 ASCII character)
Delimiter " "	20h	(1 ASCII character)
Delimiter "("	28h	(1 ASCII character)
Delimiter ")"	29h	(1 ASCII character)
Delimiter "-"	2Dh	(1 ASCII character)

◻ **Request software versions**

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	B1h
Length (2 bytes)	00h, 01h
Data: - Fixed value (1 byte)	00h

⬅ **Response**

B2h	Identifier (1 byte)
00h, 4Ah	Length (2 bytes)
xxh, ... 00h xxh, ... 00h xxh, ... 00h xxh, ... 00h xxh, ... 00h xxh, ... 00h xxh, ... 00h xxh, ... 00h	Data: - Software version (16 ASCII characters) - Padding with the espace character (20 h) if the string is smallet 16 - Separator (1 byte) - Coldfire Boot version (10 ASCII characters) - Separator (1 byte) - Eboot version (4 ASCII characters) - Separator (1 byte) - Head FPGA version (4 ASCII characters) - Separator (1 byte) - NEP FPGA version (8 ASCII characters) - Separator (1 byte) - OS version (4 ASCII characters) - Separator (1 byte) - UI version (12 ASCII characters) - Separator (1 byte) - NEP DLL version (12 ASCII characters) - Separator (1 byte)

Protocol - Request to the printer

◻ **Request for autodating parameters**

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	BBh
Length (2 bytes)	00h, 03h
Data: - Soft version A12 (3 ASCII characters)	30h, 30h 30h

⬅ **Response**

BCh	Identifier (1 byte)
00h, 92h	Length (2 bytes)
xxh, xxh xxh	Data (129 bytes): - postdate 1 modulo (2 bytes) - first weekday for coding the day of the week (1 byte)
xxh, ...	- Shift code 1 parameters (5 bytes)
xxh, ... 00h to 3Bh	- Shift code 2 parameters (8 bytes) - Number of days of the interval
xxh	- Language 1 for months of the Year (1 byte)
xxh, xxh, ... 00h, 00h 20h, 00h, 00h 00h, 00h 20h, 00h, 00h	- Shift code 3 parameters (112 bytes) - 5 bytes reserved* - 5 bytes reserved* * following parameters for each day : (x 7 bytes)
xxh, xxh, ...	- postdate 2 to 6 modulo (10 bytes)
xxh	- language 2 for months of the Year (1 byte)
xxh, ...	Date change time shift for coding the postdates
xxh, ...	Date change time shift for coding the date (3 bytes)

■ Detail of data

Postdate 1 modulo:
From 0000 to 9999 (2 bytes)

First day of the week for coding the week-day (1 byte):
01 Monday
02 Tuesday
03 Wednesday
04 Thursday
05 Friday
06 Saturday
07 Sunday

Shift code 1 parameters (5 bytes):

Starting time	00h to 17h	(1 byte)
Starting minute	00h to 3Bh	(1 byte)
Mode 12/24 H in ASCII	(A/P/SPACE)	(1 byte in ASCII)
Number of hours of the interval	00h to 17h	(1 byte)
Number of minutes of the interval	00h to 3Bh	(1 byte)

Shift code 2 parameters (8 bytes):

Starting day	00h to 1Fh	(1 byte)
Starting month	00h to 0Ch	(1 byte)
Starting hour	00h to 17h	(1 byte)
Starting minute	00h to 3Bh	(1 byte)
Mode 12/24 H in ASCII	(A/P/SPACE)	(1 byte in ASCII)
Number of days of the interval	00h to 1Fh	(1 byte)
Number of hours of the interval	00h to 17h	(1 byte)
Number of minutes of the interval	00h to 3Bh	(1 byte)

Protocol - Request to the printer

Language 1 for months of the year (1 byte):

Number	Language
00h	French
01h	English
02h	German
03h	Italian
04h	Spanish
05h	Norwegian
06h	Swedish
07h	Dutch
08h	Danish
09h	Japanese (Kanji)
0Ah	Portuguese
0Bh	Brasilian
0Ch	Hungarian
0Dh	Polish
0Eh	Turkish
0Fh	Czech
10h	Arab
11h	Chinese
12h	Korean
13h	Hebrew
14h	Russian
15h	Thai
16h	Greek
17h	Finnish
18h	Croatian
19h	Slovenian
1Ah	Vietnamese
1Bh	Iranian
1Ch	Bulgarian
1Dh	Indonesian
1Eh	Taiwanese
1Fh	Hegirien

Shift code 3 (112 bytes):
□ Shift code 3 parameters (7 bytes)

Table number for Monday	(01 to 07)	(1 byte)
Table number for Tuesday	(01 to 07)	(1 byte)
Table number for Wednesday	(01 to 07)	(1 byte)
Table number for Thursday	(01 to 07)	(1 byte)
Table number for Friday	(01 to 07)	(1 byte)
Table number for Saturday	(01 to 07)	(1 byte)
Table number for Sunday	(01 to 07)	(1 byte)

□ Tables 1 to 7 (15 bytes * 7)

Starting hour	(00h to 17h)
Starting minute	(00h to 3Bh)
Mode 12/24H in ASCII	(A/P/Space)
Number of hours of the interval	(00h to 17h)
Number of minutes of the interval	(00h to 3Bh)
5 bytes reserved	00h, 00h, 20h, 00h, 00h
5 bytes reserved	00h, 00h, 20h, 00h, 00h

Postdate 2 to 6 modulo (10 bytes)
postdate 2 modulo 0000 to 9999 (2 bytes)
postdate 3 modulo 0000 to 9999 (2 bytes)
postdate 4 modulo 0000 to 9999 (2 bytes)
postdate 5 modulo 0000 to 9999 (2 bytes)
postdate 6 modulo 0000 to 9999 (2 bytes)

Language 2 for months of the year (1 byte):
(the numbers are identical to month 1 of the year).

Date change time shift
This parameter also affects the "Day of the year" field

Shift direction (+ or -)	2dh or 20h
Hours	0 to 23 binary (17 h)
Minutes	0 to 59 binary (3Bh)

Protocol - Request to the printer

☐ **Request Shift Code 4 parameters (custom Shift Code)**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	BBh
Length (2 bytes)	00h, 00h

⬅ **Response**

BCh	Identifier (1 byte)
00h, 2Ah	Length (2 bytes)
xxh, xxh, ...	Data: - Shift Code 4 Custom parameter (42 bits)

■ Detail of data

Table number for Monday to Sunday
01 Monday
02 Tuesday
03 Wednesday
04 Thursday
05 Friday
06 Saturday
07 Sunday

Parameters for each day (x7)
- Starting hour   00h to 17h   (1 byte)
- Starting minute   00h to 3Bh   (1 bytes)
- Mode 12/24 h in ASCII   (A/P/SPACE)   1 bytes in ASCII
- Number of hours of the interval   00h to 17h   (1 byte)
- Number of minutes of the interval   00h to 3Bh   (1 byte)

❑ Request for current printing counter parameters

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	BEh
Length (2 bytes)	00h, 02h
Data: - counter parameters (2 bytes)	00h, 01h à 00h, 0Fh (1 to 15)

⬅ **Response**

BEh	Identifier (1 byte)
00h, 2Dh	Length (2 bytes)
31h	Data: - part of parameters: first byte following the Length (31h) definition to last byte.

Protocol - Request to the printer

❏ **Request cartridge Tag general information**

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	BFh
Length (2 bytes)	00h, 01h
Data:	xxh

◼ Detail of data

Channel
00h: side Ink cartridge
01h: side Additive cartridge

⬅ **Response**

BFh	Identifier (1 byte)
00h, 3Bh 00h, 36h	Length (2 bytes) - Ink cartridge Length (2 bytes) - Additive cartridge
xxh xxh xxh	Data: - Tag Type (1 byte) - Version (1 byte) - Batch number (10 bytes) - Expiration date (2 bytes) (MM/AA) - Volume (2 bytes) (current value) **If Tag = Ink** - Ink cartridge reference in place (6 bytes) - Associated Additive reference (6 bytes) - Plug & Play Inks (5x6 bytes) - List of 5 compatible inks **If Tag = Additive** - Additive cartridge reference in place (6 bytes) - Plug & Play Additives (5x6 bytes) - List of 5 compatible additives - Cartridge type (1 byte) - Tag Format(1 byte)

■ Detail of data

Tag Type
02h : ink cartridge
03h : additive cartridge

Cartridge Type
00h : standard
01h : cartridge used for rinsing

Tag Format
01h : Markem-Imaje origin
02h : no data
03h : no accessible tag

❑ **Request for general parameters by default**

COMPUTER   PRINTER

Request →

Identifier (1 byte)	CBh
Length (2 bytes)	00h, 00h

← Response

CDh	Identifier (1 byte)
xxh, xxh	Length (2 bytes)
Number of parameters (2 bytes)	1 to N
Binary (2 bytes)	Size M, in bytes, is dependent on the type of parameter processed.

Protocol - Request to the printer

◻ **Request for history of the thirty faults**

PRINTER

Request ➡

Identifier (1 byte)	CFh
Length (2 bytes)	00h, 00h

⬅ **Response**

D0h	Identifier (1 byte)
xxh, xxh	Length (2 bytes)
xxh, xxh FFh, NNh xxh, xxh, xxh... xxh, xxh, ... xxh, xxh xxh xxh	Data: - Number of records (2 bytes) - Fault number (2 bytes) -Time and date of notification (5 bytes) - Printer status on notification of the fault (26 bytes) (Print Unit information's) - Number of faults or warnings present at the same time as that treated (2 bytes) - List of faults or warnings (2 bytes)

■ Detail of data

Time and date notification

Days	xxh	1 byte
Month	xxh	1 byte
Year	xxh	1 byte
Hours	xxh	1 byte
Minute	xxh	1 byte

Printer status of notification of the fault

jet speed	xxh, xxh	2 bytes
Phase detection	xxh, xxh	2 bytes
Head temperature	xxh, xxh	2 bytes
Solvent added	xxh, xxh	2 bytes
Motor speed	xxh, xxh	2 bytes
Motor speed target	xxh, xxh	2 bytes
Pomp pressure	xxh, xxh	2 bytes
Pomp pressure target	xxh, xxh	2 bytes
Tank ink level	xxh, xxh	2 bytes
Hardware temperature	xxh, xxh	2 bytes
Ink temperature	xxh, xxh	2 bytes
Viscosity	xxh, xxh	2 bytes
Viscosity target	xxh, xxh	2 bytes

❑ Request warnings/faults (number and codes)

COMPUTER   PRINTER

Request ➡

Identifier (1 byte)	DAh
Length (2 bytes)	00h, 01h
Data: - fictitious value (1 byte)	00h

⬅ Response

D2h	Identifier (1 byte)
xxh, xxh	Length (2 bytes)
xxh xxh, xxh ... xxh, xxh	Data: - Total number of warnings/faults (1 to N) in the list (1 byte) - Warning or fault number for item 1 in the list (2 bytes) ... - Warning or fault number for item N in the list (2 bytes)

NOTE: Warning and fault numbers are organized as follows:
        Printing board $1000 \leq Faults < 1500 \leq Warnings$
        Print head $2000 \leq Faults < 2500 \leq Warnings$
        Ink circuit $4000 \leq Faults < 4500 \leq Warnings$

        ACM $4610 \leq Warnings \leq 4820$

Protocol - Request to the printer

❑ **Request active job number**

COMPUTER   **PRINTER**

Request ➡

Identifier (1 byte)	DBh
Length (2 bytes)	00h, 00h

⬅ **Response**

91h	Identifier (1 byte)
00h, 0Ah	Length (2 bytes)
xxh, xxh xxh	Data: - Job number (2 bytes) 1 to 255 0 if no message in production - Messages heading (8 bytes) ASCII Maximum 8 characters. Longer names will be truncated.

❑ Request for an autodating table

Request ➡

Identifier (1 byte)	DEh
Length (2 bytes)	00h, 01h
Data: - Type of the element of the table	xxh

■ Detail of data byte

Type of table (1 byte)

00h	Hours table
01h	Minute table
02h	Days of the week table
03h	Days of the year table
04h	Days of the month table
05h	Weeks table
06h	Months of the year table
07h	Years table
08h	Shift code 2 table
09h	Months of year table - language 1
0Ah	Shift code 3 table
0Bh	Months of year table - language 2
0Ch	Hijri calendar
0Dh	Shift code 1, alpha - 1-0
0Eh	Shift code 1, numerical
0Fh	Shift code 1, alpha
10h	Hour of date change
11h	First day of the week
12h	Suppress zero digit before figure
13h	Shift code 4

Protocol - Request to the printer

Response ←

DFh	Identifier (1 byte)
xxh xxh	Length (2 bytes)
xxh xxh, xxh	Data: - Type of table (1 byte) - Table (N byte)

■ Detail of data byte

Table

Hours table	24 x 3 characters ASCII
Minute table	60 x 3 characters ASCII
Days of the week table	7 x 3 characters ASCII
Days of the year table	366 x 3 characters ASCII
Days of the month table	31 x 3 characters ASCII
Weeks table	53 x 3 characters ASCII
Months of the year table	12 x 3 characters ASCII
Years table	10 x 3 characters ASCII
Shift code 2 table	366 x 3 characters ASCII
Months of year table - language 1	12 x 3 characters ASCII
Shift code 3 table	7 x 24 x 3 characters ASCII
Months of year table - language 2	2 x 3 characters ASCII
Hijri calendar	2 (+ (24 x 4) bytes
Shift code 1, alpha - 1-0	24 x 1 characters ASCII
Shift code 1, numerical	100 x 2 characters ASCII
Shift code 1, alpha	26 x 1
Hour of date change	HHMM (BCD)
First day of the week	00 to 06
Suppress zero digit before figure	0 h (disable) FFh (enable)
Shift code 4[1]	7 x 24 x 3

(1) each item in the table is described on three digits; the useful characters are right-justified with 1 or 2 space characters before them (20h).
Example: 20 20 41 : " A" (space,space,A)

☐ Request for the languages of the month to be printed

Request ➡

Identifier (1 byte)	E0h
Length (2 bytes)	00h, 03h
Data: - Current version (3 bytes)	92h, 32h, 00h

⬅ **Response**

C4h	Identifier (1 byte)
xxh, xxh	Length (2 bytes)
1 byte	Data: Number of available languages for the months in letter.
n byte	Name of languages separated by a character null (0x0)

■ Detail of data

See command BBh "languages of the month to be printed"

Protocol - Request to the printer

markem·imaje
A DOVER COMPANY

Protocol - Precisions/Example

Protocol - Precisions/Example

■ Reminders

❑ Binary, decimal, hexadecimal and ASCII conversion

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

A byte (8 bits) corresponds to two hexadecimal digits.
The symbol "h" signifies hexadecimal notation.

Example:

7	6	5	4	3	2	1	0	
0	1	0	0	1	0	1	0	binary data

4   A

4Ah in hexadecimal
or 74 in decimal

The ASCII standard associates an alphanumeric symbol with each byte.
Example: 4Ah corresponds to "J" in ASCII.

Protocol - Precisions/Example

❑ **ASCII character table for a job**

	2	3	4	5	6	7	8	9	A
0	SP	0	@	P	'	p	Ç	É	á
1	!	1	A	Q	a	q	ü	æ	í
2	"	2	B	R	b	r	é	Æ	ó
3	#	3	C	S	c	s	â	ô	ú
4	$	4	D	T	d	t	ä	ö	ñ
5	%	5	E	U	e	u	à	ò	Ñ
6	&	6	F	V	f	v	å	û	œ
7	'	7	G	W	g	w	ç	ü	¿
8	(	8	H	X	h	x	ê	ù	§
9	)	9	I	Y	i	y	ë	Œ	
A	*	:	J	Z	j	z	è	Ö	
B	+	;	K	[	k	{	ï	Ü	
C	,	<	L	\	l	|	î	¢	
D	-	=	M	]	m	}	ì	£	
E	.	>	N	^	n	~	Ä	Ø	
F	/	?	O	_	o		Å	ø	

Example:

ASCII characters	P	A	R	I	S
Hexadecimal code	50h/	41h/	52h/	49h/	53h

☐ **BCD (Binary Coded Decimal) encoding**

In BCD, numbers are represented as decimal digits, with each digit coded in four bits. One byte can therefore contain a two-digit number (00 to 99).
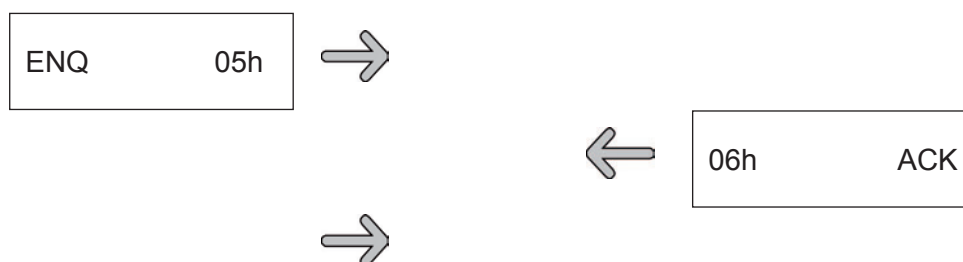
Example:

7	6	5	4	3	2	1	0	
0	1	1	1	0	1	0	0	binary data

7	4	74 in BCD

NOTE: *It is important to note that binary data in BCD encoding is different from the hexadecimal encoding for an identical decimal equivalent.*

Protocol - Precisions/Example

■ Example

❏ Sending a complete job to the library

PRODUCT:06/22/11 MADE IN FRANCE WEIGHT xxx KG

ENQ	05h

⟹

⟸ | 06h | ACK |

⟹

IDENTIFIER	9Bh

Length	01h ADh

Header:	
Total length of data	00h 00h 01h A8h
Checksum	00h 00h 1Ch 7Fh
Job type & version number	11h 01h
Job name ("EXAMPLE")	45h 58h 41h 4Dh 50h 4Ch 45h 00h 00h 00h
	00h 00h 00h 00h 00h 00h 00h 00h 00h 00h
Job number	00h 01h
Comment ("Summary")	53h 75h 6Dh 6Dh 61h 72h 79h 00h 00h 00h
	00h 00h 00h 00h 00h 00h 00h 00h 00h 00h
	00h 00h 00h 00h 00h 00h 00h 00h 00h 00h
	00h 00h

Parameters:	
Number of parameters	00h 04h

— Job parameters:	
Parameter type	01h
Parameter number	00h
Length	00h 12h

• Description of global parameters:		
Job direction	normal	
Horizontal character direction	normal	
Vertical character direction	normal	
Tacho mode	yes	10h
Manual start-up mode	no	
Printing mode	object	
Unit type	mm	

Number of repetitions	00h	

Speed measurement without tacho	no	02h
DTOP signal filter	200 µs	

Hijri calendar		05h
Tacho division		

Outbound margin	00h	03h
Return margin	00h	03h
Interval for repeating mode	00h	02h
Printing speed	01h	00h
Algorithm number	00h	00h

— Bar code parameters:						
Parameter type	04h					
Bar code number	01h					
Length	00h	1Eh				
• Description of parameters:						
Identifier (DATAMATRIX)	17h					
Bar code type	01h					
Length of parameters	00h	0Eh				
Mode	00h					
Height (number of cells)	00h	18h				
Right/left quiet zone	00h	0Ah				
Expansion of cells	01h					
Reserved	00h	00h	00h	00h	00h	00h
Length of motif data	00h	06h				
Motif data to encode	52h	45h	46h	31h	32h	33h
Length of key data	00h	00h				

Protocol - Precisions/Example

— Line Y coordinate parameters:	
Parameter type	08h
Parameter number	00h
Length of description data	00h 0Eh
Line 1 Y position	00 00
Line 2 Y position	00 08
Line 3 Y position	00 10
Line 4 Y position	00 18
Line 5 Y position	00 1F

— Number of lines parameters:	
Parameter type	09h
Number of line(s)	02h
Length	00h 04h

Text	
Delimiter for <u>first</u> line	0Ah

— Definition of <u>first block</u>:	
Parameter type	10h
Length	00h 12h
Character generator number (286d)	01h 1Eh
Algorithm number	00h 00h
Y reference	00h 01h
Reserved	00h
Expansion (number of frames)	01h
Generic parameters:	
Block locking	00h 00h
Character encoding mode (UTF8)	
Reserved	00h 00h
Length	00h 12h
Parameter type	10h

— Definition of text:	
P	50h
R	52h
O	4Fh
D	44h
U	55h
C	43h
T	54h
:	3Ah

markem·imaje
A DOVER COMPANY

— Definition of timestamp:

Type	1Ah	
Length	00h	0Eh
Month of year	50h	51h
Separator (/)	6Eh	
Day of month	49h	4Ah
Separator (/)	6Eh	
Year	55h	56h
Length	00h	0Eh
Type	1Ah	

— End of first block:

Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Eh
Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
Generic parameters	00h	00h
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

— Definition of second block:

Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
• Generic parameters:		
Block locking	00h	00h
Character encoding mode (UTF8)		
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

Protocol - Precisions/Example

— Definition of text:

	20h	
W	57h	
E	45h	
I	49h	
G	47h	
H	48h	
T	54h	
	20h	

— End of <u>second</u> block:

Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
Generic parameters	00h	00h
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

— Definition of <u>third</u> block:

Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
• Generic parameters:		
Block locking	01h	00h
Character encoding mode (UTF8)		
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

— Definition of external variable:

Type	12h	
Length	00h	0Bh
Variable number	01h	

Contents (Default: xxx)	78h	78h	78h
Variable number	01h		
Length	00h	0Bh	
Type	12h		

— End of third block:		
Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
Generic parameters	01h	00h
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

— Definition of fourth block:		
Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
• Generic parameters:		
Block locking	00h	00h
Character encoding mode (UTF8)		
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

— Definition of text:	
	20h
K	4Bh
G	47h

— End of fourth block:		
Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh

Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
Generic parameters	00h	00h
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

— Definition of fifth block:		
Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	01h
Reserved	00h	
Expansion (number of frames)	01h	
• Generic parameters:		
Block locking	01h	00h
Character encoding mode (UTF8)		
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	

— Definition of a tab:		
Type	1Eh	
Size in number of frames:(6)	06h	
Type	1Eh	

— Definition of bar code:		
Type	1Fh	
Identifier	01h	
Field type	00h	
Type	1Fh	

— End of fifth block:		
Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	01h

markem·imaje
A DOVER COMPANY

Reserved	00h	
Expansion (number of frames)	01h	
Generic parameters	01h	00h
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	
Delimiter for second line	0Ah	
— Definition of first block :		
Parameter type	10h	
Length	00h	12h
Character generator number (286d)	01h	1Bh
Algorithm number	00h	00h
Y reference	00h	09h
Reserved	00h	
Expansion (number of frames)	01h	
• Generic parameters:		
Block locking	00h	00h
Character encoding mode (UTF8)		
Reserved	00h	00h
Length	00h	12h
Parameter type	10h	
— Definition of a tab:		
Type	1Eh	
Size in number of frames:(192)	C0h	
Type	1Eh	
— Definition of text:		
	20h	
M	4Dh	
A	41h	
D	44h	
E	45h	
	20h	
I	49h	
N	4Eh	
	20h	
F	46h	
R	52h	

Protocol - Precisions/Example

A	41h
N	4Eh
C	43h
E	45h
— Definition of a tab:	
Type	1Eh
Size in number of frames:(44)	2Ch
Type	1Eh
— End of <u>first</u> block:	
Parameter type	10h
Length	00h 12h
Character generator number (286d)	01h 1Bh
Algorithm number	00h 00h
Y reference	00h 09h
Reserved	00h
Expansion (number of frames)	01h
Generic parameters	00h 00h
Reserved	00h 00h
Length	00h 12h
Parameter type	10h
End of job:	
Delimiter	0Dh
Alignment (size must be multiple of 4 bytes) :	
Filling data	00h
Type of entry (creation):	00h
Check byte:	7Bh

■ Standard fonts list

Number	Name	Format (H x W)
256	Latin	32 x 22
282	Latin	5 x 6
283	Latin	7 x 6
284	Latin	9 x 6
285	Latin	11 x 8
286	Latin	16 x 12
287	Latin	24 x 21
278	Latin Chim	5 x 6
279	Latin Chim	7 x 12
280	Latin Chim	11 x 17
281	Latin Chim	21 x 25
273	Arabic	7 x 6
274	Arabic	16 x 12
259	Arabic	24 x 20
270	Hebrew	5 x 6
271	Hebrew	7 x 6
272	Hebrew	16 x 12
273	Hebrew	24 x 20
266	Greek	7 x 6
267	Greek	16 x 12
268	Greek	24 x 21
257	Chinese	12 x 12
258	Chinese	16 x 15
262	Chinese	24 x 22
288	Japanese	7 x 10
289	Japanese	11 x 15
290	Japanese	21 x 25
275	Korean	7 x 8
276	Korean	9 x 7
277	Korean	11 x 8
263	Cyrillic	7 x 7
264	Cyrillic	16 x 13
265	Cyrillic	24 x 18

Protocol - Precisions/Example

■ V24 commands performance

Type of exchange (identifier)	
Request status (0xA6 / 0xE4)	10 ms
Request default (0xDA / 0xD2)	5 ms
Print acknowledgement in mode 9030 (0xCE)	600 µs
Print acknowledgement in mode 9410-9450 (0xE7)	550 µs
Select number of job (0x98)	1 ms
Send external variable in 9030 format (0x99) - Content = 20 characters ASCII	1,5 ms
Send external variable in 9410-9450 format (0xE8) - Content = 20 characters ASCII	1,5 ms
Request to send a new job in the store (0x9F / 0xC5)	500 ms <> 800 ms
Request to replace an existing job in the store (0x9F / 0xC5)	100 ms <> 200 ms
Request suppress a job in the store (0xC7 / 0xC5)	500 ms
Send job 9030 to print without datamatrix and without save in store (0xE3)	< 3 ms
Send job 9030 to print with datamatrix and without save in store (0xE3) - Content of datamatrix : font = 24 dots / 1 counter + 1 date	8 ms
Send job 9030 to print with datamatrix and without save in store (0xEF) - Content of datamatrix : font = 24 dots / 1 counter + 1 date	8 ms
Request to send a new job 9410-9450 in the Library with save in Store (0x9B) Data's size = 200 bytes.	600 ms < > 800 ms
Request to replace an existing job 9410-9450 in the Library with save in Store (0x9B) Data's size = 200 bytes.	80 ms < > 150 ms

Parallel link

Parallel link

■ Parallel link - Introduction

The parallel link can be used to select printing of jobs from the printer's the library, by DIRECT SELECTION or by JOB INCREMENT.

These two types of job selection entail connecting to the printer on the industrial interface terminal block board (available as an option).

◻ Configuration

The type of job selection used is configured in the following menu:
[icon] ► [icon] ► [icon] ► Operation:

Standard job: job selection via the operator interface.
Select job number: job selection by DIRECT SELECTION.
Library: job selection via the JOB INCREMENT interface.

◻ Characteristics

■ Inputs

All the input circuits feature high speed photocouplers.
Operating voltage: 5 to 35 V.
Regulated input current within the operating voltage range
Inputs D0 to D7 have a common connection COMDATA.
The inputs are active high.

■ Outputs

Two outputs are available:
– "PRINTG" output: Job synchronization.
– "COUNTR END" output: Counter end value.

These outputs feature high speed photocouplers with open collector phototransistors.
Maximum output current: 50 mA
Maximum operating voltage: 50 V.
The outputs are active low.

Parallel link

☐ **Connection**

The two types of job selection on the industrial interface board terminal blocks are connected in the same way as the serial link.
See the "Connection - General" section.
Wires are connected to the corresponding terminals on connectors J2, J3, J4 or J6 depending on the job selection type used.

■ Terminals common to the two selection types

D0	MODE-a	J4
D1	MODE-b	
D2		
D3		
D4	J2	PRINTG+
D5	PRINTG-	
D6		J6
D7	COUNTR END+	
COMDATA	COUNTR END-	

INC/TRG MESS-a   J3
INC/TRG MESS-b

■ Terminals for Job increment only

TRIGG-a

TRIGG-b

J2

LIBRARY SENSE-a

LIBRARY SENSE-b

J3

IMPORTANT: The shield of the connection cable used must be connected to the edge of the metal cable clamp on the printer. The same type of connection must be used on the computer.

Parallel link

■ Direct selection by job number

❑ Operation

■ Inputs

INC/TRG MESS: This input is used to confirm the job number present on data inputs D0 to D7 and trigger printing of the job (this input is not filtered).

D0 to D7: data for addressing the job number to be printed.

MODE = 1: addresses are coded as BCD (addressing 99 jobs)

MODE = 0: addresses are coded as hexadecimal (addressing 255 jobs)

■ Output

PRINTG: This output is active while the selected job is being printed.

❑ Signal diagram

D0 to D7
+ MODE

T1   T5

T6

INC/TRG
MESS

T2   T4

PRINTG

T3

PRINTING

T1:   Minimum time during which data D0 to D7 must be present before the
        rising edge of the INC/TRG MESS signal = 30 µs.

T2:   Maximum time between the rising edge on INC/TRG MESS and the active
        state of the PRINTG signal = 10 ms (this time depends on the job contents).

T3:   Time between activation of the PRINTG signal and the start of printing the
        job (this time depends on the head type, speed and printing delay for the
        selected job).

T4:   Minimum time between the end of printing (signal PRINTG inactive) and a
        new job selection = 10 ms.

T5:   Minimum time during which data D0 to D7 and MODE must be held after the
        rising edge of the INC/TRG MESS signal = 30 µs.

T6:   Width minimum of the pulse on INC/TRG MESS = 100 µs

Parallel link

■ Selection via the job increment interface

☐ Operation

With this type of selection, there are two modes of operation depending on the signal present on the MODE bit:
– MODE = 0: Direct selection by job number.
– MODE = 1: Selection by job number increment.

IMPORTANT: Direct selection is primarily useful for selecting the first in a series of jobs, after which the remaining jobs are selected by job number increment.

NOTE: *These two modes of operation may be used concurrently depending on the level of the MODE input.*

☐ Direct selection by job number

■ Inputs

– DATA D0 to D7:
Job number to print, in hexadecimal.

– INC/TRG MESS:
This input validates the job number present on data inputs D0 to D7.

– TRIGG:
This input triggers printing of the job selected with data inputs D0 to D7 and validated by the INC/TRG MESS input.

■ Output

– PRINTG:
This output is active while the selected job is being printed.

■ Signal diagram

T1:   Minimum time during which data D0 to D7 and MODE must be present before the rising edge of the INC/TRG MESS signal = 30 µs.

T2:   Minimum time between the rising edge of INC/TRG MESS and the rising edge of the TRIGG input = 10 ms (this time depends on the contents of the job).

T3:   Time between the rising edge of the TRIGG signal and printing of the first frame in the selected job. This time depends on the head type, speed and print delay of the selected job.

T4:   Minimum time between the end of printing (PRINTG signal returns to standby) and a new rising edge on the INC/TRG MESS input = 10 ms.

Parallel link

❑ Selection by job number increment

An initial active job must be initialized either via the operator interface or via the serial link, or via direct selection by job number.

■ Inputs

– INC/TRG MESS:

Each pulse on this input increments or decrements the job number active for printing, depending on the level of the LIBRARY SENSE input.

– LIBRARY SENSE:

When the LIBRARY SENSE input is low. the job number is incremented.
When the LIBRARY SENSE input is high. the job number is decremented.
After a change of state of the LIBRARY SENSE input, the first pulse on the INC/TRG MESS signal does not modify the number of the job selected for printing.

– TRIGG:

A pulse on this input triggers printing of the selected job.

■ Output

– PRINTG:

This output is active while the selected job is being printed.

■ Signal diagram

T1: Minimum time between the rising edge of INC/TRG MESS and the rising edge of the TRIGG input = 10 ms (this time depends on the contents of the job).

T2: Time between the rising edge of the TRIGG signal and printing of the first frame in the selected job (PRINTG signal active). This time depends on the head type, speed and print delay of the selected job.

T3: minimum Time for which the LIBRARY SENSE input must be present before the rising edge of the INC/TRG MESS input = 30 μs.

T4: minimum Time for which the LIBRARY SENSE input must be held after the rising edge of the INC./TRG MESS input = 1 ms.

T5: Minimum time between the end of printing (PRINTG signal returns to standby) and a new rising edge on the INC/TRG MESS input = 10 ms.

Parallel link

Job format specification

Job format specification

■ General Description

The formal description used on 9410-9450 explained below allows the description of any kind of marking/coding job.

■ General Structure:

Header		Parameters		Lines[1..8]		Flag = 0x0D End of Job

Number of lines is limited to 8

■ Header

Header is used to describe the job.

■ Parameters

The parameters are specific to the job and are used to define the print parameters and the variable elements according to the content of the job to be printed.
The parameters which possess several instantiations (example.:counters) are always ordered from 1 to N but an instantiation can be away (i.e.: we can have a list of counter 1,5,6,7,15)

■ Line Structure

Flag = 0x0A Start Line		Block 1		Block 2	- - - -	Block n

A line is a set of contiguous blocks. A block is a set of data that has the following common characteristics:
– Y position
– Font number
– Dilatation

The number of blocks is limited to 100. Line sizes are variable. The job size must not exceed 4 Ko of the jobs.

Lines must be aligned from lower Y position to upper Y position. The lower position is the gutter position

■ End of Job

This indicator is used as end of job. This indicator is mandatory.

Job format specification

■ Detailed Job Structure

❑ Header

Field	Size (bytes)
Total length of the file	4
Checksum	4
Job Type and Version Number	2
Job name	20
Job number	2
Summary	32

– Total length of the file:

Total number of all bytes in the job, including the 4 bytes of total length.

– Checksum:

Not used. Default value: 00h, 00h, 00h, 00h.

– Job Type : (msb byte of the word)

0x11 for job CIJ.
0x21 for production message

– Version number : (lsb byte of the word)

0x01

– Job name :

String with variable significative length terminated by \0.
String is in ASCII format.

– Job number :

0x0001 to 0x3E7

– Summary:

String with variable significative length terminated by \0. This field is used to summarise the job. String is in ASCII format.

❑ Parameters

Field	Size (bytes)
Number of parameters	2
Type and Parameter Number for this parameter Length Parameters description	2 2 N
Type and Parameter Number for this parameter Length Parameters description	2 2 N
………………………… …. …………………………	
Type and Parameter Number for this parameter Length Parameters description	2 2 N

Parameter area is a list of the various parameters.
The presence of a parameter is only required if the corresponding element is in the text (i.e: counter parameter with counter in the text)
The order is fixed and is defined with "Type" and "Parameter number".
Order = 0x01 then 0x02 then 0x03 etc….

Job format specification

   – Number of Parameters:

Number of parameters . If job contents no parameter this value = 0x0000.

     – For one parameter :

       – Type : (msb byte of the word) :

This field defines the type of associated parameter.

0x01	for print parameters
0x02	for counters
0x03	for postdates
0x04	for barcodes
0x05	for shiftcode
0x08	for LineY coordinates
0x09	for number of lines
0x0A	for DIN parameter

       – Parameter Number : (lsb byte of the word) :

This field specifies a particular type

[0x00]	for print parameters (associated with Type 0x01.
[0x01..0x0F]	for counter 1 to 15 (associated with Type 0x02).
[0x01..0x06]	for postdate 1 to 6 (associated with Type 0x03).
[0x01..0x04]	for barcode 1 to 4 (associated with Type 0x04).
[0x01]	for shiftcode (associated with Type 0x05).
[0x00]	for line Y coordinates (associated with Type 0x08.
[0x01..0x10]	to define number of lines (associated with Type 0x09).

   – Length

Total number of all bytes of this parameter, including Type, Parameter number and 2 bytes of length.

   – Parameters description

specific for each parameter

☐ **Print Parameters Definition**

Field	Size (bytes)
Type = 0x01	1 byte
Unused parameter number = 0	1 byte
Length = 0x0012	2 bytes
Job horizontal direction: 0 = normal / 1 = inverted	b7
Characters horizontal direction: 0 = normal / 1 = inverted	b6
Characters vertical direction: 0 = normal / 1 = inverted	b5
Tacho mode: 0 = no / 1 = yes	b4
Reserved = 0	b3
Printing mode: 0 = Object / 1 = Repetitive	b2
Unit type: 0 = mm / 1 = htrame	b1
Reserved = 0	b0
Multitop value : [0x00 .. 0xFF] (0x00 = inactive mode)	1 byte
Measure of speed without tachy: 0 = no / 1 = yes	b7
Dtop filter * 100µs [0x02 .. 0x7F]	b6 .. b0
Reserved = 0	b7
Not used	b6
Tacho division (in number of ½ tacho pulse) [0x01..0x3F]	b5 .. b0
Away margin [0x0003 .. 0x270F]	2 bytes
Return margin [0x0003 .. 0x270F]	2 bytes
Interval in repetitive mode [0x0002 .. 0x270F]	2 bytes
Conveyor speed in mm/s [0x0001.. [0x270F]	2 bytes
Algo number [0x0000 .. 0xFFFF]	2 bytes

– Type:
0x01 = print parameter

– Length :
Number of all bytes of this parameter, including the 4 first bytes of print parameters.

– Job horizontal direction :
normal : job is printed from the beginning to the end.
inverse : job is printed from the end to the beginning.

Job format specification

    – Characters horizontal direction :

normal : each character is printed from the beginning to the end.
inverse : each character is printed from the end to the beginning.

    – Characters vertical direction:

normal : each character is printed with normal vertical sens.
inverse : each character is printed with its vertical sens inverted.

    – Tacho mode:

no: job is printed with internal constant clock.
yes: job is printed with external variable clock.

    – Printing mode (reserved for extension always = 0) :

object: job is printed on time after start print
repetitive: job is printed as long as DTOP is active (DTOP mode) or that a new start print is received (manual mode)

    – Unit type:

0 = mm to calculate margin
1 = htrame clock to calculate margin.

    – Multitop value:

value: number of repetition of the printed job.
0x00 = inactive mode / 0x01 to 0xFF = active value

    – Measure of speed without tachy:

0 = no
1 = yes

    – Dtop filter:

value: filtration time of the photocell . Step of filtration = 200µs
0x02 to 0x7F = 100µs to 12,7ms active value

    – Tacho division:

value: number of ½ tacho pulse.
0x01 to 0x3F = active value

    – Away margin:

value: margin used in normal job sens. Defined in millimeter
3 to 9999 mm = active value

–   Return Margin:

value: margin used in inverse job sens. Defined in millimeter
3 to 9999 mm = active value

–   Interval in repetitive mode:

value: margin used in repetitive mode. Defined in millimeter
This value is the distance from end an beginning of 2 consecutive jobs.
2 to 9999 mm = active value .

–   Conveyor speed:

max speed of conveyor in mm/s

–   Algo Number:

specific algo number
0 = active value.

Job format specification

☐ **Counter Definition**

for one counter

Field	Bits msb.. lsb Binary
Type = 0x02	1 byte
Counter number = [0x01..0x0F]	1 byte
Length = 0x0031	2 bytes
Display leading zeros : 0 = no / 1 = yes 0 = reserved Init counter on top object 0 = no / 1 = yes Increment / Decrement 0 = increment / 1 = decrement Number of significant characters from 1 to 9	b7 b6 b5 b4 b3..b0
Unused = 0 Counter mode – Inactive state – Incrementing by external input A – Incrementing by external input B – Object increment – Job increment – Incrementing by preceding job counter – Other values : unused	 b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1
Unused = 0 Reserved =0 Reserved =0 Reset from external input A : 0 = no / 1 = yes Reset from external input B : 0 = no / 1 = yes Initialize counter when selecting the job to be printed ; 0 = no / 1 = yes Activation of state output on final value 0 = no / 1 = yes Overflow on next counter : 0 = no / 1 = yes	b7 b6 b5 b4 b3 b2 b1 b0
Start value	9 ASCII bytes
Final value	9 ASCII bytes
Counter step	9 ASCII bytes
Increment divider	4 binary bytes
Current value	9 ASCII bytes
Counting base 　first char 　last char	 1 ASCII byte 1 ASCII byte

A48601-A

It is an alphanumeric counter that goes from its current value to its final value
On each change its value increases or decreases by one counter step. The action of changing the value is performed on condition that a value called increment divider is counted.

Programmable inputs/outputs of the industrial interface related to the counters of the job are defined at the head initialization parameters level.

A job can have up to 15 counters.

Start value:
    from 000000000 to 999999999
    or from AAAAAAAAA to ZZZZZZZZZ
Final value:
    from 000000000 to 999999999
    or from AAAAAAAAA to ZZZZZZZZZ
Counter step:
    from 000000001 to 000000099
Increment divider:
    from 000001 to 999999
Current value:
    from 000 000000 to 999999999
    or from 000 000000 to 999999999 then AAAAAAAAA to ZZZZZZZZZ

In alphabetic mode the difference between begin and end value must be >= 4 digits

Initialization consists in initializing the current value with the start value:
  – after an init command
  – if it reached its final value
  – if the bit b2 init counter is positioned
Initialization of a counter involves initialization of all the counters of the chain.

  – Type:
0x02 = counter parameter

  – Counter number:
0x01 to 0x0F = define counter number

  – Length :
Number of all bytes of this parameter, including the 4 first bytes of print parameters.
  – Display leading zeros:

yes : non significant zeros are printed.
no : non significant zeros are not printed and replaced by spaces.

Job format specification

    – Init counter on dtop :

no : a dtop does not initialize the counter to its start value
yes : each dtop initializes the counter to its start value

    – Increment / Decrement:

increment : the counter value increases on each change (start val < final val)
decrement : the counter value decreases on each change (start val > final val)

    – Number of significant characters:

n : number of numeric characters used for the current value
0x00 to 0x09 = active value

    – Counter mode:

        – Incrementing by external input A
        – Incrementing by external input B

        This counter is incrementd by external input A (B). Input A (B) must be active >= 10ms

        – Object increment

        This counter is incremented after each job printed if object mode is active in job parameter

        – Job increment

        This counter is incremented after each job printed if repetitive mode is active in job parameter

        – Incrementing by preceding job counter

        The overflow of the preceding counter increments this counter parameter.

    – Reset from external input A (B) :

yes : this counter is reseted by external imput. This imput must be active > 10ms

    – Initialize counter when selecting the job to be printed :

no : this counter is managed with its last current value
yes : the counter is initialized when the job is selected

    – Activation of state output on final value :

yes: The state output is activated when this counter reaches its final value

    – Overflow on next counter :

yes: the overflow of this counter increments the next one.

    – Start value and final value :

Bounds of the current counter value.

–   Counter step :

Added or subtracted value on each evolution of the counter.
Min value = 000000001 / Max value = 000000099 (the 7 upper digits must be = 0)

–   Increment divider:

Number of repetitions of the current value before a counter evolution.

–   Current value:

If field 'Counter type' = 1 , current value of counter is recorded in this field otherwise this field is unused.

–   Counting base:

Defined the first character used for the numeration of this counter. Default value = '0'
Defined the last character used for the numeration of this counter. Default value = '9'
The difference between first and last character must be greater or equal to 4.

Job format specification

☐ **Postdate Definition**

for one postdate

Field	Bits msb.. lsb Binary
Type = 0x03	1 byte
Postdate number = [0x01..0x06]	1 byte
Length = 0x0008	2 bytes

Interval type:
    in number of days
    in number of weeks
    in number of months
Interval : from 0000 to 9999 (unit see type above) : 0 by default
	b15 b14
	0 0
	0 1
	1 0
	b13 .. b0

Modulo: 0001 to 9999 in number of days	2 bytes

– Type:

0x03 = postdate parameter

– Postdate number:

0x01 to 0x06 = define postdate number

– Length :

Number of all bytes of this parameter, including the 4 first bytes of print parameters.

– Interval

This delay is added to the current date.
Interval can be defined in number of days or weeks or months

– Modulo

The postdate is calculated as following: (current date + postdate duration in days) modulo (Modulo post-day value).

☐ Barcode Definition

for one barcode

Field	Bits msb.. lsb Binary
Type = 0x04	1 byte
Barcode number = [0x01..0x04]	1 byte
Length = XXXX	2 bytes
Identifier	1 byte
Free for evolution = 0000 Code type: – 0000 = mono-dimensional (barcode industrial and distribution) – 0001 = bi-dimensional (datamatrix)	b7 .. b4 b3 .. b0
Continuation of the structure depends on Code type field	
if Code type = 0000 mono-dimensional (barcode insdustrial and distribution)	
Parameters length = 0x0018	2 bytes
Mode: – Free for evolution = 0 – Control byte in the bar field – Control byte in the name field – Reverse mode	b7 .. b3 b2 b1 b0
Code height (in number of drops)	2 bytes
Quiet zone (right and left in number of rasters)	2 bytes
Border (around the main code in number of dots)	2 bytes
Width of narrow bars (in number of rasters)	2 bytes
Width of narrow spaces (in number of rasters)	2 bytes
Width of wide bars (in number of rasters)	2 bytes
Width of wide spaces (in number of rasters)	2 bytes
Dilatation (1 to 9 for distribution = ean,upc,hibc,code128)/ 1 for industrial = 2of5 and 39)	1 byte
Reserved for exension = 0	6 bytes
Bar field length = XXXX = N	2 bytes
Bar field to encode	N bytes
Name field length = 0	2 bytes

Job format specification

if Code type = 0001 = bi-dimensional (datamatrix)

Parameters length = 0x000E	2 bytes
Mode: – Free for evolution = 0 – Reverse mode	b7 .. b1 b0
Code height (in number of cells)	2 bytes
Quiet zone (right and left in number of rasters)	2 bytes
Cells dilatation (1 or 2)	1 byte
Reserved for exension = 0	6 bytes
Bar field length = XXXX = N	2 bytes
Bar field to encode	N bytes
Name field length = 0	2 bytes

– Type:
0x04 = barcode parameter

– Barcode number:
0x01 to 0x04 = define barcode number

– Length :
Number of all bytes of this parameter, including the 4 first bytes of print parameters.

– Identifier:
Barcode identifier:
00h = Interleave 2/5
01h = Code 39
02h = EAN13
04h = EAN8
06h = UPCA
08h = UPCE
17h = DATAMATRIX

– Code type :
0000 = mono-dimensional (barcode insdustrial and distribution)
0001 = bi-dimensional (datamatrix)

---

if Code type = 0000 mono-dimensional (barcode industrial and distribution)

– Parameters length :

Number of bytes of these parameters. Computation do not include bar field and name field but including the 2 bytes of bytes length.

– Mode :
- – Control byte in the bar field

    0 = do not insert control byte in the bar field
    1 = insert control byte in the bar field
- – Control byte in the name field

    0 = do not insert control byte in the name field
    1 = insert control byte in the name field
    If the control character is set the location for control byte is reserved in the field 'Bar Field' or 'Name Field'. The control character is always a fix character.
- – Reverse

    0 = barcode is printed in normal mode. Quiet zone is not marked and color bars are not inverted.
    1 = barcode is printed in video inverted mode. Quiet zone is black and color bars are inverted.
    Reverse is only applied to 'Bar field' no to 'Name field'

– Code height :

0 to 32 = code height in number of drops.
This field is only used to print "Bar field". To print "Name field" it is the 'Gc number 'in bloc parameters that is used. The height defined by gc number is not used to define height of "Bar field".

– Quiet zone :

0 to 25 = right and left in number of rasters

– Border

0 to 4 = Number of dots around the main code
- – Width of narrow bars :
- – Width of narrow spaces :
- – Width of wide bars :
- – Width of wide spaces :

1 to 4 for narrow and 2 to 9 for wide

– Dilatation:

1 to 9 for barcode distribution / 1 barcode insdustrial
The dilatation of the block is not managed by the bar field only by the name field. The bar field uses the value of dilatation which is defined in the parameters of barcode

Job format specification

   – Bar field length

Number of bytes of Bar field. If bar field is empty Bar field length = 0. The dilatation is never applied to the border.

   – Bar field

This field is composed of symbols to be encoded. The format depends on the type of code
This field can contain the following elements: Symbol and (or) Counter and (or) Date/Postdate and (or) Shiftcode and (or) external variable.
See description of the different elements and separators in preceding paragraphs.
This field is optional and is only present if Bar field length # 0.

---

if Code type = 0001 = bi-dimensional (datamatrix)

   – Parameters length :

Number of bytes of this parameters. Computation do not include bar field and name field

   – Mode :

      – Reverse

       0 = barcode is printed in normal mode. Quiet zone is not marked and color cells are not inverted.
       1 = barcode is printed in video inverted mode. Quiet zone is black and color cells are inverted.

   – Code height :

8 to 32 = number of cells. This value is calculated with the cells dilatation.

   – Quiet zone :

0 to 25 = right and left in number of rasters

– Cells dilatation:

1 or 2
The dilatation of the block is not managed by the bar field only by the name field. The bar field uses the value of dilatation which is defined in the parameters of barcode

– Bar field length

Number of bytes of Bar field. If bar field is empty Bar field length = 0

– Bar field

This field is composed of symbols to be encoded. The format depends on the type of code
This field can contain the following elements: Symbol and (or) Counter and (or) Date/Postdate and (or) Shiftcode and (or) external variable.
See description of the different elements and separators in preceding paragraphs.
This field is optional and is only present if BarField length # 0.

Job format specification

☐ **Shiftcode definition**

Field	Bits msb.. lsb Binary
Type = 0x05	1 byte
Shiftcode type = [0x01..0x03]	1 byte
Length = 0x0009	2 bytes
Start hour [0..23] or [1..12]	1 byte
Start minute [0..59]	1 byte
Start mode 12h/24h ['A' or 'P' or 'Space']	1 byte
Duration in number of hours [0..23]	1 byte
Duration in number of minutes [0..59]	1 byte

– Type:

0x05 = shiftcode parameter

– Shiftcode type:

0x01 to 0x03

This shiftcode allows printing the same value during time intervals defined for all days . It is defined by a start time (HH:MN) and a time interval (HH:MN).

– Length :

Number of all bytes of this parameter, including 2 bytes of the length and 2 first bytes of each parameter.

– Start Hour :

Hour of the beginning in number of hours
00 to 23 in 24h mode
01 to 12 in 12h mode

– Start Minute :

Minute of the beginning in number of minutes
00 to 59

– Mode:

Mode of the start time
character 'A' for AM / character 'P' for PM / Space character for 24h mode

– Duration in number of hours:

Interval between two codes in number of hours.
00 to 23

– Duration in number of minutes:

Interval between two codes in number of minutes.
00 to 53

Job format specification

❑ Guide line Y-coordinates (reserved for the job editor)

Field	Bits msb.. lsb Binary
Type = 0x08	1 byte
Unused parameter number = 0	1 byte
Length = XXXX	2 bytes
Y-position for line 1	2 bytes
Y-position for line 2	2 bytes

    repeated for all lines

– Type:

0x08 = Guide line Y-coordinates

– Length :

Number of all bytes of this parameter, including the 4 first bytes of this parameter.

– Y-Position for line n

Absolute Y-position of guide line in dots
The Y positions must be by listed in growing position of the line : 1 to N.
value = 0x01 to 0x20

❑ Number of printed lines

Field	Bits msb.. lsb Binary
Type = 0x09	1 byte
Number of lines = 0x01 to 0x08	1 byte
Length = 0x0004	2 bytes

If job contents no text this bloc is away.
This type is always a job parameter

– Type:

0x09 = Line

– Number of lines

Number of lines in this job.
value list = line number 0x01 to 0x08

– Length :

Number of all bytes of this parameter, including the 4 first bytes of this parameter

☐ Din parameters

Field	Bits msb.. lsb Binary
Type = 0x0A	1 byte
Unused parameter number = 0	1 byte
Length = 0x0006	2 bytes
Tabulation	2 bytes

– Type:

0x0A = Din parameters

– Length:

Number of all bytes of this parameter, including the 4 first bytes of print parameters.

– Tabulation :

Number of Htram between the Din messages value = 0x01 to 0x63

Job format specification

■ Lines definition

```
┌─────────────────────────────────────┐
│     Line delimiter = 0Ah (1 byte)    │
└─────────────────────────────────────┘

         ┌──────────────────┐
         │  Bloc Parameters  │
         │   10h <…>10h      │
         └──────────────────┘

    ┌────────────────────────────────────────────┐
    │                                              │
    │                                              │
    │  Symbol                                    ▯ │
    │                                              │
    │  Graphics                    11h <------> 11h│
    │                                              │
    │  External Var                12h <------> 12h│
    │                                              │
    │  Internal Editor Spacing     15h <------> 15h│
    │                                              │
    │  Date/time/Pdate/Shiftcode   1Ah <------> 1Ah│
    │                                              │
    │  Pdate/Shiftcode             1Bh <------> 1Bh│
    │                                              │
    │  Counter                     1Ch <------> 1Ch│
    │                                              │
    │  Spacing                     1Eh <------> 1Eh│
    │                                              │
    │  Bar Code                    1Fh <------> 1Fh│
    │                                              │
    └────────────────────────────────────────────┘

         ┌──────────────────┐
         │  Bloc Parameters  │
         │   10h <…>10h      │
         └──────────────────┘
```

The block structure is symmetrical in order to facilitate printing in inverted sense mode.
A line starts always with a delimiter = 0x0A
A block in never empty.

Delimiters list of the several elements that aren't fixed symbols:
10h   : Bloc parameters
11h   : Graphics
12h   : External variable
15h   : Internal editor spacing
1Ah   : Standard date/time/shiftcode
1Bh   : Postdate/shiftcode
1Ch   : Counters
1Eh   : Spacing
1Fh   : Bar code

Job format specification

❑ Bloc Parameters Structure

Field	Bits msb.. lsb Binary
Type = 0x10	1 byte
Length = 0x0012	2 bytes
Gc number [0x0001 to 0xXXXX]	2 bytes
= 0	2 bytes
Y reference (in number of dot 1 to 32)	2 bytes
0x00 (not used)	1 byte
Dilatation in number of rasters (1 to 9)	1 byte
Generic parameters	b15 ..b9
Lock block : 0 = no / 1 = yes (only used by job editor)	b8
Unused = 0	b7
Unused = 0	b6
Unused = 0	b5
Unused = 0	b4
Unused = 0	b3
Unused = 0	b2
Unused = 0	b1
Unused = 0	b0
= 0x0000	2 bytes
Length = 0x0012	2 bytes
Type = 0x10	1 byte

A block can contain different elements except for barcode
Bar codes are managed through a specific block

– Type:
0x10 = identification bloc parameters.

– Length:
Length of all bytes of this block, including length and type

– Gc Number
number of the character generator

– Y reference:
The absolute Y coordinate of the block.

    – Dilatation

characters dilatation in number of rasters: from 1 to 9

    – Generic parameters :

        – Lock block (only used by job editor)

            0 = unlock
            1 = lock: the parameters and text of this block is locked and can not be modified by job editor.

❑ Symbols

A symbol is always defined in UTF8 format.

UTF8 format
0xxxxxxxx   : symbol encoded on 1 byte : 1 to 7 bits (0x20 to 0x7F)
110xxxxx 10xxxxxx : symbol encoded on 2 bytes : 8 to 11 bits (0x80 to 0x7FF)
1110xxxx 10xxxxxx 10xxxxxx : symbol on 3 bytes : 12 to 16 bits (0x800 to 0xFFFF)

Job format specification

☐ Graphics

Field	Bits msb.. lsb Binary
Type = 0x11	1 byte
Length = XXXX	2 bytes
Drops number of raster descriptions	1 byte
Number of descriptions (=N) Raster descriptions set Number of descriptions (=N)	2 bytes N x bytes 2 bytes
Drops number of raster descriptions	1 byte
Length = XXXX	2 bytes
Type = 0x11	1 byte

– Type:
0x11 = identification Graphics.

– Length
Length of all bytes of this block, including length and type.

– Drops Number :
Number of drops of rasters.

– Number of description:
Number of description.

– Raster description:
Raster description in vertical binary format.

NOTA: The dilatation is not managed by the graphic block.

markem·imaje
A DOVER COMPANY

Job format specification

☐ External Variables

Field	Bits msb.. lsb Binary
Type = 0x12	1 byte
Length = XXXX	2 bytes
Number : 0xXX	1 byte
Variable content description	n bytes
Number : 0xXX	1 byte
Length = XXXX	2 bytes
Type = 0x12	1 byte

– Type:

0x12 = identification external variables.

– Length

Length of all bytes of this block, including length and type.

– Number

Current number in this job: 1 to 10

External variables are limited to 10 per job.
This block is used to identify part of a text to be modified from an external device.
An external variable can not be include an other external variable.
An external variable can not be define in variable element but surround this element.

The content appears as a set of job elements organized as a job. This field can contain the following elements:

Symbol
Counter
Date/Postdate
Shiftcode

Nota: Barcode is never included in external variable but external variable can be included in barcode.

See description of the different elements and separators in other paragraphs.

Job format specification

☐ **Internal editor spacing**

Field	Bits msb.. lsb Binary
Type = 0x15	1 byte
Number of white raster representing a spacing from 1 to 255	1 byte
Type = 0x15	1 byte

The number of spacing blocks is not limited.
This block is reserved to internal editor and can not be used in barcode field

NOTA: *The dilatation is not managed by the spacing bloc.*

☐ Date Time Postdate

Field	Bits msb.. lsb Binary
Type = 0x1A	1 byte
Length = XXXX	2 bytes

– Hour

41h / 42h.seconds	by default : 00 to 59
43h / 44h minutes	by default : 00 to 59
45h / 46h hours	by default : 00 to 23
47h / 48h mode (AM or PM or ' ')	by default : ' '

– Date

49h / 4Ah day of the month	by default : 01 to 31
4Bh / 4CH / 4Dh day of the year	by default: 001 to 366
4Eh / 4Fh week of the year	by default : 01 to 53
50h / 51h month of the year	by default : 01 to 12
52h / 53h / 54h month of the year in letters (language 1) : JAN-DEC	
55h / 56h year	by default : 00 to 99

– Postdate 1

57h / 58h postday of the month	by default: 01 to 31
59h / 5Ah / 5Bh postday of the year	by default: 001 to
5Ch / 5Dh postweek of the year	by default : 01 to 53
5Eh / 5Fh postmonth of the year	by default : 01 to 12
60h / 61h / 62h postmonth of the year in letters (language 1) : JAN-DEC	
63h / 64h postyear	by default : 00 to 99

– Shiftcode 1

65h shiftcode 1 letter	by default : A to Z
66h /67h shiftcode 1 numeral	by default : 01 to 99
68h shiftcode 1 letter	by default : A to Z – O – I

– Miscellaneous

69h day of the week	by default : 1 to 7
6Ah / 6Bh / 6Ch postday of the year modulo programmable	by default: 000 to 999
6Dh separator	:
6Eh separator	/
6Fh separator	.
70h separator	SP (space)
71h / 72h / 73h month of the year in letters (language 2) : JAN-DEC	
74h / 75h / 76h postmonth of the year in letters (language 2) : JAN-DEC	

Job format specification

   – Extension Date (unused today 77h to E2h : values = 0x00)

   – Century (always = 20)

E2h : century	0x32
E3h : century	0x30

Length = XXXX	2 bytes

Type = 0x1A	1 byte

   – Type:

0x1A = identification date variables.

   – Length

Length of all bytes of this block, including length and type.

Date/Time/Postdate elements are coded with ASCII characters that address a table containing the elements to be printed.
Elements begin at 41h so that confusion can't be made between elements and delimiters.

☐ Postdate extended

Field	Bits msb.. lsb Binary
Type = 0x1B	1 byte
Length = XXXX	2 bytes

– Postdate 2

41h / 42h postday of the month   by default : 01 to 31
43h / 44h / 45h postday of the year by default: 001 to 366
46h / 47h / 48h postday of the year modulo programmable by default: 000 to 999
49h / 4Ah postweek of the year   by default : 01 to 53
4Bh / 4Ch postmonth of the year   by default : 01 to 12
4Dh / 4Eh postyear   by default : 00 to 99
4Fh / 50h / 51h postday of the week in letters (week of the year definissable user table)
52h / 53h / 54h postmonth of the year in letters (month of the year definissable user table)

– Horodating table

55h / 56h / 57h table of the hours
58h / 59h / 5Ah table of the minutes
5Bh / 5Ch / 5Dh table of the month of the year
5Eh / 5Fh / 60h table of the day of the week
61h / 62h / 63h table of the day of the month
64h / 65h / 66h table of the day of the year
67h / 68h / 69h table of the week of the year
6Ah / 6Bh / 6Ch table of the year
6Dh / 6Eh / 6Fh table of the shiftcode 2

– Miscellaneous

70h separator   :
71h separator   /
72h separator   .
73h separator   SP (space)

– Shiftcode 3
74h / 75h / 76h table of the shiftcode 3

Job format specification

– Postdate 3

77h / 78h postday of the month   by default : 01 to 31
79h / 7Ah / 7Bh postday of the year by default: 001 to 366
7Ch / 7Dh / 7Eh postday of the year modulo programmable by default: 000 to 999
7Fh / 80h postweek of the year   by default : 01 to 53
81h / 82h postmonth of the year   by default : 01 to 12
83h / 84h postyear   by default : 00 to 99
85h / 86h / 87h postday of the week in letters (week of the year definissable user table)
88h / 89h / 8Ah postmonth of the year in letters (month of the year definissable user table)

– Postdate 4

8Bh / 8Ch postday of the month   by default : 01 to 31
8Dh / 8Eh / 8Fh postday of the year by default: 001 to 366
90h / 91h / 92h postday of the year modulo programmable by default: 000 to 999
93h / 94h postweek of the year   by default : 01 to 53
95h / 96h postmonth of the year   by default : 01 to 12
97h / 98h postyear   by default : 00 to 99
99h / 9Ah / 9Bh postday of the week in letters (week of the year definissable user table)
9Ch / 9Dh / 9Eh postmonth of the year in letters (month of the year definissable user table)

– Postdate 5

9Fh / A0h postday of the month   by default : 01 to 31
A1h / A2h / A3h postday of the year by default: 001 to 366
A4h / A5h / A6h postday of the year modulo programmable by default: 000 to 999
A7h / A8h postweek of the year   by default : 01 to 53
A9h / AAh postmonth of the year   by default : 01 to 12
ABh / ACh postyear   by default : 00 to 99
ADh / AEh / AFh postday of the week in letters (week of the year definissable user table)
 B0h / B1h / B2h postmonth of the year in letters (month of the year definissable user table)

– Postdate 6

B3h / B4h postday of the month   by default : 01 to 31
B5h / B6h / B7h postday of the year by default: 001 to 366
B8h / B9h / BAh postday of the year modulo programmable by default: 000 to 999
BBh / BCh postweek of the year   by default : 01 to 53
BDh / BEh postmonth of the year   by default : 01 to 12
BFh / C0h postyear   by default : 00 to 99
C1h / C2h / C3h postday of the week in letters (week of the year definissable user table)
C4h / C5h / C6h postmonth of the year in letters (month of the year definissable user table)

– Separator

C7h : separator   (
C8h : separator   )
C9h : separator   -- Century (always = 20)
CAh : century   0x32
CBh : century   0x30

Length = XXXX   2 bytes

Type = 0x1A   1 byte

– Type:
0x1B = identification Postdate variables.

– Length
Length of all bytes of this block, including length and type.

Job format specification

☐ Counter

Field	Bits msb.. lsb Binary
Type = 0x1C	1 byte
Number of the counter from 1 to 15	1 byte
Type = 0x1C	1 byte

The number of counter is not limited.

☐ Spacing

Field	Bits msb.. lsb Binary
Type = 0x1E	1 byte
Number of white raster representing a spacing from 1 to 255	1 byte
Type = 0x1E	1 byte

The number of spacing blocks is not limited.

NOTA: The dilatation is not managed by the spacing bloc.

☐ Barcode

Field	Bits msb.. lsb Binary
Type = 0x1F	1 byte
Identifier barcode = 0x01 to 0x04	1 byte
Field type = 0x00 for bar	1 byte
Type = 0x1F	1 byte

Bar codes are managed through a specific block.
The number of barcode with same bloc parameters is not limited.

■ End of Job

Code 0Dh indicates the end of job

■ 9410-9450 Network Interface Manual - Revision

. Revision index A corresponds to the first edition of this manual.
. The revision index is changed every time the document is revised.

Issue date	Document revision index	Software index
12/2014	A	