

KPM202

KPM203

KPM302

KPM303

TK202

TK203

TK302

TK303

CUSTOM ENGINEERING S.p.A.
Str. Berettine 2
43010 Fontevivo (PARMA) - Italy
Tel. : +39 0521-680111
Fax : +39 0521-610701
http: www.custom.biz

Customer Service Department:
Email : support@custom.it

© 2012 CUSTOM ENGINEERING S.p.A. – Italy. All rights reserved. Total or partial reproduction of this manual in whatever form, whether by printed or electronic means, is forbidden. While guaranteeing that the information contained in it has been carefully checked, CUSTOM ENGINEERING S.p.A. and other entities utilized in the realization of this manual bear no responsibility for how the manual is used. Information regarding any errors found in it or suggestions on how it could be improved are appreciated. Since products are subject to continuous check and improvement, CUSTOM ENGINEERING S.p.A. reserves the right to make changes in information contained in this manual without prior notification.

The pre-installed multimedia contents are protected from Copyright CUSTOM ENGINEERING S.p.A. Other company and product names mentioned herein may be trademarks of their respective companies. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. CUSTOM ENGINEERING S.p.A. assumes no responsibility with regard to the performance or use of these products.

THE IMAGES USED IN THIS MANUAL ARE USED AS AN ILLUSTRATIVE EXAMPLES. THEY COULDN'T REPRODUCE THE DESCRIBED MODEL FAITHFULLY.

UNLESS OTHERWISE SPECIFIED, THE INFORMATION GIVEN IN THIS MANUAL ARE REFERRED TO ALL MODELS IN PRODUCTION AT THE ISSUE DATE OF THIS DOCUMENT.



The format used for this manual improves use of natural resources reducing the quantity of necessary paper to print this copy.

GENERAL SAFETY INFORMATION

Your attention is drawn to the following actions that could compromise the characteristics of the product:

- Read and retain the instructions which follow.
- Follow all indications and instructions given on the device.
- Make sure that the surface on which the device rests is stable. If it is not, the device could fall, seriously damaging it.
- Make sure that the device rests on a hard (non-padded) surface and that there is sufficient ventilation.
- When positioning the device, make sure cables do not get damaged.
- Use the type of electrical power supply indicated on the device label. If uncertain, contact your dealer.
- Make sure the electrical system that supplies power to the device is equipped with a ground wire and is protected by a differential switch.
- Do not block the ventilation openings.
- Do not insert objects inside the device as this could cause short-circuiting or damage components that could jeopardize printer functioning.
- Do not carry out repairs on the device yourself, except for the normal maintenance operations given in the user manual.
- Make sure that there is an easily-accessible outlet with a capacity of no less than 15A in the vicinity of where the device is to be installed.
- Periodically perform scheduled maintenance on the device to avoid dirt build-up that could compromise the correct, safe operation of the unit.
- Before any type of work is done on the machine, disconnect the power supply.
- Do not touch the head heating line with bare hands or metal objects. Do not perform any operation inside the printer immediately after printing because the head and motor tend to become very hot.

GENERAL INSTRUCTIONS

CUSTOM ENGINEERING S.p.A. declines all responsibility for accidents or damage to persons or property occurring as a result of tampering, structural or functional modifications, unsuitable or incorrect installations, environments not in keeping with the equipment's protection degree or with the required temperature and humidity conditions, failure to carry out maintenance and periodical inspections and poor repair work.



THE CE MARK AFFIXED TO THE PRODUCT CERTIFY THAT THE PRODUCT SATISFIES THE BASIC SAFETY REQUIREMENTS.

The device is in conformity with the essential Electromagnetic Compatibility and Electric Safety requirements laid down in Directives 2006/95/CE and 2004/108/CE inasmuch as it was designed in conformity with the provisions laid down in the following Standards:

- EN 55022 Class B (*Limits and methods of measurements of radio disturbance characteristics of Information Technology Equipment*)
- EN 55024 (*Information Technology Equipment – Immunity characteristics – Limits and methods of measurement*)
- EN 60950 (*Safety of information equipment including electrical business equipment*)



GUIDELINES FOR THE DISPOSAL OF THE PRODUCT

The crossed-out rubbish bin logo means that used electrical and electronic products shall NOT be mixed with unsorted municipal waste. For more detailed information about recycling of this product, refer to the instructions of your country for the disposal of these products.

- Do not dispose of this equipment as miscellaneous solid municipal waste, but arrange to have it collected separately.
- The re-use or correct recycling of the electronic and electrical equipment (EEE) is important in order to protect the environment and the wellbeing of humans.
- In accordance with European Directive WEEE 2002/96/EC, special collection points are available to which to deliver waste electrical and electronic equipment and the equipment can also be handed over to a distributor at the moment of purchasing a new equivalent type.
- The public administration and producers of electrical and electronic equipment are involved in facilitating the processes of the re-use and recovery of waste electrical and electronic equipment through the organisation of collection activities and the use of appropriate planning arrangements.
- Unauthorised disposal of waste electrical and electronic equipment is punishable by law with the appropriate penalties.

INDEX

1 INTRODUCTION	5
1.1 Command description	5
1.2 Print direction	6
2 ESC/POS™ EMULATION	7
3 SVELTA EMULATION	107
4 ALIGNMENT: PRACTICAL APPLICATIONS	177
4.1 Alignment commands: ESC/POS™ emulation	177
4.2 Alignment commands: SVELTA emulation	179
5 PAPER SPECIFICATIONS	181
5.1 Paper with alignment notch	181
5.2 Ticket with hole	182
5.3 Paper with labels	183
5.4 Paper with one-dimensional barcode	184
5.5 Ticket with RFID tag	185
6 COMMANDS INDEX	189

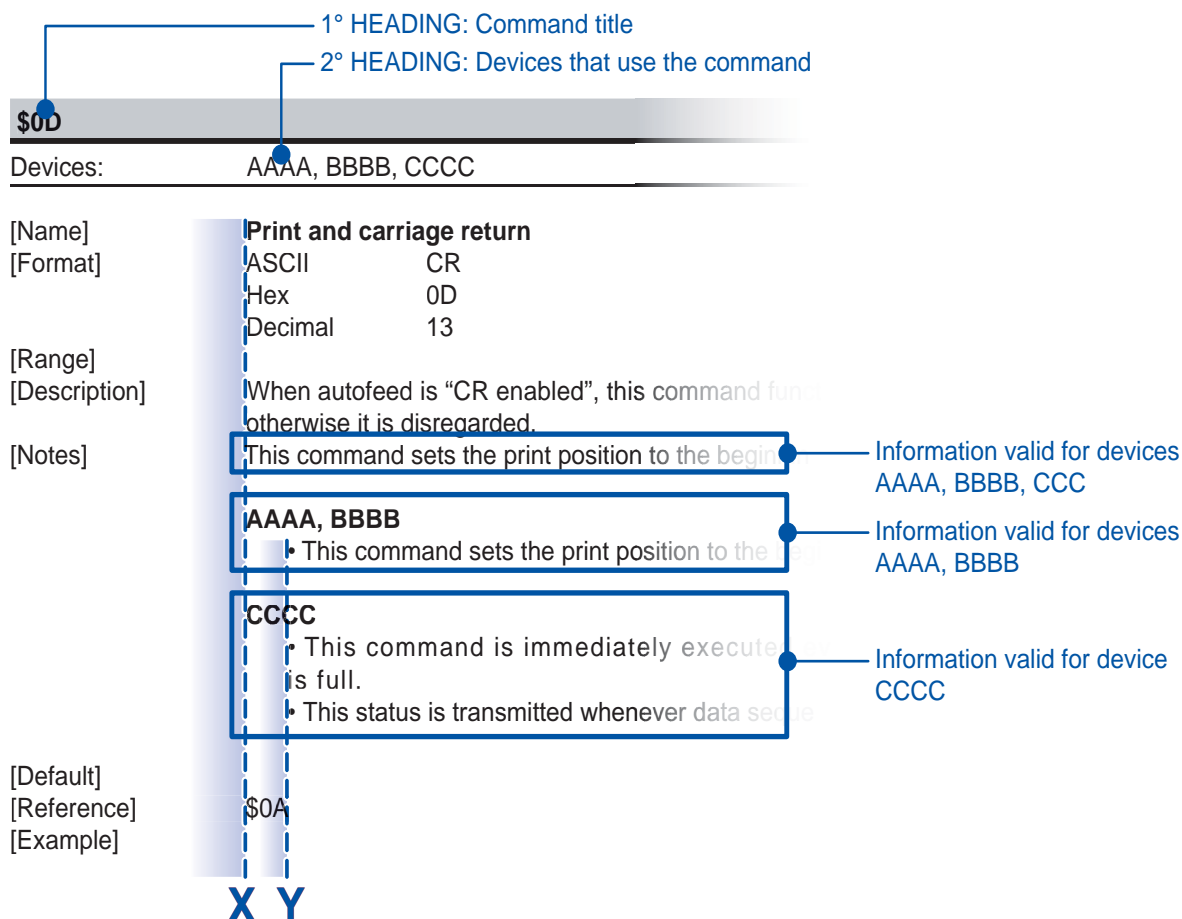
1 INTRODUCTION

1.1 Command description

Each command reported in this manual is described as shown in the following picture. In the first heading line (grey colour) is reported the hexadecimal command value. In the second heading line are listed the devices on which it is possible to use the command (for example device AAAA).

The next fields give all the information useful to use the command.

[Name] Command title
 [Format] ASCII, hexadecimal and decimal command value.
 [Range] Limits of the values the command and its variables can take
 [Description] Description of command function
 [Notes] Additional information about command use and settings .
 [Default] Default value of the command and its variables.
 [Reference] Pertaining commands related to described command.
 [Example]



The information reported in the picture are aligned with line X or line Y:

LINE X Description valid for all the devices listed in the second heading line.

LINE Y Description valid for a specific device (written in bold).

LEGEND

\$ indicates the representation of the command hexadecimal value (for example \$40 means HEX 40).

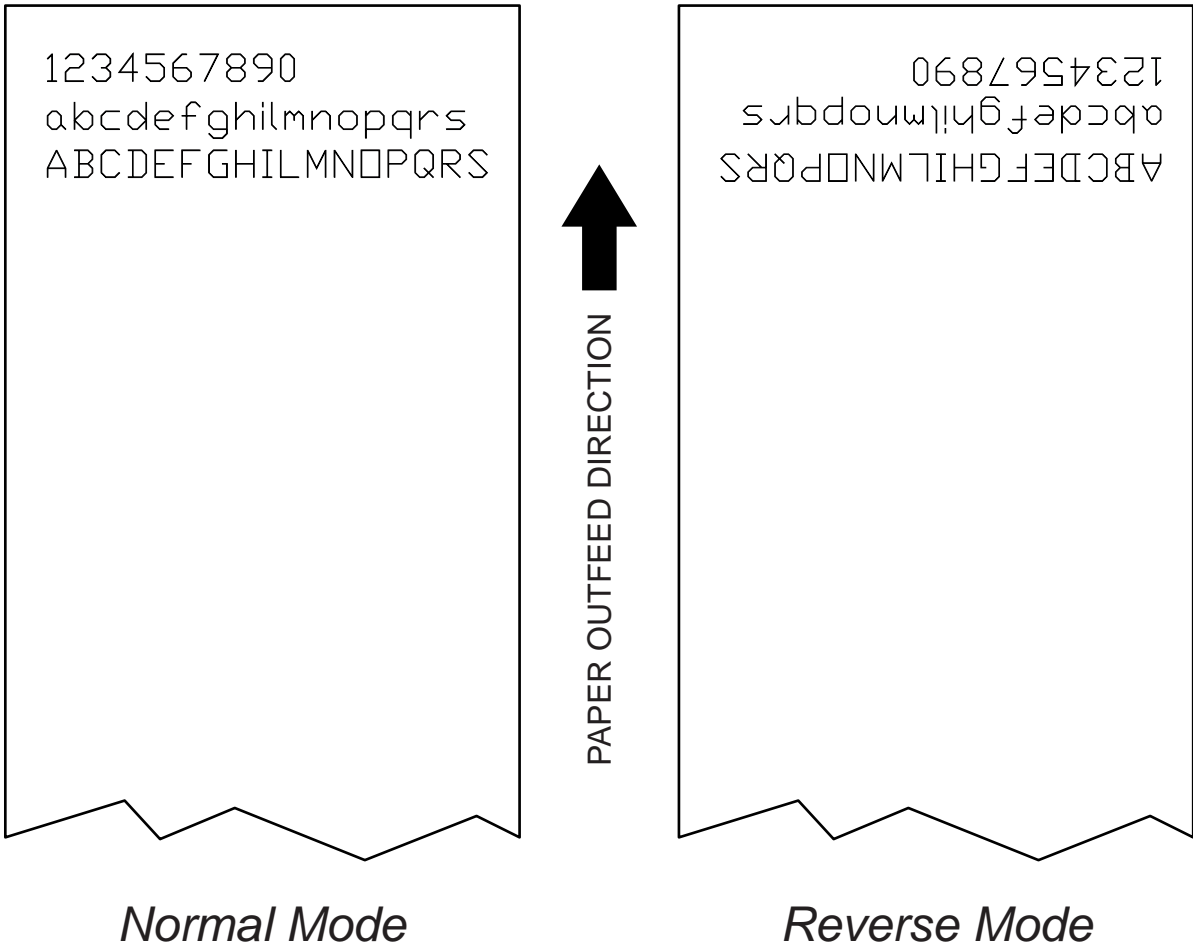
{ } indicates an ASCII character not performable.

n, m, t, x, y are optional parameters that can have different values.

Introduction

1.2 Print direction

The device has two printing direction which can be selected by means of the control characters: normal e reverse.



2 ESC/POS™ EMULATION

The following table lists all the commands for function management in ESC/POS Emulation of the device. The commands can be transmitted to the device at any moment, but they will only be carried out when the commands ahead of them have been executed. The commands are carried out when the circular buffer is free to do so.

COMMAND DESCRIPTION TABLE

Com. HEX	Com. ASCII	Description
PRINT COMMANDS		
\$0A	LF	Print and line feed
\$0D	CR	Print and carriage return
\$1B \$4A	ESC J	Print and feed paper
\$1B \$64	ESC d	Print and feed paper n lines
\$1C \$82	FS { }	Print date
\$1C \$83	FS { }	Print time
LINE SPACING COMMANDS		
\$1B \$30	ESC 0	Select 1/8-inch line spacing
\$1B \$32	ESC 2	Select 1/6-inch line spacing
\$1B \$33	ESC 3	Set line spacing using minimum units
CHARACTER COMMANDS		
\$18	CAN	Cancel current line trasmitted
\$1B \$20	ESC SP	Set character right-side spacing
\$1B \$21	ESC !	Set print mode
\$1B \$25	ESC %	Select/cancel user-defined character set
\$1B \$26	ESC &	Define user-defined characters
\$1B \$2D	ESC -	Turn underline mode on/off
\$1B \$3F	ESC ?	Cancel user-defined characters
\$1B \$45	ESC E	Select emphasized mode
\$1B \$47	ESC G	Select double-strike mode
\$1B \$4D	ESC M	Select character font
\$1B \$52	ESC R	Select international character set
\$1B \$56	ESC V	Select print mode 90° turned
\$1B \$74	ESC t	Select character code table
\$1B \$7B	ESC {	Set/cancel upside-down character printing
\$1B \$C1	ESC { }	Set/cancel cpi mode
\$1C \$65	FS e	Enable/Disable TrueType fonts encoding
\$1C \$66	FS f	TrueType fonts management
\$1D \$21	GS !	Select character size
\$1D \$42	GS B	Turn white/black reverse printing mode on/off
PRINT POSITION COMMANDS		
\$08	BS	Back space
\$09	HT	Horizontal tab
\$1B \$24	ESC \$	Set absolute position

ESC/POS™ Emulation

\$1B \$28 \$76	ESC (v	Set relative vertical print position
\$1B \$44	ESC D	Set horizontal tab position
\$1B \$5C	ESC \	Set relative print position
\$1B \$61	ESC a	Select justification
\$1D \$4C	GS L	Set left margin
\$1D \$57	GS W	Set printing area width
BIT-IMAGE COMMANDS		
\$1B \$2A	ESC *	Select image print mode
\$1D \$2A	GS *	Define downloaded bit image
\$1D \$2F	GS /	Print downloaded bit image
\$1D \$76 \$30	GS v 0	Print raster image
STATUS COMMANDS		
\$10 \$04	DLE EOT	Real-time status transmission
\$1B \$76	ESC v	Transmit printer status
\$1D \$72	GS r	Transmit status
\$1D \$E0	GS { }	Enable / disable automatic FULL STATUS back
\$1D \$E1	GS { }	Reading of length paper (cm) available before virtual paper end
\$1D \$E2	GS { }	Reading number of cuts performed from the printer
\$1D \$E3	GS { }	Reading of length (cm) of printed paper
\$1D \$E5	GS { }	Reading number of power up
BARCODE COMMANDS		
\$1C \$B0	FS { }	Sets barcode reader status
\$1C \$B1	FS { }	Get barcode reader status
\$1C \$B2	FS { }	Barcode reader trigger
\$1D \$28 \$6B	GS (k	Print two-dimensional barcode
\$1D \$48	GS H	Select printing position of HRI characters
\$1D \$66	GS f	Select font for HRI characters
\$1D \$68	GS h	Select barcode height
\$1D \$6B	GS k	Print barcode
\$1D \$77	GS w	Select horizontal size (enlargement) of barcode
MACRO FUNCTION COMMANDS		
\$1D \$3A	GS :	Set start/end of macro definition
\$1D \$5E	GS ^	Execute macro
MECHANISM CONTROL COMMANDS		
\$1B \$69	ESC i	Total cut
\$1B \$69	ESC i	Presentation mode
\$1C \$0C	FS { }	Load paper from triple feeder (1,2,3)
\$1C \$C1	FS { }	Paper recovery after cut
\$1D \$56	GS V	Select cut mode

MISCELLANEOUS COMMANDS		
\$1B \$3D	ESC =	Select device
\$1B \$40	ESC @	Initialize printer
\$1B \$63 \$35	ESC c 5	Enable/Disable keys panel
\$1C \$3C	FS <	Change printer emulation to SVELTA
\$1C \$6C	FS I	Reload paper
\$1C \$80	FS { }	Read date/time of the real time clock
\$1C \$81	FS { }	Set date/time of the real time clock
\$1C \$84	FS { }	Set user-defined date/time formats
\$1C \$90	FS { }	Get number of stored logo
\$1C \$91	FS { }	Get pictures header list
\$1C \$92	FS { }	Get pictures header info
\$1C \$93	FS { }	Print logo
\$1C \$94	FS { }	Save the image received from serial port into the flash
\$1C \$C0	FS { }	Hardware reset
\$1D \$49	GS I	Transmit printer ID
\$1D \$50	GS P	Set horizontal and vertical motion units
\$1D \$E6	GS { }	Virtual paper end limit
TICKET MANAGEMENT COMMANDS		
\$1D \$7C	GS { }	Set printing density
\$1D \$E7	GS { }	Set notch distance
\$1D \$F0	GS { }	Set printing speed
\$1D \$F6	GS { }	Ticket align at print
\$1D \$F8	GS { }	Ticket align at cut
SELECTOR MANAGEMENT COMMANDS		
\$1D \$65 \$35	GS e 5	Perform the ticket ejection
\$1D \$70 \$69	GS p i	Initialize selector
\$1D \$70 \$6F	GS p o	Set selector in "Open" position
\$1D \$70 \$73	GS p s	Set selector in "Storage" position

ESC/POS™ Emulation

Given below are more detailed descriptions of each command.

\$08

Devices: *ALL*

[Name] **Back space**

[Format] ASCII BS
Hex 08
Decimal 8

[Range]

[Description] Moves print position to previous character.

[Notes] Can be used to put two characters at the same position.

[Default]

[Reference]

[Example]

\$09

Devices: *ALL*

[Name] **Horizontal tab**

[Format] ASCII HT
Hex 09
Decimal 9

[Range]

[Description] Moves the print position to the next horizontal tab position.

[Notes]

- Ignored unless the next horizontal tab position has been set.
- If the command is received when the printing position is at the right margin, the printer executes print buffer full printing and horizontal tab processing from the beginning of the next line.
- Horizontal tab position are set using \$1B \$44

[Default]

[Reference] \$1B \$44

[Example]

\$0ADevices: *ALL*[Name] **Print and line feed**

[Format]	ASCII	LF
	Hex	0A
	Decimal	10

[Range]

[Description] Prints the data in the buffer and feeds one line based on the current line spacing.

[Notes]

- Sets the print position to the beginning of the line.
- If the buffer is empty, the printing feeds of (character height + spacing gap) dot.

[Default]

[Reference] \$1B \$32, \$1B \$33, \$0D

[Example]

\$0DDevices: *ALL*[Name] **Print and carriage return**

[Format]	ASCII	CR
	Hex	0D
	Decimal	13

[Range]

[Description] When autofeed is "CR enabled", this command functions in the same way as \$0A, otherwise it is disregarded.

[Notes]

- Sets the print position to the beginning of the line.

[Default] See "Autofeed in setup" parameter.

[Reference] \$0A

[Example]

\$10 \$04

Devices: ALL

[Name] **Real-time status transmission**
 [Format] ASCII DLE EOT n
 Hex 10 04 n
 Decimal 16 4 n

[Range] $1 \leq n \leq 4$; n=17; n=20; n=21

**KPM302 (models with triple feeder),
 TK302 (models with triple feeder)**
 n = 26

[Description] Transmits the selected printer status specified by *n* in real time according to the following parameters:
 n = 1 transmit printer status
 n = 2 transmit off-line status
 n = 3 transmit error status
 n = 4 transmit paper roll sensor status
 n = 17 transmit print status
 n = 20 transmit FULL STATUS
 n = 21 transmit printer ID

**KPM302 (models with triple feeder),
 TK302 (models with triple feeder)**
 n=26 transmit printer + triple feeder FULL STATUS

- [Notes]
- Immediately executed even when the data buffer is full.
 - This status is transmitted whenever data sequence \$10 \$04 n is received.

[Default]
 [Reference]

See tables below.

[Example]

n=1: Printer status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	-	-	-	RESERVED
3	Off	00	0	On-line.
	On	08	8	Off-line.
4	On	10	16	Not used. Fixed to On
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	Off	00	0	LF key released
	On	80	128	LF key pressed

n=2: Off-line status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	Off	00	0	Cover closed
	On	04	4	Cover opened
3	Off	00	0	Paper isn't fed by FEED. key
	On	08	8	Paper is fed by FEED. key
4	On	10	16	Not used. Fixed to On
5	Off	00	0	Paper present
	On	20	32	Printing stop due to paper end
6	Off	00	0	No error
	On	40	64	Error
7	Off	00	0	Not used. Fixed to Off

n=3: Error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	-	-	-	RESERVED
3	Off	00	0	Cutter ok
	On	08	8	Cutter error
4	On	10	16	Not used. Fixed to On
5	Off	00	0	No unrecoverable error.
	On	20	32	Unrecoverable error
6	Off	00	0	No auto-recoverable error
	On	40	64	Auto-recoverable error
7	Off	00	0	Not used. Fixed to Off

n=4: Paper roll sensor status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2,3	Off	00	0	Paper present..
	On	0C	12	Near paper end.
4	On	10	16	Not used. Fixed to On
5, 6	Off	00	0	Paper present
	On	60	96	Paper not present
7	Off	00	0	Not used. Fixed to Off

n=17: Print status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	Off	00	0	Paper drag motor off
	On	04	4	Paper drag motor on
3	-	-	-	RESERVED
4	On	10	16	Not used. Fixed to On
5	Off	00	0	Paper present
	On	20	32	Paper absent
6	-	-	-	RESERVED
7	Off	00	0	Not used. Fixed to Off

n=20: FULL status (6 bytes)

1° Byte = \$10 (DLE)

2° Byte = \$0F

3° Byte = Paper status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper present
	On	01	1	Paper not present
1	-	-	-	RESERVED
2	Off	00	0	Paper present
	On	04	4	Near paper end
3	-	-	-	RESERVED
4	-	-	-	RESERVED
5	Off	00	0	Ticket not present in output
	On	20	32	Ticket present in output
6	Off	00	0	Not virtual paper end (*).
	On	40	64	Virtual paper end (*).
7	Off	00	0	The notch is placed over the sensor
	On	80	128	The notch is not placed over the sensor

(*) Virtual paper end is set when the paper length available, readed by \$1D \$E1, is 0.

4° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	-	-	-	RESERVED

5° byte = Recoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Head temperature ok.
	On	01	1	Head temperature error
1	Off	00	0	No COM error
	On	02	2	RS232 COM error
2	-	-	-	RESERVED
3	Off	00	0	Power supply voltage ok
	On	08	8	Power supply voltage error
4	-	-	-	RESERVED
5	Off	00	0	Acknowledge command
	On	20	32	Not acknowledge command error
6	Off	00	0	Free paper path
	On	40	64	Paper jam
7	Off	00	0	Notch search ok
	On	80	128	Error in notch search

6° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Cutter ok
	On	01	1	Cutter error
1	Off	00	0	Cutter cover ok
	On	02	2	Cutter cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

n=21: transmit printer ID

1° byte = \$75 (refer to command \$1D \$49)

KPM302 (models with selector), KPM303 (models with selector)

n=1: Printer status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	-	-	-	RESERVED
3	Off	00	0	On-line.
	On	08	8	Off-line.
4	On	10	16	Not used. Fixed to On
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	Off	00	0	LF key released
	On	80	128	LF key pressed

n=2: Off-line status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	Off	00	0	Cover closed
	On	04	4	Cover opened
3	Off	00	0	Paper isn't fed by FEED. key
	On	08	8	Paper is fed by FEED. key
4	On	10	16	Not used. Fixed to On
5	Off	00	0	Paper present
	On	20	32	Printing stop due to paper end
6	Off	00	0	No error
	On	40	64	Error
7	Off	00	0	Selector in "open" position
	On	80	128	Selector in "storage" position

n=3: Error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	-	-	-	RESERVED
3	Off	00	0	Cutter ok
	On	08	8	Cutter error
4	On	10	16	Not used. Fixed to On
5	Off	00	0	No unrecoverable error.
	On	20	32	Unrecoverable error
6	Off	00	0	No auto-recoverable error
	On	40	64	Auto-recoverable error
7	Off	00	0	Not used. Fixed to Off

n=4: Paper roll sensor status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2,3	Off	00	0	Paper present..
	On	0C	12	Near paper end.
4	On	10	16	Not used. Fixed to On
5, 6	Off	00	0	Paper present
	On	60	96	Paper not present
7	Off	00	0	Selector Ok
	On	80	128	Selector error

n=17: Print status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	Off	00	0	Paper drag motor off
	On	04	4	Paper drag motor on
3	-	-	-	RESERVED
4	On	10	16	Not used. Fixed to On
5	Off	00	0	Paper present
	On	20	32	Paper absent
6	-	-	-	RESERVED
7	Off	00	0	Not used. Fixed to Off

n=20: FULL status (6 bytes)

1° Byte = \$10 (DLE)

2° Byte = \$0F

3° Byte = Paper status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper present
	On	01	1	Paper not present
1	-	-	-	RESERVED
2	Off	00	0	Paper present
	On	04	4	Near paper end
3	-	-	-	RESERVED
4	-	-	-	RESERVED
5	Off	00	0	Ticket not present in output
	On	20	32	Ticket present in output
6	Off	00	0	Not virtual paper end (*).
	On	40	64	Virtual paper end (*).
7	Off	00	0	The notch is placed over the sensor
	On	80	128	The notch is not placed over the sensor

(*) Virtual paper end is set when the paper length available, read by \$1D \$E1, is 0.

4° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	-	-	-	RESERVED

5° byte = Recoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Head temperature ok.
	On	01	1	Head temperature error
1	Off	00	0	No COM error
	On	02	2	RS232 COM error
2	-	-	-	RESERVED
3	Off	00	0	Power supply voltage ok
	On	08	8	Power supply voltage error
4	-	-	-	RESERVED
5	Off	00	0	Acknowledge command
	On	20	32	Not acknowledge command error
6	Off	00	0	Free paper path
	On	40	64	Paper jam
7	Off	00	0	Notch search ok
	On	80	128	Error in notch search

6° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Cutter ok
	On	01	1	Cutter error
1	Off	00	0	Cutter cover ok
	On	02	2	Cutter cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

n=21: transmit printer ID

1° byte = \$75 (refer to command \$1D \$49)

**KPM302 (models with triple feeder),
TK302 (models with triple feeder)**

n=26 FULL status (14 bytes)

1° Byte = \$10 (DLE)

2° Byte = \$FF

3° Byte = Paper status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper present
	On	01	1	Paper not present
1	-	-	-	RESERVED
2	Off	00	0	Paper present
	On	04	4	Near paper end
3	-	-	-	RESERVED
4	-	-	-	RESERVED
5	Off	00	0	Ticket not present in output
	On	20	32	Ticket present in output
6	Off	00	0	Not virtual paper end (*).
	On	40	64	Virtual paper end (*).
7	Off	00	0	The notch is placed over the sensor
	On	80	128	The notch is not placed over the sensor

(*) Virtual paper end is set when the paper length available, read by \$1D \$E1, is 0.

4° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	-	-	-	RESERVED

5° byte = Recoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Head temperature ok.
	On	01	1	Head temperature error
1	Off	00	0	No COM error
	On	02	2	RS232 COM error
2	-	-	-	RESERVED
3	Off	00	0	Power supply voltage ok
	On	08	8	Power supply voltage error
4	-	-	-	RESERVED
5	Off	00	0	Acknowledge command
	On	20	32	Not acknowledge command error
6	Off	00	0	Free paper path
	On	40	64	Paper jam
7	Off	00	0	Notch search ok
	On	80	128	Error in notch search

6° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	-	-	-	RESERVED
1	Off	00	0	Frontal cover ok
	On	02	2	Frontal cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

7° byte = \$49

8° byte = Feeder status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper sensor (Feeder 1 UP): paper not present
	On	01	1	Paper sensor (Feeder 1 UP): paper present
1	Off	00	0	Paper sensor (Feeder 2 CENTER): paper not present
	On	02	2	Paper sensor (Feeder 2 CENTER): paper present
2	Off	00	0	Paper sensor (Feeder 3 DOWN): paper not present
	On	04	4	Paper sensor (Feeder 3 DOWN): paper present
3	-	-	-	RESERVED
4	Off	00	0	Near paper end sensor (Feeder 1 UP): paper not present
	On	10	16	Near paper end sensor (Feeder 1 UP): paper present
5	Off	00	0	Near paper end sensor (Feeder 2 CENTER): paper not present
	On	20	32	Near paper end sensor (Feeder 2 CENTER): paper present
6	Off	00	0	Near paper end sensor (Feeder 3 DOWN): paper not present
	On	40	64	Near paper end sensor (Feeder 3 DOWN): paper present
7	-	-	-	RESERVED

9° Byte = \$41

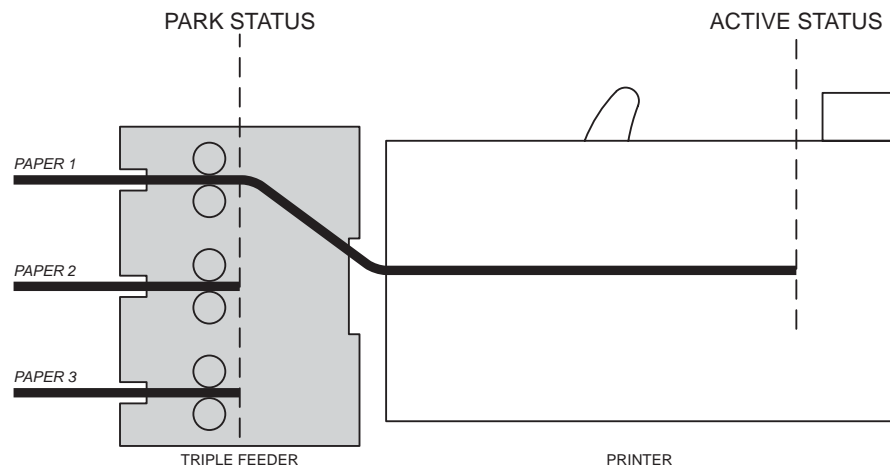
10° Byte = FEEDER 1 (UP) = 0 No paper in feeder 1
 = 1 Paper in ACTIVE STATUS
 = 7 Paper end
 = 9 Error
 = 10 Paper in PARK STATUS

11° Byte = \$42

12° Byte = FEEDER 2 (CENTER) = 0 No paper in feeder 2
 = 1 Paper in ACTIVE STATUS
 = 7 Paper end
 = 9 Error
 = 10 Paper in PARK STATUS

13° Byte = \$43

14° Byte = FEEDER 3 (DOWN) = 0 No paper in feeder 3
 = 1 Paper in ACTIVE STATUS
 = 7 Paper end
 = 9 Error
 = 10 Paper in PARK STATUS



KPM202, KPM203, TK202, TK203

n=1: Printer status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	-	-	-	RESERVED
3	Off	00	0	On-line.
	On	08	8	Off-line.
4	On	10	16	Not used. Fixed to On
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	Off	00	0	LF key released
	On	80	128	LF key pressed

n=2: Off-line status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	Off	00	0	Cover closed
	On	04	4	Cover opened
3	Off	00	0	Paper isn't fed by FEED. key
	On	08	8	Paper is fed by FEED. key
4	On	10	16	Not used. Fixed to On
5	Off	00	0	Paper present
	On	20	32	Printing stop due to paper end
6	Off	00	0	No error
	On	40	64	Error
7	Off	00	0	Not used. Fixed to Off

n=3: Error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	-	-	-	RESERVED
3	-	-	-	RESERVED
4	On	10	16	Not used. Fixed to On
5	Off	00	0	No unrecoverable error.
	On	20	32	Unrecoverable error
6	Off	00	0	No auto-recoverable error
	On	40	64	Auto-recoverable error
7	Off	00	0	Not used. Fixed to Off

n=4: Paper roll sensor status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2,3	Off	00	0	Paper present..
	On	0C	12	Near paper end.
4	On	10	16	Not used. Fixed to On
5, 6	Off	00	0	Paper present
	On	60	96	Paper not present
7	Off	00	0	Not used. Fixed to Off

n=17: Print status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Not used. Fixed to Off
1	On	02	2	Not used. Fixed to On
2	Off	00	0	Paper drag motor off
	On	04	4	Paper drag motor on
3	-	-	-	RESERVED
4	On	10	16	Not used. Fixed to On
5	Off	00	0	Paper present
	On	20	32	Paper absent
6	-	-	-	RESERVED
7	Off	00	0	Not used. Fixed to Off

n=20: FULL status (6 bytes)

1° Byte = \$10 (DLE)

2° Byte = \$0F

3° Byte = Paper status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper present
	On	01	1	Paper not present
1	-	-	-	RESERVED
2	Off	00	0	Paper present
	On	04	4	Near paper end
3	-	-	-	RESERVED
4	-	-	-	RESERVED
5	Off	00	0	Ticket not present in output
	On	20	32	Ticket present in output
6	Off	00	0	Not virtual paper end (*).
	On	40	64	Virtual paper end (*).
7	Off	00	0	Notch found
	On	80	128	Notch not found

(*) Virtual paper end is set when the paper length available, read by \$1D \$E1, is 0.

4° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	-	-	-	RESERVED

5° byte = Recoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Head temperature ok.
	On	01	1	Head temperature error
1	Off	00	0	No COM error
	On	02	2	RS232 COM error
2	-	-	-	RESERVED
3	Off	00	0	Power supply voltage ok
	On	08	8	Power supply voltage error
4	-	-	-	RESERVED
5	Off	00	0	Acknowledge command
	On	20	32	Not acknowledge command error
6	Off	00	0	Free paper path
	On	40	64	Paper jam
7	Off	00	0	Notch search ok
	On	80	128	Error in notch search

6° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	-	-	-	RESERVED
1	Off	00	0	Frontal cover ok
	On	02	2	Frontal cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

n=21: transmit printer ID 1° byte = \$75 (refer to command \$1D \$49)

\$18Devices: *ALL*

[Name]	Cancel current line transmitted		
[Format]	ASCII	CAN	
	Hex	18	
	Decimal	24	
[Description]	Deletes current line transmitted.		
[Notes]	<ul style="list-style-type: none"> • Sets the print position to the beginning of the line. • However, this command does not clear the receive buffer. 		
[Default]			
[Reference]			
[Example]			

\$1B \$20Devices: *ALL*

[Name]	Set right-side character spacing			
[Format]	ASCII	ESC	SP	n
	Hex	1B	20	n
	Decimal	27	32	n
[Range]	0 ≤ n ≤ 255			
[Description]	Sets the character spacing for the right side of the character to [n x horizontal or vertical motion units].			
[Notes]	<ul style="list-style-type: none"> • The right character spacing for double-width mode is twice the normal value. When the characters are enlarged, the right side character spacing is m (2 or 4) times the normal value. • The horizontal and vertical motion units are specified by \$1D \$50. Changing the horizontal or vertical motion units does not affect the current right side spacing. • The \$1D \$50 command can change the horizontal (and vertical) motion unit. However, the value cannot be less than the minimum horizontal movement amount. • In standard mode, the horizontal motion unit is used. • The maximum right side spacing is 32mm. 			
[Default]	n = 0			
[Reference]	\$1D \$50			
[Example]				

\$1B \$21

Devices: ALL

[Name] **Select print modes**
 [Format] ASCII ESC ! n
 Hex 1B 21 n
 Decimal 27 33 n
 [Range] $0 \leq n \leq 255$
 [Description] Selects print modes using n (see table below):

BIT	OFF/ON	HEX	Decimal	FUNCTION	11/15 cpi	15/20 cpi
0	Off	00	0	Character font A selected	18 x 24	14 x 24
	On	01	1	Character font B selected	14 x 24	10 x 24
1	-	-	-	Undefined		
2	-	-	-	Undefined		
3	Off	00	0	Expanded mode not selected		
	On	08	8	Expanded mode selected		
4	Off	00	0	Double-height mode not selected		
	On	10	16	Double-height mode selected		
5	Off	00	0	Double-width mode not selected		
	On	20	32	Double-width mode selected		
6	Off	00	0	Italic mode not selected		
	On	40	64	Italic mode selected		
7	Off	00	0	Underlined mode not selected		
	On	80	128	Underlined mode selected		

[Notes]

- The printer can underline all characters, but cannot underline the spaces set by \$09, \$1B \$24, \$1B \$5C and 90°/270° rotated characters.
- This command resets the left and right margin at default value (see \$1D \$4C, \$1D \$57).
- \$1B \$45 can also be used to turn the emphasized mode on/off. However, the last-received setting command is the effective one.
- \$1B \$2D can also be used to turn the underlining mode on/off. However, the last-received setting command is the effective one.
- \$1D \$21 can also be used to select character height/width. However, the last-received setting command is the effective one.

[Default] $n = 0$
 [Reference] \$1B \$2D, \$1B \$45, \$1D \$21
 [Example]

\$1B \$24

Devices:	ALL				
[Name]	Set absolute print position				
[Format]	ASCII	ESC	\$	nL	nH
	Hex	1B	24	nL	nH
	Decimal	27	36	nL	nH
[Range]	0 ≤ nL ≤ 255 0 ≤ nH ≤ 255				
[Description]	Sets the distance from the beginning of the line to the position at which subsequent characters are to be printed. The distance from the beginning of the line to the print position is [(nL + nH × 256) × (vertical or horizontal motion unit)] inches.				
[Notes]	<ul style="list-style-type: none"> • Settings outside the specified printable area are ignored. • The horizontal and vertical motion unit are specified by \$1D \$50. • \$1D \$50 can change the horizontal (and vertical) motion unit. However, the value cannot be less than the minimum horizontal movement amount. • In standard mode, the horizontal motion unit (x) is used. • If the setting is outside the printing area width, it sets the absolute print position, but the left or right margin is set at default value. 				
[Default]					
[Reference]	\$1B \$5C, \$1D \$50				
[Example]					

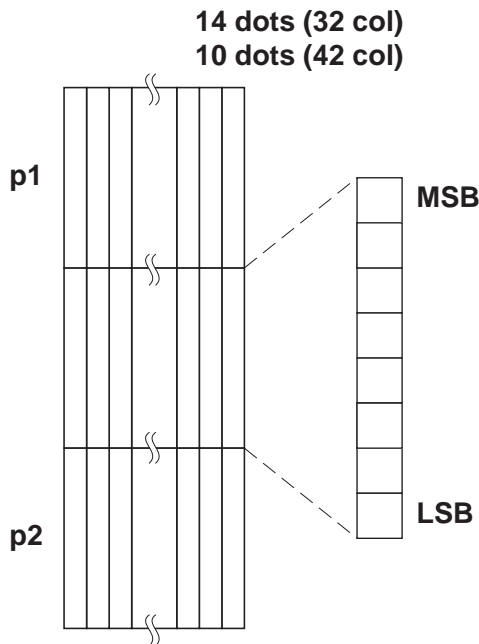
\$1B \$25

Devices:	ALL				
[Name]	Select/cancel user-defined characters				
[Format]	ASCII	ESC	%	n	
	Hex	1B	25	n	
	Decimal	27	37	n	
[Range]	0 ≤ n ≤ 255				
[Description]	Selects or cancels the user-defined character set. When the Least Significant Bit (LSB) of n is 0, the user-defined character set is cancelled. When the LSB of n is 1, the user-defined character set is selected.				
[Notes]	<ul style="list-style-type: none"> • Only the LSB of n is applicable. • When the user-defined character set is cancelled, the internal character set is automatically selected. 				
[Default]	n=0				
[Reference]	\$1B \$26, \$1B \$3F				
[Example]					

\$1B \$26

Devices: *ALL*

[Name]	Defines user-defined characters
[Format]	ASCII ESC & y c1 c2 Hex 1B 26 y c1 c2 Decimal 27 37 y c1 c2
[Range]	y = 3 32 ≤ c1 ≤ c2 ≤ 126 0 ≤ x ≤ 16 (Font (18 × 24)) 0 ≤ x ≤ 13 (Font (14 × 24)) 0 ≤ x ≤ 10 (Font 10 × 24) 0 ≤ d1 ... d (y × xk) ≤ 255 k = c2 – c1 + 1
[Description]	Defines user-defined characters. • Y specifies the number of bytes in the vertical direction. • C1 specifies the beginning character code for the definition, and C2 specifies the final code. • X specifies the number of dots in the horizontal direction.
[Notes]	• The allowable character code range is from ASCII \$20 (32) to \$7E (126) (95 characters). • It is possible to define multiple characters for consecutive character codes. If only one character is desired, use c1 = c2. • If c2 < c1, the command is not executed. • d is the dot data for the characters. The dot pattern is in the horizontal direction starting from the left. Any remaining dots on the right remain blank. • The data to define a user-defined character is (x × y) bytes. • To print a dot, set the corresponding bit to 1; to not have it print, set to 0. • This command can define different user-defined character patterns for each font. To select the font, use \$1B \$21. • The user-defined character definitions are cleared when: \$1B \$40 or \$1D \$2A or \$1B \$3F are executed or the printer is reset or the power shut off.
[Default]	Internal character set.
[Reference]	\$1B \$25, \$1B \$3F
[Example]	



\$1B \$28 \$76

Devices: ALL

[Name] Set relative vertical print position

[Format]	ASCII	ESC	(v	nL	nH
	Hex	1B	28	76	nL	nH
	Decimal	27	10	118	nL	nH

[Range] $0 \leq nL \leq 255$
 $0 \leq nH \leq 255$

[Description] Sets the print vertical position based on the current position by using the horizontal or vertical motion unit. This command sets the distance from the current position to $[(nL + nH \times 256) \times$ (horizontal or vertical motion unit)].

[Notes]

- When the starting position is specified by N motion unit to the bottom: $nL + nH \times 256 = N$
- When the starting position is specified by N motion unit to the top (negative direction), use the complement of 65536: $nL + nH \times 256 = 65536 - N$
- The horizontal and vertical motion unit are specified by \$1D \$50.
- The \$1D \$50 command can change the horizontal (and vertical) motion unit. However, the value cannot be less than the minimum horizontal movement amount.
- In standard mode, the vertical motion unit is used.

[Default]**[Reference]** \$1D \$50**[Example]**

\$1B \$2A

Devices: ALL

[Name] **Select bit image mode**

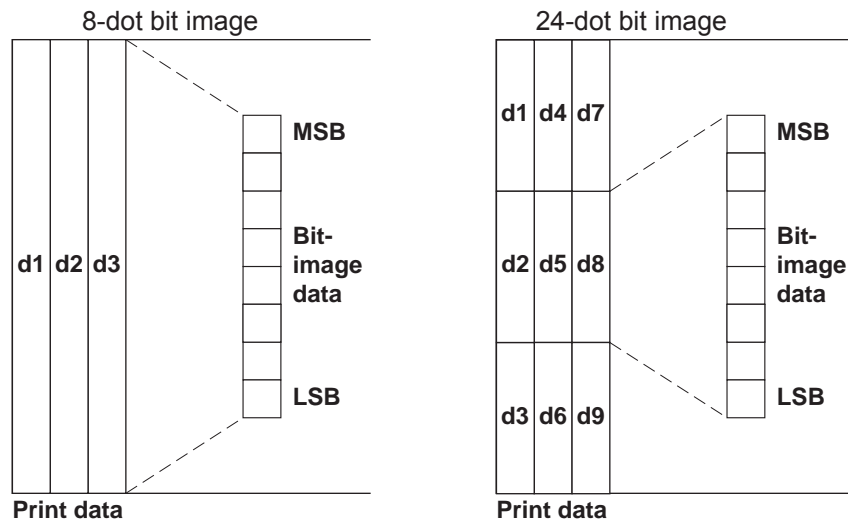
[Format] ASCII ESC * m nL nH d1...dk
 Hex 1B 2A m nL nH d1...dk
 Decimal 27 42 m nL nH d1...dk

[Range] m = 0, 1, 32, 33
 0 ≤ nL ≤ 255
 0 ≤ nH ≤ 3
 0 ≤ d ≤ 255

[Description] Selects a bit image mode using m for the number of dots specified by nL and nH, as follows:

m	MODE	VERTICAL DIRECTION		HORIZONTAL DIRECTION (*1)	
		N° dots	DPI	DPI	N° of data (k)
0	8 dot single density	8	67	100	nL + nH × 256
1	8 dot double density	8	67	200	nL + nH × 256
32	24 dot single density	24	200	100	(nL + nH × 256) × 3
33	24 dot double density	24	200	200	(nL + nH × 256) × 3

- [Notes]
- The nL and nH commands indicate the number of dots of the bit image in the horizontal direction. The number of dots is calculated using: nL + nH × 256.
 - If the bit image data input exceeds the number of dots to be printed on a line, the excess data is ignored.
 - d indicates the bit image data. Set a corresponding bit to 1 to print a dot, or to 0 to not print the dot.
 - If the value of m is outside the specified range, nL and data following it are processed as normal data.
 - If the width of the printing area set by \$1D \$4C and \$1D \$57 is less than the width required by the data set using \$1B \$2A, the excess data are ignored.
 - To print the bit image use \$1B \$4A or \$1B \$64.
 - After printing a bit image, the printer returns to normal data processing mode.
 - This command is not affected by the emphasized, double-strike, underline (etc.) print modes, except for the upside-down mode.
 - The relationship between the image data and the dots to be printed is as follows:



[Default]
 [Reference]
 [Example]

\$1B \$2D

Devices: ALL

[Name] Turn underline mode on/off

[Format] ASCII ESC - n
 Hex 1B 2D n
 Decimal 27 45 n

[Range] $0 \leq n \leq 2$
 $48 \leq n \leq 50$

[Description] Turns underline mode on or off, based on the following values of *n*:

- n* = 0, 48 Turns off underline mode
- n* = 1, 49 Turns on underline mode (1-dot thick)
- n* = 2, 50 Turns on underline mode (2-dot thick)

[Notes]

- The printer can underline all characters, but cannot underline the space and right-side character spacing.
- The printer cannot underline 90°/270° rotated characters and white/black inverted characters.
- When underline mode is turned off by setting the value of *n* to 0 or 48, the data which follows is not underlined.
- Underline mode can also be turned on or off by using \$1B \$21. Note, however, that the last received command is the effective one.

[Default] n=0
 [Reference] \$1B \$21
 [Example]

\$1B \$30

Devices: *ALL*

[Name]	Select 1/8-inch line spacing
[Format]	ASCII ESC 0 Hex 1B 30 Decimal 27 48
[Range]	
[Description]	Selects 1/8-inch line spacing
[Notes]	
[Default]	
[Reference]	\$1B \$32, \$1B \$33
[Example]	

\$1B \$32

Devices: *ALL*

[Name]	Select 1/6-inch line spacing
[Format]	ASCII ESC 2 Hex 1B 32 Decimal 27 50
[Range]	
[Description]	Selects 1/6-inch line spacing.
[Notes]	
[Default]	
[Reference]	\$1B \$30, \$1B \$33
[Example]	

\$1B \$33Devices: *ALL***[Name] Set line spacing**

[Format]

ASCII	ESC	3	n
Hex	1B	33	n
Decimal	27	51	n

[Range] $0 \leq n \leq 255$ **[Description]** Sets line spacing to [$n \times$ (vertical or horizontal motion unit)] inches.

[Notes]

- The horizontal and vertical motion unit are specified by \$1D \$50. Changing the horizontal or vertical motion unit does not affect the current line spacing.
- The \$1D \$50 command can change the horizontal (and vertical) motion unit. However, the value cannot be less than the minimum vertical movement amount.
- In standard mode, the vertical motion unit is used.
- The maximum spacing is 32.5mm.

[Default] $n = 64$ (1/6 inch)**[Reference]** \$1B \$30, \$1B \$32, \$1D \$50**[Example]****\$1B \$3D**Devices: *ALL***[Name] Select peripheral device**

[Format]

ASCII	ESC	=	n
Hex	1B	3D	n
Decimal	27	61	n

[Range] $1 \leq n \leq 3, n = 5$ **[Description]** Select the device to which the host computer sends data, using n as follows:

n	Function
1, 3	Printer enabled
2	Printer disabled
5, '5'	Select Pass-Through toward RFID module

[Notes]

- When the printer is disabled, it ignores all transmitted data until the printer is enabled through this command.
- When the Pass-through function is enabled, all transmitted data are sent on the 2nd serial.
- When the Pass-through function is enabled toward RFID module, to reactivate communication toward printer must send the \$1B \$3D \$31 \$F1 \$5A \$E0 command.

[Default] $n = 1$ **[Reference]****[Example]**

\$1B \$3F													
Devices:	ALL												
[Name]	Cancel user-defined characters												
[Format]	<table border="0"> <tr> <td>ASCII</td> <td>ESC</td> <td>?</td> <td>n</td> </tr> <tr> <td>Hex</td> <td>1B</td> <td>3F</td> <td>n</td> </tr> <tr> <td>Decimal</td> <td>27</td> <td>63</td> <td>n</td> </tr> </table>	ASCII	ESC	?	n	Hex	1B	3F	n	Decimal	27	63	n
ASCII	ESC	?	n										
Hex	1B	3F	n										
Decimal	27	63	n										
[Range]	$32 \leq n \leq 126$												
[Description]	Cancels user-defined characters.												
[Notes]	<ul style="list-style-type: none"> • This command cancels the pattern defined for the character code specified by n. • This command deletes the pattern defined for the specified character code in the font selected by \$1B \$21. • If the user-defined character has not been defined for the specified character code, the printer ignores this command. 												
[Default]													
[Reference]	\$1B \$26, \$1B \$25												
[Example]													

\$1B \$40										
Devices:	ALL									
[Name]	Initialize printer									
[Format]	<table border="0"> <tr> <td>ASCII</td> <td>ESC</td> <td>@</td> </tr> <tr> <td>Hex</td> <td>1B</td> <td>40</td> </tr> <tr> <td>Decimal</td> <td>27</td> <td>64</td> </tr> </table>	ASCII	ESC	@	Hex	1B	40	Decimal	27	64
ASCII	ESC	@								
Hex	1B	40								
Decimal	27	64								
[Range]										
[Description]	Clears the data in the print buffer and resets the printer mode to that in effect when power was turned on.									
[Notes]	<ul style="list-style-type: none"> • The data in the receiver buffer is not cleared. • The macro definitions are not cleared. 									
[Default]										
[Reference]										
[Example]										

\$1B \$44Devices: *ALL*[Name] **Set horizontal tab positions**

[Format]	ASCII	ESC	D	n1...nk	NUL
	Hex	1B	44	n1...nk	\$00
	Decimal	27	68	n1...nk	0

[Range] $1 \leq n \leq 255$
 $0 \leq k \leq 32$

[Description] Sets horizontal tab positions

- *n* specifies the column number for setting a horizontal tab position calculated from the beginning of the line.

- *k* indicates the total number of horizontal tab positions to be set.

[Notes]

- The horizontal tab position is stored as a value of [character width x *n*] measured from the beginning of the line. The character width includes the right-side character spacing and double-width characters are set with twice the width of normal characters.

- This command cancels previous tab settings.

- When setting *n* = 8, the print position is moved to column 9.

- Up to 32 tab positions (*k* = 32) can be set. Data exceeding 32 tab positions is processed as normal data.

- Send [*n*] *k* in ascending order and place a 0 NUL code at the end. When [*n*] *k* is less than or equal to the preceding value [*n*] *k* -1, the setting is complete and the data which follows is processed as normal data.

- \$1B \$44 \$00 cancels all horizontal tab positions.

- The previously specified horizontal tab position does not change, even if the character width is modified.

[Default] Default tab positions are set at intervals of 8 characters (columns 9, 17, 25, ...) for Font A when the right-side character spacing is 0.

[Reference] \$09

[Example]

\$1B \$45Devices: *ALL*[Name] **Turn emphasized mode on/off**

[Format]	ASCII	ESC	E	n
	Hex	1B	45	n
	Decimal	27	69	n

[Range] $0 \leq n \leq 255$

[Description] Turns emphasized mode on/off.

- When the LSB of *n* is 0, the emphasized mode is off.

- When the LSB of *n* is 1, the emphasized mode is on.

[Notes]

- Only the LSB of *n* is effective.

- \$1B \$21 also turns on and off the emphasized mode. However, the last received command is the effective one.

[Default] *n* = 0

[Reference] \$1B \$21

[Example]

\$1B \$47

Devices: *ALL*

[Name] Turn double-strike mode on/off
[Format] ASCII ESC G n
 Hex 1B 47 n
 Decimal 27 71 n
[Range] $0 \leq n \leq 255$
[Description] Turns double-strike mode on or off.
 • When the LSB of *n* is 0, the double-strike mode is off.
 • When the LSB of *n* is 1, the double-strike mode is on.
[Notes] • Only the LSB of *n* is effective.
 • Printer output is the same in double-strike and emphasized mode.
[Default] *n* = 0
[Reference] \$1B \$45
[Example]

\$1B \$4A

Devices: *ALL*

[Name] Print and paper feed
[Format] ASCII ESC J n
 Hex 1B 4A n
 Decimal 27 74 n
[Range] $0 \leq n \leq 255$
[Description] Prints the data in the print buffer and feeds the paper [*n* × (vertical or horizontal motion unit)] inches.
[Notes] • After printing has been completed, this command sets the print starting position to the beginning of the line.
 • The paper feed amount set by this command does not affect the values set by \$1B \$32 or \$1B \$33.
 • The horizontal and vertical motion units are specified by \$1D \$50.
 • \$1D \$50 can change the vertical (and horizontal) motion unit. However, the value cannot be less than the minimum vertical movement amount.
 • In standard mode, the vertical motion unit is used.
 • The maximum paper feed amount is 520 mm.
[Default]
[Reference] \$1D \$50
[Example]

\$1B \$4D

Devices: ALL

[Name] **Select character font**

[Format] ASCII ESC M n
 Hex 1B 4D n
 Decimal 27 77 n

[Range] n = 0, 1, 48, 49

[Description] Selects characters font depending of cpi value set (Char/Inch) as follows :

CHAR/INCH.	n	FUNCTION
A=11 cpi	0, 48	Font 11 cpi (18 × 24)
B=15 cpi	1, 49	Font 15 cpi (14 × 24)
A=15 cpi	0, 48	Font 15 cpi (14 × 24)
B=20 cpi	1, 49	Font 20 cpi (10 × 24)

[Notes]

[Default]

[Reference] \$1B \$C1

[Example]

\$1B \$52

Devices: ALL

[Name] **Select an international character set**

[Format] ASCII ESC R n
 Hex 1B 52 n
 Decimal 27 82 n

[Range] 0 ≤ n ≤ 10

[Description] Selects the international character set n according to the table below:

	HEX	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
n	CHARACTER SER												
0	U.S.A.	#	\$	@	[\]	^	`	{		}	~
1	France	#	\$	à	°	ç	§	^	`	é	ù	è	“
2	Germany	#	\$	§	Ä	Ö	Ü	^	`	ä	ö	ü	b
3	United Kingdom	£	\$	@	[\]	^	`	{		}	~
4	Denmark I	#	\$	@	Æ	Æ	Å	^	`	æ	f	å	~
5	Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
6	Italy	#	\$	@	°	\	é	^	ù	à	ò	è	ì
7	Spain I	Pt	\$	@	i	Ñ	¿	^	`	“	ñ	}	~
8	Japan	#	\$	@	[¥]	^	`	{		}	~
9	Norway	#	¤	É	Æ	Æ	Å	Ü	é	æ	f	å	ü
10	Denmark II	#	\$	É	Æ	Æ	Å	Ü	é	æ	f	å	ü

[Default] n = 0

[Reference]

[Example]

\$1B \$56

Devices: ALL

[Name] **Set 90° rotated print mode**

[Format] ASCII ESC V n
Hex 1B 56 n
Decimal 27 86 n

[Range] $0 \leq n \leq 1, 48 \leq n \leq 49$

[Description] Turns 90° rotation mode on/off. n is used as follows :

n	FUNCTION
0, 48	Turns off 90° rotation mode
1, 49	Turns on 90° rotation mode

[Notes]

- When underlined mode is turned on, the printer does not underline 90° rotated characters. All the same it's possible select the underline mode.
- Double-width and double-height commands in 90° rotation mode enlarge characters in the opposite directions from double-height and double-width commands in normal mode.
- This command is not available in Page mode.
- If this command is entered in Page mode, the printer all the same save the setting.

Default] n = 0

[Reference] \$1B \$21, \$1B \$2D

[Example]

\$1B \$5C

Devices:	ALL				
[Name]	Set relative print position				
[Format]	ASCII	ESC	\	nL	nH
	Hex	1B	5C	nL	nH
	Decimal	27	92	nL	nH
[Range]	0 ≤ nL ≤ 255 0 ≤ nH ≤ 255				
[Description]	Sets the print starting position based on the current position by using the horizontal or vertical motion unit. Sets the distance from the current position to [(nL+ nH × 256) × (horizontal or vertical motion unit)].				
[Notes]	<ul style="list-style-type: none"> • It's possible to print further on the right margin set for every font. In this case the printing continues up to the maximum border of the printer mechanism and then begins a new row. • Any setting that exceeds the printable area is ignored. • When the starting position is specified by n motion units to the right: $nL + nH \times 256 = n$ • When the starting position is specified by n motion units to the left (negative direction), use the complement of 65536: $nL + nH \times 256 = 65536 - n$ • If setting exceeds the printing area width, the left or right margin is set to the default value. • The horizontal and vertical motion unit are specified by \$1D \$50. • \$1D \$50 can change the horizontal (and vertical) motion units. However, the value cannot be less than the minimum horizontal movement amount. • In standard mode, the horizontal motion unit is used. • Setting the right value, it's possible to print characters over the right edge. 				
[Default]					
[Reference]	\$1B \$24, \$1D \$50				
[Example]					

\$1B \$61

Devices: *ALL*

[Name] **Select justification**
 [Format] ASCII ESC a n
 Hex 1B 61 n
 Decimal 27 97 n
 [Range] $0 \leq n \leq 2$
 $48 \leq n \leq 50$
 [Description] Aligns all data in one line to the specified position. *n* selects the type of justification as follows:

n	JUSTIFICATION
0, 48	Flush left
1, 49	Centred
2, 50	Flush right

[Notes] • This command is only enabled when inserted at the beginning of a line.
 • Lines are justified within the specified printing area.
 • Spaces set by \$09, \$1B \$24 and \$1B \$5C will be justified according to the previously-entered mode.

[Default] n = 0

[Reference]

[Example]

Flush left	Centered	Flush right
ABC ABCD ABCDE	ABC ABCD ABCDE	ABC ABCD ABCDE

\$1B \$63 \$35

Devices: *ALL*

[Name] **Enable/Disable keys panel**
 [Format] ASCII ESC c 5 n
 Hex 1B 63 35 n
 Decimal 27 99 53 n
 [Range] $0 \leq n \leq 255$
 [Description] Enables / disables the keys panel.
 • When the LSB of *n* is 0, the keys panel is enabled.
 • When the LSB of *n* is 1, the keys panel is disabled.
 [Notes] • Only the LSB of *n* is effective.
 • When the keys panel is disabled, the keys may only be used after the printer has been reset.
 [Default] n = 0
 [Reference]
 [Example]

\$1B \$64Devices: *ALL*[Name] **Print and feed paper *n* rows**

[Format]	ASCII	ESC	d	n
	Hex	1B	64	n
	Decimal	27	100	n

[Range] $0 \leq n \leq 255$ [Description] Prints the data in the print buffer and feeds the paper *n* rows.

[Notes]

- *n* rows paper feed is equivalent to (*n* × char height + line spacing set).
- Sets the print starting position at the beginning of the line.
- This command does not affect the line spacing set by \$1B \$32 or \$1B \$33.
- The maximum paper feed amount is 254 rows. Even if a paper feed amount of more than 254 rows is set, the printer feeds the paper only 254 rows.

[Default]

[Reference] \$1B \$32, \$1B \$33

[Example]

\$1B \$69

Devices: KPM302

KPM303

TK302

TK303

[Name] **Total cut**

[Format]	ASCII	ESC	i
	Hex	1B	69
	Decimal	27	105

[Range]

[Description] This command prints the data in the buffer and enables cutter operation. If there is no cutter, a disabling flag is set and any subsequent cut commands will be ignored.

[Notes]

- The printer waits to complete all paper movement commands before it executes a total cut.

[Default]

[Reference]

[Example]

\$1B \$69

Devices:	KPM202
	KPM203
	TK202
	TK203

[Name]	Presentation mode		
[Format]	ASCII	ESC	i
	Hex	1B	69
	Decimal	27	105

[Range]

[Description] This command activates the presentation mode of the ticket for the manual tear.

[Notes]

[Default]

[Reference]

[Example]

\$1B \$74

Devices:	ALL
----------	-----

[Name]	Select character code table			
[Format]	ASCII	ESC	t	n
	Hex	1B	74	n
	Decimal	27	116	n

[Range] n = 0, 2, 3, 4, 5, 16, 17, 18, 19, 255

[Description] Select a page n from the character code table as follows:

n	PAGE
0	0 (PC437 [U.S.A., Standard Europe])
2	2 (PC850 [Multilingual])
3	3 (PC860 [Portuguese])
4	4 (PC863 [Canadian-French])
5	5 (PC865 [Nordic])
16	16 (WPC1252)
17	17 (PC866 [Cyrillic #2])
18	18 (PC852 [Latin 2])
19	19 (PC858 for Euro symbol at position 213)
255	Space page

[Notes] WPC1252, PC866 and PC852 tables are valid only for TrueType fonts.

[Default] n = 0

[Reference] See character code tables, \$1C \$65, \$1C, \$66

[Example] For printing Euro symbol (€), the command sequence is: \$1B, \$74, \$13, \$D5

\$1B \$76Devices: *ALL*[Name] **Transmit paper sensor status**

[Format] ASCII ESC v
 Hex 1B 76
 Decimal 27 118

[Range]

[Description] When this command is received, transmit the current status of the paper sensor.

[Notes] This command is executed immediately, even when the data buffer is full (Busy). The status to be transmitted is shown in the table below:

BIT	OFF/ON	HEX	Decimal	FUNCTION
0,1	Off	00	0	Near paper-end sensor: paper present
	On	03	3	Near paper-end sensor: paper not present
2,3	Off	00	0	Paper-end sensor: paper present
	On	(0C)	(12)	Paper-end sensor: paper not present
4	Off	00	0	Not used. Fixed to Off
5	-	-	-	Undefined
6	-	-	-	Undefined
7	Off	00	0	Not used. Fixed to Off

[Default]

[Reference]

[Example]

\$1B \$7B

Devices: ALL

[Name] **Turn upside-down printing mode on/off**

[Format] ASCII ESC { n
 Hex 1B 7B n
 Decimal 27 123 n

[Range] $0 \leq n \leq 255$

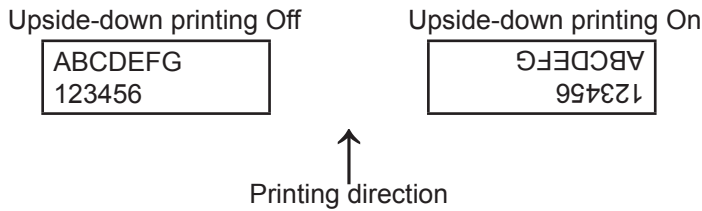
[Description] Turns upside-down printing mode on or off.
 • When the LSB of *n* is 0, the upside-down printing mode is off.
 • When the LSB of *n* is 1, the upside-down printing mode is on.
 • Only the LSB of *n* is effective.
 • This command is valid only if entered at the beginning of a line.
 • In upside-down printing mode, the printer rotates the line to be printed 180° and then prints it.
 n = 0

[Notes]

[Default]

[Reference]

[Example]



\$1B \$C1

Devices: ALL

[Name] **Set/cancel cpi mode**

[Format] ASCII ESC { } n
 Hex 1B C1 n
 Decimal 27 193 n

[Range] $0 \leq n \leq 1, 48 \leq n \leq 49$

[Description] Sets cpi mode based on the following values of n:

n	FUNCTION
0, 48	Font A = 11 cpi Font B = 15 cpi
1, 49	Font A = 15 cpi Font B = 20 cpi

[Default] n = 0

[Reference] \$1B \$21

[Example]

\$1C \$0C

Devices:	KPM302	(models with triple feeder)
	TK302	(models with triple feeder)

[Name] **Load paper from triple feeder (1, 2, 3)**

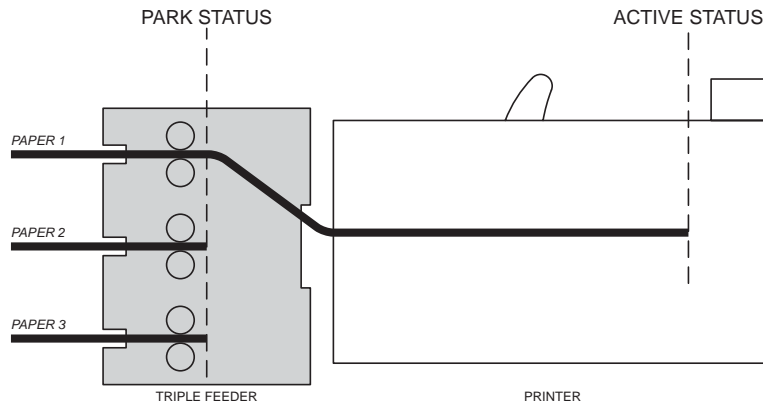
[Format]	ASCII	FS	{ }	n
	Hex	1C	0C	n
	Decimal	28	12	n

[Range] 65 ≤ n ≤ 67

[Description] Load paper inside the printer based on the following values of n:

- n= 65 paper in feeder 1
- n= 66 paper in feeder 2
- n= 67 paper in feeder 3

[Notes] • If another paper is in ACTIVE STATUS is retracted to PARK STATUS.



[Default]
 [Reference]
 [Example]

\$1C \$3C

Devices: *ALL*

[Name] **Change printer emulation to SVELTA**
 [Format] ASCII FS < S V E L >
 Hex 1C 3C 53 56 45 4C 3E
 Decimal 28 60 83 86 69 76 62

[Range]
 [Description] Change the printer emulation to SVELTA emulation.
 [Notes]
 [Default]
 [Reference]
 [Example]

\$1C \$65

Devices: *ALL*

[Name] **Enable/Disable encoding**
 [Format] ASCII FS e n
 Hex 1C 65 n
 Decimal 28 101 n

[Range] n = '0', '1', '2', 48, 49, 50
 [Description] Enable/Disable the text encoding based on the following values of n:

n	ENCODING
0, 48	Disabled
1, 49	Enable UTF-8
2, 50	Enable UTF-16

[Notes]

- This command is valid only for TrueType fonts of monospace type.
- If the text encoding is disabled, manage the characters coding by \$1B \$52 and \$1B \$74 commands.
- If the text encoding is enabled, the character's addressing respects the UNICODE™ standard (see www.unicode.org).

[Default] Disabled.
 [Reference] \$1B \$52, \$1B \$74, \$1C \$66
 [Example]

\$1C \$66

Devices: ALL

[Name] **True Type font management**

[Format] ASCII FS f m n d[0]...d[n]
 Hex 1C 66 m n d[0]...d[n]
 Decimal 28 102 m n d[0]...d[n]

[Range] $0 \leq m \leq 256$
 $0 \leq n \leq 64$

[Description] Manage the TrueType fonts depending on the following values of m

m (BIT)	FUNCTION
0	Check glyph width
1	TTF enable hinting
2	Not used
3	Not used
4	Re-enable TrueType font
5	Disable TrueType font
6	De-init TrueType font
7	Clear all

n specifies the name length of the font to use.

d[0]...d[n] specifies the font name to use.

[Notes]

- If "Check glyph width" is selected, for every character, printer checks if the glyph width is different from default width. In this case, the font will be not installed. The check may require some time (it depends on the characters number of the font).
- For "Hinting" means the font adaptation to the grid. When hinting is enabled, the characters are more legible but some characters may be too high (for example, the accented capital letters). This bit is active only when you install a new font.
- "Re-enable" function re-enables a TrueType font previously disabled.
- "Disable" function disables a TrueType font.
- "De-init" function uninstalls a font and clears the memory used by the font. Use this function only when you intend to use the font more, otherwise use the "Disable" function to speed up operations.
- "Clear all" function uninstalls all the installed fonts.
- If command is successful the printer transmits the ACK (\$06), otherwise return NACK (\$015).
- After "Disable", "Re-enable" and "Clear-all" functions, do not pass the filename of the TrueType font.

[Default]

[Reference]

[Example]

- Select the TrueType font with dimensions check, without hinting:
 \$1C \$66 \$02 \$0C "veramono.ttf"
- Return to use the embedded fonts:
 \$1C \$66 \$20 \$00
- Select the font previously disabled:
 \$1C \$66 \$10 \$00
- Uninstall a TrueType font:
 \$1C \$66 \$40 \$0C

\$1C \$6C

Devices: *ALL*

[Name] **Reload paper**

[Format] ASCII FS I
Hex 1C 6C
Decimal 28 108

[Range]

[Description] When this command is received, the printer performs a paper reloading.

[Notes] During the execution of the command, the printer indicates the paper end

[Default]

[Reference]

[Example]

\$1C \$80

Devices: ALL

[Name] **Read date/time of the real time clock**

[Format] ASCII FS { } m
 Hex 1C 80 m
 Decimal 28 128 m

[Range] $0 \leq m \leq 3$

[Description] Read date/time of the real time clock in the format specified by m values as follows:

m	FORMAT
0	DD/MM/YY hh:mm:ss
1	DDMMYYhhmmss
2	YYMMDDhhmmss
3	YYMMDDkkmmssd

where:

DD = represents the day of the date
 MM = represents the month of the date
 YY = represents the year of the date
 hh = represents the hour of the time
 mm = represents the monutes of the time
 ss = represents the seconds of the time
 d = represents the day of the wee

[Notes] • Before send the date/time, if the m parameter is valid the printer transmits the ACK (\$06) followed by the number of bytes to sent, otherwise return NACK (\$015).

[Default]

[Reference]

[Example]

To read date/time in the “DDYYMMhhmmss” format, transmit:

Host

HEX	\$1C	\$80	\$01
ASCII	FS	{ }	m

For example if the current date/time are “15 September 2006 at 10:56:20 (AM)” the printer’s answer is as follows:

HEX	\$06	\$0C	\$31	\$35	\$30	\$39	\$30	\$36	\$31	\$30	\$35	\$36	\$32	\$30
ASCII	ACK	FF	1	5	0	9	0	6	1	0	5	6	2	0

\$1C \$81

Devices: ALL

[Name] **Read date/time of the real time clock**
 [Format] ASCII FS { } m n d0...dn
 Hex 1C 81 m n d0...dn
 Decimal 28 129 m n d0...dn
 [Range] 0 ≤ m ≤ 3
 0 ≤ d0, dn ≤ 255
 [Description] Read date/time of the real time clock in the format specified by m values as follows:

m	FORMAT
0	DD/MM/YY hh:mm:ss
1	DDMMYYhhmmss
2	YYMMDDhhmmss
3	YYMMDDkmmssd

where:

- DD = represents the day of the date
- MM = represents the month of the date
- YY = represents the year of the date
- hh = represents the hour of the time
- mm = represents the minutes of the time
- ss = represents the seconds of the time
- d = represents the day of the week

- n specifies the number of characters to send.
- d0...dn are the ASCII characters relative to the date and time to set.
- If the transmission has been received correctly and the command is valid, the printer returns the ACK (\$06), otherwise return NACK (\$015).
- the day of the week is calculated automatically from the printer and then it's possible that the returned value is different from the one transmitted.

[Notes]

[Default]
 [Reference]
 [Example]

For example to set the date and time to "29 September 2006 at 13:51:00 (PM)" in the "YYM-MDDhhmmss" format, transmit:

Host

Hex	\$1C	\$81	\$02	\$0C	\$30	\$36	\$30	\$39	\$32	\$39	\$31	\$33	\$35	\$31	\$30	\$30
ASCII	FS	{ }	STX	FF	0	6	0	9	2	9	1	3	5	1	0	0

The printer's answer ACK (\$06) if the transmission is OK otherwise NACK (\$15).

\$1C \$82Devices: *ALL*[Name] **Print date**

[Format]	ASCII	FS	{ }
	Hex	1C	82
	Decimal	28	130

[Range]

[Description] Prints date in the format specified by the command \$1C \$84 with the parameter n='D'.

[Notes]

[Default] "dd/mm/yy"

[Reference] \$1C \$83, \$1C \$84

[Example]

\$1C \$83Devices: *ALL*[Name] **Print time**

[Format]	ASCII	FS	{ }
	Hex	1C	83
	Decimal	28	131

[Range]

[Description] Prints date in the format specified by the command \$1C \$84 with the parameter n='T'.

[Notes]

[Default] "hh:mm:ss"

[Reference] \$1C \$82, \$1C \$84

[Example]

\$1C \$84

Devices: ALL

[Name] **Set user-defined date/time formats**
 [Format] ASCII FS { } n d1...dk NUL
 Hex 1C 84 n d1...dk \$00
 Decimal 28 132 n d1...dk 0
 [Range] n = 'D', n = 'T'
 0 ≤ d0, dK ≤ 255
 [Description] Sets the format string for date and time used to printing (\$1C \$83, \$1C \$84).

- n specifies which user-defined string format is set
- D for date
- T for time
- d0..dk are the ASCII characters relative to user-defined date/time formats.
- The maximum length of the user-defined date/time format string is 64 chars.
- The following table shows characters used to create user-defined date/time formats:

CHARACTER	DESCRIPTION
l	Select italian language
E	Select english language (default language)
c	Select default data/time
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (for example, Sun).
dddd	Displays the day as a full name (for example, Sunday).
dddddd	Displays the date as a complete date in the short format where date values are formatted with day, month and year (the short date format is dd/mm/yy).
ddddddd	Displays the date as a complete date in the extended format where date values are formatted with day, month and year (the extended date format is dd mmmm, yyyy).
m	Displays the month as a number without a leading zero (1-12). If the character m is immediately after the character h or hh ,displays the minutes instead of month (see also the n character formatting).
mm	Displays the month as a number with leading zeros (01-12). If the character m is immediately after the character h or hh , displays the minutes instead of month (see also the nn character formatting).
mmm	Displays the month as an abbreviation (for example, Jan).
mmmm	Displays the month as a full month name (for example, January).
yy	Displays the year in two-digit numeric format with a leading zero.
yyyy	Displays the year in four digit numeric format.

CHARACTER	DESCRIPTION
h	Displays the hour as a number without leading zeros (0-23)
hh	Displays the hour as a number with leading zeros (00-23)
n	Displays the minutes as a number without leading zeros (0-59)
nn	Displays the minutes as a number with leading zeros (00-59)
s	Displays the seconds as a number without leading zeros (0-59)
ss	Displays the seconds as a number with leading zeros (00-59)
tttt	Displays the time in the extended format where time values are formatted with hour, minutes and seconds (the extended time format is h:mm:ss).
AM/PM	Using the 12-hour clock and displays the AM prefix in uppercase next to the hours that preceding midday and the PM prefix in uppercase next to the hours between midday and midnight.
am/pm	Using the 12-hour clock and displays the am prefix in lowercase next to the hours that preceding midday and the pm prefix in lowercase next to the hours between midday and midnight.
A/P	Using the 12-hour clock and displays the A prefix in uppercase next to the hours that preceding midday and the a prefix in uppercase next to the hours between midday and midnight.
a/p	Using the 12-hour clock and displays the a prefix in lowercase next to the hours that preceding midday and the a prefix in lowercase next to the hours between midday and midnight.

[Notes]
 [Default]
 [Reference]
 [Example]

For example to print the current time with the string format 'yy/mm/dd hh:mm:ss' follow these steps :

1. Send the following command to define the user-defined Time string format:

HEX	\$1C	\$84	\$54	\$79	\$79	\$2F	\$6D	\$6D	\$2F	\$64	\$64	\$20
ASCII	FS	{	T	y	y	/	m	m	/	d	d	h
	\$68	\$68	\$3A	\$6E	6E	\$3A	\$73	\$73	\$00			
	h	:	n	n	:	s	s	NUL				

The printer's answer ACK (\$06) if the transmission is OK otherwise NACK(\$15).

2. Send the following command to print the time :

HEX	\$1C	\$83	\$0A
ASCII	FS	{}	LF

Note : The character \$0A feeds one line based on the current line spacing.

If the date and time is 22 October 2006 at 17:35:27 (PM) the output string printed will be: 06/10/22 17:35:27

\$1C \$90

Devices: *ALL*

[Name] **Get number of stored logo**
 [Format] ASCII FS { }
 Hex 1C 90
 Decimal 28 144

[Range]
 [Description] This command sends to the printer the request of number of stored logo; the printer returns a bytes sequence as follows : `<PNn>`
 where *n* (in ASCII format) indicates the number of stored images.

[Notes]
 [Default]
 [Reference]
 [Example] If in the flash memory are stored 10 logos send this command :

HEX	\$1C	\$90
ASCII	FS	{ }

The printer's answer will be :

HEX	\$3C	\$50	\$4E	\$31	\$30	\$3E
ASCII	<	P	N	1	0	>

\$1C \$91

Devices: *ALL*

[Name] **Get pictures header list**
 [Format] ASCII FS { }
 Hex 1C 91
 Decimal 28 145

[Range]
 [Description] This command requests to the printer the list of stored logo. The printer returns a bytes sequence as follows : `<PL CrLf [N-ID CrLf]>`
 where:
CrLf indicates the two characters \$0D (Carriage return) and \$0A (Line Feed);
N is the number of stored logo;
[ID] indicates the file-name that identify the logo, a sequence of 16 bytes that was defined when the logo is stored. This field is optional because it's returned only if the logo has been found.

[Notes]
 [Default]
 [Reference] \$1C \$92, \$1C \$94
 [Example]

\$1C \$92

Devices: ALL

[Name] Get pictures header info

[Format]	ASCII	FS	{ }	nH	nL
	Hex	1C	92	nH	nL
	Decimal	28	146	nH	nL

[Range] 0 ≤ nH, nL ≤ 255

[Description] Gets the logo header info stored specified by n.

- n is the number of stored logo;
- The printer returns a byte sequence as follows :
<Ple[*ID*]>

where:

e indicates the search result
 e = 0 picture not found
 e = 1 picture found

[*ID*] indicates the file-name that identify the logo, a sequence of 16 bytes that was defined when the logo is stored. This field is optional because it's returned only if the logo has been found.

[Notes]**[Default]****[Reference]****[Example]**

\$1C \$93

Devices: ALL

[Name] Print logo

[Format] ASCII FS { } nH nL opt sp posH posL
 Hex 1C 93 nH nL opt sp posH posL
 Decimal 28 147 nH nL opt sp posH posL

[Range] 0 ≤ nH, nL ≤ 255

[Description] Prints logo defined by n.

- n is the number of image to print;
- opt is the option byte that specifies justification and rotation as shown in the following table:

BIT	DESCRIPTION	BIN	FUNCTION
0,1	Justification	00	Left
		01	Center
		10	Right
		11	User Define (on the basis of position specified by posH and posW)
2, 3	N.U.	00	Not used.
4, 6	N.U.	00	Not used.
7	Rotated print	0	Print normal.
		1	Print rotate.

- sp specifies the thickness of the image border.
- posH, posL specifies the logo's horizontal position (from the left border); used only with user-defined justification.

[Notes]
[Default]
[Reference]

[Example]

Example 1: To print logo no.10 centered and rotated transmits :
 \$1C \$93 \$00 \$0A \$81 \$01 \$00 \$00
 where
 \$1C \$93 //print logo command
 \$00 \$0A //Logo no. 10
 \$81 //printing rotated and centered
 \$01 //1 pixel of image border
 \$00 \$00 //Positioning not used

Example 2: To print logo no.10 not rotated and with a user-defined printing position transmits:
 \$1C \$93 \$00 \$0A \$03 \$01 \$00 \$50
 where
 \$1C \$93 //print logo command
 \$00 \$0A //Logo no. 10
 \$03 //printing with a user define positioning and not rotated
 \$01 //1 pixel of image border
 \$00 \$50 //Printing 10mm from the left border

\$1C \$94

Devices: ALL

[Name] **Save the image received from serial port into the flash**
 [Format] ASCII FS { } nH nL xDimH xDimL yDimH yDimL TbdH TbdL Id0..Idn d0..dn >
 Hex 1C 94 nH nL xDimH xDimL yDimH yDimL TbdH TbdL Id0..Idn d0..dn 3E
 Decimal 28 148 nH nL xDimH xDimL yDimH yDimL TbdH TbdL Id0..Idn d0..dn 62
 [Range] 0 ≤ nH, nL ≤ 255,
 0 ≤ xDimH, xDimL ≤ 255,
 0 ≤ yDimH, yDimL ≤ 255
 0 ≤ d0, dn ≤ 255

[Description] Saves the image received from serial port into the printer flash; if the number used to store logo is not already present inside the printer, the new logo is appended to stored logos. Otherwise the new logo is updated.

- nH and nL indicates the number of logo (2 bytes expressed in hexadecimal notation).
- xDimH and xDimL indicate the logo horizontal dimension in pixel (2 bytes expressed in hexadecimal notation); the value must be multiple of 16.
- yDimH and yDimL indicates the logo vertical dimension in pixel (2 bytes expressed in hexadecimal notation).
- TbdH and TbdL 2 bytes fixed to \$00 (RESERVED)
- Id0..Idn indicates the file-name of the logo, a sequence of 16 bytes to identify univocally the logo.
- d0 ...dn are the image data. The size of image is defined as follows :
 xSize = xDim /16; number of WORD (16 bit) in a horizontal image line
 Total Size = (xSize * yDim) *2;
- '>' is the character terminator (in ASCII) of this command.

The printer returns a sequence of bytes as follows :

<PC0> if the saving include an incorrect syntax or the memory in flash available for logos is finished (128Kbyte);

<PC1n> if the syntax command is correct and there's memory enough in flash for saving logos; n returns the status of the flash programming :

\$88 -> sector not erased
 \$77 -> error during programming
 \$AA -> Programming done.

[Notes] • If file-name length is shorter than 16 byte, add a terminator (0) and make padding to 16 characters.
 • If file-name extension is absent, it is automatically added to the name.

[Default]
 [Reference]
 [Example]

The following example shows the bytes sequence received from serial port to store a logo into the printer flash :

Offset	Hexadecimal	ASCII
00000000:	1C 94 00-08 01 C0 02-49 00 00 4C-6F 67 6F 32 36	° ° ° ' + ^ L o g o - 2 6
00000010:	2E 42 4D-50 00 00 00-00 00 00 00-00 00 00 00 00	.BMP
00000020:	00 00 00-00 00 00 00-00 00 00 00-00 00 00 00 00	
....		Image data
....		
....		
00008000:	00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00	
00008010:	00 00 3E	>

If the programming is successful, the printer's answer will be :

HEX	\$3C	\$50	\$43	\$31	\$AA	\$3E
ASCII	<	P	C	1	{	>

\$1C \$B0		
Devices:	KPM302	(models with BARCODE reader)
	KPM303	(models with BARCODE reader)
	TK302	(models with BARCODE reader)
	TK303	(models with BARCODE reader)

[Name] **Sets the barcode reader status.**
 [Format] ASCII FS { } n
 Hex 1C B0 n
 Decimal 28 176 n
 [Range] \$30 ≤ n ≤ \$36
 [Description] This command sets the operating status of the barcode reader; n identifies the status of the barcode setting as follows :

\$30 *TRIGGER ON/OFF:*
 Every trigger the barcode reader toggle the previous status.

\$31 *GOOD READ OFF:*
 Every trigger the barcode reader is turn ON and switch off after a timeout (standard) or after a correct reading.

\$32 *CONTINUOUS TRIGGER OFF:*
 Every trigger the barcode reader toggle the previous status.

\$33 *CONTINUOUS / AUTO POWER ON:*
 The barcode reader remains power on.

\$34 *FLASH:*
 Every trigger the barcode keeps scanning. The scanner flashes the light source when no code is decoded after the timeout duration elapsed. This mode can save the power resource and extend the operation life of the light source. The scanner can be waked up when there is a successful reading or with a trigger.

\$35 *TESTING:*
 If the barcode reader recognize a correct barcode the reading operation is not single, like the trigger on/off state, but is made permanent until the barcode is removed.

\$36 *FLASH/AUTO POWER ON:*
 The barcode reader remains in a continuous flashing condition, when occurs a reading the barcode reader is turned ON. This condition still stays for a standard timeout, then the barcode reader returns in a flashing condition.

[Notes] • The execution of this command clears the output buffer of barcode reader; if a scansion is executed without data acquisition by the host, all data read are deleted.

The printer returns a byte:
 ACK (\$06) The command is executed successfully.
 NACK(\$15) The command is not executed successfully.
 \$FF The n parameter send is not valid
 \$FE The barcode reader is not working or it not installed on the printer.

[Default]
 [Reference]
 [Example]

\$1C \$B1

Devices:	KPM302	(models with BARCODE reader)
	KPM303	(models with BARCODE reader)
	TK302	(models with BARCODE reader)
	TK303	(models with BARCODE reader)

[Name] **Get barcode reader status.**

[Format] ASCII FS { } n
Hex 1C B1 n
Decimal 28 177 n

[Range] $\$30 \leq n \leq \34

[Description] Reads the barcode reader parameters in base of n value :
n = \$30 STATUS:

Reads the barcode reader status. It returns :

- NACK (\$15) character if the command is not successful
- \$FE character if the barcode reader is not working or it not installed on the printer.
- ACK (\$06) character, followed by a status byte; the status to be transmitted is shown in the table below:

BIT	VALUE	FUNCTION
0, 1, 2	\$00	TRIGGER ON/OFF
	\$01	GOOD READ OFF
	\$02	CONTINUOUS TRIGGER OFF
	\$03	CONTINUOUS / AUTO POWER ON
	\$04	FLASH
	\$05	TESTING
	\$06	FLASH / AUTO POWER ON
	\$07	RESERVED
3	0	PE Off
	1	PE On
4	0	TG Off
	1	TG On
5	0	Decode OK
	1	Decode error
6, 7	-	RESERVED

- The execution of this command clears the output buffer of barcode reader; if a scansion is executed without data acquisition by the host, all data read are deleted.

n = \$31 **BYTES ON RECEPTION BUFFER:**

Indicates the number of bytes sent from barcode reader. It returns :

- NACK (\$15) character if the command is not successful or the buffer is empty
- \$FE character if the barcode reader is not working or it not installed on the printer.
- ACK (\$06) character, followed by one byte that indicates the number of bytes sent from barcode reader.

n = \$32 **BYTES READING ON OUTPUT FROM BARCODE READER**

Indicates the number of bytes sent from barcode reader. It returns :

- NACK (\$15) character if the command is not successful or the buffer is empty
- \$FE character if the barcode reader is not working or it not installed on the printer.
- ACK (\$06) character, followed by a bytes sequence B1, B2, ...Bn where n are the bytes on output from barcode reader.

n = \$33 *DELETE BYTES ON OUTPUT*

This command deletes all bytes on the output buffer from the barcode reader. It returns

- NACK (\$15) character if the command is not successful.
- \$FE character if the barcode reader is not working or it not installed on the printer.
- ACK (\$06) character if the command is successful.

n = \$34 *READING OF ONE BYTE ON OUTPUT FROM BARCODE READER*

This command reads one byte on output from barcode reader. It returns :

- NACK (\$15) character if there are no bytes on output from barcode reader.
- \$FE character if the barcode reader is not working or it not installed on the printer.
- ACK (\$06) character, followed by one byte that is the first byte present on the output FIFO from barcode reader.

[Notes] • with n = \$30 after the barcode reader executes this command, emits a beep as acoustic signalling.

[Default]

[Reference]

\$FS \$B0

[Example]

\$1C \$B2		
Devices:	KPM302	<i>(models with BARCODE reader)</i>
	KPM303	<i>(models with BARCODE reader)</i>
	TK302	<i>(models with BARCODE reader)</i>
	TK303	<i>(models with BARCODE reader)</i>

[Name] **Barcode reader Trigger.**

[Format]	ASCII	FS	{ }
	Hex	1C	B2
	Decimal	28	178

[Range]

[Description] This command execution forces a trigger of barcode reader. It returns:

- NACK (\$15) character if the command is successful.
- \$FE character if the barcode reader is not working or it not installed on the printer.
- ACK (\$06) character, if the command is successful.

[Notes]

- A trigger event may be effect on barcode reader setting, depending on the barcode reader status.
- The execution of this command clears the ouput buffer of barcode reader; if a scansion is executed without data acquisition by the host, all data read are deleted.

[Default]

[Reference]

\$FS \$B0

[Example]

\$1C \$C0Devices: *ALL***[Name] Hardware reset**

[Format]	❶	ASCII FS	{ }	\$18	\$10	\$14	\$1A
		Hex 1C	C0	18	10	14	1A
		Decimal	28	192	16	20	26
	❷	ASCII FS	{ }	\$18	\$10	\$14	\$1B
		Hex 1C	C0	18	10	14	1B
		Decimal	28	192	16	20	27

[Range]

[Description] When this command is received, the printer perform an hardware reset (like a printer power-up).

[Notes]

- This command is executed immediately, even when the data buffer is full (Busy).
- **❶** The command execution stop the communication with HOST;
- **❷** The command execution keep the communication with HOST active.

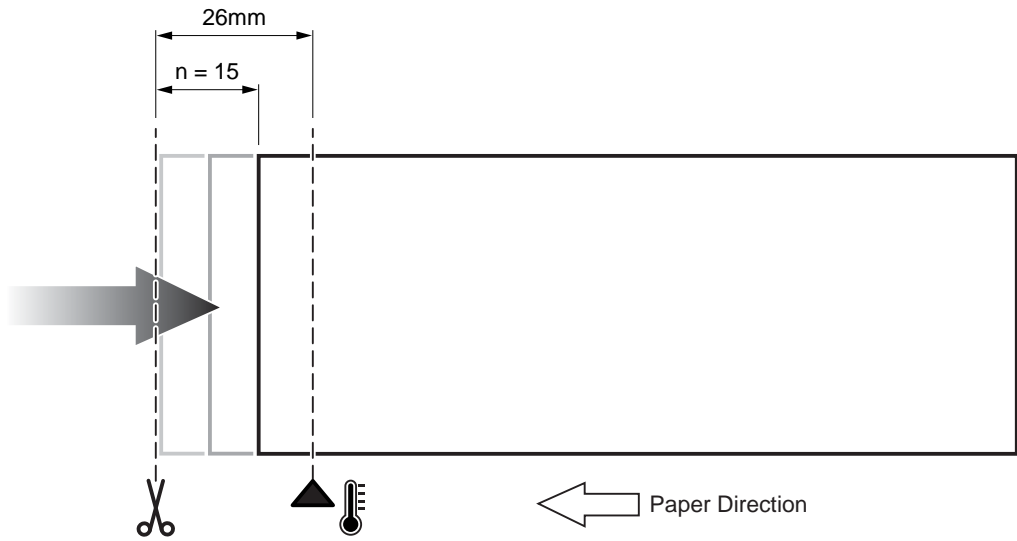
[Default]

[Reference]

[Example]

\$1C \$C1	
Devices:	KPM302
	KPM303
	TK302
	TK303

[Name] **Paper recovery after cut**
 [Format] ASCII FS { } n
 Hex 1C C1 n
 Decimal 28 193 n
 [Range] $0 \leq n \leq 24$
 [Description] Set the paper moving (in mm) toward the print head after the paper cut.



[Notes]

- Set $n = 24$ to complete recover the paper.
- WARNING: setting $n = 24$ is not recommended for paper roll with low weight.

[Default] $n = 15$ mm
 [Reference]
 [Example]

\$1D \$21

Devices: ALL

[Name] **Select character size**

[Format] ASCII GS ! n
Hex 1D 21 n
Decimal 29 33 n

[Range] $0 \leq n \leq 7$, $16 \leq n \leq 23$, $32 \leq n \leq 39$,
 $48 \leq n \leq 55$, $64 \leq n \leq 71$, $80 \leq n \leq 87$,
 $96 \leq n \leq 103$, $112 \leq n \leq 119$

[Description] Selects character height and width, as follows:

- Bits 0 to 3: to select character height (see table 2).
- Bits 4 to 7: to select character width (see table 1).

Table 1 Select character width

HEX	Decimal	WIDTH
00	0	1 (normal)
10	16	2 (width = 2x)
20	32	3 (width = 3x)
30	48	4 (width = 4x)
40	64	5 (width = 5x)
50	80	6 (width = 6x)
60	96	7 (width = 7x)
70	112	8 (width = 8x)

Table 2 Select character height

HEX	Decimal	HEIGHT
00	0	1 (normal)
01	1	2 (height = 2x)
02	2	3 (height = 3x)
03	3	4 (height = 4x)
04	4	5 (height = 5x)
05	5	6 (height = 6x)
06	6	7 (height = 7x)
07	7	8 (height = 8x)

[Notes]

- This command is effective for all characters (except HRI characters).
- If n falls outside the defined range, this command is ignored.
- Characters enlarged to different heights on the same line are aligned at the baseline or topline.
- \$1B \$21 can also be used to select character size. However, the setting of the last received command is the effective one.

[Default] $n = 0$

[Reference] \$1B \$21

[Example]

\$1D \$28 \$6B

Devices: ALL

[Name]	Print two-dimensional barcode							
[Format]	ASCII	GS	(k	pL	pH	cn	fn
	Hex	1D	28	6B	pL	pH	cn	fn
	Decimal	29	40	107	pL	pH	cn	fn

[Range]

[Description] Processes the data concerning two-dimensional barcode.

- Barcode type is specified by *cn*
- Function is specified by *fn*

cn	fn	FUNCTION	
48	65	Function 065	PDF 417: Specify the number of columns
48	66	Function 066	PDF 417: Specify the number of rows
48	67	Function 067	PDF 417: Specify the width of module
48	68	Function 068	PDF 417: Specify the module height
48	69	Function 069	PDF 417: Specify the error correction level
48	80	Function 080	PDF 417: Store the received data in the barcode save area
48	81	Function 081	PDF 417: Print the barcode data in the barcode save area
49	65	Function 065	QRcode: Specify encoding scheme
49	66	Function 066	QRcode: Specify dot size of the module
49	67	Function 067	QRcode: Specify size of barcode
49	69	Function 069	QRcode: Specify the error correction level
49	80	Function 080	QRcode: Store the received data in the barcode save area
49	81	Function 081	QRcode: Print the barcode data
51	65	Function 365	DATAMATRIX: Set encoding scheme
51	66	Function 366	DATAMATRIX: Set rotate
51	67	Function 367	DATAMATRIX: Set dot size of the module
51	68	Function 368	DATAMATRIX: Set size of barcode
51	80	Function 380	DATAMATRIX: Store the received data in the barcode save area
51	81	Function 381	DATAMATRIX: Print the barcode data in the barcode save area
52	65	Function 065	AZTEC: Specify encoding scheme
52	67	Function 067	AZTEC: Specify dot size of the module
52	68	Function 068	AZTEC: Specify size of barcode
52	69	Function 069	AZTEC: Specify the error correction level
52	80	Function 080	AZTEC: Store the received data in the barcode save area
52	81	Function 081	AZTEC: Print the barcode

[Notes]

[Default]

[Reference]

[Example]

\$1D \$28 \$6B [function 065]Devices: *ALL***[Name] Specify the number of columns of PDF417 barcode**

[Format]	ASCII	GS	(k	pL	pH	cn	fn	n
	Hex	1D	28	6B	pL	pH	cn	fn	n
	Decimal	29	40	107	pL	pH	cn	fn	n

[Range] $(pL+pH \times 256) = 3$ $(pL = 3, pH = 0)$
 cn = 48
 fn = 65
 $0 \leq n \leq 30$

[Description] Specifies the number of columns of PDF417 barcode.

- pL and pH specify the number of successive bytes to be sent
- $n = 0$ specifies auto processing
- When n is not 0, specifies the number of columns of the data area as n code word.
- When auto processing ($n = 0$) is specified, the maximum number of columns in the data area is 30 columns.

[Notes]

- The following data is not included in the number of columns:
 - start pattern and stop pattern
 - indicator code word of left and right
- Settings are effective until ESC @ is executed, the printer is reset or the power is turned off.

[Default] $n = 0$

[Reference] \$1D \$28 \$6B

[Example] To define 3 columns, the command sequence is : \$1D \$28 \$6B \$03 \$00 \$30 \$41 \$03

\$1D \$28 \$6B [function 066]Devices: *ALL***[Name] Specify the number of rows of PDF417 barcode**

[Format]	ASCII	GS	(k	pL	pH	cn	fn	n
	Hex	1D	28	6B	pL	pH	cn	fn	n
	Decimal	29	40	107	pL	pH	cn	fn	n

[Range] $(pL+pH \times 256) = 3$ $(pL = 3, pH = 0)$
 cn = 48
 fn = 66
 $n = 0, 3 \leq n \leq 20$

[Description] Specifies the number of rows of PDF417 barcode.

- pL and pH specify the number of successive bytes to be sent
- $n = 0$ specifies auto processing
- When n is not 0, specifies the number of rows of the data area as n rows.
- When auto processing ($n = 0$) is specified, the maximum number of rows is 90.

[Notes]

- Settings are effective until ESC @ is executed, the printer is reset or the power is turned off.

[Default] $n = 0$

[Reference] \$1D \$28 \$6B

[Example] To define 3 rows, the command sequence is : \$1D \$28 \$6B \$03 \$00 \$30 \$42 \$03

\$1D \$28 \$6B [function 067]

Devices: *ALL*

[Name] Specify the width of a module of PDF417 barcode

[Format]	ASCII	GS	(k	pL	pH	cn	fn	n
	Hex	1D	28	6B	pL	pH	cn	fn	n
	Decimal	29	40	107	pL	pH	cn	fn	n

[Range] (pL+pH × 256) = 3 (pL = 3, pH = 0)
 cn = 48
 fn = 67
 2 ≤ n ≤ 8

[Description] Specifies the width of a module of PDF417 barcode.

- [Notes]
- Settings are effective until ESC @ is executed, the printer is reset or the power is turned off.
 - pL and pH specify the number of successive bytes to be sent

[Default] n = 3

[Reference] \$1D \$28 \$6B

[Example] To set width = 4, the command sequence is : \$1D \$28 \$6B \$03 \$00 \$30 \$43 \$04

\$1D \$28 \$6B [function 068]

Devices: *ALL*

[Name] Specify the height of the module of PDF417 barcode

[Format]	ASCII	GS	(k	pL	pH	cn	fn	n
	Hex	1D	28	6B	pL	pH	cn	fn	n
	Decimal	29	40	107	pL	pH	cn	fn	n

[Range] (pL+pH × 256) = 3 (pL = 3, pH = 0)
 cn = 48
 fn = 68
 2 ≤ n ≤ 8

[Description] Specifies the height of the module of the PDF417 barcode.

- [Notes]
- Settings are effective until ESC @ is executed, the printer is reset or the power is turned off.
 - pL and pH specify the number of successive bytes to be sent

[Default] n = 3

[Reference] \$1D \$28 \$6B

[Example] To set height = 4, the command sequence is : \$1D \$28 \$6B \$03 \$00 \$30 \$44 \$04

\$1D \$28 \$6B [function 069]

Devices: ALL

[Name] **Specify the error correction level of PDF417 barcode**

[Format] ASCII GS (k pL pH cn fn m n
 Hex 1D 28 6B pL pH cn fn m n
 Decimal 29 40 107 pL pH cn fn m n

[Range] $(pL+pH \times 256) = 4$ ($pL = 4, pH = 0$)
 cn = 48
 fn = 69
 m = 48 $48 \leq n \leq 56$
 m = 49 $1 \leq n \leq 40$

[Description] Specifies the error correction level of PDF417 barcode.
 • pL and pH specify the number of successive bytes to be sent
 • The error correction level is specified by "level" when m = 48.
 • The error correction level is specified by "ratio" when m = 49 [$n \times 10\%$].

[Notes] • Error correction level is specified by either "level" or "ratio".
 • Error correction level specified by "level" (m = 48) is as follows. The number of the error correction code word is fixed regardless of the number of code words on the data area.

n	CORRECTION LEVEL	N. OF ERROR CORRECTION CODE WORD
48	Error correction level 0	2
49	Error correction level 1	4
50	Error correction level 2	8
51	Error correction level 3	16
52	Error correction level 4	32
53	Error correction level 5	64
54	Error correction level 6	128
55	Error correction level 7	256
56	Error correction level 8	512

• Error correction level specified by "ratio" (m = 49) is as follows. The error correction level is defined by the calculated value [number of data code word $\times n \times 0.1 = (A)$]. The number of the error correction code word is changeable in proportion to the number of the code words on the data area.

CALCULATED VALUE (A)	CORRECTION LEVEL	N. OF ERROR CORRECTION CODE WORD
0 - 3	Error correction level 1	4
4 - 10	Error correction level 2	8
11 - 20	Error correction level 3	16
21 - 45	Error correction level 4	32
46 - 100	Error correction level 5	64
101 - 200	Error correction level 6	128
201 - 400	Error correction level 7	256
> 400	Error correction level 8	512

• Settings are effective until ESC @ is executed, the printer is reset or the power is turned off.
 m = 49, n = 1 [ratio: 10%]

[Default]
 [Reference]
 [Example]

\$1D \$28 \$6B
 To set error correction=0,2 the command sequence is :\$1D \$28 \$6B \$03 \$00 \$30 \$45 \$30 \$02

\$1D \$28 \$6B [function 080]

Devices: ALL

[Name]	Store the PDF417 barcode data in the barcode save area									
[Format]	ASCII	GS	(k	pL	pH	cn	fn	m	d1...dk
	Hex	1D	28	6B	pL	pH	cn	fn	m	d1...dk
	Decimal	29	40	107	pL	pH	cn	fn	m	d1...dk
[Range]	cn = 48 fn = 80 m = 48 $0 \leq d \leq 255$ $k = (pL + pH \times 256) - 3$ <ul style="list-style-type: none"> • PDF417 barcode only with ASCII characters: $4 \leq (pL + pH \times 256) \leq 1112$ ($0 \leq pL \leq 255, 0 \leq pH \leq 4$) • PDF417 barcode only with alphanumeric characters: $4 \leq (pL + pH \times 256) \leq 1854$ ($0 \leq pL \leq 255, 0 \leq pH \leq 7$) • PDF417 barcode only with numeric characters: $4 \leq (pL + pH \times 256) \leq 2729$ ($0 \leq pL \leq 255, 0 \leq pH \leq 10$) 									
[Description]	Store the PDF417 barcode data (d1...dk) in the barcode save area.									
[Notes]	<ul style="list-style-type: none"> • Data stored in the barcode save area by this function are processed by Function 081. The data in the barcode save area are reserved after processing Function 081. • pL and pH specify the number of successive bytes to be sent • k bytes of d1...dk are processed as barcode data. • Specify only the data code word of the barcode with this function. Be sure not to include the control data in the data d1...dk because they are added automatically by the printer. • Settings are effective until ESC @ is executed, the printer is reset or the power is turned off. 									
[Default]										
[Reference]	\$1D \$28 \$6B									
[Example]										

\$1D \$28 \$6B [function 081]

Devices: ALL

[Name] **Encodes and prints the PDF417 barcode data in the barcode save area**

[Format]	ASCII	GS	(k	pL	pH	cn	fn	m
	Hex	1D	28	6B	pL	pH	cn	fn	m
	Decimal	29	40	107	pL	pH	cn	fn	m

[Range] (pL+pH × 256) = 3 (pL = 3, pH = 0)
 cn = 48
 fn = 81
 m = 48

[Description] Encodes and prints the PDF417 barcode data in the barcode save area.

[Notes]

- In standard mode, use this function when printer is "at the beginning of a line" or "there is no data in the print buffer".
- pL and pH specify the number of successive bytes to be sent
- A barcode that size exceeds the printing area cannot be printed.
- If there is any error described below in the data of the barcode save area, it cannot be printed.
 - There is no data (Function 080 is not processed).
 - If [(number of columns × number of rows) < number of code word] when auto processing is specified for number of columns and number of rows.
 - Number of code word exceeds 928 in the data area.
- When auto processing (Function 065) is specified, the number of columns is calculated by the current printing area, module width (Function 067) and the code word in the data area. Maximum number of the columns is 30.

[Default]

[Reference] \$1D \$28 \$6B

[Example] To print the PDF417 barcode data the command sequence is : \$1D \$28 \$6B \$03 \$00 \$30 \$51 \$30

\$1D \$28 \$6B [function 065]

Devices: *ALL*

[Name] **Specify encoding scheme of QRcode barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ ($pL = 3, pH = 0$)
 cn = 49
 fn = 65
 $0 \leq n \leq 1$

[Description] Specifies encoding type of QRcode barcode.

n	ENCODING SCHEME
0	QRcode
1	MicroQR

[Notes] • QRcode: Encode all extended ASCII characters data up to a maximum length of 7089 numeric digits, 4296 alphabetic characters or 2953 bytes of data.
 • pL and pH specify the number of successive bytes to be sent
 • MicroQR (a miniature version of the QRcode barcode for short message): Encode all numbers from 0 to 9 up to a maximum length of 35 characters.

[Default] n = 0

[Reference]

[Example]

\$1D \$28 \$6B [function 066]

Devices: *ALL*

[Name] **Specify dot size of the module of the QRcode barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ ($pL = 3, pH = 0$)
 cn = 49
 fn = 66
 $2 \leq n \leq 24$

[Description] Specifies numbers of dot for each pixel of QRcode barcode.

[Notes] • pL and pH specify the number of successive bytes to be sent

[Default] n = 0

[Reference]

[Example]

\$1D \$28 \$6B [function 067]

Devices: ALL

[Name] **Specify QRcode barcode size**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] (pL+pH × 256) = 3 (pL = 3, pH = 0)
 cn = 49
 fn = 67
 0 ≤ n ≤ 40

[Description] Specifies QRcode barcode eversion, as follows:

n	VERSION	n	VERSION	n	VERSION
0	AUTO	14	V14	28	V28
1	V1	15	V15	29	V29
2	V2	16	V16	30	V30
3	V3	17	V17	31	V31
4	V4	18	V18	32	V32
5	V5	19	V19	33	V33
6	V6	20	V20	34	V34
7	V7	21	V21	35	V35
8	V8	22	V22	36	V36
9	V9	23	V23	37	V37
10	V10	24	V24	38	V38
11	V11	25	V25	39	V39
12	V12	26	V26	40	V40
13	V13	27	V27		

[Notes] • pL and pH specify the number of successive bytes to be sent
 [Default] n = 0
 [Reference]
 [Example]

\$1D \$28 \$6B [function 069]

Devices: ALL

[Name] **Specify the error correction level of the QRcode barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ (pL = 3, pH = 0)
 cn = 49
 fn = 69
 $0 \leq n \leq 4$

[Description] Specifies the ECC level (Error Correction Capacity) of QRcode barcode.

n	ECC level	
0	AUTO	
1	ECC = approx 20% of barcode	Recovery Capacity = approx 7%
2	ECC = approx 37% of barcode	Recovery Capacity = approx 15%
3	ECC = approx 50% of barcode	Recovery Capacity = approx 25%
4	ECC = approx 65% of barcode	Recovery Capacity = approx 30%

[Notes] • pL and pH specify the number of successive bytes to be sent
 [Default] n = 0
 [Reference]
 [Example]

\$1D \$28 \$6B [function 080]

Devices: *ALL*

[Name]	Store the QRcode barcode data in the barcode save area									
[Format]	ASCII	GS	(k	pL	pH	cn	fn	m	d1...dk
	Hex	1D	28	6B	pL	pH	cn	fn	m	d1...dk
	Decimal	29	40	107	pL	pH	cn	fn	m	d1...dk

- [Range]
- cn = 49
 - fn = 80
 - m = 49
 - $0 \leq d \leq 255$
 - $k = (pL + pH \times 256) - 3$
 - QRcode barcode only with binary characters (8 bit):
 $4 \leq (pL + pH \times 256) \leq 2957$ ($0 \leq pL \leq 255, 0 \leq pH \leq 11$)
 - QRcode barcode only with alphanumeric characters:
 $4 \leq (pL + pH \times 256) \leq 4300$ ($0 \leq pL \leq 255, 0 \leq pH \leq 16$)
 - QRcode barcode only with numeric characters:
 $4 \leq (pL + pH \times 256) \leq 7093$ ($0 \leq pL \leq 255, 0 \leq pH \leq 27$)

- [Description]
- [Notes]
- Store the QRcode barcode data (d1...dk) in the barcode save area.
 - Data stored in the barcode save area by this function are processed by Function 081. The data in the barcode save area are reserved after processing Function 081.
 - pL and pH specify the number of successive bytes to be sent
 - k bytes of d1...dk are processed as barcode data.
 - Specify only the data code word of the barcode with this function.

[Default]
 [Reference]
 [Example]

\$1D \$28 \$6B [function 081]

Devices: *ALL*

[Name]	Prints the QRcode barcode data									
[Format]	ASCII	GS	(k	pL	pH	cn	fn	m	
	Hex	1D	28	6B	pL	pH	cn	fn	m	
	Decimal	29	40	107	pL	pH	cn	fn	m	

- [Range]
- $(pL+pH \times 256) = 3$ ($pL = 3, pH = 0$)
 - cn = 49
 - fn = 81
 - m = 49

- [Description]
- [Notes]
- Prints the QRcode barcode in the current position.
 - pL and pH specify the number of successive bytes to be sent
- [Default]
 [Reference]
 [Example]

\$1D \$28 \$6B [function 365]

Devices: ALL

[Name] **Specify the encoding scheme of DATAMATRIX barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ (pL = 3, pH = 0)
 cn = 51
 fn = 65
 $0 \leq n \leq 6$

[Description] Set the encoding scheme specified by n as follows:

n	ENCODING SCHEME
0	Ascii
1	C40
2	Text
3	X12
4	Edifact
5	Base256
6	AutoBest

[Notes] • pL and pH specify the number of successive bytes to be sent

[Default]

[Reference] \$1D \$28 \$6B

[Example] To set encoding = Ascii, the command sequence is: \$1D \$28 \$6B \$03 \$00 \$33 \$41 \$00

\$1D \$28 \$6B [function 366]

Devices: ALL

[Name] **Set rotation of DATAMATRIX barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ (pL = 3, pH = 0)
 cn = 51
 fn = 66
 n = 0, 1

[Description] Set rotate by n as follows:

n	ROTATION
0	No rotation
1	Rotation

[Notes] • pL and pH specify the number of successive bytes to be sent

[Default]

[Reference] \$1D \$28 \$6B

[Example]

\$1D \$28 \$6B [function 367]Devices: *ALL*

[Name]	Set dot size of the module of DATAMATRIX barcode									
[Format]	ASCII	GS	(k	pL	pH	cn	fn	n	
	Hex	1D	28	6B	pL	pH	cn	fn	n	
	Decimal	29	40	107	pL	pH	cn	fn	n	
[Range]	(pL+pH × 256) = 3		(pL = 3, pH = 0)							
	cn = 51									
	fn = 67									
	2 ≤ n ≤ 24									
[Description]	Set dot size of the module of the DATAMATRIX barcode. n = dot dimension									
[Notes]	• pL and pH specify the number of successive bytes to be sent									
[Default]	n = 6									
[Reference]	\$1D \$28 \$6B									
[Example]	To set dot size = 6 the command sequence is : \$1D \$28 \$6B \$03 \$00 \$33 \$43 \$06									

\$1D \$28 \$6B [function 368]

Devices: ALL

[Name] **Set size of DATAMATRIX barcode**
 [Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n
 [Range] $(pL + pH \times 256) = 3$ ($pL = 3, pH = 0$)
 cn = 51
 fn = 68
 $1 \leq n \leq 29$
 [Description] Set the size of DATAMATRIX barcode specified by n as follows:

n	BARCODE SIZE
1	10 x 10
2	12 x 12
3	14 x 14
4	16 x 16
5	18 x 18
6	20 x 20
7	22 x 22
8	24 x 24
8	26 x 26
10	32 x 32
11	36 x 36
12	40 x 40
13	44 x 44
14	48 x 48
15	52 x 52

n	BARCODE SIZE
16	64 x 64
17	72 x 72
18	80 x 80
19	88 x 88
20	96 x 96
21	104 x 104
22	120 x 120
23	132 x 132
24	144 x 144
25	8 x 18
26	8 x 32
27	12 x 26
28	12 x 36
29	16 x 36

[Notes] • pL and pH specify the number of successive bytes to be sent
 [Default] DmtxSymbolSquareAuto
 [Reference] \$1D \$28 \$6B
 [Example]

\$1D \$28 \$6B [function 380]

Devices: ALL

[Name] Store the DATAMATRIX barcode data in the barcode save area

[Format]	ASCII	GS	(k	pL	pH	cn	fn	m	d1...dk
	Hex	1D	28	6B	pL	pH	cn	fn	m	d1...dk
	Decimal	29	40	107	pL	pH	cn	fn	m	d1...dk

[Range]

cn = 51
 fn = 80
 m = 51
 $0 \leq d \leq 255$
 $k = (pL + pH \times 256) - 3$

- DATAMATRIX barcode only with ASCII characters (8 bit) :
 $4 \leq (pL + pH \times 256) \leq 1560$ ($0 \leq pL \leq 255, 0 \leq pH \leq 6$)
- DATAMATRIX barcode only with alphanumeric characters:
 $4 \leq (pL + pH \times 256) \leq 2339$ ($0 \leq pL \leq 255, 0 \leq pH \leq 9$)
- DATAMATRIX barcode only with numeric characters:
 $4 \leq (pL + pH \times 256) \leq 3120$ ($0 \leq pL \leq 255, 0 \leq pH \leq 12$)

[Description] Store the DATAMATRIX barcode data (d1...dk) in the barcode save area.

[Notes]

- Data stored in the barcode save area by this function are processed by Function 081. The data in the barcode save area reserved after processing Function 381.
- k bytes of d1...dk are processed as barcode data.
- Specify only the data code word of the barcode with this function. Be sure not to include the control data in the data d1...dk because they are added automatically by the printer.
- Settings are effective until ESC @ is executed, the printer is reset or the power is turned off.

[Default]**[Reference]**

\$1D \$28 \$6B

[Example]

\$1D \$28 \$6B [function 381]

Devices: ALL

[Name] **Encodes and prints the DATAMATRIX barcode data in the barcode save area**

[Format]	ASCII	GS	(k	pL	pH	cn	fn	m
	Hex	1D	28	6B	pL	pH	cn	fn	m
	Decimal	29	40	107	pL	pH	cn	fn	m

[Range] (pL+pH × 256) = 3 (pL = 3, pH = 0)
 cn = 51
 fn = 81
 m = 51

[Description] Encodes and prints the DATAMATRIX barcode data in the barcode save area.

- [Notes]
- In standard mode, use this function when printer is "at the beginning of a line" or "there is no data in the print buffer".
 - pL and pH specify the number of successive bytes to be sent
 - A barcode that size exceeds the printing area cannot be printed.
 - If there is any error described below in the data of the barcode save area, it cannot be printer.
 - There is no data (Function 380 is not processed).
 - If [(number of columns × number of rows) < number of code word] when auto processing is specified for number of columns and number of rows.
 - Number of code word exceeds 928 in the data area.

[Default]

[Reference] \$1D \$28 \$6B

[Example] To print the DATAMATRIX barcode data the command sequence is : \$1D \$28 \$6B \$03 \$00 \$33 \$51 \$33

\$1D \$28 \$6B [function 065]

Devices: ALL

[Name] **Specify encoding scheme of AZTEC barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ ($pL = 3, pH = 0$)
 cn = 52
 fn = 65
 $0 \leq n \leq 1$

[Description] Specifies encoding type of AZTEC barcode.

n	ENCODING SCHEME
0	FULL AZTEC
1	AZTEC RUNE

[Notes] • Full Aztec: Encode all extended ASCII characters data up to a maximum length of approximately 3823 numeric or 3067 alphabetic characters or 1914 bytes of data.
 • pL and pH specify the number of successive bytes to be sent
 • Aztec Rune (Compact Aztec Code, sometimes called Small Aztec Code): Encode all numbers from 0 to 255 up to a maximum length of 3 numbers.

[Default] n = 0

[Reference]

[Example]

\$1D \$28 \$6B [function 067]

Devices: ALL

[Name] **Specify dot size of the module of the AZTEC barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ ($pL = 3, pH = 0$)
 cn = 52
 fn = 67
 $2 \leq n \leq 24$

[Description] Specifies numbers of dot for each pixel of AZTEC barcode.

[Notes] • pL and pH specify the number of successive bytes to be sent

[Default] n = 0

[Reference]

[Example]

\$1D \$28 \$6B [function 068]

Devices: ALL

[Name] **Specify AZTEC barcode size**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 3$ (pL = 3, pH = 0)
 cn = 52
 fn = 68
 $0 \leq n \leq 36$

[Description] Specifies AZTEC barcode format (rows and columns), as follows:

n	FORMAT	n	FORMAT	n	FORMAT
0	AUTO	13	C53X53	26	C109X109
1	C15X15 Compact	14	C57X57	27	C113X113
2	C19X19 Compact	15	C61X61	28	C117X117
3	C23X23 Compact	16	C67X67	29	C121X121
4	C27X27 Compact	17	C71X71	30	C125X125
5	C19X19	18	C75X75	31	C131X131
6	C23X23	19	C79X79	32	C135X135
7	C27X27	20	C83X83	33	C139X139
8	C31X31	21	C87X87	34	C143X143
9	C37X37	22	C91X91	35	C147X147
10	C41X41	23	C95X95	36	C151X151
11	C45X45	24	C101X101		
12	C49X49	25	C105X105		

[Notes] • pL and pH specify the number of successive bytes to be sent
 [Default] n = 0
 [Reference]
 [Example]

\$1D \$28 \$6B [function 069]

Devices: ALL

[Name] **Specify the error correction level of the AZTEC barcode**

[Format] ASCII GS (k pL pH cn fn n
 Hex 1D 28 6B pL pH cn fn n
 Decimal 29 40 107 pL pH cn fn n

[Range] $(pL+pH \times 256) = 4$ (pL = 4, pH = 0)
 cn = 52
 fn = 69
 $0 \leq n \leq 4$

[Description] Specifies the ECC level (Error Correction Capacity) of AZTEC barcode.

n	ECC level
0	AUTO
1	> 10 % + 3 codewords
2	> 23 % + 3 codewords
3	> 36 % + 3 codewords
4	> 50 % + 3 codewords

- It is not possible to select both barcode size and error correction capacity for the same barcode. If both options are selected then the error correction capacity selection will be ignored.
 - pL and pH specify the number of successive bytes to be sent
- n = 0

[Notes]
 [Default]
 [Reference]
 [Example]

\$1D \$28 \$6B [function 080]

Devices: *ALL*

[Name]	Store the AZTEC barcode data in the barcode save area									
[Format]	ASCII	GS	(k	pL	pH	cn	fn	m	d1...dk
	Hex	1D	28	6B	pL	pH	cn	fn	m	d1...dk
	Decimal	29	40	107	pL	pH	cn	fn	m	d1...dk

- [Range]
- cn = 52
 - fn = 80
 - m = 52
 - $0 \leq d \leq 255$
 - $k = (pL + pH \times 256) - 3$
 - AZTEC barcode only with ASCII characters:
 - $4 \leq (pL + pH \times 256) \leq 1918$ ($0 \leq pL \leq 255, 0 \leq pH \leq 7$)
 - AZTEC barcode only with alphanumeric characters:
 - $4 \leq (pL + pH \times 256) \leq 3071$ ($0 \leq pL \leq 255, 0 \leq pH \leq 11$)
 - AZTEC barcode only with numeric characters:
 - $4 \leq (pL + pH \times 256) \leq 3836$ ($0 \leq pL \leq 255, 0 \leq pH \leq 14$)

- [Description] Store the AZTEC barcode data (d1...dk) in the barcode save area.
- [Notes]
- Data stored in the barcode save area by this function are processed by Function 081. The data in the barcode save area are reserved after processing Function 081.
 - pL and pH specify the number of successive bytes to be sent
 - k bytes of d1...dk are processed as barcode data.
 - Specify only the data code word of the barcode with this function.

[Default]
 [Reference]
 [Example]

\$1D \$28 \$6B [function 081]

Devices: *ALL*

[Name]	Prints the AZTEC barcode data									
[Format]	ASCII	GS	(k	pL	pH	cn	fn	m	
	Hex	1D	28	6B	pL	pH	cn	fn	m	
	Decimal	29	40	107	pL	pH	cn	fn	m	

- [Range]
- $(pL+pH \times 256) = 3$ ($pL = 3, pH = 0$)
 - cn = 52
 - fn = 81
 - m = 48

- [Description] Prints the AZTEC barcode in the current position.
- [Notes]
- pL and pH specify the number of successive bytes to be sent

[Default]
 [Reference]
 [Example]

\$1D \$2A

Devices: ALL

[Name] **Define downloaded bit image**

[Format] ASCII GS * x y d1...d(x × y × 8)
 Hex 1D 2A x y d1...d(x × y × 8)
 Decimal 29 42 x y d1...d(x × y × 8)

[Range] 1 ≤ x ≤ 255
 1 ≤ y ≤ 48
 x × y ≤ 1536
 0 ≤ d ≤ 255

[Description] Defines a downloaded bit image using the number of dots specified by x and y.

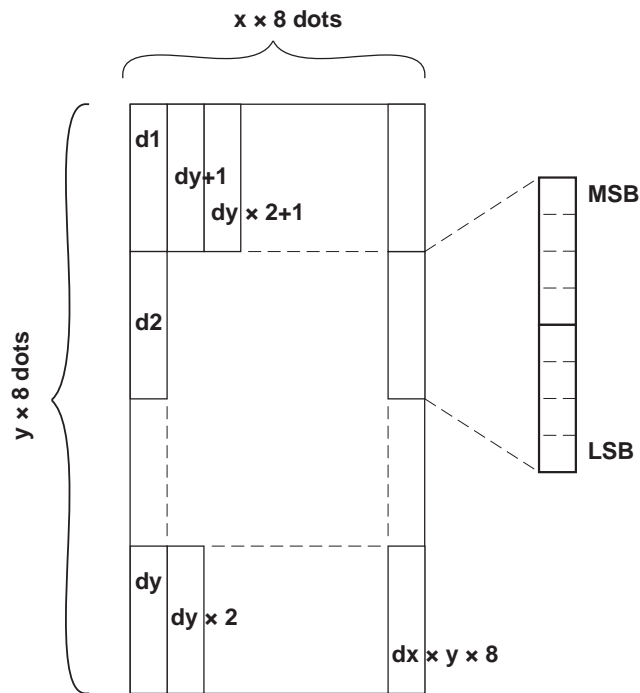
- x specifies the number of dots in the horizontal direction.
- y specifies the number of dots in the vertical direction.

[Notes]

- The number of dots in the horizontal direction is x × 8, in the vertical direction it is y × 8.
- If x × y is out of the specified range, this command is disabled.
- The d indicates bit-image data. Data (d) specifies a bit printed to 1 and not printed to 0.
- The downloaded bit image definition is cleared when:
 - 1) \$1B \$40 is executed.
 - 2) \$1B \$26 is executed.

Printer is reset or the power is turned off.

- The following figure shows the relationship between the downloaded bit image and the printed data.



[Reference] \$1D \$5C

[Example]

\$1D \$2F

Devices: ALL

[Name] **Print downloaded bit image**
 [Format] ASCII GS / m
 Hex 1D 2F m
 Decimal 29 47 m

[Range]
 [Description] Prints a downloaded bit image using the mode specified by m. *m* selects a mode from the table below :

m	MODE
0,48	Normal
1, 49	Double width
2, 50	Double height
3, 51	Quadruple

[Notes]

- This command is ignored if a downloaded bit image has not been defined.
- In standard mode, this command is effective only when there is no data in the print buffer.
- This command has no effect in the print modes (emphasized, underline, character size, or white/black reverse printing), except for upside-down printing mode.
- If the downloaded bit-image to be printed exceeds the printable area, the excess data is not printed
- If the printing area width set by \$1D \$4C and \$1D \$57 is less than one line in vertical, the following processing is performed only on the line in question:
 - 1) The printing area width is extended to the right up to one line in vertical. In this case, printing does not exceed the printable area.
 - 2) If the printing area width cannot be extended by one line in vertical, the left margin is reduced to accommodate one line in vertical.

[Reference] \$1D \$2A
 [Example]

\$1D \$3ADevices: *ALL*[Name] **Start/end macro definition**

[Format]	ASCII	GS	:
	Hex	1D	3A
	Decimal	29	58

[Description] Starts or ends macro definition.

[Notes]

- Macro definition starts when this command is received during normal operation.
- When \$1D \$5E is received during macro definition, the printer ends macro definition and clears all definitions.
- Macros are not defined when power is turned on to the machine.
- Macro content is not cancelled by the \$1B \$40 command. Therefore, \$1B \$40 may be included in the content of macro definitions.
- If the printer receives \$1D \$3A a second time after previously receiving \$1D \$3A, the printer remains in macro undefined status.
- The contents of the macro can be defined up to 1024 bytes. If the macro definition exceeds 1024 bytes, excess data is not stored.

[Default]

[Reference] \$1D \$5E

[Example]

\$1D \$42Devices: *ALL*[Name] **Turn white/black reverse printing mode on/off**

[Format]	ASCII	GS	B	n
	Hex	1D	42	n
	Decimal	29	66	n

[Range] $0 \leq n \leq 255$

[Description] Turns white/black reverse printing mode on or off.

[Notes]

- When the LSB of n is 0, white/black reverse printing is turned off.
- When the LSB of n is 1, white/black reverse printing is turned on.
- Only the LSB of n is effective.
- This command is available for both built-in and user-defined characters.
- This command does not affect bit image, downloaded bit image, bar code, HRI characters and spacing skipped by \$09, \$1B \$24 and \$1B \$5C.
- This command does not affect white space between lines.
- White/black reverse mode has a higher priority than underline mode. Even if underline mode is on, it will be disabled (but not cancelled) when white/black reverse mode is selected.

[Default] $n = 0$

[Reference]

[Example]

\$1D \$48

Devices: *ALL*

[Name] **Select printing position of Human Readable Interpretation (HRI) characters**

[Format] ASCII GS H n
 Hex 1D 48 n
 Decimal 29 72 n

[Range] $0 \leq n \leq 3, 48 \leq n \leq 51$

[Description] Selects the printing position of HRI characters when printing bar codes. *n* selects the printing positions as follows:

n	FUNCTION
0, 48	Not printed
1, 49	Above the barcode.
2, 50	Below the barcode.
3, 51	Both above and below the barcode.

[Notes] HRI characters are printed using the font specified by \$1D \$66.

[Default] *n* = 0

[Reference] \$1D \$66, \$1D \$6B

[Example]

\$1D \$49

Devices: ALL

[Name] **Transmit printer ID**
 [Format] ASCII GS l n
 Hex 1D 49 n
 Decimal 29 73 n
 [Range] $1 \leq n \leq 3, 49 \leq n \leq 51$

[Description] Transmits the printer ID specified by n follows:

n	PRINTER ID	SPECIFICATION
1, 49	Printer model ID	\$75 (KPM202, KPM203) \$75 (KPM302, KPM303) \$75 (TK302, TK303) \$75 (TK202, TK203)
2, 50	Type ID	See table below
3, 51	ROM version ID	Depends on ROM version (4 characters)

n = 2, 50 Type ID

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	2-byte characters codes not supported
1	Off	00	0	Autocutter not supplied
	On	02	2	Autocutter supplied
2	Off	00	0	Thermal paper w/o label
	On	04	4	Thermal paper label
3	-	-	-	Undefined
4	Off	00	0	Not used. Fixed to Off
5	-	-	-	Undefined
6	-	-	-	Undefined
7	Off	00	0	Not used. Fixed to Off

[Notes]

- This command is executed when the data is processed in the data buffer. Therefore, there could be a time lag between command reception and data transmission, depending on data buffer status.
- The printer only transmits 1 byte (printer ID) without confirmation that the host is ready to receive data.

[Default]
 [Reference]
 [Example]

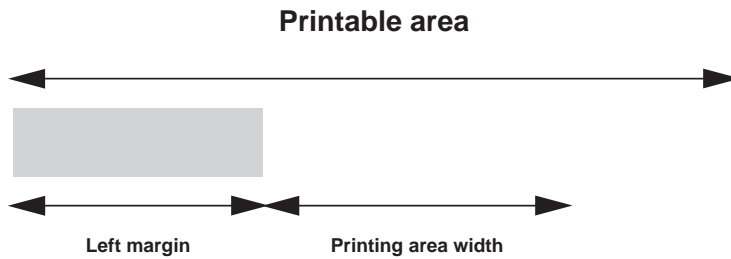
\$1D \$4C

Devices: *ALL*

[Name]	Set left margin				
[Format]	ASCII	GS	L	nL	nH
	Hex	1D	4C	nL	nH
	Decimal	29	76	nL	nH

[Range] 0 ≤ nL, nH ≤ 255

[Description] Sets the left margin.
 • The left margin is set to $[(nL + nH \times 256) \times (\text{horizontal motion unit})]$ inches.



- [Notes]
- This command is enabled only if set at the beginning of the line.
 - If the setting exceeds the printable area, the maximum value of the printable area is used.
 - If the left margin + printing area width is greater than the printable area, the printing area width is set at maximum value.
 - The horizontal and vertical motion unit are specified by \$1D \$50. Changing the horizontal or vertical motion unit does not affect the current left margin.
 - The \$1D \$50 command can change the horizontal (and vertical) motion unit.
 - However, the value cannot be less than the minimum horizontal movement amount and it must be in even units of the minimum horizontal movement amount.

[Default]
 [Reference] \$1D \$50, \$1D \$57
 [Example]

\$1D \$50

Devices: ALL

[Name] **Set horizontal and vertical motion units**
 [Format] ASCII GS P x y
 Hex 1D 50 x y
 Decimal 29 80 x y
 [Range] $0 \leq x, y \leq 255$
 [Description] Sets the horizontal and vertical motion units to 1/x inch and 1/y inch respectively. When x is set to 0, the default setting value is used. When y is set to 0, the default setting value is used.
 [Notes]

- The horizontal direction is perpendicular to the paper feed direction.
- In standard mode, the following commands use x or y, regardless of character rotation (upside-down or 90° clockwise rotation):
 - ❶ Commands using x : \$1B \$20, \$1B \$24, \$1B \$5C, \$1D \$4C, \$1D \$57.
 - ❷ Commands using y : \$1B \$33, \$1B \$4A.
- This command does not affect the previously specified values.
- The calculated result from combining this command with others is truncated to the minimum value of the mechanical pitch or an exact multiple of that value.

 [Default] $x = 204, y = 408$
 [Reference] \$1B \$20, \$1B \$24, \$1B \$5C, \$1B \$33, \$1B \$4A, \$1D \$4C, \$1D \$57
 [Example]

❶ \$1D \$56, ❷ \$1D \$56

Devices: ALL

[Name] **Select cut mode**
 [Format]

❶	ASCII	GS	V	m	
	Hex	1D	56	m	
	Decimal	29	86	m	
❷	ASCII	GS	V	m	n
	Hex	1D	56	m	n
	Decimal	29	86	m	n

 [Range]

- ❶ $m = 0, 48$
- ❷ $m = 65, 0 \leq n \leq 255$

 [Description] Selects cut mode and executes the cut command. *m* selects cut mode as follows:

m	FUNCTION
0, 48	Total cut.
65, 66	Form feed (cut position + [n × vertical motion unit]) and total cut

 KPM202, KPM203, **TK202, TK203**

m	FUNCTION
65, 66	Form feed (cut position + [n × vertical motion unit]) and total cut

 [Notes]

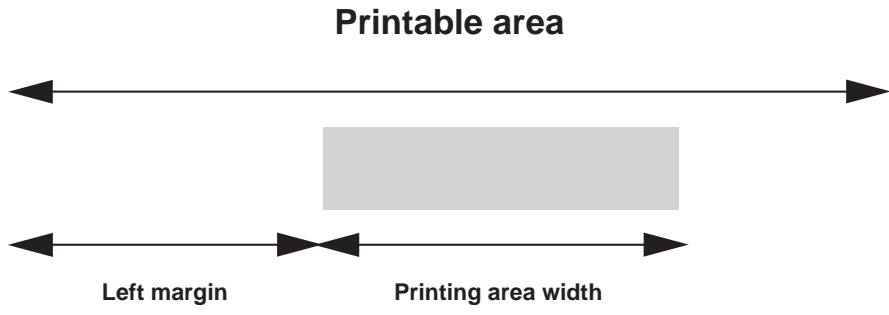
- This command is only enabled if set at the beginning of the line.
- The horizontal and vertical motion units are specified by \$1D \$50.

 [Default]
 [Reference] \$1B \$69
 [Example]

\$1D \$57

Devices: ALL

[Name]	Set printing area width				
[Format]	ASCII	GS	W	nL	nH
	Hex	1D	57	nL	nH
	Decimal	29	87	nL	nH
[Range]	0 ≤ nL, nH ≤ 255				
	0 ≤ nL + nH × 256 ≤ 832				
[Description]	Sets the printing area width to the area specified by nL and nH.				
	• The left margin is set to [(nL + nH × 256) × (horizontal motion unit)] inches.				



- [Notes]
- This command is only enabled if set at the beginning of the line.
 - If the right margin is greater than the printable area, the printing area width is set at maximum value.
 - If the printing area width = 0, it is set at the maximum value.
 - The horizontal and vertical motion units are specified by \$1D \$50. Changing the horizontal or vertical motion unit does not affect the current left margin.
 - The \$1D \$50 command can change the horizontal (and vertical) motion unit. However, the value cannot be less than the minimum horizontal movement amount and it must be in even units of the minimum horizontal movement amount.

[Default]
 [Reference] \$1D \$4C, \$1D \$50
 [Example]

\$1D \$5E

Devices:	ALL					
[Name]	Execute macro					
[Format]	ASCII	GS	^	r	t	m
	Hex	1D	5E	r	t	m
	Decimal	29	94	r	t	m
[Range]	0 ≤ r, t ≤ 255 0 ≤ m ≤ 1					
[Description]	<p>Executes a macro.</p> <ul style="list-style-type: none"> • r specifies the number of times to execute the macro. • t specifies the waiting time for executing the macro. The waiting time is t × 100 msec. for each macro execution. • m specifies macro executing mode: When the LSB of m = 0, the macro is executed r times continuously at the interval specified by t. When the LSB of m = 1, after waiting for the period specified by t, the LED indicator blinks and the printer waits for the FEED button to be pressed. After the button is pressed, the printer executes the macro once. The printer repeats the operation r times. 					
[Notes]	<ul style="list-style-type: none"> • This command has an interval of (t × 100 msec.) after a macro is executed by t. • If this command is received while a macro is being defined, the macro definition is aborted and the definition is cleared. • If the macro is not defined or if r is 0, nothing is executed. • When the macro is executed by pressing the FEED button (m=1), the paper cannot be fed using the FEED button. 					
[Default]						
[Reference]	\$1D \$3A					
[Example]						

\$1D \$65 \$35

Devices:	KPM302	<i>(models with selector)</i>			
	KPM303	<i>(models with selector)</i>			
[Name]	Perform the ticket ejection				
[Format]	ASCII	GS	e	5	
	Hex	1D	65	35	
	Decimal	29	101	53	
[Range]					
[Description]	This command performs the ejection of the printed ticket.				
[Notes]					
[Default]					
[Reference]					
[Example]					

\$1D \$66

Devices: *ALL*

[Name] **Select font for HRI characters**
 [Format] ASCII GS f n
 Hex 1D 66 n
 Decimal 29 102 n
 [Range] n = 0, 1, 48, 49
 [Description] Selects a font for the HRI characters used when printing a bar code. *n* selects a font from the following table:

n	FONT
0, 48	Font A
1, 49	Font B

[Notes] HRI characters are printed at the position specified by \$1D \$48.
 [Default] n = 0
 [Reference] \$1D \$48, \$1D \$6B
 [Example]

\$1D \$68

Devices: *ALL*

[Name] **Set bar code height**
 [Format] ASCII GS h n
 Hex 1D 68 n
 Decimal 29 104 n
 [Range] $1 \leq n \leq 255$
 [Description] Sets the height of the bar code. *n* specifies the number of vertical dots.
 [Notes]
 [Default] n = 162 (20.25 mm)
 [Reference] \$1D \$6B
 [Example]

① \$1D \$6B, ② \$1D \$6B

Devices: ALL

[Name] **Print bar code**

[Format]	①	ASCII	GS	k	m	NUL
		Hex	1D	6B	m	00
		Decimal	29	107	m	0
	②	ASCII	GS	k	m	n
		Hex	1D	6B	m	n
		Decimal	29	107	m	n

[Range]	①	0 ≤ m ≤ 20
,	②	65 ≤ m ≤ 90

[Description] Selects a bar code system and prints the bar code. *m* selects a bar code system as follows:

m	BARCODE SYSTEM	No. OF CHARACTERS	REMARKS
0	UPC-A	11 ≤ k ≤ 12	48 ≤ d ≤ 57
1	UPC-E	11 ≤ k ≤ 12	48 ≤ d ≤ 57
2	EAN13 (JAN)	12 ≤ k ≤ 13	48 ≤ d ≤ 57
3	EAN8 (JAN)	7 ≤ k ≤ 8	48 ≤ d ≤ 57
4	CODE39	1 ≤ k	48 ≤ d ≤ 57, 65 ≤ d ≤ 90, 32, 36, 37, 43, 45, 46, 47
5	ITF	1 ≤ k (even number)	48 ≤ d ≤ 57
6	CODABAR	1 ≤ k	48 ≤ d ≤ 57, 65 ≤ d1 ≤ 68, 36, 43, 45, 46, 47, 58
7	CODE93	1 ≤ k ≤ 255	1 ≤ d ≤ 127
8	CODE128	2 ≤ k ≤ 255	1 ≤ d ≤ 127
20	CODE32	8 ≤ k ≤ 9	48 ≤ d ≤ 57

65	UPC-A	11 ≤ n ≤ 12	48 ≤ d ≤ 57
66	UPC-E	11 ≤ n ≤ 12	48 ≤ d ≤ 57
67	EAN13 (JAN)	12 ≤ n ≤ 13	48 ≤ d ≤ 57
68	EAN8 (JAN)	7 ≤ n ≤ 8	48 ≤ d ≤ 57
69	CODE39	1 ≤ n ≤ 255	48 ≤ d ≤ 57, 65 ≤ d ≤ 90, 32, 36, 37, 43, 45, 46, 47
70	ITF	1 ≤ n ≤ 255	48 ≤ d ≤ 57
71	CODABAR	1 ≤ n ≤ 255	48 ≤ d ≤ 57, 65 ≤ d1 ≤ 68, 36, 43, 45, 46, 47, 58
72	CODE93	1 ≤ n ≤ 255	0 ≤ d ≤ 127
73	CODE128	2 ≤ n ≤ 255	0 ≤ d ≤ 127
90	CODE32	8 ≤ n ≤ 9	48 ≤ d ≤ 57

[Notes]

- If *d* is outside of the specified range, the printer prints the following message: "BAR CODE GENERATOR IS NOT OK!" and processes the data which follows as normal data.
- If the horizontal size exceeds the printing area, the printer only feeds the paper.
- This command feeds as much paper as is required to print the bar code, regardless of the line spacing specified by \$1B \$32 or \$1B \$33.
- After printing the bar code, this command sets the print position to the beginning of the line.
- This command is not affected by print modes (emphasized, double-strike, underline or character size), except for upside-down and justification mode.

[Notes per ❶]

- This command ends with a NUL code.
- When the bar code system used is UPC-A or UPC-E, the printer prints the bar code data after receiving 11 (without check digit) or 12 (with check digit) bytes bar code data.
- When the bar code system used is EAN13, the printer prints the bar code data after receiving 12 (without check digit) or 13 (with check digit) bytes bar code data.
- When the bar code system used is EAN8, the printer prints the bar code data after receiving 7 (without check digit) or 8 (with check digit) bytes bar code data.
- The number of data for ITF bar code must be even numbers. When an odd number of data is input, the printer ignores the last received data.

[Notes per ❷]

- If *n* is outside of the specified range, the printer stops command processing and processes the following data as normal data.

When CODE93 is used:

- The printer prints an HRI character (o) as a start character at the beginning of the HRI character string.
- The printer prints an HRI character (o) as a stop character at the end of the HRI character string.
- The printer prints an HRI character (n) as a control character (\$00 to \$1F and \$7F).

When CODE128 is used:

- When using CODE128 in this printer, please note the following regarding data transmission:
- The top part of the bar code data string must be a code set selection character (CODE A, CODE B or CODE C) which selects the first code set.
- Special characters are defined by combining two characters “{” and one character. ASCII character “f” is defined by transmitting “{” twice, consecutively.

SPECIFIC CHARACTER	DATA TRANSMISSION		
	ASCII	HEX	Decimal
SHIFT	{S	7B, 53	123, 83
CODE A	{A	7B, 41	123, 65
CODE B	{B	7B, 42	123, 66
CODE C	{C	7B, 43	123, 67
FNC1	{1	7B, 31	123, 49
FNC2	{2	7B, 32	123, 50
FNC3	{3	7B, 33	123, 51
FNC4	{4	7B, 34	123, 52
{f	{{	7B, 7B	123, 123

When UPC-E is used, introducing the barcode characters, the printer prints:

TRANSMITTED DATA											PRINTING DATA					
d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d2	d3	d9	d10	d11	
0	0-9	0-9	0	0	0	0	0	0-9	0-9	0-9	d2	d3	d9	d10	d11	0
0	0-9	0-9	1	0	0	0	0	0-9	0-9	0-9	d2	d3	d9	d10	d11	1
0	0-9	0-9	2	0	0	0	0	0-9	0-9	0-9	d2	d3	d9	d10	d11	2
0	0-9	0-9	3-9	0	0	0	0	0	0-9	0-9	d2	d3	d4	d10	d11	3
0	0-9	0-9	0-9	1-9	0	0	0	0	0	0-9	d2	d3	d4	d5	d11	4
0	0-9	0-9	0-9	0-9	1-9	0	0	0	0	5-9	d2	d3	d4	d5	d6	d11

[Default]

[Reference]

[Example]

- \$1D \$48, \$1D \$66, \$1D \$68, \$1D \$77
- ❶ Example of print the Bar Code 39
1D 6B 04 54 45 53 54 00
 - ❷ Example of print the Bar Code 39
1D 6B 45 04 54 45 53 54

\$1D \$70 \$69

Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>

[Name]	Initialize selector			
[Format]	ASCII	GS	p	i
	Hex	1D	70	69
	Decimal	29	112	105

[Range]

[Description] This command performs a movement of the selector mechanisms in the two available positions. If the selector is mechanically unable to move, the flag status indicates an error.

[Notes] At the end of the movement, selector is set in the "Open" position (default).

[Default]

[Reference]

[Example]

\$1D \$70 \$6F

Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>

[Name]	Set selector in "Open" position			
[Format]	ASCII	GS	p	o
	Hex	1D	70	6F
	Decimal	29	112	111

[Range]

[Description] This command set the selector in the "Open" position: the paper exits the printer regularly. If the selector position is already the desired one, this command does not generate any movement.

[Notes]

[Default]

[Reference]

[Example]

\$1D \$70 \$73

Devices:	KPM302	(models with selector)
	KPM303	(models with selector)

[Name] **Set selector in "Storage" position**

[Format]	ASCII	GS	p	s
	Hex	1D	70	73
	Decimal	29	112	115

[Range]

[Description] This command set the selector in the "Storage" position: paper exits the printer downwards. If the selector position is already the desired one, this command does not generate any movement.

[Notes]

[Default]

[Reference]

[Example]

\$1D \$72

Devices:	ALL
----------	-----

[Name] **Transmit status**

[Format]	ASCII	GS	r	n
	Hex	1D	72	n
	Decimal	29	114	n

[Range] n = 1, 49

[Description] Transmits the status specified by n as follows:

n	FUNCTION
1, 49	Transmits paper sensor status (as for \$1B \$76).

Paper sensor status (n = 1, 49):

BIT	OFF/ON	HEX	Decimal	FUNCTION
0, 1	Off	00	0	Near paper end sensor: paper present
	On	03	3	Near paper end sensor: paper not present
2,3	Off	00	0	Paper end sensor: paper present
	On	(0C)	(12)	Paper end sensor: paper not present
4	Off	00	0	Not used. Fixed to Off
5	-	-	-	Undefined
6	-	-	-	Undefined
7	Off	00	0	Not used. Fixed to Off

[Notes] • This command is executed when the data is processed in the data buffer. Therefore, there may be a time lag between receiving the command and transmitting the status, depending on data buffer status.

[Default]

[Reference] \$10 \$04, \$1B \$76

[Example]

\$1D \$76 \$30

Devices: ALL

[Name] **Print raster bit image**

[Format] ASCII GS v 0 m xL xH yL yH d1...dk
 Hex 1D 76 30 m xL xH yL yH d1...dk
 Decimal 29 118 48 m xL xH yL yH d1...dk

[Range] $0 \leq m \leq 3$, $48 \leq m \leq 51$
 $0 \leq xL \leq 255$
 $0 \leq xH \leq 255$ ($1 \leq xL + xH \times 256 \leq 65535$)
 $0 \leq yL \leq 255$
 $0 \leq yH \leq 8$ ($1 \leq yL + yH \times 256 \leq 2047$)
 $0 \leq d \leq 255$
 $k = (xL + xH \times 256) + (yL + yH \times 256)$
 (except for $k = 0$)

[Description] Selects raster bit image mode. The value of m selects the mode as follows:

m	MODE
0,48	Normal
1, 49	Double width
2, 50	Double height
3, 51	Quadruple

- xL, xH selects the number of data bits ($xL+xH \times 256$) in the horizontal direction for the bit image.
- yL, yH selects the number of data bits ($yL+yH \times 256$) in the vertical direction for the bit image. k indicates no. of the image data. k is an explanation parameter; it is not necessary to be transmitted.
- d indicates the image data.

- [Notes]
- In standard mode for receipt paper, this command is effective only when there is no data in the print buffer.
 - The data (d) identify as 1 a printer bit and as 0 a non-printed bit.
 - If a raster bit image is longer than one line, the surplus data aren't printed.
 - This command has no effect in all print modes (character size, emphasized, double-strike, upside-down, underline, white/black reverse printing, etc.) for raster bit image.
 - This command feed the paper as much as necessary to print the bit image without using spacing set by \$1B \$32 or \$1B \$33.
 - Do not use this command during a macro executing because this command should not be included in a macro.
 - After the printing the printing starting position moves to the beginning of the line.
 - The following table shows the relationship between the downloaded bit image and the printed data:

d1	d2	...	dx
dX+1	dX+2	...	dX × 2
:	:	...	:
...	dk-2	dk-1	dk

[Reference]
 [Example]

\$1D \$77

Devices: ALL

[Name] **Set bar code width**

[Format] ASCII GS w n
 Hex 1D 77 n
 Decimal 29 119 n

[Range] $1 \leq n \leq 6$

[Description] Sets the horizontal size of the bar code. *n* specifies the bar code width as follows:

n	MODULE WIDTH (mm)
1	0.125
2	0.25
3	0.375
4	0.5
5	0.625
6	0.75

[Notes]

[Default] $n = 3$

[Reference] \$1D \$6B

[Example]

\$1D \$7C

Devices: ALL

[Name] **Set printing density**

[Format] ASCII GS { } n
 Hex 1D 7C n
 Decimal 29 124 n

[Range] $0 \leq n \leq 8, 48 \leq n \leq 56$

[Description] Sets printing density. *n* specifies printing density as follows:

n	PRINTING DENSITY
0, 48	- 50%
1, 49	- 37.5%
2, 50	- 25%
3, 51	- 12.5%
4, 52	0%
5, 53	+ 12.5%
6, 54	+ 25%
7, 55	+ 37.5%
8, 56	+ 50%

[Notes]

• Printing density reverts to the default value when the printer is reset or turned off.

[Default] $n = 4$

[Reference]

[Example]

\$1D \$E0

Devices: ALL

[Name] **Enable / disable automatic FULL STATUS back**

[Format] ASCII GS { } n
 Hex 1D E0 n
 Decimal 29 224 n

[Range] $0 \leq n \leq 255$

[Description] Enable / disable automatic full status back. n specifies the composition of FULL STATUS as follows :

**KPM202, KPM203, KPM302 (models without triple feeder), KPM303,
 TK202, TK203, TK302 (models without triple feeder), TK303**

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Disable paper status
	On	01	1	Enable paper status
1	Off	00	0	Disable user status
	On	02	2	Enable user status
2	Off	00	0	Disable Recoverable Error Status
	On	04	4	Enable Recoverable Error Status
3	Off	00	0	Disable Unrecoverable Error Status
	On	08	8	Enable Unrecoverable Error Status
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

**KPM302 (models with triple feeder),
 TK302 (models with triple feeder)**

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Disable paper status
	On	01	1	Enable paper status
1	Off	00	0	Disable user status
	On	02	2	Enable user status
2	Off	00	0	Disable Recoverable Error Status
	On	04	4	Enable Recoverable Error Status
3	Off	00	0	Disable Unrecoverable Error Status
	On	08	8	Enable Unrecoverable Error Status
4, 5, 6, 7	Off	00	0	Disable Triple Feeder Status
	On	F0	240	Enable Triple Feeder Status

[Notes]

**KPM202, KPM203, KPM302 (models without triple feeder), KPM303,
 TK202, TK203, TK302 (models without triple feeder), TK303**

- Once enable at least one byte of the FULL STATUS, for each change of at least one of the bits which compose the required status, the status sent in automatic from the printer will be so composed as follows:

1° Byte = 0x10 (DLE)

2° Byte = n

Next bytes (depends how many bits are active in n)

**KPM302 (models with triple feeder),
TK302 (models with triple feeder)**

- Once enable at least one byte of the FULL STATUS, for each change of at least one of the bits which compose the required status, the status sent in automatic from the printer will be so composed as follows:

1° Byte = 0x10 (DLE)

2° Byte = n

- The next 4 bytes depending on how many bits are active in the low nibble of n.

- The next 8 bytes depending on the value written in the high nibble of n (as showed in the table)

Note: The value of the bits within each byte is indicated for the \$10 \$04 26 command.

[Default]

[Reference]

\$10 \$04

[Example]

\$1D \$E1

Devices: ALL

[Name] **Reading of length paper (cm) available before virtual paper-end**

[Format]

ASCII	GS	{ }
Hex	1D	E1
Decimal	29	225

[Range]

[Description]

Reading of length (cm) paper available before virtual paper-end.

The command return a string pointing out how much paper is available, for example if there are 5.1 m before the paper end, it will be: '510cm'

[Notes]

- The length of residual paper reported is just as an indication because tolerances and other factors are not taken into consideration (paper thickness, roll core diameter, roll core thickness). The virtual paper-end limit is set by the command \$1D \$E6.

- To set virtual paper-end limit, measure the length of the paper from near paper end to the end of the roll, using several of them.

[Default]

[Reference]

\$1D \$E6

[Example]

\$1D \$E2

Devices:	KPM302
	KPM303
	TK302
	TK303

[Name] **Reading number of cuts performed from the printer**

[Format]

ASCII	GS	{ }
Hex	1D	E2
Decimal	29	226

[Range]

[Description] Reading the number of cuts performed from the printer.

[Notes] The command return a string that points out how many cuts are performed by the printer, for example if there are performed 2376 cuts, it will be: '2376 cuts'

[Default]

[Reference]

[Example]

\$1D \$E3

Devices:	ALL
----------	-----

[Name] **Reading of length (cm) of printed paper**

[Format]

ASCII	GS	{ }
Hex	1D	E3
Decimal	29	227

[Range]

[Description] Reading of length (cm) of printed paper.

[Notes] The command return a string pointing out how much paper is printed, for example if the printer has print about 2515,5 m, it will be: '251550cm'

[Default]

[Reference]

[Example]

\$1D \$E5

Devices: *ALL*

[Name] **Reading number of power up**
 [Format] ASCII GS { }
 Hex 1D E5
 Decimal 29 229

[Range]

[Description] Reading number of power up of the printer.
 [Notes] The command return a string pointing out the number of turning on of the printer, for example if the printer is turned on 512 times, it will be: '512on'

[Default]

[Reference]

[Example]

\$1D \$E6

Devices: *ALL*

[Name] **Virtual paper-end limit**
 [Format] ASCII GS { } nH nL
 Hex 1D E6 nH nL
 Decimal 29 230 nH nL

[Range] $0 \leq nH \leq 255$
 $0 \leq nL \leq 255$

[Description] This command sets the limit after which is pointed out the virtual paper-end.
 [Notes]

- The calculation limit of the near paper-end is in centimetres.
- This value is expressed as $[(nH \times 256) + nL]$

[Default] nH = 0x00
 nL = 0xF0

[Reference]

[Example] To see the virtual paper-end is pointed out after 15 metres from the first detection of near paper end, it's necessary convert 15 metres in 1500 centimetres and then, calculate nH and nL value in the following mode:

$$nH = 1500 / 256 = 5$$

$$nL = 1500 - (nH \times 256) = 1500 - (5 \times 256) = 220$$

and then send the following command:

HEX:	\$1D	\$E6	\$05	\$DC
DECIMAL:	29	230	5	220

\$1D \$E7

Devices: ALL

[Name] **Set notch distance**

[Format] ASCII GS { } nL nH
 Hex 1D E7 nL nH
 Decimal 29 231 nL nH

[Range] $0 \leq nH \leq 255$
 $0 \leq nL \leq 255$

[Description] Sets notch distance in tenth mm from the beginning of the document.

[Notes]

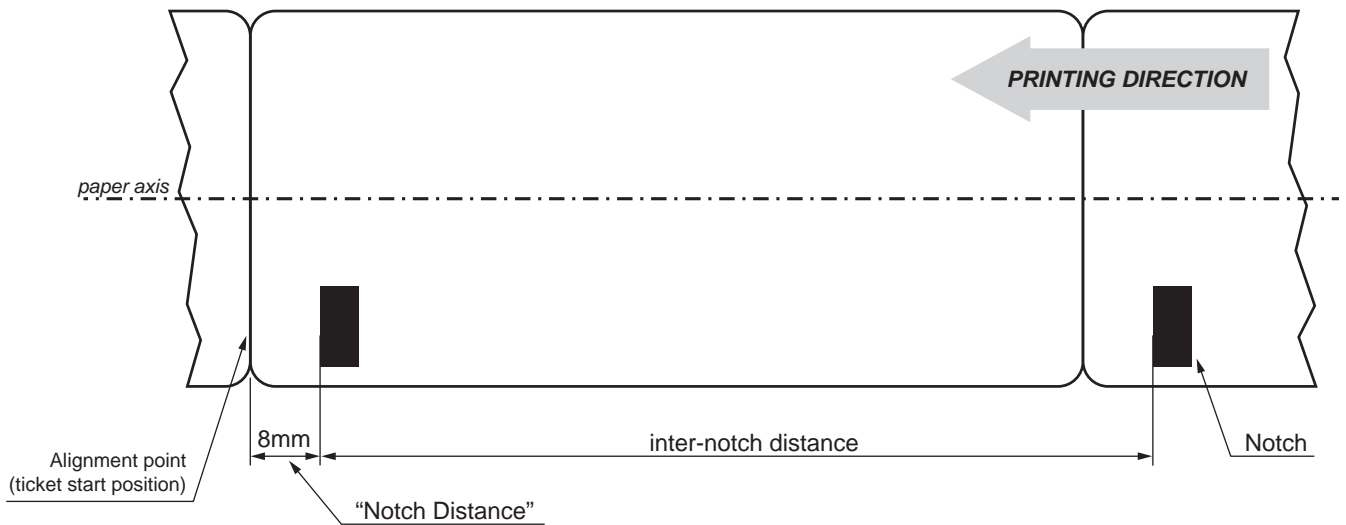
- This value is expressed as $[(nH \times 256) + nL]$
- The maximum value is 99.9 mm.
- The distance is saved in nonvolatile memory: it is therefore recommended not to send this command for each printed ticket, because the number of rewrites is limited. In many devices, however, is checked the diversity of the data before performing the rescue to avoid reaching the limit of rewrites.
- The distance defined by this command is the same that can be set with the value of the "Notch Distance" during the setup of the printer or by modifying the same parameter of the "Setup.ini" file (see User Manual for further explanation).

[Default] nH = \$00
 nL = \$00

[Reference]

[Example] To set a distance of the "Alignment Point" from the notch equal to 8mm = 80 tenths of a millimeter, send the command:
 \$1D \$E7 \$00 \$50

The following image shows a ticket with "Alignment Point" positioned at 8 mm from the notch.



\$1D \$F0

Devices: *ALL*

[Name] **Set printing speed**
 [Format] ASCII GS {} n
 Hex 1D F0 n
 Decimal 29 240 n
 [Range] $0 \leq n \leq 2$
 [Description] Sets printing speed. *n* specifies the printing speed as follows:

n	PRINTING SPEED
0	Alta qualità
1	Normale
2	Alta velocità

[Notes] • Printing speed reverts to the default value when the printer is reset or turned off.
 [Default] *n* = 1
 [Reference]
 [Example]

\$1D \$F6

Devices: *ALL*

Name] **Align the ticket**
 [Format] ASCII GS {}
 Hex 1D F6
 Decimal 29 246
 [Range]
 [Description] This command align the edge of the ticket to the alignment point set with \$1D \$E7 command as the notch distance. The printout will start at this position (see User Manual for further explanation).
 [Notes]
 [Default]
 [Reference] \$1D \$E7, \$1D \$F8
 [Example]

\$1D \$F8

Devices:	KPM302
	KPM303
	TK302
	TK303

[Name]	Align the ticket		
[Format]	ASCII	GS	{ }
	Hex	1D	F8
	Decimal	29	248
[Range]			
[Description]	This command align the edge of the ticket to the alignment point set with \$1D \$E7 command as the notch distance. The printout will start at this position (see User Manual for further explanation).		
[Notes]			
[Default]			
[Reference]	\$1D \$E7, \$1D \$F6		
[Example]			

3 SVELTA EMULATION

The following table lists all the commands for function management. The commands must be transmitted to the printer as command string enclosed between '<' character and '>' character.

COMMAND DESCRIPTION TABLE

Com. ASCII	Description
PRINT COMMANDS	
<DATE>	Print data
<p>	Printing command (cut and buffer cleaning) in reverse
<P>	Printing command (cut and buffer cleaning) in normal
<PP n, x, y, sp>	Print image in graphic page
<PR n, x, y, sp>	Print rotated image
<q>	Printing command (only buffer cleaning) in reverse
<Q>	Printing command (only buffer cleaning) in normal
<qn>	Printing command without alignment in reverse
<QN>	Printing command without alignment in normal
<TDF m Data>	Set user-defined date/time formats
<TIME>	Print time
CHARACTERS COMMAND	
<BS height, width>	Define area of the BOX mode
<F:bold>	Set bold mode
<F:clear>	Uninstall all TrueType fonts from printer
<F:draw:n>	Set drawing mode
<F:enc:ascii>	Set ASCII encoding
<F:enc:utf-8>	Set UTF-8 encoding
<F:enc:utf-16>	Set UTF-16 encoding
<F:err:n>	Get error
<F:filename.ttf>	Install new font
<F:italic>	Set italic mode
<F n>	Select the font
<F:regular>	Set regular mode
<F:rotate:aa>	Set font angle rotation
<F:size:nn>	Set font dimension
<HW height, width>	Set height and width of the current font
<NR>	Restore the text horizontal
<RL>	Rotate test 90° counter-clockwise
<RR>	Rotate test 90° clockwise
<RU>	Rotate test 180°
PRINT POSITION COMMANDS	
<LHTlength, width, notch, dimnotch>	Set the ticket dimension to print
<MM n>	Feed the paper of n step

SVELTA Emulation

<OXY x, y>	Set printing offset
<RC row, column>	Position the cursor
<T>	Get the ticket dimension to print
BIT-IMAGE COMMANDS	
<BF x1, y1, x2, y2>	Command to create filled BOX
<BV x1, y1, x2, y2>	Command to create empty BOX
<BX x1, y1, x2, y2, s, t>	Command to create parametric BOX
<CB>	Clear data in the print buffer
STATUS COMMAND	
<AFSB x>	Enable / Disable auto FULL STATUS back
<S n>	Status request
<SB x>	FULL STATUS request
BARCODE COMMANDS	
<B2D k, A, x>	Set the number of columns of two-dimensional barcode (PDF417)
<B2D k, B, x>	Set the number of rows of two-dimensional barcode (PDF417)
<B2D k, C, x>	Set the width of two-dimensional barcode (PDF417)
<B2D k, D, x>	Set the height of two-dimensional barcode (PDF417)
<B2D k, E, m, x>	Set the error correction level (PDF417)
<B2D k, P, x, d1...dn>	Store the two-dimensional barcode data in the barcode save area (PDF417)
<B2D I, A, x>	Set the height of DATAMATRIX barcode
<B2D I, B, x>	Set dot size (DATAMATRIX)
<B2D I, C, x>	Set barcode size (DATAMATRIX)
<B2D I, D, x>	Set rotation (DATAMATRIX)
<B2D I, P, x, d1...dn>	Store the two-dim. barcode data in the barcode save area (DATAMATRIX)
<B2D m, A, n>	Specify encoding scheme (AZTEC)
<B2D m, B, n>	Specify dot size (AZTEC)
<B2D m, C, n>	Specify size (AZTEC)
<B2D m, D, n>	Specify error correction level (AZTEC)
<B2D m, P, x, d0...dk>	Store the received data in the barcode save area (AZTEC)
<B2D n, A, n>	Specify encoding scheme (QRcode)
<B2D n, B, n>	Specify dot size (QRcode)
<B2D n, C, n>	Specify size (QRcode)
<B2D n, D, n>	Specify error correction level (QRcode)
<B2D n, P, x, d0...dk>	Store the received data in the barcode save area (QRcode)
<NCL x,y>	Print an horizontal code 128 barcode
<NCP x,y>	Print a vertical code 128 barcode
<NEL n>	Print horizontal EAN13 barcode
<NEP n>	Print a vertical EAN13 barcode
<NFL s>	Print horizontal ITF barcode
<NFP s>	Print a vertical ITF barcode
<NL s>	Print an horizontal code 39 barcode

<NP s>	Print a vertical code 39 barcode
<X n, M>	Define the barcode lines dimension
MISCELLANEOUS COMMANDS	
<bXnn>	Set the scan timeout of the barcode reader
	Return the scan timeout value of the barcode reader
<BC n>	Read a barcode
<BEEP 1, tt>	Emits a beep
<BMP>	Save a bitmap into flash disk
<BMPD>	Save a bitmap into SD/MMC card
<COM1>	Terminate the communication toward RFID
<COM2>	Select the communication toward RFID
<DT m>	Read date/time
<EPOS>	Change printer emulation to ESC/ POS
<INPUT n>	Load paper from triple feeder
<KEYS x>	Enable/Disable keys panel
<LOAD>	Reload paper
<SDT m Data>	Set date/time
<SVEL>	Change printer emulation to SVELTA
TICKET MANAGEMENT COMMANDS	
<BA n>	Change the ticket print intensity
<SP n>	Change speed
LOGOS MANAGEMENT COMMANDS	
<PC HexNumLogo HexXDim HexY-Dim HexTBD Id HexData>	Save the image in flash
<PE n>	Delete image
<PI n>	Get picture header info
<PL>	Get picture header list
<PN>	Get number of stored logo
COMMANDS FRO MECHANISM CONTROL	
<CUT>	Total cut
SELECTOR MANAGEMENT COMMANDS	
<EJOUT>	Perform the ticket ejection
<SELECTORI>	Initialize selector
<SELECTORO>	Set selector in "Open" position
<SELECTORS>	Set selector in "Storage" position

SVELTA Emulation

Given below are more detailed descriptions of each command.

<AFSB x>

Devices: ALL

[Name] **Enable / Disable auto FULL STATUS back**

[Format] ASCII <AFSB x >

[Range] **KPM202, KPM203, KPM302 (models without triple feeder), KPM303, TK202, TK203, TK302 (models without triple feeder), TK303**
 '0' ≤ x ≤ '9', 'A' ≤ x ≤ 'F'

KPM302 (models with triple feeder), TK302 (models with triple feeder)
 '0' ≤ x ≤ '9', 'A' ≤ x ≤ 'F' y = '0', y = 'F'

[Description]

- Enable/disable auto FULL STATUS back.
- x specify the request for FULL STATUS. where x identify the bitmask with the following table:

x	»	BIT3	BIT2	BIT1	BIT0
0	»	0	0	0	0
1	»	0	0	0	1
2	»	0	0	1	0
3	»	0	0	1	1
4	»	0	1	0	0
5	»	0	1	0	1
6	»	0	1	1	0
7	»	0	1	1	1
8	»	1	0	0	0
9	»	1	0	0	1
A	»	1	0	1	0
B	»	1	0	1	1
C	»	1	1	0	0
D	»	1	1	0	1
E	»	1	1	1	0
F	»	1	1	1	1

4° byte = Unrecoverable error status
 3° byte = Recoverable error status
 2° byte = User status
 1° byte = Full status

KPM302 (models with triple feeder), TK302 (models with triple feeder)

- y is an optional parameter; if y = 'F' then enables the transmission of the triple feeder status.

[Notes]

- Once enable at least one byte of the FULL STATUS, for each change of at least one of the bits which compose the required status, the status sent in automatic from the printer will be so composed as follows:

<SB x, CHR1 CHRn>

where:

- SB = fixed characters
- x = is the bitmask to identify the request.
- CHR1..CHRn = response bytes referred to the following tables:

all models

1° byte = Full status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper present
	On	01	1	Paper not present
1	-	-	-	RESERVED
2	Off	00	0	Paper present
	On	04	4	Near paper end
3	-	-	-	RESERVED
4	-	-	-	RESERVED
5	Off	00	0	Ticket not present in output
	On	20	32	Ticket present in output
6	Off	00	0	Not virtual paper end
	On	40	64	Virtual paper end
7	Off	00	0	The notch is placed over the sensor
	On	80	128	The notch is not placed over the sensor

KPM202, KPM203, KPM302 (models without selector), KPM303 (models without selector), TK202, TK203, TK302, TK303

2° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	-	-	-	RESERVED

KPM302 (models with selector), KPM303 (models with selector),

2° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	Off	00	0	Selector in "open" position
	On	80	128	Selector in "storage" position

all models

3° byte = Recoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Head temperature ok.
	On	01	1	Head temperature error
1	Off	00	0	No COM error
	On	02	2	RS232 COM error
2	-	-	-	RESERVED
3	Off	00	0	Power supply voltage ok
	On	08	8	Power supply voltage error
4	-	-	-	RESERVED
5	Off	00	0	Acknowledge command
	On	20	32	Not acknowledge command error
6	Off	00	0	Free paper path
	On	40	64	Paper jam
7	Off	00	0	Notch search ok
	On	80	128	Error in notch search

**KPM302 (models with triple feeder),
KPM302 (models without selector), KPM303 (models without selector),
TK302, TK303**

4° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Cutter ok
	On	01	1	Cutter error
1	Off	00	0	Cutter cover ok
	On	02	2	Cutter cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

KPM302 (models with selector), KPM303 (models with selector)

4° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Cutter ok
	On	01	1	Cutter error
1	Off	00	0	Cutter cover ok
	On	02	2	Cutter cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	Off	00	0	Selector OK
	On	80	128	Selector error

KPM202, KPM203, TK202, TK203

4° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	-	-	-	RESERVED
1	Off	00	0	Frontal cover ok
	On	02	2	Frontal cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

**KPM302 (models with triple feeder),
TK302(models with triple feeder)**

5° byte = 'I'

6° byte = Feeder sensors status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper sensor (Feeder 1 UP): paper not present
	On	01	1	Paper sensor (Feeder 1 UP): paper present
1	Off	00	0	Paper sensor (Feeder 2 CENTER): paper not present
	On	02	2	Paper sensor (Feeder 2 CENTER): paper present
2	Off	00	0	Paper sensor (Feeder 3 DOWN): paper not present
	On	04	4	Paper sensor (Feeder 3 DOWN): paper present
3	-	-	-	RESERVED
4	Off	00	0	Near paper end sensor (Feeder 1 UP): paper not present
	On	10	16	Near paper end sensor (Feeder 1 UP): paper present
5	Off	00	0	Near paper end sensor (Feeder 2 CENTER): paper not present
	On	20	32	Near paper end sensor (Feeder 2 CENTER): paper present
6	Off	00	0	Near paper end sensor (Feeder 3 DOWN): paper not present
	On	40	64	Near paper end sensor (Feeder 3 DOWN): paper present
7	-	-	-	RESERVED

7° byte = 'A'

8° Byte = FEEDER 1 (UP)

- = 0 No paper in feeder 1
- = 1 Paper in ACTIVE STATUS
- = '7' Paper end
- = 9 Error
- = 10 Paper in PARK STATUS

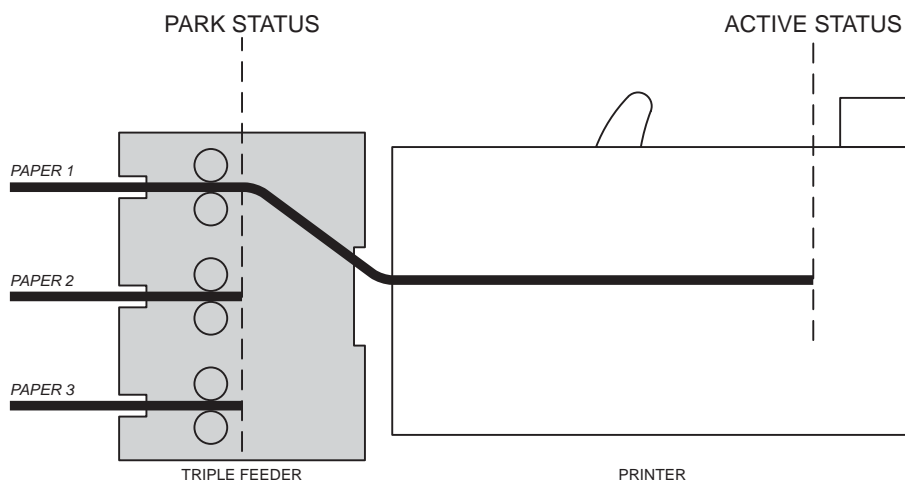
9° Byte = 'B'

10° Byte = FEEDER 2 (CENTER)

- = 0 No paper in feeder 2
- = 1 Paper in ACTIVE STATUS
- = '7' Paper end
- = 9 Error
- = 10 Paper in PARK STATUS

11° Byte = 'C'

- 12° Byte = FEEDER 3 (DOWN)
- = 0 No paper in feeder 3
 - = 1 Paper in ACTIVE STATUS
 - = '7' Paper end
 - = 9 Error
 - = 10 Paper in PARK STATUS



• The printer transmits bytes 1,2,3,4 and 6 as a pair of hexadecimal characters (between '0' and '9' or between 'A' and 'F'). For example the first byte is equal to \$A9, then will be sent from the printer the characters 'A' (\$41) and '9' (\$39).

[Default]
[Reference]
[Example]

To request the Full status (1° byte) and the User status (2°byte) proceed as follow:
see bitmask:

$BIT3 = 0$ $BIT2 = 0$ $BIT1 = 1$ $BIT0 = 1$ quindi 0011 = 3

Send the command: <AFSB3>

Possible answer: <SB3,0504>

where:

1°byte

0 = 0000	bit7 = 0 (notch found)	bit6 = 0 (not virtual paper end)	bit5 = 0 (ticket not present)	bit4 = 0 (RESERVED)
5 = 0101	bit3 = 0 (RESERVED)	bit2 = 1 (near paper end)	bit1 = 0 (RESERVED)	bit0 = 1 (Paper not present)

2°byte

0 = 0000	bit7 = 0 (RESERVED)	bit6 = 0 (FF key released)	bit5 = 0 (LF key released)	bit4 = 0 (RESERVED)
4 = 0100	bit3 = 0 (drag motor off)	bit2 = 1 (spooling)	bit1 = 0 (cover closed)	bit0 = 0 (print head down)

<bXnn>		
Devices:	KPM302	<i>(models with BARCODE reader)</i>
	KPM303	<i>(models with BARCODE reader)</i>
	TK302	<i>(models with BARCODE reader)</i>
	TK303	<i>(models with BARCODE reader)</i>

[Name]	Sets the scan timeout of the barcode reader	
[Format]	ASCII	<bXnn>
[Range]		
[Description]	<p>Sets the scan timeout of the barcode reader, using nn parameter value, expressed in tenth of second (10-1 second).</p> <p>If the X parameter value is equal to ASCII character 'e' (\$65) the nn value (the scan timeout) is stored in EEPROM. Otherwise its value is loaded into RAM so that it's possible to make different tests before save the correct value in EEPROM.</p>	
[Notes]		
[Default]	X = 3	
[Reference]		
[Example]		

		
Devices:	KPM302	<i>(models with BARCODE reader)</i>
	KPM303	<i>(models with BARCODE reader)</i>
	TK302	<i>(models with BARCODE reader)</i>
	TK303	<i>(models with BARCODE reader)</i>

[Name]	Return the scan timeout value of the barcode reader	
[Format]	ASCII	
[Range]		
[Description]	Returns the scan timeout value of the barcode reader.	
[Notes]		
[Default]		
[Reference]		
[Example]		

<B2D k, A, x>Devices: *ALL*

[Name]	Set the number of columns of two-dimensional barcode PDF417
[Format]	ASCII <B2D k, A, x>
[Range]	$0 \leq x \leq 30$
[Description]	Set the number of columns of PDF417 barcode. <ul style="list-style-type: none"> • $x = 0$ specifies auto processing • When x is not 0, specifies the number of columns of the data area as x code word.
[Notes]	<ul style="list-style-type: none"> • When auto processing ($x = 0$) is specified, the maximum number of columns in the data area is 30 columns.
[Default]	$x = 0$
[Reference]	
[Example]	

<B2D k, B, x>Devices: *ALL*

[Name]	Set the number of rows of two-dimensional PDF417 barcode
[Format]	ASCII <B2D k, B, x>
[Range]	$3 \leq x \leq 90$
[Description]	Set the number of rows of PDF417 barcode. <ul style="list-style-type: none"> • x specifies the number of rows of the data area as x rows.
[Notes]	
[Default]	
[Reference]	
[Example]	

<B2D k, C, x>

Devices: *ALL*

[Name] **Set the width of a module of two-dimensional barcode PDF417**
[Format] ASCII <B2D k, C, x>
[Range] $2 \leq x \leq 8$
[Description] Set the width of a module of PDF417 barcode.
[Notes]
[Default] $x = 3$
[Reference]
[Example]

<B2D k, D, x>

Devices: *ALL*

[Name] **Set the height of two-dimensional barcode PDF417**
[Format] ASCII <B2D k, D, x>
[Range] $2 \leq x \leq 8$
[Description] Set the height of PDF417 barcode.
[Notes]
[Default] $x = 3$
[Reference]
[Example]

<B2D k, E, m, x>

Devices: ALL

[Name] **Set the error correction level of the PDF417 barcode**

[Format] ASCII <B2D k, E, m, x>

[Range] m = 0, 1
 m = 0 0 ≤ x ≤ 8
 m = 1 1 ≤ x ≤ 40

[Description] Set the error correction level of PDF417 barcode.

- The error correction level is specified by "level" when m = 0.
- The error correction level is specified by "ratio" when m = 1 [x × 10%].
- Error correction level is specified by either "level" or "ratio".
- Error correction level specified by "level" (m = 0) is as follows. The number of the error correction code word is fixed regardless of the number of code words on the data area.

[Notes]

x	FUNCTION	N. OF ERROR CORRECTION CODE WORD
0	Error correction level 0	2
1	Error correction level 1	4
2	Error correction level 2	8
3	Error correction level 3	16
4	Error correction level 4	32
5	Error correction level 5	64
6	Error correction level 6	128
7	Error correction level 7	256
8	Error correction level 8	512

- Error correction level specified by "ratio" (m = 1) is as follows. The error correction level is defined by the calculated value [number of data code word × x × 0.1 = (A)]. The number of the error correction code word is changeable in proportion to the number of the code words on the data area.

CALCULATED VALUE (A)	CORRECTION LEVEL	N. OF ERROR CORRECTION CODE WORD
0 - 3	Error correction level 1	4
4 - 10	Error correction level 2	8
11 - 20	Error correction level 3	16
21 - 45	Error correction level 4	32
46 - 100	Error correction level 5	64
101 - 200	Error correction level 6	128
201 - 400	Error correction level 7	256
400 or more	Error correction level 8	512

- The error correction code word calculated by modulus 929.

[Default] m = 1, x = 1 [ratio: 10%]

[Reference]

[Example]

SVELTA Emulation

<B2D k, P, x, d1...dn>

Devices: *ALL*

[Name] **Store the two-dimensional PDF417 barcode data in the barcode save area**

[Format] ASCII <B2D k, P, x, d1...dn>

[Range]

[Description] Store the PDF417 barcode data (d1...dn) in the barcode save area.

- x = number of characters (= dn)

- d1...dn = barcode data

[Notes]

- n bytes of d1...dn are processed as barcode data.

- Specify only the data code word of the barcode with this function. Be sure not to include the control data in the data d1...dn because they are added automatically by the printer.

[Default]

[Reference]

[Example]

<B2D I, A, x>

Devices: *ALL*

[Name] **Specify the encoding scheme of DATAMATRIX barcode**

[Format] ASCII <B2D I, A, x>

[Range] $0 \leq x \leq 6$

[Description] Set the encoding scheme specified by x as follows:

x	ENCODING SCHEME
0	Ascii
1	C40
2	Text
3	X12
4	Edifact
5	Base256
6	AutoBest

[Notes]

[Default]

[Reference]

[Example]

<B2D I, B, x>

Devices: *ALL*

[Name] **Set dot size of the module of the DATAMATRIX barcode**
 [Format] ASCII <B2D I, B, x>
 [Range] $2 \leq x \leq 24$
 [Description] Set dot size of the module of DATAMATRIX barcode.
 x = dot dimension.
 [Notes]
 [Default] x = 6
 [Reference]
 [Example]

<B2D I, C, x>

Devices: *ALL*

[Name] **Set size of the DATAMATRIX barcode**
 [Format] ASCII <B2D I, C, x>
 [Range] $1 \leq x \leq 29$
 [Description] Set the size specified by x as follows:

x	BARCODE SIZE
1	10 x 10
2	12 x 12
3	14 x 14
4	16 x 16
5	18 x 18
6	20 x 20
7	22 x 22
8	24 x 24
8	26 x 26
10	32 x 32
11	36 x 36
12	40 x 40
13	44 x 44
14	48 x 48
15	52 x 52

x	BARCODE SIZE
16	64 x 64
17	72 x 72
18	80 x 80
19	88 x 88
20	96 x 96
21	104 x 104
22	120 x 120
23	132 x 132
24	144 x 144
25	8 x 18
26	8 x 32
27	12 x 26
28	12 x 36
29	16 x 36

[Notes]
 [Default] DmtxSymbolSquareAuto
 [Reference]
 [Example]

SVELTA Emulation

<B2D I, D, x>

Devices: *ALL*

[Name] **Set rotation of the DATAMATRIX barcode**

[Format] ASCII <B2D I, D, x>

[Range] x = 0, 1

[Description] Set rotation by x as follows:

n	ROTATION
0	No rotation
1	Rotation

[Notes]

[Default]

[Reference]

[Example]

<B2D I, P, x, d1...dn>

Devices: *ALL*

[Name] **Store the two-dimensional DATAMATRIX barcode data in the barcode save area**

[Format] ASCII <B2D I, P, x, d1...dn>

[Range]

[Description] Store the DATAMATRIX barcode data (d1...dn) in the barcode save area.

- x = number of characters (= dn)
- d1...dn = barcode data

[Notes]

- n bytes of d1...dn are processed as barcode data.
- Specify only the data code word of the barcode with this function. Be sure not to include the control data in the data d1...dn because they are added automatically by the printer.

[Default]

[Reference]

[Example]

<B2D m, A, n>Devices: *ALL*[Name] **Specify encoding scheme of AZTEC barcode**

[Format] ASCII <B2D m, A, n>

[Range] $0 \leq n \leq 1$

[Description] Specifies encoding type of AZTEC barcode.

n	ENCODING SCHEME
0	FULL AZTEC
1	AZTEC RUNE

[Notes]

- Full Aztec: Encode all extended ASCII characters data up to a maximum length of approximately 3823 numeric or 3067 alphabetic characters or 1914 bytes of data.
- Aztec Rune (Compact Aztec Code, sometimes called Small Aztec Code): Encode all numbers from 0 to 9 up to a maximum length of 3 numbers.

[Default] $n = 0$

[Reference]

[Example]

<B2D m, B, n>Devices: *ALL*[Name] **Specify dot size of the module of the AZTEC barcode**

[Format] ASCII <B2D, m, B, n>

[Range] $2 \leq n \leq 24$

[Description] Specifies numbers of dot for each pixel of AZTEC barcode.

[Notes]

[Default] $n = 0$

[Reference]

[Example]

SVELTA Emulation

<B2D m, C, n>

Devices: ALL

[Name] **Specify AZTEC barcode size**
[Format] ASCII <B2D m, C, n>
[Range] $0 \leq n \leq 36$
[Description] Specifies AZTEC barcode format (rows and columns), as follows:

n	FORMAT
0	AUTO
1	C15X15 Compact
2	C19X19 Compact
3	C23X23 Compact
4	C27X27 Compact
5	C19X19
6	C23X23
7	C27X27
8	C31X31
9	C37X37
10	C41X41
11	C45X45
12	C49X49

n	FORMAT
13	C53X53
14	C57X57
15	C61X61
16	C67X67
17	C71X71
18	C75X75
19	C79X79
20	C83X83
21	C87X87
22	C91X91
23	C95X95
24	C101X101
25	C105X105

n	FORMAT
26	C109X109
27	C113X113
28	C117X117
29	C121X121
30	C125X125
31	C131X131
32	C135X135
33	C139X139
34	C143X143
35	C147X147
36	C151X151

[Notes]
[Default] n = 0
[Reference]
[Example]

<B2D m, D, n>

Devices: *ALL*

[Name] **Specify the error correction level of the AZTEC barcode**
 [Format] ASCII <B2D m, D, n>
 [Range] $0 \leq n \leq 4$
 [Description] Specifies the ECC level (Error Correction Capacity) of AZTEC barcode.

n	ECC level
0	AUTO
1	> 10 % + 3 codewords
2	> 23 % + 3 codewords
3	> 36 % + 3 codewords
4	> 50 % + 3 codewords

- It is not possible to select both barcodesize and error correction capacity for the same barcode. If both options are selected then the error correction capacity selection will be ignored.

[Notes]
 [Default] n = 0
 [Reference]
 [Example]

<B2D m, P, n, d0...dk>

Devices: *ALL*

[Name] **Store and prints the AZTEC barcode data in the barcode save area**
 [Format] ASCII <B2D m, P, n, d0...dk> <P>
 [Range] n = n bytes of data
 [Description] Store the AZTEC barcode data (d0...dk) in the barcode save area.
 • k bytes of d0...dk are processed as barcode data.
 • Specify only the data code word of the barcode with this function.

[Notes]
 [Default]
 [Reference]
 [Example]

SVELTA Emulation

<B2D n, A, n>

Devices: *ALL*

[Name] **Specify encoding scheme of QRcode barcode**
[Format] ASCII <B2D n, A, n>
[Range] $0 \leq n \leq 1$
[Description] Specifies encoding type of AZTEC barcode.

n	ENCODING SCHEME
0	QRcode
1	MicroQR

[Notes]

- QRcode: Encode all extended ASCII characters data up to a maximum length of 7089 numeric digits, 4296 alphabetic characters or 2953 bytes of data.
- MicroQR (a miniature version of the QRcode barcode for short message): Encode all numbers from 0 to 9 up to a maximum length of 35 characters.

[Default] n = 0

[Reference]

[Example]

<B2D n, B, n>

Devices: *ALL*

[Name] **Specify dot size of the module of the QRcode barcode**
[Format] ASCII <B2D, n, B, n>
[Range] $2 \leq n \leq 24$
[Description] Specifies numbers of dot for each pixel of the module of the QRcode barcode.

[Notes]

[Default] n = 0

[Reference]

[Example]

<B2D n, C, n>

Devices: *ALL*

[Name] **Specify QRcode barcode size**
 [Format] ASCII <B2D n, C, n>
 [Range] $0 \leq n \leq 40$
 [Description] Specifies QRcode barcode format (rows and columns), as follows:

n	VERSION	n	VERSION	n	VERSION
0	AUTO	14	V14	28	V28
1	V1	15	V15	29	V29
2	V2	16	V16	30	V30
3	V3	17	V17	31	V31
4	V4	18	V18	32	V32
5	V5	19	V19	33	V33
6	V6	20	V20	34	V34
7	V7	21	V21	35	V35
8	V8	22	V22	36	V36
9	V9	23	V23	37	V37
10	V10	24	V24	38	V38
11	V11	25	V25	39	V39
12	V12	26	V26	40	V40
13	V13	27	V27		

[Notes]
 [Default] n = 0
 [Reference]
 [Example]

<B2D n, D, n>

Devices: *ALL*

[Name] **Specify the error correction level of the QRcode barcode**
 [Format] ASCII <B2D n, D, n>
 [Range] $0 \leq n \leq 4$
 [Description] Specifies the ECC level (Error Correction Capacity) of QRcode barcode.

n	ECC level	
0	AUTO	
1	ECC = approx 20% of barcode	Recovery Capacity = approx 7%
2	ECC = approx 37% of barcode	Recovery Capacity = approx 15%
3	ECC = approx 50% of barcode	Recovery Capacity = approx 25%
4	ECC = approx 65% of barcode	Recovery Capacity = approx 30%

[Notes]
 [Default] n = 0
 [Reference]
 [Example]

<B2D n, P, n, d0...dk>

Devices: *ALL*

[Name] **Store and prints the QRcode barcode data in the barcode save area**

[Format] ASCII <B2D n, P, n, d0...dk> <P>

[Range] n = n bytes of data

[Description] Store the QRcode barcode data (d0...dk) in the barcode save area.

- k bytes of d0...dk are processed as barcode data.
- Specify only the data code word of the barcode with this function.

[Notes]

[Default]

[Reference]

[Example]

<BA> n

Devices: *ALL*

[Name] **Change the ticket print intensity**

[Format] ASCII <BA n>

[Range]

[Description] Changes the ticket print intensity where n indicates the print mode. The possible values of n are as follows :

n	PRINT MODE
0	Black/white printing at 100% of maximum intensity
8	Black/white printing at 50% of maximum intensity
16	Black/white printing at 25% of maximum intensity
24	Black/white printing at 12% of maximum intensity
32	Black/white printing at 7% of maximum intensity
40	Black/white printing at 5% of maximum intensity

[Notes]

[Default]

[Reference]

[Example]

<BC n>		
Devices:	KPM302	(models with BARCODE reader)
	KPM303	(models with BARCODE reader)
	TK302	(models with BARCODE reader)
	TK303	(models with BARCODE reader)

[Name]	Read a BarCode
[Format]	ASCII <BC n>
[Range]	n = 0, 1, A, C, T, S
[Description]	<ul style="list-style-type: none"> • <u>With n = 0 the scan command is sent and the returned string is:</u> <BC0 ↵ x barcode ↵ > where <ul style="list-style-type: none"> - ↵ corresponds to CR character (\$0D). - x indicate the reading result ; the x value can be : <ul style="list-style-type: none"> '!' : the barcode is read '#' : the barcode is not correctly read - barcode is the barcode's characters read • <u>With n = 1 the returned string is :</u> <BC1 ↵ x barcode ↵ > where <ul style="list-style-type: none"> - barcode is the last barcode read through the printing commands '<p>', '<P>', '<q>', '<Q>'. • <u>With n = A returns the last barcodes read up to ten as maximum;the returned string is:</u> <BCA ↵ x barcode1 ↵ x barcode2 ↵ ... x barcode n ↵ > where <ul style="list-style-type: none"> - ↵ corresponds to CR character (\$0D). - x indicate the reading result ; the x value can be :↵ <ul style="list-style-type: none"> '!' : the barcode is read '#' : the barcode is not correctly read - barcode is the barcode's characters read • <u>With n = C the returned string is:</u> <BCC ↵ x barcode ↵ > where <ul style="list-style-type: none"> - ↵ corresponds to CR character (\$0D). - x indicate the reading result; the x value can be : <ul style="list-style-type: none"> '!' : the barcode is read '#' : the barcode is not correctly read - barcode is the barcode's characters read • <u>With n = S returns the barcode reader status; the returned string is:</u> <BCS x> where <ul style="list-style-type: none"> - x indicate the barcode reader status; the x value can be: <ul style="list-style-type: none"> '!' : the barcode reader is on '#' : the barcode reader is off • <u>With n = T enable/disable barcode reader to reading</u>
[Notes]	<ul style="list-style-type: none"> • The barcode read through the printing commands '<p>', '<P>', '<q>', '<Q>'.
[Default]	
[Reference]	
[Example]	

<BEEP 1, tt>

Devices: *ALL*

[Name] **Emits a beeb**

[Format] ASCII <BEEP1, tt>

[Range]

[Description] When this command is received, the printer emits a beeb as acoustic signalling.
tt is the beep time in milliseconds.

[Notes]

[Default]

[Reference]

[Example]

<BF x1 y1, x2, y2>

Devices: *ALL*

[Name] **Command to create filled Box**

[Format] ASCII <BF x1,y1,x2,y2>

[Range]

[Description] Create a filled box on the basis of x1, y1, x2, y2 coordinates where :

x1 -> minimum horizontal coordinate

y1 -> minimum vertical coordinate

x2 -> maximum horizontal coordinate

y2 -> maximum vertical coordinate

[Notes]

- If the coordinates are reversed, the printer automatically turns the points to create in any case the box.

- If the x2 is greater than the maximum horizontal width of graphic page, the box is drawn using the maximum width as last point.

- If the y2 is greater than the maximum length of graphic page defined by <LHT...> command, the box is drawn using the maximum length (defined by this command) as last point.

[Default]

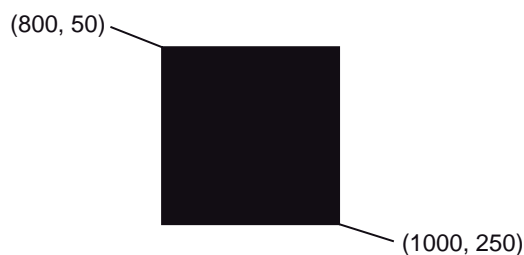
[Reference] <OXY x, y>

[Example] Ticket example that use a filled box

```
<CB><BA8>
```

```
<BF800,50,1000,250>
```

```
<q>
```



<BMP>

Devices: ALL

[Name] **Save a bitmap into flash disk**

[Format] ASCII <BMP>

[Range]

[Description] When this command is received, a bitmap with an image of the printing ticket is saved into "Prt-Ticket" folder on flash disk.

[Notes] The bitmap file name consists of data and time of ticket print.

[Default]

[Reference]

[Example]

<BMPD>

Devices: KPM302

KPM303

TK302

TK303

[Name] **Save a bitmap into SD/MMC card**

[Format] ASCII <BMPD>

[Range]

[Description] When this command is received, a bitmap with an image of the printing ticket is saved into "Prt-Ticket" folder on multimedia card.

[Notes] The bitmap file name consists of data and time of ticket print.

[Notes]

[Default]

[Reference]

[Example]

<BS height, width>

Devices: *ALL*

[Name] **Define area for the box mode**

[Format] ASCII <BS height, width>

[Range]

[Description] Defines the area where position a character. If the box dimensions are bigger than the font, then the empty spaces are filled with white spaces, whereas if the box dimensions are smaller than the font, then the font is cutted.

[Notes]

- To disable the Box Size set height and width parameters to 0 (<BS0,0>).
- This command is not active with TrueType fonts.

[Default]

[Reference]

[Example]

<BV x1, y1, x2, y2>

Devices: *ALL*

[Name] **Command to create empty Box**

[Format] ASCII <BF x1,y1,x2,y2>

[Range]

[Description] Create an empty box on the basis of x1, y1, x2, y2 coordinates where :
x1 -> minimum horizontal coordinate
y1 -> minimum vertical coordinate
x2 -> maximum horizontal coordinate
y2 -> maximum vertical coordinate

[Notes]

- The box border is fixed to 1mm (8 dots)
- If the coordinates are reversed, the printer automatically turns the points to create in any case the box.
- If the x2 is greater than the maximum horizontal width of graphic page, the box is drawn using the maximum width as last point.
- If the y2 is greater than the maximum length of graphic page defined by <LHT...> command, the box is drawn using the maximum length (defined by this command) as last point.

KPM203, KPM303, TK203, TK303

- The box border is fixed to 1mm (12 dots)

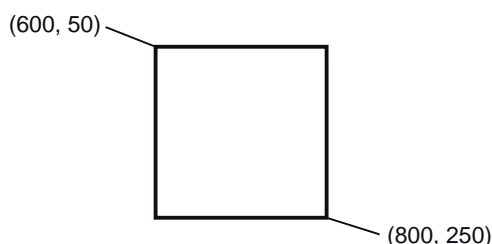
[Default]

[Reference] <OXY x, y>

[Example] Ticket example that use an empty box

<CB><BA8>

<BV600,50,800,250>



<BX x1, y1, x2, y2, s, t>

Devices: ALL

[Name] **Command to create parametric Box**
 [Format] ASCII <BX x1,y1,x2,y2, s, t >
 [Range]

[Description] Create a box defined by the following parameters where :
 x1 -> minimum horizontal coordinate
 y1 -> minimum vertical coordinate
 x2 -> maximum horizontal coordinate
 y2 -> maximum vertical coordinate
 s -> border thickness in dot (8 dot = 1mm)s ≤ 255
 t -> Fill mode 0 ≤ t ≤ 9

KPM203, KPM303, TK203, TK303
 s -> border thickness in dot (12 dot = 1mm)s ≤ 255

t	FILL MODE
0	Deletes area
1	Fills area
2..8	Fills area with specific pattern
9	The area leaves unchanged (only for rectangle border)

- [Notes]
- If t > 9 the fill mode is set to 9
 - If the coordinates are reversed, the printer automatically turns the points to create in any case the box.
 - If the x2 is greater than the maximum horizontal width of graphic page, the box is drawn using the maximum width as last point.
 - If the y2 is greater than the maximum length of graphic page defined by <LHT...> command, the box is drawn using the maximum length (defined by this command) as last point.
 - If the defined thickness is greater than the half of box width, then the thickness is set to the half of box width to print (filled box).
 - This command is not active with TrueType fonts.

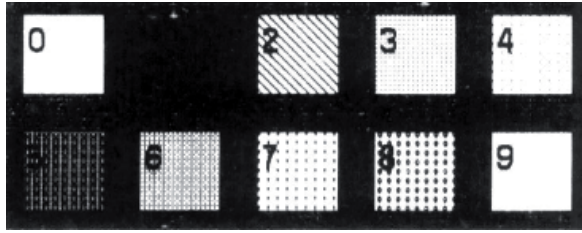
[Default]
 [Reference] <OXY x, y>

[Example]

Command sequence to generate a demo ticket with differents kinds of box

```
<CB><BA8><BS0,0>
<NR>
<BX200,100,300,200,16,0><RC120,220><F3><HW1,1>0
<BX300,100,400,200,16,1><RC120,320><F3><HW1,1>1
<BX400,100,500,200,16,2><RC120,420><F3><HW1,1>2
<BX500,100,600,200,16,3><RC120,520><F3><HW1,1>3
<BX600,100,700,200,16,4><RC120,620><F3><HW1,1>4
<BX200,200,300,300,16,5><RC220,220><F3><HW1,1>5
<BX300,200,400,300,16,6><RC220,320><F3><HW1,1>6
<BX400,200,500,300,16,7><RC220,420><F3><HW1,1>7
<BX500,200,600,300,16,8><RC220,520><F3><HW1,1>8
<BX600,200,700,300,16,9><RC220,620><F3><HW1,1>9
<q>
```

Example of what will be printed on ticket



<CB>

Devices: *ALL*

[Name] **Clear data in the print buffer**
[Format] ASCII <CB>
[Range]
[Description] Clear data in the print buffer, move the cursor to column 0, row 0, resets the text rotation, set the default font as current and disables the Box Size function during the character writing.

[Notes]
[Default]
[Reference]
[Example]

<COM1>

Devices:	KPM302	<i>(models with RFID board)</i>
	KPM303	<i>(models with RFID board)</i>
	TK302	<i>(models with RFID board)</i>
	TK303	<i>(models with RFID board)</i>

[Name]	Terminate the communication toward RFID module
[Format]	ASCII <COM1>
[Range]	
[Description]	Terminates the communication toward RFID module.
[Notes]	
[Default]	
[Reference]	
[Example]	

<COM2>

Devices:	KPM302	<i>(models with RFID board)</i>
	KPM303	<i>(models with RFID board)</i>
	TK302	<i>(models with RFID board)</i>
	TK303	<i>(models with RFID board)</i>

[Name]	Select the communication toward RFID module
[Format]	ASCII <COM2>
[Range]	
[Description]	Set the communication toward RFID module.
[Notes]	
[Default]	
[Reference]	
[Example]	

<CUT>

Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>

[Name]	Total cut
[Format]	ASCII <CUT>
[Range]	
[Description]	This command enables cutter operation. If there is no cutter, a disabling flag is set and any subsequent cut commands will be ignored.
[Notes]	<ul style="list-style-type: none">• The printer waits to complete all paper movement commands before it executes a total cut.
[Default]	
[Reference]	
[Example]	

<DATE>

Devices:	<i>ALL</i>
----------	------------

[Name]	Print date
[Format]	ASCII <DATE>
[Range]	
[Description]	Prints date in the format specified by the command '<TDF>'. [Notes]
[Default]	"dd/mm/yy"
[Reference]	<TIME>
[Example]	

<DT m>

Devices: ALL

[Name] **Read date/time**
 [Format] ASCII <DT m>
 [Range]
 [Description] Read date/time of the real time clock and send it, in the format specified by m values as follows:

m	FORMAT
0	DD/MM/YY hh:mm:ss
1	DDMMYYhhmmss
2	YYMMDDhhmmss
3	YYMMDDhhmmssd

where :

- DD = represents the day of the date
- MM = represents the month of the date
- YY = represents year of the date
- hh = represents the hour of the time
- mm = represents the minutes of the time
- ss = represents the seconds of the time
- d = indicates the day of the week

The printer's answer will be :

<DT ↵ x data↵ >

where

- ↵ corresponds to CR character (\$0D).
- x indicate the reading result ; the x value can be :
 - '!' : the command is executed successfully
 - '#' : the command is not executed successfully
- data are the ASCII characters that represent the date/time.

[Notes]
 [Default]
 [Reference]
 [Example]

To read date/time in the "DDMMYYhhmmss" format, transmit: <DT 1>
 For example if the current date/time are "15 September 2006 at 10:56:20 (AM)" the printer's answer is as follows :
 <DT ↵ ! 151006105620 ↵ > if the transmission is successfully, otherwise
 <DT ↵ # ↵ > if the transmission is not successful

SVELTA Emulation

<EJOUT>

Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>

[Name]	Perform ticket ejection
[Format]	ASCII <EJOUT>
[Range]	
[Description]	This command performs the ejection of the printed ticket.
[Notes]	
[Default]	
[Reference]	
[Example]	

<EPOS>

Devices:	<i>ALL</i>
----------	------------

[Name]	Change printer emulation to ESC/ POS
[Format]	ASCII <EPOS>
[Range]	
[Description]	Set the ESC/ POS emulation.
[Notes]	
[Default]	
[Reference]	
[Example]	

<F:bold>Devices: *ALL*

[Name] **Set bold mode**
 [Format] ASCII <F:bold>
 [Range]
 [Description] Set the bold printing mode
 [Notes] This command is active only with TrueType fonts.
 [Default]
 [Reference]
 [Example]

<F:clear>Devices: *ALL*

[Name] **Uninstall all TrueType fonts from printer**
 [Format] ASCII <F:clear>
 [Range]
 [Description] Clear the installation memory by uninstalling TrueType fonts
 [Notes]

- This command is active only with TrueType fonts.
- Use <F:err:n> command to verify the outcome of this command.

 [Default]
 [Reference] <F:err:n>
 [Example]

<F:draw:n>Devices: *ALL*

[Name] **Set drawing mode**
 [Format] ASCII <F:draw:n>
 [Range] n = '0', '1', '2'
 [Description] Set drawing mode functioning with following n values:

n = '0'	OR mode
n = '1'	XOR mode
n = '2'	AND mode

 [Notes] This command is active only with TrueType fonts.
 [Default] n = '0'
 [Reference]
 [Example]

<F:enc:ascii>

Devices: *ALL*

[Name] **Set ASCII encoding**
[Format] ASCII <F:enc:ascii>
[Range]
[Description] Set default encoding (ASCII) for TrueType fonts
[Notes] This command is active only with TrueType fonts.
[Default]
[Reference]
[Example]

<F:enc:utf-8>

Devices: *ALL*

[Name] **Set UTF-8 encoding**
[Format] ASCII <F:enc:utf-8>
[Range]
[Description] Set UTF-8 encoding for TrueType fonts
[Notes]

- This command is active only with TrueType fonts.
- The character's addressing respects the UNICODE™ standard (see www.unicode.org).

[Default]
[Reference]
[Example]

<F:enc:utf-16>

Devices: *ALL*

[Name] **Set UTF-16 encoding**
[Format] ASCII <F:enc:utf-16>
[Range]
[Description] Set UTF-16 encoding for TrueType fonts
[Notes]

- This command is active only with TrueType fonts.
- The character's addressing respects the UNICODE™ standard (see www.unicode.org).

[Default]
[Reference]
[Example]

<F:err:n>Devices: *ALL*[Name] **Get error**

[Format] ASCII <F:err:n>

[Range] n = '0', '1'

[Description] Get the last error functioning with n, where
 n = '0' Get last error
 n = '1' Get last error + internal error code

[Notes]

- Use this command to know if an error occurs during the execution of commands for TrueType fonts management (as <F:filename.ttf> or <F:clear>).
- To know the internal error codes list, contact Customer Service.
- This command is active only with TrueType fonts.

[Default]

[Reference]

[Example]

<F:filename.ttf>Devices: *ALL*[Name] **Install new font**

[Format] ASCII <F:filename.ttf>

[Range]

[Description] Install a new TrueType font.

[Notes]

- This command is active only with TrueType fonts.
- Use <F:err:n> command to verify the outcome of this command.

[Default]

[Reference] <F:err:n>

[Example]

<F:italic>Devices: *ALL*[Name] **Set italic mode**

[Format] ASCII <F:italic>

[Range]

[Description] Set the italic printing mode

[Notes] This command is active only with TrueType fonts.

[Default]

[Reference]

[Example]

<F n>

Devices: *ALL*

[Name] **Select the font**
[Format] ASCII <F n>
[Range]
[Description] Selects the current font where n indicates the font to use.
[Notes]
[Default]
[Reference]
[Example]

<F:regular>

Devices: *ALL*

[Name] **Set regular mode**
[Format] ASCII <F:regular>
[Range]
[Description] Set the regular printing mode
[Notes] This command is active only with TrueType fonts.
[Default]
[Reference]
[Example]

<F:rotate:aa>

Devices: *ALL*

[Name] **Set rotation angle for TrueType font**
[Format] ASCII <F:rotate:aa>
[Range] $0 \leq aa \leq 360$
[Description] Set rotation angle for TrueType font, functioning with aa.
[Notes]

- This command is active only with TrueType fonts.
- For TrueType fonts, it is also possible to use the commands for standard angles of rotation (<NR>, <RR>, <RL>, <RU>).

[Default] aa = 0
[Reference] <NR>, <RR>, <RL>, <RU>
[Example]

<F:size:nn>Devices: *ALL*

[Name] **Set font dimension**
 [Format] ASCII <F:size:nn>
 [Range]
 [Description] Set font dimension functioning with n.
 [Notes]

- The size is not expressed in pixels but in points
- This command is active only with TrueType fonts.

 [Default] 10 points
 [Reference]
 [Example]

<HW height, width>Devices: *ALL*

[Name] **Set height and width of the current font**
 [Format] ASCII <HW height, width>
 [Range]
 [Description] Modifies the height and width of the current font where height and width are the multiplier coefficients of height and width of how enlarge the font. Both values can be:

1:	Font dimension ×1
2:	Font dimension ×2
3:	Font dimension ×3
4:	Font dimension ×4
5:	Font dimension ×5
6:	Font dimension ×6
7:	Font dimension ×7
8:	Font dimension ×8

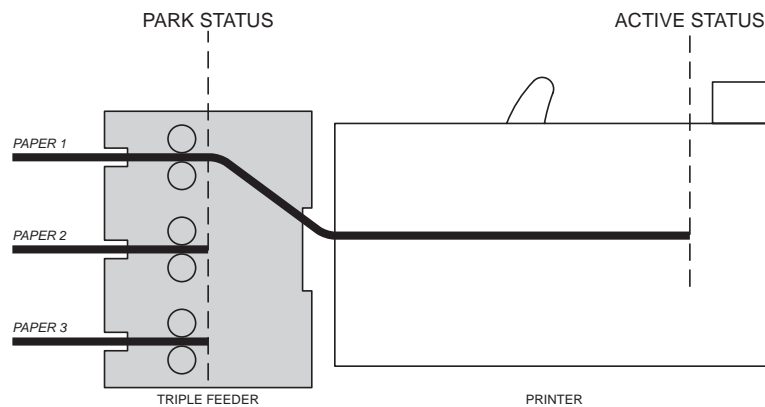
[Notes] The command is ignored if height or width has different value from that reported above.
 [Default]
 [Reference]
 [Example]

SVELTA Emulation

<INPUT n>

Devices:	KPM302	(models with triple feeder)
	TK302	(models with triple feeder)

[Name]	Load paper from triple feeder (1, 2, 3)
[Format]	ASCII <INPUT n>
[Range]	n= A, B, C
[Description]	Load paper inside the printer based on the following values of n : n= A ingresso carta 1 n= B ingresso carta 2 n= C ingresso carta 3
[Notes]	• If another paper is in ACTIVE STATUS is retracted to PARK STATUS.



[Default]
[Reference]
[Example]

<KEYS x>

Devices:	ALL
----------	-----

[Name]	Enable/Disable keys panel
[Format]	ASCII <KEYS x>
[Range]	x = 0, 1
[Description]	Enables / disables the keys panel. <ul style="list-style-type: none">• When x = 0, the keys panel is disabled.• When x = 1, the keys panel is enabled.
[Notes]	• When the keys panel is disabled, the keys may only be used after the printer has been reset.
[Default]	x = 1
[Reference]	
[Example]	

<LHT length, height, notch, dimnotch>

Devices: ALL

[Name] **Set ticket dimension to print**
 [Format] ASCII <LHT *length, height, notch, dimnotch*>

[Range]
 [Description] Sets the ticket dimension to print in the following mode:
length is the ticket length (in dot);
height is the ticket height (in dot);
notch is the distance (in dot) between the ticket upper edge and strobe backside preprinted black mark;
dimnotch is the notch dimension (in dot).

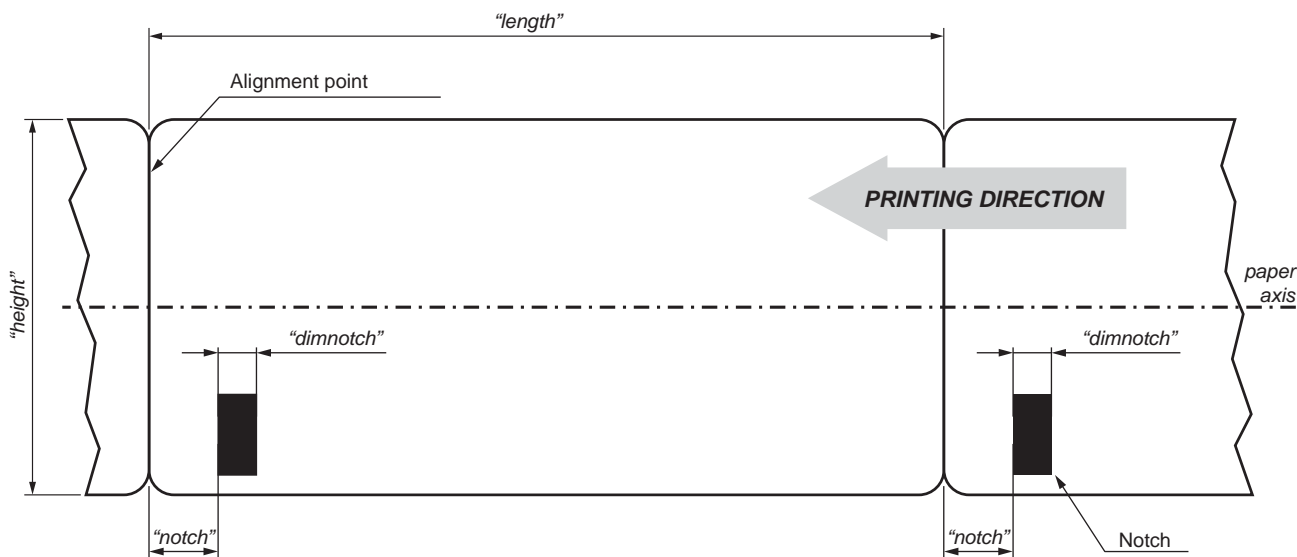
[Notes]

- If using the point (.) character as decimal separator instead of commas then the passed value are stored in nonvolatile memory.
- The parameters are saved in nonvolatile memory: it is therefore recommended not to send this command for each printed ticket, because the number of rewrites is limited. In many devices, however, is checked the diversity of the data before performing the rescue to avoid reaching the limit of rewrites.
- The parameters defined by this command are the same that can be set by modifying the same parameters of the "Setup.ini" file (see User Manual for further explanation).
- 1mm = 8 dot.

KPM203, KPM303, TK203, TK303
 • 1mm = 12 dot.

[Default]
 [Reference]
 [Example]

The following image shows a ticket with the parameters set by this command:



<LOAD>

Devices: *ALL*

[Name] **Reload paper**

[Format] ASCII <LOAD>

[Range]

[Description] When this command is received, the printer performs a paper reloading.

[Notes] During the execution of the command, the printer indicates the paper end

[Default]

[Reference]

[Example]

<MM n>

Devices: *ALL*

[Name] **Feed the paper of n step**

[Format] ASCII <MM n>

[Range]

[Description] When this command is received, the paper feed of n STEP.

[Notes] 1 STEP = 0,125 mm (1/8 mm)

[Default]

[Reference]

[Example]

<NCL x,y>Data

Devices: ALL

[Name] **Print horizontal CODE 128 barcode**

[Format] ASCII <NCL x, y>Data

[Range]

[Description] Print a CODE 128 barcode type in horizontal, where:
 x = barcode height in millimetres;
 y = byte number of the string to encode.

[Notes] • The top part of the bar code data string must be a code set selection character (CODE A, CODE B or CODE C) which selects the first code set.
 • Special characters are defined by combining two characters “{” and one character. ASCII character “{” is defined by transmitting “{{” twice, consecutively.

SPECIFIC CHARACTER	DATA TRANSMISSION		
	ASCII	HEX	Decimal
SHIFT	{S	7B, 53	123, 83
CODE A	{A	7B, 41	123, 65
CODE B	{B	7B, 42	123, 66
CODE C	{C	7B, 43	123, 67
FNC1	{1	7B, 31	123, 49
FNC2	{2	7B, 32	123, 50
FNC3	{3	7B, 33	123, 51
FNC4	{4	7B, 34	123, 52
{	{{	7B, 7B	123, 123

[Default]

[Reference]

[Example]

code A : <RC10,300><NCL 15,9>{A3456789
 code B: <RC10,300><NCL 15,9>{B3456789
 code C : <RC10,300><NCL 15,9>{C3456789

<NCP x,y>Data

Devices: *ALL*

[Name] **Print vertical CODE 128 barcode**

[Format] ASCII <NCP x, y>Data

[Range]
[Description]

Print a CODE 128 barcode type in vertical, where:
x = barcode height in millimetres;
y = byte number of the string to encode.

[Notes]

- The top part of the bar code data string must be a code set selection character (CODE A, CODE B or CODE C) which selects the first code set.
- Special characters are defined by combining two characters “{” and one character. ASCII character “{” is defined by transmitting “{{” twice, consecutively.

SPECIFIC CHARACTER	DATA TRANSMISSION		
	ASCII	HEX	Decimal
SHIFT	{S	7B, 53	123, 83
CODE A	{A	7B, 41	123, 65
CODE B	{B	7B, 42	123, 66
CODE C	{C	7B, 43	123, 67
FNC1	{1	7B, 31	123, 49
FNC2	{2	7B, 32	123, 50
FNC3	{3	7B, 33	123, 51
FNC4	{4	7B, 34	123, 52
{	{{	7B, 7B	123, 123

[Default]

[Reference]

[Example]

code A : <RC10,300><NCP15,9>{A3456789
code B : <RC10,300><NCP15,9>{B3456789
code C : <RC10,300><NCP15,9>{C3456789

<NEL n> *Data*Devices: *ALL*[Name] **Print horizontal EAN13 barcode**

[Format] ASCII <NEL n> *Data*

[Range]

[Description] Print an EAN13 barcode type in horizontal.
 The n parameter indicates the barcode height in millimetres.
 The Data parameter contains the data to convert, with start and stop characters of barcode.
 The "*" star character is the start and the stop character of the barcode.

[Notes]

[Default]

[Reference]

[Example]

<X2,L>

<RC220,20><NEL 10>*123456789012*

<NEP n> *Data*Devices: *ALL*[Name] **Print vertical EAN13 barcode**

[Format] ASCII <NEP n>*Data*

[Range]

[Description] Print an EAN13 barcode type in vertical.
 The n parameter indicates the barcode height in millimetres.
 The Data parameter contains the data to convert, with start and stop characters of barcode.
 The "*" star character is the start and the stop character of the barcode.

[Notes]

[Default]

[Reference]

[Example]

<X2,L>

<RC20,10><NEP10>*123456789012*

<NFL s> *Data*

Devices: *ALL*

[Name] **Print horizontal ITF barcode**
[Format] ASCII <NFL s> *Data*
[Range]
[Description] Print an ITF barcode type in horizontal. The s parameter indicates the barcode height in millimetres. The Data parameter contains the data to convert, with start and stop characters of barcode.
[Notes] The "*" star character is the start and the stop character of the barcode.
[Default]
[Reference]
[Example] <X2,L>
<RC220,20><NFL10>*123456*

<NFP s> *Data*

Devices: *ALL*

[Name] **Print vertical ITF barcode**
[Format] ASCII <NFP s>*Data*
[Range]
[Description] Print an ITF barcode type in vertical. The s parameter indicates the barcode height in millimetres. The Data parameter contains the data to convert, with start and stop characters of barcode.
[Notes] The "*" star character is the start and the stop character of the barcode.
[Default]
[Reference]
[Example] <X2,L>
<RC20,10><NFP10>*123456*

<NL s> *Data*Devices: *ALL*

[Name] **Print an horizontal CODE 39 barcode**
 [Format] ASCII <NL s>*Data*
 [Range]
 [Description] Print a CODE 39 barcode type in horizontal. The s parameter indicates the barcode height in millimetres. The Data parameter contains the data to convert, with start and stop characters of barcode.
 [Notes] The "*" star character is the start and the stop character of the barcode.
 [Default]
 [Reference]
 [Example] <X2,L>
 <RC220,120><NL10>*123456*

<NP s> *Data*Devices: *ALL*

[Name] **Print a vertical CODE 39 barcode**
 [Format] ASCII <NP s> *Data*
 [Range]
 [Description] Print a CODE 39 barcode type in vertical. The s parameter indicates the barcode height in millimetres. The Data parameter contains the data to convert, with start and stop characters of barcode.
 [Notes] The "*" star character is the start and the stop character of the barcode.
 [Default]
 [Reference]
 [Example] <X2,L>
 <RC120,10><NP10>*123456*

<NR>	
Devices:	ALL
[Name]	Restore the text in horizontal
[Format]	ASCII <NR>
[Range]	
[Description]	Restore the text in horizontal, without rotation.
[Notes]	
[Default]	
[Reference]	<F:rotate:aa>
[Example]	

<OXY x, y>

Devices: ALL

[Name] **Set printing offset**
 [Format] ASCII <OXY x, y>

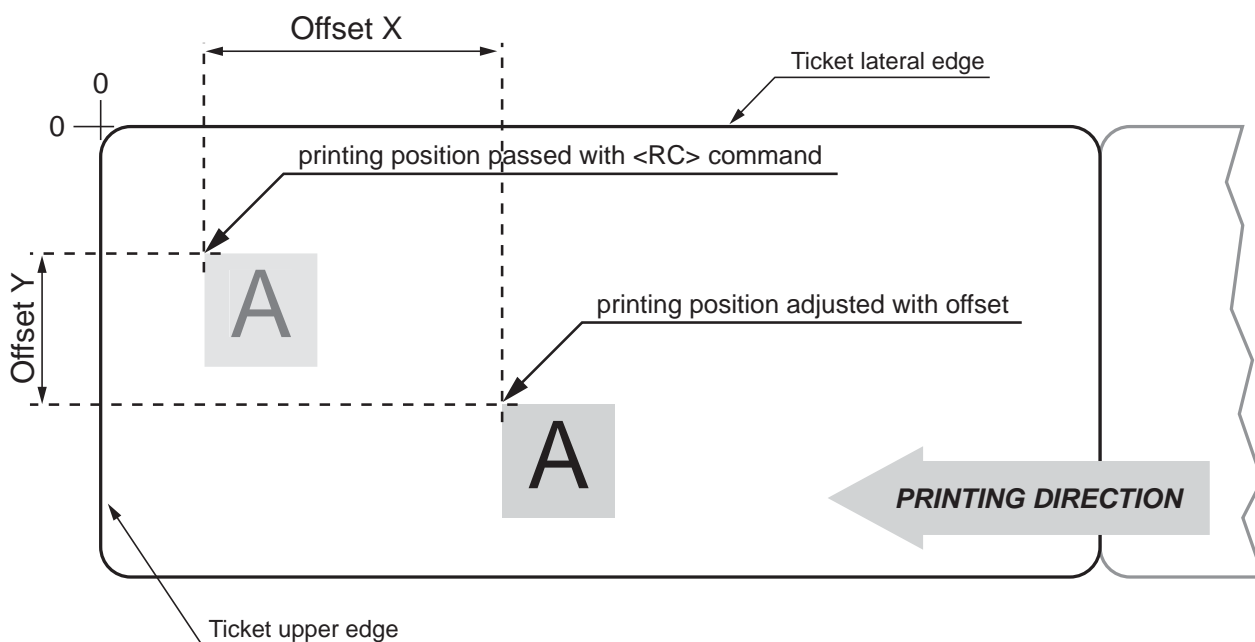
[Range]
 [Description] Sets an offset that will be added to all the transmitted positions, where:
 x is the distance (in dot) between the ticket upper edge and the starting point of printing
 y is the distance (in dot) between the ticket lateral edge and the starting point of printing

This command is useful to adjusting the printout positions, without having to modify all the transmitted positions.

- [Notes]
- If using the point (.) character as decimal separator instead of commas then the passed value are stored in EEPROM.
 - It's possible to set negative values of offset.
 - If you get negative values after adding the offset, (the printing position is outside the ticket), the printing position is set to 0.
 - 1mm = 8 dot.

KPM203, KPM303, TK203, TK303
 • 1mm = 12 dot.

[Default] <RC>
 [Reference]
 [Example]



<p>

Devices: ALL

[Name] **Printing command (cut and buffer cleaning) in reverse**

[Format] ASCII <p>

[Range]

[Description]

This command executes the following operations :

- align the ticket to notch (based on the alignment set with the <LHT> command);
- barcode reader turn ON (only for models with BARCODE reader);
- prints ticket;
- clear the data in the print buffer;
- align the ticket to cut;
- executes a ticket cut.
- recovers the portion of paper equal to the distance between cutter and printing head.

KPM202, KPM203, TK202, TK203

This command executes the following operations :

- align the ticket to notch;
- prints ticket;
- clear the data in the print buffer;
- activate the ticket presentation mode;

[Notes]

- Print ticket in reverse
- After printing, the data of the barcode read and the reading result, are stored in a circular buffer.
- To read the barcode acquired during printing, use the '<BC1>' or '<BCA>' commands.

[Default]

[Reference]

[Example]

<CB>, <LHT>

<P>

Devices: ALL

[Name] **Printing command (cut and buffer cleaning) in normal**

[Format] ASCII <P>

[Range]

[Description]

This command executes the following operations :

- align the ticket to notch (based on the alignment set with the <LHT> command);
- barcode reader turn ON (only for models with BARCODE reader);
- prints ticket;
- clear the data in the print buffer;
- align the ticket to cut;
- executes a ticket cut.
- recovers the portion of paper equal to the distance between cutter and printing head.

KPM202, KPM203, TK202, TK203

This command executes the following operations :

- align the ticket to notch;
- prints ticket;
- clear the data in the print buffer;
- activate ticket presentation mode;

[Notes]

- Print ticket in normal
- After printing, the data of the barcode read and the reading result, are stored in a circular buffer.
- To read the barcode acquired during printing, use the '<BC1>' or '<BCA>' commands.

[Default]

[Reference]

[Example]

<CB>, <LHT>

<PCHexNumLogo HexXDim HexYDim HexTBD Id Hexdata>

Devices: *ALL*

[Name] **Save the image received from serial port into flash**
[Format] ASCII <PCHexNumLogo HexXDim HexYDim HexTBD Id Hexdata>
[Range]
[Description] Save the image received from serial port into printer flash; if the number used to store logo is not already present inside the printer, the new logo is appended to stored logos, otherwise the image is overwritten and moved in the last position of flash.
 • The source image must be a monochrome bitmap.
HexNumLogo indicates the number of logo, 2 bytes expressed in hexadecimal notation;
HexXDim indicates the logo horizontal dimension in pixel, 2 bytes expressed in hexadecimal notation; the value must be multiple of 32;
HexYDim indicates the logo vertical dimension in pixel, 2 bytes expressed in hexadecimal notation;
HexTBD 2 bytes fixed to \$00 (RESERVED);
Id indicates the file-name of the logo, a sequence of 16 bytes that identify univocally the logo;
Hexdata are the image data (logo's bytes less than the first 62 bytes of the header).
 • The printer returns a sequence of bytes as follows :
 <PC0> if the saving include an incorrect syntax or the available memory in flash for logos is finished (128Kbyte);
 <PC1n> if the syntax command is correct and there's enough memory in flash for saving logos; n returns the status of the flash programming :
 \$88 -> Sector not erased
 \$77 -> Error during programming
 \$AA -> Programming done.

[Notes]

- The logo is stored into the printer flipped vertically relative to the bitmap
- The colors of monochrome bitmaps may appear reversed if the "palette" in the header of the bitmap in position 0x3B is 0xFF 0xFF 0xFF 0x00".
- If file-name length is shorter than 16 byte, add a terminator byte NULL (0x00) up to 16 characters.
- If file-name extension is absent, it is automatically added to the name.

[Default]
[Reference]
[Example]

The following example shows the bytes sequence received from serial port to store a logo into the printer flash :

```

Offset      Hexadecimal                                     ASCII
00000000: 3C 50 43 00 08 00 60 00 58 00 00 65 78 61 6D 70  <PC...`.X..examp
00000010: 6C 65 6C 6F 67 6F 38 00 00 00 00 00 00 00 2F  lelogo8.bmp
....
....
....
>
```

Image data less than the first 62 bytes

If the programming is successful, the printer's answer will be:

HEX	\$3C	\$50	\$43	\$31	\$AA	\$3E
ASCII	<	P	C	1	{	>

<PE n>Devices: *ALL*

[Name] **Delete image**
[Format] ASCII <PE n>
[Range]
[Description] Deletes image defined by n.
The printer returns a sequence of bytes as follows :
<PE0> Image n not found;
<PE1n> Image found; n returns to the flash programming status
\$88 -> Sector not erased
\$77 -> Error during erasing operation
\$AA -> Erasing done.

[Notes]
[Default]
[Reference]
[Example]

<PI n>Devices: *ALL*

[Name] **Get pictures header info**
[Format] ASCII <PI n>
[Range]
[Description] Gets the logo header info stored specified by n (express in ASCII). The printer returns a bytes sequence as follows :
<PIe[ID]> where
• e indicates the search result
 e = 0 picture not found
 e = 1 picture found

• [ID] indicates the file-name that identify the logo, a sequence of 16 bytes that was defined when the logo is stored. This field is optional because it's returned only if the logo has been found.

[Notes]
[Default]
[Reference]
[Example]

<PL>	
Devices:	ALL
[Name]	Get pictures header list
[Format]	ASCII <PL>
[Range]	
[Description]	<p>This command requests to the printer the list of stored logo. The printer returns a bytes sequence as follows :</p> <p><PL CrLf [N-ID CrLf]> where</p> <ul style="list-style-type: none"> • CrLf indicates the two characters \$0D (Carriage return) and \$0A (Line Feed); • N is the number of stored logo; • [ID] indicates the file-name that identify the logo, a sequence of 16 bytes that was defined when the logo is stored. This field is optional because it's returned only if the logo has been found.
[Notes]	<ul style="list-style-type: none"> • The fields enclosed in square bracket are repeated for all number of stored images.
[Default]	
[Reference]	
[Example]	

<PN>																					
Devices:	ALL																				
[Name]	Get number of stored logo																				
[Format]	ASCII <PN>																				
[Range]																					
[Description]	<p>This command sends to the printer the request of number of stored logo; the printer returns a bytes sequence as follows : <PNn></p> <p>where n (in ASCII format) indicates the number of stored images.</p>																				
[Notes]																					
[Default]																					
[Reference]																					
[Example]	<p>If in the flash memory are stored 10 logos send this command</p> <table border="1" style="margin-left: 40px;"> <tr> <td>HEX</td> <td>\$1C</td> <td>\$90</td> </tr> <tr> <td>ASCII</td> <td>FS</td> <td>{}</td> </tr> </table> <p>The printer's answer will be :</p> <table border="1" style="margin-left: 40px;"> <tr> <td>HEX</td> <td>\$3C</td> <td>\$50</td> <td>\$4E</td> <td>\$31</td> <td>\$30</td> <td>\$3E</td> </tr> <tr> <td>ASCII</td> <td><</td> <td>P</td> <td>N</td> <td>1</td> <td>0</td> <td>></td> </tr> </table>	HEX	\$1C	\$90	ASCII	FS	{}	HEX	\$3C	\$50	\$4E	\$31	\$30	\$3E	ASCII	<	P	N	1	0	>
HEX	\$1C	\$90																			
ASCII	FS	{}																			
HEX	\$3C	\$50	\$4E	\$31	\$30	\$3E															
ASCII	<	P	N	1	0	>															

<PP n, x, y, sp>

Devices:	ALL
[Name]	Print image in graphic page
[Format]	ASCII <PP n, x, y, sp>
[Range]	
[Description]	Prints image in graphic page where <ul style="list-style-type: none"> • <i>n</i> is the number of image to print; • <i>x</i> indicates the horizontal position inside the graphic page • <i>y</i> indicates the vertical position inside the graphic page • <i>sp</i> indicates the thickness value of the image border (express in dot).
[Notes]	<ul style="list-style-type: none"> • if <i>n</i> is a negative number the image is printed as a background image, without deleting the area below.
[Default]	
[Reference]	<OXY x, y>
[Example]	<p>Several printing commands in graphic page; in the first printing command the image no. 2 is printed with border, instead the other images are printed without border:</p> <pre><CB><n><BA8><HW1,1><BS0,0> <PP2,10,10,8> (image printed with border) <PP1,10,200,0> (image printed without border) <PP3,210,200,0> (image printed without border) <PP4,620,200,0> (image printed without border) <q></pre>

<PR n, x, y, sp>

Devices:	ALL
[Name]	Print rotated image
[Format]	ASCII <PR n, x, y, sp>
[Range]	
[Description]	Prints rotated image in graphic page where <ul style="list-style-type: none"> • <i>n</i> is the number of image to print; • <i>x</i> indicates the horizontal position inside the graphic page • <i>y</i> indicates the vertical position inside the graphic page • <i>sp</i> indicates the thickness value of the image border (express in dot).
[Notes]	<ul style="list-style-type: none"> • if <i>n</i> is a negative number the image is printed as a background image, without deleting the area below.
[Default]	
[Reference]	<OXY x, y>
[Example]	<p>Several printing commands in graphic page; in the first printing command the image no. 2 is printed with border, instead the other images are printed without border:</p> <pre><CB><n><BA8><HW1,1><BS0,0> <PR2,10,10,8> (image printed with border) <PR1,10,200,0> (image printed without border) <PR3,210,200,0> (image printed without border) <PR4,620,200,0> (image printed without border) <q></pre>

<q>	
Devices:	ALL
[Name]	Printing command (only buffer cleaning) in reverse
[Format]	ASCII <q>
[Range]	
[Description]	This command executes the following operations : <ul style="list-style-type: none">- align the ticket to notch (based on the alignment set with the <LHT> command);- barcode reader turn ON (only for models with BARCODE reader);- prints ticket;- clear the data in the print buffer; <p>KPM202, KPM203, TK202, TK203</p> <p>This command executes the following operations :</p> <ul style="list-style-type: none">- align the ticket to notch;- prints ticket;- clear the data in the print buffer;
[Notes]	<ul style="list-style-type: none">• Print ticket in reverse• After printing, the data of the barcode read and the reading result, are stored in a circular buffer.• To read the barcode acquired during printing, use the '<BC1>' or '<BCA>' commands.
[Default]	
[Reference]	<CB>, <LHT>
[Example]	

<Q>	
Devices:	ALL
[Name]	Printing command (only buffer cleaning) in normal
[Format]	ASCII <Q>
[Range]	
[Description]	This command executes the following operations : <ul style="list-style-type: none">- align the ticket to notch (based on the alignment set with the <LHT> command);- barcode reader turn ON (only for models with BARCODE reader);- prints ticket;- clear the data in the print buffer; <p>KPM202, KPM203, TK202, TK203</p> <p>This command executes the following operations :</p> <ul style="list-style-type: none">- align the ticket to notch;- prints ticket;- clear the data in the print buffer;
[Notes]	<ul style="list-style-type: none">• Print ticket in normal• After printing, the data of the barcode read and the reading result, are stored in a circular buffer.• To read the barcode acquired during printing, use the '<BC1>' or '<BCA>' commands.
[Default]	
[Reference]	<CB>, <LHT>
[Example]	

<qn>		
Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>
[Name]	Printing command without alignment in reverse	
[Format]	ASCII	<qn>
[Range]		
[Description]	<p>This command executes the following operations :</p> <ul style="list-style-type: none"> - barcode reader turn ON (only for models with BARCODE reader); - prints ticket; - clear the data in the print buffer; 	
[Notes]	<ul style="list-style-type: none"> • Print ticket in reverse • After printing, the data of the barcode read and the reading result, are stored in a circular buffer. • To read the barcode acquired during printing, use the '<BC1>' or '<BCA>' commands. 	
[Default]		
[Reference]	<CB>, <LHT>	
[Example]		

<QN>		
Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>
[Name]	Printing command without alignment in normal	
[Format]	ASCII	<QN>
[Range]		
[Description]	<p>This command executes the following operations :</p> <ul style="list-style-type: none"> - barcode reader turn ON (only for models with BARCODE reader); - prints ticket; - clear the data in the print buffer; 	
[Notes]	<ul style="list-style-type: none"> • Print ticket in normal • After printing, the data of the barcode read and the reading result, are stored in a circular buffer. • To read the barcode acquired during printing, use the '<BC1>' or '<BCA>' commands. 	
[Default]		
[Reference]	<CB>, <LHT>	
[Example]		

SVELTA Emulation

<RC row, column>

Devices: *ALL*

[Name] **Position the cursor**

[Format] ASCII <RC row, column>

[Range]

[Description] Moves the cursor at the position specified by row and column parameters.

[Notes] • The row and column values must be a number with four digit at most, otherwise the command will be ignored.

[Default]

[Reference] <OXY x, y>

[Example] To move the cursor at row (dot) 10, column (dot) 30 the command sequence is :

<RC 10,30>

<RL>

Devices: *ALL*

[Name] **Rotate text 90° counter-clockwise**

[Format] ASCII <RL>

[Range]

[Description] Rotate text 90° counter-clockwise, (to the left).

[Notes]

[Default]

[Reference] <F:rotate:aa>

[Example]

<RR>

Devices: *ALL*

[Name] **Rotate text 90° clockwise**
[Format] ASCII <RR>
[Range]
[Description] Rotate text 90° clockwise, (to the right).
[Notes]
[Default]
[Reference] <F:rotate:aa>
[Example]

<RU>

Devices: *ALL*

[Name] **Rotate text 180°**
[Format] ASCII <RU>
[Range]
[Description] Rotate text 180°.
[Notes]
[Default]
[Reference] <F:rotate:aa>
[Example]

<SB x>

Devices: *ALL*

[Name] **FULL STATUS back request**

[Format] ASCII <SB x y>

[Range] **KPM202, KPM203, KPM302 (models without triple feeder), KPM303, TK202, TK203, TK302 (models without triple feeder), TK303**
 '0' ≤ x ≤ '9', 'A' ≤ x ≤ 'F'

KPM302 (models with triple feeder), TK302 (models with triple feeder)
 '0' ≤ x ≤ '9', 'A' ≤ x ≤ 'F' y= '0', y = 'F'

[Description] • FULL STATUS back request.
 • x specify the request for FULL STATUS. where x identify the bitmask with the following table:

x	»	BIT3	BIT2	BIT1	BIT0
0	»	0	0	0	0
1	»	0	0	0	1
2	»	0	0	1	0
3	»	0	0	1	1
4	»	0	1	0	0
5	»	0	1	0	1
6	»	0	1	1	0
7	»	0	1	1	1
8	»	1	0	0	0
9	»	1	0	0	1
A	»	1	0	1	0
B	»	1	0	1	1
C	»	1	1	0	0
D	»	1	1	0	1
E	»	1	1	1	0
F	»	1	1	1	1

4° byte = Unrecoverable error status
 3° byte = Recoverable error status
 2° byte = User status
 1° byte = Full status

KPM302 (models with triple feeder), TK302 (models with triple feeder)

• y is an optional parameter; if y = 'F' then enables the transmission of the triple feeder status.

[Notes] • The status sent from the printer will be so composed as follows:

<SB x, CHR1 CHRn>

where:

- SB = fixed characters
- x = is the bitmask to identify the request.
- CHR1..CHRn = response bytes referred to the following tables:

all models

1° byte = Full status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper present
	On	01	1	Paper not present
1	-	-	-	RESERVED
2	Off	00	0	Paper present
	On	04	4	Near paper end
3	-	-	-	RESERVED
4	-	-	-	RESERVED
5	Off	00	0	Ticket not present in output
	On	20	32	Ticket present in output
6	Off	00	0	Not virtual paper end
	On	40	64	Virtual paper end
7	Off	00	0	Notch found
	On	80	128	Notch not found

KPM202, KPM203, KPM302 (models without selector), KPM303 (models without selector), TK202, TK203, TK302, TK303

2° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	-	-	-	RESERVED

KPM302 (models with selector), KPM303 (models with selector)

2° byte = User status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Printing head down
	On	01	1	Printing head up error
1	Off	00	0	Cover closed
	On	02	2	Cover opened
2	Off	00	0	No spooling
	On	04	4	Spooling
3	Off	00	0	Drag paper motor off
	On	08	8	Drag paper motor on
4	-	-	-	RESERVED
5	Off	00	0	LF key released
	On	20	32	LF key pressed
6	Off	00	0	FF key released
	On	40	64	FF key pressed
7	Off	00	0	Selector in "open" position
	On	80	128	Selector in "storage" position

all models

3° byte = Recoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Head temperature ok.
	On	01	1	Head temperature error
1	Off	00	0	No COM error
	On	02	2	RS232 COM error
2	-	-	-	RESERVED
3	Off	00	0	Power supply voltage ok
	On	08	8	Power supply voltage error
4	-	-	-	RESERVED
5	Off	00	0	Acknowledge command
	On	20	32	Not acknowledge command error
6	Off	00	0	Free paper path
	On	40	64	Paper jam
7	Off	00	0	Notch search ok
	On	80	128	Error in notch search

**KPM302 (models with triple feeder),
KPM302 (models with selector), KPM303 (models with selector),
TK302, TK303**

4° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Cutter ok
	On	01	1	Cutter error
1	Off	00	0	Cutter cover ok
	On	02	2	Cutter cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

KPM302 (models with selector), KPM303 (models with selector)

4° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Cutter ok
	On	01	1	Cutter error
1	Off	00	0	Cutter cover ok
	On	02	2	Cutter cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	Off	00	0	Selector OK
	On	80	128	Selector error

KPM202, KPM203, TK202, TK203

4° byte = Unrecoverable error status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	-	-	-	RESERVED
1	Off	00	0	Frontal cover ok
	On	02	2	Frontal cover open
2	Off	00	0	RAM ok
	On	04	4	RAM error
3	Off	00	0	EEPROM ok
	On	08	8	EEPROM error
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

**KPM302 (models with triple feeder),
TK302 (models with triple feeder)**

5° byte = 'I'

6° byte = Feeder sensors status

BIT	OFF/ON	HEX	Decimal	FUNCTION
0	Off	00	0	Paper sensor (Feeder 1 UP): paper not present
	On	01	1	Paper sensor (Feeder 1 UP): paper present
1	Off	00	0	Paper sensor (Feeder 2 CENTER): paper not present
	On	02	2	Paper sensor (Feeder 2 CENTER): paper present
2	Off	00	0	Paper sensor (Feeder 3 DOWN): paper not present
	On	04	4	Paper sensor (Feeder 3 DOWN): paper present
3	-	-	-	RESERVED
4	Off	00	0	Near paper end sensor (Feeder 1 UP): paper not present
	On	10	16	Near paper end sensor (Feeder 1 UP): paper present
5	Off	00	0	Near paper end sensor (Feeder 2 CENTER): paper not present
	On	20	32	Near paper end sensor (Feeder 2 CENTER): paper present
6	Off	00	0	Near paper end sensor (Feeder 3 DOWN): paper not present
	On	40	64	Near paper end sensor (Feeder 3 DOWN): paper present
7	-	-	-	RESERVED

7° byte = 'A'

8° Byte = FEEDER 1 (UP)

- = 0 No paper in feeder 1
- = 1 Paper in ACTIVE STATUS
- = '7' Paper end
- = 9 Error
- = 10 Paper in PARK STATUS

9° Byte = 'B'

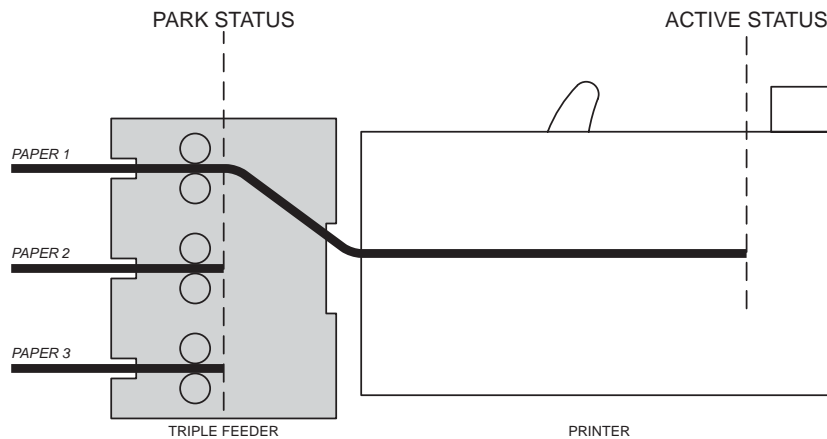
10° Byte = FEEDER 2 (CENTER)

- = 0 No paper in feeder 2
- = 1 Paper in ACTIVE STATUS
- = '7' Paper end
- = 9 Error
- = 10 Paper in PARK STATUS

11° Byte = 'C'

12° Byte = FEEDER 3 (DOWN)

- = 0 No paper in feeder 3
- = 1 Paper in ACTIVE STATUS
- = '7' Paper end
- = 9 Error
- = 10 Paper in PARK STATUS



- The printer transmits bytes 1,2,3,4 and 6 as a pair of hexadecimal characters (between '0' and '9' or between 'A' and 'F'). For example the first byte is equal to \$A9, then will be sent from the printer the characters 'A' (\$41) and '9' (\$39).

[Default]
 [Reference]
 [Example]

<SBF, 00000000> no errors
 <SBF, 04000000> near paper end
 <SBF, 01030000> paper not present, printing head up, cover open

To request the Full status (1° byte) and the User status (2°byte) proceed as follow:

see bitmask:
 BIT3 = 0 BIT2 = 0 BIT1 = 1 BIT0 = 1 quindi 0011 = 3

Send the command: <AFSB3>

Possible answer: <SB3,0504>

where:

1°byte

0 = 0000	bit7 = 0 (notch found)	bit6 = 0 (not virtual paper end)	bit5 = 0 (ticket not present)	bit4 = 0 (RESERVED)
5 = 0101	bit3 = 0 (RESERVED)	bit2 = 1 (near paper end)	bit1 = 0 (RESERVED)	bit0 = 1 (Paper not present)

2°byte

0 = 0000	bit7 = 0 (RESERVED)	bit6 = 0 (FF key released)	bit5 = 0 (LF key released)	bit4 = 0 (RESERVED)
4 = 0100	bit3 = 0 (drag motor off)	bit2 = 1 (spooling)	bit1 = 0 (cover closed)	bit0 = 0 (print head down)

<SDT m data>

Devices: *ALL*

[Name] **Set date/time of the real time clock**
 [Format] ASCII <SDT m data>
 [Range]
 [Description] Set date/time of the real time clock, in the format specified by m values as follows :

m	FORMAT
0	DD/MM/YY hh:mm:ss
1	DDMMYYhhmmss
2	YYMMDDhhmmss
3	YYMMDDhhmmssd

where:
 DD = represents the day of the date
 MM = represents the month of the date
 YY = represents year of the date
 hh = represents the hour of the time
 mm = represents the minutes of the time
 ss = represents the seconds of the time
 d = indicates the day of the week
 data are the ASCII characters relative to the date and time to set.

If the transmission has been received correctly and the command is valid, the printer returns the following string: <SDT ↵ x ↵ >

where
 - ↵ corresponds to CR character (\$0D).
 - x indicate the reading result ; the x value can be :
 '!' : the command is executed successfully
 '#': the command is not executed successfully

[Notes] • the day of the week is calculated automatically from the printer and then it's possible that the returned value is different from the one transmitted.

[Default]
 [Reference]
 [Example]

For example to set the date and time to "29 September 2006 at 13:51:00 (PM)" in the "YYM-MDDhhmmss" format transmit:
 <SDT 2 061029135100>
 The printer's answer will be :
 <SDT ↵ ! ↵ > if the transmission is successfully, otherwise
 <SDT ↵ # ↵ > if the transmission is not successfully

<SELECTORI>

Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>

[Name]	Initialize selector
[Format]	ASCII <SELECTORI>
[Range]	
[Description]	This command performs a movement of the selector mechanisms in the two available positions. If the selector is mechanically unable to move, the flag status indicates an error.
[Notes]	At the end of the movement, selector is set in the "Open" position (default).
[Default]	
[Reference]	
[Example]	

<SELECTORO>

Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>

[Name]	Set selector in "Open" position
[Format]	ASCII <SELECTORO>
[Range]	
[Description]	This command set the selector in the "Open" position: the paper exits the printer regularly. If the selector position is already the desired one, this command does not generate any movement.
[Notes]	
[Default]	
[Reference]	
[Example]	

<SELECTORS>

Devices:	KPM302	<i>(models with selector)</i>
	KPM303	<i>(models with selector)</i>

[Name] **Set selector in "Storage" position**
 [Format] ASCII <SELECTORS>
 [Range]
 [Description] This command set the selector in the "Storage" position: paper exits the printer downwards. If the selector position is already the desired one, this command does not generate any movement.
 [Notes]
 [Default]
 [Reference]
 [Example]

<S n>

Devices:	<i>ALL</i>
----------	------------

[Name] **Status request**
 [Format] ASCII <Sn>
 [Range]
 [Description] The host can ask to the printer many different status info; the n parameter indicates which type of request :

KPM302, KPM303, TK302, TK303

If n = 1 the printer return a byte that represent the status:

- \$10: Paper end
- \$11: No error
- \$19: Wrong command
- \$20: Notch error
- \$21: Heading over temperature error
- \$22: Power supply voltage error
- \$23: Cutter error

KPM202, KPM203, TK202, TK203

If n = 1 the printer return a byte that represent the status:

- \$10: Paper end
- \$11: No error
- \$19: Wrong command
- \$20: Notch error
- \$21: Heading over temperature error
- \$22: Power supply voltage error

- If n=3 the printer return ACK (\$06) if printing is properly finished, otherwise return NACK (\$15). If the request will be transmitted during printing phase, it waits the end of the process and then is sent the answer.

[Notes]
 [Default]
 [Reference]
 [Example]

<SP n>Devices: *ALL*

[Name] **Change speed**
 [Format] ASCII <SP n>
 [Range]
 [Description] Sets printing speed using n as follows :

n	PRINTING SPEED
0	High quality
1	Normal
2	High speed

[Notes]
 [Default]
 [Reference]
 [Example]

<SVEL>Devices: *ALL*

[Name] **Change printer emulation to SVELTA**
 [Format] ASCII <SVEL>
 [Range]
 [Description] Set the SVELTA emulation.

[Notes]
 [Default]
 [Reference]
 [Example]

<T>Devices: *ALL*

[Name] **Get the ticket dimension to print**
 [Format] ASCII <T>
 [Range]
 [Description] Get the ticket dimensions to print, in the Ticket Size format.

[Notes]
 [Default]
 [Reference]
 [Example]

<TDF m data>

Devices: *ALL*

[Name] **Set user-defined date/time formats**
 [Format] ASCII <TDF m data>

[Range]
 [Description] Sets the format string for date and time used to printing;
 • m specifies which user-defined string format is set
 D for date
 T for time
 • data are the ASCII characters relative to user-defined date/time formats.
 • the maximum length of the user-defined date/time format string is 64 chars.
 The following table shows characters used to create user-defined date/time formats :

CHARACTER	DESCRIPTION
I	Selects Italian language
E	Selects English language (is the default language)
c	Selects default date/time
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (for example, Sun).
dddd	Displays the day as a full name (for example, Sunday).
dddddd	Displays the date as a complete date in the short format where date values are formatted with day, month and year (the short date format is dd/mm/yy).
dddddd	Displays the date as a complete date in the extended format where date values are formatted with day, month and year (the extended date format is dd mmmm, yyyy).
m	Displays the month as a number without a leading zero (1-12). If the character m is immediately after the character h or hh , displays the minutes instead of month (see also the n character formatting).
mm	Displays the month as a number with leading zeros (01-12). If the character m is immediately after the character h or hh , displays the minutes instead of month (see also the nn character formatting).
mmm	Displays the month as an abbreviation (for example, Jan).
mmmm	Displays the month as a full month name (for example, January).
yy	Displays the year in two-digit numeric format with a leading zero.
yyyy	Displays the year in four digit numeric format.

CHARACTER	DESCRIPTION
h	Displays the hour as a number without leading zeros (0-23)
hh	Displays the hour as a number with leading zeros (00-23)
n	Displays the minutes as a number without leading zeros (0-59)
nn	Displays the minutes as a number with leading zeros (00-59)
s	Displays the seconds as a number without leading zeros (0-59)
ss	Displays the seconds as a number with leading zeros (00-59)
tttt	Displays the time in the extended format where time values are formatted with hour, minutes and seconds (the extended time format is h:mm:ss).
AM/PM	Using the 12-hour clock and displays the AM prefix in uppercase next to the hours that preceding midday and the PM prefix in uppercase next to the hours between midday and midnight.
am/pm	Using the 12-hour clock and displays the am prefix in lowercase next to the hours that preceding midday and the pm prefix in lowercase next to the hours between midday and midnight.
A/P	Using the 12-hour clock and displays the A prefix in uppercase next to the hours that preceding midday and the a prefix in uppercase next to the hours between midday and midnight.
a/p	Using the 12-hour clock and displays the a prefix in lowercase next to the hours that preceding midday and the a prefix in lowercase next to the hours between midday and midnight.

[Notes]
 [Default]
 [Reference]
 [Example]

For example to print the current time with the string format 'yy/mm/dd hh:mm:ss' follow these steps :

1. Send the following command to define the user-defined Time string format:

<TDF T yy/mm/dd hh:mm:ss>

2. Send the following command to print the time :

<TIME>

If the date and time is 22 October 2006 at 17:35:27 (PM) the output string printed will be:

06/10/22 17:35:27

<TIME>

Devices: *ALL*

[Name] **Print Time**
 [Format] ASCII <TIME>
 [Range]
 [Description] Prints time with the format specified by the command '<TDF>'.
 [Notes]
 [Default] "hh:nn:ss"
 [Reference] <DATE>
 [Example]

<X n, m>

Devices: *ALL*

[Name] **Define the barcode lines dimension**
 [Format] ASCII <X n, M>
 [Range]
 [Description] n defines the thins lines dimension (in dot) of barcode. The M parameter defines the barcode printing speed if it must be printed rotated.
 [Notes] If the M parameter = 'H' as ASCII value, the barcodes will be printed in high speed. Otherwise if the M parameter = 'L' as ASCII value the barcodes will be printed at reduced speed (only if n is less than 4).
 [Default]
 [Reference]
 [Example]

4 ALIGNMENT: PRACTICAL APPLICATIONS

The device is equipped with sensors that allows the use of alignment notch to handle:

- rolls of tickets with pre-printed and fixed length fields;
- FanFold modules of tickets with pre-printed and fixed length fields.

For further information, refer to the User Manual of each device.

4.1 Alignment commands: ESC/POS™ emulation

The commands available for managing the alignment of the ticket are the following:

- \$1D \$E7: sets the distance between the point of alignment and the notch (value of parameter "Notch Distance")
- \$1D \$F6 and \$1D \$F8: perform the ticket alignment, which is advanced to cut the ticket at the first alignment point available
- \$1C \$C1 : performs the desired recovery of the paper after the cutting operation

Print a ticket with alignment requires the following sequence of commands:

1. General settings of the ticket: character formatting, print density, margins etc..
2. Alignment command: \$1D \$F6.
3. Ticket printout: printing text, logos or any graphic.
4. Alignment command: \$1D \$F8.
5. Cut command.

NOTE:

The settings take effect from next ticket to the one already in the printer.

In the following examples, are described some sequences of commands to manage the alignment.

EXAMPLE 1

Commands sequence to print two tickets with "alignment point" used to align the cut over the edge of the notch (Notch Distance = 0mm = \$00 \$00).

{Set Notch Distance}

\$1D,\$E7,\$00,\$00,

{Alignment}

\$1D, \$F6,

{Print text}

'TICKET 1', \$0A, 'FIRST LINE', \$0A, 'SECOND LINE', \$0A

{Alignment}

\$1D, \$F8,

{Cut}

\$1B \$69,

{Alignment}

\$1D, \$F6,

{Print text}

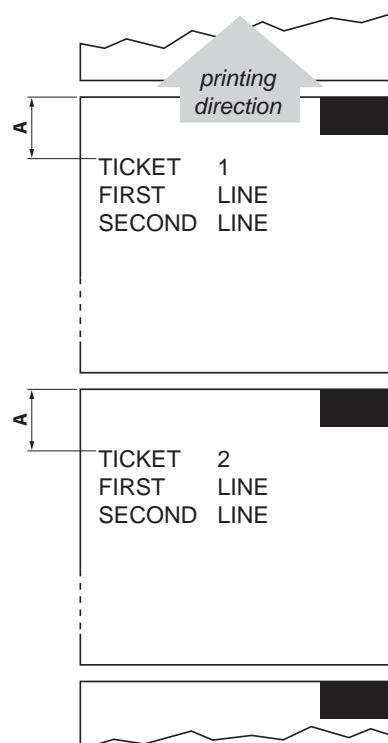
'TICKET 2', \$0A, 'FIRST LINE', \$0A, 'SECOND LINE', \$0A

{Alignment}

\$1D,\$F8,

{Cut}

\$1B \$69



Alignment: practical applications

NOTE:

The dimension A shown in the figure, represents the non-printable area, equal to the distance between cutting line and printing line that may be recoverable by the \$1C \$C1 command.

NOTE:

For a better comprehension of the images, the black mark has been represented on the heat sensitive side of the paper.

EXAMPLE 2

Commands sequence to print tickets with "alignment point" used to the cut the paper 8mm before the notch ("Notch Distance" = 8mm = 80 tenths of a millimeter = \$ 00 \$ 50).

{Set Notch Distance}

\$1D, \$E7, \$00, \$50,

{Alignment}

\$1D, \$F6,

{Print text}

'TICKET 1', \$0A, 'FIRST LINE', \$0A, 'SECOND LINE', \$0A

{Alignment}

\$1D, \$F8,

{Cut}

\$1B \$69,

{Alignment}

\$1D, \$F6,

{Print text}

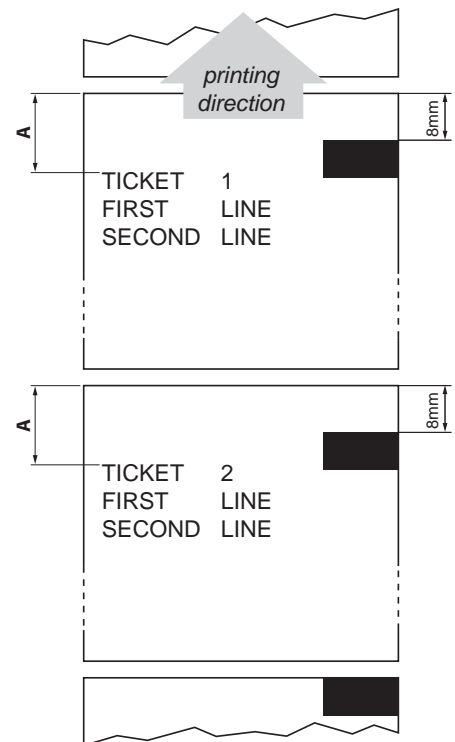
'TICKET 2', \$0A, 'FIRST LINE', \$0A, 'SECOND LINE', \$0A

{Alignment}

\$1D, \$F8,

{Cut}

\$1B \$69



NOTE:

The dimension A shown in the figure, represents the non-printable area, equal to the distance between cutting line and printing line that may be recoverable by the \$1C \$C1 command.

NOTE:

For a better comprehension of the images, the black mark has been represented on the heat sensitive side of the paper.

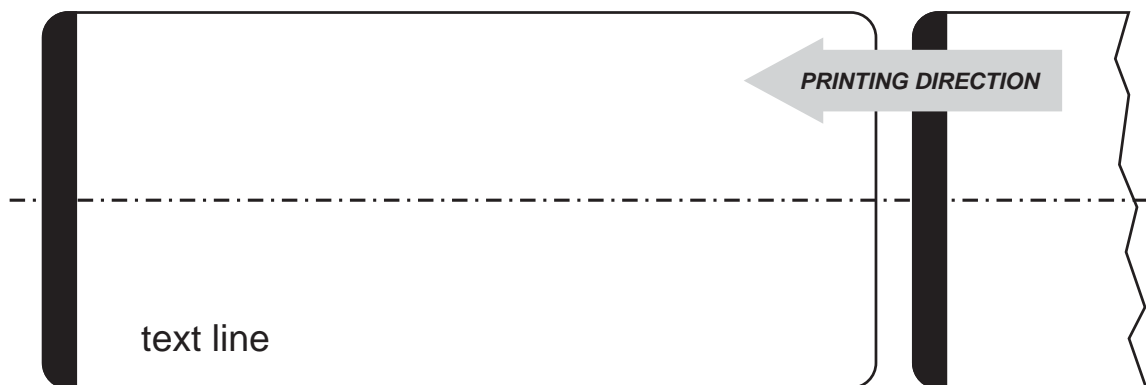
4.2 Alignment commands: SVELTA emulation

In SVELTA emulation, alignment is automatically managed if set during the printer setup procedure by the following commands:

- <p>, <P>, <q> and <Q>: printing commands that perform also the ticket alignment.
- <LHT length, height, notch, dimnotch> : defines the alignment point, the notch size and the ticket size.
- <OXY x, y>: adjusts the position of the page to be printed within the ticket.

The following example shows the commands sequence to print a ticket with "alignment point" used to cut the paper on the notch edge (Notch = 0mm).

```
{Setting of ticket properties}
<LHT1616,656,0,40>
{Setting of printing coordinates}
<RC50,624>
{Text}
text line
{Alignment, text printout and cut}
<p>
```



NOTE:

For a better comprehension of the images, the black mark has been represented on the heat sensitive side of the paper.

Alignment: practical applications

5 PAPER SPECIFICATIONS

This chapter shows the specifications for paper types available for devices related to this manual.

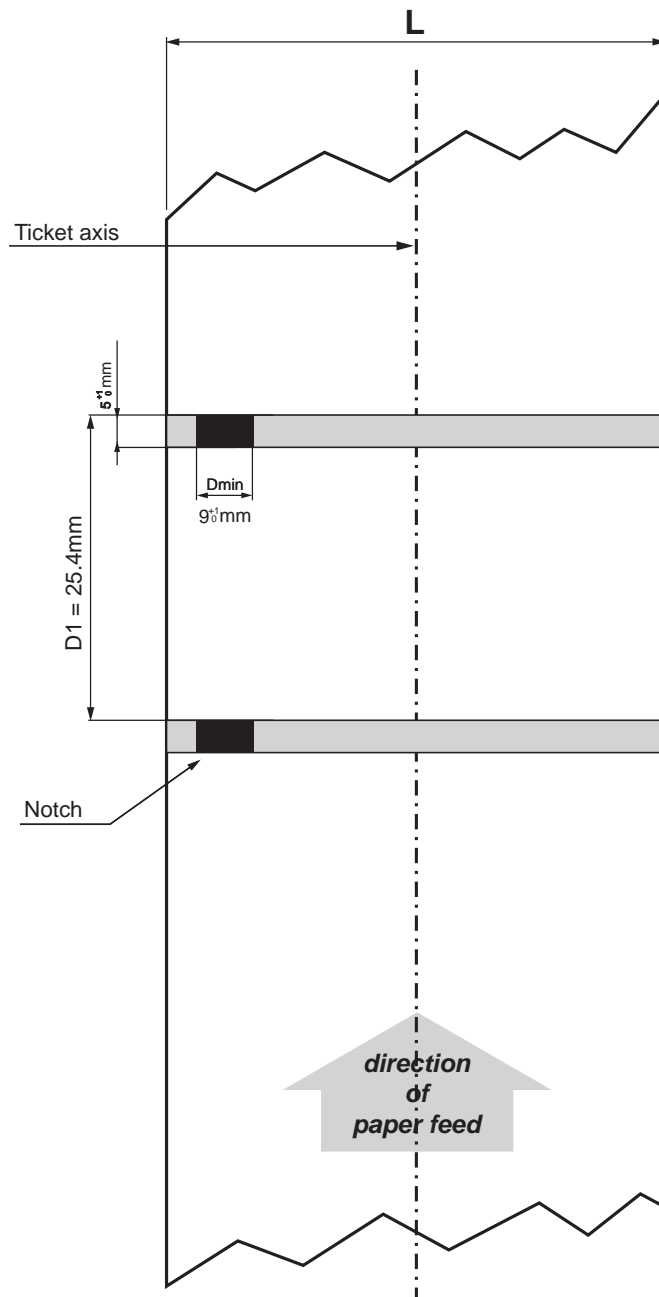
5.1 Paper with alignment notch

KPM202, KPM203

KPM302, KPM303

TK202, TK203

TK302, TK303



L = width of paper used

Dmin = minimum notch dimension

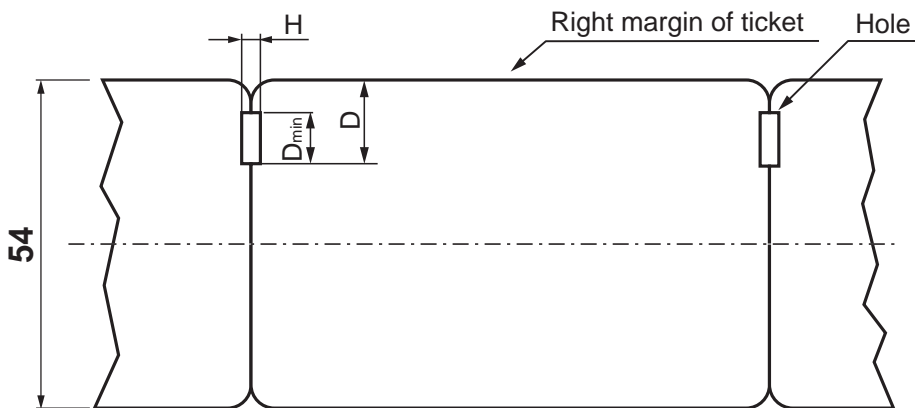
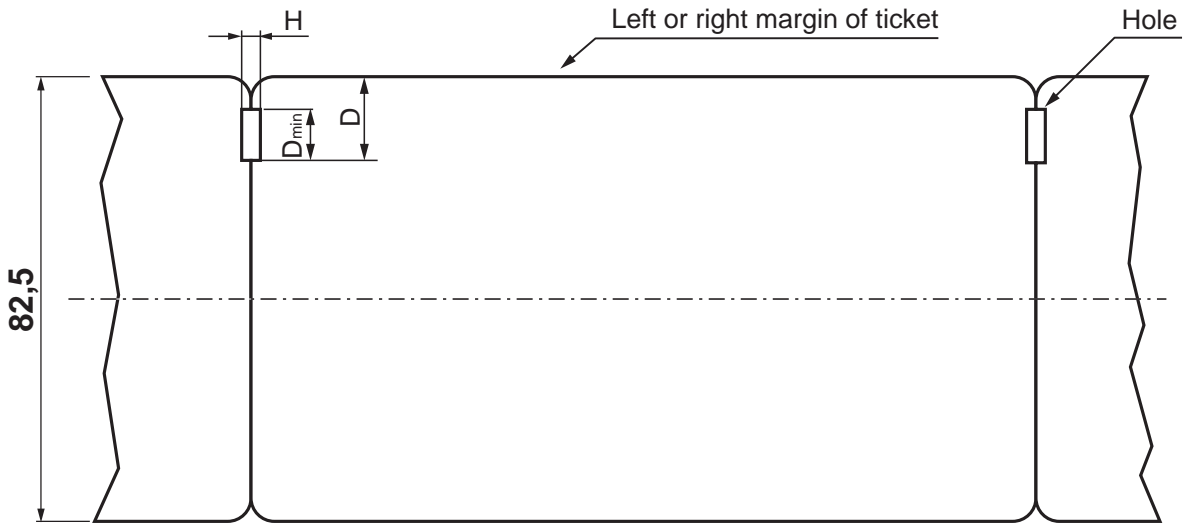
D1 = minimum notch to notch distance

Paper specifications

5.2 Ticket with hole

KPM302 (models with triple feeder)

TK302 (models with triple feeder)

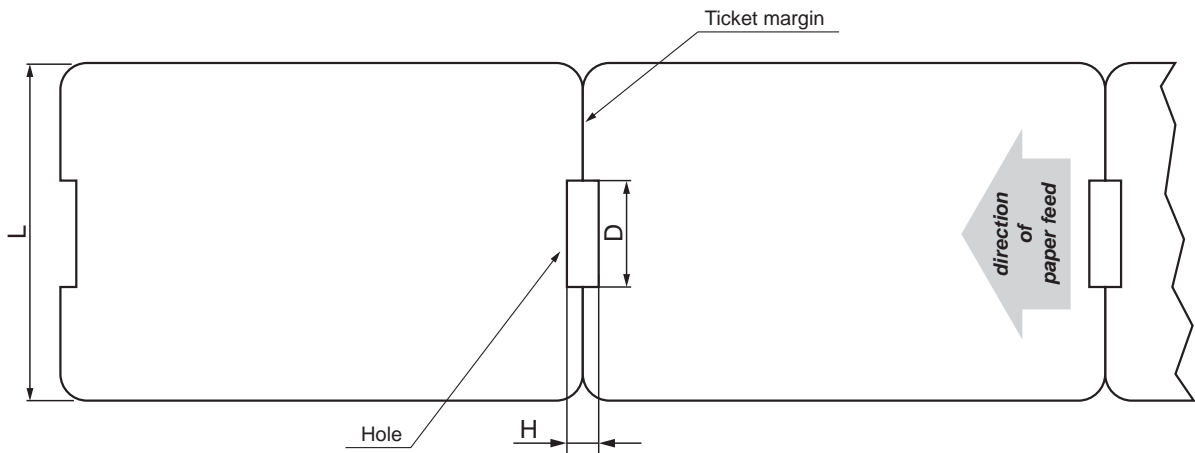


H = minimum hole height (2mm)

D_{min} = minimum hole width (10mm)

D = maximum allowed distance between hole edge and ticket edge (20mm)

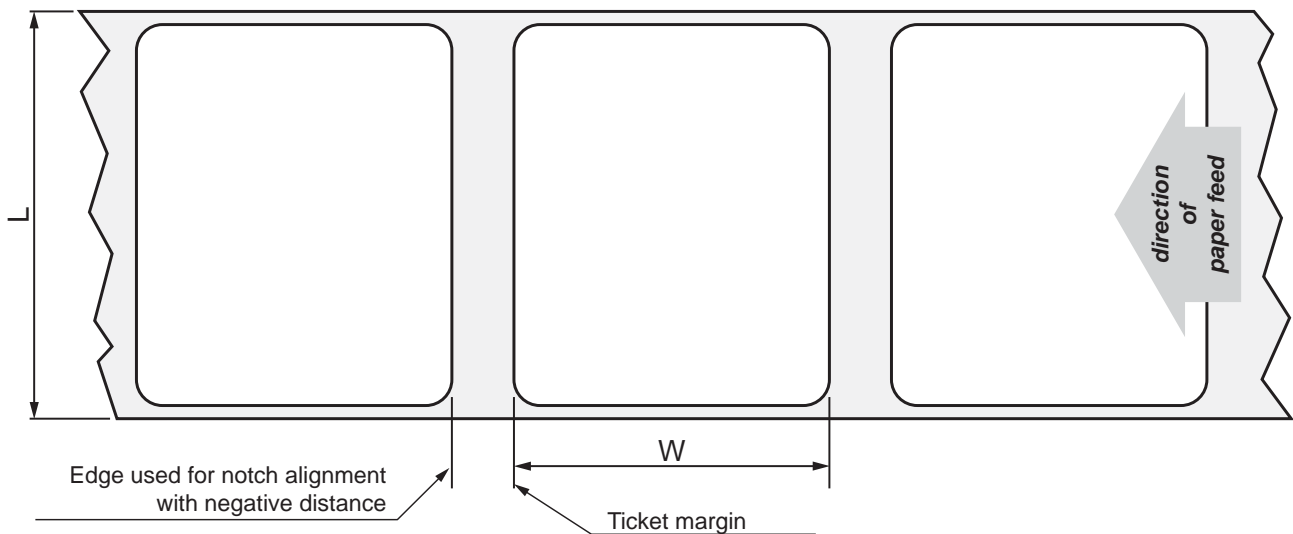
KPM202, KPM203,
KPM302 (models without triple feeder), KPM303
TK202, TK203
TK302 (models without triple feeder), TK303



L = width of paper used
 H = minimum hole height (2mm)
 D = minimum hole width (10mm)

5.3 Paper with labels

KPM302 (models without triple feeder),
KPM303
TK302 (models without triple feeder),
TK303



L = width of paper used
 W = minimum managed length 25,4mm (1")

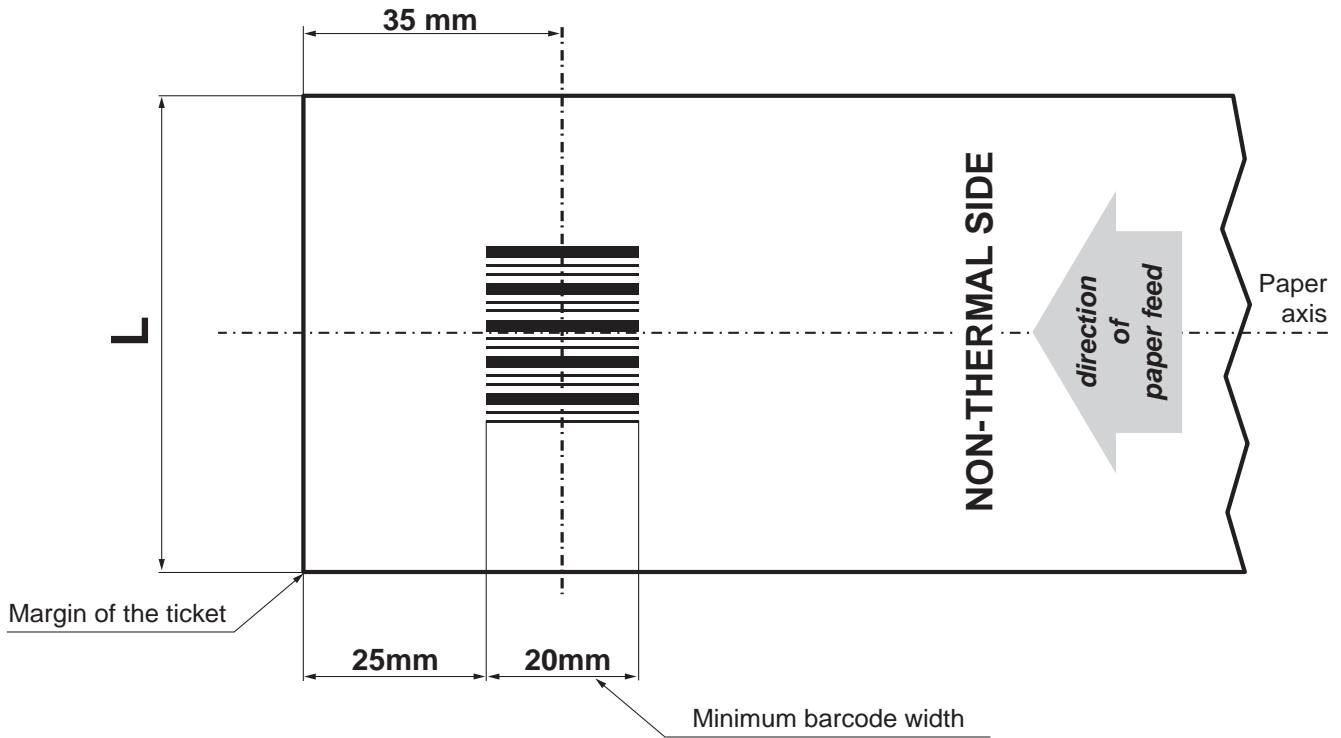
5.4 Paper with one-dimensional barcode

KPM302 (models with reader for one-dimensionale barcode)

KPM303 (models with reader for one-dimensionale barcode)

TK302 (models with reader for one-dimensionale barcode)

TK303 (models with reader for one-dimensionale barcode)



L = width of paper used

5.5 Ticket with RFID tag

KPM302 (models with RFID tag reader)

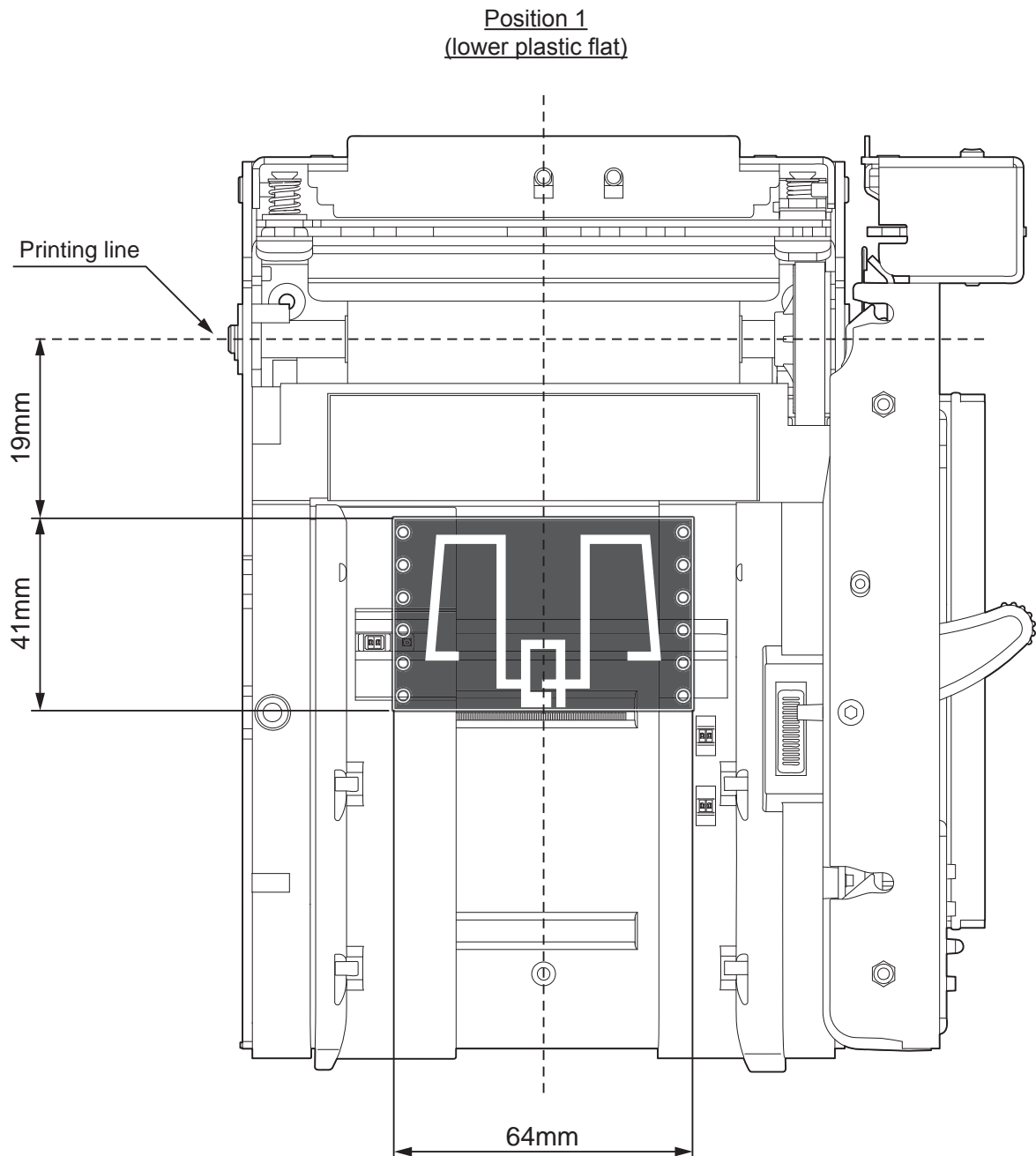
KPM303 (models with RFID tag reader)

TK302 (models with RFID tag reader)

TK303 (models with RFID tag reader)

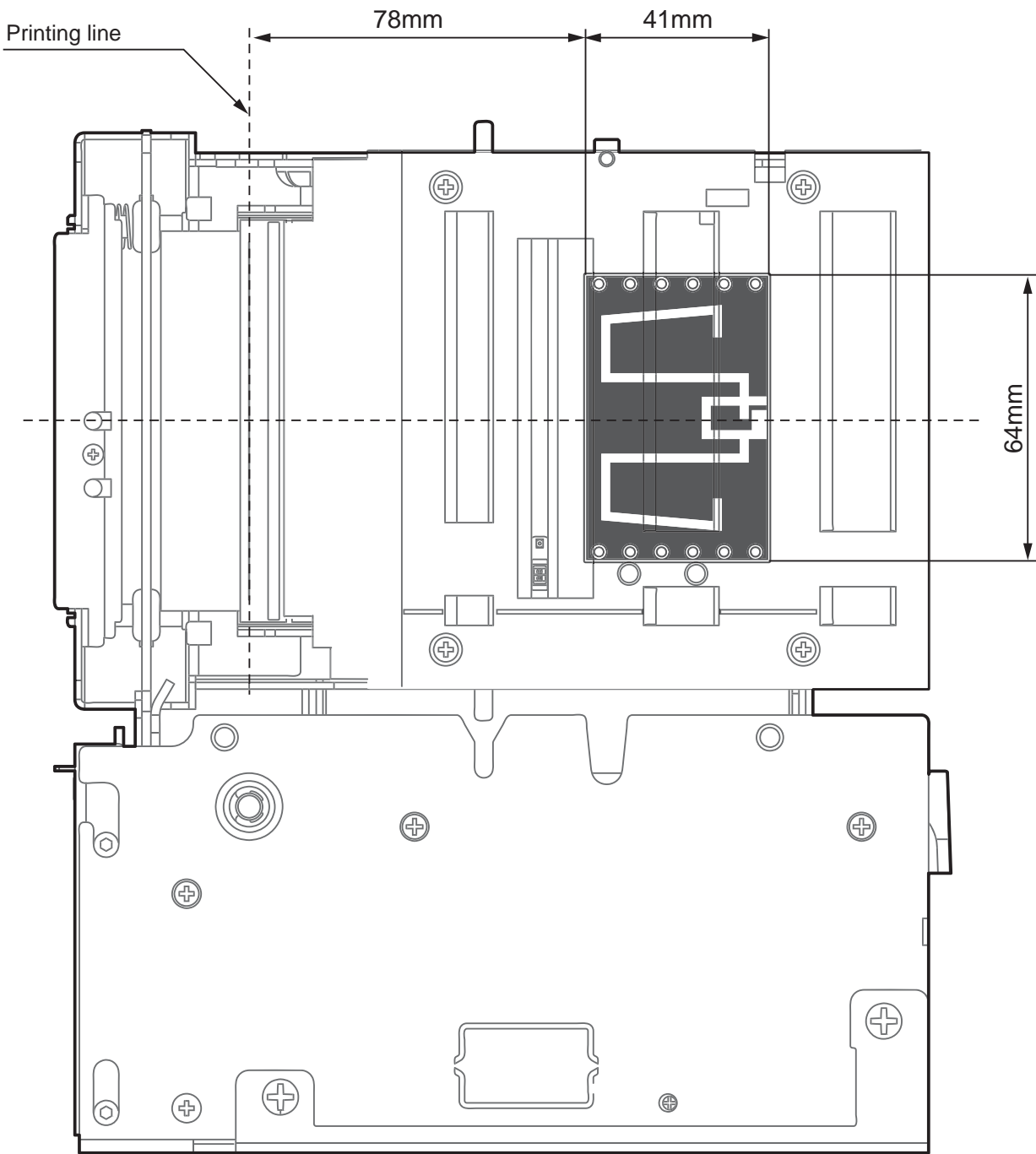
For this application the ticket dimensions are not binding but for good reading is important that the tag inside the ticket, after alignment, intersects the antenna area.

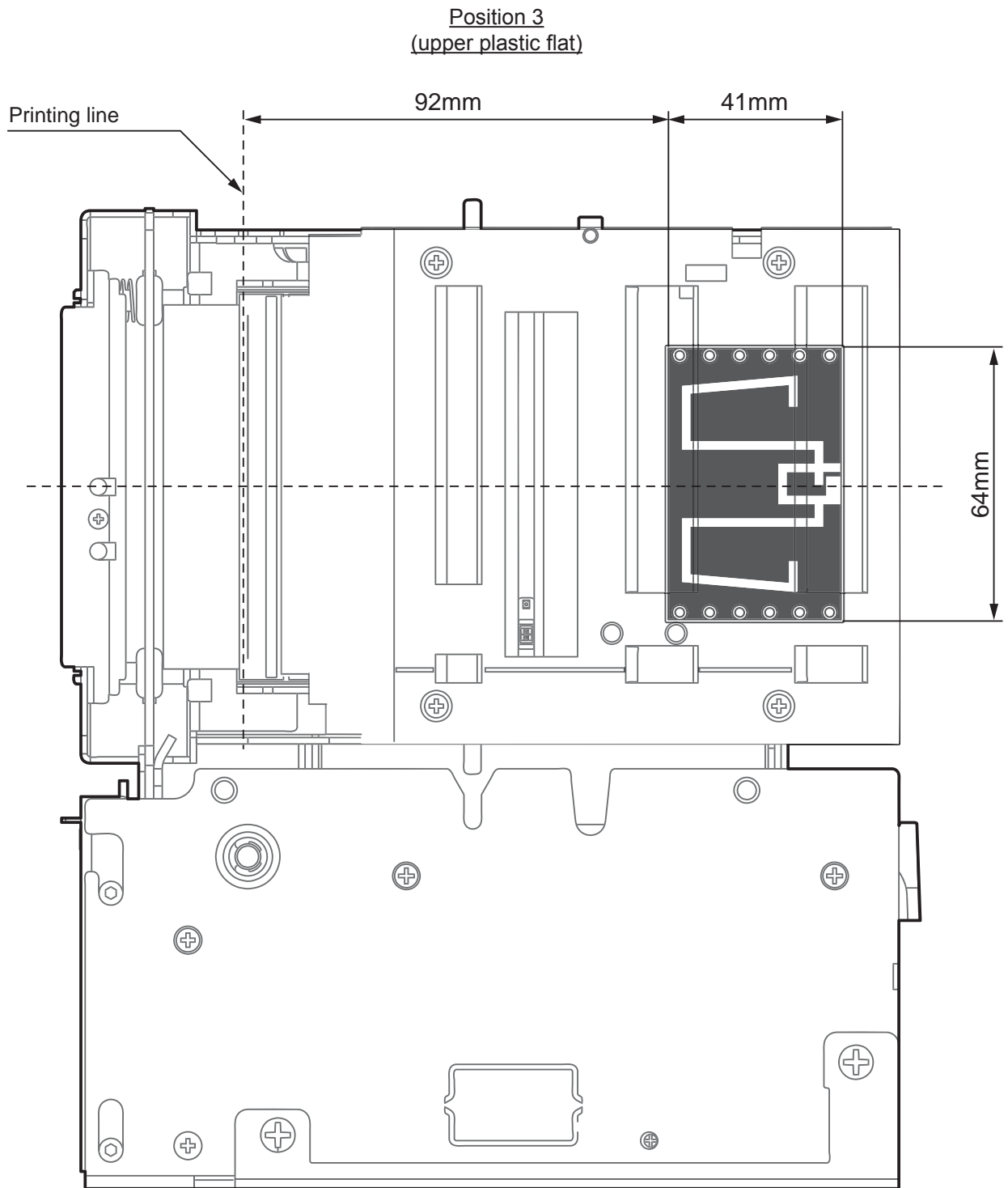
The following figure shows the antenna's area and its position under the paper guide in the RFID printer model



Paper specifications

Position 2
(upper plastic flat)





NOTE:

Using ticket with RFID tag, the minimum managed length is a credit card size ticket (84x54 mm).

6 COMMANDS INDEX

ESC/POS™ EMULATION

\$08.....	10	\$1C \$B2.....	60
\$09.....	10	\$1C \$C0.....	61
\$0A.....	11	\$1C \$C1.....	62
\$0D.....	11	\$1D \$21.....	63
\$10 \$04.....	12	\$1D \$28 \$6B.....	64
\$18.....	25	\$1D \$28 \$6B [function 065].....	65
\$1B \$20.....	25	\$1D \$28 \$6B [function 066].....	65
\$1B \$21.....	26	\$1D \$28 \$6B [function 067].....	66
\$1B \$24.....	27	\$1D \$28 \$6B [function 068].....	66
\$1B \$25.....	27	\$1D \$28 \$6B [function 069].....	67
\$1B \$26.....	28	\$1D \$28 \$6B [function 080].....	68
\$1B \$28 \$76.....	29	\$1D \$28 \$6B [function 081].....	69
\$1B \$2A.....	30	\$1D \$28 \$6B [function 065].....	70
\$1B \$2D.....	31	\$1D \$28 \$6B [function 066].....	70
\$1B \$30.....	32	\$1D \$28 \$6B [function 067].....	71
\$1B \$32.....	32	\$1D \$28 \$6B [function 069].....	72
\$1B \$33.....	33	\$1D \$28 \$6B [function 080].....	73
\$1B \$3D.....	33	\$1D \$28 \$6B [function 081].....	73
\$1B \$3F.....	34	\$1D \$28 \$6B [function 365].....	74
\$1B \$40.....	34	\$1D \$28 \$6B [function 366].....	74
\$1B \$44.....	35	\$1D \$28 \$6B [function 367].....	75
\$1B \$45.....	35	\$1D \$28 \$6B [function 368].....	76
\$1B \$47.....	36	\$1D \$28 \$6B [function 380].....	77
\$1B \$4A.....	36	\$1D \$28 \$6B [function 381].....	78
\$1B \$4D.....	37	\$1D \$28 \$6B [function 065].....	79
\$1B \$52.....	37	\$1D \$28 \$6B [function 067].....	79
\$1B \$56.....	38	\$1D \$28 \$6B [function 068].....	80
\$1B \$5C.....	39	\$1D \$28 \$6B [function 069].....	81
\$1B \$61.....	40	\$1D \$28 \$6B [function 080].....	82
\$1B \$63 \$35.....	40	\$1D \$28 \$6B [function 081].....	82
\$1B \$64.....	41	\$1D \$2A.....	83
\$1B \$69.....	41	\$1D \$2F.....	84
\$1B \$69.....	42	\$1D \$3A.....	85
\$1B \$74.....	42	\$1D \$42.....	85
\$1B \$76.....	43	\$1D \$48.....	86
\$1B \$7B.....	44	\$1D \$49.....	87
\$1B \$C1.....	44	\$1D \$4C.....	88
\$1C \$0C.....	45	\$1D \$50.....	89
\$1C \$3C.....	46	❶ \$1D \$56, ❷ \$1D \$56.....	89
\$1C \$65.....	46	\$1D \$57.....	90
\$1C \$66.....	47	\$1D \$5E.....	91
\$1C \$6C.....	48	\$1D \$65 \$35.....	91
\$1C \$80.....	49	\$1D \$66.....	92
\$1C \$81.....	50	\$1D \$68.....	92
\$1C \$82.....	51	❶ \$1D \$6B, ❷ \$1D \$6B.....	93
\$1C \$83.....	51	\$1D \$70 \$69.....	95
\$1C \$84.....	52	\$1D \$70 \$6F.....	95
\$1C \$90.....	54	\$1D \$70 \$73.....	96
\$1C \$91.....	54	\$1D \$72.....	96
\$1C \$92.....	55	\$1D \$76 \$30.....	97
\$1C \$93.....	56	\$1D \$77.....	98
\$1C \$94.....	57	\$1D \$7C.....	98
\$1C \$B0.....	58	\$1D \$E0.....	99
\$1C \$B1.....	59	\$1D \$E1.....	100
		\$1D \$E2.....	101
		\$1D \$E3.....	101

\$1D \$E5.....	102
\$1D \$E6.....	102
\$1D \$E7.....	103
\$1D \$F0.....	104
\$1D \$F6.....	104
\$1D \$F8.....	105

SVELTA EMULATION

<AFSB x>	110
<bXnn>	116
	116
<B2D k, A, x>	117
<B2D k, B, x>	117
<B2D k, C, x>	118
<B2D k, D, x>	118
<B2D k, E, m, x>	119
<B2D k, P, x, d1...dn>	120
<B2D l, A, x>	120
<B2D l, B, x>	121
<B2D l, C, x>	121
<B2D l, D, x>	122
<B2D l, P, x, d1...dn>	122
<B2D m, A, n>	123
<B2D m, B, n>	123
<B2D m, C, n>	124
<B2D m, D, n>	125
<B2D m, P, n, d0...dk>	125
<B2D n, A, n>	126
<B2D n, B, n>	126
<B2D n, C, n>	127
<B2D n, D, n>	127
<B2D n, P, n, d0...dk>	128
<BA> n.....	128
<BC n>	129
<BEEP 1, tt>	130
<BF x1 y1, x2, y2>	130
<BMP>	131
<BMPD>	131
<BS height, width>	132
<BV x1, y1, x2, y2>	132
<BX x1, y1, x2, y2, s, t>	133
<CB>	134
<COM1>	135
<COM2>	135
<CUT>	136
<DATE>	136
<DT m>	137
<EJOUT>	138
<EPOS>	138
<F:bold>	139
<F:clear>	139
<F:draw:n>	139
<F:enc:ascii>	140
<F:enc:utf-8>	140
<F:enc:utf-16>	140
<F:err:n>	141
<F:filename.ttf>	141
<F:italic>	141
<F n>	142
<F:regular>	142
<F:rotate:aa>	142
<F:size:nn>	143
<HW height, width>	143
<INPUT n>	144
<KEYS x>	144
<LHT length, height, notch, dimnotch>	145

<LOAD>..... 146
 <MM n> 146
 <NCL x,y>Data 147
 <NCP x,y>Data..... 148
 <NEL n> *Data* 149
 <NEP n> *Data* 149
 <NFL s> *Data* 150
 <NFP s> *Data* 150
 <NL s> *Data* 151
 <NP s> *Data* 151
 <NR> 152
 <OXY x, y> 153
 <p> 154
 <P> 155
 <PCHexNumLogo HexXDim HexYDim HexTBD Id
 Hexdata> 156
 <PE n>..... 157
 <PI n> 157
 <PL> 158
 <PN> 158
 <PP n, x, y, sp> 159
 <PR n, x, y, sp> 159
 <q> 160
 <Q>..... 160
 <qn> 161
 <QN> 161
 <RC row, column> 162
 <RL> 162
 <RR> 163
 <RU> 163
 <SB x>..... 164
 <SDT m data> 170
 <SELECTORI> 171
 <SELECTORO> 171
 <SELECTORS>..... 172
 <S n> 172
 <SP n>..... 173
 <SVEL> 173
 <T> 173
 <TDF m data> 174
 <TIME> 175
 <X n, m> 175

CUSTOM



M . U . R . S . T .
Ministry University
Research Scientific
T e c h n o l o g y
Authorized laboratory
n o . 5 0 8 4 6 Z Y Z

CUSTOM ENGINEERING S.p.A.

World Headquarters

Via Berettine, 2 - 43010 Fontevivo, Parma ITALY

Tel. +39 0521 680111 - Fax +39 0521 610701

info@custom.biz - www.custom.biz

All rights reserved