

COMMANDS MANUAL

PP54

PP54 EVO

CUSTOM[®]

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

Customer Service Department:
www.custom4u.it

© 2022 CUSTOM 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 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 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 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 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.

GENERAL INSTRUCTIONS

CUSTOM 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.

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.
- Do not fix indissolubly the device or its accessories such as power supplies unless specifically provided in this manual.
- When positioning the device, make sure cables do not get damaged.
- [Only OEM equipment] The equipment must be installed in a kiosk or system that provides mechanical, electrical and fire protection.
- The mains power supply must comply with the rules in force in the Country where you intend to install the equipment.
- Make sure that there is an easily-accessible outlet with a capacity of no less than 10A closely to where the device is to be installed.
- Make sure the power cable provided with the appliance, or that you intend to use is suitable with the wall socket available in the system.
- Make sure the electrical system that supplies power to the device is equipped with a ground wire and is protected by a differential switch.
- Before any type of work is done on the machine, disconnect the power supply.
- Use the type of electrical power supply indicated on the device label.
- These devices are intended to be powered by a separately certified power module having an SELV, non-energy hazardous output. (IEC60950-1 second edition).
- [Only POS equipment] The energy to the equipment must be provided by power supply approved by CUSTOM S.p.A.
- Take care the operating temperature range of equipment and its ancillary components.
- 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.
- The equipment must be accessible on these components only to trained, authorized personnel.
- Periodically perform scheduled maintenance on the device to avoid dirt build-up that could compromise the correct, safe operation of the unit.
- 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.
- Use consumables approved by CUSTOM S.p.A.



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 2014/30/EU and 2014/35/EU inasmuch as it was designed in conformity with the provisions laid down in the following Standards:

- EN 55032 (*Electromagnetic compatibility of multimedia equipment - Emission Requirements*)
- EN 55024/EN55035 (*Electromagnetic compatibility of multimedia equipment - Immunity requirements*)
- EN IEC/EN62368-1 (*Audio/video, information and communication technology equipment*)

The device is in conformity with the essential requirements laid down in Directives 2014/53/EU about devices equipped with intentional radiators. The Declaration of Conformity and other available certifications can be downloaded from the site www.custom4u.it.



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 2012/19/EU, 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.



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

INTRODUCTION



CUSTOM/POS EMULATION



TICKET LAYOUT



SCRIPT INSTRUCTIONS





INTRODUCTION

- 1 CONSULTING COMMANDS MANUAL 7
- 2 PAPER SPECIFICATIONS 9
- 3 EVENTS MANAGEMENT 11
- 4 RETURN CODES GUIDE 12
- 5 ANSWER PROTOCOL FORMAT 14



1 CONSULTING COMMANDS MANUAL

Each command reported in this manual is described as shown in the following picture. In the first heading field is reported the hexadecimal command value and the ASCII command value. In the second heading field reported the command function. In the third heading field are listed the devices on which it is possible to use the command (for example, device AAAA).

Link to index

Command value

Command function

Devices that use the command

0x0D

<CR>

Print and carriage return

Valid for

AAAA

BBBB

CCCC

[Format]

Hex

0x0D

ASCII

CR

[Range]

[Description]

When Autofeed is "CR enabled", this command function in the same way as 0x0A, otherwise it is disregarded.

[Notes]

This command sets the printing position to the beginning of the line.

AAAA

BBBB

• This command sets the printing position to the beginning of the line.

CCCC

• This command is immediately executed even when the data buffer is full.

• This status is transmitted whenever data sequence is received.

[Default]

[Reference]

0x0A

[Example]

Information valid for devices AAAA, BBBB, CCC

Information valid only for devices AAAA, BBBB

Information valid only for device CCCC



The fields shown in the scheme of the previous figure have the following meaning:

[Format]	hexadecimal and ASCII 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]	Example of using the command.

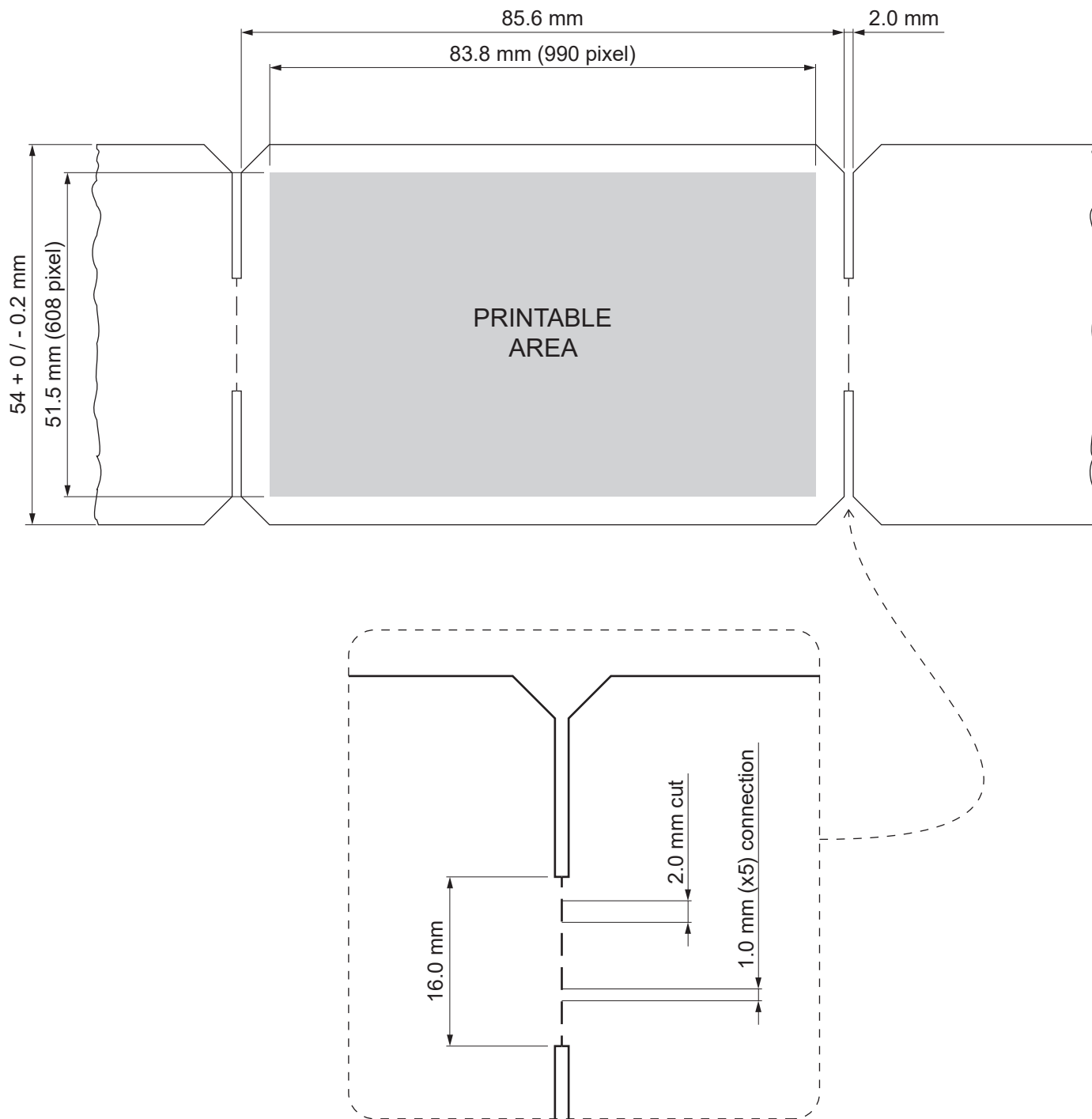
Listed below are the meanings of some of symbols that may be found in the command description:

0x	indicates the representation of the command hexadecimal value (for example 0x40 means HEX 40).
n, m, t, x, y	are optional parameters that can have different values.

2 PAPER SPECIFICATIONS

Thermal Fan-fold module with barcode

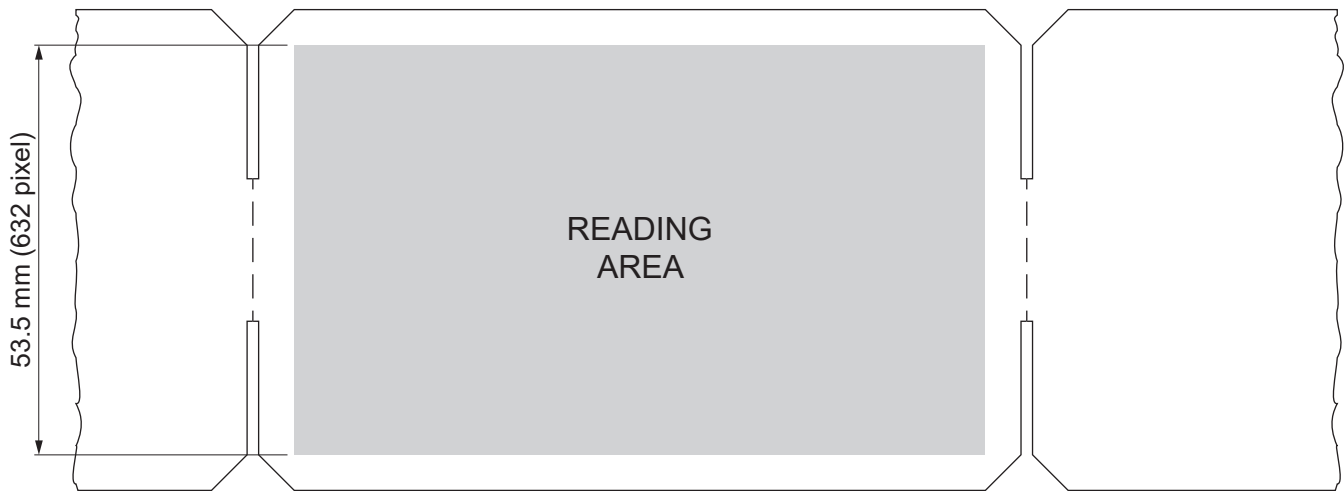
The device manages ISO 7810 ID-1 format ticket (85.60mm x 53.98 mm) with 300dpi resolution (11.8 dot/mm). The following figure shows the specifications for ISO 7810 ID-1 tickets:



Line perforation: 2.0 mm cut / 1.0 mm connection
 No imprinting on perforation area
Pull-off strength: 45 N

Barcode reading area

The following figure shows the specifications for the reading area of ticket by using the integrated barcode reader which works on a width area of 632 pixels:



3 EVENTS MANAGEMENT

The events management are the messages sent by device automatically when events occur.

Event message

Event message is generated by the device and sent to the host. The device send the status package where the second byte of the status indicates the event generated in according to the value of n parameter as follows:

n	EVENT
0x01	No barcode acquired
0x02	Barcode found
0x05	The ticket was taken by the user
0x06	Ticket loaded
0x07	Ticket lean on frontal input
0x08	Reset from command
0x09	Reset from power fail
0x0A	Alarm occurred

- With n = 0x01, n = 0x02:

Those events are generated when the reading process is completed. This inform HOST about the result of the requested operation.

- With n = 0x05:

The event is sent by device automatically when the ticket is taken by the user, after a presentation command.

- With n = 0x07:

The event is sent when ticket is placed by the outside to the frontal paper mouth of the device. It is possible to use it to understand when it is necessary to send the frontal ticket loading command (refer to the 0x1F 0x70 command with n = 'S').

- With n = 0x0A:

This event is sent to the host whenever an alarm occurs (Paper jam, Paper end, etc.).

- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.

- The <DATA> bytes are as follows:

0x010 n St0 ... ST9

- Once received the package that containing the event the status byte associated to it, returns to the value 0.

NOTE: To know the values of the bytes from ST0 to ST9 refer to the "0x10 0x04" command.



4 RETURN CODES GUIDE

If the command cannot be processed followed by a byte that returns the code of error type. In the following table is shown the list of signalling codes with message, description and command reference:

CODE	“SIGNALLING MESSAGE”	DESCRIPTION	COMMAND REFERENCE
0x01	COMMAND PARAMETER ERROR	Error in the parameters of the command	-
0x02	DEVICE NOT READY	The device is not ready to process commands because is performing another operation	-
0x03	TICKET FEEDER EMPTY	Paper feeder empty	0x1F 0x70
0x04	TICKET PRESENT	Do not run the command because the ticket is present	0x1F 0x70
0x05	TICKET NOT PRESENT	Do not run the command because the ticket is not present	0x1F 0x70
0x06	PARKING SLOT NOT EMPTY	There is a ticket in a parking slot	0x1F 0x70
0x07	PARKING SLOT EMPTY	There is no ticket in a parking slot. The parking slot is empty	0x1F 0x70
0x08	READING IN PROGRESS	The device is already reading a ticket. In this state, you can stop the reading operation (*)	0x1F 0x70
0x09	READING NOT IN PROGRESS	The device is not reading anything because is not in the reading process.	0x1F 0x70
0x0A	ALARM IN PROGRESS	There is an alert in the byte of Alarms status (9th byte of the Device status) because one or more bits are active	-
0x0B	TICKET RFID READING	The RFID ticket is in reading position	0x1F 0x70
0x0C	TICKET RFID NOT READING	The RFID ticket is not in the reading position	0x1F 0x70
0x0D	SCRIPT IN PROGRESS	A script is running	-
0x0E	UNKNOWN COMMAND	The command sent does not exist	-
0x0F	COMMAND NOT AVAILABLE	The command sent is not available because one or more options are not mounted	-
0x10	COMMAND ERROR	Generic error during command execution	-
0x11	FEEDER ALREADY SELECTED	The feeder is already selected	0x1F 0x70
0x1E	LAYOUT FIELD ERROR	Error in the layout file such as error in the declaration of the field	0x1F 0x6C
0x1F	INCORRECT DATA FIELD	Error in the layout file such as incorrect data field received	0x1F 0x64
0x20	LAYOUT MISSING	The device detects an error because the layout file is missing	0x1F 0x64
0x21	OUT OF MEMORY	The device detects an error there is no more space in the internal memory of the device to save the file	0x1F 0x64, 0x1F 0x6C, 0x1F 0x4F
0x22	OBJECT ERROR DOWNLOAD	Error during the object download procedure (Logo, Font, Script)	0x1F 0x4F



0x23	OBJECT ERROR DELETE	Error during object deletion (Logo, Font, Layout, Script)	0x1F 0x4F
0x24	TICKET CAPACITY REACHED	Maximum number of ticket data files reached; the buffer must be clean with a print command	0x1F 0x64
0x25	TICKET DATA MISSING	The device can't print, ticket data is missing	0x1F 0x70
0x80	DATA FIELD OUT OF PRINTABLE AREA	A data field exceeds the printable area limits (**)	0x1F 0x64

NOTES:

(*): If a read disable command is received by device when a ticket is processing (before the reading process is completed), the device abort operation and ejects the ticket on the frontal mouth.

(**): The ticket can be printed anyway but the printout may be different from the expected layout.



5 ANSWER PROTOCOL FORMAT

The device provides a response to any request received, the answer is always structured according to the following protocol:

<STX> <DIM> <DATA> <CKS> <ETX>

where:

<STX>	1 byte	= start of packet character, fixed to 0x02
<DIM>	2 bytes	= Size of <DATA>, the first byte is the most significant
<DATA>	DIM bytes	= Data buffer of response
<CKS>	1 byte	= checksum calculated on the <DATA> buffer using LRC algorithm
<ETX>	1 byte	= end of packet character, fixed to 0x03

In this manual, all descriptions of the responses to the various commands refer to the <DATE> field illustrated in this paragraph.

The calculation of the checksum is done in according to the LRC algorithm (Longitudinal Redundancy Check) defined by the ISO standard 1155, refer to the following code:

```

Set LRC = 0
For each byte b in the buffer
do
    Set LRC = (LRC + b) AND 0xFF
end do
Set LRC = (((LRC XOR 0xFF) + 1) AND 0xFF)
    
```

More information about the LRC algorithm are available at the web site: https://en.wikipedia.org/wiki/Longitudinal_redundancy_check

Examples

Example of ACK answer generated by the printer if command is successful:

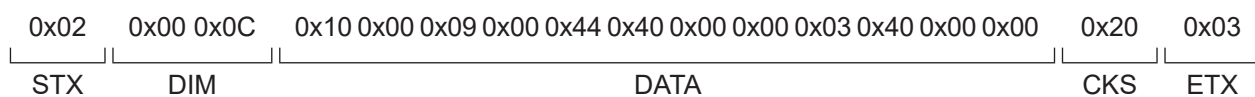
0x02	0x00 0x01	0x06	0xFA	0x03
STX	DIM	DATA	CKS	ETX

where:

0x02	—	STX
0x00 0x01	—	Size of the data package, in this case 1 byte.
0x06	—	Data package that includes only the ACK byte, 0x06
0xFA	—	Checksum calculated in according to the LRC algorithm $(0xFF - 0x06) + 0x01 = 0xFA$
0x03	—	ETX



Example of a response to the status command 0x10 0x04 0x17:



where:

- 0x02 – STX
- 0x00 0x0C – Size of the data package, in this case 12 bytes.
- 0x10...0x00 – Data package that contains the status of the printer buffer
- 0x20 – Calculated checksum according to the LRC algorithm
- 0x03 – ETX





CUSTOM/POS EMULATION

- 1 COMMANDS LISTED IN ALPHANUMERIC ORDER 18
- 2 COMMANDS LISTED BY FUNCTION 20



1 COMMANDS LISTED IN ALPHANUMERIC ORDER

0x10 0x04	<DLE EOT>	32
0x1B 0x26 0x6C	<ESC & l>	70
0x1B 0x26 0x6C	<ESC & l>	71
0x1B 0x2A 0x62	<ESC * b>	72
0x1B 0x2A 0x62	<ESC * b>	73
0x1B 0x2A 0x70	<ESC * p>	75
0x1B 0x2A 0x70	<ESC * p>	76
0x1B 0x2A 0x70	<ESC * p>	77
0x1B 0x2A 0x70	<ESC * p>	78
0x1B 0x2A 0x70	<ESC * p>	79
0x1B 0x2A 0x70	<ESC * p>	80
0x1B 0x2A 0x74	<ESC * t>	81
0x1B 0x3D 0x31	<ESC = 1>	52
0x1B 0x3D 0x35	<ESC = 5>	53
0x1C 0x80		54
0x1C 0x81		55
0x1C 0xEA		56
0x1D 0x7C		57
0x1D 0xDA 0x4C	<GS 0xDA L>	50
0x1D 0xDA 0x63	<GS 0xDA c>	51
0x1D 0xDA	<GS 0xDA>	48
0x1D 0xDA	<GS 0xDA>	49
0x1D 0xF0		58
0x1F 0x42	<US B>	37
0x1F 0x44	<US D>	40
0x1F 0x49	<US l>	59



0x1F 0x4C	.<US L>	60
0x1F 0x4F	.<US O>	30
0x1F 0x52	.<US R>	62
0x1F 0x62 0x00	.<US b>	41
0x1F 0x62	.<US b>	42
0x1F 0x64	.<US d>	64
0x1F 0x65	.<US e>	66
0x1F 0x67 0x47	.<US g G>	67
0x1F 0x69	.<US i>	45
0x1F 0x6C	.<US l>	69
0x1F 0x6F	.<US o>	47
0x1F 0x70	.<US p>	23



2 COMMANDS LISTED BY FUNCTION

TICKET MANAGEMENT COMMANDS

0x1F 0x70.....<US p>.....	23
Ticket movement	

OBJECT MANAGEMENT COMMANDS

0x1F 0x4F.....<US O>.....	30
Save/Delete command logos, fonts, layouts and script	

STATUS COMMANDS

0x10 0x04.....<DLE EOT>.....	32
Real-time device status transmission	

SCANNER COMMANDS

0x1F 0x42.....<US B>.....	37
Set a barcode search area (ROI)	
0x1F 0x44.....<US D>.....	40
Enable debug features	
0x1F 0x62 0x00.....<US b>.....	41
Reading number of barcode for each area	
0x1F 0x62.....<US b>.....	42
Return the barcodes of specified area	

I/O COMMANDS

0x1F 0x69.....<US i>.....	45
I/O status transmission	
0x1F 0x6F.....<US o>.....	47
Set I/O output	

LCD COMMANDS

0x1D 0xDA.....<GS 0xDA>.....	48
Clear LCD	



0x1D 0xDA	<GS 0xDA>	49
Write a message on LCD		
0x1D 0xDA 0x4C	<GS 0xDA L>	50
Show an image on LCD		
0x1D 0xDA 0x63	<GS 0xDA c>	51
Set LCD contrast		

MISCELLANEOUS COMMANDS

0x1B 0x3D 0x31	<ESC = 1>	52
Disable RFID Pass-Through		
0x1B 0x3D 0x35	<ESC = 5>	53
Enable RFID Pass-Through		
0x1C 0x80		54
Read date/time of the real time clock		
0x1C 0x81		55
Set date/time of the real time clock		
0x1C 0xEA		56
Transmit the device serial number		
0x1D 0x7C		57
Set printing density		
0x1D 0xF0		58
Set printing speed		
0x1F 0x49	<US I>	59
Transmit printer ID		
0x1F 0x4C	<US L>	60
Retrieve log file		
0x1F 0x52	<US R>	62
RGB LED bar management		
0x1F 0x64	<US d>	64
Send print data		
0x1F 0x65	<US e>	66
Clear the buffer of the datas files stored of the layout configuration		
0x1F 0x67 0x47	<US g G>	67
Retrieve diagnostic data		
0x1F 0x6C	<US I>	69
Save the layout configuration file		



GRAPHICS ADVANCED MODE COMMANDS

0x1B 0x26 0x6C	<ESC & l>	70
Set the page length		
0x1B 0x26 0x6C	<ESC & l>	71
Set the maximum page length		
0x1B 0x2A 0x62	<ESC * b>	72
Activate/Deactivate compression of data		
0x1B 0x2A 0x62	<ESC * b>	73
Receive graphics data		
0x1B 0x2A 0x70	<ESC * p>	75
Set absolute positioning on the X axis		
0x1B 0x2A 0x70	<ESC * p>	76
Set absolute positioning on the Y axis		
0x1B 0x2A 0x70	<ESC * p>	77
Set relative positive positioning on the X axis		
0x1B 0x2A 0x70	<ESC * p>	78
Set relative positive positioning on the Y axis		
0x1B 0x2A 0x70	<ESC * p>	79
Set relative negative positioning on the X axis		
0x1B 0x2A 0x70	<ESC * p>	80
Set relative negative positioning on the Y axis		
0x1B 0x2A 0x74	<ESC * t>	81
Set print resolution, both vertical and horizontal		



TICKET MANAGEMENT COMMANDS

0x1F 0x70

<US p>

Ticket movement

Valid for	PP54
	PP54 EVO

[Format]	Hex	1F	70	n	m	[opt]	[steps]
	ASCII	US	p	n	m	[opt]	[steps]

[Range]	n = 'C', 'E', 'F', 'I', 'K', 'L', 'P', 'R', 'S', 'T', 'V'
	0x30 ≤ m ≤ 0x34
	opt = 0x00, 0x01 (use only with n = 'T')
	0 ≤ steps ≤ 45.9 (use only with n = 'V')

[Description]	Executes a movement command of the ticket in according to the n parameter value as follows:
---------------	---

n	MODE	FUNCTION
'C' (*)	Dual Feeder	Loads the ticket from one of the two feeder paths 1,2 (*)
'E'	Eject	Ejects the ticket out of the device by one of three ejection paths
'F'	Feed	Feeds the ticket to the printing position
'I'	Reset	Executes an hardware reset and eject all tickets present
'K' (*)	Paper Unload	Unloads the ticket and place it in one of the two available feeder paths (*)
'L'	Print plastic cards	Prints a very high quality ticket (to be used for printing on rewritable plastic cards)
'P'	Park	Moves the ticket in one of the parking slot positions or in one of the RFID reading positions
'R'	Resume	Moves the ticket from one parking slot into the device,
'S'	Frontal Load (SCANNER)	Frontal ticket loading
'T'	Print	Prints a ticket
'V'	Verify	Scans the ticket during printing

Note (*): only for models with dual feeder



- m is related to the value of n and specifies the modality of the command as follows.

n = 'C' Dual Feeder

This command selects the feeder from one of the two feeder paths (1,2) of the device to load ticket specifies by m as follows:

m	Feeder Selection
0x30	Load paper from Feeder 1
0x31	Load paper from Feeder 2

- If the Feeder is already selected the return code 0x11.is sent

n = 'E' Ejection / Presentation

This command ejects the ticket out of the device into three different Path; m specifies the eject channel as follows:

m	Ejection Channel
0x30	Frontal, ticket presentation (tickets stop at the mouth to be taken by the user, async event is raised when ticket is taken)
0x31	Frontal, eject ticket
0x32	Rear, eject ticket
0x33	Bottom, eject ticket

- When the ticket is on the mouth and wait to be taken by the user, the device repeats the eject process 5 each 30 sec. in order to repositioning the ticket in the correct position if it has been moved outside by the user.
- If Read Enable, or Eject to different path commands is received while the ticket have to be taken, the device will execute it and move the ticket according to the command. No ticket taken event will be activated.

n = 'F' Feed

This command feeds the ticket in printing position, separating it from the paper stock.

- In case of models with Dual Feeder use before n = 'C'

n = 'I' Reset

This command executes an hardware reset and ejects all tickets inside the device. The way of ticket ejection has specified by m as follows:

m	Direction of ejection
0x30	Only hardware reset
0x31	Frontal, ticket ejection
0x32	Rear, ticket ejection
0x33	Bottom, ticket ejection



n = 'K' Paper Unload

This command unloads the ticket inside one of the two available feeder paths. Feeder selection is specified by m as follows:

m	Feeder Selection
0x30	Unload paper from Feeder 1
0x31	Unload paper from Feeder 2

- If the Feeder is empty the return code 0x03 is sent.

n = 'L' Print plastic card

This command prints the ticket on rewritable plastic card with m = 0x30.

Send another parameter opt that indicates the printing direction as follows:

opt	Layout orientation
0x00	Normal
0x01	180° Rotation

- The commands that perform printing only work if the layout and data files are saved (refer respectively to the [0x1F 0x4F](#) command with n = 'S', to the [0x1F 0x6C](#) command and to the [0x1F 0x64](#) command).
- If the ticket is still inside the device or is partially ejected, it's possible reprint on the same ticket sending a new print command, but only if you have already sent the ticket data using the [0x1F 0x64](#) command

n = 'P' Park

This command moves the ticket in one of the parking slot positions or in one of the RFID reading positions specifies by m as follows:

m	Park Channel
0x30	Parking slot 1 (bottom parking slot)
0x31	Parking slot 2 (rear parking slot)
0x32	Positions the ticket for reading RFID (RFID position 1)
0x33	Positions the ticket for reading RFID (RFID position 2)

- If the Parking slot 2 is not empty the feed command cannot be executed.



n = 'R' Resume

This command moves the ticket from one parking slot into the device, ready to accept any subsequent command. (e.g. Eject, Read enable or Park again). m specifies the parking slot as follows:

m	Park Channel
0x30	Resume from Parking slot 1 (bottom parking slot)
0x31	Resume from Parking slot 2 (rear parking slot)
0x32	Resume from reading position of RFID (RFID position 1)
0x33	Resume from reading position of RFID (RFID position 2)

n = 'S' Frontal Load

This command enables or disables the reading and the front-loading of a ticket. m specifies the type of reading as follows:

m	mode
0x30	Disable read
0x31	Enable read UP
0x32	Enable read BOTTOM
0x33	Load and position the ticket for reading RFID from the frontal load
0x34	Load ticket without reading



n = 'T' Print

This command prints the ticket specifying the orientation layout. m specifies the feed before printing as follows:

m	Feed
0x30	Don't feed the ticket, print the ticket present inside the device
0x31	Feed the ticket if it's not present inside the device (*)

Note (*): If they were sent up to 3 data files, it's possible to print them in a chain of tickets by using the 0x1F 0x70 'T' 0x31 command (with n = 'T' and m = 0x31)

Send another parameter opt that indicates the printing direction as follows:

opt	Layout orientation
0x00	Normal
0x01	180° Rotation

- The commands that perform printing only work if the layout and data files are saved (refer respectively to the [0x1F 0x4F](#) command with n = 'S', to the [0x1F 0x6C](#) command and to the [0x1F 0x64](#) command).
- If the ticket is still inside the device or is partially ejected, it's possible reprint on the same ticket sending a new print command, but only if you have already sent the ticket data using the [0x1F 0x64](#) command.



n = 'V' Verify

This command scans the ticket into the device during printing. m specifies the feed before printing as follows:

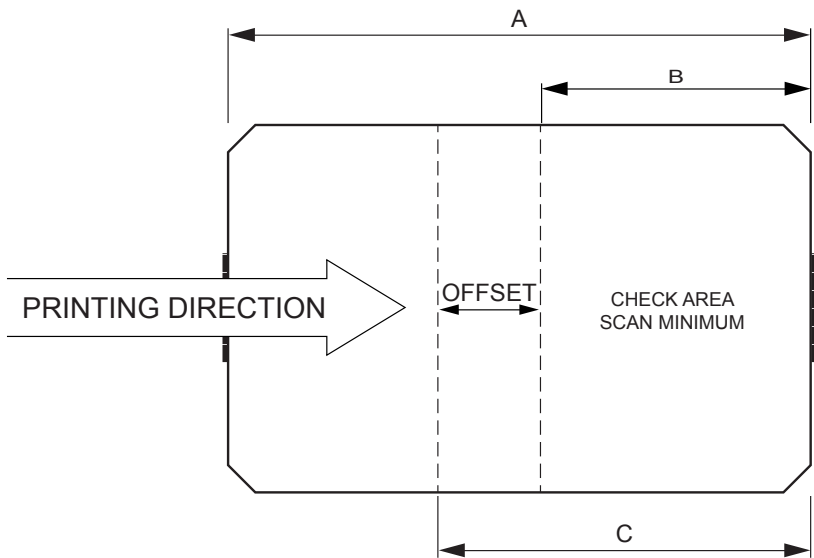
m	Feed
0x30	Don't feed the ticket, print the ticket present inside the device
0x31	Feed the ticket if it's not present inside the device

Send another parameter opt that indicates the printing direction as follows:

opt	Layout orientation
0x00	Normal
0x01	180° Rotation

Send another parameter offset that indicates the feed offset as follows:

offset	Feed
from 00 to 45.9 mm	The feed offset in mm after printing to terminate the scan



where

- A Ticket length defined by parameter "Ticket Len F1" for Feeder 1 or by parameter "Ticket Len F2" for Feeder 2 in the Setup procedure (see the User Manual of the device)
- B Autotmatic area where the scanner perform scanning after printing which is equal to 39.7 mm
- C Test area or scan area where the scanner verify the printed barcode after printing and it is composed by B + the offset value set by offset parameter

To use all the ticket to perform scanning, you must comply with the following equation:
 $C \leq A$



- The commands that perform printing only work if the layout and data files are saved (refer respectively to the [0x1F 0x4F](#) command with n = 'S', to the [0x1F 0x6C](#) command and to the [0x1F 0x64](#) command).
- When reading is enabled, device automatically transport the ticket from the frontal mouth into the reading area and try to read it (automatically perform the necessary retry). If a ticket is already present inside the device and read enable command is sent, device will be perform immediately the reading process of the already present ticket.
- When the reading process is completed, the ticket is returned to the printing position, ready for the next command (e.g. Eject, Print, or Read again).
- If a read disable command is received by device when a ticket is processing (before the reading process is completed) the device have to abort operation and ejects the ticket on the frontal mouth.
- Reading checks the length of the ticket, and rejects a ticket too short or too long, to avoid jamming caused by the insertion of not manageable ticket formats.

[Notes]

- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.
- This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table [RETURN CODES GUIDE](#) at the beginning of this document for further explanation).

[Default]

[Reference]

[0x1F 0x4F](#), [0x1F 0x64](#), [0x1F 0x6C](#)

[Example]



OBJECT MANAGEMENT COMMANDS

0x1F 0x4F

<US O>

Save/Delete command logos, fonts, layouts and script

Valid for	PP54
	PP54 EVO

[Format]	Hex	1F	4F	n	m	[o]	[s1]	[s2]	[s3]	[s4]	d0...dk
	ASCII	US	O	n	m	[o]	[s1]	[s2]	[s3]	[s4]	d0...dk

[Range]

n = 'd', 'D', 'S'
m = 'F', 'L', 'Y', 'S', 'D'
1 ≤ o ≤ 99 (use only with n = 'S')
0x00 ≤ s1, s2, s3, s4 ≤ 0xFF (use only with n = 'S')
0x00 ≤ d0, dk ≤ 0xFF (use only with n = 'S')

[Description]

This command saves or deletes logos, fonts, layouts and scripts; the command is unique for all operations.

- n identify the operation;

n	OPERATION	FUNCTION
'd'	Delete	Deletes the single object specified by m and o
'D'	Delete	Deletes the object specified by m
'S'	Save	Saves the object specified by m

- m identify the object (logo, font, layout and script).

n = 'D' Delete object

This command is used to delete logos, fonts or layouts. m specifies the objects as follows:

m	Operation Type
'F'	Delete all fonts (*.ttf)
'L'	Delete all ticket logos (*.pcx)
'Y'	Delete all layouts (*.lyt)
'S'	Delete the script.spt script file
'D'	Delete all display logos (*.bmp)

- the o parameter identifies the object number that represents the index with which the object was saved in flash.



n = 'S' Save object

This command is used to save logos and fonts. m specifies the objects as follows:

m	mode
'F'	Saves the object as font (.tff)
'L'	Saves the ticket logo as logo (.pcx) ⁽¹⁾
'S'	Saves the script.spt script file
'D'	Saves the display logo as logo (.bmp) ⁽¹⁾⁽²⁾

- The object is saved into the device flash drive at the following path:
 for fonts FLASH DRIVE\TICKET\FONTS\
 for ticket logos FLASH DRIVE\TICKET\LOGOS\
 for script FLASH DRIVE\
 for display logos FLASH DRIVE\DISPLAY\
- the o parameter identifies the object number that represents the index of saving the object in flash.
 If m = 'S' the parameter o must be set to 0.
- the s parameter identifies the object size (4 bytes expressed in hexadecimal notation such as MSB ----->LSB).

[Notes]

- To save the layout configuration using the 0x1F 0x6C command.
- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.
- This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table [RETURN CODES GUIDE](#) at the beginning of this document for further explanation).

[Default]

[Reference] 0x1F 0x6C

[Example]

NOTES:

- (1) : The image must be in black and white (1bit/pixel).
 (2) : The size in pixel of the .bmp image must be 132x29.



STATUS COMMANDS

0x10 0x04

<DLE EOT>

Real-time device status transmission

Valid for	PP54
	PP54 EVO

[Format]	Hex	10	04	17
	ASCII	DLE	EOT	0x17

[Range]

[Description] Transmits the device status at any time.

[Notes]

- Immediately executed even when the data buffer is full.
- This status is transmitted whenever data sequence 0x10 0x04 0x17 is received.
- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.
- The <DATA > bytes is as follows:
Device status (12 bytes)

1st Byte = 0x10 (DLE)

2nd Byte = 0x00 (reserved for events management)

3rd Byte = Operative status

BIT	OFF/ON	HEX	FUNCTION
0	On	01	Device in idle state; the device is ready to process commands
1	On	02	Read ticket enabled
2	On	04	Command in progress
3	On	08	Alarm in progress
4	On	10	Running script
5	On	20	Paper loaded from Feeder 1 ⁽¹⁾
6	On	40	Paper loaded from Feeder 2 ⁽¹⁾
7	-	-	RESERVED

4th Byte = Reserved



5th Byte = Sensors status

BIT	OFF/ON	HEX	FUNCTION
0	On	01	Ticket present in SENS1 (FEEDER IN)
1	On	02	Ticket present in SENS2 (FEEDER OUT)
2	On	04	Ticket present in SENS3 (IN PP54)
3	On	08	Ticket present in SENS4 (OUT1 PP54)
4	On	10	Ticket present in SENS5 (OUT2/LOAD PP54)
5	On	20	Ticket present in SENS6 (Parking slot 1)
6	On	40	Printing head up SENS7
7	On	80	Printing head down SENS8

6th Byte = Sensors status

BIT	OFF/ON	HEX	FUNCTION
0	On	01	Ticket present in SENS9 (Parking slot 2)
1	On	02	Ticket present in SENS10 (FEEDER 1 IN) ⁽¹⁾
2	On	04	Ticket present in SENS11 (FEEDER 1 OUT) ⁽¹⁾
3	On	08	Ticket present in SENS12 (FEEDER 2 IN) ⁽¹⁾
4	On	10	Ticket present in SENS13 (FEEDER 2 OUT) ⁽¹⁾
5	On	20	Diverter DOWN
6	On	40	Cover closed
7	-	-	RESERVED

7th Byte = Sensors status

BIT	OFF/ON	HEX	FUNCTION
0	On	01	Ticket present in SENS9 (Parking slot 2)
1	On	02	Ticket present in RFID position 1 (frontal)
2	On	04	Ticket present in RFID position 2 (rear)
3	-	-	RESERVED
4	-	-	RESERVED
5	-	-	RESERVED
6	-	-	RESERVED
7	-	-	RESERVED

8th Byte = Reserved sensors status

9th Byte = Alarms status

BIT	OFF/ON	HEX	FUNCTION
0	On	01	Paper not present in the paper feeder
1	On	02	Low paper ⁽²⁾
2	On	04	Cover open
3	On	08	Head temperature error
4	On	10	Reception errors (parity, frame error, overrun error)
5	On	20	Power supply voltage error
6	On	40	Command error (command not recognized)
7	On	80	Autocutter error (only for models with autocutter)



10th Byte = Alarms status

BIT	OFF/ON	HEX	FUNCTION
0	On	01	Head error (printing head locked, because it's not be able to go up and down)
1	On	02	Diverter error
2	On	04	Paper jam in the Feeder path
3	On	08	Paper jam in the device path
4	On	10	Paper jam in the Parking slot 1 path
5	On	20	Paper jam in the Parking slot 2 path
6	On	40	Low paper ⁽¹⁾ ⁽²⁾
7	On	80	Paper jam in paper path of Feeder 1 ⁽¹⁾

11th Byte = Alarms status

BIT	OFF/ON	HEX	FUNCTION
0	On	01	Paper jam in paper path of Feeder 2 ⁽¹⁾
1	-	-	RESERVED
2	-	-	RESERVED
3	-	-	RESERVED
4	-	-	RESERVED
5	-	-	RESERVED
6	-	-	RESERVED
7	-	-	RESERVED

12th Byte = Alarms status:

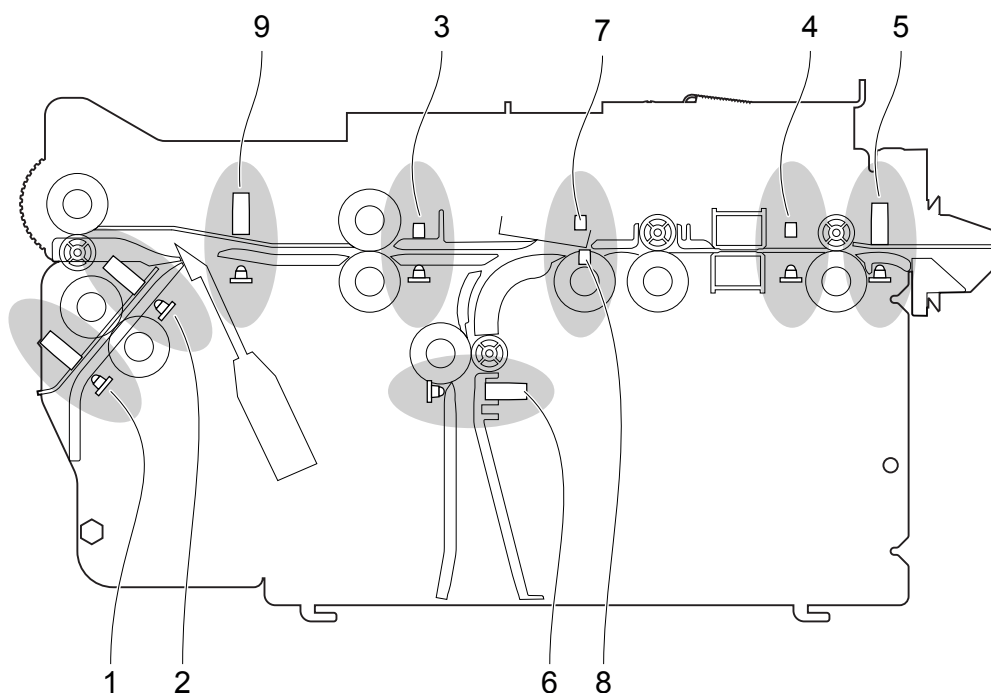
BIT	OFF/ON	HEX	FUNCTION
0	-	-	RESERVED
1	-	-	RESERVED
2	-	-	RESERVED
3	-	-	RESERVED
4	-	-	RESERVED
5	On	20	RAM error
6	On	40	External memory error
7	On	80	FPGA error

NOTES:

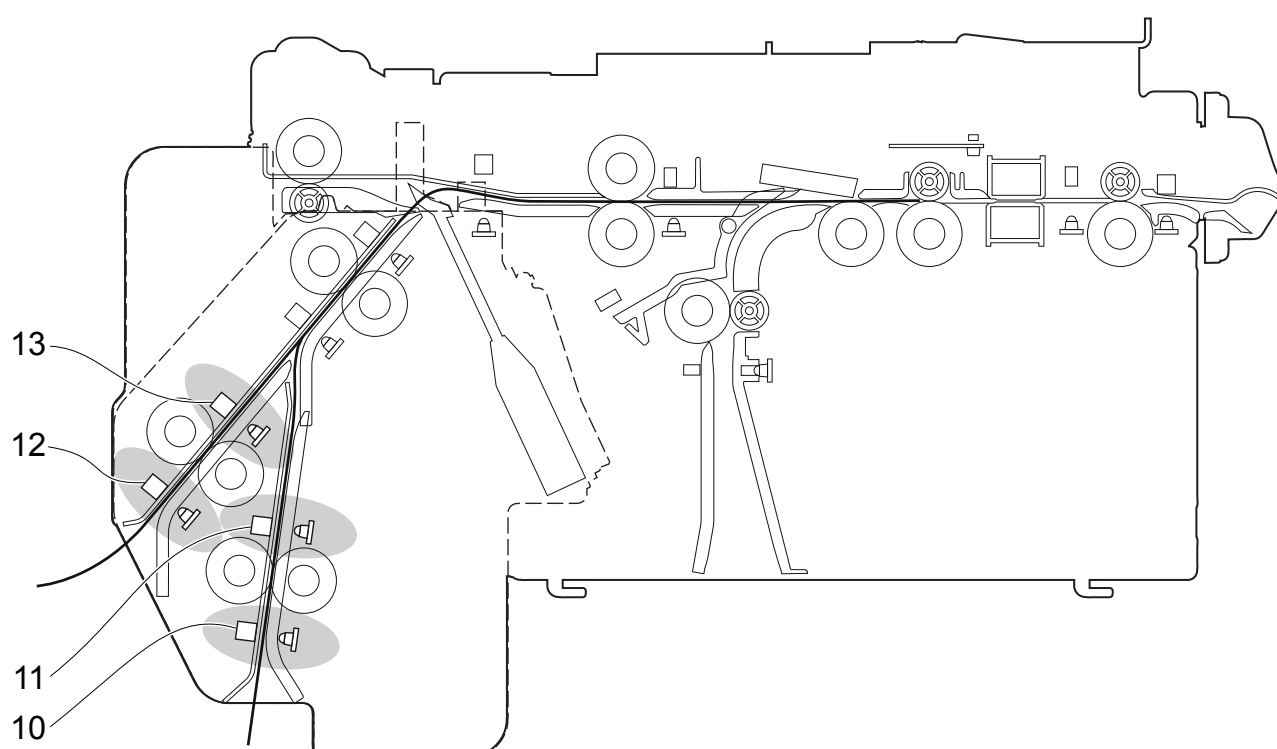
(1) : Only for models with dual feeder.

(2) : Connector for external low paper on control board.

The following figure shows the position of sensors inside the device from SENS1 to SENS9:

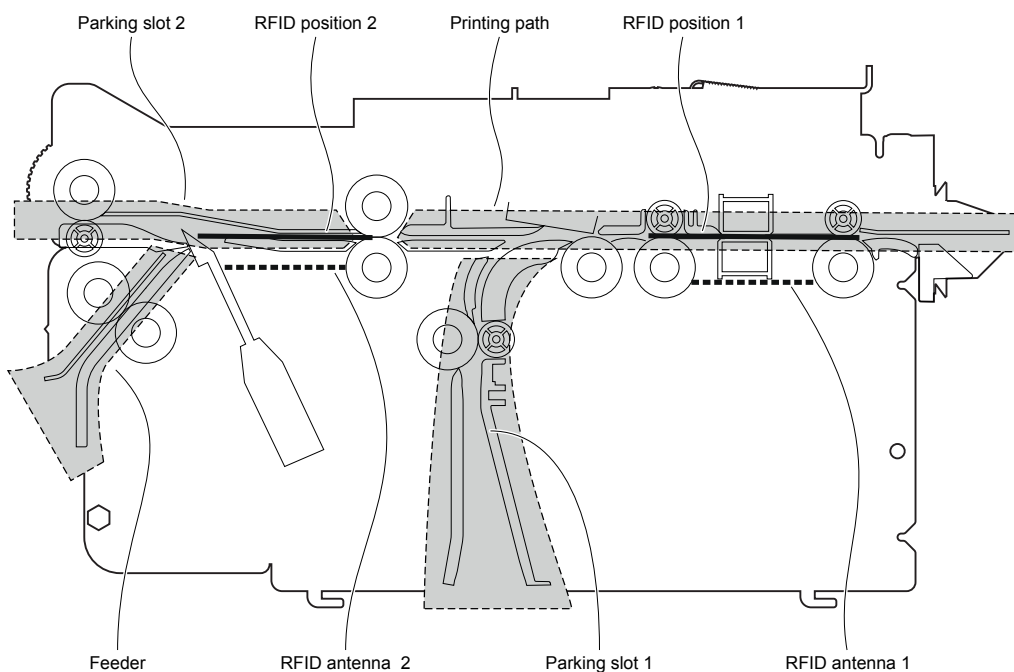


The following figure shows the position of sensors inside the device in the models with Dual Feeder from SENS10 to SENS13:

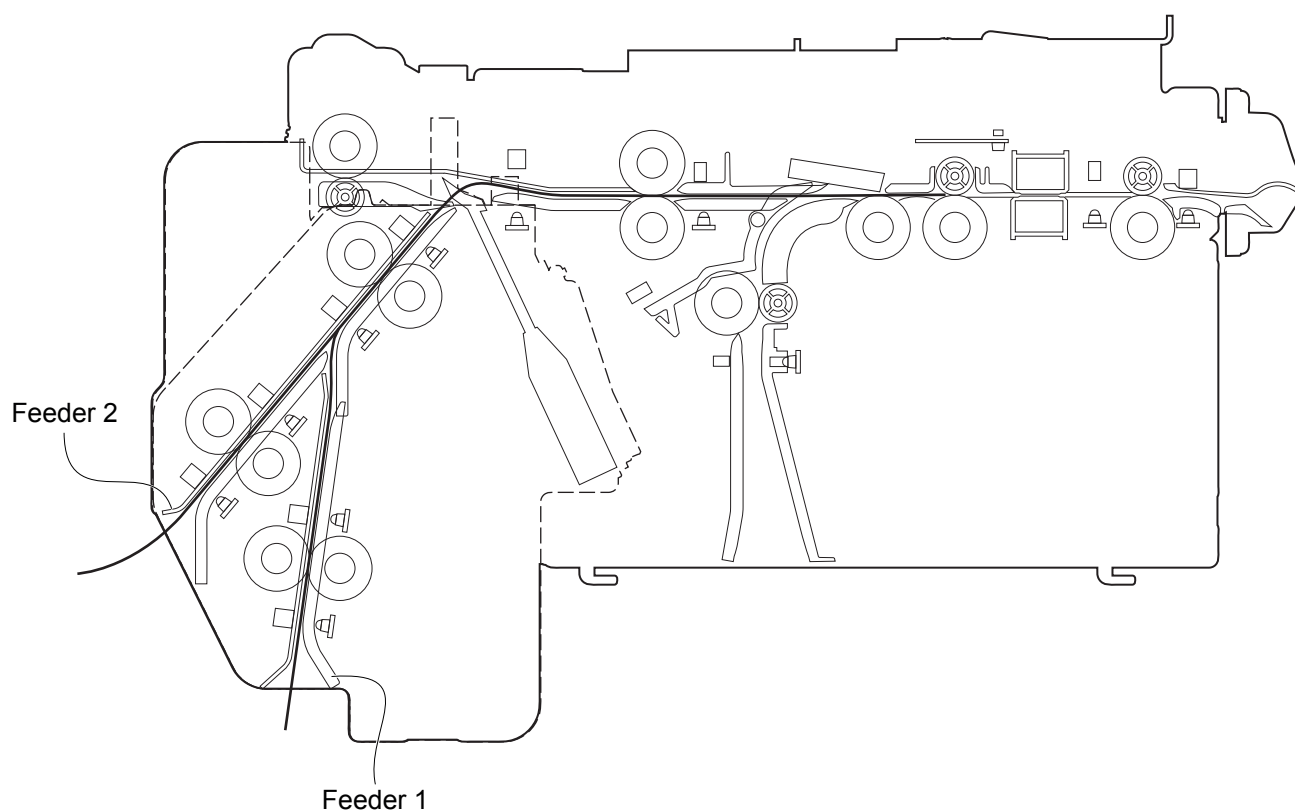




The following figure shows the paper path, the position of sensors and the two RFID antennas inside the device referred from 10th byte:



The following figure shows the two paper paths (1 and 2) inside the device in the models with dual feeder:



[Default]

[Reference]

[Example]



SCANNER COMMANDS

0x1F 0x42

<US B>

Set a barcode search area (ROI)

Valid for	PP54
	PP54 EVO

[Format]	Hex	1F	42	ncis	roiparameters	0x00
	ASCII	US	B	ncis	roiparameters	0x00

where the correct sequence of the roiparameters is:

roi	nbarcode	xH	xL	yH	yL	xDimH	xDimL	yDimH	yDimL
TYPEHH		TYPEHL		TYPELH		TYPELL		timeout	

[Range]	0x01 ≤ roi ≤ 0x05
	0x01 ≤ nbarcode ≤ 0x05
	0x00 ≤ xH ≤ 0xFF, 0 ≤ xL ≤ 0xFF
	0x00 ≤ yH ≤ 0xFF, 0 ≤ yL ≤ 0xFF
	0x00 ≤ xDimH ≤ 0xFF, 0x00 ≤ xDimL ≤ 0xFF
	0x00 ≤ yDimH ≤ 0xFF, 0x00 ≤ yDimL ≤ 0xFF
	0x00 ≤ TypeHH ≤ 0xFF, 0x00 ≤ TypeHL ≤ 0xFF
	0x00 ≤ TypeLH ≤ 0xFF, 0x00 ≤ TypeLL ≤ 0xFF
	0x00 ≤ timeout ≤ 0xFF

[Description]	Set the search areas ROI of barcode (Region of Interest). ncis specifies which CIS reader has been reading as follows:
---------------	---

ncis	FUNCTION
0x30	CIS reader is the TOP
0x31	CIS reader is the BOTTOM

- roi specifies the search area of barcode
- nbarcode specifies the number of barcodes to search inside the search area
- xH and xL specify the X coordinate of area origin point = (xH × 256) + xL
- yH and yL specify the Y coordinate of area origin point = (yH × 256) + yL
- xDimH and xDimL specify the dimension X of area = (xDimH × 256) + xDimL
- yDimH and yDimL specify the dimension Y of area = (yDimH × 256) + yDimL
- TYPEHH, TYPEHL, TYPELH and TYPELL specify 32 bits that allow to enable the search for given types of barcode within the area in according to the following table :

BARCODE TYPE	FLAG BIT
DATAMATRIX (2D)	Bit1
QRcode (2D)	Bit2
AZTEC (2D)	Bit3
PDF 417 (2D)	Bit5



CODE39 (1D)	Bit10
ITF (1D)	Bit11
CODABAR (1D)	Bit12
CODE128 (1D)	Bit13
CODE93 (1D)	Bit14
EAN13 (1D)	Bit17

- timeout specifies the timeout value for a barcode search, is expressed in multiples of 50 ms., for example if set 1 it will be 50 ms.

- [Notes]
- The ROI area is a rectangle specified by the position of the top-left corner point and the width and height identified by xDimH, xDimL, yDimH e yDimL.
 - The values of parameters xH, xL, yH, yL, xDimH, xDimL, yDimH and yDimL must be expressed in dot (1mm = 12 dot).
 - To cancel an area send this command with barcode number equal to 0
US 'B' <roi> 0x00
 - See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.
 - This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table [RETURN CODES GUIDE](#) at the beginning of this document for further explanation).

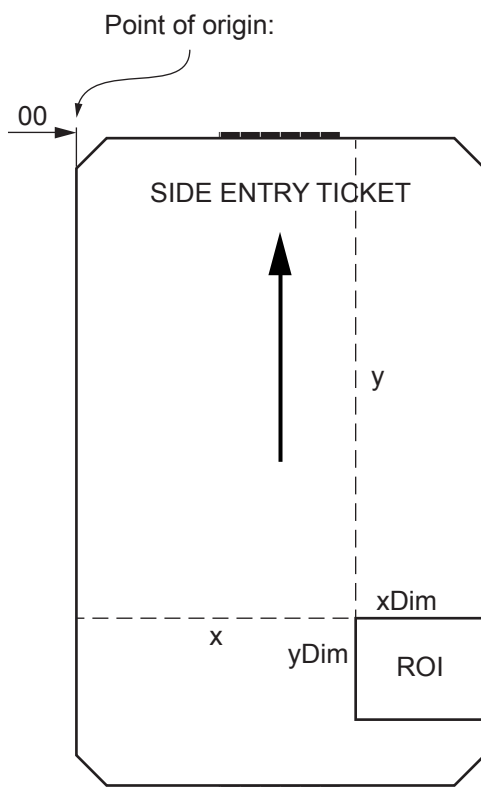
[Default]

[Reference]

[Example] For example to have an area **roi** 1, enable DataMatrix and PDF 417 barcode to search, transmit:

0x1F 0x42	0x30	0x01	0x01	0x01	0xCC	0x02	0xD0	0x00	0xA8	0x00	0xA8	0x00 0x00 0x00 0x22	0x1E	0x00
Command	Scanner position (Top scanner)	Roi index	Number of barcodes to search in that area	xH	xL	yH	yL	xDimH	xDimL	yDimH	yDimL	Barcode Type to search (4 bytes, see table Bit1 enabled and Bit5 enabled = DataMatrix and PDF417 Search)	Timeout for single barcode search in 1/50ms (must be a multiple of 50)	not used

The following figure shows an example of the position of the research area of barcode defined as ROI:





0x1F 0x44

<US D>

Enable debug features

Valid for PP54
PP54 EVO

[Format] Hex 1F 44 n
ASCII US D n

[Range] $1 \leq n \leq 3$

[Description] Debugs the barcode read in base of n value:

BIT	OFF/ON	HEX	DECIMAL	FUNCTION
0	Off	00	0	Disable barcode creation log file
	On	01	1	Enable barcode creation log file
1	Off	00	0	Disable creation image file
	On	02	2	Enable creation image file
2	Off	00	0	Disable creation log file of device status
	On	04	4	Enable creation log file of device status
3	-	-	-	RESERVED
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

- If the Bit0 is set to 1 enables the creation of a log file into the RAM of the device where puts some information of the barcode found.
- If the Bit1 is set to 1 enables the creation of a scanned image.
- If enable both of the bit (Bit0 and Bit1), the image scanned shows the research areas of barcode (ROI) and the coordinates of the corners of the barcode found.

[Notes]

- This command is used for the SW development that manages the device.
- When the device is turned on the default setting is no file is created.
- Debug operation is performed during the ticket reading, so this command have effect from the next scanning.
- The content of the RAM disk is cleared when the printer is reset or the power is turned off.
- The image and log file can be found in 'LASTSCANDRV' disk (disk that is created when you send this command)
- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.
- This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table [RETURN CODES GUIDE](#) at the beginning of this document for further explanation).

[Default] n = 0

[Reference]

[Example]



0x1F 0x62 0x00

<US b>

Reading number of barcode for each area

Valid for	PP54				
	PP54 EVO				

[Format]	Hex	1F	62	ncis	0x00
	ASCII	US	b	ncis	0x00

[Range]	$0x30 \leq ncis \leq 0x31$
---------	----------------------------

[Description]	Reads how many barcode are found in each area. ncis specifies which CIS reader has been reading as follows:
---------------	--

ncis	FUNCTION
0x30	CIS reader is the top
0x31	CIS reader is the bottom

The command return returns 5 bytes as follows:

byte number	FUNCTION
1st	number of barcode into ROI 1
2nd	number of barcode into ROI 2
3rd	number of barcode into ROI 3
4th	number of barcode into ROI 4
5th	number of barcode into ROI 5

[Notes]	• See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command. • This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table RETURN CODES GUIDE at the beginning of this document for further explanation).
---------	--

[Default]

[Reference]

[Example]



0x1F 0x62

<US b>

Return the barcodes of specified area

Valid for	PP54
	PP54 EVO

[Format]	Hex	1F	62	ncis	roi	nbarcode
	ASCII	US	b	ncis	roi	nbarcode

[Range]	0x30 ≤ ncis ≤ 0x31
	0x01 ≤ roi ≤ 0x05
	0x01 ≤ nbarcode ≤ 0x05
	nbarcode = 0xFF

[Description]	Returns the number of barcodes of a specified area.
	• ncis specifies which CIS reader has been reading as follows:

ncis	FUNCTION
0x30	CIS reader is the top
0x31	CIS reader is the bottom

- roi specifies the research area of barcode
- nbarcode specifies the ID of the barcode to read. If nbarcode = 0xFF this command return all barcodes data of a specified area.

The command returns for the specified area ROI this information:

- TYPEHH, TYPEHL, TYPELH and TYPELL specify 32 bits that allow to enable the search for given types of barcode within the area in according to the following table :

BARCODE TYPE	FLAG BIT
DATAMATRIX (2D)	Bit1
QRcode (2D)	Bit2
AZTEC (2D)	Bit3
PDF 417 (2D)	Bit5
CODE39 (1D)	Bit10
ITF (1D)	Bit11
CODABAR (1D)	Bit12
CODE128 (1D)	Bit13
CODE93 (1D)	Bit14
EAN13 (1D)	Bit17

- Q specifies the quality of the error
- xHc0 and xLc0 specify the position X of the point c0 to represent
- yHc0 and yLc0 specify the position Y of the point c0 to represent
- xHc1 and xLc1 specify the position X of the point c1 to represent
- yHc1 and yLc1 specify the position Y of the point c1 to represent
- xHc2 and xLc2 specify the position X of the point c2 to represent
- yHc2 and yLc2 specify the position Y of the point c2 to represent
- xHc3 and xLc3 specify the position X of the point c3 to represent
- yHc3 and yLc3 specify the position Y of the point c3 to represent

- DimH and DimL specify the size of data coded in the barcode
- data specifies the data of the barcode

[Notes]

- The Q specifies the quality of the barcode (from 0 to 100), where 100 identifies the maximum quality (no error correction is used) and 0 is the minimum quality.
- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.
- This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table [RETURN CODES GUIDE](#) at the beginning of this document for further explanation).

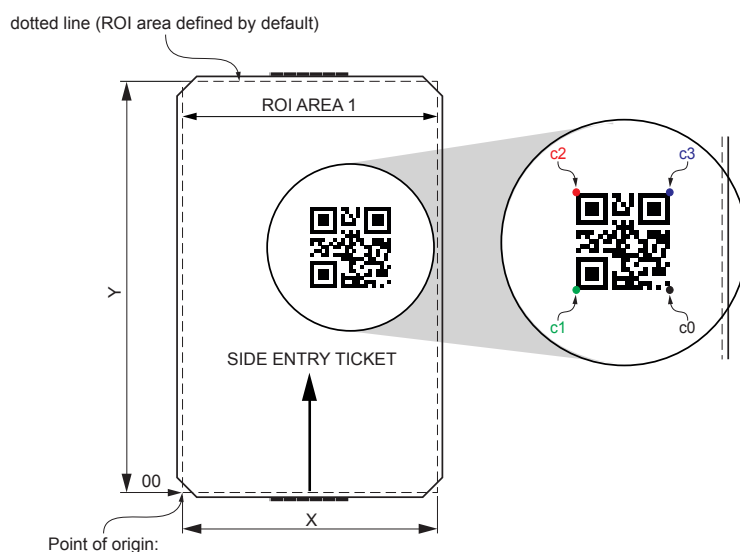
[Default]

[Reference]

0x1F 0x42

[Example]

The following figure shows an example of coordinates defined and the area of barcode:



To read the barcode showed previously, can be sent this command:

VALUE	FIELD DESCRIPTION
0x1F 0x62	Command
0x30	ncis
0x01	roi (roi index)
0x01	nbarcode (barcode index)

The printer will answer the following data stream, below the explanation of the given data:

VALUE	FIELD DESCRIPTION
0x02	STX
0x00 0x1D	Data length
0x06	ACK
0x00 0x00 0x00 0x04	4 Bytes, Barcode type QRcode
0x54	Barcode quality
0x01	xHc0
0x85	xLc0
0x01	yHc0
0x5F	yLc0
0x01	xHc1



0x0B	xLc1
0x01	yHc1
0x60	yLc1
0x01	xHc2
0x10	xLc2
0x00	yHc2
0xE5	yLc2
0x01	xHc3
0x83	xLc3
0x00	yHc3
0xE5	yLc3
0x00 0x05	Barcode data length (DimH and DimL)
0x45 0x44 0x43 0x42 0x41	Barcode Data
0x9C	LCR
0x03	ETX



I/O COMMANDS

0x1F 0x69

<US i>

I/O status transmission

Valid for	PP54
	PP54 EVO

[Format]	Hex	1F	69
	ASCII	US	i

[Range]

[Description] Transmits the device I/O status.

[Notes]

- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.
- The <DATA > bytes is as follows:

I/O status (2 bytes)

1st Byte = Input status as follows:

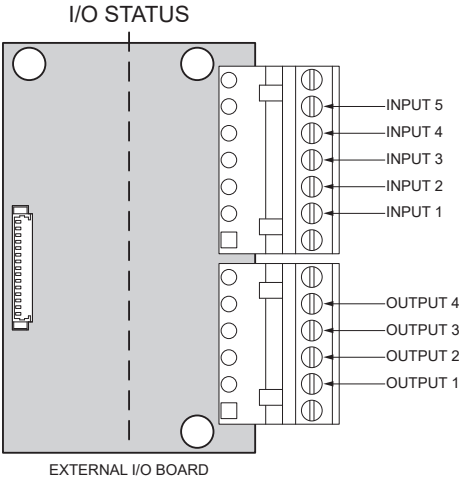
BIT	OFF/ON	HEX	DECIMAL	FUNCTION
0	Off	00	0	INPUT1 Off
	On	01	1	INPUT1 On
1	Off	00	0	INPUT2 Off
	On	02	2	INPUT2 On
2	Off	00	0	INPUT3 Off
	On	04	4	INPUT3 On
3	Off	00	0	INPUT4 Off
	On	08	8	INPUT4 On
4	Off	00	0	INPUT5 Off
	On	10	16	INPUT5 On
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED



2nd Byte = Output status as follows:

BIT	OFF/ON	HEX	DECIMAL	FUNCTION
0	Off	00	0	OUTPUT1 Off
	On	01	1	OUTPUT1 On
1	Off	00	0	OUTPUT2 Off
	On	02	2	OUTPUT2 On
2	Off	00	0	OUTPUT3 Off
	On	04	4	OUTPUT3 On
3	Off	00	0	OUTPUT4 Off
	On	08	8	OUTPUT4 On
4	-	-	-	RESERVED
5	-	-	-	RESERVED
6	-	-	-	RESERVED
7	-	-	-	RESERVED

The following figure shows the position of the I/O pin on the external input/output board referred from 1st to 2nd byte (from INPUT1 to INPUT5 and from OUTPUT1 to OUTPUT4):



[Default]

[Reference]

[Example]

0x1F 0x6F

<US o>

Set I/O output

Valid for	PP54					
	PP54 EVO					

[Format]	Hex	1F	6F	out	on	time
	ASCII	US	o	out	on	time

[Range]	$0x01 \leq \text{out} \leq 0x04$
	$0x00 \leq \text{on} \leq 0x01$
	$0x00 \leq \text{time} \leq 0xFF$

[Description]	Sets an output of the device I/O specified as follows • out parameter indicates the output from 1 to 4 • on parameter indicates the state of output as follows:
---------------	--

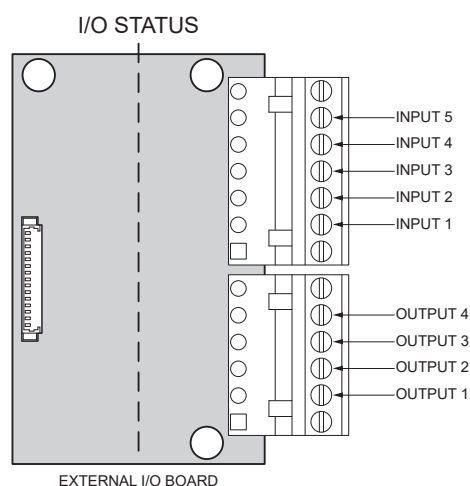
on	FUNCTION
0x00	Reset the output
0x01	Set the output

• time parameter specifies the timeout in millisecond after which the output returns to its previous state as follows:

if time = 0 no timeout;

if time is not equal to 0, after a time of <time> ms the output returns to its previous state.

The following figure shows the position of the I/O pin on the external input/output board (from OUTPUT1 to OUTPUT4):



[Notes]

[Default]

[Reference]

[Example]



LCD COMMANDS

0x1D 0xDA

<GS 0xDA>

Clear LCD

Valid for	PP54
	PP54 EVO

[Format]	Hex	1D	DA	n
	ASCII	GS	0xDA	n

[Range]	$0x30 \leq n \leq 0x31$ $n = 0x43$
---------	---------------------------------------

[Description]	This command clears the LCD in base of n value as follows:
---------------	--

n	FUNCTION
0x30	Clears the LCD display and tuns off LCD backlight
0x31	Tuns on LCD backlight
0x43	Clears the LCD display

[Notes]

[Default]

[Reference]

[Example]



0x1D 0xDA

<GS 0xDA>

Write a message on LCD

Valid for	PP54
	PP54 EVO

[Format]	Hex	1D	DA	n	d0...dk
	ASCII	GS	0xDA	n	d0...dk

[Range]	$0x41 \leq n \leq 0x42$ $0x00 \leq d0, dk \leq 0xFF$
---------	---

[Description]	- d0...dk are the ASCII characters relative to the text to write. - n defines the message position according to the following table:
---------------	--

n	FUNCTION
0x41	Write a message on the first line on the display
0x42	Write a message on the second line on the display

[Notes]	- It is only possible write one line at a time. - The string can not be longer than 20 characters. - The message must be 20 characters long, or must end with NULL.
---------	---

[Default]

[Reference]

[Example]	To write the message "HELLO" in the first line, transmit:
-----------	---

0x1D	0xDA	0x41	0x48	0x45	0x4C	0x4C	0x4F	0x00
command		n = 41	"H"	"E"	"L"	"L"	"O"	(NULL)



0x1D 0xDA 0x4C

<GS 0xDA L>

Show an image on LCD

Valid for	PP54				
	PP54 EVO				
[Format]	Hex	1D	DA	4C	filename
	ASCII	GS	0xDA	L	filename
[Range]					
[Description]	Shows an image on LCD specified by filename. • The images to display are the bitmap files that are saved into the device flash disk at the following path: FLASH DRIVE\DISPLAY\				
[Notes]	• Command valid only for B/W .bmp images. • The image must be in black and white (1bit/pixel). • The size in pixel of the .bmp image must be 132x29.				
[Default]					
[Reference]	0x1F 0x4F				
[Example]	For example to display on LCD the bitmap image “01.bmp” transmit: 0x1D 0xDA 0x4C “01.bmp”				



0x1D 0xDA 0x63

<GS 0xDA c>

Set LCD contrast

Valid for	PP54				
	PP54 EVO				
[Format]	Hex	1D	DA	63	n
	ASCII	GS	0xDA	c	n
[Range]	$0x00 \leq n \leq 0x64$				
[Description]	Sets LCD contrast (adjustable from 0% to 100%)				
[Notes]	The setting value is saved into device Flash.				
[Default]	40% (0x28)				
[Reference]					
[Example]					



MISCELLANEOUS COMMANDS

0x1B 0x3D 0x31

<ESC = 1>

Disable RFID Pass-Through

Valid for	PP54
	PP54 EVO

[Format]	Hex	1B	3D	31	F1	5A	E0
	ASCII	ESC	=	1	±	Z	0xE0

[Range]

[Description] Disables the Pass-Through toward RFID module.

[Notes]

- When the Pass-through function is disabled, all transmitted data are sent on the printer.
- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.

[Default]

[Reference]

[Example]



0x1B 0x3D 0x35

<ESC = 5>

Enable RFID Pass-Through

Valid for	PP54			
	PP54 EVO			
[Format]	Hex	1B	3D	35
	ASCII	ESC	=	5
[Range]				
[Description]				
Enables the Pass-Through toward RFID module.				
[Notes]				
• When the Pass-through function is enabled, all transmitted data are sent on the internal serial port. • See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command.				
[Default]				
[Reference]				
[Example]				



0x1C 0x80

Read date/time of the real time clock

Valid for	PP54
	PP54 EVO

[Format]	Hex	1C	80	m
	ASCII	FS	0x80	m

[Range]	$0x00 \leq m \leq 0x03$
---------	-------------------------

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

m	FORMAT
0x00	DD/MM/YY hh:mm:ss
0x01	DDMMYYhhmmss
0x02	YYMMDDhhmmss
0x03	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 minutes of the time
ss	=	represents the seconds of the time
d	=	represents the day of the week

[Notes]	• See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command.
---------	---

[Default]	
-----------	--

[Reference]	
-------------	--

[Example]	To read date/time in the “DDYYMMhhmmss” format, transmit:
-----------	---

Host

HEX	0x1C	0x80	0x01
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	0x06	0x0C	0x31	0x35	0x30	0x39	0x30	0x36	0x31	0x30	0x35	0x36	0x32	0x30
ASCII	ACK	FF	1	5	0	9	0	6	1	0	5	6	2	0



0x1C 0x81

Set date/time of the real time clock

Valid for	PP54					
	PP54 EVO					

[Format]	Hex	1C	81	m	n	d0...dn
	ASCII	FS	0x81	m	n	d0...dn

[Range]	0x00 ≤ m ≤ 0x03					
	0x00 ≤ d0, dn ≤ 0xFF					

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

m	FORMAT
0x00	DD/MM/YY hh:mm:ss
0x01	DDMMYYhhmmss
0x02	YYMMDDhhmmss
0x03	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 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

[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. • The setting is stored in the EEPROM, and continues to be stored even when the printer is switched off. • In models without real time clock, date / time it is updated automatically only printer on; in case of power off the date / time it is not updated. • See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command.					
---------	--	--	--	--	--	--

[Default]

[Reference]

[Example]	To set date and time to "29 September 2006 at 13:51:00 (PM)" in the "YYMMDDhhmmss" format, transmit:					
-----------	--	--	--	--	--	--

Host

Hex	0x1C	0x81	0x02	0x0C	0x30	0x36	0x30	0x39	0x32	0x39	0x31	0x33	0x35	0x31	0x30	0x30
ASCII	FS	{ }	STX	FF	0	6	0	9	2	9	1	3	5	1	0	0



0x1C 0xEA

Transmit the device serial number

Valid for	PP54			
	PP54 EVO			
[Format]	Hex	1C	EA	n
	ASCII	FS	0xEA	n
[Range]	n = 0x52, 0x72			
[Description]	Transmits the device serial number.			
[Notes]	• The serial number is a string of 16 alphanumeric characters. • If the printer serial number is not defined, the device returns a string of 16 characters with a value of 0x00.			
[Default]				
[Reference]				
[Example]	To read the device serial number the command sequence is: 0x1C 0xEA 0x52			
	The device returns a string of 16 alphanumeric characters just like the following: 'ABC0123456789012'			



0x1D 0x7C

Set printing density

Valid for	PP54			
	PP54 EVO			

[Format]	Hex	1D	7C	n
	ASCII	GS	0x7C	n

[Range]	$0x00 \leq n \leq 0x08$
	$0x30 \leq n \leq 0x38$

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

n	PRINTING DENSITY
0x00, 0x30	- 50%
0x01, 0x31	- 37.5%
0x02, 0x32	- 25%
0x03, 0x33	- 12.5%
0x04, 0x34	0%
0x05, 0x35	+ 12.5%
0x06, 0x36	+ 25%
0x07, 0x37	+ 37.5%
0x08, 0x38	+ 50%

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

[Default]	n = 0x04
-----------	----------

[Reference]	
-------------	--

[Example]	
-----------	--



0x1D 0xF0

Set printing speed

Valid for	PP54
	PP54 EVO

[Format]	Hex	1D	F0	n
	ASCII	GS	0xF0	n

[Range]	0x00 ≤ n ≤ 0x02
---------	-----------------

[Description]	Sets printing speed. n specifies the printing speed as follows:
---------------	---

n	PRINTING SPEED
0x00	High quality
0x01	Normal
0x02	High speed

[Notes]	Printing speed reverts to the default value when the printer is reset or turned off.
---------	--

[Default]	n = 0x02
-----------	----------

[Reference]	
-------------	--

[Example]	
-----------	--



0x1F 0x49

<US I>

Transmit printer ID

Valid for	PP54 PP54 EVO			
[Format]	Hex	1F	49	n
	ASCII	US	I	n
[Range]	n = 0x01, 0x03, 0x31, 0x33 n = 0xFF			
[Description]	Transmits the printer ID specified by n follows:			
	n	PRINTER ID		SPECIFICATION
	0x01, 0x31	Printer model ID (1 byte)		0xFF (resend the command with n=255)
	0x03, 0x33	Firmware release		Depends on Firmware release (4 character)
	0xFF	Printer model ID (2 bytes)		0x02 0x38 (PP54)
[Notes]	• See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command.			
[Default]				
[Reference]				
[Example]				



0x1F 0x4C

<US L>

Retrieve log file

Valid for	PP54
	PP54 EVO

[Format]	Hex	1F	4C	n	[dimH]	[dimL]	[indexH]	[indexL]
	ASCII	US	L	n	[dimH]	[dimL]	[indexH]	[indexL]

[Range]	n = 'P', 'S', 'G'
	0x00 ≤ dimH, dimL ≤ 0xFF (use only with n = 'S')
	0x00 ≤ indexH, indexL ≤ 0xFF (use only with n = 'G')

[Description]	This command retrieves the log file status; the maximum dimension of this log is 64Kbyte. • n specifies the log reading mode as follows:
---------------	---

n	READING MODE	FUNCTION
'P'	Get log	Get all the log file in a single page
'S'	Set log page size	Set the dimension of the log page where dimH and dimL identify the size of the receive buffer to be set (*). Device will reply with the number of packets of fixed size.
'G'	Get log page	Get log page where indexH and indexL are the index (2 bytes) of the log package that you want to receive. Each page will have size previously set with the command 0x1F 0x4C 'S'. If the index is out of range, the command will return error.

Note (*): An ACK byte will be included as data into every log page, it must be considered during log page size setting.

- The log file must be enabled with the [0x1F 0x44](#) command and the datas structure returned is:

ACK (0x06)
Row of the Log 1
...
Row of the Log n

- The printer returns the log file and the structure of the rows of this file is as follows
[S] DATE TIME > STATUS > MESSAGE

where :

'[S]'	is the identifier of the status row and it always at the beginning of the row;
DATE	is the date in the format specified by the command 0x1C 0x81;
TIME	is the time in the format specified by the command 0x1C 0x81;
'>'	is the character separator (in ASCII) of this command;
STATUS	is the device status (12 bytes) that represents the sequence of bytes returned by the command 0x10 0x04 0x17;
MESSAGE	can take different values as follows:



MESSAGE	FUNCTION
Null	No alarm active
'(ALARM)'	In that moment an alarm was active
'(TRIGGER EVENT)'	Moment when there was an alarm and the device has sent to the host the 0x0A event

[Notes]

- The command should be used immediately after receiving the 0x0A event that identifies the occurrence of an alarm (paper end, paper jam etc.).
- See paragraph [EVENTS MANAGEMENT](#) at the beginning of this manual for the table of events managed by the device.
- Once enable the creation log file of device status with the command [0x1F 0x44](#), for each change of at least one of the bits which compose the required, a new row is added to the log.
Note: The value of the bits within each byte is indicated for the 0x10 0x04 0x17 command.
- See paragraph [ANSWER PROTOCOL FORMAT](#) at the beginning of this manual for the response transmitted by the command.

[Default]

[Reference]

[0x1F 0x44](#), [0x10 0x04](#)

[Example]

For example if the “Paper end” occurs the printer’s answer is as follows:

[S]20/07/15 16:40:31,043 > 0x10 0x00 0x00 0x00 0x42 0x41 0x00 0x00 0x00 0x00 0x00 0x00	Printer OK
[S]20/07/15 16:40:31,043 > 0x10 0x0A 0x00 0x00 0x42 0x41 0x00 0x00 0x01 0x00 0x00 0x00 > (TRIGGER EVENT)	“Paper end” occurs, received the 0x0A event
[S]20/07/15 16:40:31,043 > 0x10 0x00 0x08 0x00 0x42 0x41 0x00 0x00 0x01 0x00 0x00 0x00 > (ALARM)	“Paper end” still active
[S]20/07/15 16:42:30,084 > 0x10 0x00 0x05 0x00 0x43 0x40 0x00 0x00 0x00 0x00 0x00 0x00	After feed the paper there are no alarms active



0x1F 0x52

<US R>

RGB LED bar management

Valid for	PP54 PP54 EVO					
[Format]	Hex	1F	52	n	m	d1...dk
	ASCII	US	R	n	m	d1...dk
[Range]	n = 'B', 'C', 'D', 'F', 'R' m = 'h', 'r', 'R', 's', 'w', 'W' 0x00 ≤ d ≤ 0xFF					
[Description]	Set colours for the LED bar depending on parameters values as follows: n = 'B' set the "background" colour of the LED bar. n = 'C' immediately turn on the LED bar with colour set by parameter m = 's' n = 'D' set the default values n = 'F' set the "foreground" colour of the LED bar. n = 'R' set the "rainbow" mode m = 'h' turn off the LED bar m = 'r' read the value in RAM m = 's' write the value in RAM and then the settings are effective until the printer is reset or the power is turned off m = 'R' read the value in FLASH m = 'w' write the value in RAM and then the settings are effective until the printer is reset or the power is turned off m = 'W' write the value in FLASH and then the settings continues to be stored even when the printer is switched off.					
	n	m	FUNCTION			
		'w'	writes in RAM the values for d1 = (R)ed value, d2 = (G)reen value, d3 = (B)lue value			
	'B'	'W'	writes in FLASH the values for d1 = (R)ed value, d2 = (G)reen value, d3 = (B)lue value			
	'F'	'r'	reads the value saved in RAM and returns three bytes (RGB): d1 = (R)ed value, d2 = (G)reen value, d3 = (B)lue value			
		'R'	reads the value saved in FLASH and returns three bytes (RGB): d1 = (R)ed value, d2 = (G)reen value, d3 = (B)lue value			
		'h'	turns off the LED bar			
	'C'	's'	turns on the LED bar and write in RAM the values for d1 = (R)ed value, d2 = (G)reen value, d3 = (B)lue value			



'R'	'w'	writes in RAM the value for the “rainbow” mode: d1 = 0x00 disable the “rainbow” mode, d1 = 0x01 enable the “rainbow” mode
	'W'	writes in FLASH the value for the “rainbow” mode: d1 = 0x00 disable the “rainbow” mode, d1 = 0x01 enable the “rainbow” mode
	'r'	reads from RAM the value of the “rainbow” mode and returns a byte: d1 = 0x00 the “rainbow” mode is disabled, d1 = 0x01 the “rainbow” mode is enabled
	'R'	reads from FLASH the value of the “rainbow” mode and returns a byte: d1 = 0x00 the “rainbow” mode is disabled, d1 = 0x01 the “rainbow” mode is enabled
'D'	0	set the default values

[Notes]

- In presentation mode of the ticket, if the “rainbow” mode is enabled , the LED bar create a “rainbow” effect with a continuous sequence of colours.
- The flashing of the LED bar alternates the colours set as “background” and the “foreground”.
- This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed.

[Default]

“foreground” colour = electric violet (d1 = 0x8F, d2 = 0x00, d3 = 0xFF)
“background” colour = green (d1 = 0x00, d2 = 0xFF, d3 = 0x00)

[Reference]

[Example]

- Read the value of the background colour in RAM:
0x1F 0x52 0x42 0x72
- Read the value of the background colour in FLASH:
0x1F 0x52 0x42 0x52
- Write the value of the background colour (red) in RAM:
0x1F 0x52 0x42 0x77 0xFF 0x00 0x00
- Write the value of the background colour (blue) in FLASH:
0x1F 0x52 0x42 0x57 0x00 0x00 0xFF
- Read the value of the foreground colour in RAM:
0x1F 0x52 0x46 0x72
- Read the value of the foreground colour in FLASH:
0x1F 0x52 0x46 0x52
- Write the value of the background colour (green) in RAM:
0x1F 0x52 0x42 0x77 0x00 0xFF 0x00
- Write the value of “rainbow” mode in RAM (enable):
0x1F 0x52 0x52 0x77 0x01
- Set the default values:
0x1F 0x52 0x44 0x00
- Turns on the LED bar with orange colour and writes it in RAM:
0x1F 0x52 0x43 0x73 0xFF 0x80 0x00
- Turns off the LED bar:
0x1F 0x52 0x43 0x68



0x1F 0x64

<US d>

Send print data

Valid for	PP54 PP54 EVO																		
[Format]	Hex	1F	64	02	<L>	<Sc>	<d1>	<Sc>	<d2>	<Sc>	<dn>	<Sc>	03						
	ASCII	US	d	STX	<L>	<Sc>	<d1>	<Sc>	<d2>	<Sc>	<dn>	<Sc>	ETX						
[Range]																			
[Description]	This command sends print data to the device. The command structure is as follows: 0x1F 0x64 0x02 <L> <Sc> <d1> <Sc> <d2> <Sc> <dn> <Sc> 0x03 where: <table><tr><td>L</td><td>2 numerics characters that identify the layout name followed by 2 numerics characters for identification the layout version</td></tr><tr><td>d1, d2, dn</td><td>2 numerics characters that identify the index of the field followed by the contents of the data to be printed</td></tr><tr><td>Sc</td><td>1 ASCII character (no alphanumeric) used as separator between fields</td></tr></table>													L	2 numerics characters that identify the layout name followed by 2 numerics characters for identification the layout version	d1, d2, dn	2 numerics characters that identify the index of the field followed by the contents of the data to be printed	Sc	1 ASCII character (no alphanumeric) used as separator between fields
L	2 numerics characters that identify the layout name followed by 2 numerics characters for identification the layout version																		
d1, d2, dn	2 numerics characters that identify the index of the field followed by the contents of the data to be printed																		
Sc	1 ASCII character (no alphanumeric) used as separator between fields																		
[Notes]	• The print data must be referred to a layout saved with 0x1F 0x6C command (see TICKET LAYOUT paragraph at the end this document for further explanation). • The printer can save up to 3 different print data buffers to be sent at one time for printing multiple concatenated tickets. • See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command. • This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table RETURN CODES GUIDE at the beginning of this document for further explanation).																		
[Default]																			
[Reference]	0x1F 0x70 0x54 0x31 (printing with multiple tickets), 0x1F 0x6C																		



[Example]

To save the print data of the 0101 layout composed of 3 fields where field 01 contains the text, field 02 a barcode and field 03 a logo, send this command:

```
0x1F 0x64 0x02 '0101#01AMSTERDAM# 020123456789#0301#' 0x03
```

where:

0x1F 0x64	command
0x02	start of packet character
0101	Name and layout version on which is based the printing
#	separator character declaration
01AMSTERDAM	Field 01, on the ticket will print the string 'AMSTERDAM'
#	separator character declaration
020123456789	Field 02 with data to be encoded in the barcode. It will be printed on ticket a barcode type specified in the layout file containing the data '0123456789'
#	separator character declaration
0301	Field 03 that allows printing of a logo, in this case, the printed image will be the logo with index 01
#	separator character declaration
0x03	end of packet character



0x1F 0x65

<US e>

Clear the buffer of the datas files stored of the layout configuration

Valid for	PP54		
	PP54 EVO		
[Format]	Hex	1F	65
	ASCII	US	e
[Range]			
[Description]	This command clears the buffer of the data files stored into the RAM of the device (can be up to 3).		
[Notes]			
[Default]			
[Reference]	0x1F 0x64		
[Example]			



0x1F 0x67 0x47

<US g G>

Retrieve diagnostic data

Valid for	PP54			
	PP54 EVO			

[Format]	Hex	1F	67	47
	ASCII	US	g	G

[Range]

[Description] Returns the diagnostic data to collect information about counters value in the following format:
 Kdiag[1] = Kvalue 0x0A...Kdiag[n] = Kvalue 0x0A
 where:
 Kdiag[1] = is the first diagnostic key;
 Kvalue = indicates the value of Kdiag calculated by the device.

The following table shows all the diagnostic keys (Kdiag) returned with this command as follows:

Kdiag	FUNCTION
DIAG_001	number of power on
DIAG_002	number of burster on
DIAG_003	number of diverter on
DIAG_004	number of ticket printed
DIAG_005	number of frontal ticket loading
DIAG_006	number of frontal eject
DIAG_007	number of rear eject
DIAG_008	number of bottom eject
DIAG_015	number of paper jam
DIAG_016	number of cover open
DIAG_017	number of cutter error
DIAG_018	number of print head movements
DIAG_019	number of high voltage error
DIAG_020	number of low voltage error
DIAG_021	number of printhead overheated
DIAG_022	number of driver error
DIAG_26	time of print head movements upwards. Latest value
DIAG_27	time of print head movements upwards. Lower time
DIAG_28	time of print head movements upwards. Higher time
DIAG_29	time of print head movements downwards. Latest value
DIAG_30	time of print head movements downwards. Lower time
DIAG_31	time of print head movements downwards. Higher time



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



0x1F 0x6C

<US I>

Save the layout configuration file

Valid for	PP54 PP54 EVO																											
[Format]	Hex	1F	6C	02	d1...dk	03																						
	ASCII	US	I	STX	d1...dk	ETX																						
[Range]	0x00 ≤ d ≤ 0xFF																											
[Description]	This command save the layout configuration in a file into the internal memory of the device (see TICKET LAYOUT paragraph at the end this document for further explanation). The layout files are saved into the device flash disk at the following path: FLASH DRIVE\TICKET\LAYOUTS\																											
[Notes]	- The data of the layout (d1 ... dk) are a formatted text stream such as reported in the TICKET LAYOUT paragraph at the end this document. - See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command. - This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table RETURN CODES GUIDE at the beginning of this document for further explanation).																											
[Default]																												
[Reference]	0x1F 0x64																											
[Example]	To save the data of the 0101 layout composed of 3 fields where field 01 contains the text, field 02 a barcode and field 03 a logo, send this command: 0x1F 0x6C 0x02 '0101#01CA4021028000808#02B10001009270912#03L00001073200000#' 0x03 where: <table><tr><td>0x1F 0x6C</td><td>command</td></tr><tr><td>0x02</td><td>start of packet character</td></tr><tr><td>0101</td><td>Name and layout version</td></tr><tr><td>#</td><td>separator character</td></tr><tr><td>01CA4021028000808</td><td>Field 01, is a text string in standard encoding (ASCII) with expanded and italic mode on, it will be printed using the font 02 without rotation at coordinates 28,00 and font size of 8 points</td></tr><tr><td>#</td><td>separator character</td></tr><tr><td>02B10001009270912</td><td>Field 02, is a ITF barcode type printed without rotation at coordinates 09,27 9 millimeters high with bars ratio 3:1.</td></tr><tr><td>#</td><td>separator character</td></tr><tr><td>03L00001073200000</td><td>Field 03, is a logo printed without rotation, at coordinates 07,32.</td></tr><tr><td>#</td><td>separator character</td></tr><tr><td>0x03</td><td>end of packet character</td></tr></table>						0x1F 0x6C	command	0x02	start of packet character	0101	Name and layout version	#	separator character	01CA4021028000808	Field 01, is a text string in standard encoding (ASCII) with expanded and italic mode on, it will be printed using the font 02 without rotation at coordinates 28,00 and font size of 8 points	#	separator character	02B10001009270912	Field 02, is a ITF barcode type printed without rotation at coordinates 09,27 9 millimeters high with bars ratio 3:1.	#	separator character	03L00001073200000	Field 03, is a logo printed without rotation, at coordinates 07,32.	#	separator character	0x03	end of packet character
0x1F 0x6C	command																											
0x02	start of packet character																											
0101	Name and layout version																											
#	separator character																											
01CA4021028000808	Field 01, is a text string in standard encoding (ASCII) with expanded and italic mode on, it will be printed using the font 02 without rotation at coordinates 28,00 and font size of 8 points																											
#	separator character																											
02B10001009270912	Field 02, is a ITF barcode type printed without rotation at coordinates 09,27 9 millimeters high with bars ratio 3:1.																											
#	separator character																											
03L00001073200000	Field 03, is a logo printed without rotation, at coordinates 07,32.																											
#	separator character																											
0x03	end of packet character																											



GRAPHICS ADVANCED MODE COMMANDS

0x1B 0x26 0x6C

<ESC & I>

Set the page length

Valid for	PP54					
	PP54 EVO					
[Format]	Hex	1B	26	6C	n1	50
	ASCII	ESC	&	I	n1	P
[Range]	$0 \leq n1 \leq 65535$					
[Description]	Sets the length of the page expressed in points. The resolution used for the length is 300 dpi.					
[Notes]	The n1 parameter is a string variable.					
[Default]						
[Reference]	0x1B 0x26 0x6C n1 0x52					
[Example]	To set a page length of 896 dot (112 mm) the command sequence is the following: 0x1B 0x26 0x6C 0x38 0x39 0x36 0x50					



0x1B 0x26 0x6C

<ESC & I>

Set the maximum page length

Valid for	PP54 PP54 EVO					
[Format]	Hex	1B	26	6C	n1	52
	ASCII	ESC	&	I	n1	R
[Range]	$0 \leq n1 \leq 65535$					
[Description]	Sets the maximum length of the page expressed in points. The resolution used for the length is 300 dpi. The command prints only the part of the document not empty.					
[Notes]	• The n1 parameter is a string variable.					
[Default]						
[Reference]	0x1B 0x26 0x6C n1 0x50					
[Example]	To set a page length of 896 dot (112 mm) the command sequence is the following: 0x1B 0x26 0x6C 0x38 0x39 0x36 0x52					



0x1B 0x2A 0x62

<ESC * b>

Activate/Deactivate compression of data

Valid for	PP54
	PP54 EVO

[Format]	Hex	1B	2A	62	n1	4D
	ASCII	ESC	*	b	n1	M

[Range] n1 = 0x30, 0x31, 0x32

[Description] On the bases of n1 value, the command activates or deactivates the data compression mode that concerns the data content in the 0x1B 0x2A 0x62 n1 0x57 command.

n1	FUNCTION
0x30	Deactivates data compression
0x31	Activates RLE data compression
0x32	Activates FE-RLE data compression

[Notes]

[Default] n1 = 0x30

[Reference] 0x1B 0x2A 0x62 n1 0x57

[Example] To activate the RLE data compression the command sequence is the following: 0x1B 0x2A 0x62 0x31 0x4D



0x1B 0x2A 0x62

<ESC * b>

Receive graphics data

Valid for	PP54 PP54 EVO					
[Format]	Hex	1B	2A	62	n1	57
	ASCII	ESC	*	b	n1	W
[Range]	$0 \leq n1 \leq 65535$					
[Description]	Receives the graphics data.					
[Notes]	The n1 parameter is a string variable. This command receives data either in compressed mode or not, depending on how it was previously set by the 0x1B 0x2A 0x62 n1 0x4D command: • RLE compression: The value n1, does not determine the number of bytes that are sent, but the number of bytes that will position themselves on the graphic. Therefore, it is the total number of uncompressed bytes. Compression works in the following way: • If the byte has a value that is less than 0xC0 hexadecimal (or 192 decimal), these bytes are not compressed. • If the byte has a value that is equal or greater than 0xC0 hexadecimal (or 192 decimal) the 6 least significant bytes represent the subsequent byte's multiplication factor, that can assume any value. The bytes are arranged horizontally on the graphics pages starting at the point at which the pointer is located, the "external" bytes will be ignored. • FE-RLE compression: The value of n1 determine the total number of compressed bytes. Compression works in the following way: • If 2 consecutives bytes are equal, the third byte is multiplication factor that can assume any value. The bytes are arranged horizontally on the graphics pages starting at the point at which the pointer is located, the "external" bytes will be ignored.					
[Default]						
[Reference]	0x1B 0x2A 0x62 n1 0x4D					
[Example]	The commands sequences to add, as a graphic, the following bytes sequence (in hexadecimal) are: 0xE0 0xE0 0xE0 0xE0 0xE0 0xE0 0x20 0x65 0x65 0xC9 0x88 0x88 Not compressed mode: 0x1B 0x2A 0x62 0x31 0x32 0x57 0xE0 0xE0 0xE0 0xE0 0xE0 0xE0 0x20 0x65 0x65 0xC9 0x88 0x88 RLE compression: 0x1B 0x2A 0x62 0x31 0x32 0x57 0xC6 0xE0 0x20 0xC2 0x65 0xC1 0xC9 0xC2 0x88					



where:

[0x1B 0x2A 0x62]	identify the command
[0x31 0x32]	identify that there are 12 bytes of graphics to be inserted
[0x57]	is the W character that identifies that currently 12 bytes of compressed data are pending
[0xC6 0xE0]	0xC6 is greater than 0xC0, the 2 most significant bits of byte 0xC6 are each set to 1, therefore it is compressed: the byte 0xE0 will be taken 6 times
[0x20]	this character is not compressed because $0x20 < 0xC0$
[0xC2 0x65]	0xC2 is greater than 0xC0, the 2 most significant bits of byte 0xC2 are each set to 1, therefore it is compressed: the byte 0x65 will be taken 2 times
[0xC1 0xC9]	0xC1 is greater than 0xC0, the 2 most significant bit of byte 0xC1 are each set to 1, therefore it is compressed: the byte 0xC9 will be taken 1 time
[0xC2 0x88]	0xC2 is greater than 0xC0, the 2 most significant bits of byte 0xC2 are each set to 1, therefore it is compressed: the byte 0x88 will be taken 2 times

FE-RLE compression:

0x1B 0x 2A 0x62 0x31 0x31 0x57 0xE0 0xE0 0x06 0x20 0x65 0x65 0x02 0xC9 0x88 0x88 0x02

where:

[0x1B 0x2A 0x62]	identify the command
[0x31 0x31]	identify that there are 11 bytes of graphics to be inserted
[0x57]	is the W character that identifies that currently 11 bytes of compressed data are pending
[0xE0 0xE0 0x06]	0xE0 is repeated 2 times, followed by 06 byte, therefore it is compressed: the byte 0xE0 will be taken 6 times
[0x20]	this character is not compressed because it appears once
[0x65 0x65 0x02]	0x65 is repeated 2 times, followed by 02 byte, therefore it is compressed: the byte 0x65 will be taken 2 times
[0xC9]	this character is not compressed because it appears once
[0x88 0x88 0x02]	0x88 is repeated 2 times, followed by 0x02 byte, therefore it is compressed: the byte 0x88 will be taken 2 times



<ESC * p>

0x1B 0x2A 0x70

Set absolute positioning on the X axis

Valid for	PP54					
	PP54 EVO					
[Format]	Hex	1B	2A	70	n1	58
	ASCII	ESC	*	p	n1	X
[Range]	$0 \leq n1 \leq 640$					
[Description]	Set the absolute positioning on the X axis (along the horizontal direction of the sheet) respect to the left hand margin of the sheet.					
[Notes]	• The n1 parameter is a string variable. • The resolution remains at 300 dpi.					
[Default]						
[Reference]						
[Example]	To start printing from X = 20, the command sequence is: 0x1B 0x2A 0x70 0x32 0x30 0x58					



0x1B 0x2A 0x70

<ESC * p>

Set absolute positioning on the Y axis

Valid for	PP54 PP54 EVO					
[Format]	Hex	1B	2A	70	n1	59
	ASCII	ESC	*	p	n1	Y
[Range]	0 ≤ n1 ≤ 990					
[Description]	Set the absolute positioning on the Y axis (along the vertical direction of the sheet) respect to the top margin of the sheet.					
[Notes]	• The n1 parameter is a string variable. • The resolution remains at 300 dpi.					
[Default]						
[Reference]						
[Example]	To start printing from Y = 20, the command sequence is: 0x1B 0x2A 0x70 0x32 0x30 0x59					



<ESC * p>

0x1B 0x2A 0x70

Set relative positive positioning on the X axis

Valid for	PP54 PP54 EVO						
[Format]	Hex	1B	2A	70	2B	n1	58
	ASCII	ESC	*	p	+	n1	X
[Range]	$0 \leq n1 \leq [640 - \text{current print position}]$						
[Description]	Set the relative positive positioning on the X axis (along the horizontal direction of the sheet) respect the actual position of the printout.						
[Notes]	• The n1 parameter is a string variable. • The resolution remains at 300 dpi.						
[Default]							
[Reference]							
[Example]	To move towards the right of the actual position on the X axis by 50 dot, the command sequence is: 0x1B 0x2A 0x70 0x2B 0x35 0x30 0x58						



0x1B 0x2A 0x70

<ESC * p>

Set relative positive positioning on the Y axis

Valid for	PP54 PP54 EVO						
[Format]	Hex	1B	2A	70	2B	n1	59
	ASCII	ESC	*	p	+	n1	Y
[Range]	$0 \leq n1 \leq [990 - \textit{current print position}]$						
[Description]	Set the relative positive positioning on the Y axis (along the vertical direction of the sheet) respect the actual position of the printout.						
[Notes]	The n1 parameter is a string variable. The resolution remains at 300 dpi.						
[Default]							
[Reference]							
[Example]	To move downwards by 50 dot from the current position on the Y axis, the command sequence is: 0x1B 0x2A 0x70 0x2B 0x35 0x30 0x59						



0x1B 0x2A 0x70

<ESC * p>

Set relative negative positioning on the X axis

Valid for	PP54 PP54 EVO						
[Format]	Hex	1B	2A	70	2D	n1	58
	ASCII	ESC	*	p	-	n1	X
[Range]	$0 \leq n1 \leq 640$						
[Description]	Set the relative negative positioning on the X axis (along the horizontal direction of the sheet) respect the actual position of the printout.						
[Notes]	The n1 parameter is a string variable. The resolution remains at 300 dpi.						
[Default]							
[Reference]							
[Example]	To move to the left of the actual position on the X axis by 40 dot, the command sequence is: 0x1B 0x2A 0x70 0x2D 0x34 0x30 0x58						



0x1B 0x2A 0x70

<ESC * p>

Set relative negative positioning on the Y axis

Valid for	PP54 PP54 EVO						
[Format]	Hex	1B	2A	70	2D	n1	59
	ASCII	ESC	*	p	-	n1	Y
[Range]	0 ≤ n1 ≤ 990						
[Description]	Set the relative negative positioning on the Y axis (along the vertical direction of the sheet) respect the actual position of the printout.						
[Notes]	• The n1 parameter is a string variable. • The resolution remains at 300 dpi.						
[Default]							
[Reference]							
[Example]	To move upwards by 40 dot to the left of the actual position on the Y axis, the command sequence is: 0x1B 0x2A 0x70 0x2D 0x34 0x30 0x59						



0x1B 0x2A 0x74

<ESC * t>

Set print resolution, both vertical and horizontal

Valid for	PP54 PP54 EVO					
[Format]	ASCII Hex	ESC 1B	* 2A	t 74	n1 n1	R 52
[Range]	n1 = 75, 150, 300					
[Description]	Sets print resolution, both vertical and horizontal at 75, 150 or 300 dots/inch (dpi).					
[Notes]	• The n1 parameter is a string variable.					
[Default]						
[Reference]						
[Example]	To set the resolution at 300 dpi, the command sequence is: 0x1B 0x2A 0x74 0x33 0x30 0x30 0x52					





TICKET LAYOUT

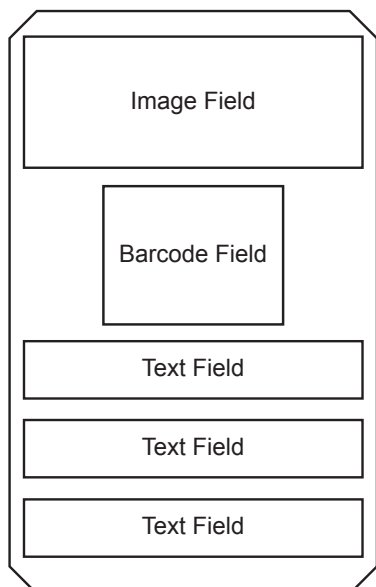
1	TICKET LAYOUT	85
---	---------------------	----



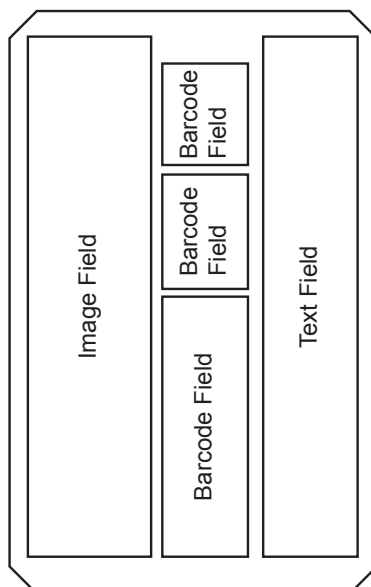
1 TICKET LAYOUT

The information printed on a ticket are organized in layouts. The printable area is 608 x 990 pixel. Layout is a template with an unique ID formed by a variable number of fields of different types (e.g. static and dynamic). Static fields can contain objects previously loaded into the device memory.

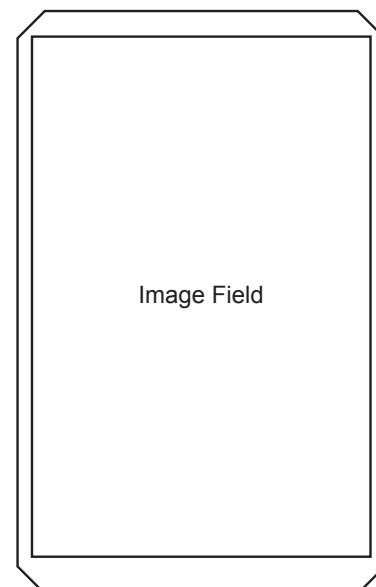
A different numbers of layouts can be stored permanently into the memory of device. The following images shows some examples of possible layouts:



Example 1



Example 2



Example 3

Each layout is a list of record containing the description of each field type and position within the printable area.

Layout structure of the Header

The Layout of the Header must be structured using the following fields that are used to compose the ticket layout.

The following table lists the name of the fields and attributes used.

FIELD NAME	TYPE	DESCRIPTION
Layout Number	Numeric (from 00 to 99)	The layout ID the unique identifier of the layout (00...99)
Layout Version	Numeric (from 00 to 99)	The layout version (00...99)
Separator	Character	The special character # or @ used between fields through whole message

LAYOUT NUMBER

This field specifies the layout ID, unique identification number of the layout composed by two digits.

LAYOUT VERSION

This field specifies the layout version because may have different versions, is represented by two digit number.

SEPARATOR

This field specifies the special character (no alphanumeric) used to separate the elements throughout the layout. When using the special character # the units of measure of vertical and horizontal positioning are expressed in mm, instead if using the special character @ the units of measure of vertical and horizontal positioning are expressed in dot.



Structure of the body of the elements descriptor, the fields structure

It contains the description of the elements. Each element descriptor will have an element descriptor number. An element descriptor describes the size, print location and other attributes of data elements. The host will send data by using element numbers.

The following table contains the descriptor of the fields used and its attributes:

ATTRIBUTES	TYPE	REMARKS
<i>Field Number</i>	2 alphanumerics	
<i>Field Type</i>	1 alphanumeric	
<i>Field Choice</i>	1 alphanumeric	
<i>Field Option_A</i>	1 numeric	
<i>Field Option_B</i>	2 alphanumerics	from 00 to 99
<i>Orientation</i>	1 alphanumeric	
<i>Vertical Position</i>		
<i>Separator #</i>	3 numerics	from 000 to 999 mm
<i>Separator @</i>	3 numerics	from 000 to 999 dot
<i>Horizontal Position</i>		
<i>Separator #</i>	2 numerics	from 00 to 99 mm
<i>Separator @</i>	3 numerics	from 000 to 999 dot
<i>Size Height</i>	2 numerics	from 00 to 99
<i>Size Width</i>	2 numerics	from 00 to 99
<i>Common Data</i>	from 0 to 20 alphanumerics	optional
<i>Separator</i>	1 alphanumeric	#, @

Field Number

The maximum number of element descriptors is 255 whereby the element numbering will be from 01 to FF represented in a two character format (hexadecimal). Elements must be shown in ascending order; however element numbers can not to be consecutives. Valid element numbers are from 01 to FF.

Field Type

- **C**: Character.
- **L**: Predefined Logo
- **B**: Barcode.

Field Choice

- *Field Type C*: defines the type of font.
A = TrueType standard encoding (ASCII)
B = TrueType UTF-8
C = TrueType UTF-16
- *Field Type L*: Reserved, Value is 0.
- *Field Type B*: defines the type of barcode as follows:

0 Code Interleaved 2 of 5 including check-digit (*)	6 PDF 417 (2D)
1 Code Interleaved 2 of 5	7 Code 128 including check-digit (*)
3 Code 39	8 Code 39 including check-digit (*)
4 DataMatrix (2D)	A EAN-13 including check-digit (*)
5 QRcode (2D)	V Aztec (2D)

NOTE (*): The check digit is calculated by the device and should not be entered in the data of barcode.



Field Option_A

This element must be defined and used only for text fields.

- **Field Type C**
 - 0 Not used
 - 1 Normal
 - 2 Emphasized
 - 3 Italic
 - 4 Emphasized/ Italic

Field Option_B

This element must be defined and used only for text fields.

- **Field Type C**
 - Indicates the index of truetype font to use. The index is the name of the file saved in FLASHDRIVE/TICKET/FONTS.
 - The maximum number of fonts that can be saved into device is 99 (from 01 to 99).

Field Orientation

For the left alignment, the horizontal position is defined as the lower left corner of the field.

For the center alignment, the horizontal position is defined as the center of the field.

For the right alignment, the horizontal position is defined as the lower right corner of the field.

For the reverse video, blackened area includes half-character-width to the left, right, on the top, and underneath the printed characters in the field. Width of the character is determined by the distance between the start of one character to the beginning of the next character.

- **Field Type C and L:**
 - Left alignment
 - 1 Rotation 0° / normal printing
 - 2 Rotation 0° / reverse printing
 - 3 Rotation 90° / normal printing
 - 4 Rotation 90° / reverse printing
 - 5 Rotation 180° / normal printing
 - 6 Rotation 180° / reverse printing
 - 7 Rotation 270° / normal printing
 - 8 Rotation 270° / reverse printing
 - Center alignment
 - A Rotation 0° / normal printing
 - B Rotation 0° / reverse printing
 - C Rotation 90° / normal printing
 - D Rotation 90° / reversed printing
 - E Rotation 180° / normal printing
 - F Rotation 180° / reverse printing
 - G Rotation 270° / normal printing
 - H Rotation 270° / reverse printing
 - Right alignment
 - I Rotation 0° / normal printing
 - J Rotation 0° / reverse printing
 - K Rotation 90° / normal printing
 - L Rotation 90° / reverse printing
 - M Rotation 180° / normal printing
 - N Rotation 180° / reverse printing
 - O Rotation 270° / normal printing
 - P Rotation 270° / reverse printing

- **Field Type B:** the value must be as follows:

1	A	I
3	C	K
5	E	M
7	G	O

Note: The barcode can not be printed in reverse, only in normal.



Field Vertical Position

- Field Type **C, L, B**: Defined as lowest point. The range is from 001 to 999, where 1 unit is defined by the type of separator used as follows:

Separator #	1 unit = 1 mm
Separator @	1 unit = 1 dot

Field Horizontal Position

- Field Type **C, L, B**: Defined as leftmost (rightmost/centering) point. The range of values and units of measure are defined by the type of separator used as follows:

Separator	Value range	units of measure	Notes
#	from 01 to 99	1 unit = 1 mm	The maximum value is 50 mm, the maximum width of the print area.
@	from 001 to 999	1 unit = 1 dot	The maximum value is 600 dots the maximum width of the print area.

Field Size Height

- Field Type **C**:
The value range is from 01 to 72 and indicates the character dimension in points.
- Field Type **B**: The value range is from 01 to 99 whereby one unit is equal 1 mm (see Note 3) and indicates the barcode height in mm. Valid for 1D barcodes and PDF417.

For barcode 2D the parameter **Size Height** will be used to specify the symbol size (rows x columns):

DATAMATRIX			symbol size from 1 to 16 ⁽¹⁾		
n	FORMAT	Information capacity	n	FORMAT	Information capacity
1	10 x 10	3	9	26 x 26	44
2	12 x 12	5	10	32 x 32	62
3	14 x 14	8	11	36 x 36	86
4	16 x 16	12	12	40 x 40	114
5	18 x 18	18	13	44 x 44	144
6	20 x 20	22	14	48 x 48	174
7	22 x 22	30	15	52 x 52	204
8	24 x 24	36	16	64 x 64	280

QR CODE			symbol size from 1 to 12 ⁽¹⁾		
n	FORMAT	Information capacity ⁽²⁾	n	FORMAT	Information capacity ⁽²⁾
1	21 x 21 (Version 1)	from 7 to 17	7	45 x 45 (Version 7)	from 64 to 154
2	25 x 25 (Version 2)	from 14 to 32	8	49 x 49 (Version 8)	from 84 to 192
3	29 x 29 (Version 3)	from 24 to 33	9	53 x 53 (Version 9)	from 98 to 230
4	33 x 33 (Version 4)	from 34 to 78	10	57 x 57 (Version 10)	from 119 to 271
5	37 x 37 (Version 5)	from 44 to 106	11	61 x 61 (Version 11)	from 137 to 321
6	41 x 41 (Version 6)	from 58 to 134	12	65 x 65 (Version 12)	from 155 to 367



AZTEC		symbol size from 1 to 15 ⁽¹⁾	
n	FORMAT	n	FORMAT
1	C15 x 15 (Compact)	9	C37 x 37
2	C19 x 19 (Compact)	10	C41 x 41
3	C23 x 23 (Compact)	11	C45 x 45
4	C27 x 27 (Compact)	12	C49 x 49
5	C19 x 19	13	C53 x 53
6	C23 x 23	14	C57 x 57
7	C27 x 27	15	C61 x 61
8	C31 x 31		

NOTES:

(1) : If the parameter's value is 0, the size of the barcode is calculated automatically based on the settings of error and dot size.

(2) : The values are reported as the minimum and maximum and change according to the set error correction value.

Field Size Width

- Field Type C:**

The value is always 00.

- Field Type B:** The value range is from 01 to 99 (see Note 3). This is a number that depending on the type of barcode, specifies different parameters and it will be coded on two digits where the 1st digit specifies magnification and the 2nd digit the bars ratio.

For barcode **1D** the **Size Width** parameter will be used to specify both barcode magnification and ratio:

The first digit specifies magnification:

- 1 = ratio duplicated with 1
- 2 = ratio duplicated with 2
- 3 = ratio duplicated with 3
- ...
- 9 = ratio duplicated with 9

The second digit specifies the bars ratio:

- 1 = 2 : 1
- 2 = 3 : 1
- 3 = 5 : 2
- 4 - 9 not used

SIZE WIDTH	MAGNIFICATION	BARS RATIO	COMMENT	WIDE BAR	NARROW BAR
11	1	2:1	(2:1) x 1	2 dots	1 dot
31	3	2:1	(2:1) x 3	6 dots	3 dots
22	2	3:1	(3:1) x 2	6 dots	2 dots

For barcode **2D** the **Size Width** parameter will be used to specify both barcode error correction level and the point size of the barcode:



The first digit specifies the error correction level of barcode (Default = 0):

CODE TYPE	CORRECTING ERRORS VALUE	PERCENTAGE OF ERROR CORRECTION/ ADDITIONAL CHARACTERS
4 (2D DataMatrix)	0	Automatic
5 QR (2D)	0	Automatic
	1	20% (L)
	2	37% (M)
	3	55% (Q)
	4	65% (H)
6 PDF 417 (2D)	0	Automatic
	1	4
	2	8
	3	16
	4	32
	5	64
	6	128
	7	256
	8	512
V Aztec (2D)	0	Automatic
	1	10%
	2	23%
	3	36%
	4	50%

The second digit specifies the point size of the barcode:

- 1 = 0.25 mm (0.75 mm for PDF417)
- 2 = 0.50 mm (1.125 mm for PDF417)
- 3 = 0.75 mm (1.50 mm for PDF417)
- 4 = 1.00 mm
- 5 = 1.25 mm
- 6 = 1.50 mm
- 7 = 1.75 mm
- 8 = 2.00 mm

Field Common Data

This option may be used to print fixed data which always appears on the layout structure (e.g. fixed text field). Also to add to the input data of a particular element, a fixed character string.

- **Field Type C:** All type of characters, except separator (maximum of 20).
- **Field Type L:** Logo number from 00 to 99.
- **Field Type B:** Maximum 12 characters.



Examples of Barcode Definition

01B10003050051011#

where:

01	B	1	0	00	3	050	05	10	1	1	#
Field number	Field Type = Barcode	Filed Choice = Barcode Interleaved 2 of 5	Field Option A	Field Option B	Orientation = 90° / Normal printing/ left align	Vertical Position	Horizontal Position	Size Height = 10 mm	Magnification = 1	Bars Ratio 2:1	Separator

01B100036000601011@

where:

01	B	1	0	00	3	600	060	10	1	1	@
Field number	Field Type = Barcode	Filed Choice = Barcode Interleaved 2 of 5	Field Option A	Field Option B	Orientation = 90° / Normal printing/ left align	Vertical Position	Horizontal Position	Size Height = 10 mm	Magnification = 1	Bars Ratio 2:1	Separator





SCRIPT INSTRUCTIONS

1	COMMANDS LISTED IN ALPHANUMERIC ORDER	94
2	SCRIPT INSTRUCTION	97
3	SCRIPT MANAGEMENT INSTRUCTIONS.....	102



1 COMMANDS LISTED IN ALPHANUMERIC ORDER

AREABAR=<nROI>,<xStart>,<yStart>	116
BEEP=<msec>	108
CMPBAR=<mode>,<barcode_string>,<label>	110
CMPDATA=<string>,<label ok>,<error_label>	111
COLRGB=RRGGBB	119
EJECT=<n>,<label>	106
ENREAD=<status>,<ncis>,<label>	109
ENRGB=<n>	120
FEED=<label>	106
GOTO=<label>	107
IN=<n>,</>,<label>	103
LCDBKLT=<status>	115
LCDCLR=	112
LCDLOGO=<filename>	115
LCDROWA=<string>	113
LCDROWB=<string>	114
OUT=<n>,<status>,<time>	103
PARK=<n>,<error_label>	120
PAUSE=<msec>	102
PRINT=<n>,<label>	105
PRNDATA=<label>,<...>	104
RESET	102
RESUME=<n>,<error_label>	121
SELFEEED=<n>,<label>	105
SENDERBAR=<portType>	118
SNDDATA=<portType>	117



TCKIN=<paper_status>,<label>	108
TCKOUT=<paper_status>,<label>	107
WAITBAR=<msec>	112





2 SCRIPT INSTRUCTION

Parking printer device has inside a script-command interpreter. The script file stored in the flash drive of the device must be named “script.spt”. The file “script.spt” is loaded only at power-on of the machine; if the file is modified, the machine must be resetted for execute new script-commands.

EXAMPLE

Script parking management of device

This script example allows to manage a vehicle entrance and exit from a parking lot. If the button connected to one of the device's input port is pressed, the printer will issue an entrance ticket and send a signal to one of the output ports to lift the parking bar and allow the car entrance.

Otherwise, if the ticket is inserted into the printer from the frontal mouth, the device will read and check the validity of the barcode data printed on it. According to the result of the barcode validity check, the printer will send a signal to open the bar or not.

BLOCK (*)	DESCRIPTION
INIT:	Script start and printer init
MAIN:	Main script function, when the printer is in IDLE waiting for a ticket to be scanned or an input from the I/O board
TICK_PRINTED:	LCD management in case of printed ticket
TICKET_PRESENTED:	Presenter and specific output management
BC_OK:	Function called when the barcode data validity has been checked, it manages the output and the LCD messages.
PRINT_TCK:	Prints the entrance ticket writing a different barcode value for every one. This function it's called by the MAIN when Input 1 is raise.
ERROR:	Error management function.

NOTE (*): Every block can be considered as a function.

In the following table is shown the lines of code written in the file “script.spt”:

INIT:	“INIT” Block, also first Line and program start
LCDBKLT=ON	Power on the LCD backlight
LCDCLR=	Clear LCD
MAIN:	“MAIN” Block
PAUSE=100	Stop for 100 milliseconds



LCDROWA= CUSTOM PARK SYSTEM	Write the message "CUSTOM PARK SYSTEM " on the upper row of the display
LCDROWB= %DTTM0	Write dynamic date and time on the lower row of the display
IN=1,1,PRINT_TCK	Check the status of input port number 1, when its value is 1 jump to label "PRINT_TCK" (A button could be connected to Input 1)
TCKOUT=NOPAPER,MAIN	Check the status of the frontal paper sensor. If there is no paper get back to "MAIN" block. If the paper is present, go to the next instruction
ENREAD=ON,1,ERROR	Enable scan in the top scanner, in case of error jump to "ERROR" label
LCDCLR=	Clear LCD
LCDROWA= READING TICKET	Write the message "READING TICKET PLEASE WAIT..." on the display.
LCDROWB= PLEASE WAIT...	
PAUSE=50	Stop for 50 milliseconds
WAITBAR=5000	Wait a successful barcode reading for maximum 5 seconds
CMPBAR=H,OUT:%DTTM1,BC_OK	Compare read barcode content with the defined string. In this case the printer compares the dates to check the ticket validity. If ok jump to "BC_OK" block, otherwise continue.
EJECT=0,ERROR	Eject the ticket. In case of error jump to "ERROR" block
LCDCLR=	Clear LCD
LCDROWA= TICKET NOT	Write the message "TICKET NOT VALID" on the display.
LCDROWB= VALID	
GOTO=TICKET_PRESENTED	Jump to "TICKET_PRESENTED" block
TICK_PRINTED:	"TICK_PRINTED" Block
LCDCLR=	Clear LCD
LCDROWA= TAKE YOUT TICKET	Write the message "TAKE YOUR TICKET" on the display.
PAUSE=1000	Stop for 1 second
TICKET_PRESENTED:	"TICKET_PRESENTED" Block, called When a ticket is ready to be taken by the user
Lcdbklt=OFF	
PAUSE=500	Display backlight blinks with 500 milliseconds interval
Lcdbklt=ON	
TCKOUT=PAPER,TICKET_PRESENTED	Loop the "TICKET_PRESENTED" until the user takes the ticket from the frontal mouth.
OUT=1,ON,500	If the ticket is taken, the Output 1 is activated for 500 milliseconds (The barrier controller could be connected to OUT 1)
GOTO=INIT	Go back to "INIT" block to restart the script cycle.
BC_OK:	"BC_OK" Block, the script enters here in case the read Barcode data is valid
LCDCLR=	Clear LCD

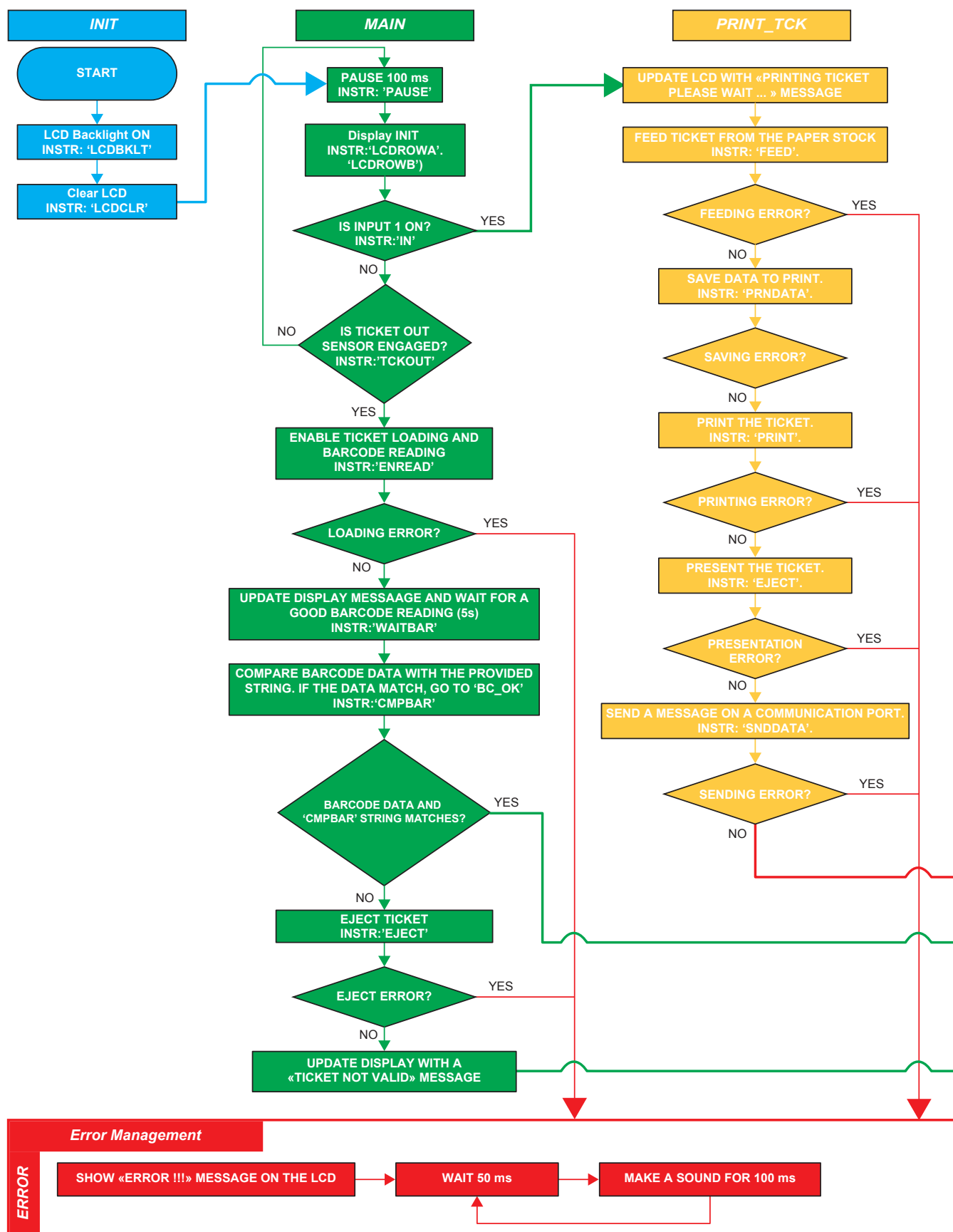


LCDLOGO=01.bmp	Draw on the display the specified image file (stored in the \DISPLAY folder into the flash disk FLASHDRIVE)
BEEP=500	Make a sound for 500 milliseconds
PAUSE=1000	Hold on for 1 second
OUT=1, ON, 500	Activate the output 1 for 500 milliseconds
PAUSE=1000	Stop for 1 second
EJECT=2, ERROR	Retract the ticket and drop it from the back, go to the "ERROR" block if the action is not possible
GOTO=INIT	Go back to "INIT" block to restart the script cycle
PRINT_TCK:	"PRINT_TCK" Block
LCDCLR=	Clear LCD
LCDROWA= PRINTING TICKET	Write the message "PRINTING TICKET PLEASE WAIT...." on the display.
LCDROWB= PLEASE WAIT....	
FEED=0, 0, ERROR	Feed new ticket. In case of error jumps to "ERROR" block
PRNDATA=ERROR, 0501#02CUSTOM SpA#03IN: %CNTR5#04PARKING#0601# 10OUT:%DTTM1+1H#	Save the specified data to print the ticket, using the 0501 layout. Barcode data is dynamic
PRINT=0, ERROR	Execute the print job using the data previously saved. In case of error jumps to "ERROR" block
EJECT=0, ERROR	Execute a ticket presentation. In case of error jumps to "ERROR" block
SNDDATA=S, NEWTICKET, ERROR	Send a message on the desired communication port "S" stands for serial port, the message is "NEWTICKET". In case of error jumps to "ERROR" block
GOTO=TICK_PRINTED	Jump to the "TICK_PRINTED" block
ERROR:	"ERROR" Block
LCDCLR=	Clear LCD
LCDROWA= ERROR !!	Write the message "ERROR !!" on the display
LCDBKLT=OFF	Power off the display
ERROR_1:	"ERROR_1" Block
PAUSE=50	
BEEP=100	Infinite BEEP loop (100 milliseconds ON / 50 milliseconds OFF). Printer in error, must be reset
GOTO=ERROR_1	

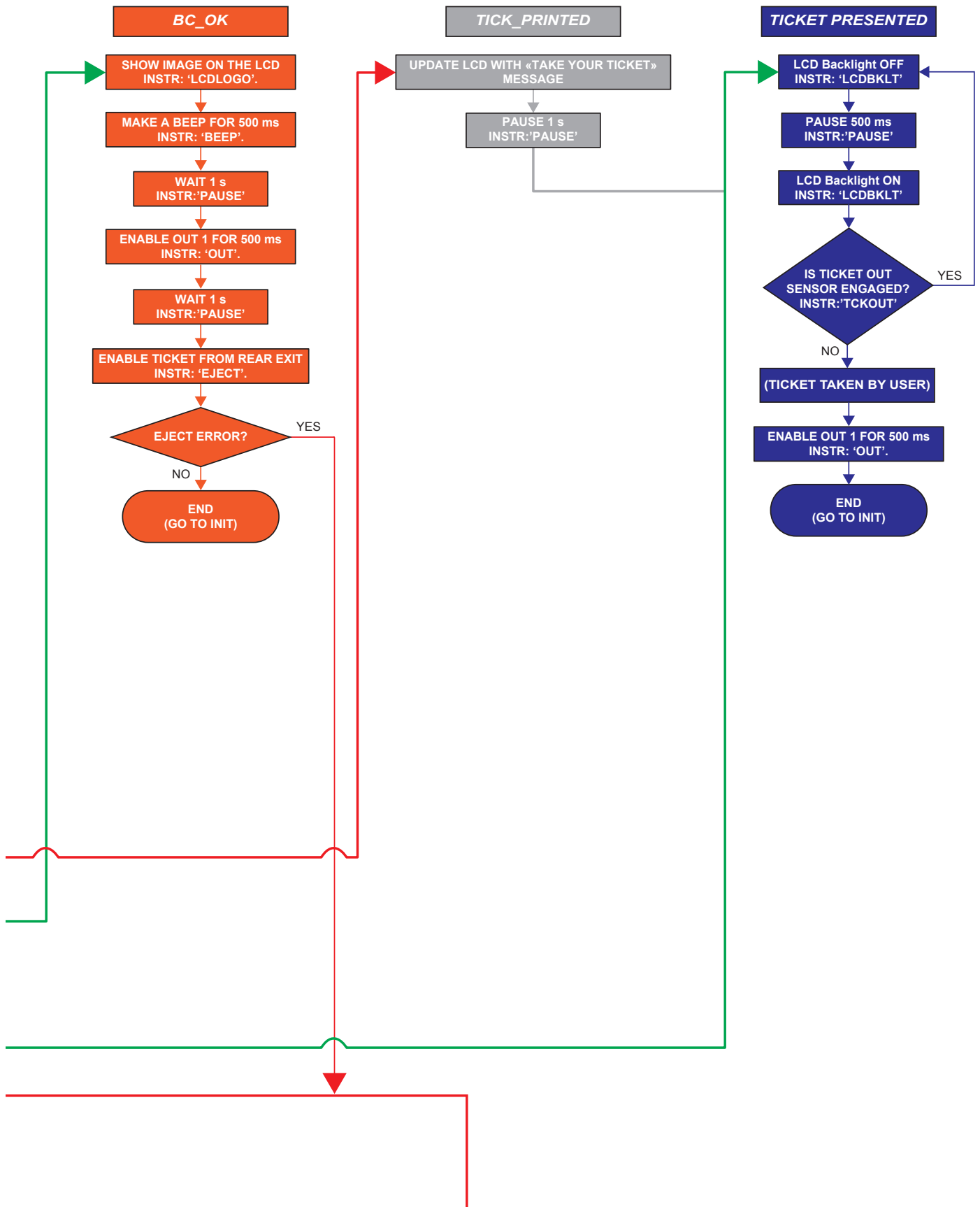
Refer to the following flowchart where is showed the script parking management of device explained in the previous table



The following flowchart shows the scheme of functioning of the script:



NOTE: The text included in single quotes indicates the instruction of the script to run.





3 SCRIPT MANAGEMENT INSTRUCTIONS

PAUSE=<msec>

Sleep time

Valid for	PP54 PP54 EVO
[Format]	PAUSE=<msec>
[Range]	
[Description]	Sleeps time where <msec> is the delay time in milliseconds.
[Notes]	The minimum value is 35msec
[Default]	
[Reference]	
[Example]	To set a sleep time for 600 msec: PAUSE=600

RESET

Reset the printer

Valid for	PP54 PP54 EVO
[Format]	RESET
[Range]	
[Description]	Resets the printer.
[Notes]	
[Default]	
[Reference]	
[Example]	



OUT=<n>,<status>,<time>

Activate/deactivate output

Valid for	PP54 PP54 EVO
[Format]	OUT=<n>,<status>,<time>
[Range]	$1 \leq n \leq 4$ status = 'ON', 'OFF' $0 \leq \text{time} \leq 255$
[Description]	Activates/deactivates output, where <n> is the output port from 1 to 4, <status> can be 'ON' or 'OFF', <time> is the activation/deactivation time in millisecond. If <time> is 0, no timeout.
[Notes]	
[Default]	
[Reference]	
[Example]	To Activate output 1 (no timeout): OUT=1,ON,0

IN=<n>,</>,<label>

Test input

Valid for	PP54 PP54 EVO
[Format]	IN=<n>,</>,<label>
[Range]	$1 \leq n \leq 5$ '0' ≤ / ≤ '1'
[Description]	Tests input, where <n> is the input port from 1 to 5, </> is the logic level ('0' or '1'), <label> is the label to jump if true.
[Notes]	
[Default]	
[Reference]	
[Example]	Jump to SENSOR 1 if input 2 is at logic level '1' IN=2,1,SENSOR 1



PRNDATA=<label>,<...>

Load data for layout

Valid for	PP54 PP54 EVO
[Format]	PRNDATA=<label>,<...>
[Range]	
[Description]	Loads data for layout (see 0x1F 0x64 command), where <label> is the label to jump if error. - With “%DATE<x>”1 can be added the date with format: “%DATE0” = DD/MM/YY “%DATE1” = YYMMDD “%DATE2” = DDMMYY - With “%TIME<x>” can be added the time with format: “%TIME0” = hh:mm “%TIME1” = hhmm “%TIME2” = hh:mm:ss “%TIME3” = hhmmss - With “%DTTM<x>” can be added the date/time with format: “%DTTM0” = DD/MM/YY hh:mm:ss “%DTTM1” = YYMMDDhhmmss “%DTTM2” = DDMMYYhhmmss “%DTTM3” = YYMMDDhhmmssd - Is possible to add minutes/hours/day at the date/time with: “%DTTM<x>+<n>M” =add <n> minutes to actual date/time “%DTTM<x>+<n>H” =add <n> hours to actual date/time “%DTTM<x>+<n>D” = add <n> days to actual date/time - With “%CNTR<x>” can be added the incremental counter with <x> specifies the number of counter digits.
[Notes]	
[Default]	
[Reference]	0x1f 0x64
[Example 1]	To save the print data with layout 0501, with date and time dynamics printed in field 03 and 10: <i>PRNDATA=ERROR,0501#02CUSTOM SpA#03%DATE0- %TIME0#04PARKING#10IN:%DATE0-%TIME0#</i>
[Example 2]	To save the print data with layout 0501, with date and time dynamics printed in field 10 (will be added one hour at the current time): <i>PRNDATA=ERROR,0501#02CUSTOM SpA#04PARKING#0601#10OUT:%DTTM1+1H#</i>
[Example 3]	To save the print data with layout 0501, with incremental counter (5 characters length formatted) in field 010: <i>PRNDATA=ERROR,0501#02CUSTOM SpA#04PARKING#0601#10COUNT:%CNTR5#</i>



PRINT=<n>,<label>

Print ticket

Valid for	PP54 PP54 EVO
[Format]	PRINT=<n>,<label>
[Range]	$0 \leq n \leq 1$
[Description]	Prints ticket, where <n> is the feed type, <label> is the label to jump on error: <n> = 0 don't feed new ticket and print ticket on printhead <n> = 1 feed ticket if not present and print ticket
[Notes]	
[Default]	
[Reference]	
[Example]	To Print ticket already present on printhead: PRINT=0,ERROR

SELFEED=<n>,<label>

Select feeder

Valid for	PP54 PP54 EVO
[Format]	SELFEED=<n>,<label>
[Range]	$0 \leq n \leq 1$
[Description]	Selects the feeder, where <n> is the feeder to load ticket, <label> is the label to jump on error: <n> = 0 select feeder 1 <n> = 1 select feeder 2
[Notes]	
[Default]	
[Reference]	
[Example]	To load paper from feeder 1: SELFEED=0,ERROR



FEED=<label>

Feed ticket

Valid for	PP54 PP54 EVO
[Format]	FEED=<label>
[Range]	
[Description]	Feeds ticket from the paper stock where <label> is the label to jump if error.
[Notes]	
[Default]	
[Reference]	
[Example]	To Feed ticket from the paper stock: FEED=ERROR

EJECT=<n>,<label>

Eject ticket

Valid for	PP54 PP54 EVO
[Format]	EJECT=<n>,<label>
[Range]	$0 \leq n \leq 3$
[Description]	Presents the ticket, where <n> is the eject type as follows: <n> = 0 frontal, ticket presentation <n> = 1 frontal, eject ticket <n> = 2 rear, eject ticket <n> = 3 bottom, eject ticket and <label> is the label to jump if error.
[Notes]	
[Default]	
[Reference]	
[Example]	Ticket presentation on the frontal mouth: EJECT=0,ERROR



GOTO=<label>

Jump to label

Valid for	PP54 PP54 EVO
[Format]	GOTO=<label>
[Range]	
[Description]	Jumps to label, where <label> is the label to jump.
[Notes]	
[Default]	
[Reference]	
[Example]	Jump to INIT label: GOTO=INIT

TCKOUT=<paper_status>,<label>

Test ticket out sensor

Valid for	PP54 PP54 EVO
[Format]	TCKOUT=<paper_status>,<label>
[Range]	
[Description]	Tests ticket out sensor (SENS5), where <paper_status> can be 'PAPER' or 'NOPAPER', <label> is the label to jump if true.
[Notes]	
[Default]	
[Reference]	
[Example]	Jump to EXIT if paper is present on ticket out sensor: TCKOUT=PAPER,EXIT



TCKIN=<paper_status>,<label>

Test ticket in sensor

Valid for	PP54 PP54 EVO
[Format]	TCKIN=<paper_status>,<label>
[Range]	
[Description]	Tests ticket in sensor (SENS1), where <paper_status> can be 'PAPER' or 'NOPAPER', <label> is the label to jump if true.
[Notes]	
[Default]	
[Reference]	
[Example]	Jump to ENDPAP if paper is not present on ticket in sensor: TCKIN=NOPAPER,ENDPAP

BEEP=<msec>

Emit a beep

Valid for	PP54 PP54 EVO
[Format]	BEEP=<msec>
[Range]	
[Description]	Emits a beep where <msec> is the beep time in milliseconds.
[Notes]	The minimum value is 5 msec
[Default]	
[Reference]	
[Example]	To set a beep time for 1 second: BEEP=1000

ENREAD=<status>,<ncis>,<label>

Enable/disable ticket loading and barcode reading

Valid for	PP54 PP54 EVO
[Format]	ENREAD=<status>,<ncis>,<label>
[Range]	'ON' ≤ status ≤ 'OFF' 1 ≤ ncis ≤ 2
[Description]	Enables/disables the ticket loading from the frontal mouth and barcode search. • <status> parameter can be 'ON' or 'OFF' • <ncis> parameter indicates wich scanner is used to barcode search <ncis> = 1 Top scanner <ncis> = 2 Bottom scanner • <label> parameter is the label to jump if error.
[Notes]	
[Default]	
[Reference]	
[Example]	Enables the barcode search in the top scanner and jump to ERROR label in case of error: ENREAD=ON,1,ERROR



CMPBAR=<mode>,<barcode_string>,<label>

Compare barcode data with provided string

Valid for	PP54 PP54 EVO
[Format]	CMPBAR=<mode>,<barcode_string>,<label>
[Range]	
[Description]	Compares barcode readed with <barcode_string>, if equal jump to <label>; • <mode> identify the compare modality as follows: '0': compare all characters; (from '1' to '95'): compare n characters; 'L': compare actual printer-date with barcode-date; compare good if barcode-date is lower than machine-date, <barcode_string> must have inside '%DTTMx' like PRNDATA command; 'H': compare actual printer-date with barcode-date; compare good if barcode-date is higher than machine-date, <barcode_string> must have inside '%DTTMx' like PRNDATA command;
[Notes]	
[Default]	
[Reference]	
[Example1]	Jump to OK label if the barcode readed is '0123456789' CMPBAR=0,0123456789,OK
[Example2]	Jump to OK label if barcode data readed are: 'OUT:' followed by date/time if higher than the date/time of the device: CMPBAR=H,OUT:%DTTM1,OK



CMPDATA=<string>,<label ok>,<error_label>

String comparison on the active communication port

Valid for	PP54 PP54 EVO
[Format]	CMPDATA=<string>,<label ok>,<error_label>
[Range]	
[Description]	Waits a string on the active communication port and compares it with the <string> parameter, if equal jump to the <label ok> label otherwise jump to the <error_label> label.
[Notes]	
[Default]	
[Reference]	
[Example]	Jump to GREEN_LIGHT label if the 'BCOK' string it's equal to the string present on the active communication port, otherwise jumps to RED_LIGHT label: CMPDATA=BCOK,GREEN_LIGHT,RED_LIGHT



WAITBAR=<msec>

Wait barcode readed

Valid for	PP54
	PP54 EVO

[Format] WAITBAR=<msec>

[Range]

[Description] Waits a barcode readed, where <msec> is the timeout in milliseconds.

[Notes] The minimum value is 32 msec

[Default]

[Reference]

[Example] Wait a barcode for max 10 second:
WAITBAR=10000

LCDCLR=

Clear LCD

Valid for	PP54
	PP54 EVO

[Format] LCDCLR=

[Range]

[Description] Clears the LCD display.

[Notes]

[Default]

[Reference]

[Example]



LCDROWA=<string>

Display the characters on upper row

Valid for	PP54 PP54 EVO
[Format]	LCDROWA=<string>
[Range]	
[Description]	Displays the characters in <string> on upper row (LCDROWA). • With “%DATE<x>” 1 can be added the date with format: “%DATE0” = DD/MM/YY “%DATE1” = YYMMDD “%DATE2” = DDMMYY • With “%TIME<x>” can be added the time with format: “%TIME0” = hh:mm “%TIME1” = hhmm “%TIME2” = hh:mm:ss “%TIME3” = hhmmss • With “%DTTM<x>” can be added the date/time with format: “%DTTM0” = DD/MM/YY hh:mm:ss “%DTTM1” = YYMMDDhhmmss “%DTTM2” = DDMMYYhhmmss “%DTTM3” = YYMMDDhhmmssd
[Notes]	The maximum length of <string> parameter is 20 characters
[Default]	
[Reference]	
[Example]	To displays on LCD “TEST ROW 1” on upper row: LCDROWA=TEST ROW 1



LCDROWB=<string>

Display the characters on lower row

Valid for	PP54 PP54 EVO
[Format]	LCDROWB=<string>
[Range]	
[Description]	Displays the characters in <string> on lower row (LCDROWB). • With “%DATE<x>”1 can be added the date with format: “%DATE0” = DD/MM/YY “%DATE1” = YYMMDD “%DATE2” = DDMMYY • With “%TIME<x>” can be added the time with format: “%TIME0” = hh:mm “%TIME1” = hhmm “%TIME2” = hh:mm:ss “%TIME3” = hhmmss • With “%DTTM<x>” can be added the date/time with format: “%DTTM0” = DD/MM/YY hh:mm:ss “%DTTM1” = YYMMDDhhmmss “%DTTM2” = DDMMYYhhmmss “%DTTM3” = YYMMDDhhmmssd
[Notes]	The maximum length of <string> parameter is 20 characters
[Default]	
[Reference]	
[Example]	To displays on LCD date and time with format DD/MM/YY hh:mm:ss on lower row: LCDROWB=%DTTM0



LCDBKLT=<status>

Enable/disable LCD backlight

Valid for	PP54 PP54 EVO
[Format]	LCDBKLT=<status>
[Range]	
[Description]	Enables/disables LCD backlight, where <status> parameter can be 'ON' or 'OFF'.
[Notes]	
[Default]	
[Reference]	
[Example]	To power on the LCD backlight: LCDBKLT=ON

LCDLOGO=<filename>

Show an image on the LCD

Valid for	PP54 PP54 EVO
[Format]	LCDLOGO=<filename>
[Range]	
[Description]	Displays a bmp 132x129 black/white on LCD backlight, where <filename> is the path and name of file. • The logos to be displayed are bitmap files that are already saved in the folder logos into the PP54 flash disk. • The display logos is saved into the PP54 flash disk at the following path: FLASH DRIVE\DISPLAY\
[Notes]	• Command valid only for B/W BMP images. • The image must be in black and white (1bit/pixel). • The size in pixel of the BMP image must be 132x29.
[Default]	
[Reference]	
[Example]	To display the "01.bmp" bitmap on LCD: LCDLOGO=FLASH DRIVE\DISPLAY\01.bmp



AREABAR=<nROI>,<xStart>,<yStart>

Set a barcode search area (ROI)

Valid for	PP54 PP54 EVO
[Format]	AREABAR=<nROI>,<xStart>,<yStart>,<xDim>,<yDim>,<Timeout>
[Range]	1 ≤ <nROI> ≤ 5 0 <xStart> 640 0 <yStart> 990 0 <xDim> 640 0 <yDim> 990
[Description]	Set the search area ROI of barcode with the following parameters: • <nROI> specifies the search area of barcode • <xStart> specifies the X coordinate of origin point, expressed in dot • <yStart> specifies the y coordinate of origin point expressed, in dot • <xDim> specifies the area size in the x direction expressed, in dot • <yDim> specifies the area size in the y direction expressed, in dot • <Timeout> specifies the timeout value for a barcode search into the roi area, is expressed in mil- lieseconds
[Notes]	
[Default]	
[Reference]	
[Example]	



SNDDATA=<portType>

Send message to the communication port

Valid for	PP54 PP54 EVO
[Format]	SNDDATA=<PortType>,<Message>,<error_label>
[Range]	<PortType>= 'S' <PortType>= 'E' <PortType>= 'U'
[Description]	Sends a message to the communication port specified by the <PortType> parameter. • With <PortType> = 'S' the message is sent to the serial port • With <PortType> = 'E' the message is sent to the ethernet port • With <PortType> = 'U' the message is sent to the USB port • <Message> = message to send • <error label> = is the label to jump if error.
[Notes]	• See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command. • This command return ACK (0x06) if command is successful, otherwise return NACK (0x15) if the command cannot be processed followed by a byte that returns the code of error type (see the table RETURN CODES GUIDE at the beginning of this document for further explanation).
[Default]	
[Reference]	
[Example]	Sends the "Message" string on the serial port and in case of error jumps to "ERROR" label SNDDATA='S',Message,ERROR



SENDER=<portType>

Send read barcode data to the communication port

Valid for	PP54 PP54 EVO
[Format]	SENDER=<PortType>,<error_label>
[Range]	<PortType>= 'S' <PortType>= 'E' <PortType>= 'U'
[Description]	Sends the read barcode data using the ENREAD command to the communication port specified by the <PortType> parameter, where <error_label> is the label to jump if error. • <PortType> identifies the communication port as follows: 'S' the communication port is serial port 'E' the communication port is ethernet port 'U' the communication port is USB port
[Notes]	• See paragraph ANSWER PROTOCOL FORMAT at the beginning of this manual for the response transmitted by the command. • The <DATA > bytes is as follows: ACK (0x06) barcode index (1 byte) size of the barcode data (2 bytes) barcode data from barcode index is repeated for each barcode
[Default]	
[Reference]	
[Example]	Sends the barcode data to serial port and jump to ERROR label in case of error: SENDER='S',ERROR



COLRGB=RRGGBB

Set RGB color of the LED bar

Valid for	PP54 PP54 EVO
[Format]	COLRGB=RRGGBB
[Range]	0x00 ≤ RR < 0xFF 0x00 ≤ GG < 0xFF 0x00 ≤ BB < 0xFF
[Description]	Sets the color in RGB for the the LED bar, where the RGB color is defined in the following way: RR = the hexadecimal value of the grade of red; GG = the hexadecimal value of the grade of green; BB = the hexadecimal value of the grade of blue;
[Notes]	• The command does not turn on the LED bar. To take effect, the setting must be followed by ENRGB=ON
[Default]	
[Reference]	
[Example 1]	To illuminate the LED bar with red color: COLRGB=FF0000
[Example 2]	To illuminate the LED bar with purple color: COLRGB=FF00FF



ENRGB=<n>

Power ON / OFF LED bar

Valid for	PP54 PP54 EVO
[Format]	ENRGB=<n>
[Range]	
[Description]	Power on/off the LED bar where <n> parameter can be 'ON' or 'OFF'.
[Notes]	
[Default]	
[Reference]	
[Example]	To power on the LED bar: ENRGB=ON

PARK=<n>,<error_label>

Move ticket in one of the parking slot

Valid for	PP54 PP54 EVO
[Format]	PARK=<n>,<error_label>
[Range]	n = 0, n = 1
[Description]	Moves ticket in one of the parking slot where <n> parameter specifies the index of parking slot as and <error_label> is the label to jump if error. • the parking slot identified by <n> parameter as defined as follows: <n>= 0 bottom parking slot <n>= 1 rear parking slot
[Notes]	
[Default]	
[Reference]	
[Example]	To move the ticket from the printing position to the bottom parking slot and jump to ERROR label in case of error: PARK=0,ERROR



RESUME=<n>,<error_label>

Move ticket from the parking slot to the printing position

Valid for	PP54 PP54 EVO
[Format]	RESUME=<n>,<error_label>
[Range]	n = 0, n = 1
[Description]	Moves ticket from one of the parking slot to the printing position where <n> parameter specifies the index of parking slot from which to move the ticket and <error_label> is the label to jump if error. • the parking slot identified by <n> parameter as defined as follows: <n>= 0 bottom parking slot <n>= 1 rear parking slot
[Notes]	
[Default]	
[Reference]	
[Example]	To move the ticket from the bottom parking slot to the printing position and jump to ERROR label in case of error: RESUME=0,ERROR





CUSTOM S.p.A.

World Headquarters

Via Berettine, 2/B - 43010 Fontevivo, Parma ITALY

Tel. +39 0521 680111 - Fax +39 0521 610701

info@custom.biz - www.custom.biz

All rights reserved

www.custom.biz