TechManual
GALEB GP-550

Contents

Programming Guide

INTRODUCTION

Fiscal printer **GALEB GP-550**, in itself has implemented two protocols, standard (which is the same protocol as printers FP-550) and the base protocol. In this guide, you will find an explanation of each command, both protocols.

The fiscal device functions under the control of an application program, and communicates with it via a asynchronous serial channel for connection to the RS232 interface, or with USB connection.

It`s purpose is to execute a previously defined set of commands, which are logically arranged depending on the type of operations, which have to be executed. This utility program doesn`t have direct access to the resources of the fiscal device, but can download data, connected with the state of the fiscal device and the fiscal memory.

The fiscal device performs the following operations:
- Save the fiscal number of the fiscal device
- Save the fiscal parameters - like the tax number of the person, registered for taxation, the date of starting operations etc.
- Save information about the owner, his/her address, names etc.
- Save data on the sum of the sales and generating clients receipts
- Save data on the daily turnover in the fiscal memory and generate daily report
- Generate reports on completed sales and the contents of the fiscal memory
- Print reports, generated by the application
- Returns data to the application program.

TAXATION CATEGORIES AND CALCULATION OF VAT (PDV)

In Serbian: PDV means (Porez na Dodatu Vrednost)

Each separate sale is recalculated in relation to a given taxation category (PDV), which defines the tax rate, applicable to the base price for the formation of the sale price. The fiscal printer work with 9 categories at the most and they are marked with the first letters of the alphabet of the respective country, for which the fiscal printer (FP) has been set-up. In the case of Serbia these letters would be A, G, D, Đ, E, Ž, I, J, K.
BTW this letter should be **А, Г, Д, Ђ, Е, Ж, И, J, К,** in cyrrilic letter.

For each of the Tax groups a tax rate is selected (in percents) which should be number in a format "99,99".

Use command 83 (53h) to set Tax Groups.

The net value of the sales is calculated with the help of the formula:

In PDV mode: NET_value = ROUND (SALE_value / (1 + TAX_rate)).

The function ROUND return the number rounded on specified number of decimal points. The value of the tax sum due is calculated with the help of the formula:

TAX_value = SALE_value - NET_value

OPERATION MODES OF THE FISCAL DEVICE

The fiscal device can work in two modes.

1. Tutorial mode – non fiscal mode.
The device has not been fiscalized. All data, necessary for its normal functioning has been entered and recorded in the fiscal memory with the exception of the PIB - Owner Tax Number. Sale and all fiscal documents are disabled, and you only can do Test device, and you can not do anything else.

2. Normal mode of operation – fiscal mode. 1
The device is fiscalized. The PIB – Owner Tax Number has been entered into the fiscal memory. All financial regulations will be applied.

1. THE DIFFERENT STATES OF THE FISCAL PRINTER

The fiscal printer can have several different states and the transitions between them are not always possible. The control of the printer and the transitions between the states, when this is possible, are controlled by the application program on the HOST (PC), which must be coordinated with the used protocol. When this protocol has not been applied correctly, the printer may shift into an undesirable state, or it's moving through a given state may be skipped, leading it to enterthe ERROR state.

A. INITIAL STATE

In this state, the date and time can be set, the initial receipt's number can be programmed and the IBFM, the fiscal module number can be entered. These operations are performed, before the device is given to it's owner, by an authorized service specialist.

THESE OPERATIONS MUST BE DONE ONLY BY SERVICEMAN ON THE CLIENT SITE.

B. AFTER FORMATTING THE FISCAL MEMORY

In this state, the type of the currency and the tax rates are entered. This is the state, in which the fiscal devices are stored in the warehouse of the manufacturer. The commands that follow are performed in the following order: 83 (53h). After these operations are performed the fiscal printer can be handed over to the client, who will use it.

C. WHEN HANDED OVER TO THE CLIENT
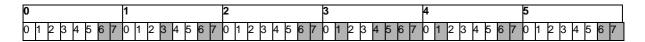
In this state, the Header and the Footer are set up - this is the beginning and the end of each separate receipt. Header can be write just when jumper J1 is in service mode. The command 43 (2Bh) has to be performed as many times as the number of the lines entered.

E. FISCALIZED FISCAL PRINTER

Fiscal receipts can be issued in this state and they will be marked as such.

CURRENT STATE OF THE FISCAL PRINTER

The current state of the device is encoded in a field which is 6 Bytes long and is transferred to each message, issued from the fiscal printer. Each byte of this field is described as follows:

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

Byte 0   **General purpose**

0.7 = 1 bit 7	is reserved
0.6 = 1 bit 6	is reserved
0.5 = 1	general error - this is OR of all errors, market with "#".
0.4 = 1#	if the printing mechanism has defected,
0.3 = 1	display is not connected,
0.2 = 1	when the date and time have not been set,
0.1 = 1#	invalid code of received command
0.0 = 1#	syntax error in received data.

Byte 1   **General purpose**

1.7 = bit 7	is reserved,
1.6 = bit 6	is reserved,
1.5 = 1	printer cover is opened,
1.4 = 1#	if the contents of the RAM is destroyed i.e., emergency clearing is performed,
1.3 = 1#	not used.
1.2 = 1#	when emergency clearing of the RAM has been performed,
1.1 = 1#	if the execution of the command is not permitted in the current fiscal mode
1.0 = 1	when at the execution of the command the sums cause an overflow in some fields.

Status 1.1 will also be set and the transfer of the data will not be executed.

Byte 2   **General purpose**

2.7 = 1 bit 7	is reserved,
2.6 = 1 bit 6	is reserved,
2.5 = 1	when a non-fiscal receipt has been opened,
2.4 = 1	paper near end for Journal printer
2.3 = 1	when a fiscal receipt has been opened,
2.2 = 1	no paper for Journal printer
2.1 = 1	paper near end. Combined flag for both printers
2.0 = 1#	no paper. Combined flag for both printers.

Byte 3   **State of the keys**

3.7 = 1 bit 7	is reserved,
3.6 = 1 bit 6	is reserved,
3.5 = 1	not in use
3.4 = 1	not in use
3.3 = 1	Baud rate is set up.
3.2 = 1	transparent display mode is set up
3.1 = 1	not in use
3.0 = 1	Auto cutting is selected.

Byte 4   **For the fiscal memory**

4.7 = 1 bit 7	is reserved
4.6 = 1 bit 6	is reserved
4.5 = 1	OR of all mistakes in fiscal memory, marked by "*" by Bytes 4 and 5,
4.4 = 1*	no free space in fiscal memory,

4.3 = 1	when there is free space for less than 50 fiscal closures,
4.2 = 1	when there is no fiscal memory module
4.1 = 1	not in use,
4.0 = 1*	error when recording in the fiscal memory

Byte 5   **For the fiscal memory**

5.7 = 1 bit 7	is reserved,
5.6 = 1 bit 6	is reserved,
5.5 = 1	when the serial number and fiscal number are programmed,
5.4 = 1	when at least one of the tax rates fields is programmed,
5.3 = 1	when the device is fiscalized,
5.2 = 1*	last closure (Z report) isn't well performed,
5.1 = 1	when the fiscal memory is formatted,
5.0 = 1*	when the fiscal memory is in the "Read Only" mode.

POWER FAILURE

The printer status is reflected in it's internal status Bytes at all times. When it is switched on (after a power OFF), the commands 74 (4Ah) and 103 (67h) initiate the utility program to gather information on the status of the device. The utility program must make a decision bout the future behavior of the printer, depending on it's status. It is guaranteed, that the fiscal memory will not be destroyed from the power failure and that the accumulated sums in the operational memory will remain valid. If the Power Failure occurs when the printer in printing, after the Power is recovered the printer will finish the printing.

PROGRAMMING AND GETTING DATA ABOUT ITEMS

The printer works with data which are programmed in it. Administration of base of items can be realized with command 107 (6Bh).

ISSUING FISCAL RECEIPTS

First, a fiscal receipt is opened, then sales are registered, the payment is done and finally, the receipt is closed. Commands in standard protocol – 48 (30h), 51 (33h), 52 (34h), 53 (35h), 56 (38h) are used. A daily financial report and clearing is performed in order to enter the information into the fiscal memory. The command 69 (45h) are used.

If using the command base protocol is a set of commands needed 0x31, 0x32, 0x33.

GENERATING REPORTS

Reports are generated only by the fiscal printer and after receiving the command from the PC. The client's program cannot perform any changes on the reports when they are generated – they look exactly as they have been set up in the fiscal memory. Initiation is done with the commands in the standard protocol:

> **50 (32h)** - report for tax changes and decimal point changes
> **69 (45h)** - daily financial report with or without clearing,
> **79 (4Fh)** - financial report - date to date,
> **111 (6Fh)** - daily fiscal report with or without clearing and printing of the article sums.

LOW LEVEL PROTOCOL

A. Protocol type - Master (Host)/ Slave

The fiscal printer executes the commands, sent by the Host, and returns a message depending on the character of the result. The fiscal printer cannot initiate communication. Only reports, resulting from the execution of given commands, are sent to the Host. Messages in the protocol are either "packed" messages or single Bytes. The fiscal printer maintains communication via RS232 or USB interface at speeds of 9600 to 115200b/s, 8N1..

B. Order of the messages

Host sends a packed message, containing the command, sent to the printer. The fiscal printer executes the operation and returns a packed answer. Host must wait for the printer's answer before sending another message. The protocol uses non-packed codes with a length of one byte for processing the necessary pauses and error mistakes.

C. Non-packed messages, time-out

Given normal operation of all Host messages, Slave answers not later than 60 ms, with a packed message or with a single-byte code. Host must have 500 ms time-out for getting the message from Slave. If, during this time, no answer comes, the message is repeated with the same logical number and the same command. After several unsuccessful attempts, Host indicates, that there is either no connection with the fiscal printer or that there is a hardware error. Non-packed messages contain one byte and are:

NAK 15 h
This code is sent from Slave, when there is an error in the control sum or in the form of the received message. When Host receives a NAK, a new message, with the same logical number, is transmitted. If those situations often occur, this is indicator that the error exist in communication. This communication is caused, because the cable is bad or of outside disturbances or because the BCC parameter is calculated wrong.

SYN 16 h
This code is sent by Slave, when the execution of the command is received needs more time. SYN is dispatched every 60 ms until the packed message is ready for an answer.

D. Packed messages

Host to printer (Send)

<1>	LEN	SEQ	CMD	DATA	<5>	BCC	<3>
1 byte	1 byte	1 byte	1 byte	?	1 byte	4 byte	1 byte
$01	$20-$F0	$20-$7F	$20-$7F	?	$05	DO $FFFF	$03

<01><len><seq><cmd><data><05><bcc><03>

Printer to Host (Receive)

<1>	LEN	SEQ	CMD	DATA	<4>	STATUS	<5>	BCC	<3>
1 byte	1 byte	1 byte	1 byte	?	1 byte	6 byte	1 byte	4 byte	1 byte
$01	$20-$F0	$20-$7F	$20-$7F	?	$04	$80-$FF	$05	DO $FFFF	$03

<01><len><seq><cmd><data><04><Status><O5><bcc><03>

<01> Preamble

length: 1 byte
value: 01h

In the fiscal printer all commands begin with this byte, except if the printer is occupied with solving previous operation and it don't reach to finish operation for 60ms.

<len> number of Bytes from <01> (without it) up to <05>(incl.) plus a fixed shift of 20h.

length: 1 byte
value: 20h - 7Fh.
The length considers amount of length next fields for Send packets:
<LEN><SEQ><CMD><DATA><05>
  1   1   1   X   1 - length of fields in bytes
  LEN=4+20H(32)=36+X(DATA).
For **Receive packet**:
<LEN><SEQ><CMD><DATA><04><STATUS><05>
 1   1   1   x   1   6   1 - length of fields in bytes
LEN=11+20H(32)=43+X(DATA).

<seq> logical number of the frame

Length: 1 byte
Value 20h-7Fh
This sequence should begin from 02h and exceed for 1. When the sequence reached the end value (7Fh), it should be resetted on value 22h. The fiscal printer records the same <seq> in the reply message. If the FP receives a message with the same <seq> as the last message received, it will not perform an operation, but will repeat the last message it has sent.

<cmd> code of the command.

Length: 1 byte
Value: 20h-7Fh
The FP writes the same <cmd> in the message reply. If the printer receives a nonexisting code, it replies with a "packed message" with zero length of the data field and sets the respective status bit.

<data> -data

Length: 0 - 203 Bytes for Host to Printer
        0 - 198 Bytes for Printer to Host
Value: 20h-FFh
The format length of the data field depends on the command. If the command has no data, then the length of this field is zero. If there is a syntax mistake, the data is set up with the respective status bit and a "packed message" is returned with zero length of the data field.

<04>   divider (only for Printer to Host messages)
Length: byte
Value: 04h.
This field define different between data and status bits in receive packet. If this bit isn't at this place, this means that content of packet is sent wrong and that it is necessary to check function which performed receiving of packet or checksum.

<status> -the field with the current state of the fiscal device
Length: 6 Bytes
Value: 80h – FFh.
Status bits will be returned at perform every command when the printer return Receive packet. For example, the account isn't printed, because there is no paper. The program has to check those bits in order to find out current state of the printer.

<05>   Postamble
Length: 1 byte
Value: 05h.
Postamble packet separate control rate from another part of the packet. At this location always must be 05.

<bcc> control sum (0000h-FFFFh)
> Length: 4 byte
> Value: 30H - 3FH
> The sum includes from <01> and without it to <05> including. Every digit from the two Bytes is sent with an added 30H. For example, the sum 1AE3h is presented as 31h, 3Ah, 3Eh, 33h. The way calculating this control rate is next:
> > For **Send packet**:
> > **<LEN><SEQ><CMD><DATA><05>**
> >   X1   X2   X3   X4   X5 - bytes at this locations
> > IF is: X1=44H, X2=22H, X3=2CH, X4='10', X5=5, then is
> > BCC=44H+22H+2C+'1'+'0'+5H=F8=00F8=30H,30H,3FH,38H.
> >   For **Receive packet**:
> > **<LEN><SEQ><CMD><DATA><04><STATUS><05>**
> >   X1   X2   X3   X4   X5   X6   X7 - bytes at this locations
> > If is: X1=44H, X2=22H, X3=2CH, X4='10', X5=4, X6=81 82 83 84, X7=05 then is
> > BCC=44H+22H+2C+'1'+'0'+4H+81+82+83+84+05=306=0306=30H,33H,30H,36H.

Notice: 30H in ASCII represent is character 0, and 39H character 9, then is from 30H to 39H = 0..9 in ASCII

<03>   terminator
> Length: 1 byte
> Value: 03h.
> This is the label for the end of packet. If it isn't at this locations, something is wrong with the function for perform packets or with communication.

> For the base protocol in the explanation of each command will be an example of input / output data.

Examples sending packages - standard protocol

Packet sent: Examples

Example 1:
Command: Moving paper **($2C)**
Parameters of command: 10 **(10 Lines)**

HEX FORM											
01	26	22	2C	31	30	05	30	30	3D	3A	03

Meaning:
$01 – PREAMBULE
$26 – LENGTH(length+sequence+command+data+postambule+$20)
$22 – sequence (OD $22 NAVIŠE DO $7F)
$2C – command - 2C is moving paper
$31, $30 - DATA (ASCII '10')
$05 – postambule
$30 $30 $3D $3A – BCC (CHECKSUM) 26+22+2C+30+31+05=$DA+30303030
$03 – TERMINATOR

EXAMPLE 2:
COMMAND: Show at the upper part of display **($2F)**
PARAMETERS OF COMMAND: TEST

HEX FORM												
01	28	22	2F	T	E	S	T	05	30	31	3B	3E	03

Meaning:
$01 – PREAMBULE
$28 – LENGTH(length+sequence+command+data+postambule+$20)
$22 – sequence (OD $22 NAVIŠE DO $7F)
$2F – COMMAND -2F is showing on the display

$54, $45, $53, $54 - DATA (ASCII 'TEST')
$05 – POSTAMBULE
$30 $31 $3B $3E – BCC (CHECKSUM) 28+22+23+54+45+53+54+05=$1BE+30303030
$03 – TERMINATOR

EXAMPLE 3:
COMMAND: DEFINE OF ARTICLE **($6B)**
PARAMETERS OF COMMAND: PA1,10,Article

01,32,22,6B,50,C0,31,2C,31,30,2C,C0,F0,F2,E8,EA,E0,EB,05,30,38,3F,3D,03

Meaning:
$01 – PREAMBULE
$32 – LENGTH (length+sequence+command+data+postambule+$20)
$22 – sequence (Upward from $ 22 to $ 7F)
$6B – COMMAND - 2F is showing on the display
$50,$C0,....$EB - DATA (ASCII 'TEST')
$05 – POSTAMBULE
$30 $38 $3F $3D – BCC (CHECKSUM)
$03 – TERMINATOR

EXAMPLE 4:

Sale of article No.1 with price of 50RSD and payment with 100RSD.

1. OPENING FISCAL ACCOUNT (**$30**)
01,2C,22,30,31,3B,30,30,30,30,2C,31,05,30,32,30,3C,03,

2. SELL OF ARTICLE 1 with price of 50 dinars (**$34**)
01,2B,23,34,53,31,2A,31,23,35,30,05,30,31,3E,3E,03,

3. TOTAL – Cash with 100RSD (**$35**)
01,27,24,35,31,30,30,05,30,31,31,36,03

4. CLOSING THE ACCOUNT (**$38**)
01,24,25,38,05,30,30,38,36,03

CHARACTER-SET OF FISCAL PRINTER GP-550

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8	Ђ										Љ		Њ		Ћ	Џ
9	ђ										љ		њ		ћ	џ
A				Ј												
B													ј			
C	А	Б	В	Г	Д	Е	Ж	З	И		К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш		Ђ	Ž	Ć	Č	Š	
E	а	б	в	г	ф	е	ж	з	и		к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш		đ	ž	ć	č	š	

LIST OF FISCAL COMMANDS IN STANDARD PROTOCOL – ALPHABETIC ORDER

hex	dec	Type	Prg	FUNCTION
21h	(33)	Dsp		Display clear text
23h	(35)	Dsp		Display a text on lower line of the display
26h	(38)	Sale		Open a non-fiscal receipt
27h	(39)	Sale		Close a non-fiscal receipt
2Ah	(42)	Sale		Print a non-fiscal free text
2Bh	(43)	I	PI	Set header and footer.
2Ch	(44)	Sale		Printer Paper feed
2Dh	(45)	Sale	SA	Printer Auto cut of receipts
2Fh	(47)	Dsp		Display a text on upper line of display
30h	(48)	Sale	SA	Open a fiscal (client's) receipt
32h	(50)	Inf		Get Tax Rates within period
33h	(51)	Sale	SA	Display a Subtotal
34h	(52)	Sale	SA	Register and display sales
35h	(53)	Sale	SA	Register Payment (Total)
38h	(56)	Sale	SA	Close a fiscal receipt
3Eh	(62)	Inf	PI	Get back date and time
3Fh	(63)	Dsp		Display date and time
40h	(64)	Inf		Info on last fiscal daily report (Z report)
41h	(65)	Inf		Info on daily taxes
43h	(67)	Inf		Get daily statistics
44h	(68)	Inf		Get free space on fiscal memory
45h	(69)	Sale	SA	Close Fiscal No (daily financial report – Z report)
46h	(70)	X		Register cash-in and cash-out
47h	(71)	X	FR	Printing report of diagnostic info
4Ah	(74)	Inf		Get 36-bit status
4Ch	(76)	Sale		Fiscal Closure Status
4Fh	(79)	FR	FR	Report summary of the fiscal memory by the date (Periodical Report)
50h	(80)			Send a Sound Signal
53h	(83)	Init	PI	Set Decimals and allowed taxes (set VAT)
59h	(89)	X		Test Fiscal Memory Size
5Ah	(90)	Inf	PI	Get the diagnostic information
61h	(97)	Inf	PI	Get Tax Rates
62h	(98)	Init	PI	Set PIB - Owner Tax Number
63h	(99)	Inf	PI	Get PIB - Owner Tax Number
64h	(100)	Dsp		Display free text on display
65h	(101)	I		Set operator's password
66h	(102)	I		Set operator's name
67h	(103)	Inf		Information on a current fiscal receipt
68h	(104)	I		Reset operators sales data
69h	(105)	FR	FR	Report on operators

6Ah	(106)	X		Set Cash Drawer opening time
6Bh	(107)	I		Define and read of Article
6Eh	(110)	Inf		Get sales data
6Fh	(111)	FR	**FR**	Report Price list
70h	(112)	Inf		Get operators data
71h	(113)	Inf		Get number of the last printed doc (receipt, rep)
72h	(114)	Inf		Info for a fiscal record or for a fiscal period
73h	(115)	I		Programming of graphic LOGO
77h	(119)	I		Get back date and time of last write in fiscal memory

A DETAILED DESCRIPTION OF THE COMMANDS - STANDARD PROTOCOL

21h (33)   Display clear text

Syntax: no data Reply: no data
Example of packet(SEQ2):   01 24 22 21 05 30 30 36 3C 03
If a fiscal receipt is opened and SW3 is OFF, only the lower line is cleared.

23h (35)   Display text on the lower line

Syntax: Text Reply: no data
Example of packet (SEQ2): 01 28 22 23 54 65 73 74 05 30 32 31 32 03
Text   A text of 20 symbols, sent directly to the display. Before it, a command is issued for the positioning and clearing of the lower line.

26h (38)   Open a non-fiscal receipt

Syntax: no data
Reply: AllReceipt, ErrCode

AllReceipt   The number of all issued receipts (fiscal and non-fiscal) from the last fiscal closure to the present moment (4 Bytes).
ErrCode   When command has been unsuccessful (1 Byte):
   a.   The fiscal memory is not formatted
   b.   There is an opened fiscal receipt
   c.   There is an opened non-fiscal receipt
   d.   The RealTimeClock is not correct.

FP performs the following operations: Header is printed out; the number of the operator is printed out. A reply is received, containing AllReceipt.

27h (39)   Close a non-fiscal receipt

Syntax: no data
Reply: AllReceipt

AllReceipt   The number of all issued receipts (fiscal and non-fiscal) from the last fiscal closure to the moment (4 Bytes).
FP performs the following operations:
   • Prints a footer, the logical number, date and time of the document is printed.
   • A reply is received, containing AllReceipt.
If SW 1.1 has been raised, the command is not performed, because a non-fiscal receipt is being opened.

2Ah (42)   **Print a non-fiscal free text**

Syntax: Text
Reply: no data

Text   Text up to 30 characters
If SW 1.1 is raised, this means that a non-fiscal receipt is opened and the text is not printed out.
If sent as part of the 30 characters to be recognized some of the reserved words with fiscal documents (eg Cash, Paid, Return, etc.). They will be left out. You Cyrillic characters belong to the set of characters will be converted into Latin version.

2Bh (43)   **Set header and footer and options**

Syntax: <item><text>
Reply: depends from the Syntax

Item   One symbol with the following content:
          "0" to "9" is the number of the lines entered. The lines of the header are from 0 - 5 and the footer from 6 - 9.
          "C" Auto cutter function On/OFF.
          "L" Graphic LOGO printing On/OFF. The Logo is defined by command 115 (73h)
          "I" Allows reading the parameters, which we have set earlier by command 43 (2Bh)
Text   Text up to 36 symbols, where:
          If <item> is a number from 0 to 9 - this is the text of the respective line If <item> = "L" - one symbol "0" or "I", where "0" forbids and "1" permits the first and second line of the Graphic LOGO. If <item> = "C" - one symbol "0" or "1" where "0" forbids and "1" permits auto cutting function.

The "header" consists of 6 lines of text, which is printed out at the beginning of each fiscal or non-fiscal receipt. The normal functioning of the printer demands the entry of at least a "header" of three lines. Header can be writing just when jumper J1 is in service mode. The "footer" is of 4 lines, printed at the end of each receipt. The "header" and "footer" are printed centered on line. This command must be executed at most 6+4 times in order to enter all the lines of the "header" and "footer".

2Ch (44)   **Printer Paper feed**

Syntax: [Lines [,Option]]
Reply: no data

Example of packet (SEQ2):   **01 26 22 2C 31 30 05 30 30 3D 3A 03**
Lines   The number of lines to which the paper will be fed. This should be a positive number, not greater than 99 (1 or 2 Bytes). If this parameter is missing 1 line feeding is assumed.
Option   Defines which paper to be fed:
          "0" no effect
          "1" Only receipt paper is fed
          "2" Only journal paper is fed
          "3" Both papers are fed.

If this parameter is missing "1" is assumed.

2Dh (45)   **Printer Auto cut of receipts**

Syntax: no data
Reply: Result

Result   "P" Successful cut
          "F" Fail cut
Example of packet (SEQ2): 01 25 22 2D 31 03 30 30 3A 3A 03
This command is used to cut the receipt. It should be noted, that the application program will take care to move the receipt at least with 2 lines forward. Otherwise the end of the receipt will be cut. If the printer is in Auto cutting mode, it will position the paper automatically. In this case it is not necessary to use this command.
When the cutter is blocked, the paper must be taken out of the cutter and to execute this command. This will put the cutter in a home position. If the fiscal receipt is opened, the command won't be performed.

2Fh (47)   Display text on the upper line

Syntax: Text
Reply: no data

Example of packet (SEQ2): 01 28 22 2F 54 65 73 74 05 30 32 31 3E 03
Text   Up to 20 characters, sent directly to the display. Before that, a command for positioning and clearing the upper line is entered. The command is rejected when a fiscal receipt is opened or SW3 is OFF.

30h (48)   Open a fiscal receipt

Syntax: <OpCode>;<OpPwd>,<TillNmb>
Reply: AllReceipt, FiscReceipt

OpCode   Operator's number (1 to 8)
OpPwd   Operator password (4 - 6 digits)
TillNmb   Operator's place number (1 - 5 digits)
AllReceipt   Number of all issued receipts (fiscal or not) from the last fiscal closure (4 Bytes).
FiscReceipt   The number of all issued fiscal receipts from the fiscalization to the present moment (4 Bytes).

The Fiscal Printer performs the following operations:
* Prints "header",
* Print Graphics Logo, footer
* Prints Owner Tax Number,
* Print number and name of operator and work place.
* Return of AllReceipt and FiscReceipt.

The command will not be successfully completed when:
* The maximum number of daily receipts has been issued.
* No free fiscal memory.
* Failure of the fiscal memory.
* No code or operator password.
* "Header" contains less than 2 lines.
* No tax number entered.
* Incorrect operator password.

When incorrect operator passwords are entered for 3 times, the printer stops functioning and must be switched off and on again for continuing work.

32h (50)   Get Tax Rates within period

Syntax: [<Start>, <End>]
Reply: "F" if tax rates aren't found for the period if it is an error. PAA, BB, CC, DD, EE, FF, GG, HH, II,
          DDMMYY – if rates are found. Last rates which are found will be returned – for allowed – its sum is returned, and for rates which are forbidden - 'DT' (Disabled Tax).

Start   The date of the start of the period – DDMMYY (6 Bytes).
End   The date of the end of the period - DDMMYY (6 Bytes).

The comma is obligatory when start and end are entered. In case the data field is empty, the information of the last entered rates is recovered. PAA, BB, CC, DD … II DDMMYY,. ..- if no rates are found where P means "PASS" and after that, all rates are counted out as well as the date of their entry.

33h (51)   Display a Subtotals

Syntax: <Print><Display>
Reply: SubTotal,Tax1,Tax2,Tax3,Tax4...

Display = "1",   the value of the subtotal will be displayed,
SubTotal   The sum of the current fiscal receipt (up to 10 Bytes)

Tax1   The sum of taxes under category 1 (up to 10 Bytes)
Tax2   The sum of taxes under category 2 (up to 10 Bytes)
Tax3   The sum of taxes under category 3 (up to 10 Bytes)
...
Tax9   The sum of taxes under category 9 (up to 10 Bytes)

The sum of all sales, registered in the fiscal receipt to the moment. The sum can be printed at demand. The calculated sum is returned to PC as well as accumulated sums for each tax category. If a surcharge or discount is entered, it is printed out on a separate line and the accumulated sums in the different tax groups are respectively corrected.

34H (52)   Register and display sales

Syntax: <S>[Sign]<PLU>[*<Qwan>][#<Price>]
        <VF>
        <VL>
Reply: no data
S   Letter "S" **Sign** One byte "+" or "-"
PLU   PLU number. Integer from 1 - 65023 (5 digit)
Quant Optional parameter, defining the quantity. The default value is 1.000. Allowable formats: 999999; #9999,9; #9,99; 9,999 or ###99,999
Price   Opcional parametar . If it set, the command will first change prise of article in the base of articles and then it will form sales item. If a price isn't specify, it will get from the base..
VF   This command remove first sold article at account.
VL   This command remove last sold article at account.

Commands VF i VL will be performed just before a first successful command for paying. (53). For reseting all account the command is being performed until the flag "Command not permitted" isn't returned. (Statusni bit 1.1).

The Fiscal Printer performs the following operations:
- Print Article, price and name
- A price of article adds at remembered in operational memory. If is operational memory full, the appropriate bits activate from registers.
- Price is displayed on upper raw, name on lower raw of display.

The command will not be successful if:
- A fiscal receipt has not been opened,
- Article does not exists
- There are more than 250 Items in receipt,
- Command **Total** is successfully executed,
- Total by Tax Group is negative,
- Total surcharge and discount is negative,
- Customer display is not connected,
- If it tries rating of article in amount which is larger than it was sold previous in current receipt.
- If it tries rating of article by price which is larger than the price which was sold in receipt.

35h (53)   Register Payment

Syntax: [[<PaidMode>]<Amount>]
Reply :<PaidCode><Amount>

PaidMode   An optional code, showing the form of payment, which can have the following values:
                "P" Payment in cash (by default),
                "C" Payments by cheques,
                "D" Payment by debit cards.
Amount   rate which is being paying /to 11 digit/.

Depending on the code, the accumulated sums in the different registers may be required in the daily report. Command 53 without argument pay rest of Receipt.
PaidCode   One byte -the result from the execution of the command.
                "F" Error
                "D" If the paid sum is smaller than the sum of the receipt. The residual amount for completing the

payment is returned to Amount.
"R" If the paid sum is greater than the sum of the receipt a Change message (POVRAT) appears and the change is returned to Amount.

Amount   Up to 1 digits with a symbol. Depends on PaidCode.

This command causes the calculation of the sums on the fiscal receipt, the printing out of the sum with a specific font and showing it on the display. The printing of an additional text is also possible - for example, a text describing the form of payment.

If the command is performed successfully, it will be generated the impulse for opening the drawer. If there are no arguments, the printer sells auto whole rate in cash.

Notice: The digits have bigger prority then the text, when they should be showed on the display.

The command will not be successful if:
- A fiscal receipt has not been opened,
- The accumulated sum is negative,
- If one of the sums from the different tax groups is negative.

After the successful execution of the command, the fiscal printer will not perform the commands 48 (30h) and 51 (33h) within the opened receipt. The command can be set when the sum of received resources of selling equal or larger than the rate of fiscal receipt.

38h (56) Close a fiscal receipt

Syntax: no data
Reply: AllReceipt, FiscReceipt,Total

AllReceipt   Number of all receipts issued from the last daily report.
FiscReceipt   Number of all fiscal receipts, after fiscalization to the present moment.
Total   It returns total of receipt as integer with sign and 12 digits.

The sums from the fiscal receipt are added to the daily sums in the RAM registers.
The command will not be executed if:
- A fiscal receipt is not opened,
- Command 53 (35h) is not successful,
- The amount paid by command 53 (35h) is less than the Total Sum of the fiscal receipt.

3Dh (61)   Set Date and Time

Syntax: <DD-MM-YY><space><HH:MM[:SS]>
Reply: no data

Changing the real-time clock is disabled from the time of making the first fiscal receipt in the current period to the moment of making daily report for the current period. Time change can be performed automatically, i.e. cash register automatically calculates and sets switch between daylight to winter calculation of time and such adjustment is imposed as a primary.
However, the fiscal printer GALEB GP-550 has a choice of manual or automatic change of time.
It always makes automatic changes daylight-winter time changeover, unless the need arises of changing the termination time. The device also has the option, so that the change will not accomplish, this selection is defined only as a service intervention.
In the case when the jumper is in active service position, time and date can be changed freely, after making daily report. Date can not be older than the date of the last daily report or the date of the fiscal. We also have active controls that are consistent with automatic correction of + / - 1 hour.

Note: All service and operator functions are unhindered allowed to 2089-the year, then, that getting into the 2090-year., the printer resumes normal to operator mode, the current settings, without the possibility of service and service interventions parameterization fiscal printer.
Any further intervention service that requires setting the expiration time of the year is not possible.

3Eh (62)   **Get back date and time**

Syntax no data
Reply <DD-MM-YY><space><HH:MM:SS>

3Fh (63)   **Display Date and Time**

Syntax: no data
Reply: no data

The lower line of the display shows current date and time: DD-MM-YY HH:MM:SS.

40h (64)   **Info on the last fiscal closure**

Syntax: no data
Reply: N,Tax1,... ,Tax9, Date

N   The number of the fiscal closure - 4 Bytes
TaxX   Net sums on all VAT code – '1, 2, 3, ...9 - 12 Bytes with symbol. For forbidden tax groups is zero.
Date   The date of the last fiscal closure - 6 Bytes. DDMMYY.

The command leads to transmitting the information on the last entry in the fiscal memory of the computer.

41h (65)   **Info on daily taxes**

Syntax: no data
Reply: Total, Tax1,...,Tax9

Total   The sum of all sales, not taxed (tax group SPACE) - 12 Bytes.
TaxX   The net sums of each VAT code - 1, 2, 3, 9 - 12 Bytes with a sign.

The net sums, divided in tax groups, are returned after the last fiscal closure until the moment of receiving the command.

43h (67)   **Get daily statistics**

Syntax: no data
Reply: Total, NegTotal, NotPaid, FiscReceipt, AllReceipt

Total   The sum of all the sales
NegTotal   The sum of all the sales with a negative price - 12 Bytes with a mark.
NotPaid   The sum of all non-paid accounts - 12 Bytes with a mark.
FiscReceipt   All issued fiscal receipts - 4 Bytes.
AllReceipt   All issued receipts up to the moment - 4 Bytes.

The sums, accumulated from the last fiscal closure to the command, are calculated and returned.

44h (68)   **Get free Fiscal No**

Syntax: no data Reply: Logical, Physical
Logical   The number of the logical places for closure - 4 Bytes,
Physical   Not used.

The number of the free Fiscal No is returned. These fields record the daily report information with clearing.

45h (69)   **Close Fiscal No (with report) /Intersection of the state**

Syntax: [<Option>[N][A]
Reply: Closure, Total, Total1,..., Total9
Option   Optional parameter, controlling the type of the generated report
            "0" – «Z» The print-out finishes with "FISCAL RECEIPT"
            "1"- "X" daily report without clearing is performed (i.e. no entry in the fiscal memory is performed and

no clearing of the registers)
"2" – "X" daily report without clearing is performed – width addition record (i.e. no entry in the fiscal memory is performed and no clearing of the registers).

N	The presence of this symbol at the end of data, forbids clearing of Operators, when a report with clear is performed.
A	The presence of this symbol at the end of data clearing summed the amounts of sales by articles. (but it don't reset the articles).
Closure	Number of the fiscal closure - 4 Bytes.
Total	The sum of all sales (no VAT) - 12 Bytes with a mark.
TotalX	The sum of each tax category, 1, 2, 3, , .9 - 12 Bytes with a mark.

The command can be performed 2000 times with argument 0. When number of rest report be less than 50, the printer every time when should be powered on, it will be printed warning about number of rest concluding rates and when every command should be performed, flag 4.3 will be set on 1. After last allowed the fiscal report is performed, fiscal memory will be locked and it can be registered more selling.

46h (70)   Register Cash-in and Cash-out

Syntax: [Amount]
Reply: ExitCode, CashSum, ServIn, ServOut

Amount	The sum to be registering (up to 9 Bytes). Depending on the mark of the digit it is interpreted as a Cash-in or a Cash-out.
ExitCode operation.	"P" - The order is entered. If the sum is "zero", the printer prints an internal receipt for registering the operation.
	"F" - The entry is forbidden. This happens when: in-cash volume is less than the entered internal cash-out, there is an opened fiscal or non-fiscal receipt,
CashSum	In-cash volume
ServIn	The sum from all commands „Internal cash-in"
ServOut	The sum of all commands "Internal cash-out"

Changes the contents of the register for "in-cash" amounts. Depending on the mark of the sum in question, it is accumulated in the register on internal cash deposit and discount. The information is not entered in the fiscal memory and is accessible until the daily closure. It is printed with the command 69 (45h) or when a daily report is initiated, without clearing from the printer itself. If the command is performed successfully, it will be generated the impulse for opening the drawer for many.

47h (71)   Report on diagnostic info

Syntax: no data
Reply: no data

This command prints an internal receipt, containing diagnostic information:
- Prints the name of fiscal device
- Software version and date fiscalization
- Prints the check sum (CRC)
- Prints the size of the database
- Prints the number of programmed items in the database
- Prints the speed of the serial port transfer
- Footer on/off
- Roll speed
- Printing intense.
- Key tone
- Adjusting the winter / summer time
- Logo on account
- Print the functions cashiers-a
- Print auto cut ON/OFF
- Print the choice of advertising ON/OFF
- Print selling items in the database YES/NO
- Print other Technical Review on the date when the same is to be executed

This command shall be not executed when a receipt is opened and there is no paper.
During a short press MENU key, the screen displays the main menu shows the choice of printing options;
pressing the "UP" make selections offered options (scroll) button until "OK" you acknowledge desired option (Print Settings (Printer Diagnostics))

4Ah (74)   Get 36-bit status

Syntax: [Option]
Reply: <SO><S1><S2><S3><S4><S5>

Option   = W Expects the printing out of all buffers of the printer,
                = X Does not wait for the printer.
Sn   Status byte N.

4Ch (76)   Status of Fiscal receipt closure

Syntax: [Option]
Reply: Open, Items, Amount [, Tender]

Option   ="T" will return the information on the current state of the account, which has to be paid by the client.
Open   = "1" if a fiscal or non-fiscal receipt is opened
                = "0" if there is no opened receipt.
Items   The number of the registered sales and the current or last fiscal receipt - 4 Bytes.
Amount   The sum of the last fiscal receipt - 9 Bytes with a sign.
Tender   The sum paid for the current or last receipt - 9 Bytes with a sign.

This command controls the application in PC to enter the status and if needed to recover and finish a fiscal operation, which was cut-off on emergency - for example at power failure.

4Fh (79)   Periodical report of fiscal memory (date-to-date)

Syntax: <Start>,<End>
Reply: no data

Start Starting date - 6 Bytes (DDMMYY)
End Ending date - 6 Bytes (DDMMYY)

The command calls the calculation and printing out of a short periodic financial report. When financial report don't exist in appropriate period, it will print null the periodical report (Regulation about look-out of fiscal documents).

50h (80)   Send Sound Signal

Syntax: no data
Reply: no data

Command sends a short Sound Signal.

53h (83)   Set mode, decimals, allowed taxes

Syntax: [<Decimals>,<TaxFlags>,<TaxX>, ...]
Reply: no data

Decimals Can be 02 or 2.
TaxFlags Is a 9-digit string, consisting of "1" and "0". Every digit represents one Tax Group ("1" to "9"). "1" means allowed tax, "0" means forbidden tax.
TaxX   Percent for Tax Group X. All tax value rates must be sets as rates which are forbidden. Values of tax rates which are forbidden, they aren't used and they are without meaning, but it must set value 0%.

The printer returns set value if in space for data there is nothing. If one parameter is set, then and another parameters must be set. The command can define tax peak 30 times including initial set. The command, in order to can perform, market should be reseted in current period. Command is allowed only immediately after fiscal closure.

5Ah (90)   Get a diagnostic information

Syntax: <Calc>
Reply: <FwRev><SP><FwDate><SP><FwTime>,<Chk>,<Sw>,<Country>,<Ser>, <FM>

Calc	if "1" , the control sum of the fiscal memory is calculated - 1 Byte.
FwRev	The version of the software utility - 4 Bytes.
SP	Space 20h.
FwDate	The date of the software utility - DDMMYY - 7 Bytes
Sp	Space (20h)
FwTime	Time of the software utility HHMM - 4 Bytes.
Chk	Control sum of the EPROM. A 4 byte string in hexadecimal format. If the control sum is 214Ah, it will be transmitted as 32h, 31h, 34h, 41h.
Sw	The keys from SW1 to SW4 - an 4 byte string with a "0" or "1".
Country	The number of the country - 1 byte.
Ser	Identification Number of Fiscal Module - 8 Bytes.
FM	The number of the Fiscal Module - 8 Bytes.

61H (97)   Get Tax Rates

Syntax: no data
Reply: Tax1, Tax2, Tax3, ...Tax9

The command returns nine tax rates separate comma.

63h (99)   Get PIB - Owner Tax Number

Syntax: no data
Reply: Text

Text   The PIB - Owner Taxation Number is presented as a string

64h (100)   Display free text in both line on the display

Syntax: Text
Reply: no data

Text   Text of up to 40 symbols, which are sent to the display. If sending of ASCII symbols, smaller than 20h (control characters) to the display is needed, these symbols are increased to 40h and are preceded by 10h.
Example:
in order to send "<ESC>K<0>" in 1Bh, 4Bh, 00h in the data field the following should be entered: 10h, 5Bh, 4Bh, 10h, 40h.

65h (101)   Set operator's password

Syntax: <OpCode>, <OldPwd>, <NewPwd>
Reply: no data

OpCode	Operator's code
OldPwd	Old password (4 - 6 digits)
NewPwd	New password (4 - 6 digits)

Enters one of the eight operator's password. The password will be asked when a fiscal receipt is opened. After initialization or clearing of the operational memory, all 8 passwords are "0000"

66h (102)   Set operator's name

Syntax: <OpCode>, <Pwd>, <OpName>
Reply: no data

OpCode   Operator's code
Pwd   Password (4 - 6 digits)
OpName   Name of the operator (up to 24 chars)

Enter one of the eight operator's names. The name and the number of the operator are printed at the beginning of each fiscal (client's) receipt. After initialization and clearing of the operational memory all operator's names are empty strings.

67h (103)   Info on the current receipt

Syntax: no data
Reply: CanVd, Tax1, Tax2, Tax3, ... Tax9

CanVd   is returning possible (sale with a zero mark) [0/1].
Tax 1   Sum, accumulated for tax group 1
Tax 2   Sum, accumulated for tax group 2
Tax 3   Sum, accumulated for tax group 3
...
Tax 9 Sum, accumulated for tax group 9

Provides information on accumulated sums under the different tax groups and weather the return of the registered items is possible.

68h (104) Reset Operators sales data

Syntax: <Operator>, <Pwd>
Reply: no data

Operator Operator number (1 ... 8 or 10 if you using base protocol)
Pwd   Operator's password (4 to 6 digits)

Clears the accumulated sums for the selected operator. If the password is not a valid one, this command will be rejected.

69h (105) Report on Operators

Syntax: no data
Reply: no data

Print the data on actual sales by operators. For each operator are printed name, number, and the total turnover.

6Ah (106) Set Drawer opening time

Syntax:[<mSec>]
Reply: no data

mSec   Drawer opening pulse length (from 5 to 25 msec)

User can set the length of the pulse for opening the Cash Drawer. After the parameter is set it will stay till a RAM RESET is performed. After a Reset the default value is 15 msec.

6Bh (107) Define and read an Articles

Syntax: <Option>[, Parameters]
Reply: ErrorCode[, Data]

Option   1 Byte, available options are: : **'P'**, **'p'**, **'D'**, **'R'**, **'F'**, **'N'**, **'f'**, **'n'**, **'X'**, **'I'**, **'C'**.
ErrorCode = P no error = F error
Parameters Depending of Option syntax is different.

Option   *= "P" Programming new article*
Syntax: <P><TaxGr><PLU>,<Sprice>,<Name>

TaxGr   A capital cyrillic letter from ('А','Г','Д','Ђ','Е','Ж','И','J','К')
PLU   An integer from 1 to 65023
Sprice   The single price
Name   Name of the article up to 32 char.

Tax group must be in 1251 ANSI code (in cyrilic). For example: Tax group A is code of $C0 (or 192 decimal).
The command will be rejected if that article already exists. If sale unit of measure different from slice, then the unit of measure must be entered in name of product or service, and the unit of measure must be separated with (/) after last character and after that to two characters which determinate an unit of measure.(e.g. Г-gram,КГ-kilogram,Т-tone,Л-liter,M-meter,M2-square meter,M3-cube meter)

Option   *= "p" programming new article*

Using of the command is the same as "P" with difference that using this command the user softer obligates to checks identify name of article before programming. This command doesn't ckeck identify of articles in the printer.

Option   *= "D" Deleting an article*

Syntax :<D> | PLU

PLU PLU number. The PLU with least number cannot be deleted. This cmd will be allowed after the daily report which is previous performed (before the first fiscal receipt) and if the total amount of sale for article equal 0.

Option   *= "R" Read an article*
Syntax :<R><PLU>
Reply :<P>,PLU>,<Grp>,<Sprice>,<Amount>,<Name>

PLU   An integer from 1 to the value, 5 digits with leading zeros
Grp   Tax group, 1 bite
Sprice   The single price, number with 2 decimal places
Amount The accumulated amount after PLU definition, number with 3 decimal places
Name   Name of the article. Up to 32 Chars.

If the article isn't in the base of the printer it will be returned syntax error. If the article doesn't define it returns 'N'.

Option   *= "C" Change a price of an article*

Syntax: <C><PLU>,<Sprice>
Reply: ErrorCode[, Data]

PLU   PLU number
Sprice   New Unit Price. Up to 10 digits
Option = "F" Find first article. Subtype of Read.
Syntax: <F>[, Parameters]
Option = "N" Find Next article. Subtype of Read.
Syntax: <N>[, Parameters]
Option = "f" Find first article with sale within a day.
Syntax: <F>[, Parameters]
Option = "n" Find Next article with sale within a day
Syntax :<N>[, Parameters]

Option = "X" Return first free PLU
Option="I" Return peak length name of artical

Subcommands "P", "p" i "C" can be get within opened fiscal receipt.
Impossible the deletion or substitution of the name of items, called the unit of measure and the tax rate from the time of making the first fiscal receipt in the current period to the moment of making daily report for the current period, it is possible that after making the daily report.

6Eh (110)   Get sales data

Syntax: no data
Reply: Cash, Debit, Cheque, Closure, Receipt

Cash   Paid in cash
Debit   Debit card payment
Cheque   Cheques payments
Closure   Current fiscal closure
Receipt   Number of the next receipt

Returns the information for the distribution of the daily sum according to the terms of payment.

6Fh (111)   Report - Price List

Syntax: <Option>
Reply: ErrorCode

Option   = "0" Only the PLU for which have sales records for the day are printed. For each PLU is printed – PLU number, Name, Unit price, Sold Quantity and Total turnover for this PLU.
                    = "1" All programmed PLUs are printed, but only with a number and unit price.

The PLUs are sorted by their number. When a Daily report with clear is performed 45h (69) and option "A" the accumulated sums are cleared.

When printing all articles from the database, you may receive the following message: **Printer stop time**!. Message represents control of excessive printer usage by user.
If the thermal printer is turned on for more than 30 seconds of continuous printing, and that was the time when the printer was turned off was a slight bit, printer, in order to protect the system activates its own function which is on display accompanied by the warning message "Printer stop time", which printer gives the system time to reduce to normal operating temperature. Thus, the corresponding waiting time is between 5 - 15 seconds, depending on the temperature and time of printing thermal printer.

The function of monitoring changes on the sale items after the reset, the value is out of the question. To use the feature must be configured via software flag in the printer after printer which tracks sales of items. Sales are monitored and quantified financially.

70h (112)   Get Operators data

Syntax: Operator
Reply: Receipts, Total, Discount, Surcharge, Void, Name

Operator   Operator's number (from 1 to 8, or 10, if you use basic protocol)
Receipts   Number of fiscal receipts issued by this Operator
Total   Number of sales and Total sum, divided by ";"
Discount   Number of discounts and Total sum of discounts
Surcharge   Number of surcharges and Total sum of surcharges
Void   Number of Voids and Total sum of Voids
Name   Operator's name

The command enables getting information which is printed when the report is performed. The sum will return as numbers with number of decimal character are set.

71h (113)   Get the number of the last printed doc (receipt, reps)
Syntax: no data
Reply: DocNum

DocNum   The number of the last printed document (7 digits)

72h (114)   Info for a fiscal No or for a fiscal period

Syntax: <Closure>, <Type>, Closure2>]
Reply: ErrorCode, TaxX (X up to 9)

Closure   Number of fiscal record
Type   Type of requested information:
                 "0" Sums according to Tax Groups
                 "1" Net sums according to Tax Groups
                 "2" Accumulated taxes for each Tax Group
                 "3" TAX rates are returned
                 "4" Sums according to Tax Groups for a certain period
                 "5" Net sums according to Tax Groups for a certain period
                 "6" Accumulated taxes for each Tax Group for a period
Closure2   number of fiscal record for "4", "5" and "6". For "0", "1","2" and "3" this field should be empty.
ErrorCode   = "F" The record is with wrong checksum and is empty
                 = "E" is empty
                 = "P" no error"
TaxX...   Sum or percentage depending on Type

73h (115) Programming a Graphic LOGO

Syntax: <RowNum>,<Data>
Reply: no data

RowNum   Shows the row which we are programming. Number from 0 to 95
Data   Graphic data. The data is transferred in hexadecimal format.

The command enables define of graphic logo which dimension is to 54x12mm (432x96 points), as user whish. Graphic logo which is sent to the printer is black-white picture and its dimension is 432x96 points. If the dimension of the picture different it must be reconstruted on some define dimension of graphic programs. Their print will be activated by command 43. The logo prints after FOOTER at beginning of every fiscal or non-fiscal receipt. For all logo defines, the command must be replied 96 times, per every line. After RESET memory of logo is empty.

77h (119) Get back date and time of last write in fiscal memory

Syntax: no data
Reply: :<P>,<DD-MM-YYYY><-><HH:mm>

DETAILED DESCRIPTION OF COMMANDS THE BASE PROTOCOL

Acronyms used in later text:

STX	start of short frame, value 0x02
STH	start of long frame, value 0x03
WAIT	wait character 0x08
LEN	length of frame
DATA	data bytes send or received in frame
CRC	frame checksum
PLU	Programmable Lookup Unit
LSB	Last Significant Bit
MSB	Most Significant Bit
ASCII	American Standard Code for Information Interchange

FRAME STRUCTURE

Short frame structure:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	1 or 2 bytes	n bytes	2 bytes

- STX=0x02
- CRC is 2 byte wide sum of all bytes from LEN to DATA including LEN and DATA, sent MSB first

Long frame structure:

STH	LEN	CMD	DATA	CRC
1 byte	2 bytes	1 or 2 bytes	n bytes	2 bytes

- STH=0x01
- LEN is length of data, 2 bytes, sent LSB first
- CRC is 2 byte wide sum of all bytes from LEN to DATA including LEN and DATA, sent MSB first

After command received from PC, if the response time is longer than 200mS, cash register sends WAIT character until operation is performed and appropriate response is sent back to the PC. For example, operations that require long time to execute like reset command. In this way communication can be stalled without an impact to the overall performance.

List of all commands:

HEX value	DEC value	Command
0x01	1	TIME SET OR READ
0x02	2	FISCALISATION
0x03	3	FISCALISATION READ
0x05	5	RESET DEVICE
0x06	6	RESET GPRS
0x08	8	READ MEMORY
0x09	9	FISCAL MEMORY STATUS
0x0A	10	READ PLU BY BARCODE
0x0B	11	PROGRAM PIB
0x0C	12	PROGRAM PLU
0x0E	14	DELETE ALL PLU
0x0F	15	PROGRAM PLU BIG
0x12	18	DELETE PLU BY BARCODE
0x13	19	PLU READ NEXT CODE
0x14	20	PROGRAM HEADER
0x15	21	RESET PLU SALE
0x16	22	PROGRAM ADVERTISEMENT MESSAGE
0x1A	26	PRINT TEXT
0x1B	27	GPRS SETTINGS
0x1D	29	READ PLU SALE
0x1E	30	TECHNICAL REVIEW
0x20	32	VATS
0x24	36	DEPARTMENTS
0x25	37	CASHIERS
0x27	39	PRINT DATA
0x28	40	BEEP
0x29	41	DAYLIGHT_SAVE_TIME
0x2A	42	DAYLIGHT_SAVE_TIME_RESPONSE
0x2C	44	READ DEVICE SETTINGS
0x2D	45	PRINT SETTINGS
0x2E	46	TEST DEVICE
0x2F	47	DEVICE STATUS
0x31	49	SELL PLU BY BARCODE
0x32	50	STORNO
0x33	51	PAYMENT
0x38	55	CASH IN / CASH OUT
0x38	56	RECEIPT STATUS
0x39	57	RECEIPT ITEM STATUS
0x3A	58	SUBSUM
0x3D	61	READ MESURE UNITS
0x3E	62	READ MODEM STATUS
0x3F	63	SHOW ON DISPLAY
0x40	64	PAPER FEED
0x42	66	PAPER CUT
0x58	88	EXECUTE FISCAL REPORTS
0x5A	90	SEND LOGO
0x5C	92	READ MEMORY ASCII
0x65	101	TEST CONNECTION
0x6A	106	PRINT PLU

TIME SET - command 0x01:

Data consists of 6 bytes that carry information in next format:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x01	6 bytes	2 bytes

HOUR	MINUTE	SECOND	DAY	MONTH	YEAR
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte

Note: year value is actual year value minus 2000

Example: set time 23:35:37, 22.09.2011

direction	hex view	ASCII view
PC -> CR	02 07 01 15 23 25 16 09 0B 00 8F	....#%....▯
CR <- PC	06 02 02 7F 00 00 81	...▯..▯
PC -> CR	06	.

TIME READ - command 0x01:

Time read command is done by sending same command as previous case with DATA omitted.

STX	LEN	CMD	CRC
1 byte	1 byte	0x01	2 bytes

Example: read time 23:42:39, 22.09.2011

direction	hex view	ASCII view
PC -> CR	02 01 01 00 02	.....
CR <- PC	06 02 07 01 15 2A 27 16 09 0B 00 98	.....*'....˜
PC -> CR	06	.

FISCALISATION - command 0x02 - FISCALISATION READ command 0x03:

Command 0x02 is used for sending and reading of IBFM, PIB and to confirm fiscalization data. First data byte after CMD defines which parameter is to read or programmed.
0x00 - IBFM data, 0x01 - PIB data and 0x02 is used to confirm fiscalization.
Command 0x03 is used for reading of IBFM, PIB
Command 0x0B is used for reading of IBFM, PIB same as 0x03 command (reserved for future use).

SEND IBFM command 0x02, 0x00:

STX	LEN	CMD	DATA			CRC
1 byte	1 byte	0x02	0x00	8 bytes		2 bytes

DATA
2 bytes IBFM letters, 6 bytes IBFM number, ASCII values, each byte one letter or digit

Example: send IBFM value AS123456

direction	hex view	ASCII view
PC -> CR	02 0A 02 00 41 53 31 32 33 34 35 36 01 D5	.....AS123456.Õ
CR <- PC	06 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81	.............□..□
PC -> CR	06	.

READ IBFM command 0x03, 0x00(same is done with 0x02, 0x01 and device converts response to 0x03)::

Time read command is done by sending 0x03 with DATA byte 0x00 only.

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x02	0x00	2 bytes

Example: read IBFM value AS123456

direction	hex view	ASCII view
PC -> CR	02 02 02 00 00 04	......
CR <- PC	06 02 09 03 41 53 31 32 33 34 35 36 01 D5	....AS123456.Õ
PC -> CR	06	.

Note: when reading IBFM received data is 9 bytes (8 bytes IBFM data and first data byte is dummy byte, in this case 0x03)

SEND PIB command 0x02, 0x01:

STX	LEN	CMD	DATA			CRC
1 byte	1 byte	0x02	0x01	9 bytes		2 bytes

DATA
9 bytes PIB number, ASCII values, each byte one letter or digit

Example: send PIB value 123456789

direction	hex view	ASCII view
PC -> CR	02 0B 02 01 31 32 33 34 35 36 37 38 39 01 EB	....123456789.ë
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

READ PIB command 0x03, 0x01(same is done with 0x02,0x01 and device converts response to 0x03):

Time read command is done by sending same command as previous case with DATA byte 0x01 only.

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x01	0x01	2 bytes

Example: read PIB value 123456789

direction	hex view	ASCII view
PC -> CR	02 02 02 01 00 05	......
CR <- PC	06 02 0A 03 31 32 33 34 35 36 37 38 39 01 EA	....123456789.ê
PC -> CR	06	.

Note: when reading PIB received data is 10 bytes(9 bytes PIB data and first data byte is dummy byte, in this case 0x03)

Confirm fiscalization command 0x02, 0x02:

When correct IBFM data, PIB data and date/time data is set, then confirming of all those parameters and finishing of fiscalization is done by next command, DATA is only one byte 0x02:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x02	0x02		2 bytes

Example:

direction	hex view	ASCII view
PC -> CR	02 02 02 02 00 06	....#%....□
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

READ MEMORY - command 0x08:

Command 0x08 is used for reading memory area. First DATA byte after CMD specifies which memory area is to be read. 0x00 is DATA FLASH memory 0x01 is RAM memory, 0x02 is flash memory and 0x03 is Fiscal memory.

Read DATA FLASH memory command 0x08, 0x00:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x08	0x00	4 bytes	2 bytes

DATA
4 bytes memory address(block of 256 bytes) from which memory is to be read

Example: read DATA FLASH memory block 100(0x64)

direction	hex view	ASCII view
PC -> CR	02 06 08 00 64 00 00 00 00 72	....d....r
CR <- PC	06 01 01 01 08 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 0A	.....ÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿ.
PC -> CR	06	.

Read RAM memory command 0x08, 0x01:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x08	0x01	4 bytes	2 bytes

DATA
4 bytes memory address(block of 256 bytes) from which memory is to be read

Example: read RAM memory block 1

direction	hex view	ASCII view
PC -> CR	02 06 08 01 01 00 00 00 00 10	....
CR <- PC	06 01 01 01 08 00 00 25 00 BC 36 00 20 00 00 00 00 00 00 00 00 04 28 00 20 02 2B 00 20 00 00 80 03 80 03 00 00 80 03 00 00 7B 03 09 04 80 00 00 00 00 14 00 40 F8 06 00 20 00 00 00 40 70 06 00 20 00 04 00 40 30 08 00 20 00 08 00 40 D0 03 00 20 00 10 00 40 70 05 00 20 00 08 01 40 40 06 00 20 00 0C 01 40 D0 09 00 20 00 00 00 00 00 AC 00 00 00 00 00 00 02 00 00 00 00 00 00 98 16 57 52 33 01 00 00 0F 00 05 00 03 00 DB 0C 7A 3A 00 20 01 00 00 20 00 00 00 00 7F FF 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B8 01 00 20 FF 00 01 00 00 00 00 00 39 30 00 00 00 00 00 00 41 72 74 69 63 6C 65 20 31 31 20 20 20 20 20 20 20 20 20 20 20 20 00 31 E8 03 00 00 06 7C 00 00 FF FF FF FF FF FF FF FF FE FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 29 A7	.......%.¼6. ... .(. .+. ..□ .□...□...{...□.. ....@ø.. ...@p.. ...@0.. ...@Ð.. ...@p.. ...@@.. ...@Ð.. .....¬. ....~.W R3.........Û.z:. ... .□ÿ..... ....,.. ÿ.......90. .....Article 11 .1è.. ..|..ÿÿÿÿÿÿÿþÿÿ ÿÿÿÿÿ........... .....)§
PC -> CR	06	.

Read PROGRAM FLASH memory command 0x08, 0x02:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x08	0x02	4 bytes	2 bytes

DATA
4 bytes memory address(block of 256 bytes) from which memory is to be read

Example: read PROGRAM FLASH memory block 1

direction	hex view	ASCII view

PC -> CR	02 06 08 02 01 00 00 00 00 11	....
CR <- PC	06 01 01 01 08 00 0C 82 44 83 44 AA F1 01 07 DA 45 01 D1 00 F0 41 F8 AF F2 09 0E BA E8 0F 00 13 F0 01 0F 18 BF FB 1A 43 F0 01 03 18 47 54 86 01 00 74 86 01 00 0A 44 10 F8 01 4B 14 F0 0F 05 08 BF 10 F8 01 5B 24 09 08 BF 10 F8 01 4B 6D 1E 05 D0 10 F8 01 3B 6D 1E 01 F8 01 3B F9 D1 64 1E 03 D0 64 1E 01 F8 01 5B FB D1 91 42 E4 D3 70 47 00 00 00 23 00 24 00 25 00 26 10 3A 28 BF 78 C1 FB D8 52 07 28 BF 30 C1 48 BF 0B 60 70 47 1F B5 14 F0 D4 FD 1F BD 10 B5 10 BD 14 F0 95 FD 11 46 FF F7 F5 FF 00 F0 15 FC 14 F0 B3 FD 03 B4 FF F7 F2 FF 03 BC 14 F0 B7 FD 00 00 09 48 80 47 09 48 00 47 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 04 48 05 49 05 4A 06 4B 70 47 00 00 6F 0C 00 08 F5 00 00 08 28 0B 00 20 28 11 00 20 28 0D 00 20 28 0D 00 20 04 46 0C E0 03 F0 71 FD 01 21 20 46 67 31	.......,DƒDªñ..Ú E.Ñ.ðAø⁻ò..°è... ð...¿û.Cð...GT†. .t†...D.ø.K.ð... ¿.ø.[$..¿.ø.Km.. Ð.ø.;m..ø.;ùÑd.. Ðd..ø.[ûÑ'BäÓpG. ..#.$.%.&.:(¿xÁû ØR.(¿0ÁH¿.`pG.µ. ðÔý.½.µ.½.ð•ý.Fÿ ÷õÿ.ð.ü.ð³ý.´ÿ÷ò ÿ.¼.ð·ý...H□G.H. Gþ¢þ¢þ¢þ¢þ¢þ¢þ¢þ ¢þ¢þ¢.H.I.J.KpG. .o...õ...(.. (.. ..(.. (.. .F.à.ðq ý.! Fg1
PC -> CR	06	.

Read Fiscal memory command 0x08, 0x03:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x08	0x03	4 bytes	2 bytes

DATA
4 bytes memory address from which memory is to be read

Example: read FISCAL memory block 100(0x64)

direction	hex view	ASCII view
PC -> CR	02 06 08 03 64 00 00 00 00 75	....d....u
CR <- PC	06 01 01 01 08 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 0A	.....ÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿ.
PC -> CR	06	.

FISCAL MEMORY STATUS - command 0x09:

Command 0x09 is used for reading fiscal status.
Read fiscal memory status 0x08:

STX	LEN	CMD	CRC
1 byte	1 byte	0x08	2 bytes

example: read fiscal memory status

direction	hex view	ASCII view
PC -> CR	02 01 09 00 0A	....
CR <- PC	06 02 2B 09 01 00 00 01 00 0F 33 20 0F 0A 0B 00 09 36 10 0A 0B 41 53 31 32 33 34 35 36 31 32 33 34 35 36 37 38 39 FF FF FF FF FF FF FF FF 0C BE	..+.......3 .... .6...AS123456123 456789ÿÿÿÿÿÿÿÿ.¾
PC -> CR	06	.

Response consists of DATA bytes organized as next:
Number of Z report - 2 bytes, coded HEX LSB first
Number of Resets - 1 byte, coded HEX
Number of VAT changes - 1 byte, coded HEX
Number of Technical reports - 1 byte, coded HEX
Fiscalization date/time - hour 1 byte, minute 1 byte, seconds 1 byte, day 1 byte, month 1 byte,
                year 1 byte (actual year -2000) coded HEX
Last date/time of record in fiscal memory - hour 1 byte, minute 1 byte, seconds 1 byte,
                day 1 byte, month 1 byte, year 1 byte (actual year -2000) coded HEX
IBFM - 8 bytes, each char/digit 1 byte HEX
PIB - 16 bytes, each char/digit 1 byte HEX

READ PLU BY CODE - command 0x0A:

Read PLU by barcode 0x0A:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x0A	4 bytes	2 bytes

DATA
4bytes HEX LSB plu code from which to perform reading

Example: read PLU barcode 123 with description "Article HWT 1234", price "12.34", VAT "2", measure unit "0", department "0"

direction	hex view	ASCII view
PC -> CR	02 05 0A D2 04 00 00 00 E5	...{........□
CR <- PC	06 02 2D 0A D2 04 00 00 41 72 74 69 63 6C 65 20 48 57 54 20 31 32 33 34 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 D2 04 00 00 00 10 FF FF 0A B2	..-.Ò...Article Galeb 1234 Ò.....ÿÿ .
PC -> CR	06	.

Response consists of DATA bytes organized as next:
- barcode 4 bytes, HEX LSB
- description 32 bytes ASCII
- price 4 bytes HEX LSB, actual price value multiplied with 100
- department 1 byte, HEX
- VAT and measurement unit share 1 byte, 4 bits LSB measurement unit, 4 bits MSB VAT
- 2 bytes reserved for future use

READ AND SEND PIB - command 0x0B:

Command 0x0B is used for sending and reading PIB number without restarting.
It used in the same way as command 0x02, and 0x03 but the device is not performing auto restart after execution, which is the case with previous commands.

PROGRAMM PLU - command 0x0C:

Program PLU command 0x0C:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x0C	n bytes		2 bytes

barcode	description	department	VAT + measurement unit	price
8 bytes	22 bytes	1byte	1byte	6 bytes

- code 4 bytes, HEX LSB
- description 32 bytes ASCII
- price 4 bytes HEX LSB, actual price value multiplied with 100
- department 1 byte, HEX
- VAT and measurement unit share 1 byte, 4 bits LSB measurement unit, 4 bits MSB VAT
- 2 bytes reserved for future use

Example: program PLU with barcode "123", description "Galeb 123", VAT "3", measurement unit "1", department "1" and price "1.00"

direction	hex view	ASCII view
PC -> CR	02 2D 0C D2 04 00 00 41 72 74 69 63 6C 65 20 48 57 54 20 31 32 33 34 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 D2 04 00 00 00 10 FF FF 0A B4	..−.Ò...Article Galeb 1234 Ò.....ÿÿà .
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

DELETE ALL PLU - command 0x0E:

Delete all PLU 0x0E:

STX	LEN	CMD	CRC
1 byte	1 byte	0x0E	2 bytes

Example: delete all PLU from cash register

direction	hex view	ASCII view
PC -> CR	02 01 0E 00 0F	.....
CR <- PC	06 08 02 02 7F 00 00 81	....•.. □
PC -> CR	06	.

A lot of wait characters 0x08 because deleting all PLU lasts long, over 10 seconds.

PROGRAMM PLU MULTI- command 0x0F:

Program PLU command 0x0C:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x0F	m*n bytes	2 bytes

barcode	description	department	VAT + measurement unit	price
8 bytes	22 bytes	1byte	1byte	6 bytes

- 1 byte PLU definition length
- barcode 4 bytes, HEX LSB
- description n bytes ASCII
- price 4 bytes HEX LSB, actual price value multiplied with 100
- department 1 byte, HEX
- VAT and measurement unit share 1 byte, 4 bits LSB measurement unit, 4 bits MSB VAT
- 2 bytes reserved for future use

DELETE PLU BY CODE - command 0x12:

Delete PLU by code 0x12:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x12	4 bytes	2 bytes

DATA
4 bytes code of the PLU to delete

Example: delete PLU with code '1234'

direction	hex view	ASCII view
PC -> CR	02 05 12 D2 04 00 00 00 ED	....Ò....í
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

READ NEXT PLU CODE - command 0x13:

Read PLU next code command 0x13:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x13	4 bytes	2 bytes

DATA
4 bytes plu code from which to perform reading

Response consists of DATA bytes organized as next:
- code of last PLU in block 4 bytes, HEX LSB

Then some number of PLU definitions organized as next:
- PLU data len 1 byte HEX
- description n bytes ASCII
- price 4 bytes HEX LSB, actual price value multiplied with 100
- department 1 byte, HEX
- VAT and measurement unit share 1 byte, 4 bits LSB measurement unit, 4 bits MSB VAT

- Number of PLU sent with the restriction of 256 bytes max frame and with PLU available. If there is for example only 2 PLU left for reading, then only those 2 PLU will be sent in the frame.

HEADER - command 0x14:

Program header 0x14:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x14	231 bytes	2 bytes

DATA
7 header rows x 33 bytes(1 byte flag and 18 bytes header row content)

Example: program HEADER

direction	hex view	ASCII view
PC -> CR	06 02 E8 14 7F 20 20 48 61 72 64 77 6F 72 6B 69 6E 67 20 54 65 63 68 6E 6F 6C 6F 67 79 20 44 2E 6F 2E 6F 2E 20 7F 20 20 20 20 20 20 20 20 20 33 37 30 30 30 20 4B 72 75 73 65 76 61 63 20 20 20 20 20 20 20 20 20 7F 20 20 20 20 20 20 20 20 20 20 20 77 77 77 2E 68 77 74 2E 72 73 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 9E 34	..è.• Galeb group d. o.o. • 1 5000 Saba c • www.galeb.com . ÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿ□4
CR <- PC	06	.

Response contains 7 header lines and each line consists of:
- flag 1 byte (0x7F print header row, 0x3F double height print header row, 0xFF do not print)
- header row content 32 bytes ASCII

Read HEADER 0x14:

STX	LEN	CMD	CRC
1 byte	1 byte	0x14	2 bytes

Example: read HEADER

direction	hex view	ASCII view
PC -> CR	02 01 14 00 15	.....
CR <- PC	06 02 E8 14 7F 20 20 48 61 72 64 77 6F 72 6B 69 6E 67 20 54 65 63 68 6E 6F 6C 6F 67 79 20 44 2E 6F 2E 6F 2E 20 7F 20 20 20 20 20 20 20 20 20 33 37 30 30 30 20 4B 72 75 73 65 76 61 63 20 20 20 20 20 20 20 20 20 7F 20 20 20 20 20 20 20 20 20 20 20 77 77 77 2E 68 77 74 2E 72 73 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 9E 34	..è.• Galeb group d. o.o. • 1 5000 Saba c • www.galeb.com . ÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿ□4
PC -> CR	06	.

Response contains 7 header lines and each line consists of:
- flag 1 byte

- header row content 32 bytes ASCII

RESET PLU SALE - command 0x15:

Reset PLU sale 0x15:

STX	LEN	CMD	CRC
1 byte	1 byte	0x15	2 bytes

Example: reset PLU sale

direction	hex view	ASCII view
PC -> CR	02 01 15 00 16	.....
CR <- PC	06 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08	.... 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 01 15 00 16
	08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08	....
	08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08	....
	01 15 00 16	....
PC -> CR	06	.

ADVERTISEMENT - command 0x16:

- Program Advertisement message command is used in the same manner as Header command, same 7 rows of 32 character length structure, with the difference of command byte , for advertisement command byte 0x17 is used.

PRINT TEXT - command 0x1A:

Print text command 0x1A:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x1A	1 byte	32 bytes	2 bytes

DATA
1 byte flags(bit 7=0 print only on receipt paper, bit7=1 print on both papers, bit6=0 print double) 32 bytes content to print

Example: Print 3 lines with content "TEST LINIJA 1", "TEST LINIJA 2",, "TEST LINIJA 3", normal print both papers

direction	hex view	ASCII view

PC -> CR	02 E1 1A FF 52 53 54 BC 55 46 48 43 BD C0 20
	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
	20 20 20 20 52 53 54 BC 55 46 48 43 BD C0 20 20
	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
	20 20 20 20 C5 41 42 56 47 44 BE 45 BF 5A 49 4A
	4B 4C 4D 4E 4F 50 20 20 20 20 20 20 20 20 20 20
	20 20 20 20 52 53 54 BC 55 46 48 43 BD C0 20 20
	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
	20 20 20 20 41 42 56 47 44 BE 45 BF 59 49 4A 4B
	4C 4D 4E 4F 50 52 53 54 BC 55 46 48 43 BD 58 C0
	20 20 20 20 9E 9F A0 A1 A2 A3 A4 A5 A6 A7 A8 A9
	AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9
	BA BB 20 20 80 81 82 83 84 85 86 87 88 89 8A 8B
	8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B
	9C 9D 20 20 50
		..á.ÿRST¼UFHC½À
		
		    RST¼UFHC½ÀJA 2
		
		    ÅABVGD¾E¿ZIJ
		KLMNOP
		    RST¼UFHC½À
		
		  ABVGD¾E¿YIJK
		LMNOPRST¼UFHC½XÀ
		    □Ÿ ¡¢£¤¥¦§¨©
		ª«¬-®¯°±²³´µ¶·¸¹
		º» □□‚ƒ„…†‡ˆ‰Š‹
		Œ□□□□‘’“”•–—˜™š›
		œ□ P.

CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

READ PLU SALE - command 0x1D:

Read only sold PLU, 0x1D:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x1D	4 bytes	2 bytes

DATA
4 bytes PLU code from which read sold

Example: read next sold PLU from code 0:

direction	hex view	ASCII view
PC -> CR	06 02 05 1D 00 00 00 00 00 22	.........."
CR <- PC	06 02 4D 1D C9 00 00 00 01 00 00 00 00 00 00 00 00 E8 03 00 00 00 00 00 00 10 0E 00 00 00 00 00 00 02 00 00 00 00 00 00 00 D0 07 00 00 00 00 00 00 24 36 00 00 00 00 00 00 03 00 00 00 00 00 00 00 E8 03 00 00 00 00 00 00 C4 09 00 00 00 00 00 00 05 2B	..M.É........... è............... ....Ð....... $6.............. è.......Ä....... ..
PC -> CR	06	.

Response consists of DATA bytes organized as next:
- code of last PLU in block 4 bytes, HEX LSB
- barcode 8 bytes, HEX LSB
- quantity 8 bytes HEX LSB, actual quantity value multiplied with 1000
- sold 8 bytes HEX LSB, actual sold value multiplied with 100

VATS - command 0x20:

Send VAT values 0x20:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x20	18 bytes	2 bytes

DATA
9 x 2 bytes HEX LSB first actual VAT values multiplied by 100

Example: Send VAT values "11.11%", "22.22%","33.33%","44.44%","55.55%","66.66%","77.77%","88.88%","99.99%"

direction	hex view	ASCII view
PC -> CR	02 13 20 57 04 AE 08 05 0D 5C 11 B3 15 0A 1A 61 1E B8 22 0F 27 04 3E	.. W.®...\.³...a .‚".'.>
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

Read VAT values 0x20:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x20	18 bytes	2 bytes

DATA
9 x 2 bytes HEX LSB first actual VAT values multiplied by 100

Example: read VAT values "**.**%", "**.**%","8.00%","18.00%","0.00%","**.**%","**.**%","**.**%","**.**%"

direction	hex view	ASCII view
PC -> CR	02 01 20 00 21	.. .!
CR <- PC	06 02 13 20 FF FF FF FF 20 03 08 07 00 00 FF FF FF FF FF FF FF FF 0C 59	... ÿÿÿÿ .....ÿÿ ÿÿÿÿÿÿ.Y
PC -> CR	06	.

DEPARTMENTS - command 0x24:

Send departments descriptions:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x24	144 bytes	2 bytes

DATA
144 bytes 8x18 bytes description ASCII for each department

Example: send departmets "Odeljenje 1", "Odeljenje 2"...

direction	hex view	ASCII view

PC -> CR	02 91 24 4F 64 65 6C 6A 65 6E 6A 65 20 31 20 20
	20 20 20 20 20 4F 64 65 6C 6A 65 6E 6A 65 20 32
	20 20 20 20 20 20 20 4F 64 65 6C 6A 65 6E 6A 65
	20 33 20 20 20 20 20 20 4F 64 65 6C 6A 65 6E
	6A 65 20 34 20 20 20 20 20 20 20 4F 64 65 6C 6A
	65 6E 6A 65 20 35 20 20 20 20 20 20 20 4F 64 65
	6C 6A 65 6E 6A 65 20 36 20 20 20 20 20 20 20 4F
	64 65 6C 6A 65 6E 6A 65 20 37 20 20 20 20 20 20
	20 4F 64 65 6C 6A 65 6E 6A 65 20 38 20 20 20 20
	20 20 20 26 D9
		.'$Odeljenje 1
		     Odeljenje 2
		       Odeljenje
		3 Odeljen
		je 4 Odelj
		enje 5 Ode
		ljenje 6 O
		deljenje 7
		Odeljenje 8
		  &Ù

CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

Read Departments 0x24: when DATA field is omitted then it is read command

STX	LEN	CMD	CRC
1 byte	1 byte	0x24	2 bytes

Example: read departments "DEP: 1", "DEP: 2" ... with zero sales

direction	hex view	ASCII view
PC -> CR	02 01 24 00 25	..$.%
CR <- PC	06 02 D1 24 44 45 50 3A 20 31 FF FF FF FF FF FF FF FF FF FF FF FF 44 45 50 3A 20 32 FF FF FF FF FF FF FF FF FF FF FF FF 44 45 50 3A 20 33 FF FF FF FF FF FF FF FF FF FF FF FF 44 45 50 3A 20 34 FF FF FF FF FF FF FF FF FF FF FF FF 44 45 50 3A 20 35 FF FF FF FF FF FF FF FF FF FF FF FF 44 45 50 3A 20 36 FF FF FF FF FF FF FF FF FF FF FF FF 44 45 50 3A 20 37 FF FF FF FF FF FF FF FF FF FF FF FF 44 45 50 3A 20 38 FF FF FF FF FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6B D1	..Ñ$DEP: 1ÿÿÿÿÿÿ ÿÿÿÿÿÿDEP: 2ÿÿÿÿ ÿÿÿÿÿÿÿÿDEP: 3ÿÿ ÿÿÿÿÿÿÿÿÿÿDEP: 4 ÿÿÿÿÿÿÿÿÿÿÿÿDEP: 5ÿÿÿÿÿÿÿÿÿÿÿÿDE P: 6ÿÿÿÿÿÿÿÿÿÿÿÿ DEP: 7ÿÿÿÿÿÿÿÿÿÿ ÿÿDEP: 8ÿÿÿÿÿÿÿÿ ÿÿÿÿ............ ....kÑ
PC -> CR	06	.

Response consists of DATA bytes organized as next:
- 8 x 18 bytes description of each department
- 8 x 8 bytes sales HEX LSB, actual sales multiplied with 100

CASHIERS - command 0x25:

Send departments descriptions:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x25	144 bytes	2 bytes

DATA
160 bytes, 16 bytes each cashier, 1 byte access level, 3 bytes access code, 12 bytes name

Example: program Cashiers, "OPERATER 1", access code 123456

direction	hex view	ASCII view
PC -> CR	02 A1 25 00 B3 15 00 4F 50 45 52 41 54 45 52 20	.¡%.³..OPERATER
	20 31 30 01 40 E2 01 4F 50 45 52 41 54 45 52 20	10.@â.OPERATER
	20 20 31 FF FF FF FF 4F 50 45 52 41 54 45 52 20	1ÿÿÿÿOPERATER
	20 20 32 FF FF FF FF 4F 50 45 52 41 54 45 52 20	2ÿÿÿÿOPERATER
	20 20 33 FF FF FF FF 4F 50 45 52 41 54 45 52 20	3ÿÿÿÿOPERATER
	20 20 34 FF FF FF FF 4F 50 45 52 41 54 45 52 20	4ÿÿÿÿOPERATER
	20 20 35 FF FF FF FF 4F 50 45 52 41 54 45 52 20	5ÿÿÿÿOPERATER
	20 20 36 FF FF FF FF 4F 50 45 52 41 54 45 52 20	6ÿÿÿÿOPERATER
	20 20 37 FF FF FF FF 4F 50 45 52 41 54 45 52 20	7ÿÿÿÿOPERATER
	20 20 38 FF FF FF FF 4F 50 45 52 41 54 45 52 20	8ÿÿÿÿOPERATER
	20 20 39 40 44	9@D
CR <- PC	06 02 02 7F 00 00 81	...•..‾
PC -> CR	06	.

Read Cashiers 0x25: when DATA field is omitted then it is read command

STX	LEN	CMD	CRC
1 byte	1 byte	0x25	2 bytes

Example: read cashiers "OPERATER 1", "OPERATER 2" ... with zero sales

direction	hex view	ASCII view
PC -> CR	02 01 25 00 26	..%.&
CR <- PC	06 02 F1 25 00 B3 15 00 4F 50 45 52 41 54 45 52	..ñ%.³..OPERATER
	20 20 31 30 FF FF FF FF 4F 50 45 52 41 54 45 52	10ÿÿÿÿOPERATER
	20 20 20 31 FF FF FF FF 4F 50 45 52 41 54 45 52	1ÿÿÿÿOPERATER
	20 20 20 32 FF FF FF FF 4F 50 45 52 41 54 45 52	2ÿÿÿÿOPERATER
	20 20 20 33 FF FF FF FF 4F 50 45 52 41 54 45 52	3ÿÿÿÿOPERATER
	20 20 20 34 FF FF FF FF 4F 50 45 52 41 54 45 52	4ÿÿÿÿOPERATER
	20 20 20 35 FF FF FF FF 4F 50 45 52 41 54 45 52	5ÿÿÿÿOPERATER
	20 20 20 36 FF FF FF FF 4F 50 45 52 41 54 45 52	6ÿÿÿÿOPERATER
	20 20 20 37 FF FF FF FF 4F 50 45 52 41 54 45 52	7ÿÿÿÿOPERATER
	20 20 20 38 FF FF FF FF 4F 50 45 52 41 54 45 52	8ÿÿÿÿOPERATER
	20 20 20 39 00 00 00 00 00 00 00 00 00 00 00 00	9............
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
	00 00 00 00 43 6C	Cl
PC -> CR	06	.

Response consists of DATA bytes organized as next:
- 10 x 16 bytes description of each department
- 10 x 8 bytes sales HEX LSB, actual sales multiplied with 100

PRINT DATA - command 0x27:

Same as PRINT TEXT command with the difference that printed data is treated as block of data and blank space is inserted and printed after the requested text printout.

BEEP - command 0x28:

Command 0x28 is used to make device beep once.
BEEP command 0x28:

STX	LEN	CMD	CRC
1 byte	1 byte	0x28	2 bytes

Example: Execute Reset "P"

direction	hex view	ASCII view
PC -> CR	02 02 28 01 00 2B	......
CR <- PC	06 02 02 7F 00 00 81	....
PC -> CR	06	.

DEVICE SETTINGS- command 0x2C:

Read Device Status 0x2F:

STX	LEN	CMD	CRC
1 byte	1 byte	0x2C	2 bytes

Example: read device status with next parameters:
Print Intensity   : 3
Print Speed   : 5
Rewinder Speed : 5
Print Footer   : 3
PC baudrate   : 19200bps
Keypad tone   : on
Cashier Logon   : on
Cashier Timeout : off
Auto switch off   : off
Advertisement   : off
LCD intensity   : 4
LCD contrast   : 4
LCD 2 intensity   : 4
LCD 2 contrast   : 4
Paper saving   : off
PLU sales   : off
Graphic logo   : off
Autocutter   : on

direction	hex view	ASCII view
PC -> CR	02 01 2C 00 2D	..,.‾
CR <- PC	06 02 0A 2C 30 53 01 80 00 44 04 F0 00 02 72	...,0S.□.D.ð..r
PC -> CR	06	.

Response consists of DATA bytes organized as next:
- 1 byte - 4bits LSB printer speed, 4bits MSB printer intensity
    (print speed can have value of 0 to 5 where 0 is fastest printing speed, print intensity can have values of 1 to 5 where 5 is highest intensity printing)
- 1 byte - 4bits LSB printer blank lines at the end of receipt, 4bits rewinder speed
    (printer blank lines value 1 to 8, rewinder speed 0 to 8)
- 1 byte - 4bits LSB PC baud rate, 4bits MSB keyboard tone
    (keypad tone value 0 is on, any other value is off, PC baud rate value 0 - 9600bps, 1 - 19200bps, 3 - 38400bps, 4 - 57600bps, 5 - 115200bps)
- 1 byte - 4bits LSB cashier timeout, 1bits MSB use/not use cashier logon
- (cashier timeout value 0 - off, 1 - 30 seconds, 2 - 60 seconds, 3 - auto a few seconds after receipt finish))
- 1 byte - 4bits LSB auto switch off timeout, 4bits MSB advertisement message choice
    (auto turn off value 0 - 10 minutes, 1 - 5 minutes, 2 - 2 minutes, 3 - off, advertisement message value 0 - do not use, 1, 2 and 3 is choice which one of the possible messages will be printed with the receipt))

- 1 byte - 4bits LSB LCD contrast, 4bits MSB LCD intensity
  (LCD contract values 0 to 7, LCD intensity values 0 to 7)
- 1 byte - 4bits LSB LCD 2 contrast, 4bits MSB LCD 2 intensity
  (LCD 2 contract values 0 to 7, LCD 2 intensity values 0 to 7)
- 1 byte - 4bits LSB paper save feature, 4bits MSB PLU sale monitoring use or not use
  (0   - OFF, other value- ON for both parameters)
- 1 byte - 4bits LSB use graphic logo or not, 4bits MSB PLU auto cutter or not
  (0- OFF, other value- ON for graphic logo, 0- ON, other value- OFF for auto cutter)

PRINT SETTINGS - command 0x2D:

Print Device Settings 0x2D, 0x00:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x2D	0x00	2 bytes

Example: print device settings

direction	hex view	ASCII view
PC -> CR	02 02 2D 01 00 30	..-../
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

Print GPRS Settings 0x2D, 0x01:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x2D	0x01	2 bytes

Example: print GPRS settings

direction	hex view	ASCII view
PC -> CR	02 02 2D 01 00 2F	..-../
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

Print GPRS state 0x2D, 0x02:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x2D	0x02	2 bytes

Example: print GPRS current state

direction	hex view	ASCII view
PC -> CR	02 02 2D 02 00 30	..-../
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

Print GPRS Settings User 0x2D, 0x03:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x2D	0x03	2 bytes

Example: print GPRS settings

direction	hex view	ASCII view
PC -> CR	02 02 2D 03 00 31	..-..1
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

Print GPRS state User 0x2D, 0x04:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x2D	0x04	2 bytes

Example: print GPRS current state

direction	hex view	ASCII view
PC -> CR	02 02 2D 04 00 32	..-../
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

TEST PRINTER - command 0x2E:

Test Printer command 0x2E:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x2E	0x01	1 byte	2 bytes

DATA
1 byte dummy

Example: test printer

direction	hex view	ASCII view
PC -> CR	02 03 2E 01 01 00 3C	......<
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

DEVICE STATUS - command 0x2F:

Read Device Status 0x2F:

STX	LEN	CMD	CRC
1 byte	1 byte	0x2F	2 bytes

Example: read device status with next parameters:
Software Version: 216
Number of Characters on Printer: 32
Number of Characters on Display: 40
Device Type: 5
Power Supply Voltage: 23.7V
PLU Frame Length: 44
PLU Description Length: 32
PLU Maximum Number: 65023
Number of Programmed PLU: 477
Next Technical: 12.09.2013.

direction	hex view	ASCII view
PC -> CR	02 01 2F 00 30	../.0
CR <- PC	06 02 15 2F D8 00 20 28 05 ED 2C 20 FF FD 00 00 DD 01 00 00 2D 6A C4 19 06 F0	.../Ø. (.í, ÿý.. Ý...-jÄ..ð
PC -> CR	06	.

Response consists of DATA bytes organized as next:
-   2 bytes HEX LSB, software version
-   1 bytes HEX number of characters on the printer
-   1 bytes HEX number of characters on the display
-   1 bytes HEX device type
-   1 bytes HEX power supply voltage in 10mV
-   1 bytes HEX frame size for PLU
-   1 bytes HEX size of the PLU description
-   4 bytes HEX LSB maximum number of PLU
-   4 bytes HEX LSB number of used(programmed) PLU
-   4 bytes HEX LSB time for next technical report, in ms from year 2000

PLU SELL BY CODE - command 0x31:

Sell PLU by barcode command 0x31:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x31	12 bytes	2 bytes

DATA
4 bytes barcode HEX LSB first, 4 bytes dummy, 4 bytes quantity HEX LSB first (actual quantity multiplied with 1000)

Example: sell PLU with code "1", quantity "1.000"

direction	hex view	ASCII view
PC -> CR	02 0D 31 01 00 00 00 00 00 00 00 E8 03 00 00 01 2A	..1........è.... *
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

PAYMENT - command 0x33:

Payment command 0x33:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x33	9 bytes	2 bytes

DATA
8 bytes amount to HEX LSB first, actual value multiplied by 100, and 1 byte payment type 0-cash, 1-cheque, 2-card

Example: pay card 1000.00

direction	hex view	ASCII view
PC -> CR	02 0A 33 A0 86 01 00 00 00 00 00 02 01 66	..3 †........f
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

STORNO - command 0x32:

Command 0x32 functions in next manner:

- DATA length is 8 then first 4 bytes represents PLU code and next 4 bytes represents quantity. If PLU code is zero (quantity is ignored) then last receipt item is canceled. If PLU quantity is zero (PLU code is ignored) then whole receipt is canceled.

Example:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x32	8 bytes		2 bytes

DATA
4bytes HEX LSB PLU code and 4 bytes quantity

Example: Storno last item on the receipt (code zero, quantity non zero)

direction	hex view	ASCII view
PC -> CR	02 09 32 00 00 00 00 01 00 00 00 00 3C	...2.........<
CR <- PC	06 02 09 32 09 00 00 00 09 00 00 00 00 4D	...2.........M
PC -> CR	06	.

Example: Storno complete receipt (code non zero, quantity zero)

direction	hex view	ASCII view
PC -> CR	02 09 32 01 00 00 00 00 00 00 00 00 3C	..2.........<
CR <- PC	06 08 02 09 32 01 00 00 00 01 00 00 00 00 3D	....2.........=
PC -> CR	06	.

CASH IN / CASH OUT - command 0x37:

CASH IN 0x37:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x37	9 bytes		2 bytes

- 8 bytes HEX LSB is the ammount
- 1 byte sign(+- input or output)

if LEN(DATA)=0 then printing of the cash state is done
if LEN(DATA)=1 then read operation is performed

direction	hex view	ASCII view
PC -> CR	02 02 37 FF 01 38	..8..;
CR <- PC	06 02 19 37 74 CE 12 00 00 00 00 00 00 00 00 00 00 00 00 00 A6 B3 7C 00 00 00 00 00 03 79	...7tÎ.......... ....¦³|......ÿ
PC -> CR	06	.

Response consists of DATA bytes organized as next:
- 4 bytes HEX LSB, receipt number
- 2 bytes HEX LSB, current fiscal state

    Bit 0 - represents if the fiscal receipt is started
    Bit 1 - represents if the payments are started
    Bit 2 - represents if the fiscal day is started
- 2 bytes HEX LSB, current items count on the receipt
- 9 x 8 bytes HEX LSB, sales of each VAT
- 3 x 8 bytes HEX LSB, amount of each payment type

Read Daily status 0x38, 0x00:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x38	0x00	2 bytes

Response consists of DATA bytes organized as next:
- 4 bytes HEX LSB, receipt number
- 4 bytes HEX LSB, Z report number
- 9 x 8 bytes HEX LSB, sales of each VAT
- 3 x 8 bytes HEX LSB, amount of each payment type
- 2 bytes HEX LSB, current state

RECEIPT STATUS - command 0x38:

Read Receipt status 0x38, 0x01:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x38	0x01	2 bytes

Example: read receipt status with next parameters
Receipt No: 2
Fiscal Receipt Started: YES
Fiscal day Started: YES
Payments Started: YES
Receipt Items Count: 6
А: 29046.60
Г: 0.00
Д: 0.00
Ђ: 0.00
Е: 0.00
Ж: 0.00
И: 0.00
Ј: 0.00
К: 0.00
CASH: 100.00

direction	hex view	ASCII view
PC -> CR	02 02 38 01 00 3B	..8..;
CR <- PC	06 02 71 38 02 00 00 00 0B 00 06 00 54 52 2C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10 27 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 C5	..q8........TR,. ....'.......... ....Å
PC -> CR	06	.

Response consists of DATA bytes organized as next:
- 4 bytes HEX LSB, receipt number
- 2 bytes HEX LSB, current fiscal state

  Bit 0 - represents if the fiscal receipt is started
  Bit 1 - represents if the payments are started
  Bit 2 - represents if the fiscal day is started
- 2 bytes HEX LSB, current items count on the receipt
- 9 x 8 bytes HEX LSB, sales of each VAT
- 4 x 8 bytes HEX LSB, amount of each payment type

Read Daily status 0x38, 0x00:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x38	0x00	2 bytes

Response consists of DATA bytes organized as next:
- 4 bytes HEX LSB, receipt number
- 4 bytes HEX LSB, Z report number
- 9 x 8 bytes HEX LSB, sales of each VAT
- 4 x 8 bytes HEX LSB, amount of each payment type
- 2 bytes HEX LSB, current state

  Bit 0 - represents if the fiscal receipt is started
  Bit 1 - represents if the payments are started
  Bit 2 - represents if the fiscal day is started

RECEIPT ITEM STATUS - command 0x39:

Read receipt item status 0x39

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x39	2 bytes	2 bytes

DATA
2 bytes HEX LSB Item position on the receipt

Example: receipt item on position 7 with PLU barcode "10", price "50.00", quantity "123.456" ...

direction	hex view	ASCII view
PC -> CR	02 02 39 05 00 40	...9..@
CR <- PC	06 02 2D 39 40 E2 01 00 0A 00 00 00 0A 00 00 00 00 00 00 00 41 72 74 69 63 6C 65 20 31 30 20 20 20 20 20 20 20 20 20 20 20 20 00 31 88 13 00 00 07 2E	..-9@â......... ....Article 10 .1^... ..
PC -> CR	06	.

Response consists of DATA bytes organized as next:
-    2 bytes HEX LSB, item number on the receipt
-    4 bytes HEX LSB sold quantity
-    the rest of the bytes is complete PLU definition as can be read with PLU read command

SUBSUM - command 0x3A:

Read SubTotal of the receipt 0x3A:

STX	LEN	CMD	CRC
1 byte	1 byte	0x08	2 bytes

Example: read SubTotal amount of the receipt which is in this case 28,946.00

direction	hex view	ASCII view
PC -> CR	02 01 3A 00 3B	..:.;
CR <- PC	06 02 09 3A 44 2B 2C 00 00 00 00 00 00 DE	...:D+,......Þ
PC -> CR	06	.

Response consists of DATA bytes organized as next:
-    8 bytes HEX LSB, actual amount multiplied with 100

READ MEASURE UNITS - command 0x3D:

Read measure units markings - command 0x3D:

STX	LEN	CMD	CRC
1 byte	1 byte	0x3D	2 bytes

Response:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x3D	18 bytes	2 bytes

DATA
9 x 2 bytes measure units description

READ MODEM STATUS - command 0x3E:

Read modem status - command 0x3E:

STX	LEN	CMD	CRC
1 byte	1 byte	0x3E	2 bytes

Response:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x3E	18 bytes	2 bytes

DATA
- 24 bytes SIM number
- 24 bytes IMEI number
- 24 bytes IMSI number
- 24 bytes operator
- 2 bytes signal quality
- 1 byte status - bit 0 SIM presence, bit 1 network registration, bit 2 working state
- 1 byte software version

SHOW ON DISPLAY - command 0x3F:

Show text on display 0x3F:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x3F	40 bytes	2 bytes

DATA
2x20 bytes of data to be displayed on the display

PAPER FEED - command 0x40:

Test Printer command 0x2E:

STX	LEN	CMD	CRC
1 byte	1 byte	0x40	2 bytes

Example: Feed paper

direction	hex view	ASCII view
PC -> CR	02 02 40 00 42	......<
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

EXECUTE FISCAL REPORT - command 0x58:

Execute Z report 0x58, 0x00:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x58	0x00	2 bytes

Example: execute Z report

direction	hex view	ASCII view
PC -> CR	02 02 58 00 00 5A	..X..Z
CR <- PC	06 08 08 08 08 08 02 02 7F 00 00 81	........□..□
PC -> CR	06	.

Execute X report 0x58, 0x00:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x58	0x00	2 bytes

Example: execute X report

direction	hex view	ASCII view
PC -> CR	02 02 58 01 00 5B	..X..[
CR <- PC	06 08 08 08 08 08 08 08 08 02 02 7F 00 00 81	..........□..□
PC -> CR	06	.

Execute Period report 0x58, 0x02:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x58	0x02	12 bytes	2 bytes

DATA
2x6 bytes of date value (2 bytes day of month, 2 bytes month, 2 bytes year - actual year - 2000) ASCII

Example: execute Period report from 01.08.2011. to 28.10.2011.

direction	hex view	ASCII view
PC -> CR	02 0E 58 02 30 31 30 38 31 31 32 38 31 30 31 31 02 C0	...X.01081128101 1.À
CR <- PC	08 08 08 08 08 08 02 02 7F 00 00 81	.........□..□
PC -> CR	06	.

SEND LOGO - command 0x5A:

Send Logo command 0x5A:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x40	n bytes	2 bytes

Data consists of the next:
- 4 bytes HEX LSB rline number (max lines 240)
- 54 bytes bit(line dot) data

Example: Send line 25

direction	hex view	ASCII view
PC -> CR	02 3B 5A 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 50 00 00 03 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 C7	..;Z........... .... fP...`.......... ....Ç
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

READ MEMORY ASCII - command 0x5C:

Command 0x5C is used for performing a reset of the device. First data byte after CMD defines which type data is to be read.
0x00 - reset, 0x01 - vat, 0x02 - Z report

Read RESETS in ASCII format 0x5C, 0x00:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x5C	0x00	1 byte	2 bytes

DATA
1 byte number of reset

Example: read RESET number 2

direction	hex view	ASCII view
PC -> CR	02 03 5C 00 02 00 61	......
CR <- PC	06 02 12 5C 00 02 32 38 31 30 31 31 20 31 37 33 39 20 50 0D 0A 03 18	...\..281011 173 9 P....
PC -> CR	06	.

Read VATS in ASCII format 0x5C, 0x01:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x5C	0x01	1 byte	2 bytes

DATA
1 byte number of VAT change

Example: read VAT number 0

direction	hex view	ASCII view
PC -> CR	02 03 5C 01 00 00 60	..\...`
CR <- PC	06 02 3D 5C 01 00 32 38 31 30 31 31 20 31 34 35 35 20 31 31 31 31 20 32 32 32 32 20 33 33 33 33 20 34 34 34 34 20 35 35 35 35 20 36 36 36 36 20 37 37 37 37 20 38 38 38 38 20 39 39 39 39 0D 0A 0B 61	..=\..281011 145 5 1111 2222 3333 4444 5555 6666 7777 8888 9999.. .a
PC -> CR	06	.

Read Z reports in ASCII format 0x5C, 0x02:

STX	LEN	CMD	DATA		CRC
1 byte	1 byte	0x5C	0x02	2 bytes	2 bytes

DATA
2 bytes HEX LSB number of Z report

Example: read Z report number 7

direction	hex view	ASCII view
PC -> CR	02 04 5C 02 07 00 00 69	..\....i
CR <- PC	06 02 34 5C 02 07 00 37 20 34 20 32 39 31 30 31 31 20 30 38 32 38 20 30 20 30 20 30 20 35 30 30 30 20 34 30 30 30 20 35 30 30 30 20 34 30 30 20 35 30 30 20 30 0D 0A 08 D2	..4\...7 4 29101 1 0828 0 0 0 500 0 4000 5000 400 500 0...Ò
PC -> CR	06	.

TEST CONNECTION - command 0x65:

Test connection command 0x65:

STX	LEN	CMD	CRC
1 byte	1 byte	0x65	2 bytes

Example: test connection

direction	hex view	ASCII view
PC -> CR	02 01 65 00 66	..e.f
CR <- PC	06 02 02 7F 00 00 81	...□..□
PC -> CR	06	.

PRINT PLU - command 0x6A:

Print all PLU 0x6A, 0x01:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x6A	0x01	2 bytes

Example: print all PLU

direction	hex view	ASCII view
PC -> CR	02 02 6A 01 00 6D	..j..m
CR <- PC	06 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81	....□..□
PC -> CR	06	.

Print sold PLU 0x6A, 0x02:

STX	LEN	CMD	DATA	CRC
1 byte	1 byte	0x6A	0x02	2 bytes

Example: print sold PLU

direction	hex view	ASCII view
PC -> CR	02 02 6A 02 00 6E	..j..n
CR <- PC	06 08 08 08 08 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81	.... □..□
PC -> CR	06	.