Table of Contents

1. COMMUNICATION PROTOCOL

The type of the protocol is Master / Slave. The communication session is always initiated by the Application Software. FPR caries out the commands send by the software application and provides a feedback depending on the result. FPR sends back an „Acknowledgement response" or „message response". All messages of the protocol are either packed or single-byte. FPR supports communication standard RS232 using the TxD, RxD and Gnd signals.

Serial port adjustment parameters:

Speed: 115200 bit/s (or 19200, 38400 , 57600 and 9600 if such is set for the FPR)

8 bit word
No parity
1stop bit

1.1. MESSAGE FORMAT FROM THE SOFTWARE APPLICATION TO THE FPR

All messages except those described in 3.4.3., sent to the FPR by the PC have the following structure:

<STX><LEN><NBL><CMD><DATA…DATA><CS><CS><ETX>

The table below contains description of the field enclosed between the symbols **<** and **>**:

Field	No. of bytes	Value
STX	1	***Message start*** – always **02h**
LEN	1	***Message length*** (number of bytes including LEN, NBL, CMD, DATA) increased by **20h** i.e. a number in the 20h - 9Fh range
NBL	1	***Message number*** increased by **20h** i.e. a number in the 20h - 9Fh range
CMD	1	***Command*** - a number in the 20h - 7Fh range(see the description of commands)
DATA.. DATA	0 - 3902	***Additional data*** – a group of data fields separated with the symbols **';'**, giving additional information needed for execution of the command (see the description of commands)
CS CS	2	***Checksum***, compiled as follows: 1) Operation XOR of all bytes from LEN to DATA inclusive = 0 .. FFh 2) Conversion of 2 bytes by adding 30h, for example: B5h -> 3Bh 35h
ETX	1	***End of message*** – always **0Ah** (LF)

The texts data of the message is sent as ASCII text with code table cp1251 (Windows 1251).

1.2. MESSAGE FORMAT FROM THE FPR TO THE SOFTWARE APPLICATION

There are several types of response depending on the message received.

1.2.1. Acknowledgement response

Positive acknowledgement – when package format is correct. It is sent when the command is acknowledged as well as when it is rejected (errors in the data sent (field <DATA…DATA>) or the command cannot be executed or the command is illegal depending on the current status of the FPR indicated by the two status bytes). It is a package message with the following format:

<ACK><NBL><STE><STE><CS><CS><ETX>

Fields description:

Field	No. of bytes	Value
ACK	1	06h
NBL	1	***No. of message*** = NBL of message related to receipt
STE STE	2	***2 error status-bytes***. A two-digit ASCII number. (see Table Errors)
CS CS	2	***Checksum***, compiled as follows: 1) Operation XOR on NBL STE и STE = 00h .. FFh 2) Conversion of 2 bytes by adding 30h, for example: B5h -> 3Bh 35h
ETX	1	0Ah (LF)

The two status-bytes are a two-digit ASCII number, in which the first digit provides information about the error in the FPR, and the second one – about a command error.

Table Errors:

Byte value	FPR errors	Byte value	Command errors
0 (30h)	OK	**0 (30h)**	OK
1 (31h)	Out of paper, printer failure	**1 (31h)**	Invalid command
2 (32h)	Registers overflow	**2 (32h)**	Illegal command
3 (33h)	Clock failure or incorrect date&time	**3 (33h)**	Z daily report is not zero
4 (34h)	Opened fiscal receipt	**4 (34h)**	Syntax error
5 (35h)	Payment residue account	**5 (35h)**	Input registers overflow
6 (36h)	Opened non-fiscal receipt	**6 (36h)**	Zero input registers
7 (37h)	Registered payment but receipt is not closed	**7 (37h)**	Unavailable transaction for correction
8 (38h)	Fiscal memory failure	**8 (38h)**	Insufficient amount on hand
9 (39h)	Incorrect password	**9 (39h)**	Not used
A (41h) : 3Ah	Missing external display		
B (42h) ; 3Bh	24hours block – missing Z report		
C (43h) < 3Ch	Overheated printer thermal head.		
D (44h) = 3Dh	Interrupt power supply in fiscal receipt (one time until status is read)		
E (45h) > 3Eh	Overflow EJ		
F (46h) ? 3Fh	Insufficient conditions		

A two-digit number is compiled depending on the type of error.
Example: Error 32 – Illegal command due to clock failure

Negative acknowledgement – It is sent when the package format is incorrect. It is 1 byte **NACK = 15h** without checksum.

Repetition request – It is sent when the FPR is busy executing the preceding command. It is 1 byte **RETRY = 0Eh** without checksum.

1.2.2. Message response

It has the format of the packed message sent by the SA to the FPR (see 3.1.) but is returned by the FPR to the SA and contains information – response to the query (see description of commands).

1.3. SHORT MESSAGES FOR TESTING THE STATUS OF THE FPR

The exchange protocol includes two unpacked single-byte codes for testing the status of the FPR, which can quickly determine the status of the device. The two codes and their meaning are shown in the table below:

Query SA	Response FPR	Meaning
04	04	FPR is on
05	05	FPR is ready for the next message
05	07	FPR out of paper
09	41	FPR is busy
09	42	FPR out of paper
09	44	FPR printer is overheated
09	40	FPR is ready

The format of the commands is described in art. 2. **DESCRIPTION OF THE COMMANDS OF FISCAL PRINTER**

2. DESCRIPTION OF THE COMMANDS
2.1. FORMAT AND PRESENTATION OF COMMANDS

All commands are described and presented using the following terms and symbols:

Key terms:

Command – the value of the CMD field of the message sent by the software application and in the message response of the the FPR.

input – structure of the fields included in the DATA field of the message sent by the software application.

output – for each command it may be one of the following:

- Acknowledgement response (see 3.2.1.).
- Structure of the fields included in the DATA field of the message response sent by the FPR (see 3.2.2).

Input data – description of the contents of the “input“ fields.

Output data – description of the contents of the “output“ fields.

Key symbols:

‘ ’ – compulsory symbol

< > – compulsory data field

<;> – field separator

[ ] – field length

{ } – non-compulsory data field

General rules:

Format of the price/value field – from 1 to 10 symbols, a floating decimal point number, preceded by +, - or SPACE.

Examples: -12.34 +56.7 8

Format of the quantity field – from 1 to 10 symbols, a floating decimal point number, up to three digits after the decimal point.

Examples: 1.234 56.78 9

Format of the rate (percentage) field – from 2 to 7 symbols, a floating decimal point number, up to two digits after the decimal point, preceded by the percent symbol - %.

Examples: -12.34% +5.67% 8.9% 10%

Payment No. 0 corresponds to the main payment – IN CASH, payment No. 4 corresponds to the special payment - VAT account, payments No. 1, 2 and 3 are programmable.

2.2. GENERAL COMMANDS

These are commands for the general functions of the FPR, related to obtaining diagnostic information and to direct access to some of the functions of the device (paper feeding, paper cutting, and display visualization).

2.2.1. Command: 20h / SP - Status

input: n. a.
output: <*Status bytes[5]*>
FPR operation: Provides detailed 5-byte information about the current status of the fiscal printer.

Input data : n. a.
Output data :

Status bytes (ST0..ST5)

Meaning of the separate bits in every byte:

ST0.0 – FM is accessible for reading only = 1 (if ST3.0, ST3.1 or ST3.2 = 1)
ST0.1 – there was a power down while a fiscal receipt was opened = 1
ST0.2 – printer overheat = 1
ST0.3 – incorrect clock= 1
ST0.4 – incorrect date = 1
ST0.5 – RAM error = 1
ST0.6 – clock failure = 1
ST0.7 – reserved for the device

ST1.0 – out of paper = 1
ST1.1 – reports accumulation overflow = 1
ST1.2 – internal for the device
ST1.3 – non-zero daily report = 1
ST1.4 – non-zero article report = 1
ST1.5 – non-zero operator report = 1
ST1.6 – non-printed copy = 1
ST1.7 - reserved for the device

ST2.0 – opened non-fiscal receipt = 1
ST2.1 – opened fiscal receipt = 1
ST2.2 – standard cash receipt = 1
ST2.3 – VAT included in the receipt = 1
ST2.4 - reserved for the device
ST2.5 – remaining less than XXX receipts in EJ
ST2.6 – Electronic Journal overflow
ST2.7 - reserved for the device

ST3.0 – no FM = 1
ST3.1 – FM failure = 1
ST3.2 – Fm overflow = 1
ST3.3 – 50 or less free FM blocks = 1
ST3.4 – numbers format: fractions = 1, whole = 0
ST3.5 – fiscalized = 1
ST3.6 – set FPR and FM numbers = 1
ST3.7 – reserved for the device

ST4.0 – automatic cutting = 1
ST4.1 – transparent display = 1
ST4.2 – reserved
ST4.3 – reserved
ST4.4 – automatic safe box opening = 1
ST4.5 – logo included in the receipt = 1
ST4.6 – reserved
ST4.7 – reserved for the device

ST5.0 – wrong SIM card= 1
ST5.1 – block – no mobile operator = 1
ST5.2 – DT have no task =1
ST5.3 – reserved
ST5.4 – reserved
ST5.5 – wrong SD card
ST5.6 – Fiscal device is unregistered = 1
ST5.7 – 1

ST6.0 – missing SIM card
ST6.1 – missing DT (modem)

2.2.2. Command: 21h / ! – Version

input: n. a.
output: *<DeviceType[1]> <;>< CertifNo[..6]><;><CertifDate[..]><;><Model><;> < CheckSum [..20]><;>*
FPR operation: Provides information about the software application version.

Input data : n. a.
Output data :

DeviceType	1 symbol for type of fiscal device – '1'- ECR, '2'-FPr
CertifNo	Certification Number of device model
CertifDate	Certification date
Model	Model name
CheckSum	Check sum

2.2.3. Command: 22h / ” – Diagnostics

input: n. a.
output: ACK
FPR operation: Prints out a diagnostic receipt.

Input data : n. a.
Output data : n. a.

2.2.4. Command: 24h / $ – Clear display

input: n. a.
output: ACK
FPR operation: Clears the display.

Input data : n. a.
Output data : n. a.

2.2.5. Command: 25h / % – Display text line 1

input: *<text[20]>*
output: ACK
FPR operation: Shows a 20-symbol text in the upper display line.

Input data :

Text	20 symbols text

Output data: n. a.

2.2.6. Command: 26h / & – Display text line 2

input: *<text[20]>*

output: ACK
FPR operation: Shows a 20-symbol text in the lower display line.

Input data :
Text 20 symbols text
Output data: n. a.

2.2.7. Command: 27h / '– Display text lines 1 and 2

input: < *text [40]*>
output: ACK
FPR operation: Shows a 40-symbol text in the two display lines.

Input data :
Text 40 symbols text
Output data: ACK

2.2.8. Command: 28h / (– Display date and time

input: n. a.
output: ACK
FPR operation: Shows the current date and time on the display.

Input data : n. a.
Output data : n. a.

2.2.9. Command: 29h /) – Cut paper

input: n. a.
output: ACK
FPR operation: Start paper cutter

Input data : n. a.
Output data : n. a.

2.2.10. Command: 2Ah / * – Safe box opening

input: n. a.
output: ACK
FPR operation: Opens the safe box.

Input data : n. a.
Output data : n. a.

2.2.11. Command: 2Bh / + – Paper feeding

input: n. a.
output: ACK
FPR operation: Feeds 1 line of paper.

Input data : n. a.
Output data : n. a.

2.3. FISCAL COMMANDS

These are commands requiring data recording in the fiscal memory of the device. Password access is required.

2.3.1.1. Command: 41h / A (1) – SET VAT and fiscal number

input: *<Password[6]> <;> <'1'> <;> <VATNo[13]><;><TypeVATNo>*
output: ACK
FPR operation: Stores the VAT and Fiscal Memory number into the operative memory.

Input data :

Password	A 6-symbol string
'1'	One symbol is compulsory 1
VATNo	13 symbol VAT number
TypeVATNo	*1 byte for type of VAT: 0- булстат, 1-ЕГН, 2- ЛНЧ, 3-служебен номер*

Output data: n. a.

2.3.1.2. Command: 41h / A (2) – Confirm VAT and fiscal number

input: *<Password[6]> <;> <'2'> <;>*
output: ACK
FPR operation: Confirm VAT and Fiscal Memory number into the operative memory.

Input data :

Password	A 6-symbol string
'2'	One symbol is compulsory 2

Output data: n. a.

2.3.2. Command: 42h / B – Tax rate change

input: *< Password [6]> <;> <TaxRateA%[2..6]> <;> <TaxRateБ%[2..6]> <;> <TaxRateB%[2..6]> <;> <TaxRateГ%[2..6]> <;> <TaxRateД%[2..6]><;> <TaxRateE%[2..6]><;> <TaxRateЖ%[2..6]><;> <TaxRateЗ%[2..6]>*
output: ACK
FPR operation: Stores a block containing the values of the tax rates into the fiscal memory. Print the values on the printer.

Input data :

Password	A 6-symbol string
Tax RateA%	Value of tax rate A with format XX.XX% - from 2 to 6 symbols
Tax RateБ%	Value of tax rate Б with format XX.XX% - from 2 to 6 symbols
Tax RateB%	Value of tax rate B with format XX.XX% - from 2 to 6 symbols
Tax RateГ%	Value of tax rate Г with format XX.XX% - from 2 to 6 symbols
Tax RateД%	Value of tax rate Д with format XX.XX% - from 2 to 6 symbols
Tax RateE%	Value of tax rate E with format XX.XX% - from 2 to 6 symbols
Tax RateЖ%	Value of tax rate Ж with format XX.XX% - from 2 to 6 symbols
Tax RateЗ%	Value of tax rate З with format XX.XX% - from 2 to 6 symbols

Output data: n. a.

2.3.3. Command: 43h / C – Change of decimal point position

input: < *Password [6]*> <;> <*'D'*>
output: ACK
FPR operation: Stores a block containing the number format into the fiscal memory. Print the current status on the printer.

Input data :

Password	A 6-symbol string
'D'	1 symbol with value 0 for whole numbers and 2 – for fractions

Output data: n. a.

2.4. PROGRAMMING COMMANDS

Set of commands, for programming the FPR configuration according to the POS requirements and the user's needs.

2.4.1. Command: 44h / D – Programming of payment types

input: *<NoPaym[1]><;> <NamePaym[10]>{<;><ExRate><CodePaym[1]}*
output: ACK
FPR operation: Preprograms the name of the type of payment.

Input data :

NoPaym	1 symbol for payment type (1 to 4)
NamePaym	10 symbols for payment type name
ExRate10	10 symbols for exchange rate of the 4th payment type, obligatory format XXXX.XXXXX maximal value 0420.00000
CodePaym	1 symbol for code payment type with name from table below or '9' or ':' for programmed name

Note: for Payment type 4 code payment type cannot be selected

'1'	Чек
'2'	Талон
'3'	В.талон
'4'	Амбалаж
'5'	Обслужв.
'6'	Повреда
'7'	Карта
'8'	Банка
'9'	Програмируемо име
':'	Програмируемо име

Output data: n. a.

2.4.2. Command: 45h / E – Programming of parameters

input: *<No.POS[4]> <;> <ParLogo[1]> <;> <ParSafe box[1]> <;> <ParAutoCut[1]> <;> <ParTranspDisp[1]> <;><ParEJ/ParArt[1]><;> < Reserv[1]><;><ParOneOper[1]>*
output: ACK
FPR operation: Programs the number of POS, printing of logo, safe box opening, display mode, cutting permission.

Input data :

NoPOS	4 symbols for number of POS

ParLogo	1 symbol of value ‘1’ for logo printout and value 0 for printout without logo
ParSafe box	1 symbol of value ‘1’ for safe box operation and value 0 for operation without safe box
ParAutoCut	1 symbol of value ‘1’ for cutter operation and value 0 for operation without cutter
ParTranspDisp	1 symbol of value ‘1’ for “transparent display” operation and value 0 for operation without “transparent display”
ParEJ / ParArt	**Device with EJ** - 1 symbol of value ‘1’ for short EJ (there is printed only fiscal receipts) and value ‘0’ for detailed EJ (there is printed all receipts) **Device with EJSD (new devices)** - 1 symbol of value ‘0’ for short “Article report” and value “1” for detailed “Article report”;
ParCurrency	1 symbol of value ‘1’ for printing currency in fiscal receipt and value ‘0’ to not print
EJFont	1 symbol of value ‘1’ for printing Low font in the journal or value ‘0’ for printing normal font
Reserv	Reserved
ParOneOper	1 symbol of value ‘1’ for work with only ONE operator

Output data: n. a.

Notes:

The logo is a graphical file in **BMP** format with dimensions 384x80 / 448 X 160 /576x80 points, which is printed at the head of every receipt

“Transparent display” is a mode, in which the FPR does not send information to the display except when executing the 25h, 26h and 27h commands. When this mode is off the FPR “uses” the display to show data during sales, at receipt finalization, etc.

2.4.3. Command: 46h / F – Programming the external display

input: *<Password[6]> <NoBytesCom1line[1]> <Com1line[8]> < NoBytesCom2line [1]> <Com2Line[8]> < NoBytesClrDis[1]> <ComClrDis[8]> <NobytesXtrCom[1]> <ComXtrCom[1]> <FlagPrecod[1]> {<PrecodTabl[64]>}*

output: ACK

FPR operation: Preprograms the external display.

Input data :

Password	A 6-symbol string
NoBytesCom1line	Number of bytes (X = 1..8), for Command: show on line 1 of the display – 1 byte
Com1line	Command string show on line 1 of the display – 8 bytes, the first X bytes are command
NoBytesCom2line	Number of bytes (Y = 1..8), for Command: show on line 2 of the display – 1 byte
Com2line	String for Command show on line 2 of the display – 8 bytes, the first Y bytes are command
NoBytesClrDis	Number of bytes (Z = 1..8), които са за Command: clear display – 1 byte
ComClrDis	String for Command clear display – 8 bytes, the first Z bytes are command
NoBytesXtrCom	Number of bytes (U = 0..8, 0 if there is no such command), for screensaver mode command – 1 byte, for hello message use line 0 of the template
ComXtrCom	String for Command for screensaver mode – 8 bytes, the first U bytes are command
FlagShift	Flag for precoding of the codetable for display cyrillization (0 – w/o precoding, 1 – with precoding) length 1 byte
PrecodTabl	Precoding table with the codes of the Cyrillic alphabet, capital and small letters

Output data: n. a.

Notes:

N command symbols should be specified for the number of bytes command. Then specify 8 bytes of control symbols, the first N of which are the command and the rest will be ignored. However, the symbols must be 8 in order to keep the format. If the display supports animation suitable for *screen-saver* - follow the above steps, otherwise set the <NoBytesXtrCom> as a 0. <FlagShift> is either 0 or 1 depending on whether a Cyrillic precoding is to be done or not. If precoding should be done input the code table.

2.4.4. Command: 47h / G – Department programming

input: *<DepNo[1..2]> <;><DepName[20]> <;> <TaxGroup[1]>*
output: ACK
FPR operation: Set data for the stated department number from the internal FPR database.

Input data:

DepNo	1 or 2 characters department number
DepName	20 characters department name
TaxGroup	1 character for tax group attachment of the department ‘А’,’Б’,’В’,’Г’,’Д’,’Е’,’Ж’ or ‘З’

Output data : n. a.

Note:

When changing the tax group attachment of department must actualize the tax groups of all articles attached to this department. Otherwise they won’t be accessible for sale (see pt.2.6.5.)

2.4.5. Command: 48h / H – Setting the date and time

input: *<DD-MM-YY[8]> <SPACE[1]> <HH:MM{:SS}[5(8)]>*
output: ACK
FPR operation: Sets the date and time and prints the current values using the RECEIPT printer.

Input data :

DD-MM-YY	A total of 8 symbols – 2 for date, month and year the last two digits, separated with the ‘-‘ symbol
SPACE	Space symbol
HH:MM{:SS}	5 or 8 symbols for hours, minutes and/or seconds, separated with the ‘:‘ symbol

Output data : n. a.

2.4.6. Command: 49h / I – Programming of header/footer lines and fiscal code

input: *<NoLine[1]> <;> <text[38]>*
output: ACK
FPR operation: Programs the contents of a header/footer lines.

Input data :

NoLine	1 symbol with value: from 1 to 7 – header lines 8 – footer line 0 – Display greeting 9 – fiscal code name
Text	Template text: 38 symbols for header/ footer line

20 symbols for Display greeting
12 symbols for fiscal code name

Output data: n. a.

2.4.7. Command: 4Ah / J – Programming of operator name and password

input: *<OpNo[2]> <;> <OpName[20]> <;> <OpPassw[4]>*
output: ACK
FPR operation: Programs the operator's name and password.

Input data :

OpNo	Symbol(s) from '1' to '20' corresponding to operator's number
OpName	20 symbols for operator's name
OpPassw	4 symbols for operator's password

Output data : n. a.

2.4.8. Command: 4Bh / K – Programming of article

input: *<PLUNo[1..5]> <;><PLUName[20]> <;> <Price[1..10]> <;> <TaxGroup[1]> <;> <DepNo[1]>*
output: ACK
FPR operation: Programs the data for a certain article (item) from the internal database. The price may have variable length, while the name field is fixed.

Input data :

PLUNo	From 1 to 5 symbols for article number
PLUName	20 symbols for article name
Price	1 to 10 symbols for article price
TaxGroup	1 symbol for article tax group – 0,1, 2,3,4,5,6 or 7
DepNo	1 symbol for article department attachment, formed in the following manner: DepNo[HEX] + 80h example: Dep01 = 81h, Dep02 = 82h … Dep19 = 93h

Output data: n. a.
Notes:
When programming department attachment, FPR checks whether the corresponding department is attached to same tax group. In case they don't match no changes will be applied. Programming of value 0 (no department attachment) is possible any time.
If no number is entered in the field of department attachment the command will execute with value 0 (no department attachment).

2.4.9. Command: 4Ch / L – Programming of Logo without setting a number (default number 0)

input: *<BMPfile[3902/9022/5822]>*
output*: Acknowledgement*
FP Action: Stores in the memory the graphic file under number 0. Prints information about loaded in the printer graphic files.
Input data:

BMPfile	*BMP file with fixed size 9022 bytes

Output data: none
Notes:
FP has the ability to store up to 10 different BMP files for logo with numbers from 0 to 9, as one of them is „active" and is printed as receipt's logo. If there is no file loaded under the number, stated as „active", FP will work as set for work without logo.

2.4.10. Command: 4Dh / M – Programming of logo with setting a number

input: *<LogoNumber[1]> <BMPfile[3902/9022/5822]>*
output: *Acknowledgement*
FP Action: Stores in the memory the graphic file under stated number. Prints information about loaded in the printer graphic files.

Input data:

LogoNumber 1 character value from '0' to '9' setting the number where the logo will be saved.

BMPfile *BMP file with fixed size 3902/9022/5822 bytes

Output data: none

2.4.11. Command: 23h / # – Set / Print active logo file number

Input*: <LogoFileNumber[1]>*
output: *Acknowledgement*
FP Action: Sets the number of logo file, which is active and will be printed as logo in the receipt header. Print information about active number.

Input data:

LogoFileNumber 1 character value from '0' to '9' or '?'. The number sets the active file, and the '?' invokes only printing of information

Output data: none

2.4.12. Command: 51h / Q – 'QE' - Enable / 'QD' - disable printing of barrcode in the end of the receipt

input: *<En/Dis[1]>*
ouput: ACK
Fpr Action: Depending of the parameter prints or doesn't barcode at the end of each clients receipt. The Barcode is EAN-13 with format specified by the next command.

Input data:

En/Dis 1 character (parameter) With value 'E' - Enable or 'D' - Disable

Output data: none

2.4.13. Command: 51h / Q – Set the format of the barcode printed in the end receipt 'QF'

input: *<'F'> <;> <CodeForm [12]>*
output: ACK
Fpr action: Defines the contents of the 12 characters of the barcode, printed in the end of each clients receipt.

Input data:

<'F'> 1 character obligatory 'F'

<CodeForm> 12 characters with possilbe combination of values:
- digits from 0 to 9 – included directly into the barcode on the corresponding possition
- 2 characters 'dd' - includes the day from the current date
- 2 characters 'mm' - includes the month from the current date
- 2 characters 'yy' - includes the last 2 digits of the year from the current date
- 4 characters 'NNNN' - Includes the receipt number
- 4 characters 'PPPP' - Includes the POS number

Output data: none

2.4.14. Command: 51h / Q – Print barcode ‘QP’

input: *<’P’> <;> <CodeType[1]> <;> <CodeLen[2]> <;> <CodeData[0..255]>*
output: ACK
Fpr Action: Prints barcode from type stated by CodeType and CodeLen and with data stated in CodeData field.

Input data:

<’P’>	1 character ‘P’
<*CodeType*>	1 symbol with possible values according to the table bellow
<*CodeLen*>	2 bytes for number of bytes according to the table
<*CodeData*>	From 0 to 255 bytes data in range according to the table

Output data: None

Table:

Barcode type	<CodeType>	<CodeLen>	Range of <CodeData>
UPC-A	**‘0’** or **‘A’**	11 or 12	Digits from ‘**0**’ to ‘**9**’
UPC-E	**‘1’** or **‘B’**	11 or 12	Digits from ‘**0**’ to ‘**9**’
JAN13 (EAN13)	**‘2’** or **‘C’**	12 or 13	Digits from ‘**0**’ to ‘**9**’
JAN8 (EAN8)	**‘3’** or **‘D’**	7 or 8	Digits from ‘**0**’ to ‘**9**’
CODE 39	**‘4’** or **‘E’**	from 1 to 10	Characters: **‘SP’ ‘$’ ‘%’ ‘+’ '-' ‘.’ ‘/'** Digits from ‘**0**’ to ‘**9**’ letters from ‘**A’** to ‘**Z**’
ITF	**‘5’** or **‘F’**	from 2 to 18 (evens only)	Digits from ‘‘**0**’ to ‘**9**’
CODABAR	**‘6’** or **‘G’**	From 1 to 15	Characters: ‘**$**’ ‘**+**’ '-' '**/**' digits from ‘**0**’ to ‘**9**’ letters from ‘**A**’ to ‘**D**’
CODE 93	**‘H’**	From 1 to 14	Bytes from **0** to **7F**
CODE 128	**‘I’**	From 1 to 12	Bytes from **0** to **7F**

The length restriction for some of the barcode types is because of the the print area not because of the barcode standard. If more data is sent the printed barcode may not be read correctly.

2.4.15. Command: 50h / P – Set invoice number range

input: *<StartNumber [10]> <;> <EndNumber[10]>*
output: ACK
FPR Action: Set invoice start and end number

Input data:

StartNumber	10 characters for start number
EndNumber	10 characters for end number

Output data: *None*

Note: To execute the command is necessary to grand following condition: the number range to be spent, not used, or not set after the last RAM reset.

2.4.16. Command: 52h / R – Programming client data base

input: *<Option[1]><ClientNo[4]><;><ClientName[26]> <;> < BuyerName [16]> <;> <VATNo[13]> <;> <FiscNo[13]><;><Address[30]>*
output: ACK
FPr Action: Program client data base.

Input data:

Option	1 symbol ‘P’ for programming
ClientNo	4 symbols for client number
ClientName	26 symbols for client name (firm name)
BuyerName	16 symbols for Buyer name (МОЛ)
VATNo	13 symbols for VAT number on client
FiscNo	13 symbols for Fiscal number on client
Address	30 symbols for address on client

Output data: *None*

2.4.17. Command: 52h / R – Reading client data base

input: *<Option[1]><ClientNo[4]*

output: *<ClientNo[4]><;><ClientName[26]> <;> < BuyerName [16]> <;> <VATNo[13]> <;> <FiscNo[13]><;><Address[30]>*

FPr Action: Program client data base.

Input data:

Option	1 symbol ‘R’ for programming
ClientNo	4 symbols for client number

Output data:

ClientNo	4 symbols for client number
ClientName	26 symbols for client name (firm name)
BuyerName	16 symbols for Buyer name
VATNo	13 symbols for VAT number on client
FiscNo	13 symbols for Fiscal number on client
Address	30 symbols for address on client

2.5. DATA READING COMMANDS

Set of commands for receiving information from the FPR about programmed values as well as additional information.

2.5.1. Command: 60h / ‘ – Reading the FPR numbers

input: n. a.

output: *<FactNo[8]> <;> <FMNo[8]>*

FPR operation: Provides information about the manufacturing number of the fiscal device.

Input data : n. a.

Output data :

FactNo	8 symbols for individual number of the fiscal device
FMNo	8 symbols for individual number of the fiscal memory

2.5.2. Command: 61h / a – Reading the TAX number

input: n. a.

output: <TAXNo[13]> <;> <TypeTAX><;> < RegNoNAP ><;> < RegDateNAP > <;>

FPR operation: Provides information about the programmed VAT number. See **2.3.1.**

Input data : n. a.

Output data :

TAXNo	13 symbols for tax number
TypeTAX	1 byte for type of TAX ‘0’ - булстат, ‘1’-ЕГН, ‘2’- ЛНЧ, ‘3’-служебен номер
RegNoNAP	Register number on the Fiscal device from NAP by registration
RegDateNAP	Date of registration

2.5.3. Command: 62h / b – Reading the tax rates

input: n. a.

output: *<xx.xx%[6]> <;> <xx.xx%[6]> <;> <xx.xx%[6]> <;> <xx.xx%[6]> <;> <xx.xx%[6]><;><xx.xx%[6]><;><xx.xx%[6]><;><xx.xx%[6]>*

FPR operation: Provides information about the current tax rates (the last values stored into the FM).

Input data : n. a.

Output data :

xx.xx%	6 symbols for tax rates of tax groups А,Б,В,Г,Д,Е,Ж and 3

2.5.4. Command: 63h / c – Reading the decimal point

input: n. a.

output: *<'D'>*

FPR operation: Provides information about the current (the last value stored into the FM) decimal point format.

Input data : n. a.

Output data:

'D'	1 symbol with value 0 for whole numbers and 2 for fractions

2.5.5. Command: 64h / d – Reading the types of payment

input: n. a.

output: *<NamePaym0[10]> <;> < NamePaym 1[10]> <;> < NamePaym 2[10]> <;> < NamePaym 3[10]> <;> < NamePaym 4[10]><;><ExRate[10]> < CodePaym 0[1]><;> < CodePaym [1]><;> < CodePaym 2[1]><;> < CodePaym 3[1]><;> < CodePaym 4[1]>*

FPR operation: Provides information about all programmed types of payment.

Input data : n. a.

Output data :

NamePaymN	10 symbols for type of payment name (0 through 4)
ExRate	10 symbols for exchange rate of payment type 4
CodePaym0	1 symbol for code of payment 0 = 0xFF (BG currency in cash)
CodePaym 1	1 symbol for code of payment 1 (default value is '7')
CodePaym 2	1 symbol for code of payment 2 (default value is '1')
CodePaym 3	1 symbol for code of payment 3 (default value is '2')
CodePaym 4	1 symbol for code of payment 4 = 0xFF (currency in cash)

2.5.6. Command: 65h / e – Reading of parameters

input: n. a.

output: *<No.POS[4]> <;> <ParLogo[1]> <;> <ParSafe box[1]> <;> <ParAutoCut[1]> <;> <ParTranspDisp[1]> <;><ParEJ/ParArt[1]><;> <ParCurrency><;> <EJfont><;> <Reserv[1]><;> <ParOneOper[1]>*

FPR operation: Provides information about the programmed number of POS and the current values of the logo and safe box options.

Input data : n. a.

Output data :

NoPOS	4 symbols for number of POS
ParLogo	1 symbol of value 1 for logo printout and value 0 for printout without logo
ParSafe box	1 symbol of value 1 for safe box operation and value 0 for operation without safe box
ParAutoCut	*1 symbol of value 1 for cutter operation and value 0 for operation without cutter*
ParTranspDisp	1 symbol of value 1 for "transparent display" operation and value 0 for

operation without “transparent display”

ParEJ / ParArt	**Device with EJ** - 1 symbol of value ‘1’ for short EJ (there is printed only fiscal receipts) and value ‘0’ for detailed EJ (there is printed all receipts) **Device with EJSD (new devices)** - 1 symbol of value ‘0’ for short “Article report” and value “1” for detailed “Article report”;
ParCurrency	1 symbol of value ‘1’ for printing currency in fiscal receipt and value ‘0’ to not print
EJFont	1 symbol of value ‘1’ for printing Low font in the journal or value ‘0’ for printing normal font
Reserv	Reserved
ParOneOper	1 symbol of value ‘1’ for work with only ONE opperator

2.5.7. Command: 66h / f – Reading of detailed printer status

input: n. a.
output: <*ParDis[1]*> <;> <*StatPRN[4]*> <;> <*FlagSecrv[1]*>
FPR operation: Provides additional status information

Input data : n. a.
Output data :

ParDis	1 symbol – connection with external display
StatPRN	4 symbols for detailed status of printer (only for printers with **ASB**) – see table below
FlagSecrv	1 symbol ‘J’ for closed jumper or ‘ ‘ for opened
Status bytes (ST0..ST3)	ST0.0 – 0 ST0.1 – 0 ST0.2 – signal level for drawer ST0.3 – printer not ready ST0.4 – 1 ST0.5 – open cover ST0.6 – paper feed status ST0.7 – 1 ST1.0 – reserved ST1.1 – reserved ST1.2 – reserved ST1.3 – cutter error ST1.4 – 0 ST1.5 – fatal error ST1.6 – overheat ST1.7 – 1 ST2.0 – JNP (journal paper near end) ST2.1 – RNP (receipt paper near end) ST2.2 – JPE (journal paper end) ST2.3 – RPE (receipt paper end) ST2.4 – 0 ST2.5 – reserved ST2.6 – reserved ST2.7 – 1 ST3.0 – print data buffer data exists ST3.1 – reserved ST3.2 – reserved ST3.3 – reserved ST3.4 – 0 ST3.5 – reserved ST3.6 – reserved

2.5.8. Command: 67h / g – Read of department registers

input*: <DepNo[2]>*

output: *< DepNo [2]> <;><DepName[20]> <;> <VATClass[1]> <;> <Turnover[11]> <;> <QtySold[11]> <;> <NoLastrep[5]> <;> <DateHour[16]>*

FPr Action: Provides information for the programmed data, the turnover from the stated department number

Input data :

DEPNo	2 symbols for department number

Output data :

DEPNo	2 symbols for department number
DEPName	20 symbols for department name
TaxGroup	1 symbol for department tax group with optional values ‘А’,’Б’,’В’,’Г’,’Д’,’Е’,’Ж’ and ‘3’
Turnover	11 symbols for accumulated turnover of the article
QtySold	11 symbols for sold quantity of the department
NoLastRep	5 symbols for the number of last Z Report
DateHour	16 symbols for date and hour on last Z Report

2.5.9. Command: 68h / h – Reading the date and time

input: n. a.

output: *<DD-MM-YYYY[10]> <SPACE[1]> <HH:MM[5]>*

FPR operation: Provides information about the current date and time.

Input data : n. a.

Output data :

DD-MM-YYYY	10 symbols – for date, month, and year, separated with the symbol ‘-‘
SPACE	Space symbol
HH:MM	5 symbols for hours, minutes, seconds separated with the symbol ‘:‘

2.5.10. Command: 69h / i – Reading the header/footer lines

input: *<No.Line[1]>*

output: *<No.Line[1]> <text[line lenght]>*

FPR operation: Provides information about the contents of the line.

Input data :

Line Number	1 symbol with value from 0 to 9

Output data:

Line Number	1 symbol with value from 0 to 9;
Text	Line text (32/38/48 symbols) or 20 symbols for display massage or 12 symbols for code name

2.5.11. Command: 6Ah / j – Reading the operator’s name and password

input: *<OpNo[2]>*

output: *< OpNo[2]> <;> <OpName[20]> <;> <OpPass[4]>*

FPR operation: Provides information about an operator’s name and password.

Input data :

OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number

Output data:

OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number
OpName	20 symbols for operator’s name
OpPassword	4 symbols for operator’s password

2.5.12. Command: 6Bh / k – Reading of article

input: *<PLUNo[5]>*

output: *<PLUNo[5]> <;> <PLUName[20]> <;> <Price[11]> <;> <TaxGroup[1]> <;> <turnover[11]> <;> <SaleQty[11]> <;> <NoLastrep[5]> <;> <DateHour[16]> <;> <DepNo[1]> <;>*

FPR operation: Provides information about the registers of the specified article

Input data :

PLUNo	From 1 to 5 symbols for article number

Output data:

PLUNo	5 symbols for article number
PLUName	20 symbols for article name
Price	11 symbols for article price
TaxGroup	1 symbol for article tax group ‘А’,’Б’,’В’,’Г’,’Д’,’Е’,’Ж’ or ‘З’
Turnover	11 symbols for turnover by this article
SaleQty	11 symbols for sale qty
NoLastRep	5 symbols for the number of last Z Report
DateHour	16 symbols for date and hour on last Z Report
DepNo	1 symbol for article department attachment, formed in the following manner: DepNo[HEX] + 80h example: Dep01 = 81h, Dep02 = 82h … Dep19 = 93h

2.5.13. Command: 6Ch / l – Logo printing

input: *<number>*

output: ACK

FPR operation: Prints the programmed graphical logo with the stated number.

Input data :

Number	Number of logo to be printed. If missing prints logo with number 0

Output data : n. a.

2.6. RECEIPT OPERATIONS COMMANDS

These commands are used mainly for FPR sales registration. The group also includes some auxiliary commands providing information for the current receipt as well as commands for RA and PO amounts.

2.6.1. Command: 2Eh / . – Non-fiscal receipt opening

input: *<OpNo[2]> <;> <OpPassw[4]> {<;> <ParEJ[1]><;><ParPrint>}*

output: ACK

FPR operation: Opens a non-fiscal receipt assigned to the specified operator (see 1.2.5.). **ParEJ** - for adding receipt to EJ no matter what it programmed type EJ (short or detailed).

Input data :

OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number
OpPassw	4 symbols for operator’s password
ParEJ	1 symbol with value ‘0’ for adding receipt to EJ by programmed type EJ (short or detailed) and ‘J’ – for always add receipt to EJ
ParPrint	1 symbol with value ‘0’ for steps printing or ‘1’ for postponed print

Output data: n. a.

2.6.2. Command: 2Fh / / – Non-fiscal receipt closure

input: n. a.

output: ACK
FPR operation: Closes the non-fiscal receipt.

Input data : n. a.
Output data : n. a.

2.6.3. Command: 30h / 0 –Fiscal receipt opening

input: *<OpNo[2]> <;> <OpPassw[4]> {{<;> <ParDetail[1]> <;> <ParVAT[1]> }<;> < ParInvoice [1]> <;> <FirmName[26]> <;> <NameClient[16]> <;> <ClientVAT[13]> <;> <ClientFiscNo[13]> <;> <Address[30]>*
output: ACK
FPR operation: Opens a fiscal receipt (or invoice) assigned to the specified operator and parameters for receipt format and VAT (and info for client – if invoice)
Input data :

OpNo	Symbol(s) from '1' to '20' corresponding to operator's number	
OpPassw	4 symbols for operator's password	
ParDetail	1 symbol with value '0' or '1' no matter when ParInvoice =1	
ParVAT	1 symbol with value '0' or '1' no matter when ParInvoice =1	
ParInvoice	1 symbol with value	'0' – standard fiscal receipt '1' – invoice '2' – standard fiscal receipt with postponed print '3' – invoice with postponed print
FirmName	26 symbols for name of firm	
NameClient	16 symbols for name of client	
ClientVAT	13 symbols for client VAT number	
ClientFiscNo	13 symbols for client Fiscal number	
Address	30 symbols for client address	

Output data: n. a.

Note:

This command have 3 ways for using: for standard fiscal receipt, for detailed fiscal receipt (invoice) and each of them combined with postponed print. (see 2.6.3.1., 2.6.3.2., 2.6.3.3.)

2.6.3.1. Command: 30h / 0 – Standard fiscal receipt opening

input: *<OpNo[2]> <;> <OpPassw[4]> {<;> <ParDetail[1]> <;> <ParVAT[1]>} <;> < ParInvoice [1]>*
output: ACK
FPR operation: Opens a fiscal receipt assigned to the specified operator and parameters for receipt format and VAT.
Input data :

OpNo	Symbol(s) from '1' to '20' corresponding to operator's number	
OpPassw	4 symbols for operator's password	
ParDetail	1 symbol with value '0' for brief format and '1' for detailed format	
ParVAT	1 symbol with value '0' or '1' for receipt with or without VAT	
ParInvoice	1 symbol with value	'0' – standard fiscal receipt '2' – standard fiscal receipt with postponed print

Output data: n. a.

Note:

When there is standard fiscal receipt: parameters after *ParInvoice* are not in use. In this case *ParInvoice* is used only if postponed print (see point 2.6.3.3.)

Parameters *ParDetail* and *ParVAT* are not obligatory but if they are to be present they should always be together. If they are not specified the command is executed with their default values equal '0'.

2.6.3.2. Command: 30h / 0 – Detailed fiscal receipt opening – Invoice

input: *<OpNo[2]> <;> <OpPassw[4]> <;> <ParDetail[1]> <;> <ParVAT[1]> <;> < ParInvoice [1]> <;> <FirmName[26]> <;> <NameClient[16]> <;> <ClientVAT[13]> <;> <ClientFiscNo[13]> <;> <Address[30]>*

output: ACK

FPR operation: Opens a fiscal receipt - invoice assigned to the specified operator and info for client.

Input data :

OpNo	Symbol(s) from '1' to '20' corresponding to operator's number
OpPassw	4 symbols for operator's password
ParDetail	1 symbol with value '0' or '1' no matter when ParInvoice =1
ParVAT	1 symbol with value '0' or '1' no matter when ParInvoice =1
ParInvoice	1 symbol with value: '1' – invoice '3' – invoice with postponed print
FirmName	26 symbols for name of firm
NameClient	16 symbols for name of client
ClientVAT	13 symbols for client VAT number
ClientFiscNo	13 symbols for client Fiscal number
Address	30 symbols for client address

Output data: n. a.

Note:

Invoice can be opened only if these are available invoice numbers see.

2.6.3.3. Command: 30h / 0 – open fiscal receipt with postponed (session) printing (speed up the printing)

This option allows speed up of the printing by avoiding the effect of the chopped printing – result of the very fast printing and not so fast communication with the AS (data is printed faster then could be sent). This is done by using the <ParInvoice[1]> as follows:

ParInvoice	1 character with value '2' or '3' respectively for standard receipt or invoice

Fpr action: Opens fiscal receipt, all the next commands will be executed but won't be printed. The data is stored to be printed in one time for the whole receipt, when AS sent information for receipt closure (36h or 38h).

2.6.4. Command: 31h / 1 – Sale/correction of article

input: *<namePLU[36]> <;> <TaxGroup[1]> <;> <Price[1..10]> {<'*'> <Qty[1..10]>} {<','> <disc/add[2..7]>}{<':'><disc/add[2..8]>}*

output: ACK

FPR operation: Registers the sale or correction of article with specified name, price, quantity, tax group and/or discount/addition on the transaction.

Input data :

namePLU	36 symbols for article's name
TaxGroup	1 symbol for article's tax group with optional values 'А','Б','В','Г','Д','Е','Ж' or '3'
Price	1 to 10 symbols for article's price
''*	1 symbol '*' indicating the presence of quantity field
Quantity	1 to 10 symbols for quantity
','	1 symbol ',' indicating the presence of discount/addition field
Disc/add	2 to 7 symbols for percentage of discount/addition
':'	1 symbol ',' indicating the presence of value discount/addition field
Disc/add	2 to 8 symbols for value of discount/addition

Output data : n. a.

Notes:

If the price field is preceded by a '-' the command is executed by the FPR as a correction/void (only if the amount of the corresponding tax group of the receipt is sufficient).

The quantity fields are not obligatory. If no value is stated for them the FPR executed the command for a default quantity of 1.000 (see 2.2.1.).

The discount/addition fields are not obligatory. The discount/addition must be in percents and is determined from the presence or absence of the '-' symbol.

2.6.5. Command: 32h / 2 – sale/correction of article from FPR database

input: <sign*[1]*> <;> <*NoPLU[5]*> {<'$'> <*Price[1..8]*>} {<'*'> <*Qty[1..10]*>} {<',’> <*disc/add[2..7]*>} {<':'> < *disc/add[2..8]* >}

output: ACK

FPR operation: Registers the sale or correction of a specified quantity of an article of the internal database of the FPR.

Input data :

Sign	1 symbol with optional value: 'SP' or '+' for sale or '-' for correction
NoPLU	5 symbols for the number of the article of FPR's database
''*	1 symbol '*' indicating the presence of quantity field
Qty	1 to 10 symbols for article's quantity sold
','	1 symbol ',' indicating the presence of discount/addition field
Disc/add	2 to 7 for percentage of discount/addition
':'	1 symbol ',' indicating the presence of discount/addition field
Disc/add	2 to 8 symbols for percentage of discount/addition (with "-" before when is discount)
'$'	1 symbol '$' indicating the presence of price
price	1 to 8 symbols for sale price

Output data : n. a.

Notes:

The FPR will perform a correction operation only if the same quantity of the article has already been sold.

If the selected article has programmed department attachment, the FPR will execute the command only if the article and the certain department are in same tax group.

2.6.6. Command: 33h / 3 – Subtotal

input: <*Print[1]*> <;> <*VisDispl[1]*> {<':'> <*Value[1..10]*>} {<',’> <*percent[2..7]*>}

output: <*Value[10]*>

FPR operation: Calculates the subtotal amount with printing and display visualization options. Provides information about values of the calculated amounts. If a percent or value discount/addition has been specified the subtotal and the discount/addition value will be printed regardless the parameter for printing.

Input data :

Print	1 symbol with value 1 or 0 stating whether the subtotal is to be printed or not
VisDispl	1 symbol with value 1 or 0 stating whether the subtotal is to be displayed or not
':'	1 symbol ':' indicating the presence of value discount/addition field
Value	1 to 10 symbols for the value of the discount/addition
','	1 symbol ',' indicating the presence of percent discount/addition field
Percent	2 to 7 symbols for the percentage value of the discount/addition

Output data:

Value	10 symbols for the value of the subtotal amount

Notes:

The discount/addition may be either values or percentages.

When the discount/addition is a percentage the amount is distributed proportionally over the turnover items and is automatically transferred to the turnovers of the corresponding tax groups.

A value discount/addition may be specified only if all sales are of articles (items) belonging to one and the same tax group.

2.6.7. Command: 34h / 4 – Sale/correction of department

input: *<nameSale[36]> <;> <DepNo[1]> <;> <Price[1..10]> {<'*'> <Qty[1..10]>} {<','> <disc/add[2..7]>}{<':'><disc/add[2..8]>}*

output: ACK

FPR operation: Registers the sale or correction in this department with specified name, price, quantity, tax group and/or discount/addition on the transaction.

Input data :

NameSale	36 symbols for name of sale
DepNo	1 symbol for article department attachment, formed in the following manner: DepNo[HEX] + 80h example: Dep01 = 81h, Dep02 = 82h … Dep19 = 93h
Price	1 to 10 symbols for article's price
''*	1 symbol '*' indicating the presence of quantity field
Quantity	1 to 10 symbols for quantity
','	1 symbol ',' indicating the presence of discount/addition field
Disc/add	2 to 7 symbols for percentage of discount/addition
':'	1 symbol ',' indicating the presence of value discount/addition field
Disc/add	2 to 8 symbols for value of discount/addition

Output data: n.a.

Notes:

Sale/correction will be collected in VAT group on department, value and qty will be gain (removed) in this department.

For correction use '-' before value price.

Fields for qty are optional. If there no set qty default value is 1.000 (see 2.2.1.).

Fields for Disc/add are optional. Disc/add can be only in percents, for discount use '-' before value. Cannot make correction with Disc/add.

2.6.8. Command: 35h / 5 – Payment

input: *<TypePaymnt[1]> <;> <NoChange[1]> <;> <amount[1..10]> {<*><ParChangeType>}*

output: ACK

FPR operation: Registers the payment in the receipt with specified type of payment and amount received (if the payment type is 1-4 the amount of change due is not obligatory.)

Input data :

TypePaymnt	1 symbol with values '0','1','2','3'or'4', according to the type of payment
NoChange	1 symbol with value 0 or 1 stating whether a change is to be calculated or not
Amount	1 to 10 characters for received amount or one character "" (quotation mark) or "-" for void all payments in receipt;
ParChangeType	1 symbols with value 0, 1 or 2 for change type. 0 is for change in cash, 1 is for the same type as the payment and 2 is for payment type 4 (currency)

Output data: n.a.

Notes:

By executing this command the FPR enters the payment mode. No further sales and/or corrections are allowed.

If the amount received is equal to or greater than the grand total amount (the amount due) the FPR quits the procedure and calculates the change in the specified type of payment except in the cases when ParNoChange is not 1 – in such cases the operator is liable for the stated amount.

If the amount received is less than the amount due a new due amount is calculated and set as a base for consequent payment.

The receipt can be finalized only when the last payment transfer is sufficient to cover the whole amount due (the grand total amount), i.e. the payment procedure has been finalized.

2.6.9. Command: 36h / 6 – Automatic receipt closure

input: n. a.
output: ACK
FPR operation: Close the fiscal receipt, paying the due amount in cash

Input data : n. a.
Output data : n. a.

2.6.10. Command: 37h / 7 – Free text printing

input: <*text[max46]***>**
output: ACK
FPR operation: Prints a free text.

Input data :

Text	Free text – max 46 symbols

Output data: ACK

2.6.11. Command: 38h / 8 – Fiscal receipt closure

input: n. a.
output: ACK
FPR operation: Closes the fiscal receipt.

Input data : n. a.
Output data : n. a.

2.6.12. Command: 39h / 9 – Void all sales and close fiscal receipt.

input: n. a.
output: ACK
FPR operation: Available only if receipt is not closed. Void all sales in receipt and close the fiscal receipt. If payment is started, then finish payment and close the receipt.

Input data : n. a.
Output data : n. a.

2.6.13. Command: 3Ah / : – Print a copy of the last document

input: *n.a.*
output: ACK

FPR operation: Prints a copy of the last receipt issued

Input data : n. a.
Output data : n. a.

2.6.14. Command: 3Bh / ; – Non-fiscal RA and PO amounts

input: <*OpNo[2]*> <;> <*OpPasswd[4]*> <;> <*TypePaymnt[1]*> <;> <*amount[1..10]*>{<'@'[1]><*Text[36]*>}

output: ACK

FPR operation: Lodges/withdraws the stated amount in the specified type of payment from the registers of the specified operator (the '-' symbol preceding the amount means withdrawal).

Input data :

OpNo	Symbol from 1 to 20 corresponding to the operator's number
OpPassw	4 symbols for operator's password
TypePaymnt	1 symbol with value '0','1','2','3'or'4', according to the type of payment
Amount	1 to 10 symbols for the amount lodged/withdrawn
'@'	Symbol @
Text	Text - 34 bytes

Output data: n. a.

2.6.15. Command: 3Eh / > – Discount/ addition

input: <*Type[1]*> <;> <*VisDispl[1]*> {<':'> <*Value[1..10]*>} {<','> <*percent[2..7]*>}**

output: ACK

FPR operation: Percent or value discount/addition over sum of transaction or over subtotal sum depended from byte "type".

Input data :

Type	1 symbol with value 2 or 1 or 0 stating type of discount/addition: 2 - defined from the device 1 - discount/addition is over the subtotal 0 - discount/addition is over the transaction sum
VisDispl	1 symbol with value 1 or 0 stating whether the subtotal is to be displayed or not (if operation is over subtotal)
':'	1 symbol ':' indicating the presence of value discount/addition field
Value	1 to 10 symbols for the value of the discount/addition
','	1 symbol ',' indicating the presence of percent discount/addition field
Percent	2 to 7 symbols for the percentage value of the discount/addition

Output data: ACK

2.7. COMMANDS FOR READING THE DATA IN FPR'S REGISTERS

This set of commands provides information about the status of FPR's registers without causing a device activity, i.e. the information is obtained through the communication interface without printing or display visualization.

2.7.1. Command: 6Dh / m – Reading of amounts by tax groups

input: n. a.

output: *<AmntTaxGrA[11]> <;> <AmntTaxGrБ[11]> <;> <AmntTaxGrB[11]> <;><AmntTaxGrГ[11]> <;><AmntTaxGrД[11]> <;> <AmntTaxGrE[11]> <;> <AmntTaxGrЖ[11]> <;> <AmntTaxGr3[11]> <;> <SumTaxGr [11]>*

FPR operation: Provides information about the accumulated amount by tax group.

Input data : n. a.

Output data :

AmntTaxGr	11 symbols for the amount accumulated in the specified tax group
SumTaxGr	11 symbols for sum of all Tax groups

2.7.2. Command: 6Eh / n – Reading of registers – 0 (on hand)

input: *<'0'>*

output: *<'0'> <;> <AmntPmnt0[11]> <;> < AmntPmnt 1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;>*

FPR operation: Provides information about the amounts on hand by type of payment.

Input data :

<'0'>	1 symbol obligatory '0'

Output data:

<'0'>	1 symbol obligatory '0'
AmntPmnt	11 symbols for the accumulated amount by payment type

2.7.3. Command: 6Eh / n – Reading of registers – 1 (general)

input: *<'1'>*

output: *<'1'> <;> <NoCust[5]> <;> <NoDisc[5]> <;> <AmntDisc[11]> <;> <NoAdd[5]> <;> <AmntAdd[11]> <;> <NoVoid[5]> <;> <AmntVoid[11]> <;>*

FPR operation: Provides information about the number of customers (number of fiscal receipt issued), number of discounts, additions and corrections made and the accumulated amounts.

Input data :

<'1'>	1 symbol obligatory '1'

Output data:

<'1'>	1 symbol obligatory '1'
NoCust	5 symbols for number of customers
NoDisc	5 symbols for number of discounts
AmntDisc	11 symbols for accumulated amount of discounts
NoAdd	5 symbols for number of additions
AmntAdd	11 symbols for accumulated amount of additions
NoVoid	5 symbols for number of corrections
AmntVoid	11 symbols for accumulated amount of corrections

2.7.4. Command: 6Eh / n – Reading of registers – 2 (RA)

input: *<'2'>*

output: *<'2'> <;> <AmntPmnt0[11]> <;> < AmntPmnt 1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;> <NoRA[5]> <;> < SumPmnt[11]> <;>*

FPR operation: Provides information about the RA amounts by type of payment and the total number of operations.

Input data :

<'2'>	1 symbol obligatory '2'

Output data:

<'2'>	1 symbol obligatory '2'
AmntPmnt	11 symbols for RA by type of payment
NoRA	5 symbols for the total number of operations
SumPmnt	11 symbol for sum of all payments

2.7.5. Command: 6Eh / n – Reading of registers – 3 (PO)

input: *<'3'>*

output: *<'3'> <;> <AmntPmnt0[11]> <;><AmntPmnt 1[11]> <;>< AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;>< AmntPmnt 4[11]><;><NoPO[5]> <;>< SumPmnt[11]> <;>*

FPR operation: Provides information about the PO amounts by type of payment and the total number of operations.

Input data :

<'3'>	1 symbol obligatory '3'

Output data:

<'3'>	1 symbol obligatory '3'
AmntPmnt	11 symbols for PO amount by type of payment
NoPO	5 symbols for the total number of operations
SumPmnt	11 symbol for sum of all payments

2.7.6. Command: 6Eh / n – Reading of registers – 4 (received)

input: *<'4'>*

output: *<'4'> <;> < AmntPmnt 0[11]> <;> < AmntPmnt 1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;>*

FPR operation: Provides information about the amounts received from sales by type of payment.

Input data :

<'4'>	1 symbol obligatory '4'

Output data:

<'4'>	1 symbol obligatory '4'
AmntPmnt	11 symbols for amount received from sales by type of payment

2.7.7. Command: 6Eh / n – Reading of registers – 5 (counters)

input: *<'5'>*

output: *<'5'> <;> <NoREP[5]> <;> <NoLastFMBlock[5]> <;> <NoEJ[5]> <;> <DateTime[16]> <;>*

FPR operation: Provides information about the current reading of the daily-report-with-zeroing counter, the number of the last block stored in FM, the number of EJ and the date and time of the last block storage in the FM.

Input data :

<'5'>	1 symbol obligatory '5'

Output data:

<'5'>	1 symbol obligatory '5'
NoREP	5 symbols for number of the last report
NoLastFMBlock	5 symbols for number of the last FM block
NoEJ	5 symbols for number of EJ
DateTime	16 symbols for date and time of the last block storage in FM

2.7.8. Command: 6Eh / n – Reading of registers – 6 (returned)

input: *<'6'>*

output: *<'6'> <;> < AmntPmnt 0[11]> <;> < AmntPmnt 1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;>*

FPR operation: Provides information about the amounts returned as change by type of payment.

Input data :

<'6'>	1 symbol obligatory '6'

Output data:

<'6'>	1 symbol obligatory '6'
AmntPmnt	11 symbols for amount returned as change by type of payment

2.7.9. Command: 6Eh / n – Reading of registers – 7 Sums in FP

input: *<'7'>*
output: *<'7'> <;> <SumTurn[14]> <;> <SumVAT[14]> <;>*
FPR operation: Provides information about the current reading of the daily-report-with-zeroing counter, the number of the last block stored in FM, the number of EJ and the date and time of the last block storage in the FM.

Input data :

<'7>	1 symbol obligatory '7'

Output data:

<'7'>	1 symbol obligatory '7'
SumTurn	14 symbols for sum of turnover in FP
SumVAT	14 symbols for sum of VAT value in FP

2.7.10. Command: 6Eh / n – Reading of registers – 9 electronic signature of last daily report

input: *<'9'>*
output: *<'9'> <;> <SYGN[40]> <;>*
FPR operation: Provides information about electronic signature of last daily report.

Input data :

<'9'>	1 symbol obligatory '9'

Output data:

<'9'>	1 symbol obligatory '9'
SYGN	40 symbols electronic signature

2.7.11. Command: 6Eh / n – Reading of registers – : Sum of daily turnover from EJ-SD

input: *<':'>*
output: *<':'> <;> <DailyTurnover[14]> <;>*
FPR operation: Provides information about electronic signature of last daily report.

Input data :

<':'>	1 symbol obligatory '9'

Output data:

<':'>	1 symbol obligatory '9'
DailyTurnover	14 symbols for daily turnover

2.7.12. Command: 6Fh / o – Reading of operator's report – 1 (general)

input: *<'1'> <;> <Op.No[2]>*
output: *<'1'> <;> <OpNo[2]> <;> <NoCust[5]> <;> <NoDisc[5]> <;> <AmntDisc[11]> <;> <NoAdd[5]> <;> <AmntAdd[11]> <;> <NoVoid[5]> <;> <AmntVoid[11]> <;>*
FPR operation: Provides information about the total number of customers, discounts, additions, corrections and accumulated amounts by operator.

Input data :

<'1'>	1 symbol obligatory '1'
OpNo	Symbol(s) from '1' to '20' corresponding to operator's number

Output data:

<'1'>	1 symbol obligatory '1'
OpNo	Symbol(s) from '1' to '20' corresponding to operator's number
NoCust	5 symbols for number of customers
NoDisc	5 symbols for number of discounts
AmntDisc	11 symbols for accumulated amount of discounts
NoAdd	5 symbols for number of additions
AmntAdd	11 symbols for accumulated amount of additions
NoVoid	5 symbols for number of corrections
AmntVoid	11 symbols for accumulated amount of corrections

2.7.13. Command: 6Fh / o – Reading of operator's report – 2 (RA)

input: *<'2'> <;> <OpNo[2]>*

output: *<'2'> <;> <OpNo[2]> <;> <AmntPmnt0[11]> <;> < AmntPmnt 1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;> <NoRA[5]> <;>*

FPR operation: Provides information about the RA by type of payment as well as the total number of operations by specified operator.

Input data :

<'2'>	1 symbol obligatory '2'
OpNo	Symbol(s) from '1' to '20' corresponding to operator's number

Output data:

<'2'>	1 symbol obligatory '2'
OpNo	Symbol(s) from '1' to '20' corresponding to operator's number
AmntPmnt	11 symbols for the RA by type of payment
NoRA	5 symbols for the total number of operations

2.7.14. Command: 6Fh / o – Reading of operator's report – 3 (PO)

input: *<'3'> <;> <OpNo[2]>*

output: *<'3'> <;> <OpNo[2]> <;> <AmntPmnt0[11]> <;> <AmntPmnt1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;> <NoPO[5]> <;>*

FPR operation: Provides information about the PO by type of payment as well as the total number of operations by specified operator

Input data :

<'3'>	1 symbol obligatory '3'
OpNo	Symbol(s) from '1' to '20' corresponding to operator's number

Output data:

<'3'>	1 symbol obligatory '3'
OpNo	Symbol(s) from '1' to '20' corresponding to operator's number
AmntPmnt	11 symbols for the PO by type of payment
NoPO	5 symbols for the total number of operations

2.7.15. Command: 6Fh / o – Reading of operator's report – 4 (received)

input: *<'4'> <;> < OpNo[2]>*

output: *<'4'> <;> <OpNo[2]> <;> < AmntPmnt 0[11]> <;> < AmntPmnt 1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;>*

FPR operation: Provides information about the amounts received from sales by type of payment and specified operator.

Input data :

<'4'>	1 symbol obligatory '4'

OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number

Output data:

<’4’>	1 symbol obligatory ‘4’
OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number
AmntPmnt	11 symbols for amounts received by type of payment

2.7.16. Command: 6Fh / o – Reading of operator’s report – 5 (counters)

input: *<’5’> <;> < OpNo[2]>*
output: *<’5’> <;> < OpNo[2]> <;> <NoRep[5]> <;> <DateTime[16]> <;>*
FPR operation: Provides information about the number of the last operator’s report and its date and time.

Input data :

<’5’>	1 symbol obligatory ‘5’
OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number

Output data:

<’5’’>	1 symbol obligatory ‘5’
OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number
NoRep	5 symbols for number of the last report
DateTime	16 symbols for date and time of the last operator’s report

2.7.17. Command: 6Fh / o – Reading of operator’s report – 6 (returned)

input: *<’6’> <;> < OpNo[2]>*
output: *<’6’> <;> <OpNo[2]> <;> < AmntPmnt 0[11]> <;> < AmntPmnt 1[11]> <;> < AmntPmnt 2[11]> <;> < AmntPmnt 3[11]> <;> < AmntPmnt 4[11]> <;>*
FPR operation: Provides information about the amounts returned as change by different payment types for the specified operator.

Input data :

<6’>	1 symbol obligatory ‘6’
OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number

Output data:

<’6’>	1 symbol obligatory ‘6’
OpNo	Symbol(s) from ‘1‘ to ‘20‘ corresponding to operator’s number
AmntPmnt	11 symbols for amounts received by type of payment

2.7.18. Command: 70h / p – Read invoice number range

input: n. a.
output: *<StartNumber[10]> <;><EndNumber[10]>*
FPR operation: Provides information about the number of the last issued receipt.

Input data : n. a.
Output data :

StartNumber	10 symbols for current number
EndNumber	10 symbols for end number

2.7.19. Command: 71h / q – Reading of last receipt number

input: n. a.
output: *<NoLastIsRec[4]> <;>*
FPR operation: Provides information about the number of the last issued receipt.

Input data : n. a.
Output data :

NoLastIsRec 4 symbols for the number of the last issued receipt by FPR

2.7.20. Command: 72h / r – Reading information about the current receipt

input: n. a.

output: *<ParOpenRec[1]> <;> {<NoSales[3]> <;> <SbtotTaxGrA[11]> <;> <SbtotTaxGrБ[11]> <;> <SbtotTaxGrB[11]> <;> <ParForbVoid[1]> <;> <ParVATinRec[1]> <;> <ParDetRec[1]> <;> <ParInitPmnt[1]> <;> <ParFinPmnt[1]> <;> <ParPowDown[1]> <;> <ParInv[1]> <;><ChangeAmount[11]> <;> <ParChangeType[1]><;><SbtotTaxGrГ[11]> <;><SbtotTaxGrД[11]><;><SbtotTaxGrE[11]><;><SbtotTaxGrЖ[11]><;><SbtotTaxGr3[11]>}*

FPR operation: Provides information about the current status of the receipt. If ParOpenRec has a '0' value (not opened), the subsequent fields will not be sent.

Input data : n. a.

Output data :

ParOpenRec	1 symbol with value '0' or '1' resp. initiated/finalized receipt
NoSales	3 symbols for number of sales
SbtotTaxGr	11 symbols for subtotal by resp. tax group
ParForbVoid	1 symbol with value '0' or '1' resp. allowed / forbidden
ParVATinRec	1 symbol with value '0' or '1' resp. with/without printing
ParDetRec	1 symbol with value '0' or '1' resp. brief/detailed format
ParInitPmnt	1 symbol with value '0' or '1' resp. initiated/not initiated payment
ParFinPmnt	1 symbol with value '0' or '1' resp finalized/not finalized payment
ParInv	1 symbol with value '0' or '1' resp. standard receipt / invoice receipt
ParPowDown	1 symbol with value '0' or '1' resp. no power down/power down
ChangeAmount	11 symbols the amount of the due change in the stated payment type
ParChangeType	1 symbol with value '0', '1' or '2' indicating the change type correspondingly – in cash, in the same payment type or in currency

2.7.21. Command: 73h / s – Reading the last date of a daily report

input: n. a.

output: *<date[10]> <;> <NoLastDRep[4]> <;> <NoLastRAMReset[4]> <;>*

FPR operation: Provides information about the date and number of the last daily report and the last reset of the operative memory (RAM reset).

Input data : n. a.

Output data :

Date	10 symbols for data in DD-MM-YYYY format
NoLastDRep	4 symbols for the number of the last daily report
NoLastRAMReset	4 symbols for the number of the last RAM reset

2.7.22. Command: 74h / t – Reading of free FM blocks

input: n. a.

output: *<FreeFMBl[4]> <;>*

FPR operation: Provides information about the number of the remaining free blocks in the Fiscal Memory.

Input data : n. a.

Output data :

FreeFMBl 4 symbols for the number of free blocks in the FM

2.7.23. Command: 75h / u – Reading of FM contents

input: n. a.
output: ACK +
end number of packed messages for every block stored in FM:

<Nsegm[4]> <CodStor[1]> <DateStor[16]> <Status[1]> <ReadData [~]> +
a message for end of string: <Nsegm[4]><'@'>

FPR operation: Provides consequently information about every single block stored in the FM starting with Acknowledgements and ending with end message.

Input data : n. a.
Output data :

Nsegm	4 symbols for physical FM block number
CodStor	1 symbol stating the type of the stored block with the following values: 0 – factory number of FPR 1 – tax number, decimal point position and tax rate at fiscalization 4 – daily financial report 5 – RAM reset 6 – change of tax rates 7 – change of decimal point position : – deleted EJ
dateStor	16 symbols for the date and time of block storing
Status	1 symbol 0 or 1 resp. for correct/incorrect block checksum
ReadData	Total fields of data read
<'@'>	1 symbol obligatory '@' for end of string

2.8. REPORTS PRINTING COMMANDS

Set of commands for printing of reports generated by FPR.

2.8.1. Command: 76h / v – Report by department

input: <Param>
output: ACK
FPR operation: Prints all special events stored in the FM (see 2.2.4.).

Input data :

Param	1 symbol with value 'X' or 'Z' for – without or with zeroing

Output data : n. a.

2.8.2. Command: 77h / w – Special FM report

input: n. a.
output: ACK
FPR operation: Prints all special events stored in the FM (see 2.2.4.).

Input data : n. a.
Output data : n. a.

2.8.3. Command: 77h / w – Report by payments

input: <Param>
output: ACK
FPR operation: Prints all payments from the FM.

Input data :

Param	1 symbol with value 'P'

Output data : n. a.

2.8.4. Command: 78h / x – Detailed FM report by number of blocks

input: <*StartNoRep[4]*> <;> <*EndNoRep [4]*>

output: ACK
FPR operation: Prints a detailed FM report by initial and end FM block number.

Input data :
StartNoStat 4 symbols for the initial FM block number included in the report
EndNoStat 4 symbols for the final FM block number included in the report
Output data: n. a.

2.8.5. Command: 78h / x – Detailed FM report of payments by number of blocks

input: <*StartNoRep[4]*> <;> <*EndNoRep [4]*> <;><*'P'*>

output: ACK
FPR operation: Prints a detailed FM report by initial and end FM block number.

Input data :
StartNoStat 4 symbols for the initial FM block number included in the report
EndNoStat 4 symbols for the final FM block number included in the report
'P' 1 symbol 'P'
Output data: n. a.

2.8.6. Command: 79h / y – Brief FM report by number of blocks

input: <*StartNoStat[4]*> <;> <*EndNoStat [4]*>
output: ACK
FPR operation: Prints a brief FM report by initial and end FM block number.

Input data :
StartNoStat 4 symbols symbols for the initial FM block number included in the report
EndNoStat 4 symbols for the final FM block number included in the report
Output data: n. a.

2.8.7. Command: 79h / y – Brief FM report of payments by number of blocks

input: <*StartNoStat[4]*> <;> <*EndNoStat [4]*> <;><*'P'*>
output: ACK
FPR operation: Prints a brief FM report by initial and end FM block number.

Input data :
StartNoStat 4 symbols symbols for the initial FM block number included in the report
EndNoStat 4 symbols for the final FM block number included in the report
'P' 1 symbol 'P'
Output data: n. a.

2.8.8. Command: 7Ah / z – Detailed FM report by date

input: <*StartDate[6]*> <;> <*EndDate[6]*>
output: ACK
FPR operation: Prints a detailed FM report by initial and end date.

Input data :

StartDate 6 symbols for initial date in the DDMMYY format
EndDate 6 symbols for final date in the DDMMYY format
Output data: n. a.

2.8.9. Command: 7Ah / z – Detailed FM report of payments by date

input: <*StartDate[6]*> <;> <*EndDate[6]*> <;><*'P'*>
output: ACK
FPR operation: Prints a detailed FM report by initial and end date.

Input data :
StartDate 6 symbols for initial date in the DDMMYY format
EndDate 6 symbols for final date in the DDMMYY format
'P' 1 symbol 'P'
Output data: n. a.

2.8.10. Command: 7Bh / { – Brief FM report by date

input: <*StartDate[6]*> <;> <*EndDate[6]*>
output: ACK
FPR operation: Prints a brief FM report by initial and end date.

Input data :
StartDate 6 symbols for initial date in the DDMMYY format
EndDate 6 symbols for final date in the DDMMYY format
Output data: n. a.

2.8.11. Command: 7Bh / { – Brief FM report of payments by date

input: <*StartDate[6]*> <;> <*EndDate[6]*> <;><*'P'*>
output: ACK
FPR operation: Prints a brief FM report by initial and end date.

Input data :
StartDate 6 symbols for initial date in the DDMMYY format
EndDate 6 symbols for final date in the DDMMYY format
'P' 1 symbol 'P'
Output data: n. a.

2.8.12. Command: 7Ch / | – Daily fiscal report X or Z

Input: <*Type[1]*>
Output: *ACK*
FP Action: Depending on the parameter prints:

- daily fiscal report with zeroing and fiscal memory record, preceded by Electronic Journal report print ('Z');
- daily fiscal report without zeroing (**'X'**);

Input data:
Type 1 character (parameter) with following values: 'X', 'Z'
Output data: None

2.8.13. Command: 7Ch / | – Daily fiscal report EJ

Input: <*Type[1]*>
Output: *ACK*
FP Action: print and zeroing EJ.

Input data:

Type	1 character (parameter) with following values: 'E'

Output data: None

2.8.14. Command: 7Ch / | – Daily fiscal report: Electronic Journal report from date to date

Input: *<'JX'><;><'D'><;><StartDate[6]> <;> <EndDate[6]>*
Output: *ACK*
FP Action: Reading/Printing Electronic Journal Report from Start Date to End Date.

Input data:

'JX'	2 symbols 'J0' for reading Electronic Journal or 'J1' for printing Electronic Journal
'D'	1 symbol 'D'
StartDate	6 symbols for initial date in the DDMMYY format
EndDate	6 symbols for final date in the DDMMYY format

Output data: n. a.

2.8.15. Command: 7Ch / | – Daily fiscal report: Electronic Journal report from number receipt to number receipt

Input: *<'JX'><;><'N'><;><StartNoRec[5]> <;> <EndNoRec[5]>*
Output: *ACK*
FP Action: Reading/Printing Electronic Journal Report from Number receipt to Number receipt.

Input data:

'JX'	2 symbols 'J0' for reading Electronic Journal or 'J1' for printing Electronic Journal
'N'	1 symbol 'N'
StartNoRec	5 symbols for initial receipt number included in report
EndNoRec	5 symbols for final receipt number included in report

Output data: n. a.

2.8.16. Command: 7Ch / | – Daily fiscal report: Electronic Journal report from number Z report to number Z report

Input: *<'JX'><;><'Z'><;><StartNoRec[4]> <;> <EndNoRec[4]>*
Output: *ACK*
FP Action: Reading/Printing Electronic Journal Report from Number report to Number report.

Input data:

'JX'	2 symbols 'J0' for reading Electronic Journal or 'J1' for printing Electronic Journal
'Z'	1 symbol 'Z'
StartNoRec	4 symbols for initial number report
EndNoRec	4 symbols for final number report

Output data: n. a.

2.8.17. Command: 7Ch / | – Daily fiscal report: Electronic Journal report from beginning to end

Input: *<'JX'><;><'*'>*

Output: *ACK*
FP Action: Reading/Printing all Electronic Journal report.

Input data:

'JX'	2 symbols 'J0' for reading Electronic Journal or 'J1' for printing Electronic Journal
'*'	1 symbol '*'

Output data: n. a.

2.8.18. Command: 7Dh / } – Operator's report

input: < *ParZero [1]*> <;> <*NoOper[1]*>
output: ACK
FPR operation: Prints an operator's report for a specified operator (0 = all operators) with or without zeroing ('**Z**' or **'X'**). When a '**Z**' value is specified the report should include all operators.

Input data :

ParZero	1 symbol with value X or Z , resp. without or with zeroing
OpNo	Symbol from '0' to '9' corresponding to operator's number (0 = all operators)

Output data: n. a.

2.8.19. Command: 7Eh / ~ – Article report

input: < *ParZero [1]*>
output: ACK
FPR operation: Prints an article report with or without zeroing ('**Z**' or **'X'**).

Input data :

ParZero	1 symbol with value X or Z , resp. without or with zeroing

Output data: n. a.

2.8.20. Command: 7Fh / ▒ – Extended daily report

input: <*ParZero[1]*>
output: ACK
FPR operation: Prints an extended daily financial report (an article report followed by a daily financial report) with or without zeroing ('**Z**' or **'X'**).

Input data :

ParZero	1 symbol with value X or Z , resp. without or with zeroing

Output data: n. a.

3. SOFTWARE APPLICATION REQUIREMENTS

3.1. RULES FOR USING THE COMMANDS

The commands should be used observing the following rules:

- Do not send a subsequent command prior to receiving a response of the preceding one.
- Observe the sequence of sent and received messages.
- The number of the message in every subsequent command should differ from that in the preceding one.
- Observe the two status bites of the Acknowledgment response.
- When the information received is insufficient request detailed status information – Command 20h.
- Use unpacked messages (see 3.3) to check the standby status of the FPR.

3.2. SAMPLE SALE TRANSACTION OF FPR

The sale transaction controlled by a software application (SA) is a procedure, which consists of several commands, of which obligatory are: initiation of customer fiscal receipt (command 30h), sale registration (command 31h or 32h), payment (command 35h) and finalization of the fiscal receipt (command 38h).

Sample command sequence for issuing a customer fiscal receipt:

- fiscal receipt opening (command 30h) – contains information about the operator's number and password, the type of receipt – detailed/brief, with/without VAT printing (see 3.2.1.);
- sale registration (command 31h) – contains information about the article's name, price and tax group as well as non-compulsory information about the quantity sold and value/percent discount/addition;
- subtotal amount (command 33h) – contains non-compulsory parameters for printing, external display visualization and value/percent discount/addition of the amount accumulated;
- current receipt information (command 72h) – requires a response from the FPR, which contains the current parameters of the receipt, the number of sales, the accumulated amounts by tax groups, information about initiated or finalized payments;
- calculation and payment of VAT on VAT account (command 36h) – performs automatic calculation of VAT in the receipt and its payment on VAT account;
- payment (command 35h) – contains information about the amount due and type of payment, which may cover partially or in full the grand total amount due as well as a parameter for calculation of change due;
- fiscal receipt closure (command 2Fh).

3.3. PROGRAMMERS SUPPORT

For easier programming of ***FP*** we have created a library called **ZFPlib**. The library contains all commands needed for the normal operation of FPR as well as some of the most common service commands. The latest version of the library may be downloaded at the following web address: http://sourceforge.net/projects/zfplib/. **ZFPlib** may be used as a C++ library and/or COM component. The library is an open-source published with Mozilla Public License 1.1 and contains: protocol description, the programming interface (API), as well as some sample programs written in various programming languages. One of these sample programs maintains the overall functionality of the library and may be used for testing and studying of the protocol and the program interface and even for routine programming. **You do not have to study the communication protocol in detail in order to use the library**.

4. AUXILARY GS PROTOCOL (COMMANDS 1Dh)

GS command	Message from the AS	FP Answer
Information	**<1Dh><Info>** **where**: **1Dh** – one byte (hex) for message start **Info** – character 'I'	<Info><print line length> <;> <Articles number> <;> <Departments number> <;> <Operators number> <;> <Tax groups number><;> <header lines><;> <payments number><;> <logo number><;> <'0' - reserved><;><clients number>
Disable communication	**<1Dh><=><0><F><nnnn>** **where:** **1Dh –** one byte (hex) for message start **=** – character '=' **0** – character '0' **F** – character 'F'; **nnnn** – four characters for the device logical number (programmed with command E – parameters)	**ATTENTION!!!** **FP stops answering all commands except the next command (enable)**
Enable communication	**<1Dh><=><1><F><nnnn>** **where:** **1Dh –** one byte (hex) for message start **=** – character '=' **0** – character '0' **F** – character 'F'; **nnnn – four characters for the device logical number (programmed with command E – parameters)**	FP continue working
Set communication baudrate	**<1Dh><s><n>** **where:** **1Dh –** one byte (hex) for message start **s** – character 's' **n** – one character for the baudrate: 0 - 9600 bps 1 - 19200 bps 2 - 34800 bps 3 - 57600 bps 4 - 115200 bps	**ATTENTION!!!** FP continues its normal work with the new baudrate. **Possible must change the communication parameters**