

DATECS WP-500 KL

Programmer's Manual



The information contained in this document is subject to change without notice. No part of this document may be reproduced or transmitted, in any form or by any means, mechanical, electrical or electronic without the prior written permission of Datecs Ltd.

Version: October, 2015



Content

Description of the program interface	4
Low level protocol	4
A) Protocol type - Master (Host) / Slave	4
B) Sequence of the messages	4
C) Non-wrapped messages – time-out	4
D) Wrapped messages	5
Message composition, syntax and meanings	5
Status bits of the cash register	6
Command interface	8
Command description	9
Command: 38 (26h) - Opening a non-fiscal receipt	9
Command: 39 (27h) - Closing a non-fiscal receipt	9
Command: 42 (2Ah) - Printing of a free non-fiscal text	9
Command: 44 (2Ch) - Paper feed	9
Command: 45 (2Dh) - Check for mode connection with PC	9
Command: 48 (30h) - Open fiscal receipt	9
Command: 49 (31h) - Registration of sale	10
Command: 50 (32h) - Return the active VAT rates	10
Command: 51 (33h) - Subtotal	11
Command: 53 (35h) - Payments and calculation of the total sum (TOTAL)	11
Command: 54 (36h) - Printing of a free fiscal text	12
Command: 56 (38h) - Close fiscal receipt	12
Command: 57 (39h) - Print client information	12
Command: 58 (3Ah) - Registering the sale of a programmed item	12
Command: 60 (3Ch) - Cancel fiscal receipt	13
Command: 61 (3Dh) - Set date and time	13
Command: 62 (3Eh) - Read date and time	13
Command: 64 (40h) - Information on the last fiscal entry	13
Command: 65 (41h) - Information on daily taxation	14
Command: 66 (42h) - Set invoice interval	14
Command: 68 (44h) - Number of remaining entries for Z-reports in FM	14
Command: 69 (45h) - Reports	14
Command: 70 (46h) - Cash in and Cash out operations	15
Command: 71 (47h) - Print diagnostic information	15
Command: 72 (48h) - Fiscalization	15
Command: 74 (4Ah) - Reading the Status	15
Command: 76 (4Ch) - Status of the fiscal transaction	15
Command: 80 (50h) - Play sound	16
Command: 83 (53h) - Programming of VAT rates	16
Command: 86 (56h) - Date of the last fiscal record	16
Command: 87 (58h) - Get item groups information	16
Command: 88 (58h) - Get department information	16
Command: 89 (59h) - Test of Fiscal Memory	17
Command: 90 (5Ah) - Diagnostic information	17
Command: 92 (5Ch) - Printing of separating line	18
Command: 94 (5Eh) - Fiscal memory report by date	19
Command: 95 (5Fh) - Fiscal memory report by number of Z-report	19
Command: 98 (62h) - Programming of TAX number	19
Command: 99 (63h) - Reading the programmed TAX number	19
Command: 100 (64h) - Reading an error	19
Command: 101 (65h) - Set operator password	20
Command: 103 (67h) - Information for the current receipt	20
Command: 105 (69h) - Report operators	20
Command: 106 (6Ah) - Drawer opening	20
Command: 107 (6Bh) - Defining and reading items	21
Command: 109 (6Dh) - Print duplicate receipt	23
Command: 110 (6Eh) - Additional daily information	23
Command: 111 (65h) - PLU report	24
Command: 112 (70h) - Information for operator	24
Command: 116 (74h) - Reading FM	25
Command: 124 (7Ch) - Search receipt number by period	25
Command: 125 (7Dh) - Info EJ	25
Command: 140 (8Ch) - Defining and reading clients	27
Command: 255 (FFh) - Programming	28

Description of the program interface

The fiscal device operates under the control of an application program, with which communicates via RS232 (or LAN) serial connection. The device executes a previously set of wrapped commands, arranged according to the type of the operations which have to be executed. The application program does not have a direct access to the resources of the fiscal device although it can detect data connected with the status of the fiscal device and the fiscal control unit.

Low level protocol

A) Protocol type - Master (Host) / Slave

The fiscal printer performs the commands sent by the Host and returns messages, which depend on the result. The fiscal printer cannot instigate asynchronous communications itself. Only responses to commands from the Host are sent to the Host. These messages are either wrapped or single byte control codes. The fiscal printer maintains the communication via the RS232 serial connection at baud rates of 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 b/s, 8N1.

B) Sequence of the messages

Host sends a wrapped message, containing a command for the fiscal printer. ECR executes the requested operation and response with a wrapped message. Host has to wait for a response from the fiscal printer before to send another message. The protocol uses non-wrapped messages with a length one byte for processing of the necessary pauses and error conditions.

C) Non-wrapped messages – time-out

When the transmitting of messages from the Host is normal, Slave answers not later than 60 ms either with a wrapped message or with a 1 byte code. Host must have 500 ms of time-out for receiving a message from Slave. If there is no message during this period of time the Host will transmit the message again with the same sequence number and the same command. After several unsuccessful attempts Host must indicate that there is either no connection to the fiscal printer or there is a hardware fault.

Non-wrapped messages consist of one byte and they are:

A) NAK 15H

This code is sent by Slave when an error in the control sum or the form of the received message is found. When Host receives a NAK it must again send a message with the same sequence number.

B) SYN 16H

This code is sent by Slave upon receiving a command which needs longer processing time. SYN is sent every 60 ms until the wrapped message is not ready for transmitting.



D) Wrapped messages

a) Host to fiscal printer (Send)

<01><LEN><SEQ><CMD><DATA><05><BCC><03>

b) Fiscal printer to Host (Receive)

<01><LEN><SEQ><CMD><DATA><04><STATUS><05><BCC><03>

Where:

<01> Preamble. - 1 byte long. Value: 01H.

<LEN> Number of bytes from <01> preamble (excluded) to <05> (included) plus the fixed offset of 20H.

Length: 4 bytes. Each digit from the two bytes is sent after 30H is added to it. For example the sum 1AE3H is presented as 31H, 3AH, 3EH, 33H.

<SEQ> Sequence number of the frame.

Length : 1 byte. Value: 20H – FFH. The fiscal printer saves the same <SEQ> in the return message. If the ECR gets a message with the same <SEQ> as the last message received it will not perform any operation, but will repeat the last sent message.

<CMD> The code of the command.

Length: 4 byte. The fiscal printer saves the same <CMD> in the return message. If the fiscal printer receives a non-existing code it returns a wrapped message with zero length in the data field and sets the respective status bit. Each digit from the two bytes is sent after 30H is added to it. For example the sum 1AE3H is presented as 31H, 3AH, 3EH, 33H.

<DATA> Data.

Length: 0-213 bytes for Host to fiscal printer, 0-218 bytes for Fiscal printer to Host. Value: 20H – FFH. The format and length of the field for storing data depends on the command. If the command has no data the length of this field is zero. If there is a syntax error the respective status bit is established in the data and a wrapped message is returned with zero field length.

<04> Separator (only for fiscal printer-to-Host messages),

Length: 1 byte. Value: 04H.

<STATUS> The field with the current status of the fiscal device.

Length: 8 bytes. Value: 80H-FFH.

<05> Postamble

Length: 1 byte. Value: 05H.

<BCC> Control sum (0000H-FFFFH),

Length: 4 bytes. Value of each byte: 30H-3FH. The sum includes between <01> preamble (excluded) to <05>. Each digit from the two bytes is sent after 30H is added to it. For example the sum 1AE3H is presented as 31H, 3AH, 3EH, 33H.

<03> Terminator, Length: 1 byte. Value: 03H.

Message composition, syntax and meanings

a) The data field depends on the command.

b) The parameters sent to the fiscal printer may be separated with a [t] and/or may have a fixed length.

c) The separator([t]) between the parameters shows that it is mandatory.

d) Some of the parameters are mandatory and others are optional. Optional parameters can be left empty, but after them must have separator ([t]).

The symbols with ASCII codes under 32 (20H) have special meanings and their use is explained whenever necessary. If such a symbol has to be sent for some reason (for example in an ESCAPE-command to the display) it must be preceded by 16 (10H) with an added offset 40H.

Example: when we write **255,language[t][t][t]** for the data field then in that field there will be **6C 61 6E 67 75 61 67 65 09 09 09** where each hexadecimal digit is an ASCII value.



Status bits of the cash register

The current status of the device is coded in field 8 bytes long which is sent within each message of the cash register. Description of each byte in this field:

Byte 0: General purpose

- ▲0.7 = 1 Always 1.
- ▲0.6 = 0 Always 0.
- ▲0.5 = 1 General error - this is OR of all errors marked with #.
- ▲0.4 = 0 Always 0.
- ▲0.3 = 0 Always 0.
- ▲0.2 = 0 Always 0.
- ▲0.1 = 1# Command code is invalid.
- ▲0.0 = 1# Syntax error.

Byte 1: General purpose

- ▲1.7 = 1 Always 1.
- ▲1.6 = 0 Always 0.
- ▲1.5 = 0 Always 0.
- ▲1.4 = 0 Always 0.
- ▲1.3 = 0 Always 0.
- ▲1.2 = 0 Always 0.
- ▲1.1 = 1# Command is not permitted.
- ▲1.0 = 1# Overflow during command execution.

Byte 2: General purpose

- ▲2.7 = 1 Always 1.
- ▲2.6 = 0 Always 0.
- ▲2.5 = 1 Nonfiscal receipt is open.
- ▲2.4 = 1 EJ nearly full.
- ▲2.3 = 1 Fiscal receipt is open.
- ▲2.2 = 1 EJ is full.
- ▲2.1 = 0 Always 0.
- ▲2.0 = 1# End of paper.

Byte 3: Not used

- ▲3.7 = 1 Always 1.
- ▲3.6 = 0 Always 0.
- ▲3.5 = 0 Always 0.
- ▲3.4 = 0 Always 0.
- ▲3.3 = 0 Always 0.
- ▲3.2 = 0 Always 0.
- ▲3.1 = 0 Always 0.
- ▲3.0 = 0 Always 0.

Byte 4: Fiscal memory

- ▲4.7 = 1 Always 1.
- ▲4.6 = 0 Always 0.
- ▲4.5 = 1 OR of all errors marked with '*' from Bytes 4 и 5.
- ▲4.4 = 1* Fiscal memory is full.
- ▲4.3 = 1 There is space for less then 50 reports in Fiscal memory.
- ▲4.2 = 1 Serial number and number of FM are set.
- ▲4.1 = 1 Tax number is set.
- ▲4.0 = 1* Error while writing in FM.

**Byte 5: Fiscal memory**

- ▲ 5.7 = 1 Always 1.
- ▲ 5.6 = 0 Always 0.
- ▲ 5.5 = 0 Always 0.
- ▲ 5.4 = 1 VAT are set at least once.
- ▲ 5.3 = 1 ECR is fiscalized.
- ▲ 5.2 = 0 Always 0.
- ▲ 5.1 = 1 FM is formatted.
- ▲ 5.0 = 0 Always 0.

Byte 6: Not used

- ▲ 6.7 = 1 Always 1.
- ▲ 6.6 = 0 Always 0.
- ▲ 6.5 = 0 Always 0.
- ▲ 6.4 = 0 Always 0.
- ▲ 6.3 = 0 Always 0.
- ▲ 6.2 = 0 Always 0.
- ▲ 6.1 = 0 Always 0.
- ▲ 6.0 = 0 Always 0.

Byte 7: Not used

- ▲ 7.7 = 1 Always 1.
- ▲ 7.6 = 0 Always 0.
- ▲ 7.5 = 0 Always 0.
- ▲ 7.4 = 0 Always 0.
- ▲ 7.3 = 0 Always 0.
- ▲ 7.2 = 0 Always 0.
- ▲ 7.1 = 0 Always 0.
- ▲ 7.0 = 0 Always 0.



Command interface

This is example command syntax:

```
{Parameter1}<SEP>{Parameter2}<SEP>{Parameter3}<SEP><DateTime><SEP>
```

Note

<SEP> - this tag must be inserted after each parameter to separate different parameters. It's value is '[t]' (tab). It is the same for all commands.

Mandatory parameters:

- **Parameter1** - This parameter is mandatory, it must be filled;
- **Parameter3** - This parameter is mandatory, it must be filled;
 - **A** - Possible value of Parameter3;
Answer(1) - if Parameter3 has value 'A' see Answer(1);
 - **B** - Possible value of Parameter3;
Answer(2) - if Parameter3 has value 'B' see Answer(2);
- **DateTime** - Date and time format: DD-MM-YY hh:mm:ss DST
 - **DD** - Day
 - **MM** - Month
 - **YY** - Year
 - **hh** - Hours
 - **mm** - Minutes
 - **ss** - Seconds
 - **DST** - Text DST. If exist means that summer time is active.

Optional parameters:

- **Parameter2** - This parameter is optional it can be left blank, but separator must exist. Default: X;

Note

If left blank parameter will be used with value, after "Default:" in this case 'X', but in some cases blank parameter may change the meaning of the command, which will be explained for each command;

Answer(X) - This is the default answer of the command.

Under each command there will be list with possible answers.

Answer when command fail to execute is the same for all commands, so it will not be explained after each command.

Answer when command fail to execute:

```
{ErrorCode}<SEP>
```

- **ErrorCode** - Indicates an error code;



Command description

Command: 38 (26h) - Opening a non-fiscal receipt

Parameters of the command: none

Answer: {ErrorCode}<SEP>{SlipNumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **SlipNumber** - Current slip number (1...9999999);
-

Command: 39 (27h) - Closing a non-fiscal receipt

Parameters of the command: none

Answer: {ErrorCode}<SEP>{SlipNumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **SlipNumber** - Current slip number (1...9999999);
-

Command: 42 (2Ah) - Printing of a free non-fiscal text

Parameters of the command: {Text}<SEP>

Optional parameters:

- ⤴ **Text** - text of 0...40 symbols;

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 44 (2Ch) - Paper feed

Parameters of the command: {Lines}<SEP>

Optional parameters:

- ⤴ **Lines** - Number of lines to feed from 1 to 99. Default: 1;

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 45 (2Dh) - Check for mode connection with PC

Parameters of the command: none

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 48 (30h) - Open fiscal receipt

Parameters of the command: {OpCode}<SEP>{OpPwd}<SEP>{TillNmb}<SEP>{Invoice}<SEP>

Mandatory parameters:

- ⤴ **OpCode** - Operator number from 1...30;
- ⤴ **OpPwd** - Operator password, ascii string of digits. Length from 1...8;
- ⤴ **TillNmb** - Number of point of sale from 1...99999;
- ⤴ **Invoice** - If this parameter has value 'I' it opens an invoice receipt. If left blank it opens fiscal receipt;

Answer: {ErrorCode}<SEP>{SlipNumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **SlipNumber** - Current slip number (1...9999999);
-

**Command: 49 (31h) - Registration of sale**

Parameters of the command: {PluName}<SEP>{TaxCd}<SEP>{Price}<SEP>{Quantity}<SEP>
{DiscountType}<SEP>{DiscountValue}<SEP>{Department}<SEP>

Mandatory parameters: {PluName},{TaxCd},{Price}

- ⤴ **PluName** - Name of product, up to 72 characters not empty string;
- ⤴ **TaxCd** - Tax code;
 - ⤴ '1' - vat group A;
 - ⤴ '2' - vat group B;
 - ⤴ '3' - vat group C;
 - ⤴ '4' - vat group D;
 - ⤴ '5' - vat group E;
 - ⤴ '6' - vat group F;
 - ⤴ '7' - vat group G;
 - ⤴ '8' - vat group H;
- ⤴ **Price** - Product price, with sign '-' at void operations. Format: 3 decimals; up to *9999999.999
- ⤴ **Department** - Number of the department 0..99; If '0' - Without department;

Optional parameters: {Quantity},{DiscountType},{DiscountValue}

- ⤴ **Quantity** - Quantity of the product (default: 1.000); Format: 2 decimals; up to *999999.99

!!! Max value of {Price} * {Quantity} is *9999999.99. !!!

- ⤴ **DiscountType** - type of discount.
 - ⤴ '0' or empty - no discount;
 - ⤴ '1' - surcharge by percentage;
 - ⤴ '2' - discount by percentage;
 - ⤴ '3' - surcharge by sum;
 - ⤴ '4' - discount by sum; If {DiscountType} is non zero, {DiscountValue} have to contain value.
The format must be a value with two decimals.
- ⤴ **DiscountValue** - value of discount. a number from 0.00 to 21474836.47 If {DiscountType} is zero or empty, this paramter must be empty.

Answer: {ErrorCode}<SEP>{SlipNumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **SlipNumber** - Current slip number (1...9999999);

Command: 50 (32h) - Return the active VAT rates

Parameters of the command: none

Answer:

{ErrorCode}<SEP>{nZreport}<SEP>{TaxA}<SEP>{TaxB}<SEP>{TaxC}<SEP>{TaxD}<SEP>{TaxE}<SEP>
>{TaxF}<SEP>{TaxG}<SEP>{TaxH}<SEP>{EntDate}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **nZreport** - Number of first Z report;
- ⤴ **TaxX** - Value of Tax group X (0.00...99.99 taxable,100.00=disabled);
- ⤴ **EntDate** - Date of entry (format DD-MM-YY);

**Command: 51 (33h) - Subtotal**

Parameters of the command: {Print}<SEP>{DiscountType}<SEP>{DiscountValue}<SEP>

Optional parameters: {Print},{DiscountType},{DiscountValue}

- ⤴ **Print** - print out;
 - ⤴ '0' - default, no print out;
 - ⤴ '1' - the sum of the subtotal will be printed out;
- ⤴ **DiscountType** - type of discount.
 - ⤴ '0' or empty - no discount;
 - ⤴ '1' - surcharge by percentage;
 - ⤴ '2' - discount by percentage;
 - ⤴ '3' - surcharge by sum;
 - ⤴ '4' - discount by sum; If {DiscountType} is non zero, {DiscountValue} have to contain value. The format must be a value with two decimals.
- ⤴ **DiscountValue** - value of discount.
 - ⤴ a number from 0.00 to 21474836.47 for sum operations;
 - ⤴ a number from 0.00 to 100.99 for percentage operations; If {DiscountType} is zero or empty, this paramter must be empty.

Answer: {ErrorCode}<SEP>{SlipNumber}<SEP>{Subtotal}<SEP>{TaxA}<SEP>{TaxB}<SEP>{TaxC}<SEP>{TaxD}<SEP>{TaxE}<SEP>{TaxF}<SEP>{TaxG}<SEP>{TaxH}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **SlipNumber** - Current slip number (1...99999999);
- ⤴ **Subtotal** - Subtotal of the receipt (0.00...99999999.99 or 0...999999999 depending dec point position);
- ⤴ **TaxX** - Receipts turnover by vat groups (0.00...99999999.99 or 0...999999999 depending dec point position);

Command: 53 (35h) - Payments and calculation of the total sum (TOTAL)

Parameters of the command:

Syntax 1: {PaidMode}<SEP>{Amount}<SEP>

- ⤴ **PaidMode** - Type of payment;
 - ⤴ '0' - cash;
 - ⤴ '1' - credit card;
 - ⤴ '2' - debit card;
 - ⤴ '3' - other pay#3
 - ⤴ '4' - other pay#4
 - ⤴ '5' - other pay#5
- ⤴ **Amount** - Amount to pay (0.00...99999999.99 or 0...999999999 depending dec point position);

Syntax 2: {PaidMode}<SEP>{Amount}<SEP>{Change}<SEP>

- ⤴ **PaidMode** - Type of payment;
 - ⤴ '6' - Foreign currency
- ⤴ **Amount** - Amount to pay (0.00...99999999.99 or 0...999999999 depending dec point position);
- ⤴ **Change** - Type of change. Only if PaidMode = '6';
 - ⤴ '0' - current currency;
 - ⤴ '1' - foreign currency;

Answer: {ErrorCode}<SEP>{Status}<SEP>{Amount}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **Status** - Indicates an error;
 - ⤴ 'D' - The command passed, return when the paid sum is less than the sum of the receipt. The residual sum due for payment is returned to Amount;
 - ⤴ 'R' - The command passed, return when the paid sum is greater than the sum of the receipt. A message "CHANGE" will be printed out and the change will be returned to Amount;
- ⤴ **Amount** - The sum tendered (0.00...99999999.99 or 0...999999999 depending dec point position);

**Command: 54 (36h) - Printing of a free fiscal text****Parameters of the command:** {Text}<SEP>

Optional parameters:

- ⤴ **Text** - text of 0...40 symbols;

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 56 (38h) - Close fiscal receipt**Parameters of the command:** none**Answer:** {ErrorCode}<SEP>{SlipNumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **SlipNumber** - Current slip number (1...9999999);
-

Command: 57 (39h) - Print client information**Parameters of the command:** {TAXN}<SEP>

Mandatory parameters: {TAXN}

- ⤴ **TAXN** - Client's tax number. ascii string of digits 8...13

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 58 (3Ah) - Registering the sale of a programmed item**Parameters of the command:**

{PluCode}<SEP>{Quantity}<SEP>{Price}<SEP>{DiscountType}<SEP>{DiscountValue}<SEP>

Mandatory parameters: {PluCode}

- ⤴ **PluCode:** The code of the item. from 1...100000

Optional parameters: {Quantity}, {DiscountType}, {DiscountValue}

- ⤴ **Quantity:** Example: 1.000; Max: 999999.999;
- ⤴ **DiscountType** - type of discount.
 - ⤴ '0' or empty - no discount;
 - ⤴ '1' - surcharge by percentage;
 - ⤴ '2' - discount by percentage;
 - ⤴ '3' - surcharge by sum;
 - ⤴ '4' - discount by sum;
- ⤴ **DiscountValue** - value of discount. a number from 0.00 to 21474836.47 If {DiscountType} is zero or empty, this paramter must be empty.

Answer: {ErrorCode}<SEP>{SlipNumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **SlipNumber** - Current slip number (1...9999999);
-

**Command: 60 (3Ch) - Cancel fiscal receipt**

Parameters of the command: none

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 61 (3Dh) - Set date and time

Parameters of the command: {DateTime}<SEP>

Mandatory parameters:

- ⤴ **DateTime** - Date and time in format: "DD-MM-YY hh:mm:ss DST";
 - ⤴ **DD** - Day;
 - ⤴ **MM** - Month;
 - ⤴ **YY** - Year;
 - ⤴ **hh** - Hour;
 - ⤴ **mm** - Minute;
 - ⤴ **ss** - Second;
 - ⤴ **DST** - Text "DST" if exist time is Summer time;

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 62 (3Eh) - Read date and time

Parameters of the command: none

Answer: {ErrorCode}<SEP>{DateTime}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **DateTime** - Date and time in format: "DD-MM-YY hh:mm:ss DST";
 - ⤴ **DD** - Day;
 - ⤴ **MM** - Month;
 - ⤴ **YY** - Year;
 - ⤴ **hh** - Hour;
 - ⤴ **mm** - Minute;
 - ⤴ **ss** - Second;
 - ⤴ **DST** - Text "DST" if exist time is Summer time;
-

Command: 64 (40h) - Information on the last fiscal entry

Parameters of the command: {Type}<SEP>

- ⤴ **Type** - Type of returned data. Default: 0;
 - ⤴ 0 - Turnover on TAX group;
 - ⤴ 1 - Amount on TAX group;

Answer: {ErrorCode}<SEP>{nRep}<SEP>{SumA}<SEP>{SumB}<SEP>{SumC}<SEP>{SumD}<SEP>
{SumE}<SEP>{SumF}<SEP>{SumG}<SEP>{SumH}<SEP>{Date}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **nRep** - Number of report 1...1825;
- ⤴ **SumX** - Depend on **Type**. X is the letter of TAX group (0.00...9999999.99 or 0...999999999 depending dec point position);
- ⤴ **Date** - Date of fiscal record in format DD-MM-YY;

**Command: 65 (41h) - Information on daily taxation****Parameters of the command:** {Type}<SEP>

- ⤴ **Type** - Type of returned data. Default: 0;
 - ⤴ 0 - Turnover on TAX group;
 - ⤴ 1 - Amount on TAX group;

Answer: {ErrorCode}<SEP>{nRep}<SEP>{SumA}<SEP>{SumB}<SEP>{SumC}<SEP>{SumD}<SEP>{SumE}<SEP>{SumF}<SEP>{SumG}<SEP>{SumH}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **nRep** - Number of report (1...1825);
- ⤴ **SumX** - Depend on **Type**. X is the letter of TAX group (0.00...9999999.99 or 0...999999999 depending dec point position);

Command: 66 (42h) - Set invoice interval**Parameters of the command:** {End}<SEP>

- ⤴ If the current invoice counter didn't reached the end of the interval.

{Start}<SEP>{End}<SEP>

- ⤴ If the current invoice counter have reached the end of the interval.
- ⤴ **Start** - The starting number of the interval. Max 10 digits (1...9999999999).
- ⤴ **End** - The ending number of the interval. Max 10 digits (1...9999999999).

Answer: {ErrorCode}<SEP>{Start}<SEP>{End}<SEP>{Current}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **Start** - The current starting value of the interval (1...9999999999)
- ⤴ **End** - The current ending value of the interval (1...9999999999)
- ⤴ **Current** - The current invoice receipt number (1...9999999999)

Command: 68 (44h) - Number of remaining entries for Z-reports in FM**Parameters of the command:** none**Answer:** {ErrorCode}<SEP>{ReportsLeft}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **ReportsLeft** - The number of remaining entries for Z-reports in FM (1...1825).

Command: 69 (45h) - Reports**Parameters of the command:** {ReportType}<SEP>

Mandatory parameters:

- ⤴ **ReportType** - Report type;
 - ⤴ 'X' - X report;
 - ⤴ 'Z' - Z report;

Answer: {ErrorCode}<SEP>{nRep}<SEP>{TotA}<SEP>{TotB}<SEP>{TotC}<SEP>{TotD}<SEP>{TotE}<SEP>{TotF}<SEP>{TotG}<SEP>{TotH}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **nRep** - Number of Z-report (1...1825);
- ⤴ **TotX** - Total sum accumulated by TAX group X - fiscal operations (0.00...9999999.99 or 0...999999999 depending dec point position);



Command: 70 (46h) - Cash in and Cash out operations

Parameters of the command: {Type}<SEP>{Amount}<SEP>

Mandatory parameters:

- ⤴ **Type** - type of operation;
 - ⤴ '0' - cash in;
 - ⤴ '1' - cash out;
- ⤴ **Amount** - the sum (0.00...9999999.99 or 0...999999999 depending dec point position);

Answer: {ErrorCode}<SEP>{CashSum}<SEP>{CashIn}<SEP>{CashOut}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **CashSum** - cash in safe sum (0.00...9999999.99 or 0...999999999 depending dec point position);
 - ⤴ **CashIn** - total sum of cash in operations (0.00...9999999.99 or 0...999999999 depending dec point position);
 - ⤴ **CashOut** - total sum of cash out operations (0.00...9999999.99 or 0...999999999 depending dec point position);
-

Command: 71 (47h) - Print diagnostic information

Parameters of the command: {InfoType}<SEP>

Optional parameters: {InfoType}

- ⤴ **InfoType** - type of the information printed;
 - ⤴ '0' - general information about the device;

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 72 (48h) - Fiscalization

Parameters of the command: {SerialNumber}<SEP>{TINnumber}<SEP>

Mandatory parameters:

- ⤴ **SerialNumber** - Serial Number (Two letters and six digits: XX123456);
- ⤴ **TAXnumber** - TAX number (max 13 characters, of which a min 9 digits);

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 74 (4Ah) - Reading the Status

Parameters of the command: none

Answer: {ErrorCode}<SEP>{StatusBytes}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **StatusBytes** - Status Bytes (See the description of the status bytes).
-

Command: 76 (4Ch) - Status of the fiscal transaction

Parameters of the command: none

Answer:

{ErrorCode}<SEP>{IsOpen}<SEP>{Number}<SEP>{Items}<SEP>{Amount}<SEP>{Paid}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **IsOpen** - 1 - Receipt is open, 0 - receipt is closed;
 - ⤴ **Number** - The number of the current or the last receipt (1...9999999);
 - ⤴ **Items** - number of sales registered on the current or the last fiscal receipt (0...9999999);
 - ⤴ **Amount** - The sum from the current or the last fiscal receipt (0.00...9999999.99 or 0...999999999 depending dec point position);
 - ⤴ **Paid** - The sum paid for the current or the last receipt (0.00...9999999.99 or 0...999999999 depending dec point position);
-

**Command: 80 (50h) - Play sound****Parameters of the command:** {Hz}<SEP>{mSec}<SEP>

Mandatory parameters:

- ⤴ **Hz** - Frequency (0...65535);
- ⤴ **mSec** - Time in milliseconds (0...65535);

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 83 (53h) - Programming of VAT rates**Parameters of the command:** {TaxA}<SEP>{TaxB}<SEP>{TaxC}<SEP>{TaxD}<SEP>{TaxE}<SEP>
{TaxF}<SEP>{TaxG}<SEP>{TaxH}<SEP>{decimal_point}<SEP>

Mandatory parameters:

- ⤴ **TaxX** - Value of VAT rate X;
 - ⤴ **0.00...99.99** - enabled;
 - ⤴ **100.00** - disabled;
- ⤴ **decimal_point** - value: 0 or 2(if decimal_point = 0 - work with integer prices. If decimal_point = 2 - work with fract prices);

Answer: {ErrorCode}<SEP>{RemainingChanges}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **RemainingChanges** - number of remaining changes (1...30);
-

Command: 86 (56h) -Date of the last fiscal record**Parameters of the command:** none**Answer:** {ErrorCode}<SEP>{DateTime}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **DateTime** - The date and the time of the last fiscal record in format: DD-MM-YYYY hh:mm:ss;
-

Command: 87 (58h) -Get item groups information**Parameters of the command:** {ItemGroup}<SEP>

Mandatory parameters:

- ⤴ **ItemGroup** - Number of item group;
 - ⤴ **Answer:** {ErrorCode}<SEP>{TotSales}<SEP>{TotSum}<SEP>{Name}<SEP>
 - ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **TotSales** - Number of sales for this item group for day;
 - ⤴ **TotSum** - Accumulated sum for this item group for day;
 - ⤴ **Name** - Name of item group;
-

Command: 88 (58h) - Get department information**Parameters of the command:** {Department}<SEP>

Mandatory parameters:

- ⤴ **Department** - Number of department;
 - ⤴ **Answer:**
{ErrorCode}<SEP>{TaxGr}<SEP>{Price}<SEP>{TotSales}<SEP>{TotSum}<SEP>{Name}<SEP>
 - ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-



- ⤴ **TaxGr** - Tax group of department;
- ⤴ **Price** - Price of department;
- ⤴ **TotSales** - Number of sales for this department for day;
- ⤴ **TotSum** - Accumulated sum for this department for day;
- ⤴ **Name** - Name of department;

Command: 89 (59h) - Test of Fiscal Memory

Parameters of the command: {Write}<SEP>

Optional parameters:

- ⤴ **Write** - Write test. Default: 0;
 - ⤴ **0** - Read test.
 - ⤴ **1** - Write and read test;

Answer: {ErrorCode}<SEP>{Records}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **Records** - Number of records left (0...16).

Command: 90 (5Ah) - Diagnostic information

This is example command syntax:

Syntax 1: {Param}<SEP>

Optional parameters:

- ⤴ **none** - Diagnostic information without firmware checksum;
Answer(1)
- ⤴ **'1'** - Diagnostic information with firmware checksum;
Answer(1)
- ⤴ **'#'** - Device identification
Answer(2)

Syntax 2: {Param}

Optional parameters:

- ⤴ **none** - Diagnostic information without firmware checksum;
Answer(3)
- ⤴ **'1'** - Diagnostic information with firmware checksum;
Answer(3)
- ⤴ **'#'** - Device identification
Answer(4)

Answer(1): {ErrorCode}<SEP>{Name}<SEP>{FwRev}<SEP>{FwDate}<SEP>{FwTime}<SEP>
{Checksum}<SEP>{Sw}<SEP>{SerialNumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **Name** - Device name (up to 32 symbols).
- ⤴ **FwRev** - Firmware version. 6 symbols;
- ⤴ **FwDate** - Firmware date DDMMYY. 7 symbols;
- ⤴ **FwTime** - Firmware time hhmm. 4 symbols.
- ⤴ **Checksum** - Firmware checksum. 4 symbols;
- ⤴ **Sw** - Switch from Sw1 to Sw8. 8 symbols (not used at this device, always 00000000);
- ⤴ **SerialNumber** - Serial Number (Two letters and six digits: XX123456);

Answer(2): {ErrorStatus}<SEP>{DevType}<SEP>{Country}<SEP>{DevName}<SEP>
{CountryCode}<SEP>{Major}<SEP>{Minor}<SEP>{Build}<SEP>{DevNumber}<SEP>
{HardwareVer}<SEP>{BootloaderVer}<SEP>



- ⤴ **ErrorStatus** - Indicates an error;
 - ⤴ **'P'** - The command passed;
 - ⤴ **'F'** - The command failed;
- ⤴ **DevType** - Type of fiscal device = 1
- ⤴ **Country** - Country code = 26
- ⤴ **DevName** - Device name
- ⤴ **CountryCode** - Code of dealer = 01
- ⤴ **Major** - Official major version number= XX
- ⤴ **Minor** - Official build number = XX
- ⤴ **Build** - Internal build number = XX
- ⤴ **DevNumber** - Device Serial Number
- ⤴ **HardwareVer** - Hardware Version
- ⤴ **BootloaderVer** - Bootloader Version

Answer(3): {Name},{FwRev}{Sp}{FwDate}{Sp}{FwTime},{Checksum},{Sw},{DevNumber}

- ⤴ **Name** - Device name.
- ⤴ **FwRev** - Firmware version. 6 symbols;
- ⤴ **Sp** - Space. 1 symbol;
- ⤴ **FwDate** - Firmware date DDMMYY. 7 symbols;
- ⤴ **FwTime** - Firmware time hhmm. 4 symbols.
- ⤴ **Checksum** - Firmware checksum. 4 symbols;
- ⤴ **Sw** - Switch from Sw1 to Sw8. 8 symbols;
- ⤴ **DevNumber** - Device Serial Number

Answer(4): {DevType},{Country},{DevName},{CountryCode},{Major},{Minor},{Build},{DevNumber},{HardwareVer},{BootloaderVer}

- ⤴ **DevType** - Type of fiscal device = 1
- ⤴ **Country** - Country code = 26
- ⤴ **DevName** - Device name
- ⤴ **CountryCode** - Code of dealer = 01
- ⤴ **Major** - Official major version number= XX
- ⤴ **Minor** - Official build number = XX
- ⤴ **Build** - Internal build number = XX
- ⤴ **DevNumber** - Device Serial Number
- ⤴ **HardwareVer** - Hardware Version
- ⤴ **BootloaderVer** - Bootloader Version

Command: 92 (5Ch) - Printing of separating line

Parameters of the command: {Type}<SEP>

Mandatory parameters:

- ⤴ **Type** - Type of the separating line.
 - ⤴ **'1'** - Separating line with the symbol '-';
 - ⤴ **'2'** - Separating line with the symbols '-' and ' ';
 - ⤴ **'3'** - Separating line with the symbol '=';

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

**Command: 94 (5Eh) - Fiscal memory report by date****Parameters of the command:** {Type}<SEP>{Start}<SEP>{End}<SEP>

Mandatory parameters:

⤴ **Type** - 0 - short; 1 - detailed;

Optional parameters:

⤴ **Start** - Start date. Default: Date of fiscalization (format DD-MM-YY);⤴ **End** - End date. Default: Current date (format DD-MM-YY);**Answer:** {ErrorCode}<SEP>⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Command: 95 (5Fh) - Fiscal memory report by number of Z-report**Parameters of the command:** {Type}<SEP>{First}<SEP>{Last}<SEP>

Mandatory parameters:

⤴ **Type** - 0 - short; 1 - detailed;

Optional parameters:

⤴ **First** - First block in the report. Default: 1;⤴ **Last** - Last block in the report. Default: number of last Z report;**Answer:** {ErrorCode}<SEP>⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Command: 98 (62h) - Programming of TAX number**Parameters of the command:** {TAXnumber}<SEP>

Mandatory parameters:

⤴ **TAXnumber** - TAX number (max 13 characters, of which a min 9 digits);**Answer:** {ErrorCode}<SEP>⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Command: 99 (63h) - Reading the programmed TAX number**Parameters of the command:** none**Answer:** {ErrorCode}<SEP>{TAXnumber}<SEP>⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;⤴ **TAXnumber** - TAX number (max 13 characters, of which a min 9 digits);

Command: 100 (64h) - Reading an error**Parameters of the command:** {Code}<SEP>

Mandatory parameters:

⤴ **Code** - Code of the error;**Answer:** {ErrorCode}<SEP>{Code}<SEP>{ErrorMessage}<SEP>⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;⤴ **Code** - Code of the error, to be explained;⤴ **ErrorMessage** - Explanation of the error in **Code**;

**Command: 101 (65h) - Set operator password****Parameters of the command:** {OpCode}<SEP>{OldPwd}<SEP>{NewPwd}<SEP>

Mandatory parameters:

- ⤴ **OpCode** - Operator number from 1...30;
- ⤴ **NewPwd** - Operator password, ascii string of digits. Length from 1...8;

Optional parameters:

- ⤴ **OldPwd** - Operator old password or administrator (oper29 & oper30) password. Can be blank if service jumper is on.

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 103 (67h) - Information for the current receipt**Parameters of the command:** none**Answer:**

{ErrorCode}<SEP>{SumVATA}<SEP>{SumVATB}<SEP>{SumVATC}<SEP>{SumVATD}<SEP>{SumVATE}<SEP>{SumVATF}<SEP>{SumVATG}<SEP>{SumVATH}<SEP>{Inv}<SEP>{InvNum}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
 - ⤴ **SumVATx** - The current accumulated sum on VATx (0.00...99999999.99 or 0...999999999 depending dec point position);
 - ⤴ **Inv** - '1' if it is expanded receipt; '0' if it is simplified receipt;
 - ⤴ **InvNmb** - Number of the next invoice (up to 10 digits)
-

Command: 105 (69h) - Report operators**Parameters of the command:** {FirstOper}<SEP>{LastOper}<SEP>{Clear}<SEP>

Optional parameters:

- ⤴ **FirstOper** - First operator. Default: 1 (1...30);
- ⤴ **LastOper** - Last operator. Default: Maximum operator number (1...30);
- ⤴ **Clear** - Clear registers for operators. Default: 0;
 - ⤴ **'0'** - Does not clear registers for operators.
 - ⤴ **'1'** - Clear registers for operators.

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
-

Command: 106 (6Ah) - Drawer opening**Parameters of the command:** {mSec}<SEP>

Optional parameters:

- ⤴ **mSec** - The length of the impulse in milliseconds. (0...65535)

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

**Command: 107 (6Bh) - Defining and reading items****Parameters of the command:** {Option}<SEP>{Parameters}<SEP>

Mandatory parameters: {Option}

- ⤴ **'T'** - Items information;

Syntax: {Option}<SEP>

Answer(3)

- ⤴ **'P'** - Item programming;

Syntax:

{Option}<SEP>{PLU}<SEP>{TaxGr}<SEP>{Dep}<SEP>{Group}<SEP>{PriceType}<SEP>{Price}<SEP>{AddQty}<SEP>{Quantity}<SEP>{Bar1}<SEP>{Bar2}<SEP>{Bar3}<SEP>{Bar4}<SEP>{Name}<SEP>

Mandatory parameters:

- ⤴ **PLU** - Item number (1...100000);

- ⤴ **TaxGr** - VAT group (letter 'A'...'H' or cyrillic 'A'...'З');

- ⤴ **Dep** - Department (0...99);

- ⤴ **Group** - Stock group (1...99);

- ⤴ **PriceType** - Price type ('0' - fixed price, '1' - free price, '2' - max price; for details see user's manual);

- ⤴ **Price** - Price (0.00...99999999.99 or 0...999999999 depending dec point position);

- ⤴ **Quantity** - Stock quantity (0.001...99999.999);

- ⤴ **Name** - Item name (up to 32 symbols);

Optional parameters:

- ⤴ **AddQty** - A byte with value 'A',

- ⤴ **BarX** - Barcode X (up to 13 digits);

Answer(1)

- ⤴ **'A'** - Change of the available quantity for item;

Syntax: {Option}<SEP>{PLU}<SEP>{Quantity}<SEP>

Mandatory parameters:

- ⤴ **PLU** - Item number (1...100000);

- ⤴ **Quantity** - Stock quantity (0.001...99999.999);

Answer(1)

- ⤴ **'D'** - Item deleting;

Syntax: {Option}<SEP>{firstPLU}<SEP>{lastPLU}<SEP>

Mandatory parameters:

- ⤴ **firstPLU** - First item to delete (1...100000); If this parameter has value 'A', all items will be deleted(lastPLU must be empty).

Optional parameters:

- ⤴ **lastPLU** - last item to delete (1...100000). Default: {firstPLU}; ;

Answer(1)

- ⤴ **'R'** - Reading item data;

Syntax: {Option}<SEP>{PLU}<SEP>

Mandatory parameters:

- ⤴ **PLU** - Item number (1...100000);

Answer(2)

- ⤴ **'F'** - Returns data about the first found programmed item;

Syntax: {Option}<SEP>{PLU}<SEP>

Optional parameters:

- ⤴ **PLU** - Item number (1...100000). Default: 0;

Answer(2)

- ⤴ **'L'** - Returns data about the last found programmed item;

Syntax: {Option}<SEP>{PLU}<SEP>

Optional parameters:

- ⤴ **PLU** - Item number (1...100000). Default: 100000;



Answer(2)

- ⤴ **'N'** - Returns data for the next found programmed item;

Syntax: {Option}<SEP>

Note

The same command with option 'F' or 'L' must be executed first. This determines whether to get next('F') or previous ('L') item.

Answer(2)

- ⤴ **'f'** - Returns data about the first found item with sales on it;

Syntax: {Option}<SEP>{PLU}<SEP>

Optional parameters:

- ⤴ **PLU** - Item number(1...100000). Default: 0;

Answer(2)

- ⤴ **'l'** - Returns data about the last found item with sales on it;

Syntax: {Option}<SEP>{PLU}<SEP>

Optional parameters:

- ⤴ **PLU** - Item number (1...100000). Default: 100000;

Answer(2)

- ⤴ **'n'** - Returns data for the next found item with sales on it;

Syntax: {Option}<SEP>

Note

The same command with option 'f' or 'l' must be executed first. This determines whether to get next('f') or previous ('l') item. *Answer(2)*

- ⤴ **'X'** - Find the first not programmed item;

Syntax: {Option}<SEP>{PLU}<SEP>

Optional parameters:

- ⤴ **PLU** - Item number(1...100000). Default: 0;

Answer(4)

- ⤴ **'x'** - Find the last not programmed item;

Syntax: {Option}<SEP>{PLU}<SEP>

Optional parameters:

- ⤴ **PLU** - Item number (1...100000). Default: 100000;

Answer(4)

Answer(1): {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Answer(2):

{ErrorCode}<SEP>{PLU}<SEP>{TaxGr}<SEP>{Dep}<SEP>{Group}<SEP>{PriceType}<SEP>{Price}<SEP>{Turnover}<SEP>{SoldQty}<SEP>{StockQty}<SEP>{Bar1}<SEP>{Bar2}<SEP>{Bar3}<SEP>{Bar4}<SEP>{Name}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **PLU** - Item number (1...100000);
- ⤴ **TaxGr** - VAT group (letter 'A'...'H' or cyrillic 'A'...'З');
- ⤴ **Dep** - Department (0...99);
- ⤴ **Group** - Stock group (1...99);
- ⤴ **PriceType** - Price type ('0' - fixed price, '1' - free price, '2' - max price; for details see user's manual) ;
- ⤴ **Price** - Price (0.00...9999999.99 or 0...999999999 depending dec point position) ;
- ⤴ **Turnover** - Accumulated amount of the item (0.00...9999999.99 or 0...999999999 depending dec point position) ;
- ⤴ **SoldQty** - Sold out quantity (0.001...99999.999);
- ⤴ **StockQty** - Current quantity (0.001...99999.999);
- ⤴ **BarX** - Barcode X (up to 13 digits) ;
- ⤴ **Name** - Item name (up to 32 symbols);



Answer(3): {ErrorCode}<SEP>{Total}<SEP>{Prog}<SEP>{NameLen}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **Total** - Total count of the programmable items (100000);
- ⤴ **Prog** - Total count of the programmed items (0...100000);
- ⤴ **NameLen** - Maximum length of item name (32);

Answer(4): {ErrorCode}<SEP>{PLU}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **PLU** - Item number (1...100000);

Command: 109 (6Dh) - Print duplicate receipt

Parameters of the command: none

Answer: {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Command: 110 (6Eh) - Additional daily information

Parameters of the command: {Type}<SEP>

Optional parameters:

- ⤴ **Type** - Type of information. Default: 0;
 - ⤴ **'0'** - Payments;
Answer(1)
 - ⤴ **'1'** - Negative payments;
Answer(2)
 - ⤴ **'2'** - Number and sum of sells;
Answer(3)
 - ⤴ **'3'** - Number and sum of discounts and surcharges;
Answer(4)
 - ⤴ **'4'** - Number and sum of correction and annulled receipts;
Answer(5)
 - ⤴ **'5'** - Number and sum of cash in and cash out operations;
Answer(6)

Answer 1: {ErrorCode}<SEP>{Pay1}<SEP>{Pay2}<SEP>{Pay3}<SEP>{Pay4}<SEP>
{Pay5}<SEP>{Pay6}<SEP>{ForeignPay}<SEP>

- **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- **PayX** - Value paid by payment X (0.00...9999999.99 or 0...999999999 depending dec point position);.
- **ForeignPay** - Value paid by foreign currency (0.00...9999999.99 or 0...999999999 depending dec point position);.

Answer 2: {ErrorCode}<SEP>{Pay1}<SEP>{Pay2}<SEP>{Pay3}<SEP>{Pay4}<SEP>
{Pay5}<SEP>{Pay6}<SEP>{ForeignPay}<SEP>

- **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- **PayX** - Value paid by payment X for return (0.00...9999999.99 or 0...999999999 depending dec point position);

Answer 3: {ErrorCode}<SEP>{Num}<SEP>{Sum}<SEP>

- **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- **Num** - number of clients;
- **Sum** - sum of the sells.



Answer 4: {ErrorCode}<SEP>{qSur}<SEP>{sSur}<SEP>{qDis}<SEP>{sDis}<SEP>

- **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- **qSur** - number of surcharges;.
- **sSur** - sum of surcharges;.
- **qDis** - number of discounts;.
- **sDis** - sum of discounts;.

Answer 5: {ErrorCode}<SEP>{qVoid}<SEP>{sVoid}<SEP>{qAnul}<SEP>{sAnul}<SEP>

- **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- **qVoid** - number of corrections;.
- **sVoid** - sum of corrections;.
- **qAnul** - number of annulled;.
- **sAnul** - sum of annulled;.

Answer 6: {ErrorCode}<SEP>{qCashIn1}<SEP>{sCashIn1}<SEP>{qCashOut1}<SEP>

{sCashOut1}<SEP>{qCashIn2}<SEP>{sCashIn2}<SEP>{qCashOut2}<SEP>{sCashOut2}<SEP>

- **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- **qCashIn1** - number of cash in operations;.
- **sCashIn1** - sum of cash in operations;.
- **qCashOut1** - number of cash out operations;.
- **sCashOut1** - sum of cash out operations;.
- **qCashIn2** - number of cash in operations in alternate currency;.
- **sCashIn2** - sum of cash in operations in alternate currency;.
- **qCashOut2** - number of cash out operations in alternate currency;.
- **sCashOut2** - sum of cash out operations in alternate currency;.

Command: 111 (65h) - PLU report

Parameters of the command: {Type}<SEP>{FirstPLU}<SEP>{LastPLU}<SEP>

Mandatory parameters:

- ▲ **Type** - Type of report;
 - ▲ '0' - PLU turnovers;
 - ▲ '1' - PLU turnovers with clearing;
 - ▲ '2' - PLU parameters;
 - ▲ '3' - PLU stock;

Optional parameters:

- ▲ **FirstPLU** - First PLU in the report (1...100000). Default: 1;
- ▲ **LastPLU** - Last PLU in the report (1...100000). Default: Maximum PLU in the ECR;

Answer: {ErrorCode}<SEP>

- ▲ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Command: 112 (70h) - Information for operator

Parameters of the command: {Operator}<SEP>

Mandatory parameters:

- ▲ **Operator** - Number of operator (1...30);

Answer: {ErrorCode}<SEP>{Receipts}<SEP>{Total}<SEP>{nDiscount}<SEP>

{Discount}<SEP>{nSurcharge}<SEP>{Surcharge}<SEP>{nVoid}<SEP>{Void}<SEP>

- ▲ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;



- ⤴ **Receipts** - Number of fiscal receipts, issued by the operator (0...65535);
- ⤴ **Total** - Total accumulated sum (0.00...9999999.99 or 0...999999999 depending dec point position);
- ⤴ **nDiscount** - Number of discounts (0...65535);;
- ⤴ **Discount** - Total accumulated sum of discounts with sign (0.00...9999999.99 or 0...999999999 depending dec point position);
- ⤴ **nSurcharge** - Number of surcharges (0...65535);
- ⤴ **Surcharge** - Total accumulated sum of surcharges with sign(0.00...9999999.99 or 0...999999999 depending dec point position);
- ⤴ **nVoid** - Number of corrections (0...65535);
- ⤴ **Void** - Total accumulated sum of corrections with sign(0.00...9999999.99 or 0...999999999 depending dec point position);

Command: 116 (74h) - Reading FM.

Parameters of the command: {Operation}<SEP>{Address}<SEP>{nBytes}<SEP>

Mandatory parameters:

- ⤴ **Operation** - type of operation = '0';
- ⤴ **Address** - Start address 0...FFFFFF (format ascii-hex).
- ⤴ **nBytes** - Number of bytes (1...106)

Answer: {ErrorCode}<SEP>{Data}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **Data** - Data read. Number of bytes is equal to **nBytes** requested, multiplied by 2;

Command: 124 (7Ch) - Search receipt number by period

Parameters of the command: {StartDate}<SEP>{EndDate}<SEP>Type<SEP>

Optional parameters:

- ⤴ **StartDate** - Start date for searching. Default: Date of first document;
- ⤴ **EndDate** - End date for searching. Default: Date of last document;
- ⤴ **Type** - Receipt type;
 - ⤴ '0' - all types;
 - ⤴ '1' - fiscal receipts;
 - ⤴ '2' - daily z reports;
 - ⤴ '3' - invoice receipts;
 - ⤴ '4' - nonfiscal receipts;

Answer: {ErrorCode}<SEP>{StartDate}<SEP>{EndDate}<SEP>{FirstDoc}<SEP>{LastDoc}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **StartDate** - Start date for searching, see DateTime format described at the beginning of the document;
- ⤴ **EndDate** - End date for searching, see DateTime format described at the beginning of the document;
- ⤴ **FirstDoc** - First document in the period, depending "Type" (1...9999999);
- ⤴ **LastDoc** - Last document in the period, depending "Type" (1...9999999);

Command: 125 (7Dh) - Info EJ

Parameters of the command: {Option}<SEP>{DocNum}<SEP>{RecType}<SEP>

Mandatory parameters:

- ⤴ **Option** :
 - ⤴ '0' - Set document to read;
Answer(1)
 - ⤴ '1' - Read one line as text. Must be called multiple time to read whole document;
Answer(2)
 - ⤴ '2' - Read as data. Must be called multiple time to read whole document;



Answer(3)

- ⤴ **'3'** - Print document;

Answer(4)

Optional parameters:

- ⤴ **DocNum** - Number of document (1...9999999). Needed for **Option** = 0.
- ⤴ **RecType** - Document type. Needed for **Option** = 0.
 - ⤴ **'0'** - all types;
 - ⤴ **'1'** - fiscal receipts;
 - ⤴ **'2'** - daily z reports;
 - ⤴ **'3'** - invoice receipts;
 - ⤴ **'4'** - nonfiscal receipts;

Answer(1):

{ErrorCode}<SEP>{DocNumber}<SEP>{RecNumber}<SEP>{Date}<SEP>{Type}<SEP>{Znumber}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **DocNumber** - Number of document (global 1...9999999) ;
- ⤴ **RecNumber** - Number of document (depending "Type");
- ⤴ **Date** - Date of document, see DateTime format described at the beginning of the document;
- ⤴ **Type** - Type of document;
 - ⤴ **'0'** - all types;
 - ⤴ **'1'** - fiscal receipts;
 - ⤴ **'2'** - invoice receipts;
 - ⤴ **'3'** - daily z reportss;
 - ⤴ **'4'** - non fiscal receipts;
 - ⤴ **'5'** - payout receipts;
- ⤴ **Znumber**- number of Z report (1...1825);

Answer(2): {ErrorCode}<SEP>{TextData}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **TextData** - Document text (up to 42 chars);

Answer(3): {ErrorCode}<SEP>{Data}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;
- ⤴ **Data** - Document data, structured information in base64 format. Detailed information in other document;

Answer(4): {ErrorCode}<SEP>

- ⤴ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

**Command: 140 (8Ch) - Defining and reading clients****Parameters of the command:** {Option}<SEP>{Parameters}<SEP>

Mandatory parameters: {Option}

- ▲ **'T'** - Clients information;

Syntax: {Option}<SEP>

Answer(3)

- ▲ **'P'** - Clients programming;

Syntax:

{Option}<SEP>{FIRM}<SEP>{Name}<SEP>{TypeTAXN}<SEP>{TAXN}<SEP>{RecName}<SEP>{VATN}<SEP>{Addr1}<SEP>{Addr2}<SEP>

Mandatory parameters:

- ▲ **FIRM** - Client number, index of record (1...1000);
- ▲ **Name** - Client's name (up to 36 chars);
- ▲ **TAXN** - Client's tax number (8...13 chars);
 - ▲ **TypeTAXN** - Type of TAXN: '0' - BULSTAT; '1' - EGN; '2' - LNCH; '3' - service number;
- ▲ **RecName** - Reciever's name (up to 36 chars);
- ▲ **VATN** - VAT number of the client (up to 14 chars);
- ▲ **Addr1** - Client's address - line 1 (up to 36 chars);
- ▲ **Addr2** - Client's address - line 2 (up to 36 chars);

Answer(1)

- ▲ **'D'** - Client deleting;

Syntax: {Option}<SEP>{firstFIRM}<SEP>{lastFIRM}<SEP>

Mandatory parameters:

- ▲ **firstFIRM** - First client to delete (1...1000); If this parameter has value 'A', all clients will be deleted(lastFIRM must be empty).

Optional parameters:

- ▲ **lastFIRM** - last client to delete (1...1000). Default: {firstFIRM}; ;

Answer(1)

- ▲ **'R'** - Reading client data;

Syntax: {Option}<SEP>{FIRM}<SEP>

Mandatory parameters:

- ▲ **FIRM** - Client number (1...1000);

Answer(2)

- ▲ **'F'** - Returns data about the first found programmed client;

Syntax: {Option}<SEP>{FIRM}<SEP>

Optional parameters:

- ▲ **FIRM** - Client number (0...1000). Default: 0;

Answer(2)

- ▲ **'L'** - Returns data about the last found programmed client;

Syntax: {Option}<SEP>{FIRM}<SEP>

Optional parameters:

- ▲ **FIRM** - Client number (1...1000). Default: 1000;

Answer(2)

- ▲ **'N'** - Returns data for the next found programmed client;

Syntax: {Option}<SEP>

Note: The same command with option 'F' or 'L' must be executed first. This determines whether to get next('F') or previous ('L') client. Answer(2)

- ▲ **'T'** - Find a client by tax number;

Syntax: {Option}<SEP>{TAXN}<SEP>

- \b TAXN - Client's tax number (8...13 chars);

Answer(2)

- ▲ **'X'** - Find the first not programmed client;



Syntax: {Option}<SEP>{FIRM}<SEP>

Optional parameters:

▲ **FIRM** - Client number (0...1000). Default: 0;

Answer(4)

▲ **'x'** - Find the last not programmed client;

Syntax: {Option}<SEP>{FIRM}<SEP>

Optional parameters:

▲ **FIRM** - Client number (1...1000). Default: 1000;

Answer(4)

Answer(1): {ErrorCode}<SEP>

▲ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Answer(2):

{ErrorCode}<SEP>{FIRM}<SEP>{TAXN}<SEP>{TypeTAXN}<SEP>{VATN}<SEP>{Name}<SEP>{RecName}<SEP>{Addr1}<SEP>{Addr2}<SEP>

▲ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

▲ **FIRM** - Client number, index of record (1...1000);

▲ **Name** - Client's name (up to 36 chars);

▲ **TAXN** - Client's tax number (8...13 chars);

▲ **TypeTAXN** - Type of TAXN: '0' - BULSTAT; '1' - EGN; '2' - LNCH; '3' - service number;

▲ **RecName** - Reciever's name (up to 36 chars);

▲ **VATN** - VAT number of the client (up to 14 chars);

▲ **Addr1** - Client's address - line 1 (up to 36 chars);

▲ **Addr2** - Client's address - line 2 (up to 36 chars);

Answer(3): {ErrorCode}<SEP>{Total}<SEP>{Prog}<SEP>{NameLen}<SEP>

▲ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

▲ **Total** - Total count of the programmable clients (1000);

▲ **Prog** - Total count of the programmed clients (0...1000);

▲ **NameLen** - Maximum length of client name (36);

Answer(4): {ErrorCode}<SEP>{FIRM}<SEP>

▲ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

▲ **FIRM** - Client number (1...1000);

Command: 255 (FFh) - Programming

Parameters of the command: {Name}<SEP>{Index}<SEP>{Value}<SEP>

Mandatory parameters:

▲ **Name** - Variable name;

▲ **Device settings;**

▲ **FpComPort** - Number of COM port for communication with PC (0 - disable; 1 - COM1; 2 - COM2; 3 – USB; 4 - Bluetooth)

▲ **FpComBaudRate** - Baud rate of COM port for communication with PC (from 0 to 9)

▲ **ComPortDevice** - Assign peripheral device to COM port. (0 - none; 1 - Barcode; 2 - scale; 3 - external display;) Number of COM port is determined by "**Index**".

▲ **ComPortBaudRate** - Baud rate of COM port that has peripheral device assigned.(from 0 to 999999) Number of COM port is determined by "**Index**".

▲ **ComPortProtocol** - Protocol for communication with peripheral device assigned COM port. (from 0 to 9), if device is scale; Number of COM port is determined by "**Index**".

▲ **ECR parameters;**

▲ **EcrLogNumber** - Logical number in the workplace (from 1 to 9999);

▲ **EcrExtendedReceipt** - Type of the receipt(1 - extended(invoice), 0 - simplified);

▲ **EcrDoveriteli** - Work with constituents: 1-enable(in one receipt only one constituent), 0



- disable;
- ⤴ **EcrWithoutPasswords** - Work without passwords (1 - enable; 0 - disable);
- ⤴ **EcrAskForPassword** - Require password after each receipt (1 - enable; 0 - disable);
- ⤴ **EcrAskForVoidPassword** - Require password for void operations (1 - enable; 0 - disable);
- ⤴ **EcrConnectedOperReport** - When making Z-report, automatically make "Operator report" (1 - enable; 0 - disable);
- ⤴ **EcrConnectedDeptReport** - When making Z-report, automatically make "Report by Departments" (1 - enable; 0 - disable);
- ⤴ **EcrConnectedPluSalesReport** - When making Z-report, automatically make "Report by PLU with turnovers" (1 - enable; 0 - disable);
- ⤴ **EcrConnectedGroupsReport** - When making Z-report, automatically make "Group report" (1 - enable; 0 - disable);
- ⤴ **EcrConnectedCashReport** - When making Z-report, automatically make "Ecr report" (1 - enable; 0 - disable);
- ⤴ **EcrUserPeriodReports** - Periodic reports (1 - enable; 0 - disable) ;
- ⤴ **EcrPluDailyClearing** - When making Z-report, automatically clear PLU turnover (1 - enable; 0 - disable);
- ⤴ **EcrOnlyAdminCloseAccn** - Only administrator closes receipts (in restaurant mode), (1 - enable; 0 - disable);
- ⤴ **EcrSafeOpening** - Open drawer on every total (1 - enable; 0 - disable);
- ⤴ **EcrScaleBarMask** - Text up to 10 symbols. If second number of the weight barcode match any of the symbols in this string, barcode will be interpreted as normal barcode.
- ⤴ **EcrNumberBarcode** - Count of used barcodes for each programmed article (1...4);
- ⤴ **AutoPowerOff** - Minutes to automatically turn off ECR if it is idle. (0 - disable; from 1 minute to 15 minutes);
- ⤴ **BkLight_AutoOff** - Minutes to automatically turn off Backlight of the display if ECR is idle. (0 - disable; from 1 minute to 5 minutes);
- ⤴ **PlayGames** - Play games (1 - enable; 0 - disable);
- ⤴ **RegModeOnIdle** - Time to clear display after last receipt in milliseconds(1 - 2 147 483 647);
- ⤴ **Currencies**
 - ⤴ **CurrNameLocal** - Locam currency name(up to 3 chars);
 - ⤴ **CurrNameForeign** - Foreign currency name(up to 3 chars);
 - ⤴ **ExchangeRate** - Exchange rate(from 0 to 999999999, decimal point is before last five digits);
- ⤴ **Header of the receipt**
 - ⤴ **Header** - Text up to 42 symbols. Header line is determined by "**Index**", Index 0 is for line 1, Index 9 is for line 10;
- ⤴ **Footer of the receipt**
 - ⤴ **Footer** - Text up to 42 symbols. Footer line is determined by "**Index**". Index 0 is for line 1, Index 9 is for line 10;
- ⤴ **RecText of the receipt**
 - ⤴ **RecText** - Free text up to 40 symbols. Printed at the end of the receipt;
- ⤴ **Operators;**
 - ⤴ **OperName** - Name of operator. Text up to 20 symbols. Number of operator is determined by "**Index**";
 - ⤴ **OperPasw** - Password of operator. Text up to 8 symbols. (Require Service jumper) Number of operator is determined by "**Index**";
- ⤴ **Payments**
 - ⤴ **PayName** - Name of payment. Text up to 10 symbols. Number of payment is determined by "**Index**";
 - ⤴ **Payment_forbidden** - Forbid the payment (1- forbidden, 0 - not forbidden). Number of payment is determined by "**Index**";



- ^ **Department keys**
 - ^ **DPxx_PluCode** - Number of PLU assigned to department key. (0 - Key is department; from 1 to 99999 for assigning PLU). Number of key is determined by "**Index**";
- ^ **Keys discount and surcharge**
 - ^ **KeyNDB_value** - Value for value surcharge; Value is in cents. (from 0 to 999999999);
 - ^ **KeyNDB_percentage** - Percentage for percentage surcharge; Value is in hundredths (0.01) of a percent. (from 0 to 9999);
 - ^ **KeyOTS_value** - Value for value discount; Value is in cents. (from 0 to 999999999);
 - ^ **KeyOTS_percentage** - Percentage for percentage discount; Value is in hundredths (0.01) of a percent. (from 0 to 9999);
 - ^ **KeyNDB_forbidden** - Forbid the surcharge key (1- forbidden, 0 - not forbidden);
 - ^ **KeyOTS_forbidden** - Forbid the discount key (1- forbidden, 0 - not forbidden);
- ^ **Service**
 - ^ **ServPasw** -Password of the Service man. Text up to 8 symbols;(Require Service jumper)
 - ^ **ServMessage** - Message that will be printed when "ServDate" is reached, up to 42 symbols.(Require Service jumper) Message line is determined by "**Index**";
 - ^ **ServiceDate** - Service date(Format: DD-MM-YY HH:MM:SS);
- ^ **Receipt parameters;**
 - ^ **PrnQuality** - Contrast of Printing (from 0 to 20);
 - ^ **DublReceipts** - Print receipt duplicate (1 - enable, 0 -disable);
 - ^ **IntUseReceipts** - Number of internal receipts(from 0 to 9);
 - ^ **BarcodePrint** - Print PLU barcode in the receipt (1 - enable, 0 -disable);
 - ^ **LogoPrint** - Print logo in the receipt (1 - enable, 0 -disable);
 - ^ **DoveritelPrint** - Print depratment name in the beginning of the receipt (1 - enable, 0 - disable);
 - ^ **ForeignPrint** - Print total sum in foreign currency (1 - enable, 0 -disable, 2 - print exchange rate);
 - ^ **VatPrintEnable** - Print VAT rates in the receipt (1 - enable, 0 -disable);
- ^ **Modem and network**
 - ^ **SimPin** - PIN code of SIM card. Text up to 16 symbols;
 - ^ **LanMAC** - MAC address of the LAN controller(up to 12 chars);
 - ^ **DHCPenable** Enable use of DHCP (1 - enable, 0 -disable);
 - ^ **LAN_IP** - IP address when DHCP is disabled(up to 15 chars);
 - ^ **LAN_NetMask** - Net mask when DHCP is disabled(up to 15 chars);
 - ^ **LAN_Gateway** - Default gateway when DHCP is disabled(up to 15 chars);
 - ^ **LAN_PriDNS** - Primary DNS when DHCP is disabled(up to 15 chars);
 - ^ **LAN_SecDNS** - Second DNS when DHCP is disabled(up to 15 chars);
- ^ **Seller data**
 - ^ **Seller** - Seller data for invoice (up to 42 symbols);
- ^ **Variables for FM (Read Only)**
 - ^ **nZreport** - Number of current Z-report;
 - ^ **nReset** - Number of current memory failure;
 - ^ **nVatChanges** - Number of current VAT change;
 - ^ **nTAXnumberChanges** - Number of current TAX number changes (0 - not programmed; 1 - programmed);
 - ^ **valVat** - Current value of VAT. Number of VAT is determined by "**Index**";
 - ^ **IDnumber** - Serial number of the ECR;
 - ^ **FMnumber** - Number of FM;
 - ^ **TAXnumber** - TAX number;
- ^ **Variables for FM (Read and Write)**
 - ^ **TAXlabel** - TAX number label(up to 10 chars);
- ^ **Internal variables (Read Only)**
 - ^ **nBon** - global number of receipts;
 - ^ **nInvoice** - Number of invoices;



- ^ **InvoiceRangeBeg** - Start of the invoice range(from 0 to 9999999999);
- ^ **InvoiceRangeEnd** - End of the invoice range(from 0 to 9999999999);
- ^ **EJNumber** - Number of current EJ;
- ^ **Item Groups**
 - ^ **ItemGroups_name** - Name of item group. Text up to 32 symbols. Number of item group is determined by "**Index**";
- ^ **Department registers**
 - ^ **Dept_name** - Name of department. Text up to 32 symbols. Number of department is determined by "**Index**";
 - ^ **Dept_ext_name** - Extended name of department. Text up to 32 symbols. Number of department is determined by "**Index**";
 - ^ **Dept_price** - Programmed price of department(from 0 to 9999999999). Number of department is determined by "**Index**";
 - ^ **Dept_vat** - VAT group of department(from 1 to 8). Number of department is determined by "**Index**";

Optional parameters:

^ **Index** - Used for index if variable is array. For variable that is not array can be left blank. Default: 0;
Note: For example: Header[], Index 0 refer to line 1. Index 9 refer to line 10.

^ **Value** - If this parameter is blank ECR will return current value (*Answer(2)*). If the value is set, then ECR will program this value (*Answer(1)*);

Answer(1): {ErrorCode}<SEP>

^ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

Answer(2): {ErrorCode}<SEP>{VarValue}<SEP>

^ **ErrorCode** - Indicates an error code. If command passed, **ErrorCode** is 0;

- **VarValue** - Current value of the variable;